


# APLIKATIF LOGIKA & ALGORITMA

(dengan C++, C# & Java)

**Fauziah, S.Kom., M.M.S.I.**

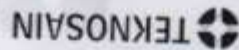
 **TEKNOSAIN**



Semua informasi tentang buku ini, silahkan scan QR Code di cover belakang buku ini

Tajuk Entrti Utama: Fauziah  
 Aplikasiif Logika dan Algoritma dengan C++, C#, dan Java/Fauziah  
 - Edisi Pertama, Cet. Ke-1. - Yogyakarta: Teknosain, 2016  
 xviii + 226 hlm.; 25 cm  
 Bibliografi: 217 - 219; Gloss.: 221 - 225  
 ISBN : 978-602-6324-10-8  
 E-ISBN : 978-602-6324-11-5  
 1. Komputer Pemrograman  
 I. Judul  
 005.1

Ruko Jambusari 7A Yogyakarta 55283  
 Telp: 0274-889398; Fax: 0274-889057; E-mail: info@teknosain.com  
 Hak Cipta dilindungi undang-undang. Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apa pun, secara elektronik maupun mekanis, termasuk memfotokopi, merekam, atau dengan teknik perekaman lainnya, tanpa izin tertulis dari penerbit.



Hak Cipta © 2016 pada penulis

oleh Fauziah, S.Kom, MMSI

Aplikatif Logika dan Algoritma dengan C++, C#, dan Java





## KATA PENGANTAR

**M**engapa kalian perlu belajar tentang Bahasa Pemrograman yaitu (C++, C# dan Java)? Tujuan pembelajaran menurut Kurikulum yang digunakan untuk Mahasiswa Tingkat awal dewasa ini antara lain; untuk memahami konsep pemrograman dasar baik pada mata kuliah Algoritma Pemrograman maupun Dasar-dasar Pemrograman, mengembangkan keterampilan menggunakan logika dan berfikir secara efisien, praktis serta sederhana dengan tujuan dapat mengembangkan sikap kritis, kreatif, apresiatif, dan mandiri dalam penggunaan bahasa pemrograman dasar, serta dapat mengembangkan pola pikir dan mampu menyelesaikan setiap permasalahan yang berkaitan dengan bahasa pemrograman di tingkat lanjut.

Buku yang penulis susun ini berisi kumpulan contoh-contoh program dan keterangan dari tiap logika yang ada, sehingga membantu mahasiswa untuk dapat memahami semua perintah, instruksi (*statement*) yang digunakan dalam bahasa pemrograman baik dalam bentuk *User Defined Words* (Kata-kata yang dideklarasikan *user* yaitu; pengguna, atau *programmer*) dan *Keyword* (Kata-kata yang dideklarasikan oleh bahasa pemrograman secara baku atau standar).

Oleh karena itu, Buku yang berisi *Aplikatif Logika & Algoritma* (dengan bahasa pemrograman C++, C# dan Java) perlu diperkenalkan, dipraktikkan, dan dikuasai oleh Mahasiswa sedini mungkin agar

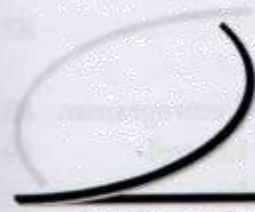
Mahasiswa memiliki bekal untuk dapat membantu memecahkan masalah yang berkaitan dengan bahasa pemrograman. Mengapa demikian? Karena contoh-contoh yang disajikan dalam buku ini banyak membantu Mahasiswa untuk dapat belajar secara cepat, singkat dan sederhana.

Buku *Aplikatif Logika & Algoritma* dengan bahasa pemrograman ini disajikan dalam upaya mengajar, mendidik, dan melatih Mahasiswa untuk belajar menggunakan logika dan berfikir secara efisien sehingga Mahasiswa mampu dan diharapkan memiliki pengetahuan bagaimana cara menyelesaikan masalah dan dapat menuangkannya dengan menggunakan bahasa pemrograman yang terus berkembang dengan cepat dari waktu ke waktu.

Penulis

Fauziah, S.Kom, M.M.SI.





## DAFTAR ISI

<b>KATA PENGANTAR</b>	v
<b>DAFTAR ISI</b>	vii
<b>DAFTAR GAMBAR</b>	ix
<b>BAB 1 PENGENALAN STRUKTUR DASAR C++, C# dan JAVA</b>	1
1.1 Sejarah Bahasa Pemrograman C++, C# dan Java	2
1.2 Struktur Program Bahasa C++	4
1.3 Struktur Program C# dan Java	6
1.4 Statement <i>Input/Output</i> pada C++, C# dan Java	12
1.5 Contoh Program Sederhana dengan Menggunakan Bahasa Pemograman C++, C# dan Java	13
<b>BAB 2 VARIABEL DAN TIPE DATA PADA PROGRAM C++, C# DAN JAVA</b>	19
2.1 Pengertian Variabel, Deklarasi dan Contoh	20
2.2 Pengertian Tipe Data, Deklarasi dan Contoh	29
<b>BAB 3 OPERATOR PADA BAHASA PEMROGRAMAN C++, C# DAN JAVA</b>	43
3.1 Pengertian Operator pada Bahasa Pemrograman	44
3.2 Macam-macam Operator pada Bahasa Pemrograman	44

<b>BAB 4</b>	<b>STRUKTUR KONTROL PADA BAHASA PEMROGRAMAN</b>	<b>87</b>
4.1	Pengertian Struktur Kontrol pada Bahasa Pemrograman	88
4.2	Contoh Program Menggunakan Struktur Kontrol pada Bahasa Pemrograman C++, C# dan Java	89
4.3	<i>Flowchart</i> Struktur Kontrol Logika <i>If</i>	93
4.4	Struktur Kontrol dengan Logika <i>Switch</i>	113
<b>BAB 5</b>	<b>ARRAY PADA BAHASA PEMROGRAMAN</b>	<b>173</b>
5.1	Pengertian <i>Array</i> dan Macamnya	174
5.2	Contoh Program dengan <i>Array</i>	175
<b>BAB 6</b>	<b>PROCEDURE PADA BAHASA PEMROGRAMAN</b>	<b>197</b>
	<b>DAFTAR PUSTAKA</b>	<b>217</b>

-0000-



87

88

89

93

113

173

174

175

197

217

## DAFTAR GAMBAR

Gambar 1.1	Area Kerja Turbo C++	4
Gambar 1.2	Area Kerja C# dengan <i>Microsoft Visual Studio 2008</i>	6
Gambar 1.3	Tampilan <i>New Project</i>	8
Gambar 1.4	Tampilan Nama <i>Project</i> yang Dibuat	8
Gambar 1.5	Area Kerja Java	10
Gambar 1.6	Tampilan <i>New Project</i>	11
Gambar 1.7	Tampilan Program Awal Java	11
Gambar 1.8	Tampilan <i>Output</i> dari Program C++	14
Gambar 1.9	Hasil <i>Output</i> Mencari Nilai Luas	15
Gambar 1.10	Tampilan <i>Output</i> Program Java	16
Gambar 2.1	Tampilan Hasil dengan Deklarasi Variabel	22
Gambar 2.2	Tampilan <i>Output</i> untuk Contoh Variabel Konstan	23
Gambar 2.3	Tampilan <i>Output</i> Menggunakan <i>Variable Input</i> pada Visual C#	24
Gambar 2.4	Tampilan <i>Output</i> Menggunakan <i>Variabel Input</i> dan Konstan	26
Gambar 2.5	Tampilan <i>Output</i> dengan Variabel Konstan pada Java	27
Gambar 2.6	Tampilan <i>Output</i> Menggunakan <i>Variabel Input</i> pada Java	28
Gambar 2.7	Tampilan <i>Output</i> Menggunakan <i>Variabel Input</i> pada Java	28
Gambar 2.8	Tampilan <i>Output</i> Menggunakan Tipe Data Angka	31
Gambar 2.9	Tampilan <i>Output</i> Menggunakan Tipe Data <i>Character</i>	32



Gambar 2.10	Tampilan <i>Output</i> Menggunakan Kombinasi Tipe Data <i>Integer</i> dan <i>Character</i>	33
Gambar 2.11	Tampilan <i>Output</i> Menggunakan Tipe Data Angka	34
Gambar 2.12	Tampilan <i>Output</i> Menggunakan Tipe Data Angka dan Huruf	36
Gambar 2.13	Tampilan <i>Output</i> Menggunakan Tipe Data Angka	37
Gambar 2.14	Tampilan <i>Output</i> Menggunakan Tipe Data <i>Character</i>	38
Gambar 2.15	Hasil <i>Output</i> dengan Kombinasi Tipe Data Angka dan Huruf	40
Gambar 3.1	Tampilan <i>Output</i> Menggunakan <i>Operator Assignment</i>	48
Gambar 3.2	Tampilan <i>Output</i> Menggunakan <i>Operator Aritmatika</i>	50
Gambar 3.3	Tampilan <i>Output</i> Menggunakan <i>Operator Aritmatika</i> dengan <i>Data Input</i>	51
Gambar 3.4	Menggunakan <i>Operator Aritmatika</i> dengan <i>Data Input</i>	52
Gambar 3.5	Tampilan <i>Output</i> Menggunakan <i>Operator Increment</i> dan <i>Decrement</i>	53
Gambar 3.6	Tampilan <i>Output Operator Increment &amp; Decrement</i> dengan <i>Variabel Input</i>	55
Gambar 3.7	Tampilan <i>Output Operator Logika And</i> dan <i>OR</i> Menggunakan <i>Variabel Konstan</i>	56
Gambar 3.8	Tampilan <i>Output Logika AND</i> dan <i>OR</i> Menggunakan <i>Data Inputan</i>	57
Gambar 3.9	Tampilan <i>Output Operator Aritmatika</i> pada Program <i>C#</i> Menggunakan <i>Data Konstan</i>	58
Gambar 3.10	Tampilan <i>Output Operator Aritmatika</i> Menggunakan <i>Data Inputan</i>	59
Gambar 3.11	Tampilan <i>Output Logika Operator And</i> dan <i>Or</i> Menggunakan <i>Variabel Konstan</i>	61
Gambar 3.12	Tampilan <i>Output Operator Logika And</i> dan <i>Or</i> Menggunakan <i>Data Input</i>	62
Gambar 3.13	Tampilan <i>Output Operator Logika Increment Decrement</i> Menggunakan <i>Data Input</i>	63
Gambar 3.14	Tampilan <i>Output Operator Logika Increment Decrement</i> Menggunakan <i>Data Konstan</i>	65



Tipe Data	33	Gambar 3.15	Tampilan <i>Output</i> Menggunakan Kombinasi Operator Aritmatika	70
Logika	34	Gambar 3.16	Tampilan <i>Output</i> Menggunakan Operator Aritmatika Variabel Konstan	71
Logika	36	Gambar 3.17	Tampilan <i>Output</i> Operator Matematika Menggunakan Data <i>Inputan</i>	73
Logika	37	Gambar 3.18	Tampilan <i>Output</i> Operator <i>Increment</i> & <i>Decrement</i> Menggunakan Data Konstan	74
Character	38	Gambar 3.19	Tampilan <i>Output</i> Menggunakan Operator <i>Increment</i> & <i>Decrement</i> dengan Data <i>Input</i>	76
Logika	40	Gambar 3.20	Tampilan <i>Output</i> Menggunakan Operator Logika <i>And</i> , <i>Or</i> , <i>Xor</i> & <i>Negasi</i> dengan Data Konstan	78
Segment	48	Gambar 3.21	Tampilan <i>Output</i> Menggunakan Operator <i>And</i> , <i>Or</i> , <i>XOR</i> dan <i>Negasi</i> dengan Variabel <i>Input</i>	81
Matematika	50	Gambar 4.1	Tampilan <i>Output</i> Menggunakan Logika <i>If</i> dengan Data Konstan	90
Matematika	51	Gambar 4.2	Tampilan <i>Output</i> Menggunakan Logika <i>If</i> dengan Data Konstan	91
Data Input	52	Gambar 4.3	Tampilan <i>Output</i> dengan Logika <i>If</i> Menggunakan Variabel <i>Input</i> /Data <i>Input</i>	93
Segment	53	Gambar 4.4	Tampilan <i>Output</i> dengan Logika <i>If</i> Menggunakan Variabel <i>Input</i> /Data <i>Input</i>	93
Segment	55	Gambar 4.4	Tampilan <i>Output</i> Menggunakan Logika <i>IF</i>	98
IF	56	Gambar 4.5	Tampilan <i>Output</i> Menggunakan Logika <i>IF</i>	98
Menggunakan	57	Gambar 4.6	Tampilan <i>Output</i> Menggunakan Program C# dengan Logika <i>IF</i> Data <i>Input</i>	101
Program	58	Gambar 4.7	Tampilan <i>Output</i> Menggunakan Program C# dengan Logika <i>IF</i> Data <i>Input</i>	101
Menggunakan	59	Gambar 4.8	Tampilan <i>Output</i> Menggunakan Program C# dengan Logika <i>IF</i> Data <i>Input</i>	102
	61	Gambar 4.9	Tampilan <i>Output</i> Menggunakan Program C# dengan Logika <i>IF</i> Data <i>Input</i>	102
	62	Gambar 4.10	Tampilan <i>Output</i> Menggunakan Program C# dengan Logika <i>IF</i> Data <i>Input</i>	103
	63			
	65			



Gambar 4.11	Tampilan <i>Output</i> Menggunakan Variabel Konstan	106
Gambar 4.12	Tampilan <i>Output</i> Menggunakan Variabel Inputan	109
Gambar 4.13	Tampilan <i>Output</i> dengan Logika <i>IF</i> Menggunakan Variabel Konstan	111
Gambar 4.14	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel <i>Input</i>	116
Gambar 4.15	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel <i>Input</i>	116
Gambar 4.16	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel <i>Input</i>	117
Gambar 4.17	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel <i>Input</i>	117
Gambar 4.18	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Konstan	119
Gambar 4.19	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Konstan	121
Gambar 4.20	Tampilan <i>Output</i> Data Konstan dengan Pilihan	123
Gambar 4.21	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Konstan	125
Gambar 4.22	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel <i>Input</i>	128
Gambar 4.23	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel <i>Input</i>	128
Gambar 4.24	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel <i>Input</i>	129
Gambar 4.25	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel <i>Input</i>	129
Gambar 4.26	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel <i>Input</i>	130
Gambar 4.27	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel <i>Input</i>	130
Gambar 4.25	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel <i>Input</i>	131



Gambar 4.26	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Input	131
Gambar 4.27	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Input	132
Gambar 4.28	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Input	132
Gambar 4.29	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Input	133
Gambar 4.30	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Konstan dengan Kota Tujuan="1"	135
Gambar 4.31	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Konstan dengan kota Tujuan="2"	136
Gambar 4.32	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Konstan dengan Kota Tujuan="3"	136
Gambar 4.33	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Konstan dengan kota tujuan="4"	137
Gambar 4.34	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Konstan dengan Kota Tujuan="5"	137
Gambar 4.35	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Konstan dengan Kota Tujuan="6"	138
Gambar 4.36	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Konstan dengan Kota Tujuan="7"	138
Gambar 4.37	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Konstan dengan Kota Tujuan="8"	139
Gambar 4.38	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Konstan dengan Kota Tujuan="9"	139
Gambar 4.39	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Konstan dengan Kota Tujuan="10"	140
Gambar 4.40	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Konstan dengan Kota Tujuan="11" [Tidak Tersedia pada Pilihan Kondisi]	140
Gambar 4.41	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Konstan dengan int angka=1	142



Gambar 4.42	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Konstan dengan int angka=2	142
Gambar 4.43	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Konstan dengan int angka=3	142
Gambar 4.44	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Konstan dengan int angka=4	142
Gambar 4.45	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Konstan dengan int angka=5	143
Gambar 4.46	Tampilan <i>Output</i> Dengan Logika <i>Switch</i> Menggunakan Variabel Konstan dengan int angka=6	143
Gambar 4.47	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Konstan dengan int angka=7	143
Gambar 4.48	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel Konstan dengan int angka=8	143
Gambar 4.49	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel <i>Input</i>	146
Gambar 4.50	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel <i>Input</i>	147
Gambar 4.51	Tampilan <i>Output</i> dengan Logika <i>Switch</i> Menggunakan Variabel <i>Input</i>	147
Gambar 4.52	Tampilan <i>Output</i> dengan Logika <i>For</i> untuk Menampilkan Nilai 1-10 dengan Kondisi Perulangan $i \leq 10$	150
Gambar 4.53	Tampilan <i>Output</i> dengan Logika <i>For</i> untuk Menampilkan nilai 1-9 dengan Kondisi Perulangan $i \leq 10$ dan Pertambahan 2	151
Gambar 4.54	Tampilan <i>Output</i> dengan Logika <i>For</i> untuk Menampilkan nilai 0-5, Nilai Genap dan Nilai Ganjil dengan Kondisi $x \leq 5$	153
Gambar 4.55	Tampilan <i>Output</i> dengan Logika <i>For Decrement</i> atau Nilai Penurunan $x--$ , $y--$ dan $z--$	155
Gambar 4.56	Tampilan <i>Output</i> dengan Logika <i>For</i> untuk Menampilkan nilai 0-5, Nilai Genap dan Nilai Ganjil dengan Kondisi $x \leq 5$	156



Gambar 4.57	Tampilan <i>Output</i> dengan Logika <i>For</i> untuk Menampilkan nilai 0-9	157
Gambar 4.58	Tampilan <i>Output</i> dengan Logika <i>For</i> untuk Menampilkan nilai 0-9	158
Gambar 4.59	Tampilan <i>Output</i> dengan Logika <i>For</i> untuk Menampilkan nilai 10-1 dengan Perulangan <i>Decrement</i> [Perulangan Menurun]	159
Gambar 4.60	Tampilan <i>Output</i> dengan Logika <i>For</i> untuk Menampilkan Nilai 10-1 dengan Perulangan <i>Decrement</i> [Perulangan Menurun]	160
Gambar 4.61	Tampilan <i>Output</i> dengan Logika <i>For</i> untuk Menampilkan Nilai 10-1 dengan Perulangan <i>Decrement</i> [Perulangan Menurun]	161
Gambar 4.62	Tampilan <i>Output</i> dengan Logika <i>Foreach</i> untuk Menampilkan Nama dengan Data Tetap	162
Gambar 4.63	Tampilan <i>Output</i> dengan Logika <i>For</i> Menggunakan Java untuk Mencetak Nama dari Indeks 0 Sampai 9 [Ada 10× Perulangan]	163
Gambar 4.64	Tampilan <i>Output</i> dengan Logika <i>For</i> Menggunakan Java untuk Mencetak Nilai Baris dan Nilai Kolom Mulai dari Indeks ke 0 Sampai Indeks ke 4 [Ada 5 Elemen Data Baris dan Kolom]	163
Gambar 4.65	Tampilan <i>Output</i> dengan Logika <i>While</i> Menggunakan Java untuk Mencetak Nama Saya adalah: Tsarwan Kasyfi Sebanyak 10×	165
Gambar 4.66	Tampilan <i>Output</i> dengan Logika <i>While</i> Menggunakan Java untuk Mencetak Nama Saya adalah: Tsarwan Kasyfi Sebanyak 10×	166
Gambar 4.67	Tampilan <i>Output</i> dengan Logika <i>While</i> Bersarang Menggunakan Java untuk mencetak Nilai Segitiga	167
Gambar 4.68	Tampilan <i>Output</i> dengan Logika <i>While</i> Bersarang Menggunakan Java untuk Mencetak Nilai Segitiga	168



Gambar 4.69	Tampilan <i>Output</i> dengan Logika <i>While</i> Bersarang Menggunakan Java untuk Mencetak Nilai Segitiga dengan Logika <i>Increment</i> dan <i>Decrement</i>	169
Gambar 4.70	Tampilan Hasil dengan Logika <i>For</i> Sesuai <i>Flowchart</i> 4.1	171
Gambar 4.71	Logika <i>For</i> dengan Nilai <i>Decrement</i> [-]	172
Gambar 5.1	Tampilan <i>Array</i> 1 Dimensi untuk Mencetak Nilai Sesuai Input dan Menyimpan Nilai Indeks Sesuai Logika <i>For</i> yang Digunakan	176
Gambar 5.2	Tampilan <i>Array</i> 1 Dimensi Menggunakan <i>Array</i> Karakter	178
Gambar 5.3	Tampilan <i>Array</i> 1 Dimensi [Angka dan Karakter] untuk Mencetak Nilai Sesuai Input dan Menyimpan Nilai Indeks Sesuai Logika <i>For</i> yang Digunakan	179
Gambar 5.4	Tampilan <i>Array</i> 1 Dimensi untuk Mencetak Nama Kota dengan 5 Buah Elemen Data Kota dan Panjang Maksimum dari Masing-masing Kota adalah 10 Karakter	180
Gambar 5.5	Tampilan <i>Array</i> 1 Dimensi untuk Mencetak Mahasiswa dan Nilai dengan <i>Array</i> 1 Dimensi	183
Gambar 5.6	Tampilan Hasil Sistem Akademik Berdasarkan Nilai dari Data Mahasiswa dengan Kombinasi <i>Array</i> 1 Dimensi dan <i>Array</i>	184
Gambar 5.7	Tampilan Hasil <i>Array</i> 1 Dimensi Menggunakan Data Karakter	185
Gambar 5.8	Tampilan Hasil <i>Array</i> 2 Dimensi Menggunakan Data Matrik dengan Ordo 2 Baris dan 2 Kolom	187
Gambar 5.9	Tampilan Hasil <i>Array</i> 2 Dimensi Menggunakan Data Nama dengan Pendeklarasian Nama untuk 2 Baris dan 2 Kolom	189
Gambar 5.10	Tampilan Hasil <i>Array</i> 2 Dimensi Menggunakan Data matriks dengan Ordo $10 \times 10$ dan Hasil yang Tercetak Merupakan Perkalian dari Isi Elemen Baris dan Kolom	192
Gambar 5.11	Tampilan Hasil <i>Array</i> 1 Dimensi Menggunakan Data Konstan Terdiri dari 5 Buah Elemen Data	193

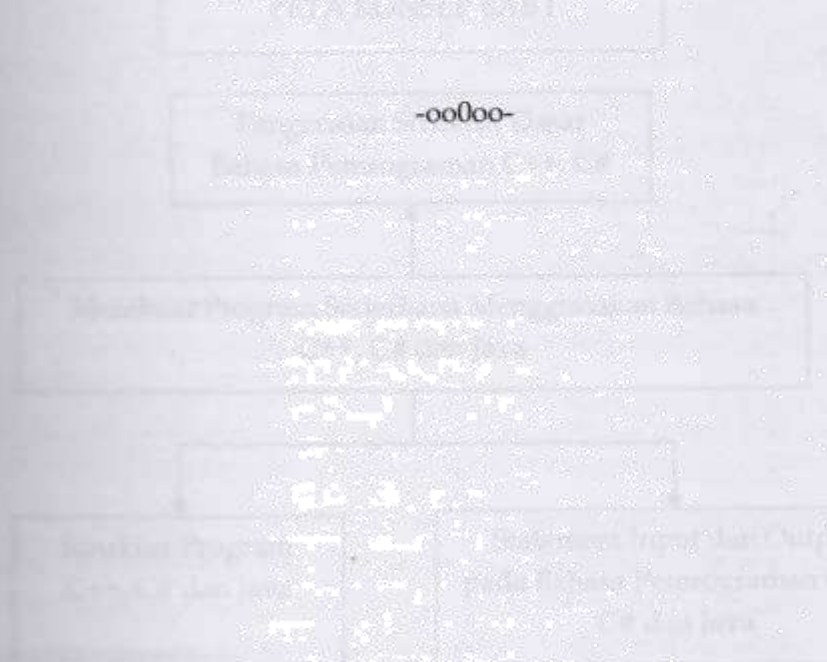


Logika dan Algoritma

169  
171  
172  
176  
178  
179  
180  
183  
184  
185  
187  
189  
192  
193

Daftar Gambar

- Gambar 5.12 Tampilan Hasil Array 2 Dimensi Menggunakan Data konstan Terdiri dari 2 Elemen
- Gambar 6.1 Tampilan Output Menggunakan Procedure
- Gambar 6.2 Hasil Output menggunakan Procedure dengan Beberapa Nama Paragraph Procedure
- Gambar 6.3 Tampilan Output Procedure dengan Java
- Gambar 6.4 Tampilan Procedure and Function
- Gambar 6.5 Tampilan Contoh Program Mencari Luas dan Keliling
- Gambar 6.6 Tampilan Contoh Program Mencari Luas dan Volume Tabung
- Gambar 6.7 Tampilan Contoh Program Mencari Luas dan Keliling Layang-layang
- Gambar 6.8 Tampilan Contoh Program Mencari Luas dan Volume Kubus
- Gambar 6.9 Tampilan Saat Dijalankan dengan Menekan Ctrl +





Sumber: [defenseindustrydaily.com](http://defenseindustrydaily.com)

# BAB I

## PENDAHULUAN

### Pengenalan Struktur C++, C# dan JAVA

#### **Tujuan Pembelajaran**

Setelah mempelajari bab ini Mahasiswa diharapkan mampu:

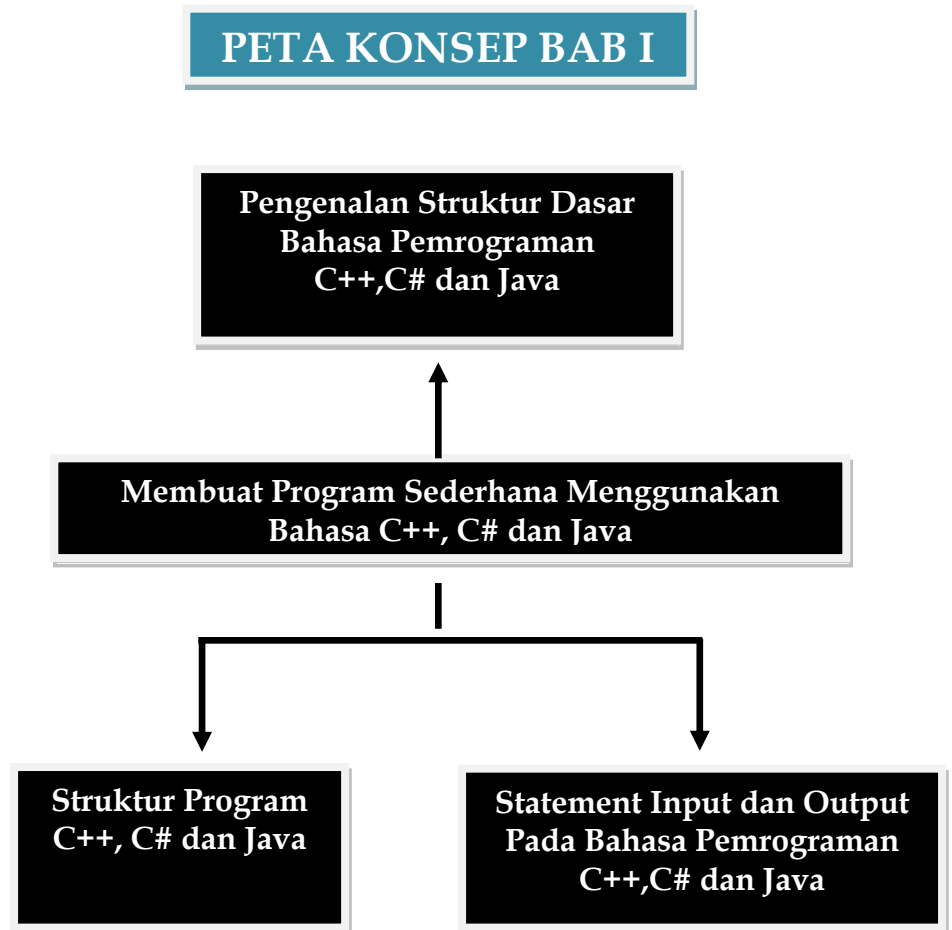
Menjelaskan struktur dasar bahasa pemrograman C++, C# dan Java serta mampu membuat program sederhana dengan logika yang singkat dan jelas

Dewasa ini Perkembangan bahasa pemrograman sudah sangat berkembang cepat dari waktu ke waktu sehingga kitapun harus



memahami berbagai macam karakteristik dari setiap struktur bahasa pemrograman yang kita gunakan agar kita dapat membedakan masing – masing struktur yang ada pada bahasa pemrograman tersebut, karena setiap bahasa pemrograman memiliki struktur, karakteristik dan fungsi masing – masing, namun setiap bahasa pemrograman mempunyai 2 istilah yang sering kita dapatkan yaitu “User Defined Words ( Kata-Kata yang dideklarasikan oleh pengguna/programmer) dan Reserved word / Keyword ( Kata Kunci yang ada pada setiap bahasa pemrograman yang kita gunakan), dan untuk contohnya dapat kita lihat pada contoh program yang akan kita bahas di bab ini.

**Kata Kunci :** User Defined Words, Reserved Word, Keyword,  
Bahasa Pemrograman C++, C#, Java





# 1.1 SEJARAH BAHASA PEMROGRAMAN

## BAHASA PEMROGRAMAN C++

Awal perkembangan bahasa pemrograman C++ antara lain :

1. Pada tahun 1978, Dennis M. Ritchie dan Brian W.Kernighan dari laboratorium yang terkenal saat itu yaitu *AT&T Laboratories* berhasil mengembangkan bahasa pemrograman yang awalnya adalah bahasa B (yang diciptakan oleh Ken Thompson yang merupakan perkembangan dari bahasa pemrograman "*Basic Combined*" yang diciptakan oleh Martin Richard) sehingga menjadi bahasa C
2. Kemudian pada tahun 1980 seorang Bjarne Stroustrup dari *AT&T Bell Laboratories* dan pada tahun 1985 kemudian dikembangkanlah bahasa C++, dan sejak

# SEJARAH BAHASA PEMROGRAMAN

## BAHASA PEMROGRAMAN C#

Awal perkembangan bahasa pemrograman C# antara lain :

1. Pada tahun 2000 *Microsoft* telah meluncurkan sebuah bahasa pemrograman yang merupakan pengembangan dari beberapa bahasa pemrograman sebelumnya
2. Bahasa Pemrograman yang dikembangkan adalah bahasa C# (C Sharp) *Programming* yang dikembangkan oleh sebuah pengembang dari tim *Microsoft* dipimpin oleh Anders Hejlsberg dan Scott Wiltamuth.
3. Bahasa Pemrograman C# memiliki bentuk umum yang hampir sama dengan bahasa C, C++ dan Java



# SEJARAH BAHASA PEMROGRAMAN

## BAHASA PEMROGRAMAN JAVA

Awal perkembangan bahasa pemrograman JAVA antara lain

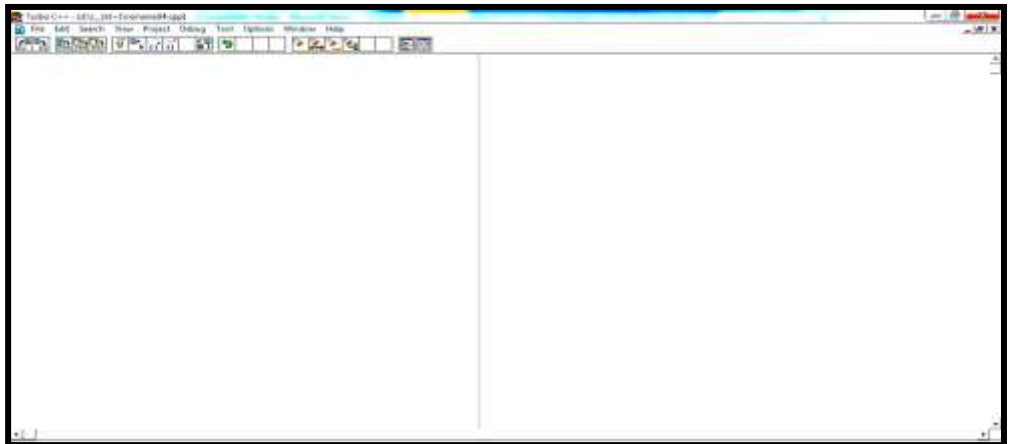
5. Bahasa Java merupakan jenis bahasa pemrograman *high level language* ( bahasa pemrograman tingkat tinggi) yang mudah dipahami dan dipelajari dan termasuk jenis bahasa pemrograman yang berorientasi pada objek (*Object Oriented Programming*)
6. Bahasa Pemrograman yang dikembangkan Pada tahun 1991 oleh James Gosling dan kawan – kawan (*Sun Micro System*)
7. Java merupakan jenis bahasa pemrograman yang sangat *familier* dan banyak digunakan untuk berbagai jenis aplikasi dan *game*
8. Arsitektur dari teknologi iava meliputi: *Java Virtual*



### Area Kerja Turbo C++

Berikut ini adalah area kerja turbo C++, langkah awal yang dilakukan untuk membuka area kerja turbo C++ adalah :

1. Klik Menu start
2. Pilih All Program kemudian pilih Turbo C++, maka akan muncul tampilan awal area kerja turbo C++ sebagai berikut:



Gambar 1.1 Area Kerja Turbo C++



# 1.2 Struktur Program Bahasa C++

## STRUKTUR PROGRAM BAHASA C++

Struktur Program Bahasa C++ adalah sebagai berikut :

1. Komentar : Merupakan baris program yang tidak dieksekusi oleh *compiler* dan mesin, komentar hanya sebagai keterangan atau penjelasan dari kode program (program yang sedang kita ketikkan), komentar menggunakan *//*
2. *Header* merupakan bagian yang paling atas biasanya berisi inputoutput Stream.header yang fungsinya untuk memerintahkan *compiler* untuk menyertakan perintah *input/output* dan diawali dengan menggunakan *symbol #*.  
Misalnya : `include<iostream.h>`

4. Struktur berikutnya adalah kurung buka kurawal menandakan bahwa kita dapat mulai mengetikkan *variable*, kode program yang isinya adalah *statement* (pernyataan – pernyataan, dimana awal pembuka dari pernyataan kita menggunakan Tanda petik di awal dan diakhir isi pernyataan ) dan kemudian di tutup dengan kurung kurawal akhir. Untuk penulisan *statement* pernyataan kita menggunakan *statement COUT* (yaitu character output) “menampilkan semua instruksi/pernyataan yang kita ketikkan ke layar monitor”.

```
{
```

```
    Berisi statement (pernyataan-pernyataan)
```

```
}
```

5. *Return* yang merupakan sebuah pernyataan atau *statement* yang dapat menghentikan program dan mengembalikan nilai ke dalam fungsi *main* dengan nilai awal nol (0).

Berikut ini adalah contoh penulisan struktur Bahasa Pemrograman C++

```
// Tampilan Pertamaku
```

```
... . . . . .
```



# Keterangan Kode Program

Dari kode program “Tampilan Pertamaku”

1. `// Tampilan pertamaku` : Merupakan komentar atau keterangan yang berisi penjelasan Program yang sedang kita buat.
2. `#include <iostream.h>` : merupakan *header* yang menyertakan *input* dan *output* untuk hasil tampilan *COUT* dan *CIN* ( *Character Output* dan *Character Input* )
3. `int main()` merupakan program utama yang dikenali oleh *compiler* dan akan mengeksekusi program.
4. `{` merupakan kurung kurawal buka sebagai penanda awal kode program atau *statement*, kemudian diikuti dengan perintah *COUT* ( *Character Output* yang artinya kita akan mencetak hasil di layar monitor, menggunakan tanda petik diawal dan diakhir misalnya :

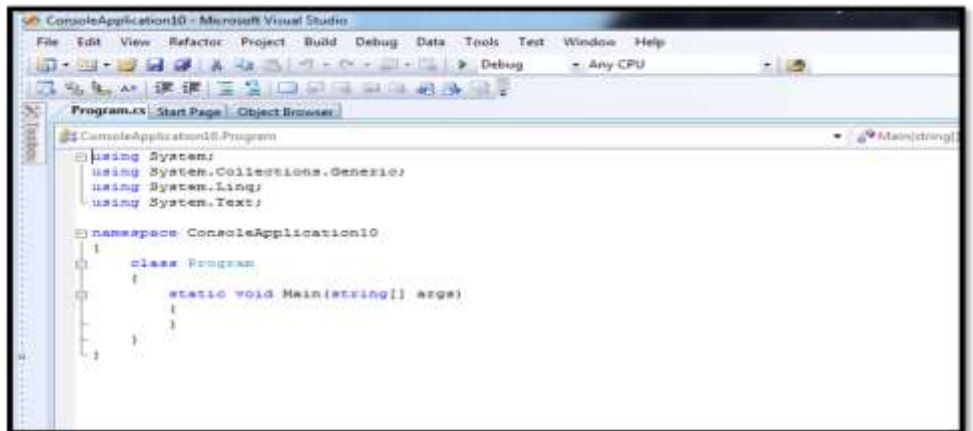
```
cout << "Hallo Nama Saya Tsarwan Kasyfillah Nasution";  
cout << "Ini Tampilan Pertamaku";
```

# Area Kerja C#

## Area Kerja C#

Berikut ini adalah area kerja C#, langkah awal yang dilakukan untuk membuka area kerja C# adalah :

1. Klik Menu start
2. Pilih All Program kemudian pilih Microsoft Visual Studio 2008 , maka akan muncul tampilan awal area kerja sebagai berikut:



Gambar 1.2 Area Kerja C# dengan Microsoft Visual Studio 2008



# Struktur Program Bahasa C#

## STRUKTUR PROGRAM BAHASA C#

Struktur Program Bahasa C# adalah sebagai berikut :

1. **Komentar** : Merupakan baris program yang tidak dieksekusi oleh *compiler* dan mesin, komentar hanya sebagai keterangan atau penjelasan dari kode program (program yang sedang kita ketikkan), komentar menggunakan (*//*)
2. **Keyword Using** merupakan bagian yang paling atas digunakan untuk memberitahukan compiler class apa yang kita gunakan dan sudah default standar secara otomatis akan muncul ketika kita membuat program pada Bahasa pemrograman C#.
3. **Keyword namespace** merupakan bagian yang terdapat pada struktur C# dan digunakan untuk mendeklarasikan

## STRUKTUR PROGRAM BAHASA C#

5. Struktur berikutnya adalah *static void Main(string[] args,* merupakan struktur dari bahasa c# merupakan *method* utama yang akan dijalankan pada *class* yang sudah kita deklarasikan.

6. Kemudian struktur berikutnya adalah kurung kurawal buka yang menandakan bahwa kita akan mulai mengetikkan semua *statement* atau perintah dengan menggunakan statement *Console.WriteLine* dan ditutup dengan kurung kurawal

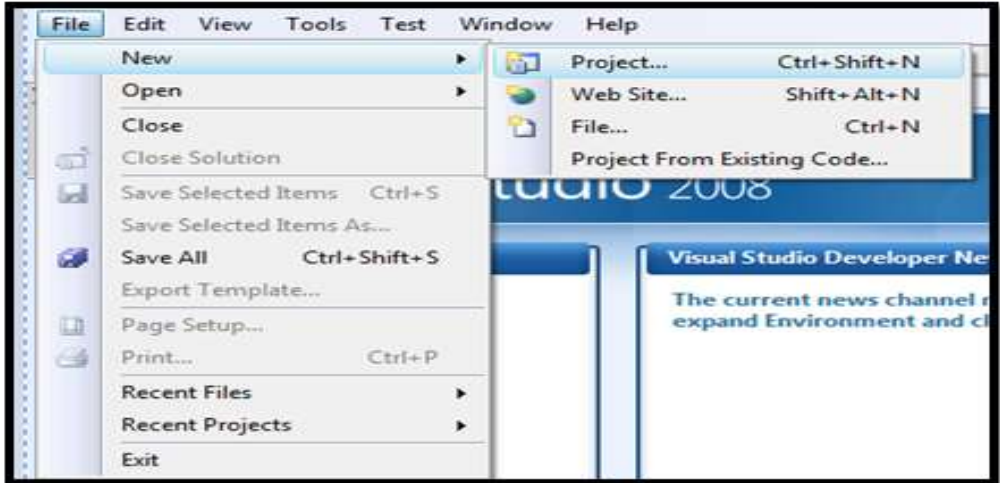
Berikut ini adalah contoh struktur Pada Bahasa Pemrograman C#

Langkah yang dilakukan adalah :

1. Klik start
2. Pilih Microsoft Visual Studio 2008
3. Klik File, pilih New
4. Pilih Project
5. Kemudian Pilih Project Type Visual C#
6. Kemudian Pilih Console Application

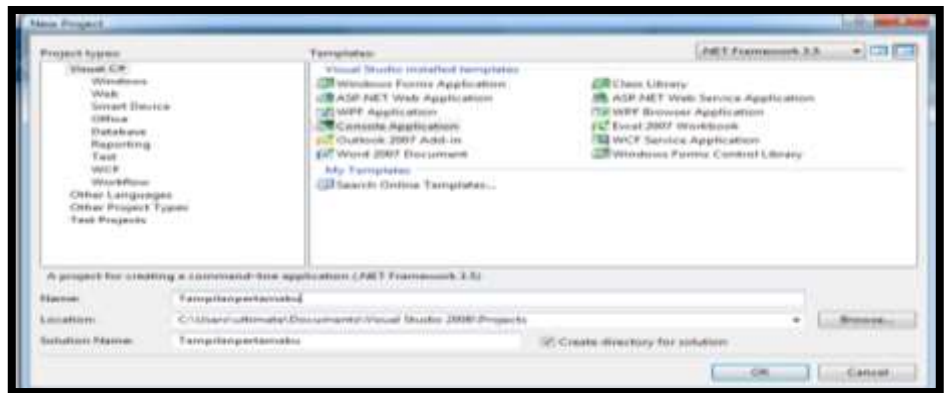


- Tampilan New Project



Gambar 1.3 Tampilan New Project

- Tampilan pilihan C# Console Application dan mengisikan nama project yang akan dibuat



Gambar 1.4 Tampilan Nama Project yang dibuat

Berikut ini adalah contoh penulisan struktur Bahasa Pemrograman C#:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace belajar
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hallo Nama Saya : Tsarwan
                               Kasyfillah Nasution");
            Console.WriteLine("Saya Baru Belajar Bahasa
                               Pemrograman C#");
            Console.WriteLine("Inilah tampilan Pertama
                               Saya, Terima Kasih");
        }
    }
}
```

## Keterangan Kode Program

1. `using System`

“Merupakan bagian paling atas yang menyatakan *compiler* tentang *class* apa yang digunakan”

2. `class Program`.”Merupakan nama kelas yang sedang dibuat”

3. `static void Main(string[] args)`.” Merupakan *method*



# Keterangan Kode Program

4. Isi statement yang diawali dengan { (kurung kurawal) dan ditutup dengan kurung kurawal }, seperti berikut ini:

```
{  
    Console.WriteLine("Hallo Nama Saya : Tsarwan  
                        Kasyfillah Nasution");  
    Console.WriteLine("Saya Baru Belajar Bahasa  
                        Pemrograman C#");  
    Console.WriteLine("Inilah tampilan Pertama  
                        Saya, Terima Kasih");  
}
```

Menggunakan statement `Console.WriteLine` yang artinya menampilkan teks pada layar monitor.

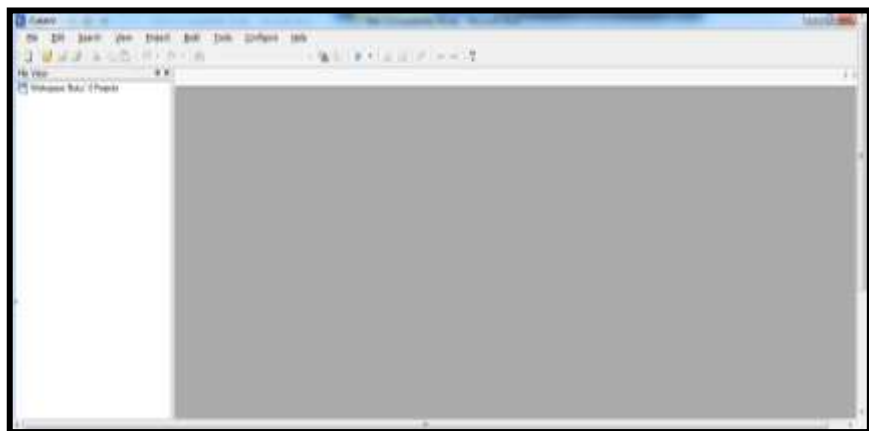
**Catatan : Untuk Menjalankan Kode Program Pada C# tekan**



### Area Kerja Java

Berikut ini adalah area kerja Java, langkah awal yang dilakukan untuk membuka area kerja Java adalah :

1. Klik Menu start
2. Pilih JCreator
3. Maka akan muncul tampilan awal area kerja sebagai berikut:



Gambar 1.5 Area Kerja Java

# Struktur Program Bahasa Java

## STRUKTUR PROGRAM BAHASA JAVA

1. Nama *Class* yaitu menunjukkan nama kelas yang sudah kita buat atau definisikan pada saat awal membuat *project* baru.  
Misalnya : “*public class* BUKU1”
2. *static void Main(string[] args):*” Merupakan *method* yang akan dijalankan pada *class* yang sudah dideklarasikan
3. Kemudian struktur berikutnya adalah kurung kurawal buka yang menandakan bahwa kita akan mulai mengetikkan semua *statement* atau perintah dengan menggunakan *statement System.out.println* dan ditutup dengan kurung kurawal

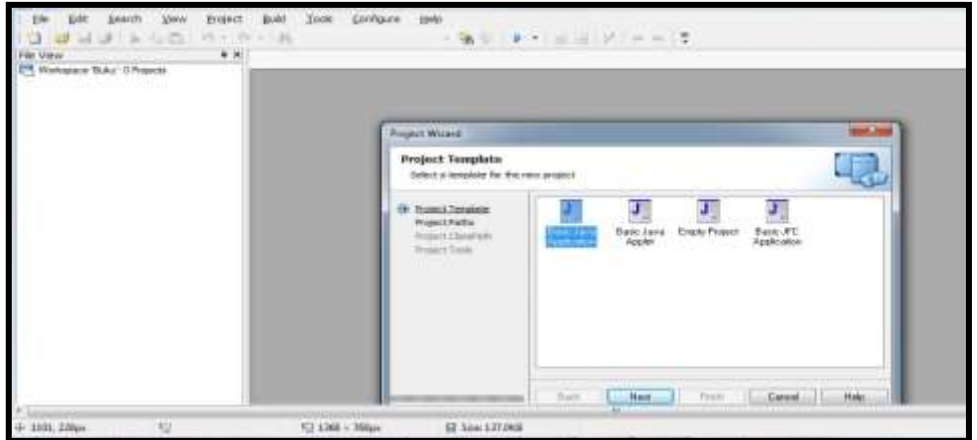
Berikut ini adalah contoh struktur Pada Bahasa Pemrograman Java

Langkah yang dilakukan adalah :

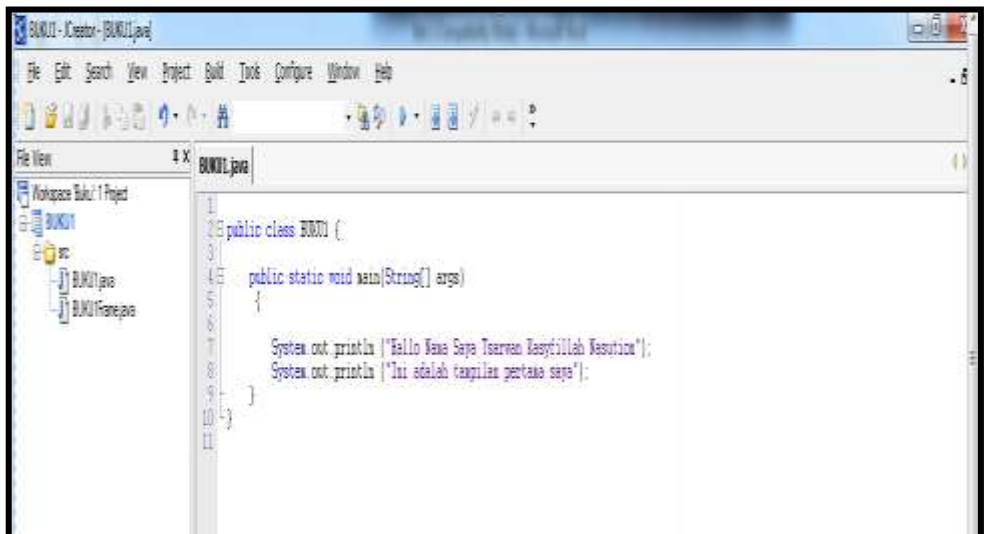


## Tampilan Pada Bahasa Pemrograman Java

- Tampilan *New Project*



Gambar 1.6 Tampilan *New Project*



Gambar 1.7 Tampilan Program awal Java

Berikut ini adalah contoh penulisan struktur Bahasa Pemrograman *Java*:

```
public class BUKU1 {  
  
    public static void main(String[] args)  
  
    {  
  
        System.out.println ("Hallo Nama Saya Tsarwan  
Kasyfillah Nasution");  
  
        System.out.println ("Ini adalah tampilan pertama  
saya");  
  
    }  
}
```

## Keterangan Kode Program

1. *public class*: “Merupakan nama class yang kita deklarasikan pada saat membuat nama project diawal”

# Keterangan Kode Program

3. Isi *statement* yang diawali dengan { (kurung kurawal) dan ditutup dengan kurung kurawal }, seperti berikut ini:

```
{  
  
    System.out.println ("Hallo Nama Saya Tsarwan  
  
    Kasyfillah Nasution");  
  
    System.out.println ("Ini adalah tampilan pertama  
  
    saya");  
  
}  
}
```

Menggunakan *statement* `System.out.println` yang artinya menampilkan teks pada layar monitor.

**Catatan :** Untuk Menjalankan Program Pada Java tekan



# Statement Input/Output

## Statement *Output* dan *Input*

### 1. Statement *Output* dan *Input* Pada Bahasa Pemrograman C++

- Statement *Output* Pada C++

“Merupakan jenis perintah atau *instruksi* yang digunakan untuk menampilkan hasil di layar monitor” *Statement* yang digunakan adalah *COUT (Character Output)*

- Statement *Input* Pada C++

“Merupakan jenis perintah atau *instruksi* yang digunakan untuk menginputkan segala sesuatu

## 2. Statement *Output* dan *Input* Pada Bahasa Pemrograman C#

- Statement *Output* Pada C#

“Merupakan jenis perintah atau *instruksi* yang digunakan untuk menampilkan hasil di layar monitor” *Statement* yang digunakan adalah `Console.WriteLine`

- Statement Input Pada C++

“Merupakan jenis perintah atau *instruksi* yang

## 3. Statement *Output* dan *Input* Pada Bahasa Pemrograman Java

- Statement *Output* Pada Java

“Merupakan jenis perintah atau *instruksi* yang digunakan untuk menampilkan hasil di layar monitor” *Statement* yang digunakan adalah `System.out.println`

- Statement *Input* Pada Java

“Merupakan jenis perintah atau *instruksi* yang

# 1.4 Contoh Program Sederhana

## Program Sederhana Menggunakan Bahasa Pemrograman

### C++, C# & Java

- Program C++

```
#include <iostream.h>

int main ()
{
    // inisialisasi variabel :
    int nilai1, nilai2;
    int hasil;
    // proses :
    cout<<"Masukkan nilai1: ";

    cin>>nilai1;

    cout<<"Masukkan nilai2: ";

    cin>>nilai2;

    // mencetak hasil penjumlahan :
    hasil = nilai1+nilai2;

    cout << hasil;

    // menghentikan program :
    return 0;
```



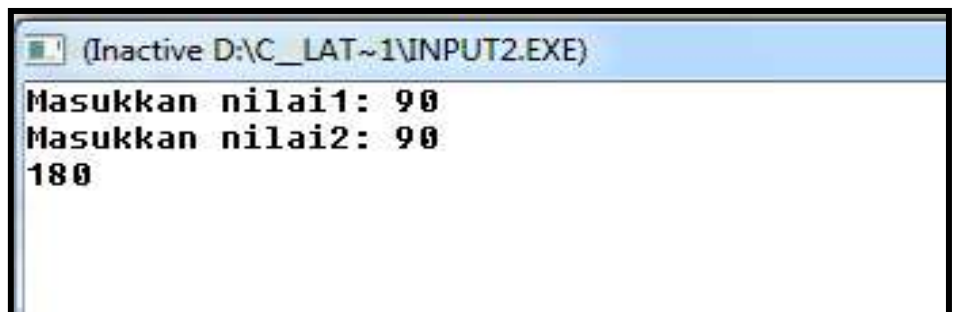
# Keterangan Kode Program

1. Program di atas mempunyai 3 buah *variable* yaitu nilai1, nilai2 dan hasil
2. Menggunakan perintah *input* dan *output* yaitu dengan *COUT* dan *CIN*
3. Proses yang dilakukan adalah dengan rumus :

$$\text{hasil} = \text{nilai1} + \text{nilai2}$$

**Note** : Jangan lupa setiap akhir baris **statement** kita menggunakan **semicolon** (symbol titik koma ;)

4. Untuk menjalankan program kita menekan tombol pada



```
(Inactive D:\C_LAT~1\INPUT2.EXE)
Masukkan nilai1: 90
Masukkan nilai2: 90
180
```

Gambar 1.8 Tampilan *Output* dari Program C++

- **Program C#**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace integer
{
    class Program
    {
        static void Main(string[] args)
        {

            int pj; //variabel unt menampung nilai panjang
            int lb; //variabel unt menampung nilai lebar
            int L; //variabel yg menampung hasil Luas
            Console.Write("Inputkan nilai panjang :");
            //konversi dari tipe string ke integer
            pj = Convert.ToInt32(Console.ReadLine());
            Console.Write("Inputkan nilai lebar :");
            lb = Convert.ToInt32(Console.ReadLine());
            L = pj*lb;
            Console.WriteLine("Jadi Hasil Luasnya adalah" + L);

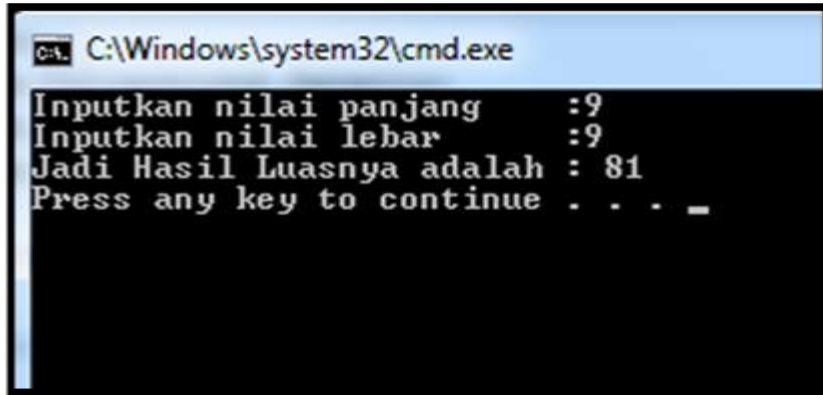
        }
    }
}
```

# Keterangan Kode Program

1. Program di atas mempunyai 3 buah *variable* yaitu panjang, lebar dan Luas
2. Menggunakan perintah *output* yaitu dengan *statement* :  
`Console.Write("Masukkan panjang :");`
3. Menggunakan perintah *input* yaitu dengan *statement* :  
`pj = Convert.ToInt32(Console.ReadLine());`
4. Menggunakan *statement Convert* yaitu perintah yang digunakan untuk mengubah nilai *string* menjadi *integer* (bilangan bulat)
5. Proses yang dilakukan adalah:  $L = pj * lb$ ;

**Note** : Jangan lupa setiap akhir baris *statement* kita menggunakan *semicolon* (*symbol*/titik koma ;)





```
C:\Windows\system32\cmd.exe
Inputkan nilai panjang      :9
Inputkan nilai lebar        :9
Jadi Hasil Luasnya adalah  : 81
Press any key to continue . . . _
```

Gambar 1.9 Hasil *output* Mencari Nilai Luas

- **Program Java**

```
//nama file berkas : bukujava.java

import java.io.*;

public class bukujava
{

    public static void main(String[] args) throws Exception
    {

        // menginputkan nama

        DataInputStream nama=new DataInputStream (System.in);

        System.out.print("Inputkan nama      : ");

        String snama=nama.readLine();

        //menginputkan nilai1

        DataInputStream nilai1=new DataInputStream (System.in);

        System.out.print("Inputkan nilai1 : ");

        String snilai1=nilai1.readLine();

        //menginputkan nilai2

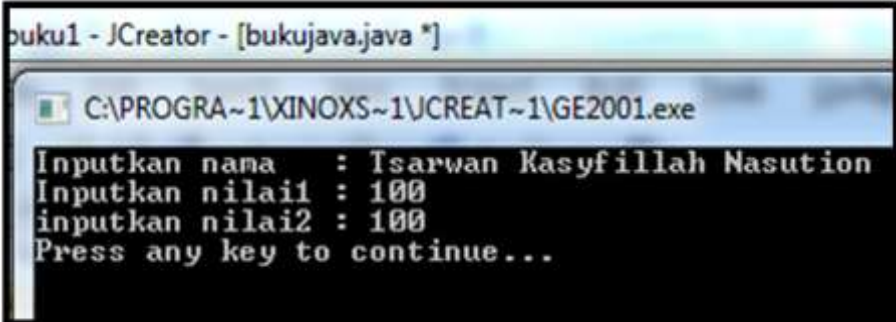
        DataInputStream nilai2=new DataInputStream (System.in);

        System.out.print("inputkan nilai2 : ");
```

# Keterangan Kode Program

1. Program di atas mempunyai 3 buah *variable* yaitu nama, nilai1 dan nilai2
2. Menggunakan perintah *output* yaitu dengan *statement* :  
`System.out.print("Inputkan nama :");`
3. Menggunakan perintah *input* yaitu dengan *statement* `String snama=nama.readLine();`

**Note** : Jangan lupa setiap akhir baris *statement* kita menggunakan *semicolon* (*symbol*/titik koma ;)



```
buku1 - JCreator - [bukujava.java *]  
C:\PROGRA~1\XINOX~1\JCREAT~1\GE2001.exe  
Inputkan nama : Tsarwan Kasyfillah Nasution  
Inputkan nilai1 : 100  
inputkan nilai2 : 100  
Press any key to continue...
```

Gambar 1.10 Tampilan *Output* Program Java

# PENGAYAAN

## Latihan Soal

Jawablah pertanyaan di bawah ini dengan baik dan benar

1. Tuliskan sejarah singkat Bahasa Pemrograman C++, C# dan Java !
2. Sebutkan kelebihan dan kelemahan dari bahasa pemrograman C++, C# dan Java !
3. Tuliskan jenis perintah input dan output yang digunakan pada bahasa C++, C# dan Java !
4. Buatlah masing – masing contoh program menggunakan C++, C# dan Java !
5. Analisis setiap kode program yang dibuat mulai



# GLOSARIUM

## GLOSARIUM BAB 9

1. Bahasa Pemrograman : Kumpulan instruksi atau perintah yang digunakan oleh *programmer* atau *user*
2. *High Level Language* : Jenis bahasa pemrograman tingkat tinggi yang *programmer* buat dan pahami
3. Contoh Bahasa Pemrograman *high Level Language* : Bahasa Pemrograman C++, C# dan Java
4. *Statement Output* : Jenis perintah atau instruksi yang digunakan untuk mencetak sesuatu nilai pada layar monitor

pada Java (`System.out.print`)

6. *Statement input* : Jenis perintah atau instruksi yang digunakan untuk menginputkan sesuatu nilai melalui *keyboard*
7. Contoh *statement input* : Pada C++ (*COUT*), pada C# (`Console.ReadLine()`), Pada Java (`readLine()`).
8. *Convert* : digunakan untuk mengkonversikan nilai pada bahasa pemrograman C# yaitu nilai *string* ke *integer*
9. *User Defined Words* : Kata-kata yang didefinisikan atau dideklarasikan oleh pengguna, *programmer* ataupun *user*
10. *Contoh User Defined Words* misalnya pendeklarasian nama *variable*, nilai1, nilai2, nama, hasil, panjang, lebar dan luas misalnya
11. *Keyword* : Kata kunci yang sudah baku atau standar yang ada pada bahasa pemrograman
12. Contoh *Keyword* : `include`, `cin`, `cout`, `Concole.Write`, `Console.readline`,



Sumber: [defenseindustrydaily.com](http://defenseindustrydaily.com)

## BAB II

### VARIABEL &

### TIPE DATA

(MENGUNAKAN

VARIABEL

DAN TIPE DATA

PADA BAHASA

PEMROGRAMAN

C++, C# dan JAVA)

#### **Tujuan Pembelajaran**

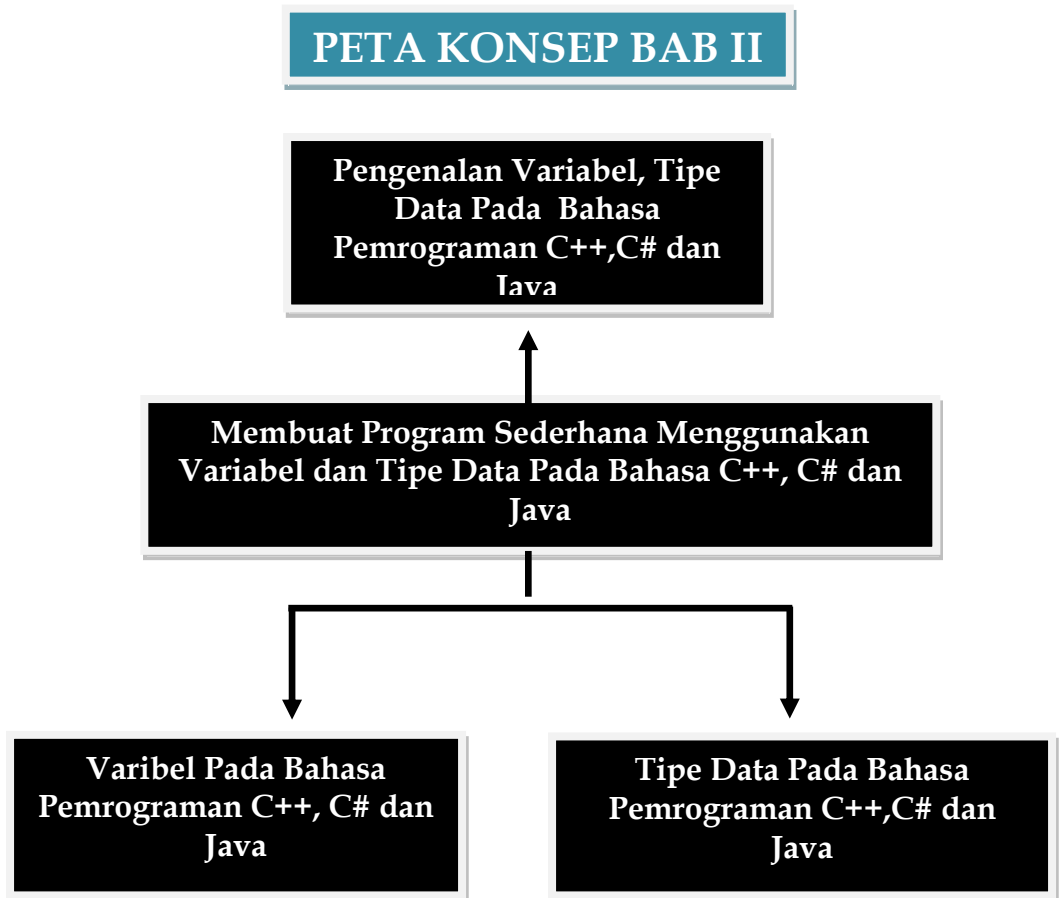
Setelah mempelajari bab ini Mahasiswa diharapkan mampu:

Menjelaskan perbedaan berbagai jenis variabel dan tipe data yang digunakan pada bahasa pemrograman C++, C# dan Java serta mampu mengimplementasikan variabel yang ada ke dalam bahasa pemrograman C++, C# dan Java.

Varibel dan tipe data merupakan komponen atau bagian terpenting yang ada dalam bahasa pemrograman, karena *variable*

dan tipe data dideklarasikan secara bersamaan, didalam sebuah kode blok program

**Kata Kunci** : *variable*, tipe data, blok program



## 2.1 Pengertian Variabel, Deklarasi dan Contoh

- **Pengertian Variabel**

*Variabel* merupakan pengenal (*identifier*) dari suatu data yang kita deklarasikan pada bahasa pemrograman, *variable* dapat dikatakan juga sebagai wadah (penyimpanan data/alokasi *storage*) yang kita deklarasikan dengan tujuan untuk memudahkan *programmer* memanggil atau menggunakan kembali data yang ada pada kode program yang dibuatnya.

Saat mendeklarasikan *variable* ada aturan yang harus dipenuhi diantaranya adalah :

- a. Penamaan *variable* menggunakan kata – kata yang dideklarasikan oleh *user*, oleh karena itu *variable* merupakan *user defined words*



c. Penamaan *variable* diawali dengan huruf, minimal 1 buah huruf diawal.

d. Penamaan *Variabel* tidak boleh menggunakan spasi

e. Jika ada 2 kata maka gunakan penghubung *hyphen ( underscore | \_ )*.

Pendeklarasian <i>Variabel</i> Benar	Pendeklarasian <i>Variabel</i> Salah
<ul style="list-style-type: none"><li>• Nilai1 (diawali dengan huruf)</li></ul>	<ul style="list-style-type: none"><li>• 1Nilai (diawali dengan angka)</li></ul>
<ul style="list-style-type: none"><li>• Nilai1(Tidak menggunakan spasi)</li></ul>	<ul style="list-style-type: none"><li>• Nilai 1(menggunakan spasi)</li></ul>
<ul style="list-style-type: none"><li>• IF1(menggunakan <i>user defined words</i> karena</li></ul>	<ul style="list-style-type: none"><li>• IF(menggunakan <i>keyword</i>)</li></ul>

ada angka1)

Macam tipe *variable* antara lain :

- a. *Variabel* konstan :” merupakan jenis *variable* yang sudah memiliki nilai atau *value*.”
- b. *Variabel inputan* :”merupakan jenis *variable* yang tidak memiliki nilai atau *value*, jika kita ingin memberikan nilai maka ada *statement* tambahan yang digunakan yaitu dengan perintah *input variable* dan kita dapat menginputkannya melalui *keyboard*.”

- Contoh *Variabel* konstan

```
Int a= 5;
```

```
String nama="Tsarwan Kasyfillah";
```

- Contoh *Variabel Input*

```
Int a;
```

```
#include <iostream.h>

int main ()

{

//deklarasi variabel dengan input

int x1, y1, z1,a, b, c;

cout<<"Inputkan nilai x1: ";

cin>>x1;

cout<<"Inputkan nilai y1: ";

cin>>y1;

cout<<"Inputkan nilai z1: ";

cin>>z1;

//proses

a = x1+y1*z1;

b = x1*y1-z1;

c = x1+z1-y1;

// mencetak hasil :

cout<<"Jadi hasil nilai a adalah : "<<a<<"\n";

cout<<"Jadi hasil nilai b adalah : "<<b<<"\n";

cout<<"Jadi hasil nilai c adalah : "<<c<<"\n".
```

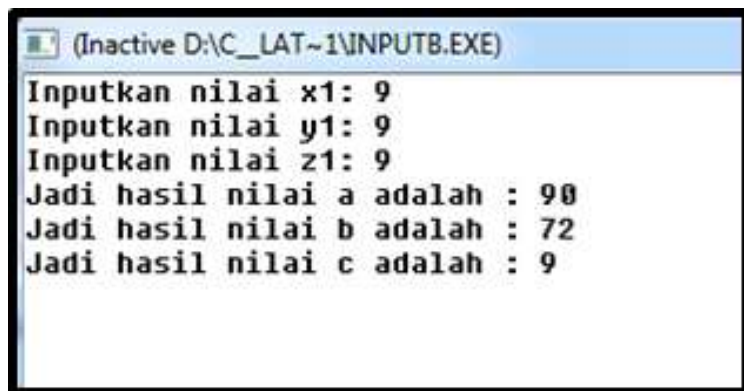
- *Variabel* yang digunakan pada program tersebut adalah:

Int x1,y1,z1,a,b,c : merupakan *variable* yang digunakan dengan nama x1,y1,z1,a,b dan c

*Variabel input* = x1,y1 dan z1

*Variabel output* = a,b dan c

- Proses yang ada adalah penjumlahan, pengurangan dan perkalian



```
(Inactive D:\C_LAT~1\INPUTB.EXE)
Inputkan nilai x1: 9
Inputkan nilai y1: 9
Inputkan nilai z1: 9
Jadi hasil nilai a adalah : 90
Jadi hasil nilai b adalah : 72
Jadi hasil nilai c adalah : 9
```

Gambar 2,1 Tampilan Hasil dengan deklarasi variabel

- Contoh program dengan variable konstan (tetap) pada program C++

```
#include <iostream.h>
```

```
int main ()
```

```
{
```

```
// inialisasi variabel :
```

```
int a, b;
```

```
int hasil;
```

```
// proses :
```

```
a = 5;
```

```
b = 2;
```

```
a = a + 1;
```

```
hasil = a - b;
```

```
// cetak hasilnya :
```

```
cout << hasil;
```



- *Variabel* yang digunakan pada program tersebut adalah:

Int a,b,hasil : merupakan *variable* yang digunakan dengan nama a,b dan hasil

Jenis *Variabel* : *konstan* yaitu terlihat nilai:

a=5;

b=2;

*Variabel output* = hasil

- Proses yang ada adalah penjumlahan dan pengurangan

Maka *output* dari program dengan *variable* yang dideklarasikan menggunakan *variable konstan* adalah :

.....



Gambar 2.2 Tampilan *Output* Untuk Contoh Variabel Konstan

- Contoh program dengan variable inputan pada program C#

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace COBA2
{
    class Program
    {
        static void Main(string[] args)
        {
            // deklarasi variabel

            string nama, kelas, hobi;
            Console.WriteLine("Selamat belajar C#");
            Console.WriteLine(" BIODATAKU      ");

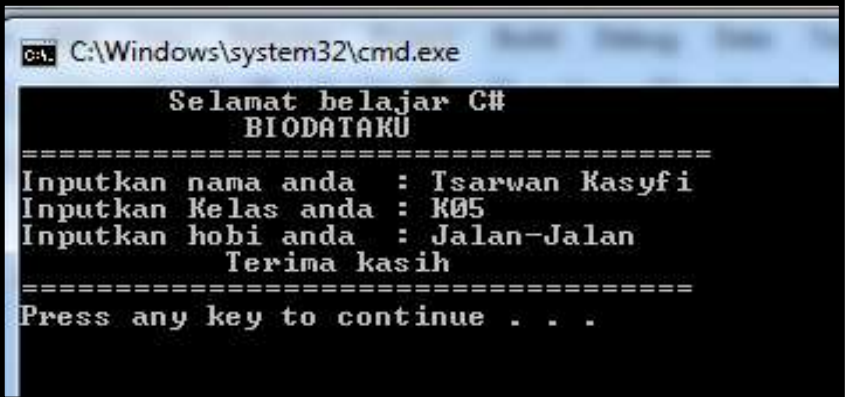
            Console.WriteLine("=====");
            Console.WriteLine("Inputkan nama anda :");
            nama = Console.ReadLine();
            Console.WriteLine("Inputkan Kelas anda :");
            kelas = Console.ReadLine();
            Console.WriteLine("Inputkan hobi anda :");
            hobi = Console.ReadLine();
            Console.WriteLine("Terima kasih ");
            Console.WriteLine("=====");

        }
    }
}
```

- *Variabel* yang digunakan pada program tersebut adalah:

```
string nama, kelas, hobi;
```

Maka *output* dari program dengan *variable* yang dideklarasikan menggunakan *variable* konstan adalah :



```
C:\Windows\system32\cmd.exe

Selamat belajar C#
BIODATAKU
=====
Inputkan nama anda : Tsarwan Kasyfi
Inputkan Kelas anda : K05
Inputkan hobi anda : Jalan-Jalan
Terima kasih
=====
Press any key to continue . . .
```

Gambar 2.3 Tampilan *Output* menggunakan *variable input* pada *Visual C#*

- Contoh program dengan variable konstan (tetap) dan *inputan* pada program C#

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace konstan
{
    class Program
    {
        static void Main(string[] args)
        {
            // variabel inputan
            int jml,hrg,total;
            long tlp;
            string plgn, nb;

            // variabel konstan
            hrg=3500000;

            // Statement input dan output

            Console.WriteLine("=====\n\n");
            Console.WriteLine("Rekapitulasi Pembelian Produk PT Untung Selalu\n\n");
            Console.WriteLine("=====\n\n");
            Console.Write("Inputkan Nama Produk           : ");
            nb = Console.ReadLine();
            Console.Write("Inputkan Jumlah Produk           : ");
            jml=Convert.ToInt32(Console.ReadLine());
            Console.Write("\nHarga Produk dengan Nilai Konstan adalah: ");
            Console.Write("{0}",hrg);
            Console.Write("\nInputkan Nama Pelanggan           : ");
            plgn=Console.ReadLine();
            Console.Write("\nInputkan No.Telp Pelanggan       : ");
            tlp=Convert.ToInt64(Console.ReadLine());
            total = (hrg*jml);
            Console.WriteLine("Total Pembayaran Adalah           : " + total);
            Console.WriteLine("\n Terima Kasih Atas Kunjungan Anda ");
            Console.WriteLine("\n                Sampai Jumpa Lagi di PT Untung Selalu ");
            Console.WriteLine("\n=====");
            Console.ReadLine();
        }
    }
}
```

- *Variabel* yang digunakan pada program tersebut adalah:

```
// variabel inputan
int jml,hrg,total;
long tlp;
string plgn, nb;
// variabel konstan
hrg=3500000;
```

Dengan *Variabel inputan* adalah : jml, hrg, total, tlp, plg dan nb dan *variabel* konstan adalah : hrg.

Maka *output* dari program dengan *variable* yang dideklarasikan menggunakan *variable* konstan dan *inputan* adalah :

```
C:\Windows\system32\cmd.exe

=====

Rekapitulasi Pembelian Produk PT Untung Selalu

=====

Inputkan Nama Produk           : Hardisk
Inputkan Jumlah Produk         : 3
Harga Produk dengan Nilai Konstan adalah: 3500000
Inputkan Nama Pelanggan       : Fauziah, S.Kom, MMSI
Inputkan No.Telp Pelanggan    : 081383955321
Total Pembayaran Adalah       : 10500000

Terima Kasih Atas Kunjungan Anda

Sampai Jumpa Lagi di PT Untung Selalu

=====
```



Gambar 2.4 Tampilan Output Menggunakan Variabel Input dan Konstan

- Contoh program dengan variable konstan (tetap) pada program Java

```
//PROGRAM DENGAN DATA KONSTAN
```

```
class konstan {
```

```
public static void main(String[] args) {
```

```
// Deklarasi Variabel dengan Nilai Konstan
```

```
byte Na = 4; // Nilai a adalah = 1
```

```
short Nb = 4; // Nilai b adalah = 2
```

```
int Nc = 1, Nd; // Nilai c adalah = 1& Hasil disimpan Pada Variabel d
```

```
Nd = Na + Nb + Nc; // Proses Perhitungan Nd dengan menjumlahkan nilai a,b dan c
```

```
System.out.println("Maka HasilMenggunakan Variabel Konstan Adalah = " + Nd);
```

```
// Output dan menyimpannya Pada Nilai d
```

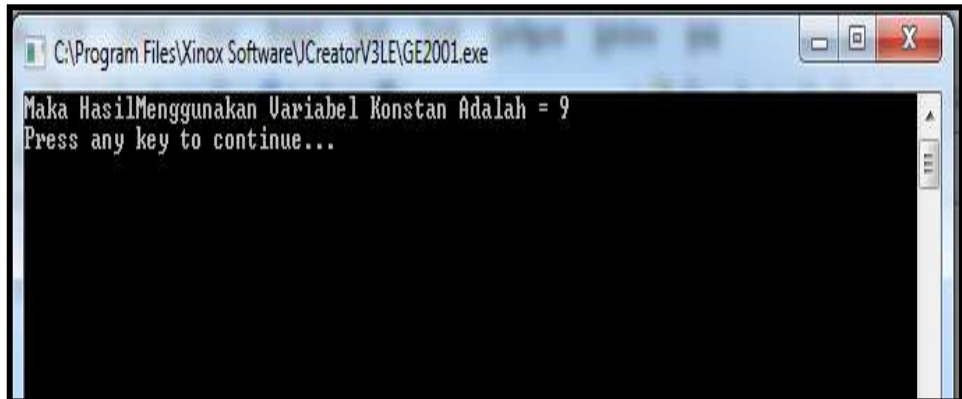
```
// dan memanggil kembali setelah proses
```

```
}
```

```
}
```

*Variabel* yang digunakan pada program tersebut adalah:

*Byte* Na, *Short* Nb dan *Int* Nc, Nd dengan nilai masing-masing variable



Gambar 2.5 Tampilan *Output* Dengan Variabel Konstan Pada Java

- Contoh program dengan variable *inputan* pada program Java

```
//nama file berkas : kondisi2.java
import java.io.*;
public class InputBaru
{

    public static void main(String[] args) throws Exception
    {
        // menginputkan nama
        DataInputStream nama=new DataInputStream (System.in);
        System.out.print("Inputkan nama  :");
        String snama=nama.readLine();

        //menginputkan nilai1
        DataInputStream nilai1=new DataInputStream (System.in);
        System.out.print("Inputkan nilai1 : ");
        String snilai1=nilai1.readLine();

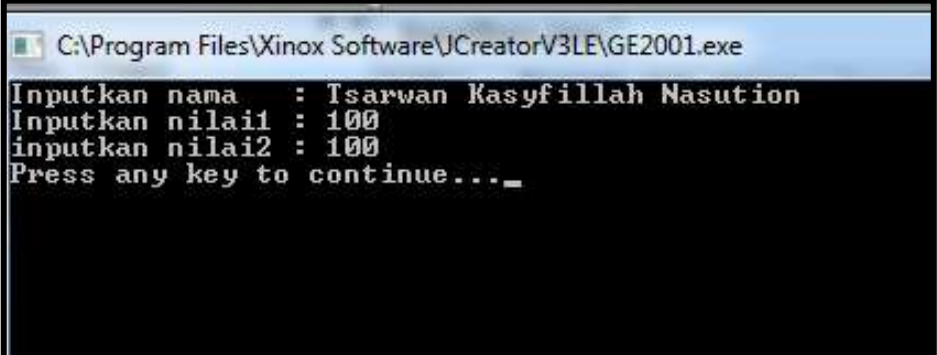
        //mendinputkan nilai2
```

- *Variabel* yang digunakan pada program tersebut adalah:

*String nama, nilai1 dan nilai2*

Dengan *Variabel inputan* adalah : nama, nilai1 dan nilai2

Maka *output* dari program dengan *variable* yang dideklarasikan menggunakan *variable* konstan *inputan* adalah :



```
C:\Program Files\Xinox Software\JCreatorV3LE\GE2001.exe
Inputkan nama   : Tsarwan Kasyfillah Nasution
Inputkan nilai1 : 100
inputkan nilai2 : 100
Press any key to continue...
```

Gambar 2.6 Tampilan *Output* Menggunakan Variabel Input Pada Java

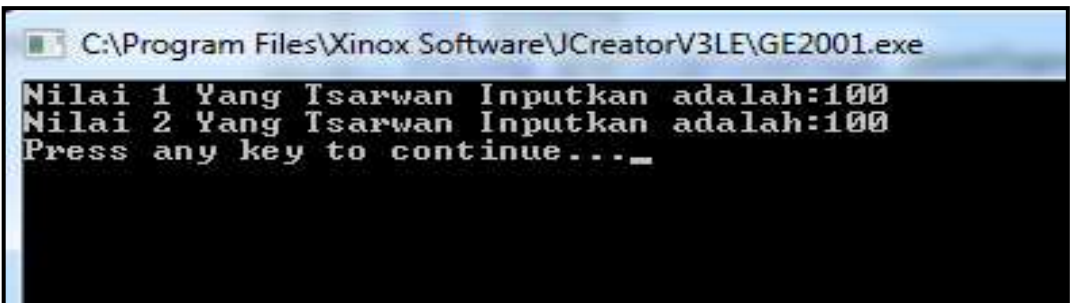
- Contoh program dengan variable *inputan* pada program Java

```
import javax.swing.JOptionPane;
public class Input2 {
public static void main(String args[]) {
int nilai1;
int nilai2;
String str=JOptionPane.showInputDialog("Masukkan nilai 1 :");
nilai1=Integer.parseInt(str);
String str1=JOptionPane.showInputDialog("Masukkan nilai 2 :");
nilai2=Integer.parseInt(str1);

System.out.println("Nilai 1 Yang Tsarwan Inputkan adalah:"+ nilai1);
System.out.println("Nilai 2 Yang Tsarwan Inputkan adalah:"+ nilai2);

System.exit(0);
}
}
```

*Variabel* yang digunakan pada program tersebut adalah:



```
C:\Program Files\Xinox Software\JCreatorV3LE\GE2001.exe
Nilai 1 Yang Tsarwan Inputkan adalah:100
Nilai 2 Yang Tsarwan Inputkan adalah:100
Press any key to continue..._
```

# 2.2 Pengertian Tipe Data, Deklarasi dan Contoh

Gambar 2.7 Tampilan *Output* Menggunakan Variabel Input Pada Java

yang digunakan pada  
deklarasikan *variable*,  
menentukan apakah  
dengan jenis datanya.  
angka memiliki nilai  
nilai bulat, ada juga  
pecahan itu pun  
alagi untuk tipe data  
ka dengan huruf, oleh  
harus mengenal tipe

ang kita gunakan

ar kapasitas memori  
a pendeklarasian tipe  
nyimpanan (*storage*)  
mpengaruhi proses

plikatif Logika&Algoritma(C++,C# dan Java)

- **Macam-Macam Tipe Data**

Tipe data yang umum digunakan dalam bahasa pemrograman C++ diantaranya adalah :

- a. Tipe Data Angka

Untuk jenis tipe data angka memiliki nilai dan panjang field yang berbeda

- *Integer (int)* : “ Merupakan tipe data yang digunakan untuk menyimpan nilai dengan bilangan bulat positif tanpa ada titik *decimal* pada bilangan tersebut, misalnya 1,2,3,... dst

Cara mendeklarasikannya pun sederhana yaitu :

1. Dengan nilai konstan misalnya

```
Intx=5;
```

2. Dengan data inputan

```
Intx;
```

67

- *Floating Point*.”Merupakan tipe data yang digunakan untuk menyimpan data angka



o Tipe data *floating Point* terbagi atas :

1. *Float* :”merupakan tipe data yang menyimpan nilai pecahan dengan panjang *field* 32 bit

2. *Double* dan *Long* :”merupakan tipe data yang menyimpan nilai pecahan dengan panjang *field* 64 bit

Cara mendeklarasikannyapun sama yaitu :

1. Dengan nilai konstan

*Float* x = 3.12;

*Double* x = 3.122222;

2. Dengan nilai inputan

*Float* x;

*Double* x;

b. Tipe Data Karakter :”merupakan tipe data yang digunakan untuk menyimpan nilai karakter ( 1 buah

c. Tipe data string :”Merupakan tipe data yang menyimpan nilai dari gabungan beberapa karakter.

Cara mendeklarasikannya adalah :

```
Char Kota [10];
```

# CONTOH PROGRAM MENGGUNAKAN TIPE DATA

## Contoh Program C++ Menggunakan Tipe Data Angka

```
#include <iostream.h>

int main ()

{

// inialisasi variabel :

float nilai1, nilai2;

int hasil;

double hasil1;

float hasil2;

// Menginputkan nilai :

cout<<"Inputkan nilai1: "<<;

cin>>nilai1;

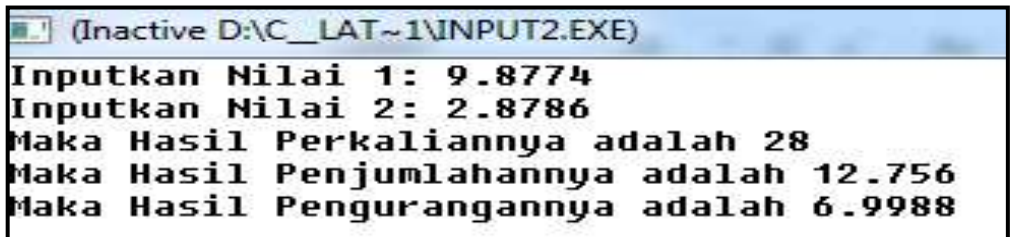
cout<<"Inputkan nilai2: ";

cin>>nilai2;

// memproses Perkalian, Penjumlahan dan Pengurangan :
```

```
// mencetak hasil Perkalian, Penjumlahan dan Pengurangan :  
  
cout << "Maka Hasil Perkaliannya adalah " << hasil<<"\n";  
  
cout << "Maka Hasil Penjumlahannya adalah " << hasil1<<"\n";  
  
cout << "Maka Hasil Pengurangannya adalah " << hasil2<<"\n";  
  
// menghentikan program :  
  
return 0;  
  
}
```

Tipe data yang digunakan pada program tersebut adalah:



```
(Inactive D:\C_LAT~1\INPUT2.EXE)  
Inputkan Nilai 1: 9.8774  
Inputkan Nilai 2: 2.8786  
Maka Hasil Perkaliannya adalah 28  
Maka Hasil Penjumlahannya adalah 12.756  
Maka Hasil Pengurangannya adalah 6.9988
```

Gambar 2.8 Tampilan *Output* Menggunakan Tipe Data Angka

Terlihat hasil *output* berbeda untuk tipe data *integer*, *float* dan *double* dengan masing-masing panjang *field* yang sudah diatur oleh tipe data tersebut, jika kita menggunakan integer maka hasil nilai yang tersimpan bilangan bulat berbeda dengan *float* dan *double* menghasilkan nilai pecahan.

### Contoh Program C++ Menggunakan Tipe Data Huruf dan Gabungan beberapa huruf

```
#include<iostream.h>
main()
{

// Deklarasi Variabel untuk 1 buah huruf
char x;

// Deklarasi Variabel lebih dari 1 huruf

char i[20],j[20],k[20],l[20];

// Menginputkan nilai variabel

cout<<"Inputkan Sebuah huruf  : ";
cin>>x;
cout<<"Inputkan Nama Anda   : ";
cin>>i;
cout<<"Inputkan Kelas Anda   : ";
cin>>i;
```

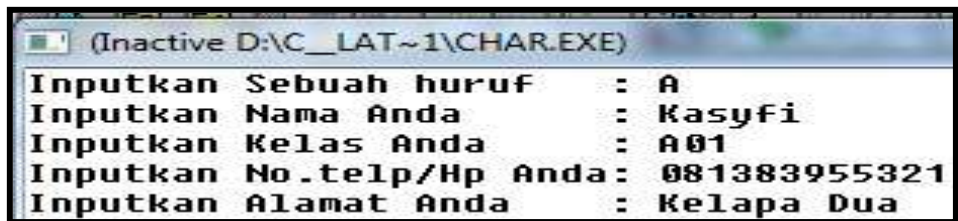
Variabel yang digunakan pada program tersebut adalah

1. *Char* x: menampung nilai dengan nama *variablenya* adalah x dengan panjang *field* 1 huruf
2. *Char* i[20] : menampung nilai dengan nama *variablenya* adalah nama dan panjang *field* yang dideklarasikan maksimum adalah 20 huruf
3. *Char* j[20] : menampung nilai dengan *variablenya* adalah kelas dan panjang *field* yang dideklarasikan maksimum adalah 20 huruf
4. *Char* k[20] : menampung nilai dengan *variablenya* adalah nomor telp/HP dan panjang *field* yang dideklarasikan maksimum 20 huruf

**Catatan : ingat nomor telp/HP tidak dipergunakan untuk kalkulasi/perhitungan jadi kita cukup deklarasikan tipe datanya dengan *char* saja bukan dengan *integer***



Maka *output* dari program tersebut :



```
(Inactive D:\C_LAT~1\CHAR.EXE)
Inputkan Sebuah huruf      : A
Inputkan Nama Anda        : Kasyfi
Inputkan Kelas Anda       : A01
Inputkan No.telp/Hp Anda  : 081383955321
Inputkan Alamat Anda      : Kelapa Dua
```

Gambar 2.9 Tampilan *Output* Menggunakan tipe data *Character*

### Contoh Program C++ Menggunakan Tipe Data campuran

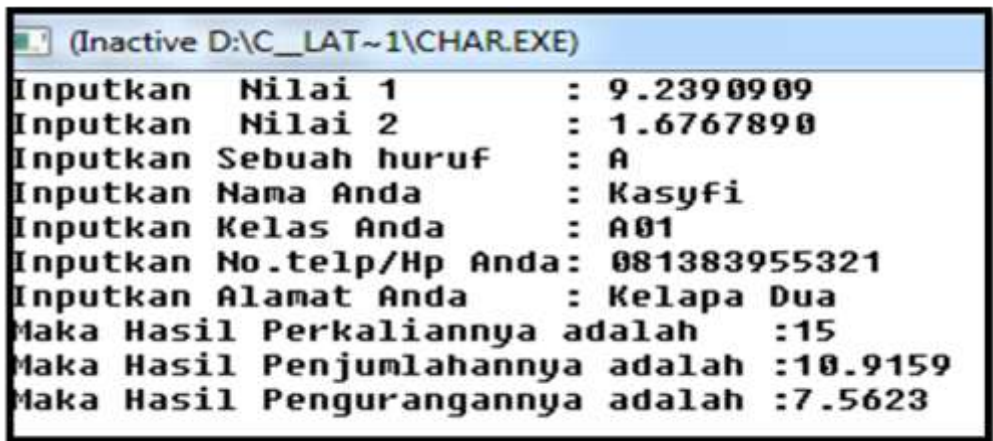
```
#include<iostream.h>
main()
{
// Deklarasi Variabel untuk 1 buah huruf
char x;
// Deklarasi Variabel lebih dari 1 huruf dan angka
char i[20],j[20],k[20],l[20];
// inialisasi variabel :
float nilai1, nilai2;
74 int hasil;
double hasil1;
float hasil2;
// Menginputkan nilai variabel
cout<<"Inputkan Nilai 1   : ";
cin>>nilai1;
cout<<"Inputkan Nilai 2   : ";
// Menginputkan nilai variabel
```

```
cout<<"Inputkan Kelas Anda   :";
cin>>j;
cout<<"Inputkan No.telp/Hp Anda: ";
cin>>k;
cout<<"Inputkan Alamat Anda   :";
cin>>l;

// memproses Perkalian, Penjumlahan dan Pengurangan :
hasil = nilai1*nilai2;
hasil1 = nilai1+nilai2;
hasil2 =nilai1-nilai2;

// mencetak hasil Perkalian, Penjumlahan dan Pengurangan :
cout << "Maka Hasil Perkaliannya adalah " << hasil<<"\n";
cout << "Maka Hasil Penjumlahannya adalah " << hasil1<<"\n";
cout << "Maka Hasil Pengurangannya adalah " << hasil2<<"\n";

// menghentikan program :
return 0;
}
```



```
(Inactive D:\C_LAT~1\CHAR.EXE)
Inputkan Nilai 1      : 9.2390909
Inputkan Nilai 2      : 1.6767890
Inputkan Sebuah huruf : A
Inputkan Nama Anda    : Kasyfi
Inputkan Kelas Anda   : A01
Inputkan No.telp/Hp Anda: 081383955321
Inputkan Alamat Anda  : Kelapa Dua
Maka Hasil Perkaliannya adalah :15
Maka Hasil Penjumlahannya adalah :10.9159
Maka Hasil Pengurangannya adalah :7.5623
```

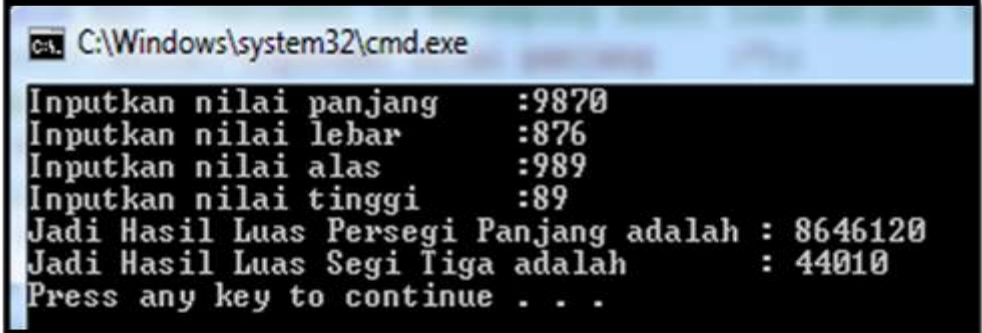
Gambar 2.10 Tampilan *Output* Menggunakan Kombinasi Tipe Data *Integer* dan *Character*

# CONTOH PROGRAM MENGGUNAKAN TIPE DATA

## Contoh Program C# Menggunakan Tipe Data Angka

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace integer
{
    class Program
    {
        static void Main(string[] args)
        {
            int pj; //variabel nilai panjang dengan tipe data Integer
            int lb; //variabel nilai lebar dengan tipe data long integer
            long L; //variabel hasil Luas dengan tipe data Long
            long al; //variabel nilai alas dengan tipe data Long
            int t; //variabel nilai tinggi dengan tipe data integer
            double LL; //variabel hasil Luas dengan tipe data double
            Console.Write("Inputkan nilai panjang      :");
            //konversi dari tipe string ke integer
            pj = Convert.ToInt32(Console.ReadLine());
            Console.Write("Inputkan nilai lebar      :");
            lb = Convert.ToInt32(Console.ReadLine());
            L = pj*lb;
            Console.Write("Inputkan nilai alas      :");
            //konversi dari tipe string ke integer
            al = Convert.ToInt64(Console.ReadLine());
            Console.Write("Inputkan nilai tinggi      :");
            //konversi dari tipe string ke integer
            t = Convert.ToInt32(Console.ReadLine());
            LL = (al * t) / 2;
            Console.WriteLine(" Jadi Hasil Luas Persegi Panjang adalah : " + L);
            Console.WriteLine("Jadi Hasil Luas Segi Tiga adalah      : " + LL);
        }
    }
}
```

Tipe Data Yang digunakan pada program tersebut adalah : tipe data angka dengan berbagai macam nilai yaitu *integer*, *long* dan *double*.  
Maka output dari program tersebut adalah :



```
C:\Windows\system32\cmd.exe
Inputkan nilai panjang      :9870
Inputkan nilai lebar       :876
Inputkan nilai alas        :989
Inputkan nilai tinggi      :89
Jadi Hasil Luas Persegi Panjang adalah : 8646120
Jadi Hasil Luas Segi Tiga adalah      : 44010
Press any key to continue . . .
```

Gambar 2.11 Tampilan *Output* Menggunakan Tipe Data Angka

### Contoh Program C# Menggunakan Tipe Data Angka dan Huruf

Catatan : Untuk data gabungan beberapa huruf pada C# menggunakan tipe data *string*

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication11
{
    class Program
    {
        static void Main(string[] args)
        {
            // variabel inputan
            int jml;
            long tlp;
            double total;
            double hrg;
            string plgn, nb;

            // Statement input dan output

            Console.WriteLine("=====\n\n");
            Console.WriteLine("Rekapitulasi Pembelian Produk PT Untung Selalu\n\n");
            Console.WriteLine(" Menggunakan Tipe Data Angka dan Huruf\n\n");
            Console.WriteLine("=====\n\n");
            Console.Write("Inputkan Nama Produk           : ");
            nb = Console.ReadLine();
            Console.Write("Inputkan Jumlah Produk           : ");
            jml = Convert.ToInt32(Console.ReadLine());
            Console.Write("\nHarga Produk adalah           : ");
            hrg = Convert.ToInt64(Console.ReadLine());
            Console.Write("\nInputkan Nama Pelanggan         : ");
            plgn = Console.ReadLine();
            Console.Write("\nInputkan No.Telp Pelanggan      : ");
            tlp = Convert.ToInt64(Console.ReadLine());
            total = (hrg * jml);
```



```
Console.WriteLine("=====\n\n");

Console.WriteLine("\n Terima Kasih Atas Kunjungan Anda ");
Console.WriteLine("\n Sampai Jumpa Lagi di PT Untung Selalu ");

Console.WriteLine("\n=====");
Console.ReadLine();
    }
}
}
```

Tipe Data Yang digunakan pada program tersebut adalah : tipe data angka dengan berbagai macam nilai yaitu *integer*, *long* dan *double*. dan tipe data *string*

```
cmd C:\Windows\system32\cmd.exe
-----
Rekapitulasi Pembelian Produk PT Untung Selalu
Menggunakan Tipe Data angka dan Huruf
-----
Inputkan Nama Produk           : Komputer
Inputkan Jumlah Produk        : 9
Harga Produk adalah           : 9817281
Inputkan Nama Pelanggan       : Fauziah, S.Kom, MMSI
Inputkan No.Telp Pelanggan    : 081383955321
-----
Total Pembayaran adalah       : 88355529
-----
Terima Kasih Atas Kunjungan Anda
Sampai Jumpa Lagi di PT Untung Selalu
-----
```

Gambar 2.12 Tampilan Output menggunakan tipe data angka dan huruf

### Contoh Program Java Menggunakan Tipe Data Angka

```
import java.io.*;
public class typedata {
public static void main(String[] args)throws Exception {
BufferedReader input = new BufferedReader
(new InputStreamReader (System.in));
String nama,mtk,alpro,fisika,bing,manaj, pl, pti;
double a,b,c,d,e,f,g,x,y;
System.out.println("Menggunakan Tipe Data Angka");
System.out.println("=====");
System.out.print ("Inputkan Nama Anda:      : ");
nama = input.readLine();
System.out.print ("Inputkan Nilai Matematika  :");
mtk = input.readLine();
System.out.print ("Inputkan Nilai Algoritma   :");
alpro = input.readLine();
System.out.print ("Inputkan Nilai Fisika     :");
fisika = input.readLine();
System.out.print ("Inputkan Nilai Bahasa Inggris :");
bing = input.readLine();
System.out.print ("Inputkan Nilai Manajemen  :");
manaj = input.readLine();
System.out.print ("Inputkan Nilai Perangkat Lunak:");
pl = input.readLine();
System.out.print ("Inputkan Nilai PTI       :");
pti = input.readLine();
System.out.println("=====");
```

```
x = (a+b+c)/3;
y = (d+e+f+g)/4;

System.out.println("=====
=");
System.out.println("Hasil Rata - Rata Nilai Eksak dan Non Eksak Adalah");
System.out.println("=====
=");
System.out.println("");
System.out.println("nilai eksak : "+ x);
System.out.println("nilai non eksak : " + y);
System.out.println("");
}
```

```
C:\Program Files\Xinox Software\JCreatorV3LE\GE2001.exe
Menggunakan Tipe Data Angka
-----
Inputkan Nama Anda           : Fauziah, S.Kom, MMSI
Inputkan Nilai Matematika    : 90.89
Inputkan Nilai Algoritma     : 90.87
Inputkan Nilai Fisika        : 89.09
Inputkan Nilai Bahasa Inggris : 98.76
Inputkan Nilai Manajemen     : 78.90
Inputkan Nilai Perangkat Lunak : 98.77
Inputkan Nilai PTI           : 96.76
-----
Hasil Rata - Rata Nilai Eksak dan Non Eksak Adalah
-----
nilai eksak : 90.28333333333335
nilai non eksak : 93.2975
Press any key to continue...
```

Gambar 2.13 Tampilan *Output* menggunakan tipe data angka

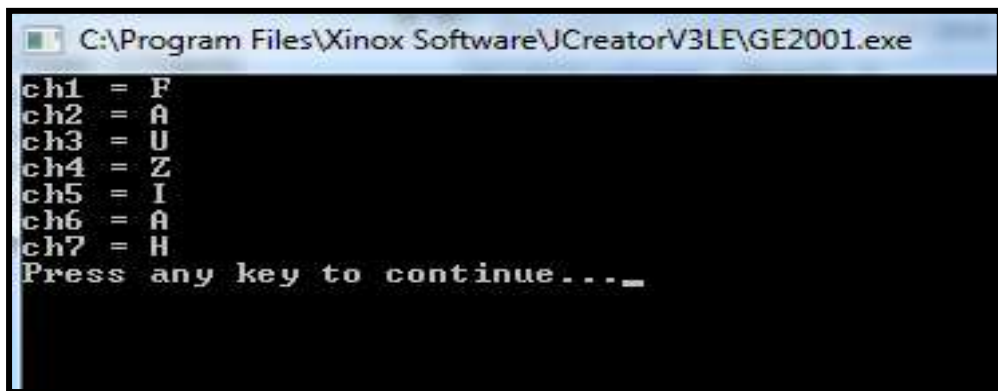
### Contoh Program Java Menggunakan Tipe Data Huruf

```
class char1{
    public static void main(String[] args) {

        char ch1 = 'F';
        char ch2 = 'A';
        char ch3 = 'U';
        char ch4 = 'Z';
        char ch5 = 'I';
        char ch6 = 'A';
        char ch7 = 'H';

        System.out.println("ch1 = " + ch1);
        System.out.println("ch2 = " + ch2);
        System.out.println("ch3 = " + ch3);
        System.out.println("ch4 = " + ch4);
        System.out.println("ch5 = " + ch5);
        System.out.println("ch6 = " + ch6);
        System.out.println("ch7 = " + ch7);
    }
}
```

Tipe data yang digunakan pada program tersebut adalah *Character*, dan *output* dari program tersebut adalah :



The screenshot shows a window titled "C:\Program Files\Xinox Software\JCreatorV3LE\GE2001.exe". The output displayed in the console is:

```
ch1 = F
ch2 = A
ch3 = U
ch4 = Z
ch5 = I
ch6 = A
ch7 = H
Press any key to continue..._
```

Gambar 2.14 Tampilan *Output* Menggunakan Tipe Data *Character*

### Contoh Program Java Menggunakan Kombinasi Tipe Data Huruf dan Angka

```
import java.io.*;
public class indata {
public static void main(String[] args)throws Exception {
BufferedReader input = new BufferedReader
(new InputStreamReader (System.in));
        char ch1 = 'F';
        char ch2 = 'A';
        char ch3 = 'U';
        char ch4 = 'Z';
        char ch5 = 'I';
        char ch6 = 'A';
        char ch7 = 'H';
        String nama,kls,AImt,NoHP;
        int a = 3; int b=2; float c,d,e,f;

System.out.println(" Menggunakan Tipe Data Angka & Kombinasi Huruf" );
System.out.println("=====
=");
System.out.print ("Inputkan Nama Anda      : ");
nama = input.readLine();
System.out.print ("Inputkan Kelas Anda      : ");
kls = input.readLine();
```

```
System.out.println("=====
=");
System.out.println("        Hasil Nilai Akhir        ");
System.out.println("=====
=");
System.out.println("");
System.out.println("Hasil Nilai Akhir : "+f);
System.out.println("=====
=");
System.out.println("    Hasil Data Karakter Dengan Data Konstan    ");
System.out.println("=====
=");
System.out.println("");
System.out.println("ch1 = " + ch1);
System.out.println("ch2 = " + ch2);
System.out.println("ch3 = " + ch3);
System.out.println("ch4 = " + ch4);
System.out.println("ch5 = " + ch5);
System.out.println("ch6 = " + ch6);
System.out.println("ch7 = " + ch7);
System.out.println("=====
=");
System.out.println("    Hasil Data Angka Dengan Data Konstan    ");
System.out.println("=====
=");
        c = a + b;
        d = a * b;
```

```
C:\Program Files\Xinox Software\JCreatorV3LE\GE2001.exe
Menggunakan Tipe Data Angka & Kombinasi Huruf
=====
Inputkan Nama Anda           : Fauziah,S.Kom,MMSI
Inputkan Kelas Anda          : A01
Inputkan Alamat Anda         : Kelapa Dua, Depok
Inputkan No. Handphone Anda  : 081383955321
=====
Memunculkan Hasil Tipe Data Angka& Kombinasi Huruf
=====
                               Hasil Nilai Akhir
=====
Hasil Nilai Akhir : 7.0
=====
                               Hasil Data Karakter Dengan Data Konstan
=====
ch1 = F
ch2 = A
ch3 = U
ch4 = Z
ch5 = I
ch6 = A
ch7 = H
=====
                               Hasil Data Angka Dengan Data Konstan
=====
5.0
6.0
1.0
Press any key to continue...
```

Gambar 2.15 Hasil Output Dengan Kombinasi Tipe Data Angka dan Huruf



# PENGAYAAN

Jawablah pertanyaan di bawah ini dengan jelas dan benar

1. Apa itu tipe Data
2. Tuliskan macam – macam tipe data
3. Buatlah Program dengan menggunakan tipe data angka menggunakan nilai konstan dan inputan untuk mencari : Luas dan Keliling
  - a. Persegi Panjang
  - b. Lingkaran
  - c. Persegi
  - d. Segi Tiga
4. Buatlah program dengan menggunakan tipe data gabungan huruf/*string* menggunakan nilai konstan

# GLOSARIUM



1. Variabel dan tipe data merupakan komponen atau bagian terpenting yang ada dalam bahasa pemrograman
2. *Variabel* merupakan pengenal (*identifier*) dari suatu data yang kita deklarasikan pada bahasa pemrograman
3. *variable* dapat dikatakan juga sebagai wadah (penyimpanan data/alokasi *storage*) yang kita deklarasikan dengan tujuan untuk

4. *Variabel/konstan* :” merupakan jenis *variable* yang sudah memiliki nilai atau *value*

5. Contoh *Variabel/konstan*

```
Int a= 5;
```

```
String nama=”Tsarwan Kasyfillah”;
```

6. *Variabel inputan* :”merupakan jenis *variable* yang tidak memiliki nilai atau *value*, jika kita ingin memberikan nilai maka ada *statement* tambahan yang digunakan yaitu dengan perintah *input variable* dan kita dapat menginputkannya melalui *keyboard*

7. Contoh *Variabel Input*

```
Int a;
```

```
String nama;
```

8. *Tipe data* merupakan jenis data yang digunakan pada saat kita membuat atau mendefinisikan *variable*, oleh karena itu *tipe data* sangat menentukan apakah data yang kita gunakan

10. *Floating Point*:"Merupakan tipe data yang digunakan untuk menyimpan data angka dengan nilai pecahan misalnya 27.72; 54.36 dst. Namun Pada data *floating point* kita dapat mengenal macam – macamnya dan mengetahui panjang *field* yang tersimpan pada masing-masing tipe datanya.

11. Tipe Data Karakter :”merupakan tipe data yang digunakan untuk menyimpan nilai karakter ( 1 buah huruf).

12. Tipe data string :”Merupakan tipe data yang



## BAB III

### OPERATOR PADA BAHASA PEMROGRAMAN C++,C# dan JAVA

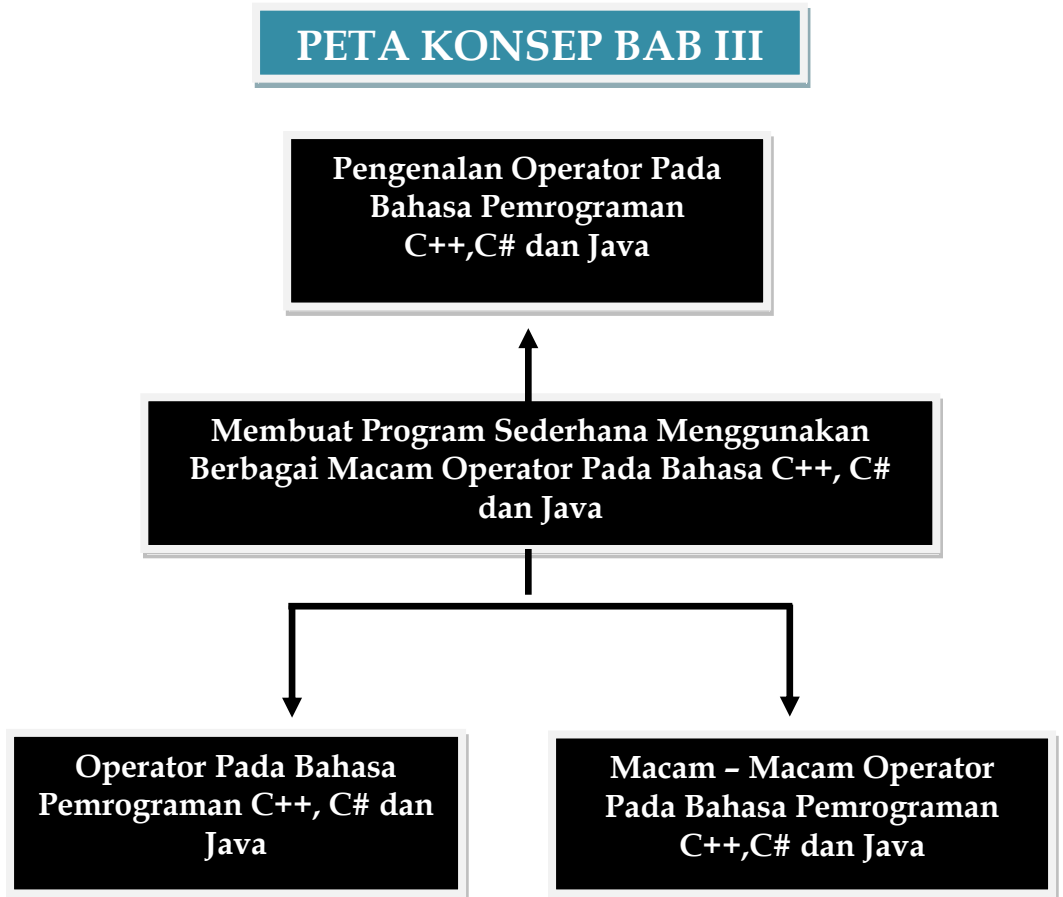
Sumber: [defenseindustrydaily.com](http://defenseindustrydaily.com)

#### **Tujuan Pembelajaran**

Setelah mempelajari bab ini Mahasiswa diharapkan mampu:  
Menjelaskan berbagai macam operator yang digunakan pada bahasa pemrograman C++, C# dan Java, serta mampu memberikan contoh dengan program menggunakan berbagai macam operator tersebut.

Kode program yang dibuat oleh pengguna atau *programmer* tidak terlepas dari penggunaan berbagai macam *operator*, apalagi dalam program menggunakan berbagai macam kalkulasi atau perhitungan, maka *operator* pun digunakan.

Kata Kunci: *operator*, program, macam *operator*, *programmer*



# 3.1 Pengertian Operator Pada Bahasa Pemrograman

- **Pengertian *Operator***

*Operator* merupakan *symbol* atau berbagai macam lambang yang sering kita gunakan dalam bahasa pemrograman dan memiliki fungsi masing – masing.

Selain *operator* kita juga harus mengenal *operand*, yaitu pengenal atau *variable* yang digunakan untuk mengoperasikan suatu nilai, misalnya saja

*Int* X = A + B;

Keterangan :

{A,B dan X } : merupakan *operand* (Variabel)



# 3.2 Macam-Macam Operator Pada Bahasa Pemrograman

- **Macam – Macam *Operator* Pada Bahasa Pemrograman**

Macam-Macam *Operator* yang digunakan pada bahasa pemrograman diantaranya adalah :

*a. Operator Assignment*

Merupakan jenis *operator* yang berfungsi untuk memberikan suatu nilai(*value*) kepada *variable* yang ditunjuk, biasanya *operator assignment* digunakan untuk jenis data *variable* konstan. *Operator assignment* dapat juga digunakan untuk perhitungan proses dengan rumus tertentu. *Symbol* yang digunakan pada *operator assignment* adalah “[=]” lambang sama dengan

Contoh pendeklarasian operator assignment

Keterangan :

Pada contoh 1 dan 2 *operator assignment* digunakan untuk memberikan nilai kepada *variable* x dengan nilai 5 dan untuk *variable string* dengan nilai "Tsarwan Kasyfi". Sedangkan pada contoh yang 3 *operator assignment* digunakan untuk melakukan proses perhitungan nilai  $x=a+b$ ; dimana hasilnya akan disimpan pada *variable* x dan prosesnya dilakukan oleh *variable* a dan b

b. Operator Logika

Merupakan jenis operator yang memberikan nilai true(benar) dan false(salah).

Simbol yang digunakan pada operator logika adalah:

Logika And (Logika dan)[symbol yang digunakan &&]

Logika Or (Logika atau)[symbol yang digunakan ||]

Logika Negasi(Logika ingkaran) [symbol yang digunakan adalah [ ! ]

Tabel Logika And [&&]

A	B	Hasil A&&B
1	1	1
1	0	0
0	1	0
0	0	0

Keterangan Tabel :

Untuk jenis logika *And* [&&] menggunakan nilai 1[*true*/benar] dan nilai 0[*false*/salah]. Pada logika *and* terlihat pada kedua buah *variable* harus bernilai benar, jika salah satunya salah maka hasil akhirnya salah.

Tabel Logika Or [||]

A	B	Hasil A  B
1	1	1
1	0	1
0	1	1
0	0	0

Keterangan Tabel :

Untuk jenis logika *Or* `||` menggunakan nilai 1[*true*/benar] dan nilai 0[*false*/salah]. Pada logika *or* terlihat pada kedua buah *variable* harus bernilai benar atau salah satunya boleh bernilai benar, maka hasilnya akan benar dan jika kedua buah *variable* bernilai salah maka hasilnya akan salah.

Tabel Logika *Negasi* `!`/ Ingkaran

A	!A
1	0
0	1

c. *Operator Relasional*

Merupakan jenis *operator* yang digunakan untuk menghubungkan 2 buah *variable* atau *operand* yang menghasilkan suatu keputusan. *Symbol – symbol* yang bisa digunakan untuk *operator relasional* adalah

d. *Operator Increment dan Decrement*

Merupakan jenis *operator* yang digunakan untuk melakukan proses penambahan atau pengurangan nilai [ yaitu nilai peubah naik atau nilai turun ] dan biasanya digunakan pada jenis *structur control* perulangan.

Logika *Operator Increment* (nilai penambahan dengan *symbol ++*)

Misalnya  $x++$  artinya nilai  $[x = x+1]$

Logika *Operator Decrement* (nilai pengurangan dengan *symbol --*)

Misalnya  $x--$  artinya nilai  $[x = x-1]$

e. *Operator Aritmatika*

# Contoh Program C++ Operator

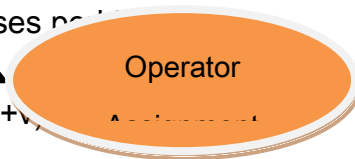
- Contoh Program dengan C++ Menggunakan *Operator*

```
// Program 3.1
#include <iostream.h>
int main ()
{
// inisialisasi variabel :
long int P, l,a,t, s,u,v;
long int K, S;
// proses :
cout<< " ===== " << endl;
cout<< " |Menghitung Sisa Pendapatan Tsarwan| " << endl;
cout<< " ===== " << endl;
cout<<"Masukkan Pendapatan          : ";
cin>>P;
cout<<"Pembayaran Uang Kos          : ";
cin>>l;
"\n" ;
cout<<"Pembayaran Uang Transport      : ";
```

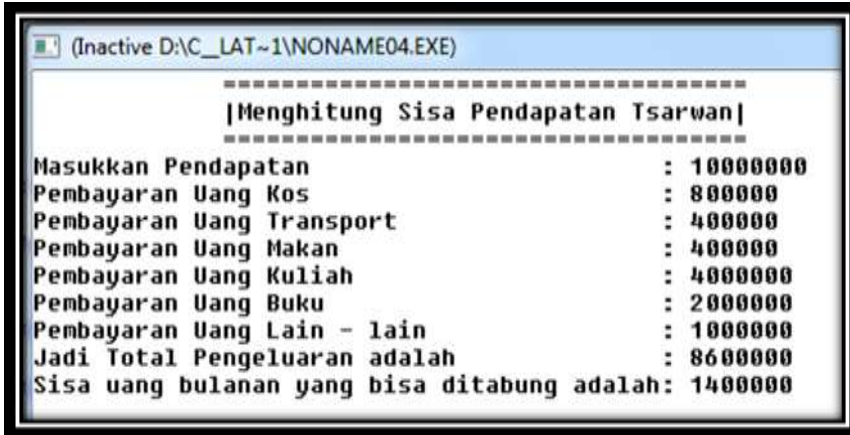
```
cin>>u;
cout<<"Pembayaran Uang Lain - lain      : ";
cin>>v;
// cetak hasilnya :
K = (l+a+t+s+u+v);
S = (P-K);
cout<<"Jadi Total Pengeluaran adalah      : "<<K<<endl;
cout<<"Sisa uang bulanan yang bisa ditabung adalah: "<<S<<endl;
// menghentikan program :
return 0;
}
```

Pada Program 3.1, menggunakan *operator Assignment* yaitu pada proses no 1

```
K = (l+a+t+s+u+v);
S = (P-K);
```







```
(Inactive D:\C_LAT~1\NONAME04.EXE)
=====
|Menghitung Sisa Pendapatan Tsarwan|
=====
Masukkan Pendapatan                : 10000000
Pembayaran Uang Kos                 : 800000
Pembayaran Uang Transport           : 400000
Pembayaran Uang Makan               : 400000
Pembayaran Uang Kuliah              : 4000000
Pembayaran Uang Buku                : 2000000
Pembayaran Uang Lain - lain         : 1000000
Jadi Total Pengeluaran adalah       : 8600000
Sisa uang bulanan yang bisa ditabung adalah: 1400000
```

Gambar 3.1 Tampilan Output menggunakan *operator Assignment*

- Menggunakan *Operator Aritmatika*

Program berikut adalah program menggunakan *operator aritmatika* dengan symbol [+,-,\* dan /]. Pada program berikut ini kita menggunakan data konstan dan tipe data *integer*.

```
//Program 3.2
#include<iostream.h>
int main ()
{
int x=8, y=4, a= 6, b=3,c=1;
cout<<"=====
==" <<endl;
cout<<"      =====" <<endl;
cout<<"          Menggunakan Operator Aritmatika          "<<endl;
cout<<"      =====" <<endl;
cout<<"=====
==" <<endl;
cout<<x<<"+"<<y<<" Menggunakan Operator Aritmatika Penjumlahan Maka Hasilnya =" <<
x+y<<endl;
cout<<x<<"*"<<y<<" Menggunakan Operator Aritmatika Perkalian   Maka Hasilnya =" <<
x*y<<endl;
cout<<x<<"-"<<y<<" Menggunakan Operator Aritmatika Pengurangan Maka Hasilnya =" <<
x-y<<endl;
cout<<x<<":"<<y<<" Menggunakan Operator Aritmatika Pembagian   Maka Hasilnya =" <<
x/y<<endl;
cout<<a<<"+"<<b<<" Menggunakan Operator Aritmatika Penjumlahan Maka Hasilnya =" <<
a+b<<endl;
cout<<a<<"*"<<b<<" Menggunakan Operator Aritmatika Perkalian   Maka Hasilnya =" <<
a*b<<endl;
cout<<a<<"-"<<b<<" Menggunakan Operator Aritmatika Pengurangan Maka Hasilnya =" <<
a-b<<endl;
```

```
cout<<b<<:".<<c<<" Menggunakan Operator Aritmatika Pembagian Maka Hasilnya =" <<
b/c<<endl;
cout<<"=====
==" " <<endl;
cout<<" ===== " <<endl;
cout<<" ===== " <<endl;
cout<<" ===== " <<endl;
cout<<" @ Created By :Tsarwan Kasyfillah" <<endl;
```



The screenshot shows a terminal window titled "(Inactive D:\C\_LAT~1\OPERARIT.EXE)". The output of the program is as follows:

```
=====
                Menggunakan Operator Aritmatika
=====
8+4 Menggunakan Operator Aritmatika Penjumlahan Maka Hasilnya =12
8*4 Menggunakan Operator Aritmatika Perkalian Maka Hasilnya =32
8-4 Menggunakan Operator Aritmatika Pengurangan Maka Hasilnya =4
8:4 Menggunakan Operator Aritmatika Pembagian Maka Hasilnya =2
6+3 Menggunakan Operator Aritmatika Penjumlahan Maka Hasilnya =9
6*3 Menggunakan Operator Aritmatika Perkalian Maka Hasilnya =18
6-3 Menggunakan Operator Aritmatika Pengurangan Maka Hasilnya =3
6:3 Menggunakan Operator Aritmatika Pembagian Maka Hasilnya =2
3+1 Menggunakan Operator Aritmatika Penjumlahan Maka Hasilnya =4
3*1 Menggunakan Operator Aritmatika Perkalian Maka Hasilnya =3
3-1 Menggunakan Operator Aritmatika Pengurangan Maka Hasilnya =2
3:1 Menggunakan Operator Aritmatika Pembagian Maka Hasilnya =3
=====
                @ Created By :Tsarwan Kasyfillah
```

Gambar 3.2 Tampilan output Menggunakan Operator Aritmatika

## CONTOH PROGRAM DENGAN OPERATOR MENGGUNAKAN DATA KONSTAN

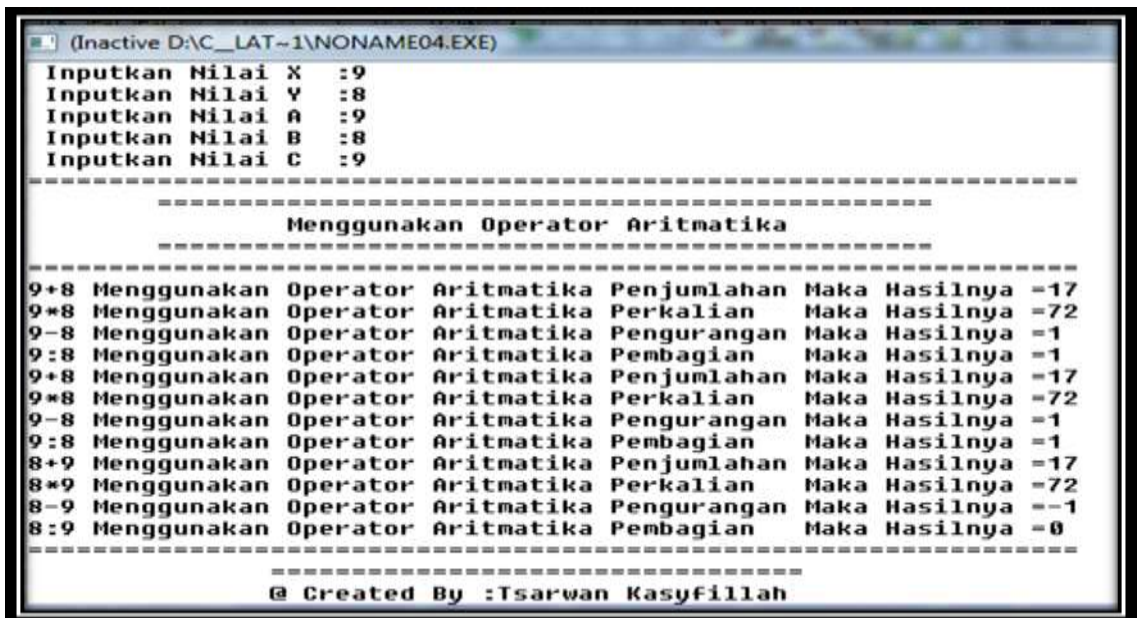
```
//Program 3.3
#include<iostream.h>
int main ()
{
// operator aritmatika dengan data inputan
int x, y, a, b,c;
cout<<" Inputkan Nilai X :"; cin>>x;
cout<<" Inputkan Nilai Y :"; cin>>y;
cout<<" Inputkan Nilai A :"; cin>>a;
cout<<" Inputkan Nilai B :"; cin>>b;
cout<<" Inputkan Nilai C :"; cin>>c;
cout<<"=====
==" <<endl;
cout<<"      =====" <<endl;
cout<<"      Menggunakan Operator Aritmatika      "<<endl;
cout<<"      =====" <<endl;
cout<<"=====
==" <<endl;
cout<<x<<"+"<<y<<" Menggunakan Operator Aritmatika Penjumlahan Maka Hasilnya =" <<
x+y<<endl;
cout<<x<<"*"<<y<<" Menggunakan Operator Aritmatika Perkalian   Maka Hasilnya =" <<
x*y<<endl;
cout<<x<<"-"<<y<<" Menggunakan Operator Aritmatika Pengurangan Maka Hasilnya =" <<
x-y<<endl;
cout<<x<<"/"<<y<<" Menggunakan Operator Aritmatika Pembagian   Maka Hasilnya =" <<
x/v<<endl;
```

```

cout<<b<<"-"<<c<<" Menggunakan Operator Aritmatika Pengurangan Maka Hasilnya =" <<
b-c<<endl;
cout<<b<<" ":"<<c<<" Menggunakan Operator Aritmatika Pembagian Maka Hasilnya =" <<
b/c<<endl;
cout<<"=====
== " <<endl;
cout<<" ===== " <<endl;
cout<<" @ Created By :Tsarwan Kasyfillah" <<endl;
// menghentikan program :
return 0;
}

```

Pada Program 3.2 menggunakan *operator aritmatika* yaitu penjumlahan,



Gambar 3.3 Tampilan Output Menggunakan Operator Aritmatika dengan Data Input

- Contoh Program dengan C++Menggunakan  
*Operator Perbandingan*

Program 3.4

```
#include<iostream.h>
int main ()
{
// Operator aritmatika dengan data inputan

int x, y; // inisialisasi Variabel input

cout<< " ===== " << endl;
cout<< "| Operator Pembanding Data Inputan | " << endl;
cout<< " ===== " << endl;

// Proses Data Input

cout<<" Inputkan Nilai X :"; cin>>x;
cout<<" Inputkan Nilai Y :"; cin>>y;

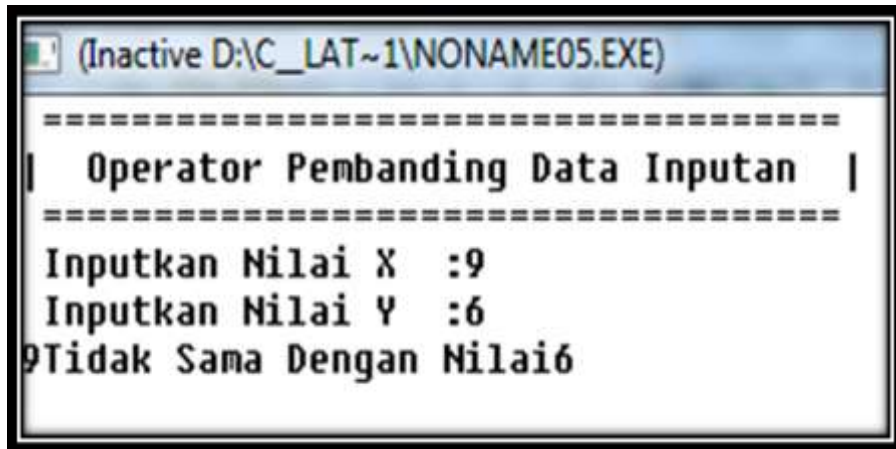
// Operator Pembanding Dengan Pengecekan Nilai

if(x==y)cout<< x <<"Sama Dengan Nilai"<< y <<endl;
else if(x!=y) cout<< x <<"Tidak Sama Dengan Nilai"<< y <<endl;
```

- Contoh Program dengan C++

### Menggunakan *Operator Perbandingan*

Dari program 3.4 menggunakan *operator* perbandingan dengan data *input*, maka *output* yang dihasilkan adalah sebagai berikut :



```
(Inactive D:\C_LAT~1\NONAME05.EXE)
=====  
| Operator Pembanding Data Inputan |  
=====  
Inputkan Nilai X :9  
Inputkan Nilai Y :6  
Tidak Sama Dengan Nilai6
```

Gambar 3.4 Menggunakan *Operator Aritmatika* Dengan Data *Input*

- Contoh Program dengan C++ Menggunakan *Operator Increment* dan *Decrement* Menggunakan Variabel Konstan

```
//Program 3.5
#include<iostream.h>
int main ()
{
// Operator increment & Decrement dengan data variabel konstan
int x=5;
int y=8;
cout<< " ===== " << endl;
cout<< " |   Operator Increment & Decrement   | " << endl;
cout<< " ===== " << endl;
// Proses Data Increment dan Decrement
cout<<" Hasil Nilai X Sebelum Dilakukan Operasi Increment Adalah :"<<x<<endl;
x++;
cout<<" Hasil Nilai X Setelah Dilakukan Operasi Increment Adalah :"<<x<<endl;
cout<<" Hasil Nilai Y Sebelum Dilakukan Operasi Increment Adalah :"<<y<<endl;
y++;
cout<<" Hasil Nilai Y Setelah Dilakukan Operasi Increment Adalah :"<<y<<endl;
cout<<" Hasil Nilai X Sebelum Dilakukan Operasi Decrement Adalah :"<<x<<endl;
x--;
cout<<" Hasil Nilai X Setelah Dilakukan Operasi Decrement Adalah :"<<x<<endl;
cout<<" Hasil Nilai Y Sebelum Dilakukan Operasi Decrement Adalah :"<<y<<endl;
y--;
cout<<" Hasil Nilai Y Setelah Dilakukan Operasi Decrement Adalah :"<<y<<endl;
cout<< " ===== " << endl;
cout<< " |           Terima Kasih Selamat Mencoba           | " << endl;
cout<< " ===== " << endl;
cout<< " ===== " << endl;
cout<< " |   Created by:@Tsarwan Kasyfi           | " << endl;
cout<< " ===== " << endl;
// menghentikan program :
return 0;
```



```
=====
|                               Operator Increment & Decrement                               |
=====
Hasil Nilai X Sebelum Dilakukan Operasi Increment Adalah :5
Hasil Nilai X Setelah Dilakukan Operasi Increment Adalah :6
Hasil Nilai Y Sebelum Dilakukan Operasi Increment Adalah :8
Hasil Nilai Y Setelah Dilakukan Operasi Increment Adalah :9
Hasil Nilai X Sebelum Dilakukan Operasi Decrement Adalah :6
Hasil Nilai X Setelah Dilakukan Operasi Decrement Adalah :5
Hasil Nilai Y Sebelum Dilakukan Operasi Decrement Adalah :9
Hasil Nilai Y Setelah Dilakukan Operasi Decrement Adalah :8
=====
|                               Terima Kasih Selamat Mencoba                               |
=====
|                               Created by:@Tsarwan Kasyfi                               |
=====
```

Gambar 3.5 Tampilan *Output* Menggunakan *Operator Increment dan Decrement*

- Contoh Program dengan C++Menggunakan *Operator Increment dan Decrement* Menggunakan Variabel Inputan

```
//Program 3.6
#include<iostream.h>
int main ()
{
// Operator increment & Decrement dengan data variabel inputan
int x,y;
cout<<" Inputkan Nilai X :";
cin>>x;
cout<<" Inputkan Nilai Y :";
cin>>y;
cout<< " ===== " << endl;
cout<< " | Operator Increment & Decrement | " << endl;
cout<< " ===== " << endl;
// Proses Data Increment dan Decrement
cout<<" Hasil Nilai X Sebelum Dilakukan Operasi Increment Adalah :"<<x<<endl;
x++;
cout<<" Hasil Nilai X Setelah Dilakukan Operasi Increment Adalah :"<<x<<endl;
cout<<" Hasil Nilai Y Sebelum Dilakukan Operasi Increment Adalah :"<<y<<endl;
y++;
cout<<" Hasil Nilai Y Setelah Dilakukan Operasi Increment Adalah :"<<y<<endl;
cout<<" Hasil Nilai X Sebelum Dilakukan Operasi Decrement Adalah :"<<x<<endl;
x--;
cout<<" Hasil Nilai X Setelah Dilakukan Operasi Decrement Adalah :"<<x<<endl;
cout<<" Hasil Nilai Y Sebelum Dilakukan Operasi Decrement Adalah :"<<y<<endl;
y--;
cout<<" Hasil Nilai Y Setelah Dilakukan Operasi Decrement Adalah :"<<y<<endl;
cout<< " ===== " << endl;
cout<< " | Terima Kasih Selamat Mencoba | " << endl;
cout<< " ===== " << endl;
cout<< " ===== " << endl;
cout<< " | Created by @Teguh Kurniawan | " << endl;
```

```
(Inactive D:\C_LAT~1\JNCDECIN.EXE)
Inputkan Nilai X :9
Inputkan Nilai Y :8
=====
|           Operator Increment & Decreament           |
=====
Hasil Nilai X Sebelum Dilakukan Operasi Increment Adalah :9
Hasil Nilai X Setelah Dilakukan Operasi Increment Adalah :10
Hasil Nilai Y Sebelum Dilakukan Operasi Increment Adalah :8
Hasil Nilai Y Setelah Dilakukan Operasi Increment Adalah :9
Hasil Nilai X Sebelum Dilakukan Operasi Decreament Adalah :10
Hasil Nilai X Setelah Dilakukan Operasi Decreament Adalah :9
Hasil Nilai Y Sebelum Dilakukan Operasi Decreament Adalah :9
Hasil Nilai Y Setelah Dilakukan Operasi Decreament Adalah :8
=====
|           Terima Kasih Selamat Mencoba           |
=====
|           Created by:@Tsarwan Kasyfi           |
=====
```

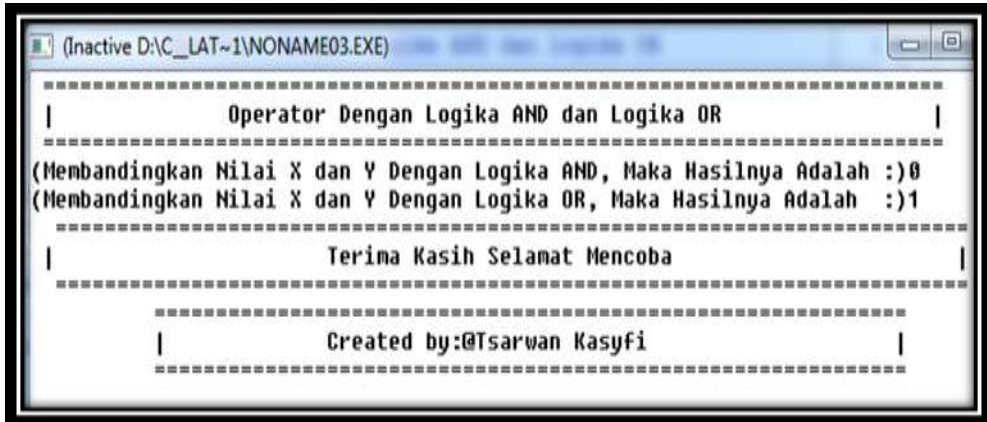
Gambar 3.6 Tampilan Output Operator Increment & Decreament Dengan Variabel Input

- Contoh Program dengan C++Menggunakan *Operator Logika AND dan OR* Menggunakan Variabel Konstan

```
//Program 3.7
#include<iostream.h>
int main ()
{
// Operator increment & Decreament dengan data variabel inputan
int x=100;
int y=1;
cout<< " ===== "
<< endl;
cout<< " | Operator Dengan Logika AND dan Logika OR          | " << endl;
cout<< " ===== "
<< endl;
// Proses Data Dengan Logika AND dan OR
cout<<"(Membandingkan Nilai X dan Y Dengan Logika AND, Maka Hasilnya Adalah :)"
<<(x>=0 && y>=10)<<endl;
cout<<"(Membandingkan Nilai X dan Y Dengan Logika OR, Maka Hasilnya Adalah :)"
<<(x>=0 || y>=10)<<endl;

cout<< " ===== "
<< endl;
cout<< " |      Terima Kasih Selamat Mencoba          | " << endl;
cout<< " ===== "
<< endl;
cout<< "      ===== " << endl;
cout<< " |      Created by:@Tsarwan Kasyfi          | " << endl;
cout<< "      ===== " << endl;

// menghentikan program :
return 0;
```

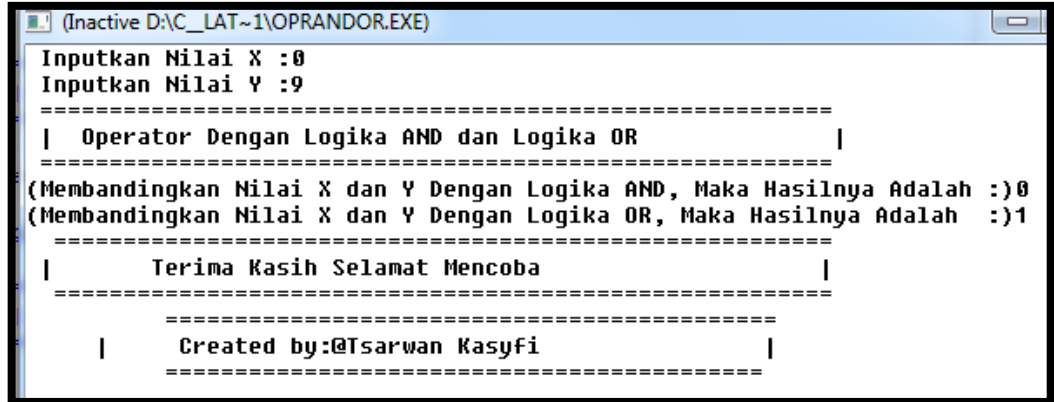


```
(Inactive D:\C_LAT~1\NONAME03.EXE)
=====
|                               Operator Dengan Logika AND dan Logika OR                               |
=====
(Membandingkan Nilai X dan Y Dengan Logika AND, Maka Hasilnya Adalah :)0
(Membandingkan Nilai X dan Y Dengan Logika OR, Maka Hasilnya Adalah :)1
=====
|                               Terima Kasih Selamat Mencoba                               |
=====
|                               Created by:@Tsarwan Kasyfi                               |
=====
```

Gambar 3.7 Tampilan Output Operator Logika And dan OR menggunakan Variabel Konstan

- Contoh Program dengan C++Menggunakan *Operator Logika AND dan OR* Menggunakan Variabel Inputan

```
//Program 3.8
#include<iostream.h>
int main ()
{
// Operator increment & Decrement dengan data variabel inputan
int x;int y;
cout<<" Inputkan Nilai X :";
cin>>x;
cout<<" Inputkan Nilai Y :";
cin>>y;
cout<< " ===== "
<< endl;
cout<< " | Operator Dengan Logika AND dan Logika OR          | " << endl;
cout<< " ===== "
<< endl;
// Proses Data Increment dan Decrement
cout<<"(Membandingkan Nilai X dan Y Dengan Logika AND, Maka Hasilnya Adalah :)"
<<(x>=0 && y>=10)<<endl;
cout<<"(Membandingkan Nilai X dan Y Dengan Logika OR, Maka Hasilnya Adalah :)"
<<(x>=0 || y>=10)<<endl;
cout<< " ===== "
<< endl;
cout<< " |      Terima Kasih Selamat Mencoba          | " << endl;
cout<< " ===== "
<< endl;
cout<< "      ===== " << endl;
cout<< " |      Created by:@Tsarwan Kasyfi          | " << endl;
cout<< "      ===== " << endl;
// menghentikan program :
return 0;
```



```
(Inactive D:\C_LAT~1\OPRANDOR.EXE)
Inputkan Nilai X :0
Inputkan Nilai Y :9
=====
| Operator Dengan Logika AND dan Logika OR |
=====
(Membandingkan Nilai X dan Y Dengan Logika AND, Maka Hasilnya Adalah :)0
(Membandingkan Nilai X dan Y Dengan Logika OR, Maka Hasilnya Adalah :)1
=====
| Terima Kasih Selamat Mencoba |
=====
| Created by:@Tsarwan Kasyfi |
=====
```

Gambar 3.8 Tampilan *Output* Logika AND dan OR Menggunakan *Data Inputan*

# Contoh Program C# Dengan Operator Aritmatika

- Contoh Program C# Menggunakan Operator Aritmatika Dengan Data Konstan

```
// Program 3.9
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace operator1
{
    class Program
    {
        static void Main(string[] args)
        {
            // Deklarasi Data Konstan
            int a = 9, b = 4;
            // Deklarasi Penyimpanan Hasil Penjumlahan,
            // Penyimpanan Hasil Pengurangan, Perkalian dan Pembagian
            int Δ R C;
```



```
// Mencetak Hasil
System.Console.WriteLine("=====");
System.Console.WriteLine("Menggunakan Operator Aritmatika Dengan Data
Konstan ");
System.Console.WriteLine("===== ");
System.Console.WriteLine(" Hasil penjumlahan : " + A);
System.Console.WriteLine(" Hasil pengurangan : " + B);
System.Console.WriteLine(" Hasil perkalian : " + C);
System.Console.WriteLine(" Hasil pembagian : " + D);
System.Console.WriteLine("===== ");
System.Console.WriteLine(" Terima Kasih @Created By Fauziah Kasyfi 2015 ");
System.Console.WriteLine("===== ");
System.Console.ReadLine();

}
}
}
```



```
C:\Windows\system32\cmd.exe
=====
Menggunakan Operator Aritmatika Dengan Data Konstan
=====
      Hasil penjumlahan : 13
      Hasil pengurangan : 5
      Hasil perkalian   : 36
      Hasil pembagian   : 2
=====
Terima Kasih @Created By Fauziah Kasyfi 2015
=====
```

Gambar 3.9 Tampilan *Output Operator Aritmatika* Pada Program C# Menggunakan Data

- **Contoh Program C# Menggunakan Operator Logika Dengan Data Inputan**

```
//Program 3.10
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace operator1
{
    class Program
    {
        static void Main(string[] args)
        {
            // Deklarasi Nilai Input a dan b
            int a, b;
            // Deklarasi Nilai Untuk Penyimpanan Hasil Penjumlahan[X],
            Pengurangan[Y]
            // Perkalian[Z], Pembagian[V] dan Modulo[W]
            int X, Y, Z, W;
            decimal V;
            // Proses input Nilai a dan b
            System.Console.WriteLine("=====
```

```
// Proses Kalkulasi Dengan Operator Aritmatika
```

```
X = a + b;
```

```
Y = a - b;
```

```
Z = a * b;
```

```
V = a / b;
```

```
W = a % b;
```

```
// Mencetak Hasil Keluaran
```

```
System.Console.WriteLine("=====  
====");
```

```
System.Console.WriteLine("Mencetak Hasil Dengan Menggunakan Operator Aritmatika");
```

```
System.Console.WriteLine("=====  
====");
```

```
System.Console.WriteLine("Hasil penjumlahan      : " + X);
```

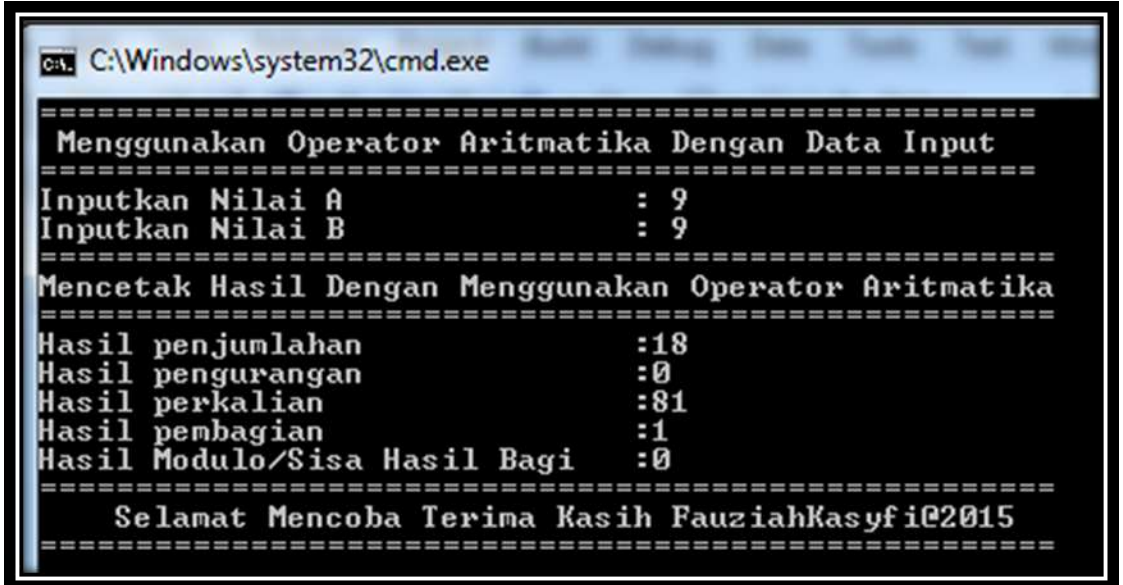
```
System.Console.WriteLine("Hasil pengurangan    : " + Y);
```

```
System.Console.WriteLine("Hasil perkalian      : " + Z);
```

```
System.Console.WriteLine("Hasil pembagian     : " + V);
```

```
System.Console.WriteLine("Hasil Modulo/Sisa Hasil Bagi : " + W);
```

```
System.Console.WriteLine("=====  
====");
```



```
C:\Windows\system32\cmd.exe

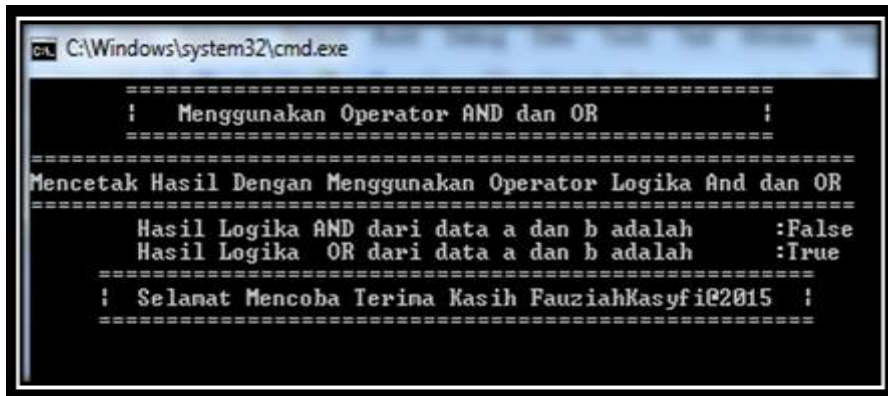
=====
Menggunakan Operator Aritmatika Dengan Data Input
=====
Inputkan Nilai A      : 9
Inputkan Nilai B      : 9
=====
Mencetak Hasil Dengan Menggunakan Operator Aritmatika
=====
Hasil penjumlahan     :18
Hasil pengurangan     :0
Hasil perkalian       :81
Hasil pembagian       :1
Hasil Modulo/Sisa Hasil Bagi :0
=====
Selamat Mencoba Terima Kasih FauziahKasyfi@2015
=====
```

Gambar 3.10 Tampilan *Output Operator Aritmatika Menggunakan Data Inputan*.

# Contoh Program C# Menggunakan Logika And & Or

```
//Program 3.11
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace operator2
{
    class Program
    {
        static void Main(string[] args)
        {
            // Deklarasi Nilai Konstan a dan b
            int a= 100, b=20;
            // Deklarasi Penyimpanan Nilai dengan Variabel X dan W
            Boolean X, W;
            System.Console.WriteLine(" =====");
            System.Console.WriteLine(" Menggunakan Operator AND dan OR " );
            System.Console.WriteLine(" =====");
            // Proses Analisis Logika Dengan Operator AND dan OR
            X= (a<=100 && b>=100);
            W= (a<=100 || b>=100);
            // Mencetak Hasil Keluaran
            System.Console.WriteLine("=====
            ====");
            System.Console.WriteLine("Mencetak Hasil Dengan Menggunakan Operator Logika And dan
            OR");
            System.Console.WriteLine("=====
```

Dari kode program 3.11 terlihat Menggunakan *Operator Logika And* dan *Or* dengan nilai *variable* konstan yaitu nilai yang sudah ditentukan untuk *variable*  $a=100$  dan  $b = 20$ , sedangkan untuk perbandingan logika menggunakan *variable*  $X$  dan  $W$ , dimana nilai  $X= (a \leq 100 \ \&\& \ b \geq 100)$ ; artinya melakukan pengecekan nilai : apakah nilai  $a \leq 100$  (*True/Benar*) dan nilai  $b \geq 100$  (*False/Salah*), dari pernyataan tersebut, maka hasil yang diperoleh adalah *False/Salah*. Langkah Selanjutnya adalah melakukan pengecekan untuk nilai  $W= (a \leq 100 \ || \ b \geq 100)$ ; artinya apakah nilai  $a \leq 100$  (*True/Benar*) atau nilai  $b \geq 100$  (*False/Salah*). Karena Menggunakan logika *Or* maka



```
cmd C:\Windows\system32\cmd.exe
=====
: Menggunakan Operator AND dan OR :
=====
Mencetak Hasil Dengan Menggunakan Operator Logika And dan OR
=====
Hasil Logika AND dari data a dan b adalah :False
Hasil Logika OR dari data a dan b adalah :True
=====
: Selamat Mencoba Terima Kasih FauziahKasyfi02015 :
=====
```

Gambar 3.11 Tampilan Output Logika Operator And dan Or Menggunakan Variabel Konstan

```
//Program 3.12
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace operator4
{
    class Program
    {
        static void Main(string[] args)
        {
            // Deklarasi Nilai Input a dan b
            int a, b;
            // Deklarasi Penyimpanan Nilai dengan Variabel X dan W
            Boolean X, W;
            // Proses input Nilai a dan b

System.Console.WriteLine("=====");
System.Console.WriteLine(" Menggunakan Operator Logika And dan Or
Dengan Data Input ");
System.Console.WriteLine("=====");
System.Console.Write( "    Inputkan Nilai A                : ");
a = int.Parse(System.Console.In.ReadLine());
System.Console.Write( "    Inputkan Nilai B                : ");
b = int.Parse(System.Console.In.ReadLine());
// Proses Analisis Logika Dengan Operator AND dan OR
X= (a<=100 && b>=100);
W= (a<=100 || b>=100);
// Mencetak Hasil Keluaran
System.Console.WriteLine("=====");
System.Console.WriteLine("Mencetak Hasil Dengan Menggunakan Operator
Logika And dan OR");
System.Console.WriteLine("=====");
System.Console.WriteLine("Hasil Logika AND dari data A dan B adalah
:" + X);
System.Console.WriteLine("Hasil Logika OR dari data A dan B adalah
:" + W);
System.Console.WriteLine(" =====");
System.Console.WriteLine("Selamat Mencoba Terima Kasih
FauziahKasyfi@2015 ");
System.Console.WriteLine("=====");
        System.Console.ReadLine();
    }
}
}
```

Dari kode program 3.12 terlihat Menggunakan *Operator Logika And* dan *Or* dengan nilai *variable inputan* yaitu nilai *variable a* dan *b*, sedangkan untuk perbandingan logika menggunakan *variable X* dan *W*, dimana nilai  $X = (a \leq 100 \ \&\& \ b \geq 100)$ ; artinya melakukan pengecekan nilai. Pengecekan nilai pada program 3.12 berbeda dengan program 3.11, Pada program 3.12 nilai hasil *True* atau *False* didapatkan berdasarkan nilai *a* dan *b* yang kita *inputkan*. Untuk lebih jelasnya dapat kita lihat pada *output*

Gambar 3.12 Tampilan *Output Operator Logika And* dan *Or* Menggunakan Data *Input*



```
// Program 3.13
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace operator5
{
    class Program
    {
        static void Main(string[] args)
        {
            // Deklarasi Nilai Input a dan b
            int a, b;
            // Proses input Nilai a dan b

            System.Console.WriteLine("=====
            ===");
            System.Console.WriteLine("Menggunakan Operator Increment & Decrement Dengan Data
            Konstan");
            System.Console.WriteLine("=====
            ===");
            System.Console.Write("          Inputkan Nilai A          : ");
            a = int.Parse(System.Console.In.ReadLine());
            System.Console.Write("          Inputkan Nilai B          : ");
            b = int.Parse(System.Console.In.ReadLine());
            // Proses Analisis Logika Dengan Operator Increment
            a++;
            b++;
            // Mencetak Hasil Keluaran
            System.Console.WriteLine("=====
            ===");
            System.Console.WriteLine("Mencetak Hasil Dengan Menggunakan Increment & Decrement");
            System.Console.WriteLine("=====
            ===");
        }
    }
}
```

Dari kode program 3.13 terlihat Menggunakan *Operator Increment* dan *Decrement* dengan nilai *variable inputan* yaitu nilai *variable a* dan *b*, Kemudian Proses pengecekan *logika increment* dan *decrement* berdasarkan data yang *diinput*. Untuk lebih jelasnya dapat kita lihat pada *output*



Gambar 3.13 Tampilan Output Operator Logika Increment Decrement  
Menggunakan Data Input

```
// Program 3.14
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace operator6
{
    class Program
    {
        static void Main(string[] args)
        {
            // Deklarasi Nilai Varibel Konstan a dan b
            int a=8, b=9;

            System.Console.WriteLine("=====
            ===");
            System.Console.WriteLine("Menggunakan Operator Increment & Decrement Dengan Data
            Konstan");
            System.Console.WriteLine("=====
            ===");
            System.Console.Write(" Nilai A Dengan Data Konstan Adalah      : "+a);
            System.Console.Write(" Nilai B Dengan Data Konstan Adalah      : "+b);
            // Proses Analisis Logika Dengan Operator Increment
            a++;
            b++;
            // Mencetak Hasil Keluaran
            System.Console.WriteLine("=====
            ===");
            System.Console.WriteLine("Mencetak Hasil Dengan Menggunakan Increment & Decrement");
            System.Console.WriteLine("=====
            ===");
```

Dari kode program 3.14 terlihat Menggunakan *Operator Increment* dan *Decrement* dengan nilai *variable konstan* yaitu nilai *variable a=8* dan *b=9*, Kemudian Proses pengecekan *logika increment* dan *decrement* berdasarkan data *konstan*. Untuk lebih jelasnya dapat



```
C:\Windows\system32\cmd.exe
=====
Menggunakan Operator Increment & Decrement Dengan Data Konstan
=====
Nilai A Dengan Data Konstan Adalah      : 8
Nilai B Dengan Data Konstan Adalah      : 9
=====
Mencetak Hasil Dengan Menggunakan Increment & Decrement
=====
Hasil Nilai A Sesudah Operasi Increment Adalah:9
Hasil Nilai B Sesudah Operasi Increment Adalah:10
Hasil Nilai A Sesudah Operasi Decrement Adalah:8
Hasil Nilai B Sesudah Operasi Decrement Adalah:9
=====
Selamat Mencoba Terima Kasih FauziahKasyfi@2015
=====
```

Gambar 3.14 Tampilan Output Operator Logika Increment Decrement  
Menggunakan Data Konstan

# Contoh Program C# Dengan Kombinasi Operator

```
// Program 3.15
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace KombinasiOp
{
    class Program
    {
        static void Main(string[] args)
        {
            String NAMA, KLS, NPM, SKS;
            // Deklarasi Variabel Untuk Mata Kuliah Aplikasi Manajemen Perkantoran
            // Terdiri dari Nilai UTS,UAS, Tugas, Kehadiran dan Nilai Akhir AMP
            Int32 AMPuts, AMPuas, AMPtgs, AMPhdr, NAAMP;
            // Deklarasi Variabel Untuk Mata Kuliah Matematika Teknologi Informasi
            // Terdiri dari Nilai UTS,UAS, Tugas, Kehadiran dan Nilai Akhir MTI
            Int32 MTKuts,MTKuas,MTKtgs,MTKhdr,NAMTK;
            // Deklarasi Variabel Untuk Mata Kuliah Matematika Lanjut
            // Terdiri dari Nilai UTS,UAS, Tugas, Kehadiran dan Nilai Akhir
            Matematika Lanjut
            Int32 MATLANuts,MATLANuas,MATLANtgs,MATLANhdr,NAMATLAN;
            // Deklarasi Variabel Untuk Mata Kuliah Listrik Magnet
            // Terdiri dari Nilai UTS,UAS, Tugas, Kehadiran dan Nilai Akhir Listrik
            Magnet
            Int32 LMuts, LMuas, LMTgs, LMhdr, NALM;
            // Deklarasi Variabel Untuk Mata Kuliah Praktikum Listrik Magnet
            // Terdiri dari Nilai UTS,UAS, Tugas, Kehadiran dan Nilai Akhir
            Praktikum Listrik Magnet
            Int32 PLMuts, PLMuas, PLMtgs, PLMhdr, NAPLM;
            // Deklarasi Variabel Untuk Mata Kuliah Pendidikan Kewarganegaraan
            // Terdiri dari Nilai UTS,UAS, Tugas, Kehadiran dan Nilai Akhir
            Pendidikan Kewarganegaraan
            Int32 PKNuts, PKNuas, PKNtgs, PKNhdr, NAPKN;
            // Deklarasi Variabel Untuk Mata Kuliah Pengantar Teknologi Informasi
            // Terdiri dari Nilai UTS,UAS, Tugas, Kehadiran dan Nilai Akhir
            Pengantar Teknologi Informasi
            Int32 PTIuts, PTIuas, PTItgs, PTIhdr, NAPTI;
            // Deklarasi Variabel Untuk Mata Kuliah Bahasa Indonesia
            // Terdiri dari Nilai UTS,UAS, Tugas, Kehadiran dan Nilai Akhir Bahasa
            Indonesia
            Int32 BInuts, BInuas, BIntgs, BInhdr, NABIn;
```

```
// Deklarasi Variabel Untuk Mata Kuliah Bahasa Inggris
// Terdiri dari Nilai UTS,UAS, Tugas, Kehadiran dan Nilai Akhir toefl
Int32 TPut, TPuas, TPtgs, TPhdr, NATP;
//Deklarasi Variabel Untuk Mata Kuliah Praktikum Algoritma Pemrograman
I
// Terdiri dari Nilai UTS,UAS, Tugas, Kehadiran dan Nilai Akhir
Algoritma Pemrograman I
Int32 APut, APuas, APtgs, APhdr, NAAP;
// Deklarasi Variabel Untuk Mata Kuliah Praktikum Algoritma Pemrograman
I
// Terdiri dari Nilai UTS,UAS, Tugas, Kehadiran dan Nilai Akhir
Praktikum Algoritma Pemrograman I
Int32 PAPut, PAPuas, PAPtgs, PAPhdr, NAPAP;
// Prose Input Data Nama, Kelas, NPM dan Jumlah SKS
Console.WriteLine("          TRANSKIP NILAI AKADEMIK          ");
Console.WriteLine("=====");
Console.Write("    Inputkan Nama  Anda          : ");
NAMA = Console.ReadLine();
Console.Write("    Inputkan Kelas  Anda          : ");
KLS = Console.ReadLine();
Console.Write("    Inputkan NPM   Anda          : ");
NPM = Console.ReadLine();
Console.Write("    Inputkan Jumlah SKS Anda      : ");
SKS = Console.ReadLine();
// Proses Input Untuk Nilai Mata Kuliah AMP
Console.WriteLine("          Berikut Terlampir Nama Mata Kuliah   : ");
Console.WriteLine("\n\tAplikasi Manajemen Perkantoran");
Console.Write("\t\t Inputkan Nilai UTS          : ");
AMPut = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai UAS          : ");
AMPuas = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai Tugas          : ");
AMPtgs = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai Kehadiran : ");
AMPhdr = Convert.ToInt32(Console.ReadLine());
NAAMP = (AMPut + AMPuas + AMPtgs + AMPhdr)/4;
Console.WriteLine("\tNilai Akhir Mata Kuliah Aplikasi Manajemen
Perkantoran : " + NAAMP);
//Proses Input Untuk Nilai Mata Kuliah MTI
Console.WriteLine("\n\tMatematika Teknologi Informasi I");
Console.Write("\t\t Inputkan Nilai UTS          : ");
MTKut = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai UAS          : ");
MTKuas = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai Tugas          : ");
MTKtgs = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai Kehadiran : ");
MTKhdr = Convert.ToInt32(Console.ReadLine());
NAMTK = (MTKut + MTKuas + MTKtgs + MTKhdr) / 4;
Console.WriteLine("\tNilai Akhir Mata Kuliah Matematika Teknologi
Informasi I : " + NAMTK);
```

```
//Proses Input Untuk Nilai Mata Kuliah MAT LAN
Console.WriteLine("\n\tMatematika Lanjut I");
Console.Write("\t\t Inputkan Nilai UTS      : ");
MATLANuts = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai UAS      : ");
MATLANuas = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai Tugas    : ");
MATLANtgs = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai Kehadiran : ");
MATLANhdr = Convert.ToInt32(Console.ReadLine());
NAMATLAN = (MATLANuts + MATLANuas + MATLANtgs +MATLANhdr) / 4;
Console.WriteLine("\tNilai Akhir Mata Kuliah Matematika Lanjut      : "
+ NAMATLAN);
//Proses Input Untuk Nilai Mata Kuliah Listrik Magnet
Console.WriteLine("\n\tListrik dan Magnet");
Console.Write("\t\t Inputkan Nilai UTS      : ");
LMuts = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai UAS      : ");
LMuas = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai Tugas    : ");
LMtgs = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai Kehadiran : ");
LMhdr = Convert.ToInt32(Console.ReadLine());
NALM = (LMuts + LMuas + LMtgs + LMhdr) / 4;
Console.WriteLine("\tNilai Akhir Mata Kuliah Listrik dan Magnet : " +
NALM);
//Proses Input Untuk Nilai Mata Kuliah Prak. Listrik Magnet
Console.WriteLine("\n\tPraktikum Listrik dan Magnet");
Console.Write("\t\t Inputkan Nilai UTS      : ");
PLMuts = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai UAS      : ");
PLMuas = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai Tugas    : ");
PLMtgs = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai Kehadiran : ");
PLMhdr = Convert.ToInt32(Console.ReadLine());
NAPLM = (PLMuts + PLMuas + PLMtgs + PLMhdr) / 4;
Console.WriteLine("\tNilai Akhir Mata Kuliah Praktikum Listrik dan
Magnet      : " + NAPLM);
//Proses Input Untuk Nilai Mata Kuliah PKN
Console.WriteLine("\n\tPendidikan Kewarganegaraan");
Console.Write("\t\t Inputkan Nilai UTS      : ");
PKNuts = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai UAS      : ");
PKNuas = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai Tugas    : ");
PKNtgs = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai Kehadiran : ");
PKNhdr = Convert.ToInt32(Console.ReadLine());
NAPKN = (PKNuts + PKNuas + PKNtgs + PKNhdr) / 4;

Console.WriteLine("\tNilai Akhir Mata Kuliah Pendidikan Kewarganegaraan
: " + NAPKN);
```

```
//Proses Input Untuk Nilai Mata Kuliah PTI
Console.WriteLine("\n\tPengantar Teknologi Komunikasi dan Informasi");
Console.Write("\t\t Inputkan Nilai UTS      : ");
PTIuts = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai UAS      : ");
PTIuas = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai Tugas      : ");
PTItgs = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai Kehadiran : ");
PTIhdr = Convert.ToInt32(Console.ReadLine());
NAPTI = (PTIuts + PTIuas + PTItgs + PTIhdr) / 4;
Console.WriteLine("\tNilai Akhir Mata Kuliah Pengantar Tek.Kom. dan
Inf.      : " + NAPTI);
//Proses Input Untuk Nilai Mata Kuliah B.Indonesia
Console.WriteLine("\n\tPengantar Bahasa Indonesia");
Console.Write("\t\t Inputkan Nilai UTS      : ");
BINuts = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai UAS      : ");
BINuas = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai Tugas      : ");
BIntgs = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai Kehadiran : ");
BINhdr = Convert.ToInt32(Console.ReadLine());
NABIn = (BINuts + BINuas + BIntgs + BINhdr) / 4;
Console.WriteLine("\t Nilai Akhir Bahasa Indonesia : " + NABIn);
//Proses Input Untuk Nilai Mata Kuliah  Prak. Algoritma Pemrograman I
Console.WriteLine("\n\tPraktikum Algoritma dan Pemrograman I");
Console.Write("\t\t Inputkan Nilai UTS      : ");
PAPuts = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai UAS      : ");
PAPuas = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai Tugas      : ");
PAPtgs = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai Kehadiran : ");
PAPhdr = Convert.ToInt32(Console.ReadLine());
NAPAP = (PAPuts + PAPuas + PAPtgs + PAPhdr) / 4;
Console.WriteLine("\tNilai Akhir Mata Kuliah Praktikum Alg.& Pem. I
: " + NAPAP);
//Proses Input Untuk Nilai Mata Kuliah Toefl Preparation
Console.WriteLine("\n\tTOEFL Preparation I");
Console.Write("\t\t Inputkan Nilai UTS      : ");
TPuts = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai UAS      : ");
TPuas = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai Tugas      : ");
TPtgs = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai Kehadiran : ");
TPhdr = Convert.ToInt32(Console.ReadLine());
NATP = (TPuts + TPuas + TPtgs + TPhdr) / 4;
Console.WriteLine("\t\t\tNilai TOEFL Preparation I : " + NATP);
```



```
//Proses Input Untuk Nilai Mata Kuliah Algoritma Pemrograman I
Console.WriteLine("\n\tAlgoritma dan Pemrograman Informatika I");
Console.Write("\t\t Inputkan Nilai UTS      : ");
APuts = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai UAS      : ");
APuas = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai Tugas    : ");
APTgs = Convert.ToInt32(Console.ReadLine());
Console.Write("\t\t Inputkan Nilai Kehadiran : ");
APhdr = Convert.ToInt32(Console.ReadLine());
NAAP = (APuts + APuas + APTgs + APhdr) / 4;
Console.WriteLine("\tNilai Akhir Mata Kuliah Alg.& Pem.I : " + NAAP);
Console.WriteLine("=====|");
Console.WriteLine("| REKAPITULASI NILAI AKHIR DARI SEMUA MATA KULIAH
SEMESTER I |");
Console.WriteLine("=====|");
Console.WriteLine("Nilai Akhir Mata Kuliah Aplikasi Manajemen Perkantoran
Adalah :"+NAAMP);
Console.WriteLine("Nilai Akhir Mata Kuliah Matematika Teknologi Informasi
Adalah :"+NAMTK);
Console.WriteLine("Nilai Akhir Mata Kuliah Matematika Lanjut Adalah
:"+NAMATLAN);
Console.WriteLine("Nilai Akhir Mata Kuliah Pengantar Teknologi Informasi
Adalah :"+NAPTI);
Console.WriteLine("Nilai Akhir Mata Kuliah Listrik Magnet Adalah
:"+NALM);
Console.WriteLine("Nilai Akhir Mata Kuliah Praktikum Listrik Magnet Adalah
:"+NAPLM);
Console.WriteLine("Nilai Akhir Mata Kuliah Bahasa Indonesia Adalah
:"+NABIn);
Console.WriteLine("Nilai Akhir Mata Kuliah B. Inggris Toefl Preparation
Adalah :"+NATP);
Console.WriteLine("Nilai Akhir Mata Kuliah Algoritma dan Pemrograman I
Adalah :"+NAAP);
Console.WriteLine("Nilai Akhir Mata Kuliah Prak.Algoritma dan Pemrog. I
Adalah :"+NAPAP);
Console.WriteLine("|=====");

    }
}
}
```

# Keterangan Kode Program

## Keterangan Kode Program 3.15

Dari program 3.15 terlihat ada banyak *variable* yang dideklarasikan dengan data *inputan* yaitu :

1. Tipe data *string* digunakan untuk menginput Nama, Kelas, NPM dan SKS.
2. Tipe Data *Integer* digunakan menginput nilai setiap mata kuliah dengan komponen nilai uts, nilai uas, nilai tugas, nilai kehadiran, dan Nilai Akhir. Pada contoh program 3.15 terlihat ada 11 komponen mata kuliah yang terdiri dari : Mata Kuliah Aplikasi Manajemen Perkantoran, Matematika Teknologi Informasi, Matematika Lanjut, Bahasa Inggris (*Toefl Preparation*), Bahasa Indonesia, Fisika Listrik dan Magnet, Praktikum Fisika Listrik Magnet, Pengantar Teknologi Informasi dan Komunikasi, Algoritma Pemrograman I, Prak. Algoritma Pemrograman I, dan Pendidikan Kewarganegaraan.
3. Untuk Menghitung komponen nilai akhir, rumus yang digunakan adalah dengan mencari nilai rata-rata dari komponen nilai yang diinputkan



```
Maka output dari program 3.15 adalah sebagai berikut :
=====
TRANSKIP NILAI AKADEMIK
=====
Inputkan Nama Anda      : Tsarwan Kasyfillah Nst
Inputkan Kelas Anda     : A01
Inputkan NPM Anda      : 11215001
Inputkan Jumlah SKS Anda : 24
Berikut Terlampir Nama Mata Kuliah :

Aplikasi Manajemen Perkantoran
    Inputkan Nilai UTS      : 100
    Inputkan Nilai UAS     : 100
    Inputkan Nilai Tugas   : 100
    Inputkan Nilai Kehadiran : 100
Nilai Akhir Mata Kuliah Aplikasi Manajemen Perkantoran : 100

Matematika Teknologi Informasi I
    Inputkan Nilai UTS      : 90
    Inputkan Nilai UAS     : 100
    Inputkan Nilai Tugas   : 100
    Inputkan Nilai Kehadiran : 100
Nilai Akhir Mata Kuliah Matematika Teknologi Informasi I : 97

Matematika Lanjut I
    Inputkan Nilai UTS      : 90
    Inputkan Nilai UAS     : 100
    Inputkan Nilai Tugas   : 100
    Inputkan Nilai Kehadiran : 100
Nilai Akhir Mata Kuliah Matematika Lanjut : 97

Listrik dan Magnet
    Inputkan Nilai UTS      : 90
    Inputkan Nilai UAS     : 100
    Inputkan Nilai Tugas   : 100
    Inputkan Nilai Kehadiran : 100
Nilai Akhir Mata Kuliah Listrik dan Magnet : 97

Praktikum Listrik dan Magnet
    Inputkan Nilai UTS      : 100
    Inputkan Nilai UAS     : 100
    Inputkan Nilai Tugas   : 100
    Inputkan Nilai Kehadiran : 100
Nilai Akhir Mata Kuliah Praktikum Listrik dan Magnet : 100

Pendidikan Kewarganegaraan
    Inputkan Nilai UTS      : 100
    Inputkan Nilai UAS     : 100
    Inputkan Nilai Tugas   : 100
    Inputkan Nilai Kehadiran : 100
Nilai Akhir Mata Kuliah Pendidikan Kewarganegaraan : 100

Pengantar Teknologi Komunikasi dan Informasi
    Inputkan Nilai UTS      : 100
    Inputkan Nilai UAS     : 100
    Inputkan Nilai Tugas   : 100
    Inputkan Nilai Kehadiran : 100
Nilai Akhir Mata Kuliah Pengantar Tek.Kom. dan Inf. : 100
```

```
C:\Windows\system32\cmd.exe

Pengantar Bahasa Indonesia
  Inputkan Nilai UTS      : 100
  Inputkan Nilai UAS      : 100
  Inputkan Nilai Tugas    : 100
  Inputkan Nilai Kehadiran : 100
  Nilai Akhir Bahasa Indonesia : 100

Praktikum Algoritma dan Penrograman I
  Inputkan Nilai UTS      : 90
  Inputkan Nilai UAS      : 100
  Inputkan Nilai Tugas    : 100
  Inputkan Nilai Kehadiran : 100
  Nilai Akhir Mata Kuliah Praktikum Alg.& Pem. I      : 97

TOEFL Preparation I
  Inputkan Nilai UTS      : 100
  Inputkan Nilai UAS      : 100
  Inputkan Nilai Tugas    : 100
  Inputkan Nilai Kehadiran : 100
  Nilai TOEFL Preparation I : 100

Algoritma dan Penrograman Informatika I
  Inputkan Nilai UTS      : 90
  Inputkan Nilai UAS      : 100
  Inputkan Nilai Tugas    : 100
  Inputkan Nilai Kehadiran : 100
  Nilai Akhir Mata Kuliah Alg.& Pem.I      : 97
=====
: REKAPITULASI NILAI AKHIR DARI SEMUA MATA KULIAH SEMESTER I :
=====
Nilai Akhir Mata Kuliah Aplikasi Manajemen Perkantoran Adalah :100
Nilai Akhir Mata Kuliah Matematika Teknologi Informasi Adalah :97
Nilai Akhir Mata Kuliah Matematika Lanjut Adalah :97
Nilai Akhir Mata Kuliah Pengantar Teknologi Informasi Adalah :100
Nilai Akhir Mata Kuliah Listrik Magnet Adalah :97
Nilai Akhir Mata Kuliah Praktikum Listrik Magnet Adalah :100
Nilai Akhir Mata Kuliah Bahasa Indonesia Adalah :100
Nilai Akhir Mata Kuliah B. Inggris Toefl Preparation Adalah :100
Nilai Akhir Mata Kuliah Algoritma dan Penrograman I Adalah :97
Nilai Akhir Mata Kuliah Prak.Algoritma dan Pemrog. I Adalah :97
=====
Press any key to continue . . . _
```

Gambar 3.15 Tampilan Output Menggunakan Kombinasi Operator Aritmatika

# CONTOH PROGRAM OPERATOR ARITMATIKA DENGAN JAVA

- Contoh Program Operator Aritmatika Dengan Java

## // Program 3.16

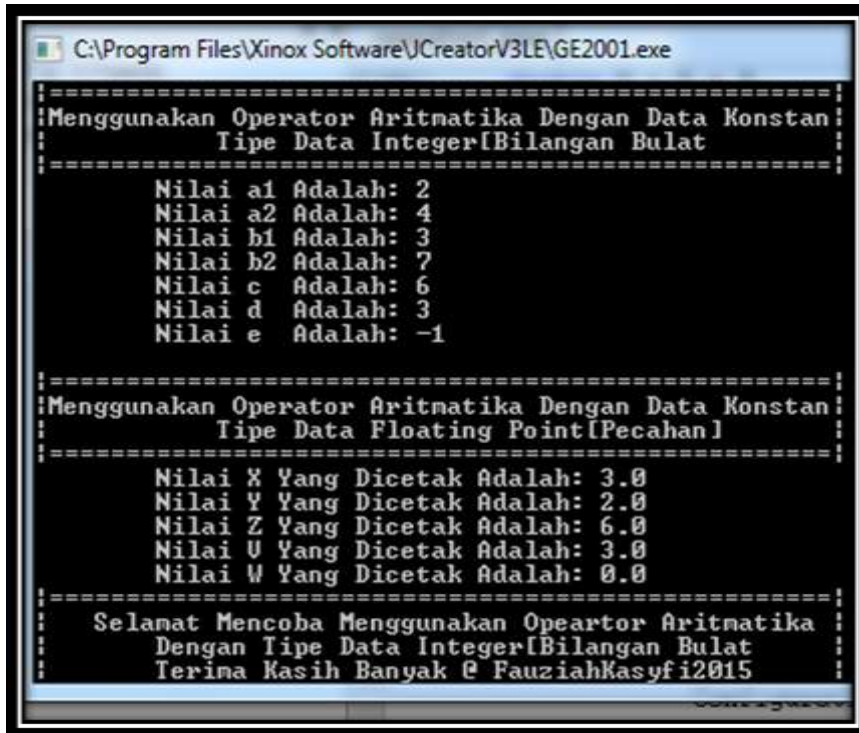
```
class operator1 {
public static void main(String[] args) {
System.out.println("|=====|");
System.out.println("|Menggunakan Operator Aritmatika Dengan Data Konstan|");
System.out.println("|   Tipe Data Integer[Bilangan Bulat   |");
System.out.println("|=====|");
;

// Deklarasi Variabel Konstan
int a1 = 2;
int a2 = 2+a1;
int b1 = a2 - 1;
int b2 = b1+a2;
int c = a1 * b1;
int d = c / 2;
int e = -a2 + 3 ;

System.out.println("Nilai a1 Adalah: " + a1);
System.out.println("Nilai a2 Adalah: " + a2);
System.out.println("Nilai b1 Adalah: " + b1);
System.out.println("Nilai b2 Adalah: " + b2);
```

```
System.out.println("| Tipe Data Floating Point[Pecahan]   |");
System.out.println("=====
|");
    double X = 2 + 1;
    double Y = X - 1;
    double Z = Y * X;
    double V = Z / 2;
    double W = V-X;

    System.out.println("Nilai X Yang Dicitak Adalah: " + X);
    System.out.println("Nilai Y Yang Dicitak Adalah: " + Y);
    System.out.println("Nilai Z Yang Dicitak Adalah: " + Z);
    System.out.println("Nilai V Yang Dicitak Adalah: " + V);
    System.out.println("Nilai W Yang Dicitak Adalah: " + W);
System.out.println("=====|")
);
System.out.println("| Selamat Mencoba Menggunakan Opeartor Aritmatika
|");
System.out.println("|Dengan Tipe Data Integer[Bilangan Bulat   |");
System.out.println("|Terima Kasih Banyak @ FauziahKasyfi2015   |");
```



```
C:\Program Files\Xinox Software\UCreatorV3LE\GE2001.exe
=====
: Menggunakan Operator Aritmatika Dengan Data Konstan
: Tipe Data Integer[Bilangan Bulat
: =====
:
: Nilai a1 Adalah: 2
: Nilai a2 Adalah: 4
: Nilai b1 Adalah: 3
: Nilai b2 Adalah: 7
: Nilai c Adalah: 6
: Nilai d Adalah: 3
: Nilai e Adalah: -1
:
: =====
: Menggunakan Operator Aritmatika Dengan Data Konstan
: Tipe Data Floating Point[Pecahan]
: =====
:
: Nilai X Yang Dicetak Adalah: 3.0
: Nilai Y Yang Dicetak Adalah: 2.0
: Nilai Z Yang Dicetak Adalah: 6.0
: Nilai U Yang Dicetak Adalah: 3.0
: Nilai W Yang Dicetak Adalah: 0.0
:
: =====
: Selamat Mencoba Menggunakan Opeartor Aritmatika
: Dengan Tipe Data Integer[Bilangan Bulat
: Terima Kasih Banyak @ FauziahKasyfi2015
: =====
```

Gambar 3.16 Tampilan Output Menggunakan Operator Aritmatika Variabel Konstan



- Contoh Program *Operator Aritmatika* Dengan Java Menggunakan *Data Input*

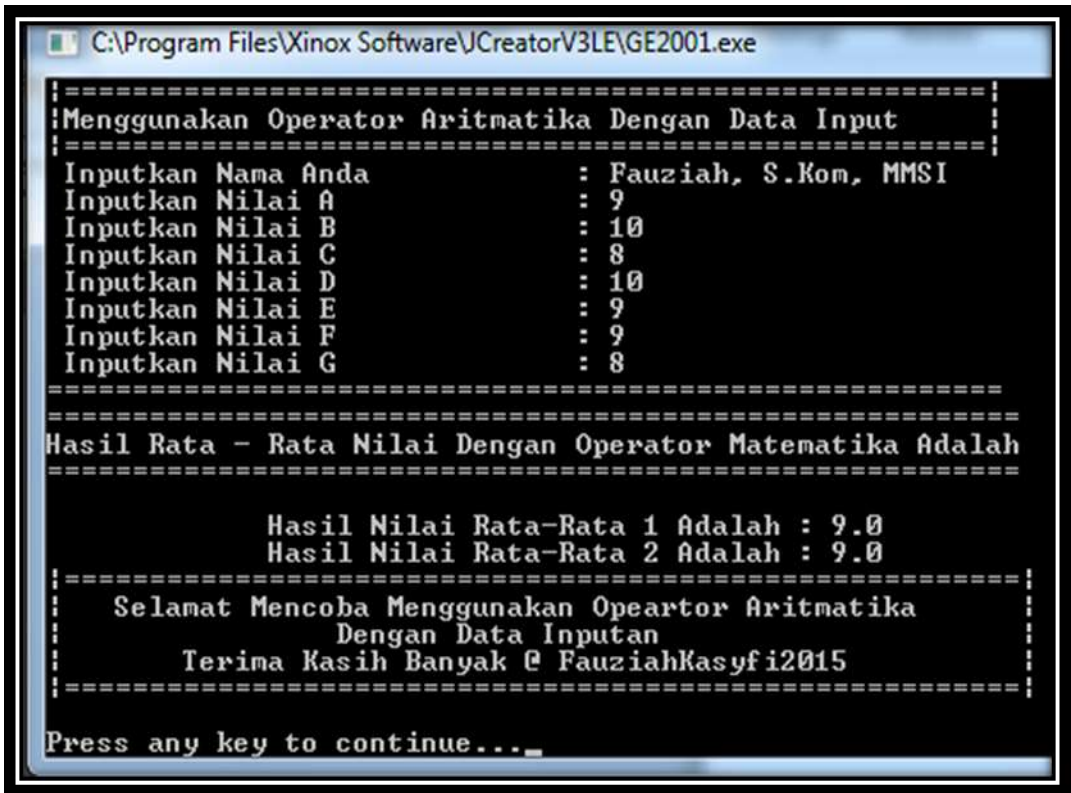
// Program 3.17

```
import java.io.*;
public class operator2 {
public static void main(String[] args)throws Exception {
BufferedReader input = new BufferedReader
(new InputStreamReader (System.in));
String nama,NA,NB,NC,ND,NE, NF, NG;
double a,b,c,d,e,f,g,x,y;
System.out.println("|=====
===|");
System.out.println("|Menggunakan Operator Aritmatika Dengan Data Input  |");
System.out.println("|=====
=|");
System.out.print (" Inputkan Nama Anda      : ");
nama = input.readLine();
System.out.print (" Inputkan Nilai A      : ");
NA = input.readLine();
System.out.print (" Inputkan Nilai B      : ");
NB = input.readLine();
System.out.print (" Inputkan Nilai C      : ");
NC = input.readLine();
System.out.print (" Inputkan Nilai D      : ");
ND = input.readLine();
System.out.print (" Inputkan Nilai E      : ");
```

```
c = Double.parseDouble(NC);
d = Double.parseDouble(ND);
e = Double.parseDouble(NE);
f = Double.parseDouble(NF);
g = Double.parseDouble(NG);

x = (a+b+c)/3;
y = (d+e+f+g)/4;

System.out.println("=====
===");
System.out.println("Hasil Rata - Rata Nilai Dengan Operator Matematika Adalah");
System.out.println("=====
===");
System.out.println("");
System.out.println("        Hasil Nilai Rata-Rata 1 Adalah : " + x);
System.out.println("        Hasil Nilai Rata-Rata 2 Adalah : " + y);
System.out.println("|=====
==|");
System.out.println("| Selamat Mencoba Menggunakan Opeartor Aritmatika    |");
System.out.println("|        Dengan Data Inputan                |");
System.out.println("| Terima Kasih Banyak @ FauziahKasyfi2015    |");
System.out.println("|=====
==|");
System.out.println("");
```



```
=====
: Menggunakan Operator Aritmatika Dengan Data Input
: =====
Inputkan Nama Anda           : Fauziah, S.Kom, MMSI
Inputkan Nilai A             : 9
Inputkan Nilai B             : 10
Inputkan Nilai C             : 8
Inputkan Nilai D             : 10
Inputkan Nilai E             : 9
Inputkan Nilai F             : 9
Inputkan Nilai G             : 8
=====
Hasil Rata - Rata Nilai Dengan Operator Matematika Adalah
=====
                Hasil Nilai Rata-Rata 1 Adalah : 9.0
                Hasil Nilai Rata-Rata 2 Adalah : 9.0
=====
                Selamat Mencoba Menggunakan Opeartor Aritmatika
                Dengan Data Inputan
                Terima Kasih Banyak © FauziahKasyfi2015
=====
Press any key to continue..._
```

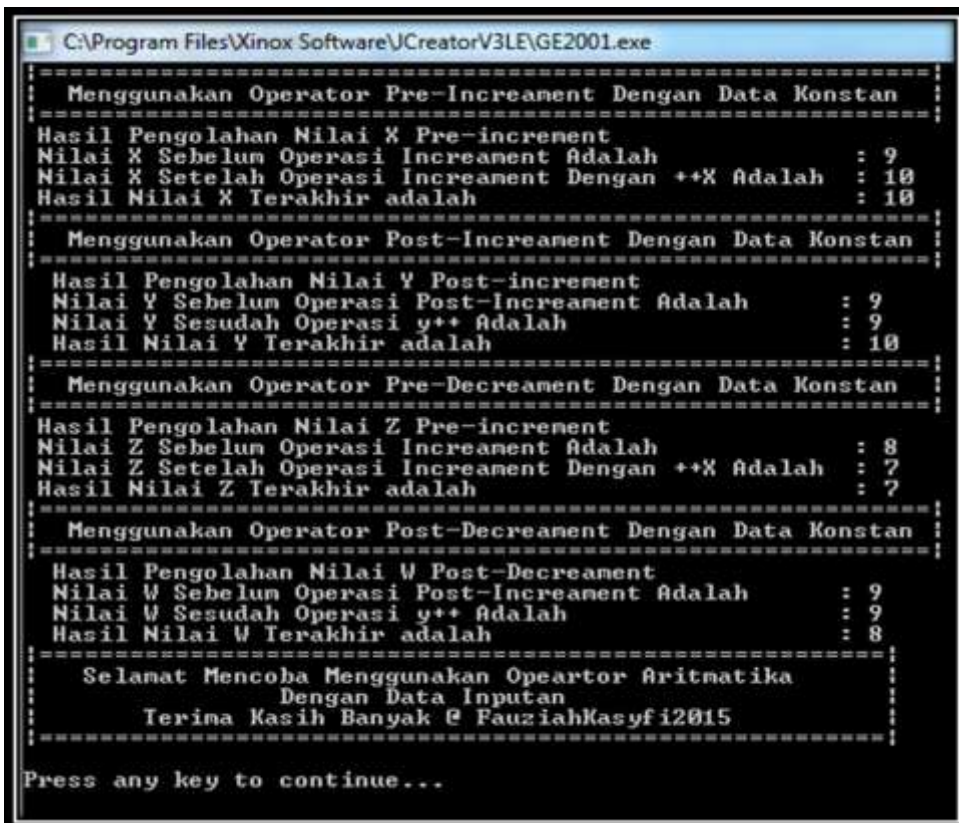
Gambar 3.17 Tampilan Output *Operator* Matematika Menggunakan Data *Inputan*

## Decrement

```
// Program 3.18
class operator3 {
public static void main(String[] args) {
int x=9;
System.out.println("=====
|");
System.out.println(" Menggunakan Operator Pre-Increment Dengan Data Konstan |");
System.out.println("=====
|");
System.out.println(" Hasil Pengolahan Nilai X Pre-increment");
System.out.println(" Nilai X Sebelum Operasi Increment Adalah      : " + x);
System.out.println(" Nilai X Setelah Operasi Increment Dengan ++X Adalah : " + ++x);
System.out.println(" Hasil Nilai X Terakhir adalah          : " + x);
System.out.println("=====
|");
System.out.println(" Menggunakan Operator Post-Increment Dengan Data Konstan |");
System.out.println("=====
|");
int y=9;
System.out.println(" Hasil Pengolahan Nilai Y Post-increment");
System.out.println(" Nilai Y Sebelum Operasi Post-Increment Adalah      : " + y);
System.out.println(" Nilai Y Sesudah Operasi y++ Adalah          : " + y++);
System.out.println(" Hasil Nilai Y Terakhir adalah          : " + y);
int z=8;
System.out.println("=====
|");
System.out.println(" Menggunakan Operator Pre-Decrement Dengan Data Konstan |");
System.out.println("=====
|");
System.out.println(" Hasil Pengolahan Nilai Z Pre-increment");
System.out.println(" Nilai Z Sebelum Operasi Increment Adalah      : " + z);
System.out.println(" Nilai Z Setelah Operasi Increment Dengan ++X Adalah : " + --z);
```

### Keterangan Kode Program

Dari Program 3.18 Menggunakan *operator increment* dan *decrement* dengan data konstan dan *variable* yang dideklarasikan adalah nilai  $x=9, y=9, z=8, w=9$ . Untuk operasi *increment* nilai yang telah dideklarasikan akan bertambah, misalnya terlihat pada statement:  $x=x+1$  dan  $y=y+1$ , sedangkan untuk operasi *Decrement* nilai *variable* akan berkurang satu



```
=====
Menggunakan Operator Pre-Increment Dengan Data Konstan
=====
Hasil Pengolahan Nilai X Pre-increment
Nilai X Sebelum Operasi Increment Adalah          : 9
Nilai X Setelah Operasi Increment Dengan ++X Adalah : 10
Hasil Nilai X Terakhir adalah                     : 10
=====
Menggunakan Operator Post-Increment Dengan Data Konstan
=====
Hasil Pengolahan Nilai Y Post-increment
Nilai Y Sebelum Operasi Post-Increment Adalah     : 9
Nilai Y Sesudah Operasi y++ Adalah                : 9
Hasil Nilai Y Terakhir adalah                     : 10
=====
Menggunakan Operator Pre-Decrement Dengan Data Konstan
=====
Hasil Pengolahan Nilai Z Pre-increment
Nilai Z Sebelum Operasi Increment Adalah          : 8
Nilai Z Setelah Operasi Increment Dengan ++X Adalah : 7
Hasil Nilai Z Terakhir adalah                     : 7
=====
Menggunakan Operator Post-Decrement Dengan Data Konstan
=====
Hasil Pengolahan Nilai W Post-Decrement
Nilai W Sebelum Operasi Post-Increment Adalah     : 9
Nilai W Sesudah Operasi y++ Adalah                : 9
Hasil Nilai W Terakhir adalah                     : 8
=====
Selamat Mencoba Menggunakan Opeartor Aritmatika
Dengan Data Inputan
Terima Kasih Banyak @ FauziahKasyfi2015
=====
Press any key to continue...
```

Gambar 3.18 Tampilan Output Operator Increment& Decrement Menggunakan Data Konstan

- Contoh Program Java Menggunakan *Operator Increment* dan *Decrement* Dengan Data *Input*

```
// Program 3.19
import java.io.*;
public class operator5 {
public static void main(String[] args)throws Exception {
BufferedReader input = new BufferedReader
(new InputStreamReader (System.in));
String nama,kelas,alamat,NA,NB,NC,ND;
double a,b,c,d,x,x1,y,y1,z,z1,v,v1,w,w1,u,u1,p,p1,q;
System.out.println("=====
==|");
System.out.println("| Menggunakan Operator Increment & Decrement Dengan Data
Input|");
System.out.println("=====
==|");
System.out.print ("      Inputkan Nama Anda      : ");
nama = input.readLine();
System.out.print ("      Inputkan Kelas Anda      : ");
kelas = input.readLine();
System.out.print ("      Inputkan Alamat Anda      : ");
alamat = input.readLine();
System.out.print ("      Inputkan Nilai A      : ");
NA = input.readLine();
System.out.print ("      Inputkan Nilai B      : ");
NB = input.readLine();
```

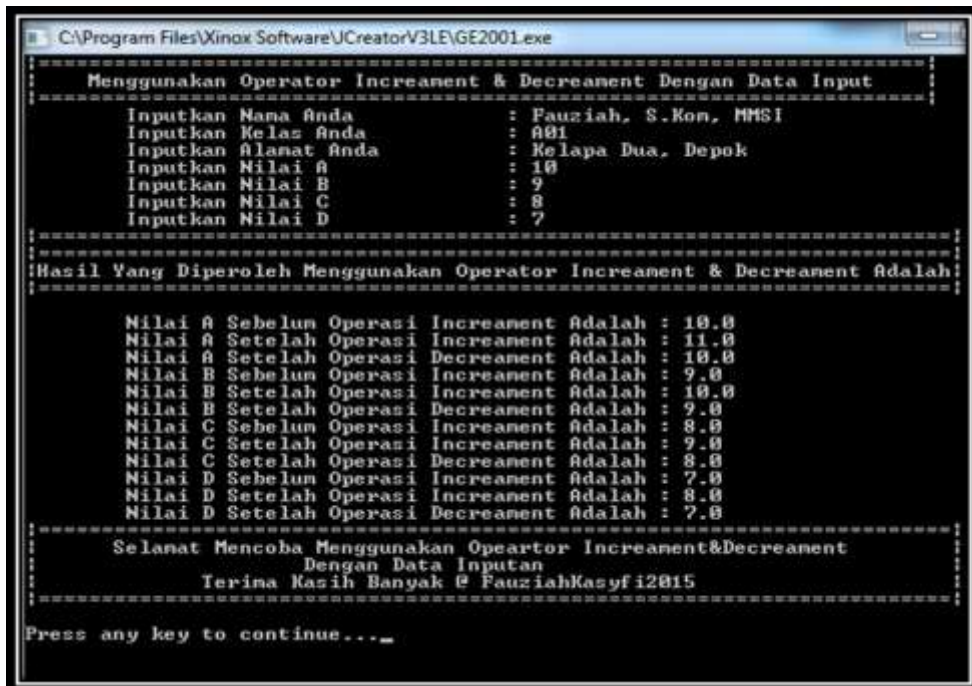
```
y = --a;
z = b;
z1= ++b;
w = --b;
v = c;
v1= ++c;
u = --c;
p = d;
p1=++d;
q=-d;

System.out.println("|=====
==|");
System.out.println("|Hasil Yang Diperoleh Menggunakan Operator Increment &
Decrement Adalah|");
System.out.println("|=====
=|");
System.out.println("");
System.out.println("    Nilai A Sebelum Operasi Increment Adalah : " + x);
System.out.println("    Nilai A Setelah Operasi Increment Adalah : " + x1);
System.out.println("    Nilai A Setelah Operasi Decrement Adalah : " + y);
System.out.println("    Nilai B Sebelum Operasi Increment Adalah : " + z);
System.out.println("    Nilai B Setelah Operasi Increment Adalah : " + z1);
System.out.println("    Nilai B Setelah Operasi Decrement Adalah : " + w);
System.out.println("    Nilai C Sebelum Operasi Increment Adalah : " + v);
System.out.println("    Nilai C Setelah Operasi Increment Adalah : " + v1);
System.out.println("    Nilai C Setelah Operasi Decrement Adalah : " + u);
System.out.println("    Nilai D Sebelum Operasi Increment Adalah : " + p);
```

### Keterangan Kode Program

Dari Program 3.19 Menggunakan *operator increment* dan *decrement* dengan data inputan dan *variable* yang dideklarasikan adalah beberapa nilai yaitu nilai input menggunakan tipe data *string* adalah :nama, kelas, npm dan alamat. Untuk data *input* menggunakan *variable* NA,NB,NC,ND dan untuk *variable* penyimpanan data menggunakan double a,b,c,d,x,x1,y,y1,z,z1,v,v1,w,w1,u,u1,p,p1,q.

Operasi yang digunakan adalah operasi *increment* dan



```
CAProgram Files\Xinox Software\UCreatorV3\LE\GE2001.exe
-----
Mengggunakan Operator Increment & Decreament Dengan Data Input
-----
Inputkan Nama Anda      : Fauziah, S.Kom, MMSI
Inputkan Kelas Anda     : 001
Inputkan Alamat Anda    : Kelapa Dua, Depok
Inputkan Nilai A        : 10
Inputkan Nilai B         : 9
Inputkan Nilai C         : 8
Inputkan Nilai D         : 7
-----
Hasil Yang Diperoleh Menggunakan Operator Increment & Decreament Adalah:
-----
Nilai A Sebelum Operasi Increment Adalah : 10.0
Nilai A Setelah Operasi Increment Adalah : 11.0
Nilai A Setelah Operasi Decreament Adalah : 10.0
Nilai B Sebelum Operasi Increment Adalah : 9.0
Nilai B Setelah Operasi Increment Adalah : 10.0
Nilai B Setelah Operasi Decreament Adalah : 9.0
Nilai C Sebelum Operasi Increment Adalah : 8.0
Nilai C Setelah Operasi Increment Adalah : 9.0
Nilai C Setelah Operasi Decreament Adalah : 8.0
Nilai D Sebelum Operasi Increment Adalah : 7.0
Nilai D Setelah Operasi Increment Adalah : 8.0
Nilai D Setelah Operasi Decreament Adalah : 7.0
-----
Selamat Mencoba Menggunakan Opeartor Increment&Decreament
Dengan Data Inputan
Terima Kasih Banyak @ FauziahKasyf i2015
-----
Press any key to continue..._
```

Gambar 3.19 Tampilan output Menggunakan Operator Increment & Decrement Dengan Data

Input



**Contoh Program Java Menggunakan Logika And, Or dan Xor Data Konstan**

## Program 3.20

```

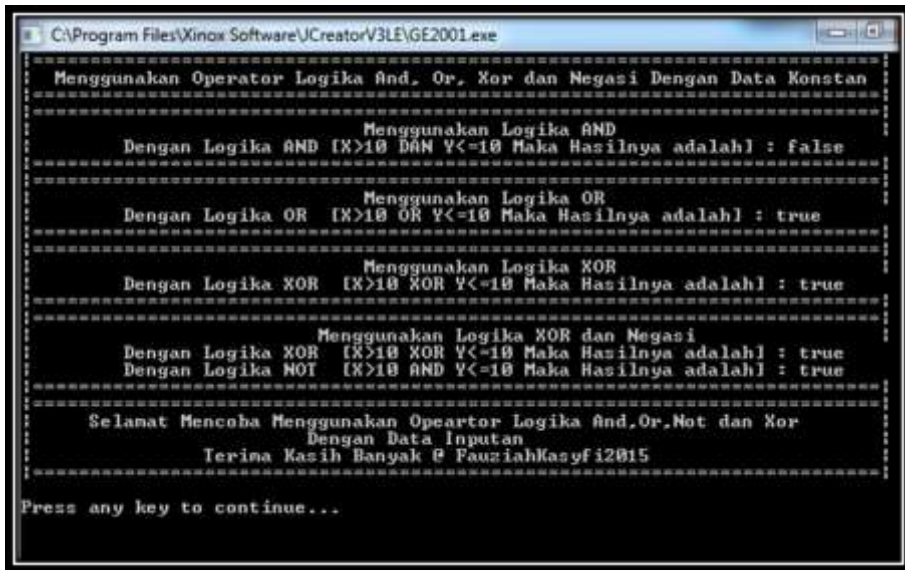
class operator6 {
    public static void main(String[] args) {
        int x=9;// deklarasasi nilai x
        int y=10;// deklarasi nilai y
        boolean z,v,w,u,q;// menyimpan nilai boolean
        // deklarasi nilai dengan logika and dan disimpan pada variabel z
        z= x>10 && y<=10;
        System.out.println("=====");
        System.out.println("| Menggunakan Operator Logika And, Or, Xor dan Negasi Dengan
        Data Konstan |");
        System.out.println("=====");
        System.out.println("=====");
        System.out.println("|      Menggunakan Logika AND      |");
        System.out.println("|Dengan Logika AND [X>10 DAN Y<=10 Maka Hasilnya adalah: "+ z);
        System.out.println("=====");
        // deklarasi nilai dengan logika or dan disimpan pada variabel v
        v= x>10 || y<=10;
        System.out.println("=====");
        System.out.println("|      Menggunakan Logika OR      |");
        System.out.println("|Dengan Logika OR [X>10 OR Y<=10 Maka Hasilnya adalah: " + v);
        System.out.println("=====");
        // deklarasi nilai dengan logika xor dan disimpan pada variabel w
        w= x>10 ^ y<=10;
        System.out.println("=====");
        System.out.println("|      Menggunakan Logika XOR      |");
        System.out.println("|Dengan Logika XOR [X>10 XOR Y<=10 Maka Hasilnya adalah: "+ w);
        System.out.println("=====");
        // deklarasi nilai dengan logika xor dan negasi disimpan pada variabel u
        q= x>10 ^ y<=10;
        u=!z;
        System.out.println("=====");
        System.out.println("|      Menggunakan Logika XOR dan Negasi      |");
    }
}

```

### Keterangan Kode Program

Dari Program 3.20 Menggunakan *operator Logika And* [&&], *Or* [||], *Xor* [^] dan *Negasi* [!] data konstan *variable* yang dideklarasikan adalah beberapa nilai yaitu nilai  $x=9$  dan nilai  $y=10$ . Kemudian untuk menyimpan nilainya kita menggunakan tipe data Boolean [nilai true atau false]. Proses yang dilakukan adalah :

1. Untuk logika And proses yang dilakukan adalah :  
 $z=x>10 \ \&\& \ y\leq 10$ ; Mengecek nilai apakah  $x>10$  [False] && mengecek nilai apakah  $y\leq 10$  [true], Pada Logika And, Jika nilai [false] && [true] maka hasilnya adalah [false]
2. Untuk logika Or proses yang dilakukan adalah :  
 $v=x>10 \ || \ y\leq 10$ ; Mengecek nilai apakah  $x>10$  [False] || mengecek nilai apakah  $y\leq 10$  [true], Pada Logika Or, Jika nilai [false] && [true] maka hasilnya adalah [true]
3. Untuk logika XOr proses yang dilakukan adalah :  
 $w= x>10 \ \wedge \ y\leq 10$ ; Mengecek nilai apakah  $x>10$  [False] || mengecek nilai apakah  $y\leq 10$  [true], Pada Logika XOr, Jika nilai [false] && [true] maka hasilnya adalah [true]
4. Untuk logika Negasi proses yang dilakukan adalah :  
 $z=x>10 \ \&\& \ y\leq 10$ ; Mengecek nilai apakah  $x>10$  [False] && mengecek nilai apakah  $y\leq 10$  [true], Pada Logika And, Jika nilai [false] && [true] maka hasilnya adalah [false], kemudian dilakukan proses negasi dari hasil logika And yaitu [!z], maka hasilnya adalah [true]



```
C:\Program Files\Xinox Software\JCreatorV3LE\GE2001.exe
Mengggunakan Operator Logika And, Or, Xor dan Negasi Dengan Data Konstan
-----
Mengggunakan Logika AND
Dengan Logika AND [X>10 DAN Y<=10] Maka Hasilnya adalah : false
-----
Mengggunakan Logika OR
Dengan Logika OR [X>10 OR Y<=10] Maka Hasilnya adalah : true
-----
Mengggunakan Logika XOR
Dengan Logika XOR [X>10 XOR Y<=10] Maka Hasilnya adalah : true
-----
Mengggunakan Logika XOR dan Negasi
Dengan Logika XOR [X>10 XOR Y<=10] Maka Hasilnya adalah : true
Dengan Logika NOT [X>10 AND Y<=10] Maka Hasilnya adalah : true
-----
Selamat Mencoba Menggunakan Opeartor Logika And,Or,Not dan Xor
Dengan Data Inputan
Terima Kasih Banyak @ Fauziahkasyfi2015
-----
Press any key to continue...
```

Gambar 3.20 Tampilan output Menggunakan Operator Logika And, Or, Xor & Negasi Dengan Data Konstan

- **Contoh Program Dengan Logika And, Or, Xor dan Negasi Menggunakan Data Input**

```
//Program 3.21
import java.io.*;
public class operator7 {
public static void main(String[] args)throws Exception {
BufferedReader input = new BufferedReader
(new InputStreamReader (System.in));
// deklarasi input
String nama,kelas,alamat,NA,NB,NC,ND;
// deklarasi variable input nilai
double a,b,c,d;
// deklarasi penyimpanan logika boolean
boolean v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16,v17,v18,v19,v20,v21,v22,v23,v24,v25,v26,v27,v28,v29,v30,v31,v32,v33,v34,v35,v36,v37,v38,v39,v40,v41,v42,v43,v44,v45,v46,v47,v48,v49,v50,v51,v52,v53,v54,v55,v56,v57,v58,v59,v60,v61,v62,v63,v64,v65,v66,v67,v68,v69,v70,v71,v72,v73,v74,v75,v76,v77,v78,v79,v80,v81,v82,v83,v84,v85,v86,v87,v88,v89,v90,v91,v92,v93,v94,v95,v96,v97,v98,v99,v100,v101,v102,v103,v104,v105,v106,v107,v108,v109,v110,v111,v112,v113,v114,v115,v116,v117,v118,v119,v120,v121,v122,v123,v124,v125,v126,v127,v128,v129,v130,v131,v132,v133,v134,v135,v136,v137,v138,v139,v140,v141,v142,v143,v144,v145,v146,v147,v148,v149,v150,v151,v152,v153,v154,v155,v156,v157,v158,v159,v160,v161,v162,v163,v164,v165,v166,v167,v168,v169,v170,v171,v172,v173,v174,v175,v176,v177,v178,v179,v180,v181,v182,v183,v184,v185,v186,v187,v188,v189,v190,v191,v192,v193,v194,v195,v196,v197,v198,v199,v200,v201,v202,v203,v204,v205,v206,v207,v208,v209,v210,v211,v212,v213,v214,v215,v216,v217,v218,v219,v220,v221,v222,v223,v224,v225,v226,v227,v228,v229,v230,v231,v232,v233,v234,v235,v236,v237,v238,v239,v240,v241,v242,v243,v244,v245,v246,v247,v248,v249,v250,v251,v252,v253,v254,v255,v256,v257,v258,v259,v260,v261,v262,v263,v264,v265,v266,v267,v268,v269,v270,v271,v272,v273,v274,v275,v276,v277,v278,v279,v280,v281,v282,v283,v284,v285,v286,v287,v288,v289,v290,v291,v292,v293,v294,v295,v296,v297,v298,v299,v300,v301,v302,v303,v304,v305,v306,v307,v308,v309,v310,v311,v312,v313,v314,v315,v316,v317,v318,v319,v320,v321,v322,v323,v324,v325,v326,v327,v328,v329,v330,v331,v332,v333,v334,v335,v336,v337,v338,v339,v340,v341,v342,v343,v344,v345,v346,v347,v348,v349,v350,v351,v352,v353,v354,v355,v356,v357,v358,v359,v360,v361,v362,v363,v364,v365,v366,v367,v368,v369,v370,v371,v372,v373,v374,v375,v376,v377,v378,v379,v380,v381,v382,v383,v384,v385,v386,v387,v388,v389,v390,v391,v392,v393,v394,v395,v396,v397,v398,v399,v400,v401,v402,v403,v404,v405,v406,v407,v408,v409,v410,v411,v412,v413,v414,v415,v416,v417,v418,v419,v420,v421,v422,v423,v424,v425,v426,v427,v428,v429,v430,v431,v432,v433,v434,v435,v436,v437,v438,v439,v440,v441,v442,v443,v444,v445,v446,v447,v448,v449,v450,v451,v452,v453,v454,v455,v456,v457,v458,v459,v460,v461,v462,v463,v464,v465,v466,v467,v468,v469,v470,v471,v472,v473,v474,v475,v476,v477,v478,v479,v480,v481,v482,v483,v484,v485,v486,v487,v488,v489,v490,v491,v492,v493,v494,v495,v496,v497,v498,v499,v500,v501,v502,v503,v504,v505,v506,v507,v508,v509,v510,v511,v512,v513,v514,v515,v516,v517,v518,v519,v520,v521,v522,v523,v524,v525,v526,v527,v528,v529,v530,v531,v532,v533,v534,v535,v536,v537,v538,v539,v540,v541,v542,v543,v544,v545,v546,v547,v548,v549,v550,v551,v552,v553,v554,v555,v556,v557,v558,v559,v560,v561,v562,v563,v564,v565,v566,v567,v568,v569,v570,v571,v572,v573,v574,v575,v576,v577,v578,v579,v580,v581,v582,v583,v584,v585,v586,v587,v588,v589,v590,v591,v592,v593,v594,v595,v596,v597,v598,v599,v600,v601,v602,v603,v604,v605,v606,v607,v608,v609,v610,v611,v612,v613,v614,v615,v616,v617,v618,v619,v620,v621,v622,v623,v624,v625,v626,v627,v628,v629,v630,v631,v632,v633,v634,v635,v636,v637,v638,v639,v640,v641,v642,v643,v644,v645,v646,v647,v648,v649,v650,v651,v652,v653,v654,v655,v656,v657,v658,v659,v660,v661,v662,v663,v664,v665,v666,v667,v668,v669,v670,v671,v672,v673,v674,v675,v676,v677,v678,v679,v680,v681,v682,v683,v684,v685,v686,v687,v688,v689,v690,v691,v692,v693,v694,v695,v696,v697,v698,v699,v700,v701,v702,v703,v704,v705,v706,v707,v708,v709,v710,v711,v712,v713,v714,v715,v716,v717,v718,v719,v720,v721,v722,v723,v724,v725,v726,v727,v728,v729,v730,v731,v732,v733,v734,v735,v736,v737,v738,v739,v740,v741,v742,v743,v744,v745,v746,v747,v748,v749,v750,v751,v752,v753,v754,v755,v756,v757,v758,v759,v760,v761,v762,v763,v764,v765,v766,v767,v768,v769,v770,v771,v772,v773,v774,v775,v776,v777,v778,v779,v780,v781,v782,v783,v784,v785,v786,v787,v788,v789,v790,v791,v792,v793,v794,v795,v796,v797,v798,v799,v800,v801,v802,v803,v804,v805,v806,v807,v808,v809,v810,v811,v812,v813,v814,v815,v816,v817,v818,v819,v820,v821,v822,v823,v824,v825,v826,v827,v828,v829,v830,v831,v832,v833,v834,v835,v836,v837,v838,v839,v840,v841,v842,v843,v844,v845,v846,v847,v848,v849,v850,v851,v852,v853,v854,v855,v856,v857,v858,v859,v860,v861,v862,v863,v864,v865,v866,v867,v868,v869,v870,v871,v872,v873,v874,v875,v876,v877,v878,v879,v880,v881,v882,v883,v884,v885,v886,v887,v888,v889,v890,v891,v892,v893,v894,v895,v896,v897,v898,v899,v900,v901,v902,v903,v904,v905,v906,v907,v908,v909,v910,v911,v912,v913,v914,v915,v916,v917,v918,v919,v920,v921,v922,v923,v924,v925,v926,v927,v928,v929,v930,v931,v932,v933,v934,v935,v936,v937,v938,v939,v940,v941,v942,v943,v944,v945,v946,v947,v948,v949,v950,v951,v952,v953,v954,v955,v956,v957,v958,v959,v960,v961,v962,v963,v964,v965,v966,v967,v968,v969,v970,v971,v972,v973,v974,v975,v976,v977,v978,v979,v980,v981,v982,v983,v984,v985,v986,v987,v988,v989,v990,v991,v992,v993,v994,v995,v996,v997,v998,v999,1000
}
```

- **Contoh Program Dengan Logika And, Or, Xor dan Negasi Menggunakan Data Input**

```
System.out.print ("      Inputkan Nama Anda      : ");
nama = input.readLine();
System.out.print ("      Inputkan Kelas Anda      : ");
kelas = input.readLine();
System.out.print ("      Inputkan Alamat Anda      : ");
alamat = input.readLine();
System.out.print ("      Inputkan Nilai A      : ");
NA = input.readLine();
System.out.print ("      Inputkan Nilai B      : ");
NB = input.readLine();
System.out.print ("      Inputkan Nilai C      : ");
NC = input.readLine();
System.out.print ("      Inputkan Nilai D      : ");
ND = input.readLine();
System.out.println("=====
===");
a = Double.parseDouble(NA);
b = Double.parseDouble(NB);
c = Double.parseDouble(NC);
d = Double.parseDouble(ND);
//proses logika
x = a>=10 ;
x1= b<=10;
y = x&& x1;
z = x||x1;
z1= x^x1;
w = !y;
v = !z;
v1= !z1;
u = !x;
System.out.println("-----
```

```
System.out.println("    Hasil Nilai v1= !z1 [negasi Xor] Adalah : " + v1);
System.out.println("    Hasil Nilai u = !x [negasi x ] Adalah : " + u);
System.out.println("=====
|");
System.out.println("|Selamat Mencoba Menggunakan Opeartor Logika And, Xor, Or,& Negasi|");
System.out.println("|    Dengan Data Inputan                |");
System.out.println("|    Terima Kasih Banyak @ FauziahKasyfi2015                |");
System.out.println("=====
==|");
System.out.println("");
}
}
```

### Keterangan Kode Program

Dari Program 3.21 Menggunakan logika And, Or, Xor, dan Negasi, variable yang digunakan adalah variable input dengan tipe data berbeda yaitu :

1. Tipe data String yaitu :nama,kelas,alamat,NA,NB,NC,ND
2. Tipe Data Input Nilai dengan Double yaitu nilai : double a,b,c,d
3. Tipe Data Boolean yaitu tipe data yang digunakan untuk menyimpan nilai true dan false untuk nilai : boolean x,x1,y,z,z1,v,w,v1,u
4. Pengecekan Logika menggunakan Proses :

x = a>=10 ; melakukan pengecekan nilai apakah nilai x yaitu apakah nilai a yang diinputkan >= 10, jika ya bernilai [true] dan jika tidak bernilai [false]

x1= b<=10; melakukan pengecekan nilai apakah nilai x1 yaitu apakah nilai b yang diinputkan <=10, jika ya bernilai [true] dan jika tidak bernilai [false]

y = x&& x1;melakukan pengecekan nilai x dan x1 menggunakan logika

```
C:\Program Files\Xinox Software\JCreatorV3LE\GE2001.exe
-----
Menggunakan Operator Logika And, Or, XOR dan Negasi Dengan Data Input
-----
Inputkan Nama Anda           : Fauziah, S.Kom, MMSI
Inputkan Kelas Anda          : A01
Inputkan Alamat Anda         : Kelapa Dua, Depok
Inputkan Nilai A              : 9
Inputkan Nilai B              : 10
Inputkan Nilai C              : 20
Inputkan Nilai D              : 30
-----
: Hasil Yang Diperoleh Menggunakan Operator Logika And, Or, Xor & Negasi Adalah :
-----
Hasil Nilai x = a>=10 Adalah           : false
Hasil Nilai x1 = b<=10 Adalah           : true
Hasil Nilai y = x&&x1 [logika And] Adalah : false
Hasil Nilai z = x||x1 [logika or ] Adalah : true
Hasil Nilai z1 = x^x1 [Logika xor] Adalah : true
Hasil Nilai w = !y [negasi And] Adalah : true
Hasil Nilai v = !z [negasi or ] Adalah : false
Hasil Nilai v1 = !z1 [negasi Xor] Adalah : false
Hasil Nilai u = !x [negasi x ] Adalah : true
-----
Selamat Mencoba Menggunakan Opeartor Logika And, Xor, Or, & Negasi
Dengan Data Inputan
Terima Kasih Banyak @ FauziahKasyfi2015
-----
Press any key to continue..._
```

Gambar 3.21 Tampilan Output Menggunakan Operator And, Or, XOR dan Negasi Dengan Variabel Input

# PENGAYAAN

Jawablah pertanyaan di bawah ini dengan jelas dan benar

1. Apa itu *Operator*
2. Jelaskan Perbedaan *operator* dan *operand*
3. Berikan Contoh Macam – macam *operator*
4. Buatlah Program dengan menggunakan Macam – macam *operator* dengan data konstan dan data *inputan* dan *screen shoot* tampilan yang dihasilkan dari masing-masing kode program yang dibuat.

# GLOSARIUM

## GLOSARIUM BAB 9

1. Operator merupakan *symbol* atau berbagai macam lambang yang sering kita gunakan dalam bahasa pemrograman dan memiliki fungsi masing – masing.
2. *operand*, yaitu pengenal atau *variable* yang digunakan untuk mengoperasikan suatu nilai  
Contoh Operator adalah : *Operator* matematika, *symbol* yang digunakan adalah [+,-,\*,/,%]
3. Contoh *operand* adalah : variable a,b,c dll
4. Contoh Operator adalah : *Operator* matematika, *symbol* yang digunakan adalah [+,-,\*,/,%]
5. *Operator Assignment* : Merupakan jenis *operator* yang berfungsi untuk memberikan suatu nilai(*value*) kepada *variable* yang ditunjuk, biasanya *operator assignment* digunakan untuk jenis data *variable* konstan *Operator*



6. Contoh *operator assignment* adalah  $Int\ x=5$
7. Operator Logika Merupakan jenis operator yang memberikan nilai true(benar) dan false(salah). Simbol yang digunakan pada operator logika adalah: Logika And (Logika dan)[symbol yang digunakan &&] Logika Or (Logika atau)[symbol yang digunakan ||] Logika Negasi(Logika ingkaran) [symbol yang digunakan adalah [ ! ]
8. *Operator Relasional*, Merupakan jenis *operator* yang digunakan untuk menghubungkan 2 buah *variable* atau *operand* yang menghasilkan suatu keputusan
9. *Operator Increment dan Decrement*, Merupakan jenis *operator* yang digunakan untuk melakukan proses penambahan atau pengurangan nilai [ yaitu nilai peubah naik atau nilai turun 1

# INFO FLOWCHART

Flowchart digunakan dalam bahasa pemrograman untuk menjelaskan logika program yang dibuat oleh programmer dengan tujuan memudahkan pembacaan logika.

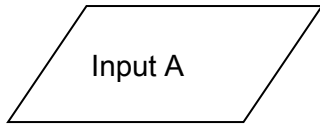
Flowchart merupakan diagram alir menjelaskan logika program yang dibuat berupa symbol – symbol atau bagan yang memiliki fungsi/arti masing- masing.

Simbol flowchart yang sering digunakan adalah :

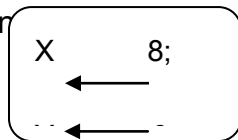
1. Simbol terminator :” merupakan symbol awal dan akhir, berisi st **START/END** selesai. Simbol terminator start dan end adalah

2. Simbol deklarasi data / variable yang berisi nilai

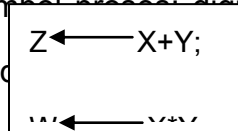
- a. Untuk *Variabel* dengan nilai *inputan* maka *symbol flowchart* yang digunakan adalah :



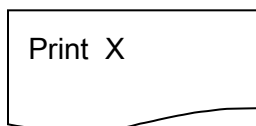
- b. Untuk *variable* dengan nilai konstan, maka *symbol flowchart* yang digunakan adalah :



3. Simbol *process* digunakan untuk melakukan proses komputasi atau deklarasi atau *inputkan*.

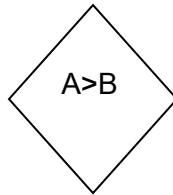


4. Simbol *output* digunakan untuk menghasilkan keluaran atau untuk cetakan.



Artinya mencetak tulisan atau output "Nilai X"

5. Simbol keputusan [*Decision Making*] : merupakan *symbol* yang digunakan untuk melakukan pengecekan kondisi apakah bernilai benar atau salah. *Symbol* kotak keputusan biasanya kita gunakan untuk logika percabangan.



Melakukan pengecekan nilai apakah nilai  $A > B$  jika ya maka lakukan *statement* berikutnya jika tidak biasanya kita

Untuk lebih jelasnya dapat kita lihat contoh kasus yang Diambil dari contoh program 3.1

Menggunakan *operator aritmatika* untuk menghitung sisa pendapatan dengan komponen pengeluaran adalah sebagai berikut :

- a. Pendapatan [*diinput*] *variable* [P]
- b. Pembayaran uang kos [*diinput*] *variable* [I]
- c. Pembayaran transport [*diinput*] *variable* [a]
- d. Pembayaran uang makan [*diinput*] *variable* [t]

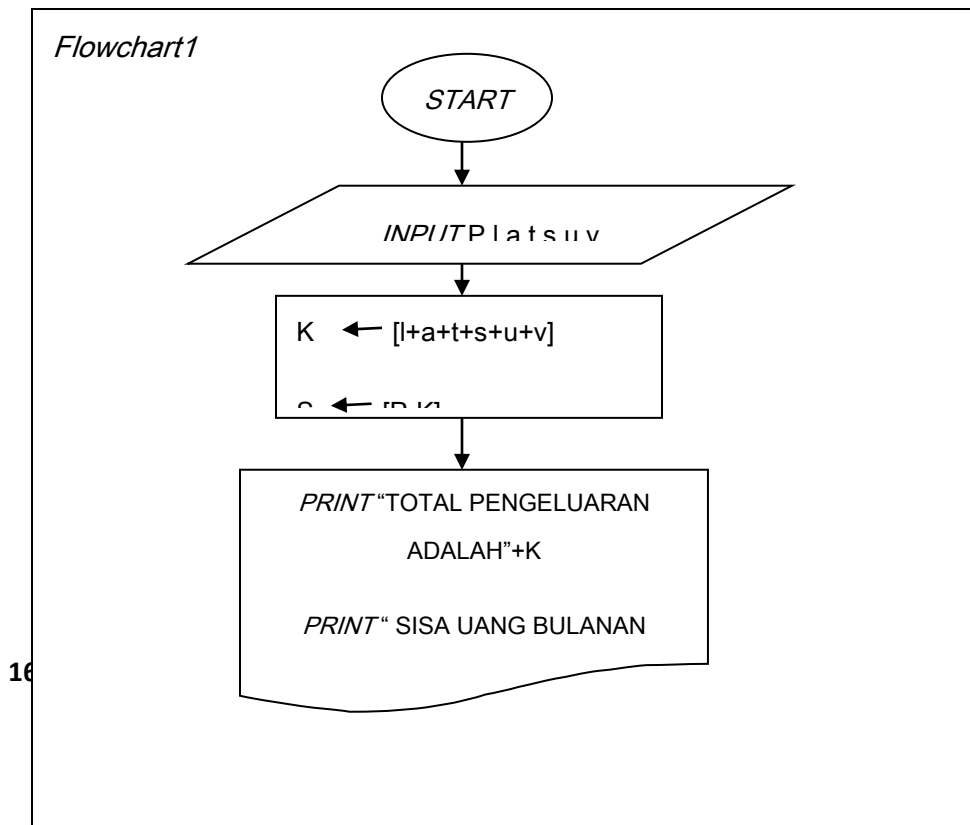
- e. Pembayaran uang kuliah [*diinput*] variable [s]
- f. Pembayaran uang buku [*diinput*] variable [u]
- g. Pembayaran lain-lain [*diinput*] variable [v]

Untuk total pengeluaran [K] menggunakan rumus :

$$[K=l+a+t+s+u+v]$$

Untuk sisa gunakan rumus : pendapatan – semua komponen pengeluaran [S= P-K]

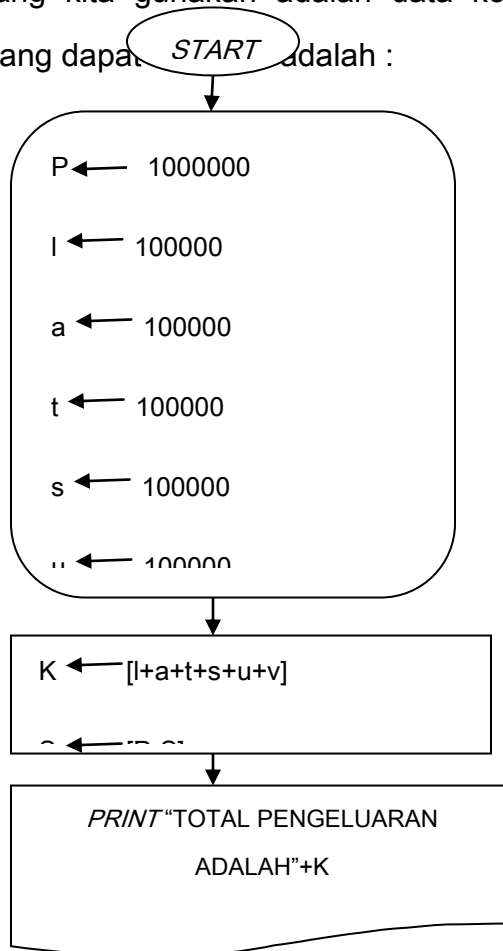
Maka *Flowchartnya* adalah sebagai berikut:



Dari *flowchart* 1 terlihat data yang *diinput* adalah pendapatan, beberapa komponen pengeluaran, untuk total pengeluaran menggunakan rumus K dan untuk sis menggunakan rumus S.

Jika data yang kita gunakan adalah data konstan, maka *flowchart* 2 yang dapat *START* adalah :

*Flowchart2*



Dari contoh *flowchart2* terlihat menggunakan data konstan dengan contoh data Pendapatan 1000000, komponen pengeluaran yang terdiri dari :

- a. Pembayaran uang kos [l=100000]
- b. Pembayaran transport [a=100000]
- c. Pembayaran uang makan [t=100000]
- d. Pembayaran uang kuliah [s=100000]
- e. Pembayaran uang buku [u=100000]
- f. Pembayaran lain-lain [v=100000]

Untuk komponen total pengeluaran menggunakan rumus yang sama yaitu  $[K=l+a+t+s+u+v]$ , dan untuk Sisa menggunakan rumus  $[S=P-K]$ .

**Ingat ! Jika kita menggunakan data konstan maka hasil**



Sumber: [defenseindustrydaily.com](http://defenseindustrydaily.com)

# BAB IV

## STRUCTURE CONTROL PADA BAHASA PEMROGRAMAN

### Tujuan Pembelajaran

Setelah mempelajari bab ini Mahasiswa diharapkan mampu:

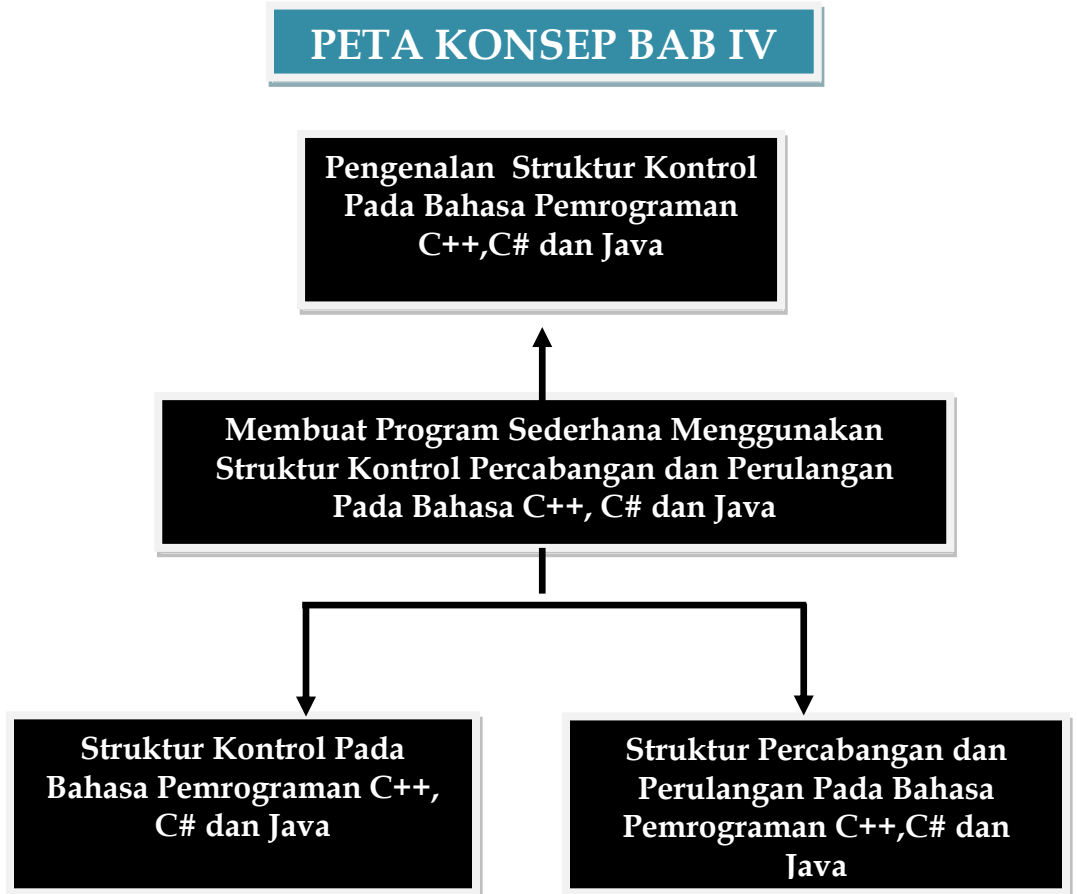
Menjelaskan *structure control* yang digunakan pada bahasa pemrograman C++, C# dan Java, dan dapat memberikan beberapa contoh program menggunakan struktur kontrol percabangan dan perulangan dengan jelas dan benar

Pada Kode program yang dibuat oleh pengguna atau *programmer* tidak terlepas dari penggunaan struktur control



yang digunakan untuk mengendalikan proses dalam pembuatan program. Pada struktur control digunakan logika percabangan dan perulangan.

Kata Kunci: *program, programmer, struktur control, percabangan, perulangan.*



# 4.1 Pengertian Struktur Kontrol Pada Bahasa Pemrograman

- **Pengertian Struktur Kontrol**

Struktur *control* pada bahasa pemrograman banyak digunakan yaitu untuk mengendalikan proses pada program. Struktur *control* yang umum digunakan adalah :

- a. Struktur *control* percabangan [*branching*]
- b. Struktur *control* perulangan [*looping*]

### **Struktur Control Percabangan**

Merupakan statement yang digunakan untuk melakukan pengecekan satu atau lebih kondisi tertentu dengan berbagai macam pilihan. Struktur *control* percabangan yang sering digunakan adalah struktur logika *If* dan struktur logika *switch*.

- **Macam Logika Statement If**

- ✚ If tanpa else

- ✚ If dengan else

Bentuk umum Logika If tanpa else [ hanya 1 buah kondisi]

```
If (Condition) Statement/Pernyataan;
```

Bentuk u

```
If (Condition1) Statement1/Pernyataan1;
```

disi]

```
Else
```

Ber

```
If (Condition1) Statement1/Pernyataan1;
```

```
Else
```

```
If (Condition2) Statement2/Pernyataan2;
```

```
Else
```

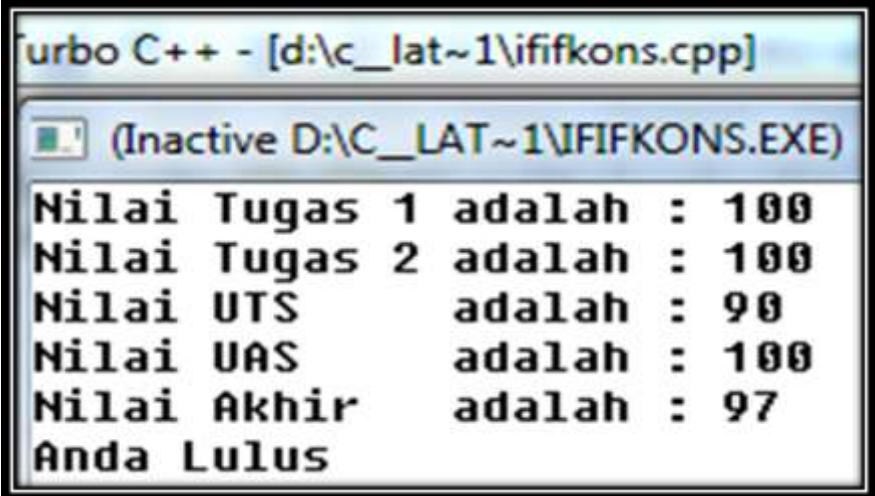
Untuk lebih jelasnya, kita dapat melihat contoh program berikut :

## CONTOH PROGRAM DENGAN C++ MENGGUNAKAN LOGIKA IF

### Contoh Program C++ Menggunakan Logika IF dengan Data Konstan

```
// Program 4.1
#include <iostream.h>
int main ()
{
// inialisasi variabel :
int a, b, c, d; // a: tugas1, b:tugas2, c: UTS dan d: UAS
int X; // Nilai Akhir
// Proses Data Konstan Dengan nilai tetap
a = 100;
b = 100;
c = 90;
d= 100 ;
// Proses perhitungan nilai akhir dengan mencari rata-rata
X = (a+b+c+d)/4;
// Mencetak Hasil Tampilan :
cout<<"Nilai Tugas 1 adalah : "<<a<<"\n";
cout<<"Nilai Tugas 2 adalah : "<<b<<"\n";
cout<<"Nilai UTS   adalah : "<<c<<"\n";
cout<<"Nilai UAS   adalah : "<<d<<"\n";
cout<<"Nilai Akhir  adalah : "<<X<<"\n";
// Menggunakan logika if untuk pengecekan 2 kondisi Jika Nilai
// Akhir >60, Maka Mahasiswa Lulus dan Jika <60 Maka Mahasiswa
```

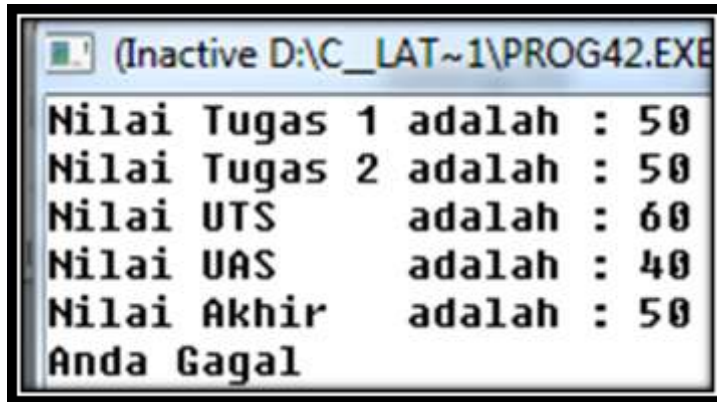
Dari program 4.1 terlihat data yang digunakan adalah data konstan yaitu untuk nilai tugas1[a], tugas2[b], UTS[c] dan UAS[d] untuk proses nilai akhir menggunakan *variable* X yaitu  $[X=(a+b+c+d)/4]$ . Pada *Logika If* menggunakan kondisi sebagai berikut; Jika nilai Akhir $[X>60]$ , maka mahasiswa lulus dan Jika nilai Akhir  $[X<60]$ , maka mahasiswa gagal. Terlihat kondisi yang digunakan adalah 2 buah kondisi, maka logika *If* yang kita deklarasikan pada program adalah 1 saja sesuai dengan



```
turbo C++ - [d:\c_lat~1\iffkons.cpp]
(Inactive D:\C_LAT~1\FIFKONS.EXE)
Nilai Tugas 1 adalah : 100
Nilai Tugas 2 adalah : 100
Nilai UTS          adalah : 90
Nilai UAS          adalah : 100
Nilai Akhir       adalah : 97
Anda Lulus
```

Gambar 4.1 Tampilan *Output* Menggunakan Logika *If* dengan Data Konstan

```
// Program 4.2
#include <iostream.h>
int main ()
{
// inialisasi variabel :
int a, b, c, d; // a: tugas1, b:tugas2, c: UTS dan d: UAS
int X; // Nilai Akhir
// Proses Data Konstan Dengan nilai tetap
a = 50;
b = 50;
c = 60;
d = 40;
// Proses perhitungan nilai akhir dengan mencari rata-rata
X = (a+b+c+d)/4;
// Mencetak Hasil Tampilan :
cout<<"Nilai Tugas 1 adalah : "<<a<<"\n";
cout<<"Nilai Tugas 2 adalah : "<<b<<"\n";
cout<<"Nilai UTS   adalah : "<<c<<"\n";
cout<<"Nilai UAS   adalah : "<<d<<"\n";
cout<<"Nilai Akhir  adalah : "<<X<<"\n";
// Menggunakan logika if untuk pengecekan 2 kondisi Jika Nilai
// Akhir >60, Maka Mahasiswa Lulus dan Jika <60 Maka Mahasiswa
// gagal, Terlihat ada 2 Kondisi maka If yang digunakan 1 saja.
if ( X>60)
cout << "Anda Lulus" << "\n";
else
cout << "Anda Gagal" << "\n";
// menghentikan program :
return 0;
}
```



Gambar 4.2 Tampilan *Output* Menggunakan Logika *If* dengan Data Konstan

### Contoh Program C++ Menggunakan Logika IF dengan Data Inputan

```
// Program 4.3
#include <iostream.h>
int main ()
{
// inisialisasi variabel :
int a, b, c, d; // a: tugas1, b:tugas2, c: UTS dan d: UAS
int X; // Nilai Akhir
cout <<
"|=====|" <<
endl;
cout << "|Menggunakan Logika Struktur Kontrol Percabangan Dengan IF|" << endl;
cout <<
"|=====|" <<
endl;
cout << "          Inputkan Nilai Tugas 1 : ";
```

```
// Proses perhitungan nilai akhir dengan mencari rata-rata
X = (a+b+c+d)/4;
// Mencetak Hasil Tampilan :
cout <<
"|=====|" <<
endl;
cout << "|    Mencetak Komponen Nilai Yang Sudah Diinputkan    |" << endl;
cout <<
"|=====|" <<
endl;
cout<<"    Nilai Tugas 1 adalah : "<<a<<"\n";
cout<<"    Nilai Tugas 2 adalah : "<<b<<"\n";
cout<<"    Nilai UTS   adalah : "<<c<<"\n";
cout<<"    Nilai UAS   adalah : "<<d<<"\n";
cout<<"    Nilai Akhir  adalah : "<<X<<"\n";
// Menggunakan logika if untuk pengecekan 2 kondisi Jika Nilai
// Akhir >60, Maka Mahasiswa Lulus dan Jika <60 Maka Mahasiswa
// gagal, Terlihat ada 2 Kondisi maka If yang digunakan 1 saja.
if ( X>60)
cout << "          Selamat Anda Lulus" << "\n";
else
cout << "          Maaf Anda Gagal" << "\n";
cout <<
"|=====|" <<
endl;
cout << "| Selamat Menggunakan Logika IF Terima Kasih FauziahKasyfi |" << endl;
cout <<
"|=====|" <<
endl;
// menghentikan program :
return 0;
```





```
(Inactive D:\C_LAT~1\PROG43.EXE)
=====
Menggunakan Logika Struktur Kontrol Percabangan Dengan IF
=====
          Inputkan Nilai Tugas 1 : 90
          Inputkan Nilai Tugas 2 : 90
          Inputkan Nilai UTS      : 90
          Inputkan Nilai UAS      : 90
=====
          Mencetak Komponen Nilai Yang Sudah Diinputkan
=====
          Nilai Tugas 1 adalah    : 90
          Nilai Tugas 2 adalah    : 90
          Nilai UTS      adalah   : 90
          Nilai UAS      adalah   : 90
          Nilai Akhir   adalah   : 90
          Selamat Anda Lulus
=====
          Selamat Menggunakan Logika IF Terima Kasih FauziahKasyfi
=====
```

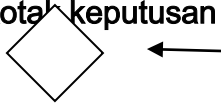
Gambar 4.3 Tampilan Output Dengan Logika If Menggunakan Variabel Input/Data Input

```
(Inactive D:\C_LAT~1\PROG43.EXE)
=====
Menggunakan Logika Struktur Kontrol Percabangan Dengan IF
=====
          Inputkan Nilai Tugas 1 : 60
          Inputkan Nilai Tugas 2 : 60
          Inputkan Nilai UTS      : 50
          Inputkan Nilai UAS      : 50
=====
          Mencetak Komponen Nilai Yang Sudah Diinputkan
=====
          Nilai Tugas 1 adalah    : 60
          Nilai Tugas 2 adalah    : 60
          Nilai UTS      adalah   : 50
          Nilai UAS      adalah   : 50
          Nilai Akhir   adalah   : 55
          Maaf Anda Gagal
=====
          Selamat Menggunakan Logika IF Terima Kasih FauziahKasyfi
=====
```

Gambar 4.4 Tampilan Output Dengan Logika If Menggunakan Variabel Input/Data Input

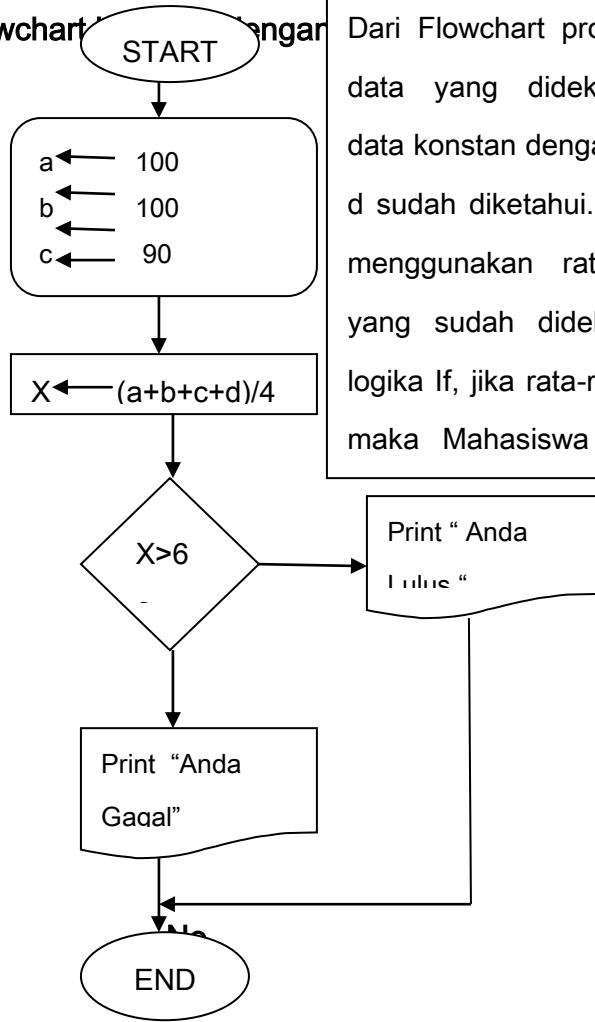
### Flowchart Struktur Kontrol Logika If

Catatan : Untuk logika if symbol flowchart yang kita gunakan adalah symbol kotak keputusan



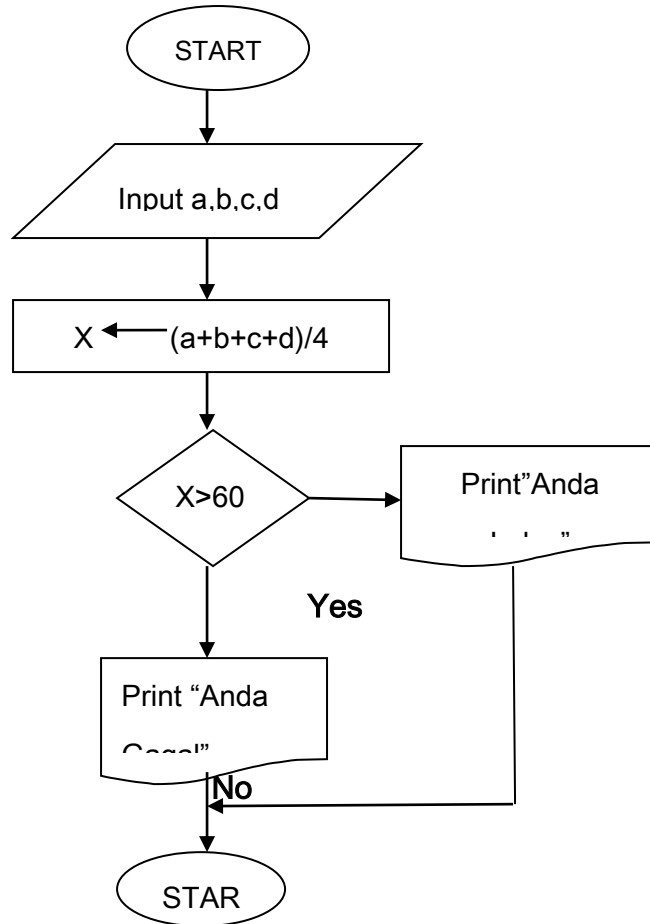
Symbol kotak keputusan

a. Flowchart dengan



Dari Flowchart program 4.1 terlihat data yang dideklarasikan adalah data konstan dengan nilai a, b, c dan d sudah diketahui. Untuk Nilai Akhir menggunakan rata-rata dari nilai yang sudah dideklarasikan. Untuk logika If, jika rata-rata nilai akhir > 60, maka Mahasiswa lulus, jika tidak

b. Flowchart Logika If dengan data inputan [program 4.3]

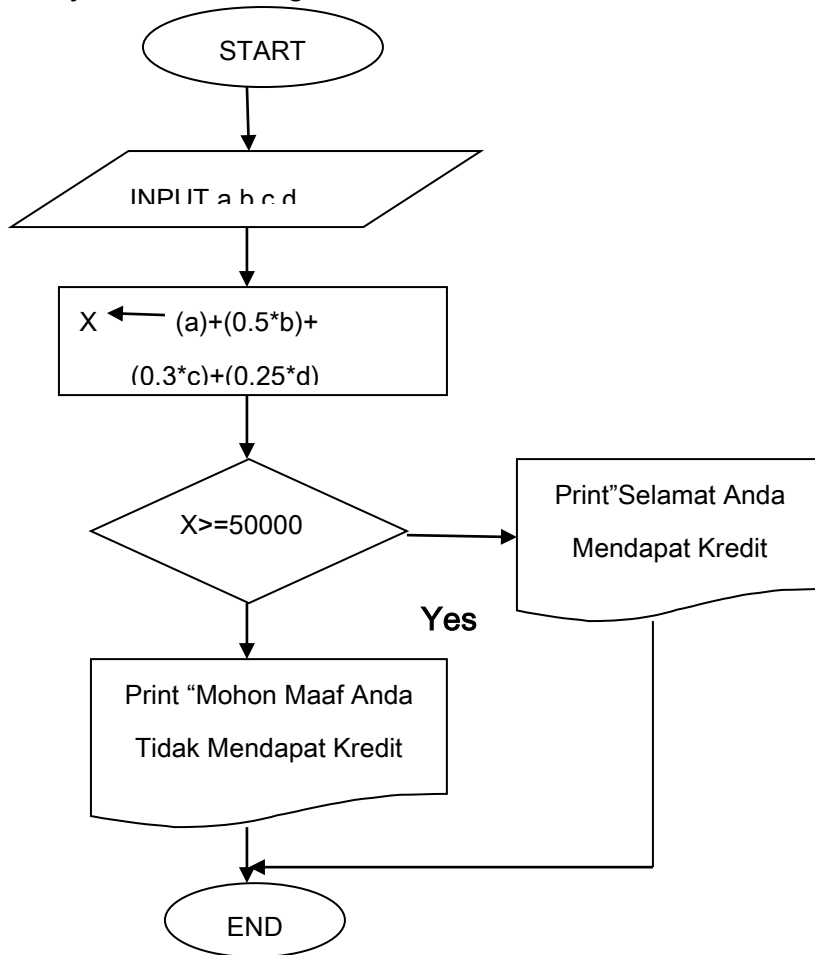


Dari Flowchart program 4.3 terlihat data yang dideklarasikan adalah data input dengan nilai a, b, c dan d sudah diketahui. Untuk Nilai Akhir menggunakan rata-rata dari nilai yang sudah dideklarasikan. Untuk logika If, jika rata-rata nilai

```
// Program 4.4
#include <iostream.h>
int main ()
{
// inisialisasi variabel :
// Deklarasi Variabel nama[i], nama keluarga[j], no hp[k], alamat[l]
long a, b, c, d; // a: Pendapatan ayah, b:ibu, c: anak dan d: lain-lain
char i[20],j[20],k[20],l[20];
long X; // Nilai Akhir
cout <<
"|=====|" <<
endl;
cout << "|          Layanan Kredit Kendaraan Dengan Logika IF          |" << endl;
cout <<
"|=====|" <<
endl;
cout <<
"|=====|" <<
endl;
cout << "|Menggunakan Logika Struktur Kontrol Percabangan Dengan IF |" << endl;
cout <<
"|=====|" <<
endl;
cout << " Inputkan Pendapatan Ayah   : ";
cin>>a;
cout << " Inputkan Pendapatan Ibu     : ";
cin>>b;
cout << " Inputkan Pendapatan Anak    : ";
cin>>c;
cout << " Inputkan Pendapatan Lain    : ";
```

```
// Menggunakan logika if untuk pengecekan 2 kondisi Jika total dengan menghitung
// komponen : pendapatan ayah dihitung penuh, ibu setengah, anak 1/3 dan lain-lain
// 1/4. Jika Hasil yang diperoleh dari pendapatan adalah sebagai berikut:
// Pendapatan>=5000000, Maka Anda Mendapat Kredit dan Jika Pendapatan
<5000000
// Maka tidak akan mendapatkan Layanan Kredit Kendaraan
if (X>=5000000)
cout << "          Selamat Anda Mendapat Kredit Kendaraan" << "\n";
else
cout << "          Maaf Anda Tidak Mendapat Kredit Apapun" << "\n";
// Mencetak Hasil Tampilan :
cout <<
"|=====|" <<
endl;
cout << "|      Mencetak Komponen Data Yang Sudah Diinputkan      |" << endl;
cout <<
"|=====|" <<
endl;
cout<<" Nama Anda          : "<<i<<"\n";
cout<<" Nama Keluarga Anda   : "<<j<<"\n";
cout<<" No HP Anda           : "<<k<<"\n";
cout<<" Alamat Anda          : "<<l<<"\n";
cout<<" Status Keuangan Anda : "<<X<<"\n";
// Menggunakan logika if untuk pengecekan 2 kondisi Jika Rata-Rata
// Pendapatan>=5000000, Maka Anda Mendapat Kredit dan Jika Pendapatan
<5000000
// Maka tidak akan mendapatkan Layanan Kredit Kendaraan
cout <<
"|=====|" <<
```

Dari program 4.4 terlihat menggunakan data *input*, dan logika *if* maka *flowchart*nya adalah sebagai berikut :



Dari *Flowchart* dan program 4.4 merupakan contoh program untuk mendapatkan kredit kendaraan dengan data *input*, yaitu pendapatan ayah, ibu, anak dan lain-lain. Perhitungan pendapatan adalah: "Pendapatan ayah dihitung penuh, pendapatan ibu dihitung setengah, pendapatan anak dihitung sepertiga dan pendapatan lain-lain dihitung seperempat. Dengan Kondisi, Jika hasil perhitungan pendapatan  $\geq 5000000$ , maka

Dari program 4.4 maka output yang dihasilkan adalah :

```

(Inactive D:\C_LAT~1\PRO44.EXE)
=====
Menggunakan Logika Struktur Kontrol Percabangan Dengan IF
=====
Inputkan Pendapatan Ayah      : 5000000
Inputkan Pendapatan Ibu       : 4000000
Inputkan Pendapatan Anak      : 6000000
Inputkan Pendapatan Lain      : 2000000
Inputkan Nama Anda            : FauziahKasyfi
Inputkan Nama Keluarga        : Abdi
Inputkan No.telp/Hp Anda      : 081383955321
Inputkan Alamat               : Kelapa Dua
                               Selamat Anda Mendapat Kredit Kendaraan
=====
Mencetak Komponen Data Yang Sudah Diinputkan
=====
Nama Anda                     : FauziahKasyfi
Nama Keluarga Anda            : Abdi
No HP Anda                    : 081383955321
Alamat Anda                   : Kelapa
Status Keuangan Anda          : 25500000
=====
Selamat Menggunakan Logika IF Terima Kasih FauziahKasyfi
=====
    
```

Gambar 4.4 Tampilan Output Menggunakan Logika IF

```

(Inactive D:\C_LAT~1\PRO44.EXE)
=====
Layanan Kredit Kendaraan Dengan Logika IF
=====
Menggunakan Logika Struktur Kontrol Percabangan Dengan IF
=====
Inputkan Pendapatan Ayah      : 500000
Inputkan Pendapatan Ibu       : 500000
Inputkan Pendapatan Anak      : 500000
Inputkan Pendapatan Lain      : 100000
Inputkan Nama Anda            : Kasyfi
Inputkan Nama Keluarga        : Abdi
Inputkan No.telp/Hp Anda      : 081383955321
Inputkan Alamat               : Kelapa Dua
                               Maaf Anda Tidak Mendapat Kredit Apapun
=====
Mencetak Komponen Data Yang Sudah Diinputkan
=====
Nama Anda                     : Kasyfi
Nama Keluarga Anda            : Abdi
No HP Anda                    : 081383955321
Alamat Anda                   : Kelapa
Status Keuangan Anda          : 2275000
=====
Selamat Menggunakan Logika IF Terima Kasih FauziahKasyfi
=====
    
```

Gambar 4.5 Tampilan Output Menggunakan Logika IF



## CONTOH PROGRAM C# DENGAN LOGIKA IF

```
//Program 4.5 Menggunakan Variabel Input
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace if4
{
    class Program
    {
        static void Main(string[] args)
        {
            String nama, npm, alm, ang, pr;
            Console.WriteLine("Inputkan Nama Mahasiswa      : ");
            nama = Console.ReadLine();
            Console.WriteLine("Inputkan Nomor Pokok Mahasiswa: ");
            npm = Console.ReadLine();
            Console.WriteLine("Inputkan Alamat Mahasiswa      : ");
            alm = Console.ReadLine();
            Console.WriteLine("Inputkan Angkatan Mahasiswa    : ");
            ang = Console.ReadLine();
            Console.WriteLine("Program Studi Mahasiswa        : ");
            pr = Console.ReadLine();
            Console.WriteLine("Inputkan Nilai Kehadiran       : ");
            double K = Convert.ToSingle(Console.ReadLine());
            Console.WriteLine("Inputkan Nilai Tugas Sblm UTS  : ");
            double TG = Convert.ToSingle(Console.ReadLine());
            Console.WriteLine("Inputkan Nilai Tugas Stlh UTS  : ");
            double AY = Convert.ToSingle(Console.ReadLine());
            Console.WriteLine("Inputkan Nilai UTS              : ");
            double L = Convert.ToSingle(Console.ReadLine());
            Console.WriteLine("Inputkan Nilai UAS              : ");
            double A = Convert.ToSingle(Console.ReadLine());
            double NA = (K*0.1)+(0.25 * TG)+(0.25 * AY)+(0.2 * L)+(0.2*A);
            string ket;
            if (NA >= 90)
                ket = "Selamat Anda Mendapat Nilai A";
            else if (NA >= 80)
                ket = "Selamat Anda Mendapat Nilai B";
            else if (NA >= 70)
                ket = "Selamat Anda Mendapat Nilai C";
            else if (NA >= 60)
                ket = "Nilai Anda Adalah D";
            else
                ket = "Maaf Anda Harus Mengulang ";
        }
    }
}
```

```

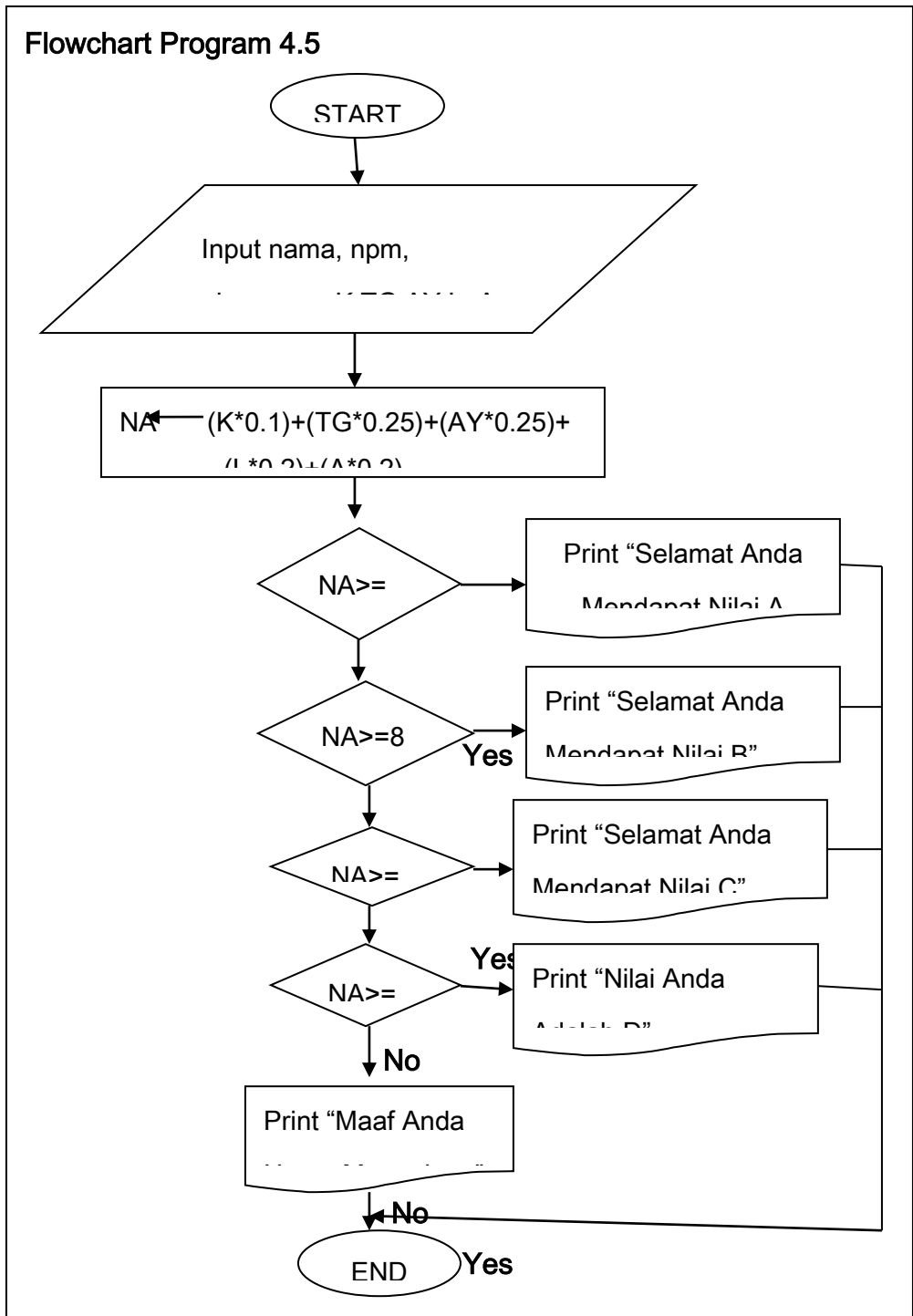
Console.WriteLine("Nilai Akhir Anda Adalah      : {0}", NA);
Console.WriteLine("=====");
Console.WriteLine("=====");
Console.WriteLine("Rangkuman Hasil Perolehan Nilai "+ nama);
Console.WriteLine("=====");
Console.WriteLine("=====");
Console.WriteLine("-----");
Console.WriteLine("| Nama      | Rangkuman Nilai      |");
Console.WriteLine("| Mahasiswa| KH      | T1      | T2      | UTS |
UAS | ");
Console.WriteLine("-----");
Console.WriteLine("| " + nama + " | "+ AY + " |" + TG + " |" + K +
"| " + L + "
| " + A + " |");
Console.WriteLine("-----");
Console.WriteLine("      Keterangan Nilai Anda Adalah: " + ket);
Console.WriteLine("::::::::::::::::::Thank You:::::::::::::::::: ");

    }
}
}

```

#### Keterangan Kode Program :

Dari program4.5 merupakan contoh program dengan logika *if* menggunakan program C# data *input* yaitu terdiri dari nama, npm, alamat, angkatan dan program studi dengan tipe data *string*. Untuk *input* nilai terdiri dari nilai kehadiran, nilai tugas sebelum UTS, nilai tugas setelah UTS, nilai UTS dan nilai UAS. Proses perhitungan nilai akhir menggunakan perhitungan nilai kehadiran dihitung 10%, nilai tugas sebelum UTS dan Sesudah UTS masing-masing 25%, Nilai UTS 20% dan Nilai UAS 20%.Kondisi yang digunakan adalah : Jika Nilai Akhir  $\geq 90$ , Maka Mendapat Nilai A. Jika Nilai Akhir  $\geq 80$  Mendapat Nilai B. Jika Nilai Akhir  $\geq 70$  Maka Mendapat Nilai C, Jika Nilai Akhir  $\geq 60$  Mendapatkan nilai D, dan Jika tidak memenuhi Kriteria yang telah



```
C:\Windows\system32\cmd.exe
Inputkan Nama Mahasiswa : Tsarwan Kasyfillah
Inputkan Nomor Pokok Mahasiswa: 112150001
Inputkan Alamat Mahasiswa : Kelapa Dua
Inputkan Angkatan Mahasiswa : 2015
Program Studi Mahasiswa : Tehnik Informatika
Inputkan Nilai Kehadiran : 90
Inputkan Nilai Tugas Sblm UTS : 90
Inputkan Nilai Tugas Stlh UTS : 90
Inputkan Nilai UTS : 90
Inputkan Nilai UAS : 90
Nilai Akhir Anda Adalah : 90

=====
Rangkuman Hasil Perolehan Nilai Tsarwan Kasyfillah
=====

| Nama | | Rangkuman Nilai | | | |
| Mahasiswa | KH | T1 | T2 | UTS | UAS |
| Tsarwan Kasyfillah | 90 | 90 | 90 | 90 | 90 |

Keterangan Nilai Anda Adalah: Selamat Anda Mendapat Nilai A
:::::::::::::::::::Thank You:::::::::::::::::::
Press any key to continue . . .
```

Output Dari Program 4.5 Adalah :

Gambar 4.6 Tampilan Output Menggunakan Program C# Dengan Logika IF Data Input

```
C:\Windows\system32\cmd.exe
Inputkan Nama Mahasiswa : Abdi Husairi Nst
Inputkan Nomor Pokok Mahasiswa: 12090909
Inputkan Alamat Mahasiswa : Mangga Besar
Inputkan Angkatan Mahasiswa : 2015
Program Studi Mahasiswa : Sistem Informasi
Inputkan Nilai Kehadiran : 80
Inputkan Nilai Tugas Sblm UTS : 80
Inputkan Nilai Tugas Stlh UTS : 80
Inputkan Nilai UTS : 80
Inputkan Nilai UAS : 80
Nilai Akhir Anda Adalah : 80

=====
Rangkuman Hasil Perolehan Nilai Abdi Husairi Nst
=====

| Nama | | Rangkuman Nilai | | | |
| Mahasiswa | KH | T1 | T2 | UTS | UAS |
| Abdi Husairi Nst | 80 | 80 | 80 | 80 | 80 |

Keterangan Nilai Anda Adalah: Selamat Anda Mendapat Nilai B
:::::::::::::::::::Thank You:::::::::::::::::::
Press any key to continue . . .
```

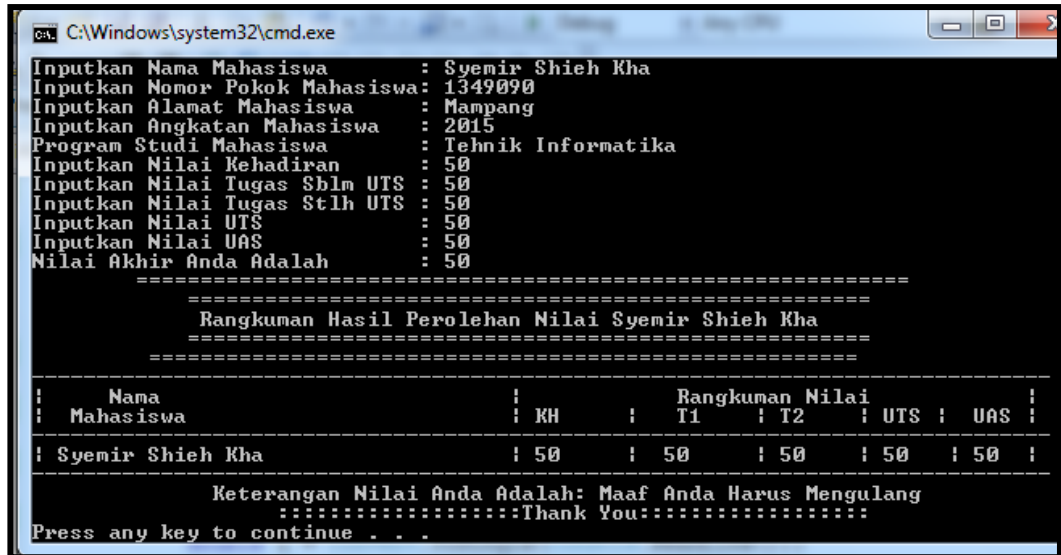
Gambar 4.7 Tampilan Output Menggunakan Program C# Dengan Logika IF Data Input

```
C:\Windows\system32\cmd.exe
Inputkan Nama Mahasiswa : Kahmir Solikhin N
Inputkan Nomor Pokok Mahasiswa: 19090909
Inputkan Alamat Mahasiswa : Cimanggis
Inputkan Angkatan Mahasiswa : 2015
Program Studi Mahasiswa : Tehnik Informatika
Inputkan Nilai Kehadiran : 70
Inputkan Nilai Tugas Sblm UTS : 70
Inputkan Nilai Tugas Stlh UTS : 70
Inputkan Nilai UTS : 70
Inputkan Nilai UAS : 70
Nilai Akhir Anda Adalah : 70
=====
Rangkuman Hasil Perolehan Nilai Kahmir Solikhin N
=====
| Nana | | Rangkuman Nilai | | | |
| Mahasiswa | KH | T1 | T2 | UTS | UAS |
| Kahmir Solikhin N | 70 | 70 | 70 | 70 | 70 |
Keterangan Nilai Anda Adalah: Selamat Anda Mendapat Nilai C
:::::::::::Thank You:::::::::::
Press any key to continue . . .
```

Gambar 4.8 Tampilan Output Menggunakan Program C# Dengan Logika IF Data Input

```
C:\Windows\system32\cmd.exe
Inputkan Nama Mahasiswa : Saza Khultimah F
Inputkan Nomor Pokok Mahasiswa: 9080890
Inputkan Alamat Mahasiswa : Klender
Inputkan Angkatan Mahasiswa : 2015
Program Studi Mahasiswa : Sistem Informasi
Inputkan Nilai Kehadiran : 60
Inputkan Nilai Tugas Sblm UTS : 60
Inputkan Nilai Tugas Stlh UTS : 60
Inputkan Nilai UTS : 60
Inputkan Nilai UAS : 60
Nilai Akhir Anda Adalah : 60
=====
Rangkuman Hasil Perolehan Nilai Saza Khultimah F
=====
| Nana | | Rangkuman Nilai | | | |
| Mahasiswa | KH | T1 | T2 | UTS | UAS |
| Saza Khultimah F | 60 | 60 | 60 | 60 | 60 |
Keterangan Nilai Anda Adalah: Nilai Anda Adalah D
:::::::::::Thank You:::::::::::
Press any key to continue . . .
```

Gambar 4.9 Tampilan Output Menggunakan Program C# Dengan Logika IF Data Input



```
C:\Windows\system32\cmd.exe
Inputkan Nama Mahasiswa      : Syemir Shieh Kha
Inputkan Nomor Pokok Mahasiswa: 1349090
Inputkan Alamat Mahasiswa    : Mampang
Inputkan Angkatan Mahasiswa  : 2015
Program Studi Mahasiswa     : Teknik Informatika
Inputkan Nilai Kehadiran     : 50
Inputkan Nilai Tugas Sblm UTS: 50
Inputkan Nilai Tugas Stlh UTS: 50
Inputkan Nilai UTS           : 50
Inputkan Nilai UAS           : 50
Nilai Akhir Anda Adalah     : 50

=====
                    Rangkuman Hasil Perolehan Nilai Syemir Shieh Kha
=====
-----
| Nama                               |                               | Rangkuman Nilai |                               |
| Mahasiswa                           | KH | T1 | T2 | UTS | UAS |
|-----|-----|-----|-----|-----|-----|
| Syemir Shieh Kha                   | 50 | 50 | 50 | 50 | 50 |
|-----|-----|-----|-----|-----|-----|
                    Keterangan Nilai Anda Adalah: Maaf Anda Harus Mengulang
                    :::::::::::::::::::::Thank You:::::::::::::::::::
Press any key to continue . . .
```

Gambar 4.10 Tampilan Output Menggunakan Program C# Dengan Logika IF Data Input

# Contoh Program C# Logika If Variabel Konstan

```
m="112898989";  
015";
```

**Logika& Algoritma(C++,C# dan Java)**

```
ah : "+nama);  
wa Adalah : "+npm);  
alah : "+alm);  
Adalah : "+ang);  
isw Adalah : "+pr);  
g diinput : "+K);  
UTS diinput: "+TG);  
S diinput : "+AY);  
"+S);
```

```
Console.WriteLine("=====");
Console.WriteLine("=====");
Console.WriteLine(" Rangkuman Hasil Perolehan Nilai " + nama);
Console.WriteLine("=====");
Console.WriteLine("=====");
Console.WriteLine("-----");
Console.WriteLine("| Nama      | Rangkuman Nilai      |");
Console.WriteLine("|Mahasiswa| KH      | T1      | T2      | UTS |
UAS | ");
Console.WriteLine("-----");
Console.WriteLine("| " + nama + " | " + AY + "      | " + TG + "
| " + K + "      | " + L + "      | " + A + " | ");
Console.WriteLine("-----");
Console.WriteLine(" Keterangan Nilai Anda Adalah: " + ket);
Console.WriteLine(" :::::::::::Thank You::::::::: ");

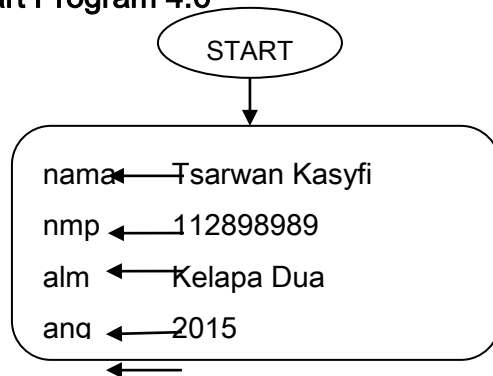
    }
}
}
```

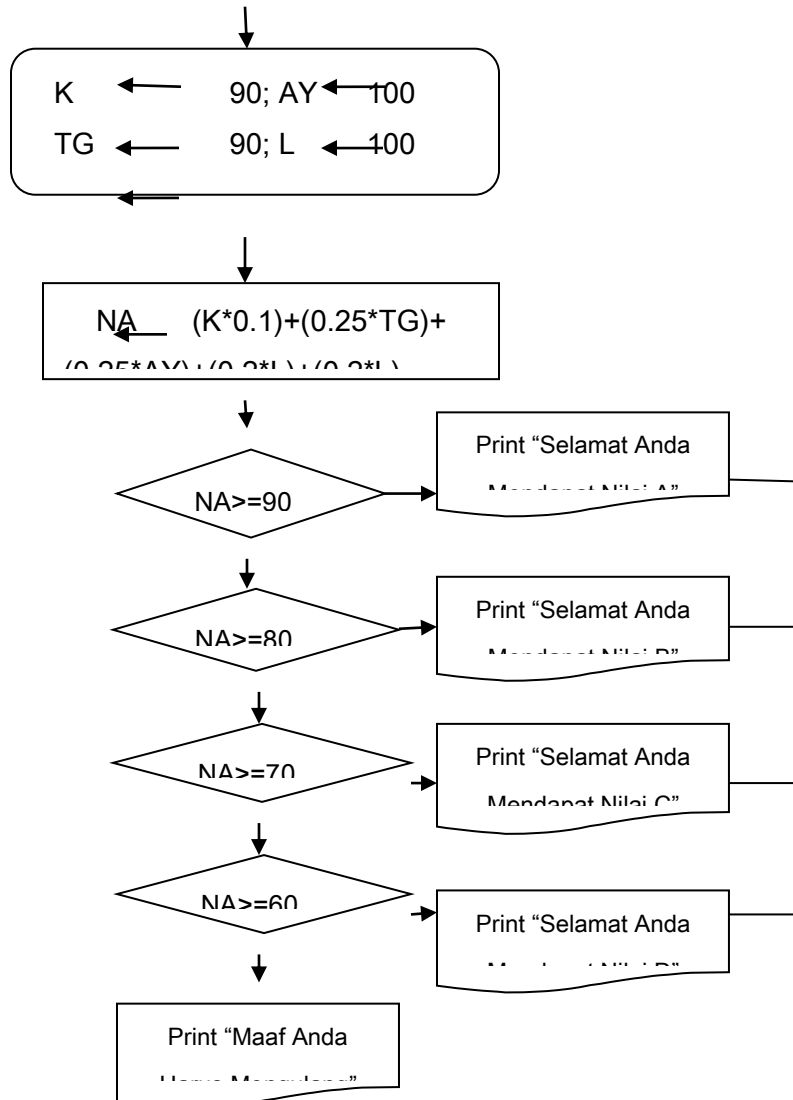


**Keterangan Kode Program :**

Dari program4.6 merupakan contoh program dengan logika *if* menggunakan program C# data Konstan yaitu terdiri dari nama, npm, alamat, angkatan dan program studi dengan tipe data *string*. Untuk data konstan nilai terdiri dari nilai kehadiran, nilai tugas sebelum UTS, nilai tugas setelah UTS, nilai UTS dan nilai UAS. Proses perhitungan nilai akhir menggunakan perhitungan nilai kehadiran dihitung 10%, nilai tugas sebelum UTS dan Sesudah UTS masing-masing 25%, Nilai UTS 20% dan Nilai UAS 20%.Kondisi yang digunakan adalah : Jika Nilai Akhir  $\geq 90$ , Maka Mendapat Nilai A. Jika Nilai Akhir  $\geq 80$  Mendapat Nilai B. Jika Nilai Akhir  $\geq 70$  Maka Mendapat Nilai C, Jika Nilai Akhir  $\geq 60$  Mendapatkan nilai D, dan Jika tidak memenuhi Kriteria yang telah

**Flowchart Program 4.6**





Dari *flowchart* program 4.6 menggunakan data konstan, maka *output* yang dapat ditampilkan adalah sebagai berikut :

```

C:\Windows\system32\cmd.exe
Nama Mahasiswa Adalah      : Isarwan Kasyfi
Nomor Pokok Mahasiswa Adalah : 112898989
Alamat Mahasiswa Adalah    : Kelapa Dua
Angkatan Mahasiswa Adalah  : 2015
Program Studi Mahasiswa Adalah : Sistem Informasi
Nilai Kehadiran Yang diinput : 90
Nilai Tugas Sebelum UTS diinput : 90
Nilai Tugas Stlh UTS diinput : 100
Nilai UTS diinput          : 100
Nilai UAS diinput          : 100
Nilai Akhir Anda Adalah    : 96.5

Rangkuman Hasil Perolehan Nilai Isarwan Kasyfi
  
```



# CONTOH PROGRAM LOGIKA IF DENGAN JAVA

## Contoh Program Java Menggunakan Logika IF dengan Data Inputan

```
//Program 4.7
import java.io.*;
public class Logikaif1 {
public static void main(String[] args)throws Exception {
BufferedReader input = new BufferedReader
(new InputStreamReader (System.in));
String nama,NPM, Alamat, Angkatan, Semester, NA,NB,NC,ND,NE, NF, NG;
double a,b,c,d,e,f,g,x,y;
System.out.println("=====
=");
System.out.println("| Menggunakan Operator Aritmatika Dengan Data Input &
Logika IF |");
System.out.println("=====
=");
// Menginputkan Biodata Siswa
System.out.print (" Inputkan Nama Anda      : ");
nama = input.readLine();
System.out.print (" Inputkan Nomor Pokok Siswa  : ");
NPM = input.readLine();
System.out.print (" Inputkan Alamat Anda      : ");
Alamat = input.readLine();
System.out.print (" Inputkan Angkatan Anda    : ");
Angkatan = input.readLine();
```

# CONTOH PROGRAM LOGIKA IF DENGAN JAVA

## Contoh Program Java Menggunakan Logika IF dengan Data Inputan

```
//Program 4.7
import java.io.*;
public class Logikaif1 {
public static void main(String[] args)throws Exception {
BufferedReader input = new BufferedReader
(new InputStreamReader (System.in));
String nama,NPM, Alamat, Angkatan, Semester, NA,NB,NC,ND,NE, NF, NG;
double a,b,c,d,e,f,g,x,y;
System.out.println("=====
=|");
System.out.println("| Menggunakan Operator Aritmatika Dengan Data Input &
Logika IF |");
System.out.println("=====
=|");
// Menginputkan Biodata Siswa
System.out.print (" Inputkan Nama Anda      : ");
nama = input.readLine();
System.out.print (" Inputkan Nomor Pokok Siswa  : ");
NPM = input.readLine();
System.out.print (" Inputkan Alamat Anda      : ");
Alamat = input.readLine();
```

# CONTOH PROGRAM LOGIKA IF DENGAN JAVA

```

System.out.print (" Inputkan Nilai Fisika      : ");
NA = input.readLine();
System.out.print (" Inputkan Nilai Matematika  : ");
NB = input.readLine();
System.out.print (" Inputkan Nilai Kimia      : ");
NC = input.readLine();
System.out.print (" Inputkan Nilai Biologi   : ");
ND = input.readLine();
// Menginputkan nilai Non Eksak terdiri dari:
// E: Nilai Ekonomi, F: Nilai Akuntansi, G: Nilai B. Inggris
System.out.println("|=====|");
System.out.println("|  Menginputkan Nilai Non Eksak    |");
System.out.println("|=====|");
System.out.print (" Inputkan Ekonomi      : ");
NE = input.readLine();
System.out.print (" Inputkan Nilai Akuntansi   : ");
NF = input.readLine();
System.out.print (" Inputkan Nilai Bahasa Inggris : ");
NG = input.readLine();
System.out.println("|=====|");
a = Double.parseDouble(NA);
b = Double.parseDouble(NB);
c = Double.parseDouble(NC);
d = Double.parseDouble(ND);
e = Double.parseDouble(NE);
f = Double.parseDouble(NF);

```

# CONTOH PROGRAM LOGIKA IF DENGAN JAVA

```
System.out.println("");  
// Logika if  
if (x > y)  
System.out.println("Selamat Anda ditempatkan di Jurusan IPA");  
else if (x < y)  
System.out.println("Selamat Anda ditempatkan di Jurusan IPS");  
else  
System.out.print("Silahkan Anda Boleh Memilih Jurusan IPA atau IPS");  
System.out.println("|=====|  
|=|");  
System.out.println("| Selamat Mencoba Menggunakan Logika IF      |");  
System.out.println("|      Dengan Data Inputan              |");
```

Sys  
Sys  
|=|)  
Sys  
}  
}  
**Keterangan Kode Program**  
Dari program 4.7 terlihat menggunakan *variable Input* dengan nama *variable* adalah : nama,NPM, Alamat, Angkatan, Semester, NA,NB,NC,ND,NE, NF, NG dengan tipe data *string*. Untuk *variable input* data nilai eksak adalah : a[nilai Fisika], b[nilai Matematika], c[nilai Kimia], d[nilai Biologi], Untuk *variable input* nilai non eksak adalah :d[nilai Ekonomi], e[nilai Akuntansi], e[nilai Bahasa Inggris] dengan tipe data *double*. Untuk Nilai akhir eksak menggunakan *variable* [x] dan nilai non eksak menggunakan *variable* [y], dengan tipe data *double*. Proses perhitungan nilai eksak dan non eksak menggunakan  $x=[a+b+c+d]/4$

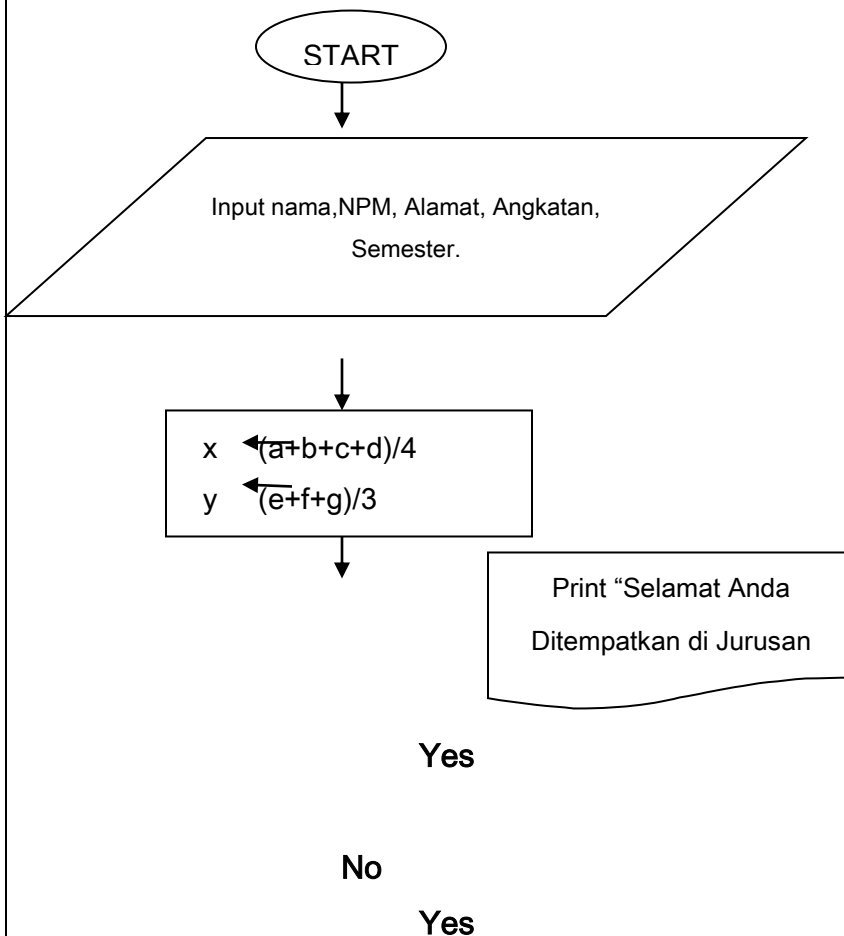
Untuk logika If menggunakan 3 buah kondisi yaitu :

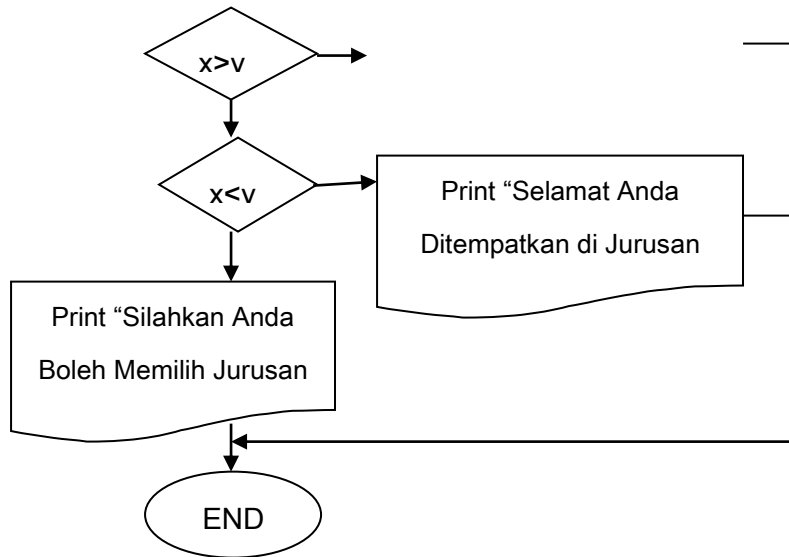
- a. Jika nilai rata-rata nilai eksak  $>$  dari nilai rata-rata non eksak maka “ Selamat Anda ditempatkan di Jurusan IPA”.
- b. Jika nilai rata-rata nilai eksak  $<$  dari nilai rata-rata non eksak maka “ Selamat Anda ditempatkan di Jurusan IPS”
- c. Jika nilai rata-rata nilai eksak = rata-rata nilai non eksak maka “Silahkan Anda Boleh Memilih Jurusan IPA atau IPS”



```
Copyright Files Software Director of Jember 2015
Megggunakan Operator Aritmatika Dengan Data Input & Logika IP
Inputkan Nama Anda           : Tsarwan Kazyfillah Nasution
Inputkan Nomor Pokok Siswa   : 1909101910
Inputkan Alamat Anda         : Kelapa Dua, Depok
Inputkan Angkatan Anda       : 2015
Inputkan Semester Anda       : 1[Satu]
Menginputkan Nilai Eksak
Inputkan Nilai Fisika         : 90
Inputkan Nilai Matematika     : 90
Inputkan Nilai Kimia          : 90
Inputkan Nilai Biologi        : 90
Menginputkan Nilai Non Eksak
Inputkan Ekonomi              : 90
Inputkan Nilai Akuntansi      : 80
Inputkan Nilai Bahasa Inggris : 100
Hasil Rata - Rata Nilai Eksak dan Non Eksak
Hasil Nilai Eksak Adalah     : 92.66666666666667
Hasil Nilai Non Eksak Adalah : 90.8
Selamat anda ditempatkan di Jurusan IPA
Selamat Mencoba Menggunakan Logika IP
Dengan Data Inputan
Terima Kasih Banyak @ FauziahKazyfi2015
Press any key to continue...
```

Flowchart Program 4.7





## CONTOH PROGRAM LOGIKA IF DENGAN JAVA

```
// program 4.8
class IF2 {
public static void main(String[] args) {
String nama="Tsarwan Kasyfillah Nasution";
String NPM= "12909090";
String Alamat="Kelapa Dua, Depok";
String Angkatan="2015";
double x,y;
double a= 90;
double b= 90;
double c= 100;
double d= 100;
double e= 88.8;
double f= 90.2;double g=88.20;
System.out.println("=====
=");
System.out.println(" Menggunakan Operator Aritmatika Dengan Data Konstan
```

## CONTOH PROGRAM LOGIKA IF DENGAN JAVA

```
// Mencetak nilai Non Eksak terdiri dari:
// e: Nilai Ekonomi, f: Nilai Akuntansi, g: Nilai B. Inggris
System.out.println("=====|");
System.out.println("    Mencetak Nilai Non Eksak    |");
System.out.println("=====|");
System.out.println(" Nilai Ekonomi Adalah    :"+e);
System.out.println(" Nilai Akuntansi Adalah   :"+f);
System.out.println(" Nilai Bahasa Inggris Adalah :"+g);
System.out.println("=====|");
// Menghitung nilai rata-rata Eksak dan Non Eksak
x = (a+b+c)/3;
y = (d+e+f+g)/4;
System.out.println("=====|")
;
System.out.println("|Hasil Rata - Rata Nilai Eksak dan Non Eksak|");
System.out.println("=====|")
;
```

### Keterangan Kode Program

Dari program 4.8 terlihat menggunakan *variable Konstan* dengan nama *variable* adalah : nama,NPM, Alamat, Angkatan, Semester, NA,NB,NC,ND,NE, NF, NG dengan tipe data *string*. Untuk *variable Konstan* data nilai eksak adalah : a[nilai Fisika], b[nilai Matematika], c[nilai Kimia], d[nilai Biologi], Untuk *variable* input nilai non eksak adalah :d[nilai Ekonomi], e[nilai Akuntansi], e[nilai Bahasa Inggris] dengan tipe data *double*. Untuk Nilai akhir eksak menggunakan *variable* [x] dan nilai non eksak menggunakan *variable* [y], dengan tipe data *double*. Proses perhitungan nilai eksak dan non eksak menggunakan  $x=[a+b+c+d]/4$  untuk nilai non eksak  $[e+f+g]/3$ , dengan tipe data *double*.

Untuk logika If menggunakan 3 buah kondisi yaitu :

- a. Jika nilai rata-rata nilai eksak > dari nilai rata-rata non eksak maka “ Selamat Anda ditempatkan di Jurusan IPA”.
- b. Jika nilai rata-rata nilai eksak < dari nilai rata-rata non eksak maka



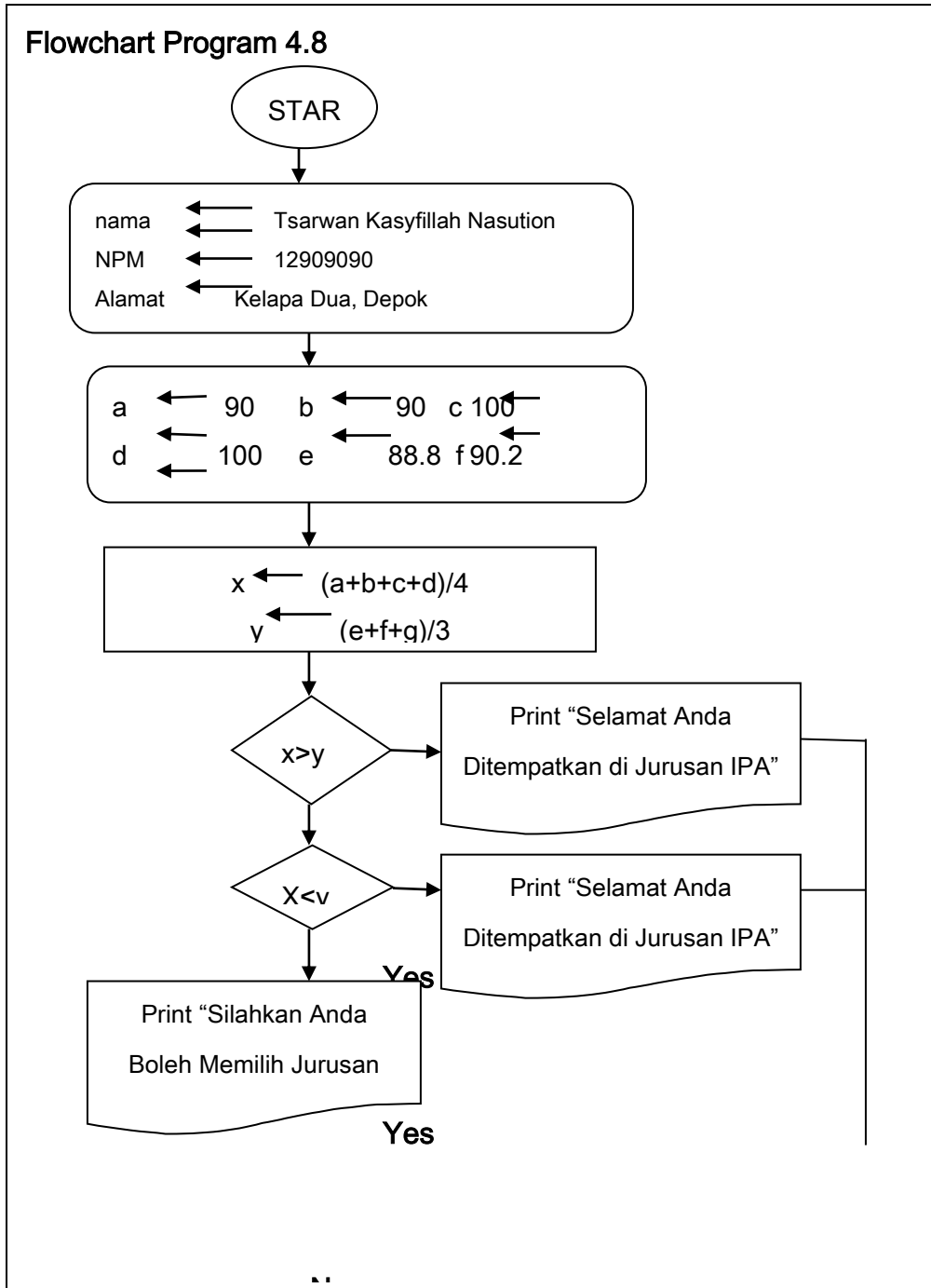
Output Dari Program 4.8 Adalah :

```
C:\Program Files\Xinox Software\JCreatorV3LE\GE2001.exe
=====
: Menggunakan Operator Aritmatika Dengan Data Konstan dan Logika IF
: =====
Nama Anda Adalah      : Tsarwan Kasyfillah Nasution
Nomor Pokok Siswa Adalah : 12909090
Alamat Anda Adalah    : Kelapa Dua, Depok
Angkatan Anda Adalah  : 2015
=====
: Mencetak Nilai Eksak      :
: =====
Nilai Fisika Adalah    : 90.0
Nilai Matematika Adalah : 90.0
Nilai Kimia Adalah     : 100.0
Nilai Biologi Adalah   : 100.0
=====
: Mencetak Nilai Non Eksak :
: =====
Nilai Ekonomi Adalah   :88.8
Nilai Akuntansi Adalah :90.2
Nilai Bahasa Inggris Adalah :88.2
=====
: Hasil Rata - Rata Nilai Eksak dan Non Eksak :
: =====
Hasil Nilai Eksak Adalah : 93.33333333333333
Hasil Nilai Non Eksak Adalah : 91.8

Selamat Anda ditempatkan di Jurusan IPA
=====
: Selamat Mencoba Menggunakan Logika IF
: Dengan Data Inputan
: Terima Kasih Banyak @ FauziahKasyfi2015
: =====
Press any key to continue..._
```

Gambar 4.13 Tampilan Output Dengan Logika IF Menggunakan Variabel Konstan

### Flowchart Program 4.8



## 4.4 Struktur Control Dengan Logika Switch

### 2. Struktur Control Dengan Logika Switch

*Struktur Control* dengan Logika *Switch*, merupakan *struktur control* yang digunakan untuk mengecek suatu kondisi dengan banyak pilihan. *Struktur logika switch* digunakan untuk menerangkan kode program jika kita menggunakan logika *IF*

**Switch** (*ekspresi*)

{

**Case** Konstanta1 :

Pernyataan1 ;

**Break** ;

**Case** Konstanta2 :

Pernyataan2 ;

**Break** ;

**Case** Konstanta3 :

Pernyataan3 ;

**Break** ;

:

:

**Case** KonstantaN :

PernyataanN ;

- *Switch* Ekspresi Berisi variabel yang telah kita deklarasikan.
- *Case* Konstanta; Berisi Nilai Konstanta [nilai tetap]
- Pernyataan1; Berisi instruksi atau pernyataan
- *Break*; menghentikan proses sementara Pada Blok Program
- *Default* : Pilihan yang



# Struktur Control Dengan Logika Switch

## Contoh Program Menggunakan Logika *Switch* Dengan C++ Data *Input*

```
// Program 4.9
//Program Menentukan Jenis Pembayaran Cicilan Kendaraan
#include <iostream.h>
#include <iomanip.h>
//Tampilan Menu
int main()
{
cout<<"-----"<<endl;
cout<<"-----"<<endl;
cout<<"-----Jenis Pembayaran Cicilan Kendaraan-----"<<endl;
cout<<"-----"<<endl;
cout<<"-----"<<endl;
cout<<"1. Motor"<<endl;
cout<<"2. Mobil"<<endl;
cout<<"3. Truk"<<endl;
//Tampilan Pilihan Jenis Cicilan
cout<<"Inputkan Jenis Pembayaran Cicilan : ";
int JenPil;// mendeklarasikan variabel ekspresi
cin>>JenPil;
//Menggunakan Logika Switch
switch(JenPil)
{
case 1:
cout<<"-----"<<endl;
cout<<"-----"<<endl;
```

# Struktur Control Dengan Logika Switch

## Contoh Program Menggunakan Logika *Switch* Dengan C++ Data *Input*

```
case 2:
cout<<"-----"<<endl;
cout<<"-----"<<endl;
cout<<"|      Jenis Kendaraan Mobil      |"<<endl;
cout<<"|-----|"<<endl;
cout<<"|Jika Membayar Tunai      | Rp.120,000,000  |"<<endl;
cout<<"|-----|"<<endl;
cout<<"|  Jika Membayar Cicil*   |Rp.1,000,000/Bulan|"<<endl;
cout<<"-----"<<endl;
cout<<"-----"<<endl;
cout<<" "<<endl;
cout<<" *Proses Pembayaran Jika Kredit Selama 10Tahun "<<endl;
break;
case 3:
cout<<"-----"<<endl;
cout<<"-----"<<endl;
cout<<"|      Jenis Kendaraan Truk      |"<<endl;
cout<<"|-----|"<<endl;
cout<<"|Jika Membayar Tunai|   Rp.100,000,000   |"<<endl;
cout<<"|-----|"<<endl;
cout<<"|Jika Membayar Cicil[Kredit]*|Rp.900,000/Bulan |"<<endl;
cout<<"-----"<<endl;
cout<<"-----"<<endl;
cout<<" "<<endl;
cout<<" *Proses Pembayaran Jika Kredit Selama 10Tahun "<<endl;
```

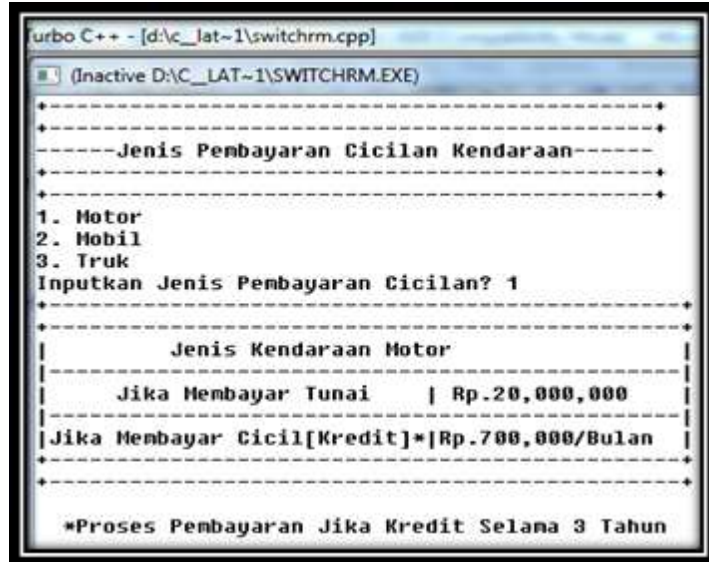
# Keterangan Kode Program

Dari Program 4.9 menggunakan data *input* dengan nama variabel adalah `JenPil`[Jenis Pilihan] dengan tipe integer, yaitu dengan menginputkan nilai numerik[1.2.3 dan seterusnya]. Untuk jenis pilihan yang digunakan pada program 4.9 adalah 3 buah jenis pilihan yaitu :

- a. Jika memilih pilihan 1, maka akan menjalankan statement jenis kendaraan bermotor.
- b. Jika memilih pilihan 2, maka akan menjalankan statement jenis kendaraan mobil
- c. Jika memilih pilihan 3, maka akan menjalankan statement jenis kendaraan truk.

Selain 3 pilihan yang dipilih, maka akan muncul pernyataan “Mohon Maaf Pilihan Anda Tidak Tertera *Input* Kembali Data Anda Yang Benar”. Untuk setiap akhir dari pernyataan, kita menggunakan statement *Break*, yang fungsinya adalah untuk menghentikan proses pilihan dan program hanya membaca sub pilihan yang kita pilih, misalnya jenis pilihan [1], dan proses yang terjadi hanya mengecek pernyataan – pernyataan 1 saja, tanpa melihat pernyataan berikutnya. Fungsi *Break* disini juga memberikan keterangan bahwa antar jenis pilihan tidak akan

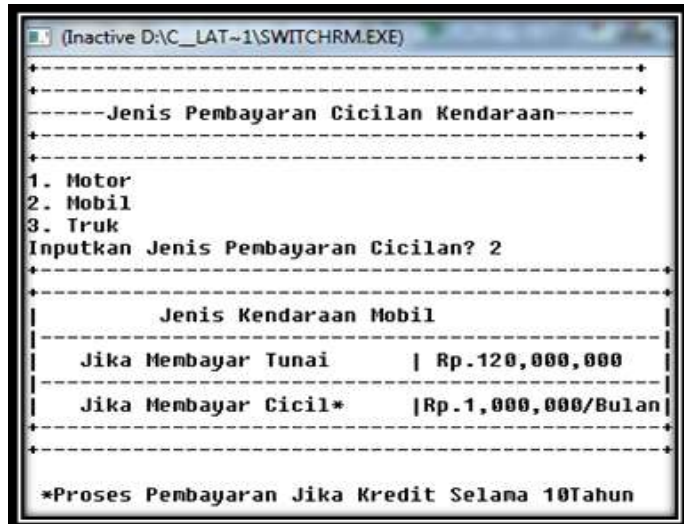
- Output Program dengan Pilihan 1



```
urbo C++ - [d:\c_lat~1\switchrm.cpp]
(Inactive D:\C_LAT~1\SWITCHRM.EXE)
-----Jenis Pembayaran Cicilan Kendaraan-----
1. Motor
2. Mobil
3. Truk
Inputkan Jenis Pembayaran Cicilan? 1
-----
|          Jenis Kendaraan Motor          |
|-----|-----|
| Jika Membayar Tunai      | Rp.20,000,000 |
|-----|-----|
| Jika Membayar Cicil[Kredit]*|Rp.700,000/Bulan|
|-----|-----|
*Proses Pembayaran Jika Kredit Selama 3 Tahun
```

Gambar 4.14 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel/ Input*

- Output Program dengan Pilihan 2



```
(Inactive D:\C_LAT~1\SWITCHRM.EXE)
-----Jenis Pembayaran Cicilan Kendaraan-----
1. Motor
2. Mobil
3. Truk
Inputkan Jenis Pembayaran Cicilan? 2
-----
|          Jenis Kendaraan Mobil          |
|-----|-----|
| Jika Membayar Tunai      | Rp.120,000,000 |
|-----|-----|
| Jika Membayar Cicil*    |Rp.1,000,000/Bulan|
|-----|-----|
*Proses Pembayaran Jika Kredit Selama 10Tahun
```

- Output Program dengan Pilihan 3

```
(Inactive D:\C_LAT~1\SWITCHRM.EXE)
-----Jenis Pembayaran Cicilan Kendaraan-----
1. Motor
2. Mobil
3. Truk
Inputkan Jenis Pembayaran Cicilan? 3
-----
| Jenis Kendaraan Truk |
| Jika Membayar Tunai| Rp.100,000,000 |
| Jika Membayar Cicil[Kredit]*|Rp.900,000/Bulan |
-----
*Proses Pembayaran Jika Kredit Selama 10Tahun
```

Gambar 4.16 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel* Input

- Output Program dengan menginputkan jenis pilihan yang tidak ada pada program misalnya kita inputkan 4

```
(Inactive D:\C_LAT~1\SWITCHRM.EXE)
-----Jenis Pembayaran Cicilan Kendaraan-----
1. Motor
2. Mobil
3. Truk
Inputkan Jenis Pembayaran Cicilan? 4
-----
Mohon Maaf Pilihan Anda Tidak Tertera---
Input Kembali Data Anda Yang Benar ---
-----
```

Gambar 4.17 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel* Input

## Contoh Program Menggunakan Logika Switch Dengan Data Konstan

```
//Program 4.10
//Program Menentukan Jenis Pembayaran Cicilan Kendaraan
#include <iostream.h>
#include <iomanip.h>
//Tampilan Menu
int main()
{
cout<<"-----"<<endl;
cout<<"-----"<<endl;
cout<<"-----Jenis Pembayaran Cicilan Kendaraan-----"<<endl;
cout<<"-----"<<endl;
cout<<"-----"<<endl;
cout<<"1. Motor"<<endl;
cout<<"2. Mobil"<<endl;
cout<<"3. Truk"<<endl;
//Tampilan Pilihan Jenis Cicilan
// mendeklarasikan variabel konstan jenis pilihan 1
int JenPil=1;
//Menggunakan Logika Switch
switch(JenPil)
{
case 1:
cout<<"-----"<<endl;
cout<<"-----"<<endl;
cout<<"| Jenis Kendaraan Bermotor |"<<endl;
cout<<"|-----|"<<endl;
```

```
cout<<" +-----+"<<endl;
cout<<" +-----+"<<endl;
cout<<"|    Jenis Kendaraan Motor    |"<<endl;
cout<<"|-----|"<<endl;
cout<<"|  Jika Membayar Tunai    | Rp.120,000,000  |"<<endl;
cout<<"|-----|"<<endl;
cout<<"|  Jika Membayar Cicil*   |Rp.1,000,000/Bulan|"<<endl;
cout<<" +-----+"<<endl;
cout<<" +-----+"<<endl;
cout<<" "<<endl;
cout<<" *Proses Pembayaran Jika Kredit Selama 10Tahun "<<endl;
break;
case 3:
cout<<" +-----+"<<endl;
cout<<" +-----+"<<endl;
cout<<"|    Jenis Kendaraan Truk    |"<<endl;
cout<<"|-----|"<<endl;
cout<<"|Jika Membayar Tunai|   Rp.100,000,000   |"<<endl;
cout<<"|-----|"<<endl;
cout<<"|Jika Membayar Cicil[Kredit]*|Rp.900,000/Bulan |"<<endl;
cout<<" +-----+"<<endl;
cout<<" +-----+"<<endl;
cout<<" "<<endl;
cout<<" *Proses Pembayaran Jika Kredit Selama 10Tahun "<<endl;
break;
default:
cout<<" "<<endl;
```

## Keterangan Kode Program

Dari Program 4.10 menggunakan data *konstan* dengan nama variabel adalah JenPil[Jenis Pilihan] dengan tipe integer, yaitu dengan memberikan nilai numerik misalnya nilai jenis pilihan [JenPil=1], maka secara otomatis Proses program akan menuju ke Switch 1.

Jika memilih pilihan 1, maka akan menjalankan statement jenis kendaraan bermotor dan akan memproses isi data yang ada pada pilihan 1 yang berisi “Jenis Kendaraan Motor”

Untuk lebih jelasnya dapat dilihat *output* berikut ini :

### Menggunakan Data Konstan Dengan Memberikan Nilai Tetap Untuk Jenis Pilihan [JenPil=1]

```
(inactive D:\C_LAT-1\SWITCHKO.EXE)
----->----->
----->----->
-----Jenis Pembayaran Cicilan Kendaraan----->
----->----->
1. Motor
2. Mobil
3. Truk
----->----->
Jenis Kendaraan Motor
----->----->
Jika Membayar Tunai | Rp.20,000,000
----->----->
Jika Membayar Cicil[Kredit]=|Rp.700,000/Bulan
----->----->
*Proses Pembayaran Jika Kredit Selama 3 Tahun
```



## Contoh Program Menggunakan Logika Switch Dengan Data Konstan

```
//Program 4.11
//Program Menentukan Jenis Pembayaran Cicilan Kendaraan
#include <iostream.h>
#include <iomanip.h>
//Tampilan Menu
int main()
{
cout<<"-----" <<endl;
cout<<"-----" <<endl;
cout<<"-----Jenis Pembayaran Cicilan Kendaraan-----" <<endl;
cout<<"-----" <<endl;
cout<<"-----" <<endl;
cout<<"1. Motor" <<endl;
cout<<"2. Mobil" <<endl;
cout<<"3. Truk" <<endl;
//Tampilan Pilihan Jenis Cicilan
// mendeklarasikan variabel konstan jenis pilihan 1
int JenPil=2;
//Menggunakan Logika Switch
switch(JenPil)
{
case 1:
cout<<"-----" <<endl;
cout<<"-----" <<endl;
cout<<"|      Jenis Kendaraan Motor      |" <<endl;
cout<<"|-----|" <<endl;
```

```
cout<<" +-----+"<<endl;
cout<<" +-----+"<<endl;
cout<<"|      Jenis Kendaraan Mobil      |"<<endl;
cout<<"|-----|"<<endl;
cout<<"| Jika Membayar Tunai   | Rp.120,000,000 |"<<endl;
cout<<"|-----|"<<endl;
cout<<"| Jika Membayar Cicil*   |Rp.1,000,000/Bulan|"<<endl;
cout<<" +-----+"<<endl;
cout<<" +-----+"<<endl;
cout<<" "<<endl;
cout<<" *Proses Pembayaran Jika Kredit Selama 10Tahun "<<endl;
break;
case 3:
cout<<" +-----+"<<endl;
cout<<" +-----+"<<endl;
cout<<"|      Jenis Kendaraan Truk      |"<<endl;
cout<<"|-----|"<<endl;
cout<<"|Jika Membayar Tunai|   Rp.100,000,000   |"<<endl;
cout<<"|-----|"<<endl;
cout<<"|Jika Membayar Cicil[Kredit]*|Rp.900,000/Bulan |"<<endl;
cout<<" +-----+"<<endl;
cout<<" +-----+"<<endl;
cout<<" "<<endl;
cout<<" *Proses Pembayaran Jika Kredit Selama 10Tahun "<<endl;
break;
default:
cout<<" "<<endl;
```

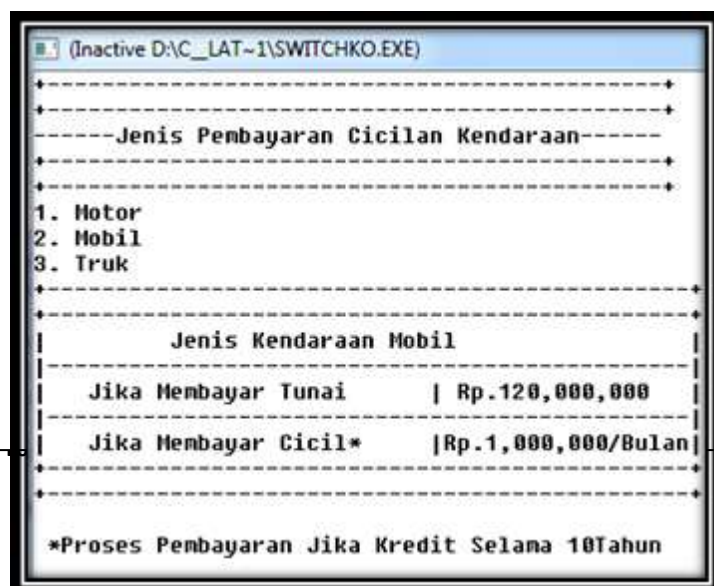
## Keterangan Kode Program

Dari Program 4.11 menggunakan data *konstan* dengan nama variabel adalah `JenPil[Jenis Pilihan]` dengan tipe integer, yaitu dengan memberikan nilai numerik misalnya nilai jenis pilihan [`JenPil=2`], maka secara otomatis Proses program akan menuju ke Switch 2.

Jika memilih pilihan 2, maka akan menjalankan statement jenis kendaraan Mobil dan akan memproses isi data yang ada pada pilihan 2 yang berisi "Jenis Kendaraan Mobil"

Untuk lebih jelasnya dapat dilihat *output* berikut ini :

### Menggunakan Data Konstan Dengan Memberikan Nilai Tetap Untuk Jenis Pilihan [`JenPil=2`]



```
(Inactive D:\C_LAT-1\SWITCHKO.EXE)
----- Jenis Pembayaran Cicilan Kendaraan -----
1. Motor
2. Mobil
3. Truk
Jenis Kendaraan Mobil
Jika Membayar Tunai | Rp.120,000,000
Jika Membayar Cicil* |Rp.1,000,000/Bulan
*Proses Pembayaran Jika Kredit Selama 10Tahun
```

## Contoh Program Menggunakan Logika Switch Dengan Data Konstan

```
//Program 4.12
//Program Menentukan Jenis Pembayaran Cicilan Kendaraan
#include <iostream.h>
#include <iomanip.h>
//Tampilan Menu
int main()
{
    cout<<"-----"<<endl;
    cout<<"-----"<<endl;
    cout<<"-----Jenis Pembayaran Cicilan Kendaraan-----"<<endl;
    cout<<"-----"<<endl;
    cout<<"-----"<<endl;
    cout<<"1. Motor"<<endl;
    cout<<"2. Mobil"<<endl;
    cout<<"3. Truk"<<endl;
    //Tampilan Pilihan Jenis Cicilan
    // mendeklarasikan variabel konstan jenis pilihan 1
    int JenPil=3;
    //Menggunakan Logika Switch
    switch(JenPil)
    {
    case 1:
        cout<<"-----"<<endl;
        cout<<"-----"<<endl;
        cout<<"|    Jenis Kendaraan Motor    |"<<endl;
        cout<<"|-----|"<<endl;
```

```
cout<<" +-----+"<<endl;
cout<<" +-----+"<<endl;
cout<<"|      Jenis Kendaraan Mobil      |"<<endl;
cout<<"|-----|"<<endl;
cout<<"| Jika Membayar Tunai | Rp.120,000,000 |"<<endl;
cout<<"|-----|"<<endl;
cout<<"| Jika Membayar Cicil* |Rp.1,000,000/Bulan|"<<endl;
cout<<" +-----+"<<endl;
cout<<" +-----+"<<endl;
cout<<" "<<endl;
cout<<" *Proses Pembayaran Jika Kredit Selama 10Tahun "<<endl;
break;
case 3:
cout<<" +-----+"<<endl;
cout<<" +-----+"<<endl;
cout<<"|      Jenis Kendaraan Truk      |"<<endl;
cout<<"|-----|"<<endl;
cout<<"|Jika Membayar Tunai| Rp.100,000,000 |"<<endl;
cout<<"|-----|"<<endl;
cout<<"|Jika Membayar Cicil[Kredit]*|Rp.900,000/Bulan |"<<endl;
cout<<" +-----+"<<endl;
cout<<" +-----+"<<endl;
cout<<" "<<endl;
cout<<" *Proses Pembayaran Jika Kredit Selama 10Tahun "<<endl;
break;
default:
cout<<" "<<endl;
```

# Keterangan Kode Program

Dari Program 4.12 menggunakan data *konstan* dengan nama variabel adalah `JenPil[Jenis Pilihan]` dengan tipe integer, yaitu dengan memberikan nilai numerik misalnya nilai jenis pilihan [`JenPil=3`], maka secara otomatis Proses program akan menuju ke Switch 3.

Jika memilih pilihan 3, maka akan menjalankan statement jenis kendaraan Mobil dan akan memproses isi data yang ada pada pilihan 3 yang berisi “Jenis Kendaraan Truk”

Untuk lebih jelasnya dapat dilihat *output* berikut ini :

**Menggunakan Data Konstan Dengan Memberikan Nilai Tetap  
Untuk Jenis Pilihan [`JenPil=3`]**

```
(Inactive D:\C_LAT~1\SWITCHKO.EXE)
+-----+
+-----+
+-----Jenis Pembayaran Cicilan Kendaraan-----+
+-----+
+-----+
1. Motor
2. Mobil
3. Truk
+-----+
+-----+
|           Jenis Kendaraan Truk           |
+-----+
| Jika Membayar Tunai|      Rp.100,000,000      |
+-----+
| Jika Membayar Cicil[Kredit]*|Rp.900,000/Bulan |
+-----+
+-----+
*Proses Pembauran Jika Kredit Selama 10Tahun
```

## Contoh Program Menggunakan Logika Switch Dengan Data Konstan

```
//Program 4.13
//Program Menentukan Jenis Pembayaran Cicilan Kendaraan
#include <iostream.h>
#include <iomanip.h>
//Tampilan Menu
int main()
{
cout<<"-----" << endl;
cout<<"-----" << endl;
cout<<"-----Jenis Pembayaran Cicilan Kendaraan-----" << endl;
cout<<"-----" << endl;
cout<<"-----" << endl;
cout<<"1. Motor" << endl;
cout<<"2. Mobil" << endl;
cout<<"3. Truk" << endl;
//Tampilan Pilihan Jenis Cicilan
// mendeklarasikan variabel konstan jenis pilihan 1
int JenPil=4;
//Menggunakan Logika Switch
switch(JenPil)
{
case 1:
cout<<"-----" << endl;
cout<<"-----" << endl;
cout<<"|      Jenis Kendaraan Motor      |" << endl;
cout<<"|-----|" << endl;
```

```
cout<<"-----+"<<endl;
cout<<"-----+"<<endl;
cout<<"    Jenis Kendaraan Mobil          |"<<endl;
cout<<"|-----|"<<endl;
cout<<"|  Jika Membayar Tunai    | Rp.120,000,000  |"<<endl;
cout<<"|-----|"<<endl;
cout<<"|  Jika Membayar Cicil*   |Rp.1,000,000/Bulan|"<<endl;
cout<<"-----+"<<endl;
cout<<"-----+"<<endl;
cout<<" "<<endl;
cout<<" *Proses Pembayaran Jika Kredit Selama 10Tahun "<<endl;
break;
case 3:
cout<<"-----+"<<endl;
cout<<"-----+"<<endl;
cout<<"    Jenis Kendaraan Truk          |"<<endl;
cout<<"|-----|"<<endl;
cout<<"|Jika Membayar Tunai|   Rp.100,000,000   |"<<endl;
cout<<"|-----|"<<endl;
cout<<"|Jika Membayar Cicil[Kredit]*|Rp.900,000/Bulan |"<<endl;
cout<<"-----+"<<endl;
cout<<"-----+"<<endl;
cout<<" "<<endl;
cout<<" *Proses Pembayaran Jika Kredit Selama 10Tahun "<<endl;
break;
default:
cout<<" "<<endl;
```



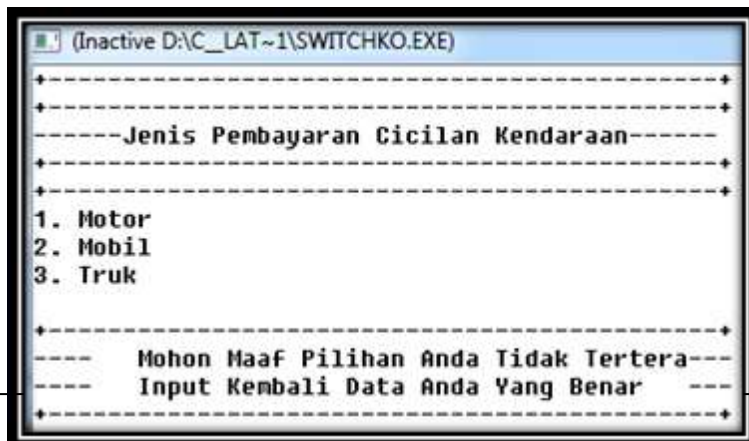
## Keterangan Kode Program

Dari Program 4.13 menggunakan data *konstan* dengan nama variabel adalah `JenPil[Jenis Pilihan]` dengan tipe integer, yaitu dengan memberikan nilai numerik misalnya nilai jenis pilihan [`JenPil=4`], maka secara otomatis Proses program akan menuju ke logika default.

Jika memilih pilihan 4, maka akan menjalankan statement yang berisi proses “Mohon Maaf Pilihan Anda Tidak Tertera Input Kembali Data Anda Yang Benar”

Untuk lebih jelasnya dapat dilihat *output* berikut ini :

**Menggunakan Data Konstan Dengan Memberikan Nilai Tetap  
Untuk Jenis Pilihan [`JenPil=4`]**



```
(Inactive D:\C_LAT~1\SWITCHKO.EXE)
-----
-----
----- Jenis Pembayaran Cicilan Kendaraan -----
-----
1. Motor
2. Mobil
3. Truk
-----
-----
----- Mohon Maaf Pilihan Anda Tidak Tertera -----
----- Input Kembali Data Anda Yang Benar -----
-----
-----
```

# Struktur Control Dengan Logika Switch

## Contoh Program Menggunakan Logika Switch Dengan Bahasa Pemrograman C# Data Input

```
//program 4.14
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace switch1
{
    class Program
    {
        static void Main(string[] args)
        {
            String a, x, y;
            Console.WriteLine("
+=====+");
            Console.WriteLine(" |Rental Mobil Wilayah Jabodetabek dan Bandung |");
            Console.WriteLine("
+=====+");
            Console.WriteLine(" Inputkan Nama Anda      : ");
```

```
Console.WriteLine(" | 3. Jakarta Selatan    |");
Console.WriteLine(" | 4. Jakarta Barat      |");
Console.WriteLine(" | 5. Bogor              |");
Console.WriteLine(" | 6. Tangerang          |");
Console.WriteLine(" | 7. Bekasi             |");
Console.WriteLine(" | 8. Lembang           |");
Console.WriteLine(" | 9. Garut              |");
Console.WriteLine(" | 10. Kota Lain di Bandung |");
Console.WriteLine(" +-----+");
String k;
double b;
Console.Write(" Inputkan Nomor Kota Tujuan : ");
k = Console.ReadLine();
Console.Write(" Inputkan Jarak Tujuan [Km] : ");
b = double.Parse(Console.ReadLine());
switch (k)
{
case "1":
Console.WriteLine(" Total Biaya : Rp " + (20000 * b) + ",-");
break;
case "2":
Console.WriteLine(" Total Biaya : Rp " + (15000 * b) + ",-");
break;
case "3":
Console.WriteLine(" Total Biaya : Rp " + (10000 * b) + ",-");
break;
case "4":
Console.WriteLine(" Total Biaya : Rp " + (20000 * b) + ",-");
```

```
case "8":
Console.WriteLine(" Total Biaya : Rp " + (25000 * b) + ",-");
break;
case "9":
Console.WriteLine(" Total Biaya : Rp " + (40000 * b) + ",-");
break;
case "10":
Console.WriteLine(" Total Biaya : Rp " + (40000 * b) + ",-");
break;
default:
Console.WriteLine(" Maaf Tidak ada pilihan");
break;
}
Console.WriteLine("
=====");
Console.WriteLine("Terima Kasih Telah Menggunakan Jasa Rental Kami ");
Console.WriteLine("
=====");
Console.ReadLine();
}
}
}
```

## Keterangan Kode Program

Dari Program 4.14 menggunakan data *input* dengan nama variabel adalah `String a, x, v` | a=variabel deklarasi nama. x= variabel alamat.

- b. Jika memilih pilihan 2, maka akan menjalankan statement jenis kota tujuan Jakarta Timur dengan tarif tiap Km adalah “Rp.15.000/km”
- c. Jika memilih pilihan 3, maka akan menjalankan statement jenis kota tujuan Jakarta Selatan dengan tarif tiap Km adalah “Rp.10.000/km”
- d. Jika memilih pilihan 4, maka akan menjalankan statement jenis kota tujuan Jakarta Barat dengan tarif tiap Km adalah “Rp.20.000/km”
- e. Jika memilih pilihan 5, maka akan menjalankan statement jenis kota tujuan Bogor dengan tarif tiap Km adalah “Rp.25.000/km”
- f. Jika memilih pilihan 6, maka akan menjalankan statement jenis kota tujuan Tangerang dengan tarif tiap Km adalah “Rp.25.000/km”
- g. Jika memilih pilihan 7, maka akan menjalankan statement jenis kota tujuan Bekasi dengan tarif tiap Km adalah “Rp.30.000/km”
- h. Jika memilih pilihan 8, maka akan menjalankan statement jenis kota tujuan Lembang dengan tarif tiap Km adalah “Rp.25.000/km”
- i. Jika memilih pilihan 9, maka akan menjalankan statement jenis kota tujuan Garut dengan tarif tiap Km adalah “Rp.40.000/km”
- j. Jika memilih pilihan 10, maka akan menjalankan statement jenis kota tujuan Bandung Lainnya dengan tarif tiap Km adalah “Rp.40.000/km”. dan jika diluar dari pilihan yang ada akan menjalankan statement “ Maaf tidak ada pilihan. Untuk lebih jelasnya dapat kita lihat output berikut ini :

### Output Jika Memilih Kota Tujuan : 1

```
C:\Windows\system32\cmd.exe
+-----+
|Rental Mobil Wilayah Jabodetabek dan Bandung |
+-----+
Inputkan Nama Anda      : Tsarwan Kasyfi
Inputkan Alamat Anda    : Kelapa Dua, Depok
Inputkan Nomor HP Anda  : 081383955321
+-----+
: Kota Tujuan Rental :
: 1. Jakarta Utara
: 2. Jakarta Timur
: 3. Jakarta Selatan
: 4. Jakarta Barat
: 5. Bogor
: 6. Tangerang
: 7. Bekasi
: 8. Lembang
: 9. Garut
: 10. Kota Lain di Bandung
+-----+
Inputkan Nomor Kota Tujuan : 1
Inputkan Jarak Tujuan [Km] : 5
Total Biaya : Rp 100000,-
+-----+
Terima Kasih Telah Menggunakan Jasa Rental Kami
+-----+
```

Gambar 4.22 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel/ Input*

### Output Jika Memilih Kota Tujuan : 2

```
C:\Windows\system32\cmd.exe
+-----+
|Rental Mobil Wilayah Jabodetabek dan Bandung |
+-----+
Inputkan Nama Anda      : Abdi Husairi Nst
Inputkan Alamat Anda    : Kelapa Dua, Depok
Inputkan Nomor HP Anda  : 081383955321
+-----+
: Kota Tujuan Rental :
: 1. Jakarta Utara
: 2. Jakarta Timur
: 3. Jakarta Selatan
: 4. Jakarta Barat
: 5. Bogor
: 6. Tangerang
: 7. Bekasi
: 8. Lembang
: 9. Garut
: 10. Kota Lain di Bandung
+-----+
Inputkan Nomor Kota Tujuan : 2
Inputkan Jarak Tujuan [Km] : 15
Total Biaya : Rp 225000,-
+-----+
Terima Kasih Telah Menggunakan Jasa Rental Kami
+-----+
```

Gambar 4.23 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel/ Input*  
226 | Aplikasi Logika & Algoritma (C++, C# dan Java)

### Output Jika Memilih Kota Tujuan : 3

```
C:\Windows\system32\cmd.exe
+-----+
|Rental Mobil Wilayah Jabodetabek dan Bandung |
+-----+
Inputkan Nama Anda      : Fauziah, MMSI
Inputkan Alamat Anda   : Condet, Balekambang
Inputkan Nomor HP Anda  : 081383955321
+-----+
| Kota Tujuan Rental :
| 1. Jakarta Utara
| 2. Jakarta Timur
| 3. Jakarta Selatan
| 4. Jakarta Barat
| 5. Bogor
| 6. Tangerang
| 7. Bekasi
| 8. Lembang
| 9. Garut
| 10. Kota Lain di Bandung
+-----+
Inputkan Nomor Kota Tujuan : 3
Inputkan Jarak Tujuan [km] : 20
Total Biaya : Rp 200000,-
+-----+
Terima Kasih Telah Menggunakan Jasa Rental Kami
+-----+
```

Gambar 4.24 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel/ Input*

### Output Jika Memilih Kota Tujuan : 4

```
C:\Windows\system32\cmd.exe
+-----+
|Rental Mobil Wilayah Jabodetabek dan Bandung |
+-----+
Inputkan Nama Anda      : Hj. Mastanah
Inputkan Alamat Anda   : Alnabruk II Condet
Inputkan Nomor HP Anda  : 081383955321
+-----+
| Kota Tujuan Rental :
| 1. Jakarta Utara
| 2. Jakarta Timur
| 3. Jakarta Selatan
| 4. Jakarta Barat
| 5. Bogor
| 6. Tangerang
| 7. Bekasi
| 8. Lembang
| 9. Garut
| 10. Kota Lain di Bandung
+-----+
Inputkan Nomor Kota Tujuan : 4
Inputkan Jarak Tujuan [km] : 5
Total Biaya : Rp 100000,-
+-----+
Terima Kasih Telah Menggunakan Jasa Rental Kami
+-----+
```

### Output Jika Memilih Kota Tujuan : 5

```
C:\Windows\system32\cmd.exe
+-----+
|Rental Mobil Wilayah Jabodetabek dan Bandung |
+-----+
Inputkan Nama Anda      : Hj.Chodidjah
Inputkan Alamat Anda   : Kalibata
Inputkan Nomor HP Anda  : 081383955321
+-----+
Kota Tujuan Rental :
1. Jakarta Utara
2. Jakarta Timur
3. Jakarta Selatan
4. Jakarta Barat
5. Bogor
6. Tangerang
7. Bekasi
8. Lembang
9. Garut
10. Kota Lain di Bandung
+-----+
Inputkan Nomor Kota Tujuan : 5
Inputkan Jarak Tujuan [Km] : 6
Total Biaya : Rp 150000,-
+-----+
Terima Kasih Telah Menggunakan Jasa Rental Kami
+-----+
```

Gambar 4.26 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel* Input

### Output Jika Memilih Kota Tujuan :6

```
C:\Windows\system32\cmd.exe
+-----+
|Rental Mobil Wilayah Jabodetabek dan Bandung |
+-----+
Inputkan Nama Anda      : Fauziah, MMSI
Inputkan Alamat Anda   : Depok
Inputkan Nomor HP Anda  : 081383955321
+-----+
Kota Tujuan Rental :
1. Jakarta Utara
2. Jakarta Timur
3. Jakarta Selatan
4. Jakarta Barat
5. Bogor
6. Tangerang
7. Bekasi
8. Lembang
9. Garut
10. Kota Lain di Bandung
+-----+
Inputkan Nomor Kota Tujuan : 6
Inputkan Jarak Tujuan [Km] : 4
Total Biaya : Rp 100000,-
+-----+
Terima Kasih Telah Menggunakan Jasa Rental Kami
+-----+
```

Gambar 4.27 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel* Input



### Output Jika Memilih Kota Tujuan : 7

```
C:\Windows\system32\cmd.exe
+=====+
|Rental Mobil Wilayah Jabodetabek dan Bandung |
+=====+
Inputkan Nama Anda      : Farhanan Taufiq
Inputkan Alamat Anda    : Ciracas
Inputkan Nomor HP Anda  : 081383955321
+-----+
: Kota Tujuan Rental :
: 1. Jakarta Utara
: 2. Jakarta Timur
: 3. Jakarta Selatan
: 4. Jakarta Barat
: 5. Bogor
: 6. Tangerang
: 7. Bekasi
: 8. Lembang
: 9. Garut
: 10. Kota Lain di Bandung
+-----+
Inputkan Nomor Kota Tujuan : 7
Inputkan Jarak Tujuan [Kml] : 8
Total Biaya : Rp 240000,-
+-----+
Terima Kasih Telah Menggunakan Jasa Rental Kami
+=====+
```

Gambar 4.25 Tampilan Output Dengan Logika Switch Menggunakan Variabel Input

### Output Jika Memilih Kota Tujuan : 8

```
C:\Windows\system32\cmd.exe
+=====+
|Rental Mobil Wilayah Jabodetabek dan Bandung |
+=====+
Inputkan Nama Anda      : Isarwan Abdi
Inputkan Alamat Anda    : Bogor
Inputkan Nomor HP Anda  : 081383955321
+-----+
: Kota Tujuan Rental :
: 1. Jakarta Utara
: 2. Jakarta Timur
: 3. Jakarta Selatan
: 4. Jakarta Barat
: 5. Bogor
: 6. Tangerang
: 7. Bekasi
: 8. Lembang
: 9. Garut
: 10. Kota Lain di Bandung
+-----+
Inputkan Nomor Kota Tujuan : 8
Inputkan Jarak Tujuan [Kml] : 12
Total Biaya : Rp 300000,-
+-----+
Terima Kasih Telah Menggunakan Jasa Rental Kami
+=====+
```

### Output Jika Memilih Kota Tujuan : 9

```
C:\Windows\system32\cmd.exe
+-----+
|Rental Mobil Wilayah Jabodetabek dan Bandung |
+-----+
Inputkan Nama Anda      : Fauziah Kasyfi
Inputkan Alamat Anda   : Petamburan
Inputkan Nomor HP Anda : 081383955321
+-----+
Kota Tujuan Rental :
1. Jakarta Utara
2. Jakarta Timur
3. Jakarta Selatan
4. Jakarta Barat
5. Bogor
6. Tangerang
7. Bekasi
8. Lembang
9. Garut
10. Kota Lain di Bandung
+-----+
Inputkan Nomor Kota Tujuan : 9
Inputkan Jarak Tujuan [Km] : 8
Total Biaya : Rp 320000,-
+-----+
Terima Kasih Telah Menggunakan Jasa Rental Kami
+-----+
```

Gambar 4.27 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel/ Input*

### Output Jika Memilih Kota Tujuan : 10

```
C:\Windows\system32\cmd.exe
+-----+
|Rental Mobil Wilayah Jabodetabek dan Bandung |
+-----+
Inputkan Nama Anda      : Fauziah Nasir
Inputkan Alamat Anda   : Setia Budi
Inputkan Nomor HP Anda : 081383955321
+-----+
Kota Tujuan Rental :
1. Jakarta Utara
2. Jakarta Timur
3. Jakarta Selatan
4. Jakarta Barat
5. Bogor
6. Tangerang
7. Bekasi
8. Lembang
9. Garut
10. Kota Lain di Bandung
+-----+
Inputkan Nomor Kota Tujuan : 10
Inputkan Jarak Tujuan [Km] : 8
Total Biaya : Rp 320000,-
+-----+
Terima Kasih Telah Menggunakan Jasa Rental Kami
+-----+
```

Gambar 4.29 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel/ Input*

Output Jika Memilih Kota Tujuan diluar dari pilihan yang disediakan  
misalnya kita input : 11, maka tampilan menjadi :



```
C:\Windows\system32\cmd.exe
+=====+
|Rental Mobil Wilayah Jabodetabek dan Bandung |
+=====+
|Inputkan Nama Anda       : Isarwan Abdi    |
|Inputkan Alamat Anda    : Tanah Abang   |
|Inputkan Nomor HP Anda  : 081383955321 |
+-----+
| Kota Tujuan Rental : |
| 1. Jakarta Utara   |
| 2. Jakarta Timur  |
| 3. Jakarta Selatan|
| 4. Jakarta Barat  |
| 5. Bogor           |
| 6. Tangerang       |
| 7. Bekasi          |
| 8. Lembang        |
| 9. Garut           |
| 10. Kota Lain di Bandung |
+-----+
|Inputkan Nomor Kota Tujuan : 11 |
|Inputkan Jarak Tujuan [Kml] : 1 |
|Maaf Tidak ada pilihan |
+=====+
|Terima Kasih Telah Menggunakan Jasa Rental Kami |
+=====+
```

Gambar 4.30 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel Input*

Keterangan Gambar :

Dari Gambar 4.19 sampai gambar 4.30 merupakan *output* yang menjelaskan tentang logika switch dengan pilihan 1-10 dan pilihan 11 yang tidak tercantum pada program, terlihat jelas masing-masing pilihan akan menjalankan proses yang telah kita *inputkan*.

# Struktur Control Dengan Logika Switch

## Contoh Program Menggunakan Logika *Switch* Dengan Bahasa Pemrograman C# Data Konstan

```
//program 4.15
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace switch1
{
    class Program
    {
        static void Main(string[] args)
        {
            String a="Kasyfi";
            String x="Kelapa Dua";
            String y = "081383955321";
            Console.WriteLine(" +=====+");
            Console.WriteLine(" |Rental Mobil Wilayah Jabodetabek dan Bandung
```

```
Console.WriteLine(" | Kota Tujuan Rental :  |");
Console.WriteLine(" | 1. Jakarta Utara      |");
Console.WriteLine(" | 2. Jakarta Timur      |");
Console.WriteLine(" | 3. Jakarta Selatan    |");
Console.WriteLine(" | 4. Jakarta Barat      |");
Console.WriteLine(" | 5. Bogor               |");
Console.WriteLine(" | 6. Tangerang           |");
Console.WriteLine(" | 7. Bekasi              |");
Console.WriteLine(" | 8. Lembang            |");
Console.WriteLine(" | 9. Garut               |");
Console.WriteLine(" | 10. Kota Lain di Bandung |");
Console.WriteLine(" +-----+");
String k= "1";
double b= 10;
Console.WriteLine(" Kota Tujuan Anda Adalah : " + k);
Console.WriteLine(" Jarak Tujuan [Km]Adalah : " + b);
switch (k)
{
    case "1":
        Console.WriteLine(" Total Biaya : Rp " + (20000 * b) + ",-");
        break;
    case "2":
        Console.WriteLine(" Total Biaya : Rp " + (15000 * b) + ",-");
        break;
    case "3":
```

```
case "6":  
Console.WriteLine(" Total Biaya : Rp " + (25000 * b) + ",-");  
break;  
case "7":  
Console.WriteLine(" Total Biaya : Rp " + (30000 * b) + ",-");  
break;  
case "8":  
Console.WriteLine(" Total Biaya : Rp " + (25000 * b) + ",-");  
break;  
case "9":  
Console.WriteLine(" Total Biaya : Rp " + (40000 * b) + ",-");  
break;  
case "10":  
Console.WriteLine(" Total Biaya : Rp " + (40000 * b) + ",-");  
break;  
default:  
Console.WriteLine(" Maaf Tidak ada pilihan");  
break;  
}  
Console.WriteLine(" =====");
```

## Keterangan Kode Program

Dari Program 4.15 menggunakan data *konstan* dengan nama *variabel* `String a="Kasyfi"; String x="Kelapa Dua";String y = "081383955321";` [Dengan `String` nama adalah =a, string alamat adalah=x dan string No. Handphone adalah y]. `String k= "1";` [string k merupakan pilihan kota tujuan dan secara otomatis akan menjalankan proses pada *switch* 1 dengan kota tujuan “Jakarta Utara”, `double b= 10` [double b merupakan jenis jarak yang ditempuh dalam satuan Km, dan secara otomatis akan dihitung berdasarkan switch 1 yaitu dengan rumus : `Total Biaya : Rp " + (20000 * b)`. Apabila `String k= "2"` ;maka secara otomatis Proses Switch 2 dengan kota tujuan “ Jakarta Timur” dan apabila jarak kita isikan secara konstan dengan nilai misalnya `double b= 10`, maka secara otomatis proses akan menghitung nilai yang kita deklarasikan secara konstan, dan begitu seterusnya sampai `String k= "10"`. Dan untuk pengecualian jika `String k= "11"` maka akan menui ke logika *default* yang berisi *statement* " `Maaf`

Output Dengan Data Konstan dan Jika Pilihan Kota yang diisikan dengan nilai kota tujuan="1",maka outputnya sebagai berikut

```
C:\Windows\system32\cmd.exe
+-----+
|Rental Mobil Wilayah Jabodetabek dan Bandung |
+-----+
>Nama Anda Adalah           : Kasyfi
>|Alamat Anda Adalah        : Kelapa Dua
>|Nomor HP Anda Adalah      : 081383955321
+-----+
>|Kota Tujuan Rental      :
>|1. Jakarta Utara
>|2. Jakarta Timur
>|3. Jakarta Selatan
>|4. Jakarta Barat
>|5. Bogor
>|6. Tangerang
>|7. Bekasi
>|8. Lembang
>|9. Garut
>|10. Kota Lain di Bandung
+-----+
>|Kota Tujuan Anda Adalah : 1
>|Jarak Tujuan [Km]Adalah : 10
>|Total Biaya : Rp 200000,-
+-----+
>|Terima Kasih Telah Menggunakan Jasa Rental Kami
+-----+
```

Gambar 4.31 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel Konstan* dengan kota tujuan="1"

Output Dengan Data Konstan dan Jika Pilihan Kota yang diisikan dengan nilai kota tujuan="2",maka outputnya sebagai berikut

```
C:\Windows\system32\cmd.exe
+-----+
|Rental Mobil Wilayah Jabodetabek dan Bandung |
+-----+
>Nama Anda Adalah           : Kasyfi
>|Alamat Anda Adalah        : Kelapa Dua
>|Nomor HP Anda Adalah      : 081383955321
+-----+
>|Kota Tujuan Rental      :
>|1. Jakarta Utara
>|2. Jakarta Timur
>|3. Jakarta Selatan
>|4. Jakarta Barat
>|5. Bogor
>|6. Tangerang
>|7. Bekasi
>|8. Lembang
>|9. Garut
>|10. Kota Lain di Bandung
+-----+
>|Kota Tujuan Anda Adalah : 2
>|Jarak Tujuan [Km]Adalah : 10
>|Total Biaya : Rp 150000,-
+-----+
>|Terima Kasih Telah Menggunakan Jasa Rental Kami
+-----+
```



Output Dengan Data Konstan dan Jika Pilihan Kota yang diisikan dengan nilai kota tujuan="3", maka outputnya sebagai berikut

```
C:\Windows\system32\cmd.exe
+-----+
|Rental Mobil Wilayah Jabodetabek dan Bandung |
+-----+
Nama Anda Adalah           : Kasyfi
Alamat Anda Adalah        : Kelapa Dua
Nomor HP Anda Adalah      : 081383955321
+-----+
Kota Tujuan Rental :
1. Jakarta Utara
2. Jakarta Timur
3. Jakarta Selatan
4. Jakarta Barat
5. Bogor
6. Tangerang
7. Bekasi
8. Lembang
9. Garut
10. Kota Lain di Bandung
+-----+
Kota Tujuan Anda Adalah : 3
Jarak Tujuan [Kml]Adalah : 10
Total Biaya : Rp 100000,-
+-----+
Terima Kasih Telah Menggunakan Jasa Rental Kami
+-----+
```

Gambar 4.33 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel*/Konstan dengan kota tujuan="3"

Output Dengan Data Konstan dan Jika Pilihan Kota yang diisikan dengan nilai kota tujuan="4", maka outputnya sebagai berikut

```
C:\Windows\system32\cmd.exe
+-----+
|Rental Mobil Wilayah Jabodetabek dan Bandung |
+-----+
Nama Anda Adalah           : Kasyfi
Alamat Anda Adalah        : Kelapa Dua
Nomor HP Anda Adalah      : 081383955321
+-----+
Kota Tujuan Rental :
1. Jakarta Utara
2. Jakarta Timur
3. Jakarta Selatan
4. Jakarta Barat
5. Bogor
6. Tangerang
7. Bekasi
8. Lembang
9. Garut
10. Kota Lain di Bandung
+-----+
Kota Tujuan Anda Adalah : 4
Jarak Tujuan [Kml]Adalah : 10
Total Biaya : Rp 200000,-
+-----+
Terima Kasih Telah Menggunakan Jasa Rental Kami
+-----+
```

Gambar 4.34 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel*/Konstan dengan kota tujuan="4"

Output Dengan Data Konstan dan Jika Pilihan Kota yang diisikan dengan nilai kota tujuan="5",maka outputnya sebagai berikut

```
C:\Windows\system32\cmd.exe
+-----+
|Rental Mobil Wilayah Jabodetabek dan Bandung |
+-----+
Nama Anda Adalah           : Kasyfi
Alamat Anda Adalah        : Kelapa Dua
Nomor HP Anda Adalah      : 081383955321
+-----+
Kota Tujuan Rental :
1. Jakarta Utara
2. Jakarta Timur
3. Jakarta Selatan
4. Jakarta Barat
5. Bogor
6. Tangerang
7. Bekasi
8. Lembang
9. Garut
10. Kota Lain di Bandung
+-----+
Kota Tujuan Anda Adalah : 5
Jarak Tujuan [Km]Adalah : 10
Total Biaya : Rp 250000,-
+-----+
Terima Kasih Telah Menggunakan Jasa Rental Kami
+-----+
```

Gambar 4.35 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel* Konstan dengan kota tujuan="5"

Output Dengan Data Konstan dan Jika Pilihan Kota yang diisikan dengan nilai kota tujuan="6",maka outputnya sebagai berikut

```
C:\Windows\system32\cmd.exe
+-----+
|Rental Mobil Wilayah Jabodetabek dan Bandung |
+-----+
Nama Anda Adalah           : Kasyfi
Alamat Anda Adalah        : Kelapa Dua
Nomor HP Anda Adalah      : 081383955321
+-----+
Kota Tujuan Rental :
1. Jakarta Utara
2. Jakarta Timur
3. Jakarta Selatan
4. Jakarta Barat
5. Bogor
6. Tangerang
7. Bekasi
8. Lembang
9. Garut
10. Kota Lain di Bandung
+-----+
Kota Tujuan Anda Adalah : 6
Jarak Tujuan [Km]Adalah : 10
Total Biaya : Rp 250000,-
+-----+
Terima Kasih Telah Menggunakan Jasa Rental Kami
+-----+
```

Gambar 4.36 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel* Konstan

Output Dengan Data Konstan dan Jika Pilihan Kota yang diisikan dengan nilai kota tujuan="7",maka outputnya sebagai berikut



```
C:\Windows\system32\cmd.exe
:Rental Mobil Wilayah Jabodetabek dan Bandung :
:-----:
Nama Anda Adalah           : Kasyfi
Alamat Anda Adalah        : Kelapa Dua
Nomor HP Anda Adalah      : 08138395321
:-----:
Kota Tujuan Rental :
1. Jakarta Utara
2. Jakarta Timur
3. Jakarta Selatan
4. Jakarta Barat
5. Bogor
6. Tangerang
7. Bekasi
8. Lembang
9. Garut
10. Kota Lain di Bandung
:-----:
Kota Tujuan Anda Adalah : 7
Jarak Tujuan [Km]Adalah : 10
Total Biaya : Rp 300000,-
:-----:
Terima Kasih Telah Menggunakan Jasa Rental Kami
:-----:
```

Gambar 4.37 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel/Konstan* dengan kota tujuan="7"

Output Dengan Data Konstan dan Jika Pilihan Kota yang diisikan dengan nilai kota tujuan="8",maka outputnya sebagai berikut



```
C:\Windows\system32\cmd.exe
:Rental Mobil Wilayah Jabodetabek dan Bandung :
:-----:
Nama Anda Adalah           : Kasyfi
Alamat Anda Adalah        : Kelapa Dua
Nomor HP Anda Adalah      : 08138395321
:-----:
Kota Tujuan Rental :
1. Jakarta Utara
2. Jakarta Timur
3. Jakarta Selatan
4. Jakarta Barat
5. Bogor
6. Tangerang
7. Bekasi
8. Lembang
9. Garut
10. Kota Lain di Bandung
:-----:
Kota Tujuan Anda Adalah : 8
Jarak Tujuan [Km]Adalah : 10
Total Biaya : Rp 250000,-
:-----:
Terima Kasih Telah Menggunakan Jasa Rental Kami
:-----:
```

Gambar 4.38 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel/Konstan* dengan kota tujuan="8"

Output Dengan Data Konstan dan Jika Pilihan Kota yang diisikan dengan nilai kota tujuan="9",maka outputnya sebagai berikut

```
C:\Windows\system32\cmd.exe
:Rental Mobil Wilayah Jabodetabek dan Bandung :
-----
Nama Anda Adalah           : Kasyfi
Alamat Anda Adalah         : Kelapa Dua
Nomor HP Anda Adalah       : 081383955321
-----
Kota Tujuan Rental :
1. Jakarta Utara
2. Jakarta Timur
3. Jakarta Selatan
4. Jakarta Barat
5. Bogor
6. Tangerang
7. Bekasi
8. Lembang
9. Garut
10. Kota Lain di Bandung
-----
Kota Tujuan Anda Adalah : 9
Jarak Tujuan [Km]Adalah : 10
Total Biaya : Rp 400000,-
-----
Terima Kasih Telah Menggunakan Jasa Rental Kani
-----
```

Gambar 4.39 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel/Konstan* dengan kota tujuan="9"

Output Dengan Data Konstan dan Jika Pilihan Kota yang diisikan dengan nilai kota tujuan="10",maka outputnya sebagai berikut

```
C:\Windows\system32\cmd.exe
:Rental Mobil Wilayah Jabodetabek dan Bandung :
-----
Nama Anda Adalah           : Kasyfi
Alamat Anda Adalah         : Kelapa Dua
Nomor HP Anda Adalah       : 081383955321
-----
Kota Tujuan Rental :
1. Jakarta Utara
2. Jakarta Timur
3. Jakarta Selatan
4. Jakarta Barat
5. Bogor
6. Tangerang
7. Bekasi
8. Lembang
9. Garut
10. Kota Lain di Bandung
-----
Kota Tujuan Anda Adalah : 10
Jarak Tujuan [Km]Adalah : 10
Total Biaya : Rp 400000,-
-----
Terima Kasih Telah Menggunakan Jasa Rental Kani
-----
```

Gambar 4.40 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel/Konstan* dengan kota tujuan="10"

Output Dengan Data Konstan dan Jika Pilihan Kota yang diisikan dengan nilai kota tujuan="11", untuk pilihan 11 tidak tertera dalam program, maka akan menjalankan proses default, maka outnya sebagai berikut



```
C:\Windows\system32\cmd.exe
+-----+
|Rental Mobil Wilayah Jabodetabek dan Bandung |
+-----+
Nama Anda Adalah           : Kasyfi
Alamat Anda Adalah        : Kelapa Dua
Nomor HP Anda Adalah      : 081383955321
+-----+
: Kota Tujuan Rental :
: 1. Jakarta Utara
: 2. Jakarta Timur
: 3. Jakarta Selatan
: 4. Jakarta Barat
: 5. Bogor
: 6. Tangerang
: 7. Bekasi
: 8. Lembang
: 9. Garut
: 10. Kota Lain di Bandung
+-----+
Kota Tujuan Anda Adalah : 11
Jarak Tujuan [Km] adalah : 10
Maaf Tidak ada pilihan
+-----+
Terima Kasih Telah Menggunakan Jasa Rental Kari
+-----+
```

Gambar 4.40 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel/Konstan* dengan kota tujuan="11"[tidak tersedia pada pilihan kondisi]

**Keterangan Gambar :**

Dari Gambar 4.30 sampai gambar 4.40 merupakan *output* yang menjelaskan tentang logika switch dengan pilihan 1-10 dan pilihan 11 yang tidak tercantum pada program, terlihat jelas masing-masing pilihan akan menjalankan proses yang telah kita berikan nilainya secara konstan, terlihat dari nama, alamat dan

# Struktur Control Dengan Logika Switch

## Contoh Program Menggunakan Logika *Switch* Dengan Bahasa Pemrograman Java Data Konstan

//program 4.16

```
class switchko{
    public static void main(String[] args) {
        int angka = 5;
        switch (angka) {
            case 1:
                System.out.println("Huruf Abjad ke-" + angka + " adalah A");
                break;
            case 2:
                System.out.println("Huruf Abjad ke-" + angka + " adalah B");
                break;
            case 3:
                System.out.println("Huruf Abjad ke-" + angka + " adalah C");
                break;
            case 4:
                System.out.println("Huruf Abjad ke-" + angka + " adalah D");
                break;
            case 5:
```

case 7:

```
System.out.println("Huruf Abjad ke-" + angka + " adalah G");
```

```
break;
```

```
default:
```

```
System.out.println("Tidak ada Huruf Abjad Yang Sesuai" + huruf);
```

```
}
```

```
}
```

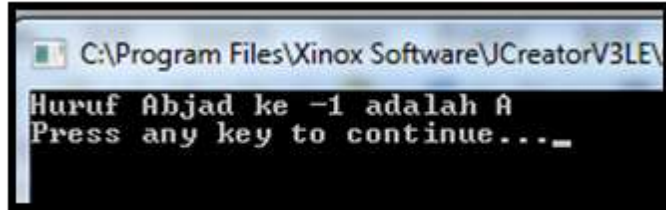
```
}
```

## Keterangan Kode Program

Dari program 4.16 terlihat program menggunakan *variabel* konstan untuk mencetak tampilan huruf abjad A sampai G dan data konstan yang telah diberikan nilainya adalah pada bagian `int angka = 5;` yang artinya program akan menuju proses *switch* 5, yaitu mencetak abjad "E", dan jika kita memberikan angka=1, maka akan mencetak abjad "A" begitu seterusnya, namun untuk contoh program yang kita gunakan sampai angka=7; yaitu mencetak huruf abjad "G" dan jika tidak ada pada pilihan *switch*, misalnya kita menginputkan angka=8, maka proses akan menuju ke pilihan *default* yaitu berisi pesan " Tidak Ada Huruf Abjad

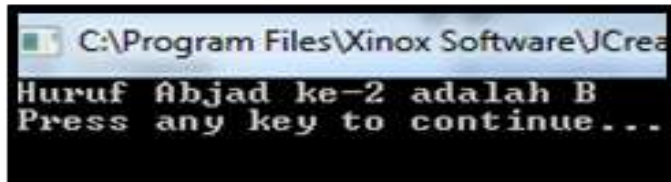


### Output jika nilai konstan pada int angka=1



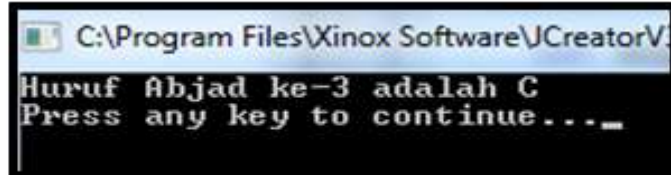
Gambar 4.41 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel/Konstan* dengan int angka=1

### Output jika nilai konstan pada int angka=2



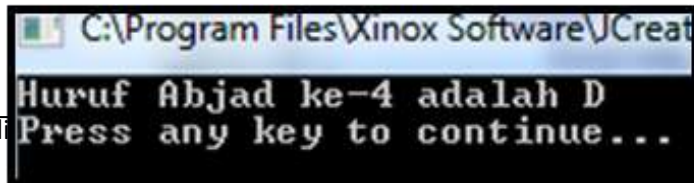
Gambar 4.42 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel/Konstan* dengan int angka=2

### Output jika nilai konstan pada int angka=3




Gambar 4.43 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel/Konstan* dengan int angka=3

### Output jika nilai konstan pada int angka=4





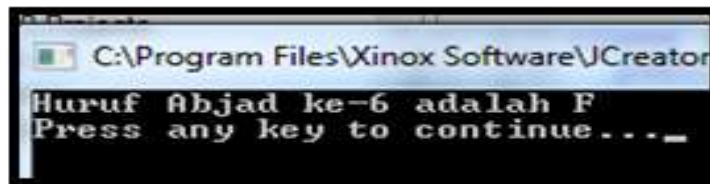
### Output jika nilai konstan pada int angka=5



```
C:\Program Files\Xinox Software\CreatorV3
Huruf Abjad ke-5 adalah E
Press any key to continue..._
```

Gambar 4.45 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel Konstan* dengan int angka=5

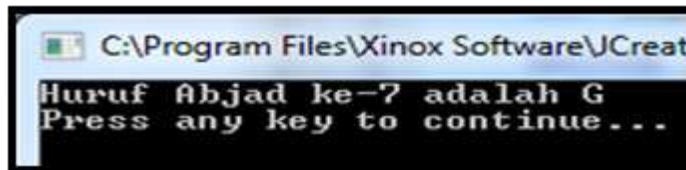
### Output jika nilai konstan pada int angka=6



```
C:\Program Files\Xinox Software\Creator
Huruf Abjad ke-6 adalah F
Press any key to continue..._
```

Gambar 4.46 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel Konstan* dengan int angka=6

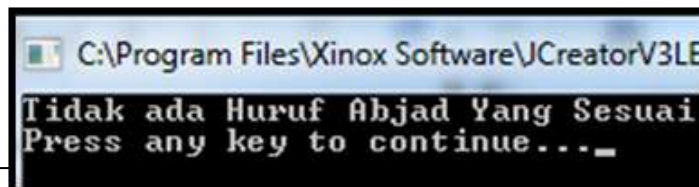
### Output jika nilai konstan pada int angka=7



```
C:\Program Files\Xinox Software\Creat
Huruf Abjad ke-7 adalah G
Press any key to continue..._
```

Gambar 4.47 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel Konstan* dengan int angka=7

### Output jika nilai konstan pada int angka=8



```
C:\Program Files\Xinox Software\CreatorV3LE
Tidak ada Huruf Abjad Yang Sesuai
Press any key to continue..._
```

## Keterangan Gambar

Dari gambar 4.41 sampai dengan gambar 4.48 terlihat jelas output yang dihasilkan dari masing – masing nilai yang sudah kita berikan dengan nilai *konstan* yaitu nilai 1 sampai dengan 8, hal ini membuktikan bahwa proses *logika switch* akan berjalan sesuai dengan instruksi atau perintah yang kita berikan.

## Struktur Control Dengan Logika Switch

### Contoh Program Menggunakan Logika *Switch* Dengan Bahasa Pemrograman Java Data *Inputan*

```
//program 4.16
import java.util.*;
public class SWITCH22
{
public static void main(String args[])
{
while(true){
System.out.println("=====");
System.out.println("  Menu Bank Untung Selalu :");
```

```
System.out.println("[4]. Proses Kliring");
System.out.println("[5]. Proses Lainnya");
System.out.println("[6]. Keluar");
System.out.print(" Inputkan Pilihan anda : ");
String input = System.console().readLine();
int nilai = Integer.parseInt(input);
System.out.println("-----");
switch(nilai)
{
case 1 : System.out.println("Anda memilih Proses Cek Saldo");
System.out.print("Inputkan Nama Anda      :");
String input1 = System.console().readLine();
System.out.print("Inputkan Alamat Anda      :");
String input2 = System.console().readLine();
System.out.print("Inputkan Nomor Rekening Anda:");
String input3 = System.console().readLine();
System.out.println("Terima Kasih Telah Menggunakan Layanan
Kami");
System.out.println("=====
=====");
break;
case 2 : System.out.println("Anda memilih Proses Menabung");
System.out.print("Inputkan Nama Anda      :");
String input4 = System.console().readLine();
System.out.print("Inputkan Alamat Anda      :");
```

```
break;
case 3 : System.out.println("Anda memilih Proses Transfer");
System.out.print("Inputkan Nama Anda      :");
String input7 = System.console().readLine();
System.out.print("Inputkan Alamat Anda    :");
String input8 = System.console().readLine();
System.out.print("Inputkan Nomor Rekening Anda:");
String input9 = System.console().readLine();
System.out.println("Terima Kasih Telah Menggunakan Layanan
Kami");
System.out.println("=====
=");
break;
case 4 : System.out.println("Anda memilih Proses Kliring");
System.out.print("Inputkan Nama Anda      :");
String input10 = System.console().readLine();
System.out.print("Inputkan Alamat Anda    :");
String input11 = System.console().readLine();
System.out.print("Inputkan Nomor Rekening Anda:");
String input12 = System.console().readLine();
System.out.println("Terima Kasih Telah Menggunakan Layanan Kami");
System.out.println("=====
=");
break;
```

```
case 6 : System.out.println("Anda memilih keluar dari Sistem,  
Terima Kasih Banyak");  
System.exit(0);  
break;  
default: System.out.println("Mohon Maaf Nilai Yang Anda  
Inputkan Salah, Coba Lagi");  
break;  
}  
}  
}  
}
```

## Keterangan Kode Program

Dari Program 4.16 Dapat dijelaskan sebagai berikut :

- Menggunakan Logika *switch*, yaitu dengan melakukan proses seleksi dengan beberapa pilihan yaitu, pilihan 1 sampai dengan 6. Untuk Pilihan [1]:Proses Cek Saldo, [2] Proses Menabung, [3] Proses Transfer,[4] Proses Kliring, [5]:Proses Lainnya dan [6]: Keluar dari Sistem. Namun Jika nilai yang diinputkan adalah nilai yang tidak terdapat pada pilihan misalnya nilai [7] maka akan tercetak pesan dari default Yaitu :”Mohon Maaf Nilai Yang Anda Inputkan Salah, Coba Lagi”].
- Untuk jenis data dan variable yang kita gunakan adalah:  
Jenis Data : *String*, merupakan jenis data yang digunakan untuk menyimpan nilai karakter atau gabungan dari beberapa karakter

### Output Jika Menginputkan Pilihan [1,2, dan 3]

```
C:\Program Files\Xinox Software\JCreatorV3LE\GE2001.exe
=====
Menu Bank Untung Selalu :
=====
[1]. Proses Cek Saldo
[2]. Proses Menabung
[3]. Proses Transfer
[4]. Proses Kliring
[5]. Proses Lainnya
[6]. Keluar
Inputkan Pilihan anda : 1
-----
Anda memilih Proses Cek Saldo
Inputkan Nama Anda           :Fauziah, S.Kom, MMSI
Inputkan Alamat Anda         :Kelapa Dua Depok
Inputkan Nomor Rekening Anda:08080808
Terima Kasih Telah Menggunakan Layanan Kami
=====
Menu Bank Untung Selalu :
=====
[1]. Proses Cek Saldo
[2]. Proses Menabung
[3]. Proses Transfer
[4]. Proses Kliring
[5]. Proses Lainnya
[6]. Keluar
Inputkan Pilihan anda : 2
-----
Anda memilih Proses Menabung
Inputkan Nama Anda           :Fauziah, S.Kom, MMSI
Inputkan Alamat Anda         :Kelapa Dua, Depok
Inputkan Nomor Rekening Anda:08080808
Terima Kasih Telah Menggunakan Layanan Kami
=====
Menu Bank Untung Selalu :
=====
[1]. Proses Cek Saldo
[2]. Proses Menabung
[3]. Proses Transfer
[4]. Proses Kliring
[5]. Proses Lainnya
[6]. Keluar
Inputkan Pilihan anda : 3
-----
Anda memilih Proses Transfer
Inputkan Nama Anda           :Fauziah, S.Kom, MMSI
Inputkan Alamat Anda         :Kelapa Dua, Depok
Inputkan Nomor Rekening Anda:08080808
Terima Kasih Telah Menggunakan Layanan Kami
=====
```

Gambar 4.49 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel* Input

### Output Jika Menginputkan Pilihan [4,5,6 dan 7]

```
C:\Program Files\Xinox Software\CreatorV3LE\GE2001.exe
=====
Menu Bank Untung Selalu :
=====
[1]. Proses Cek Saldo
[2]. Proses Menabung
[3]. Proses Transfer
[4]. Proses Kliring
[5]. Proses Lainnya
[6]. Keluar
Inputkan Pilihan anda : 4
-----
Anda memilih Proses Kliring
Inputkan Mana Anda      :Fauziah, S.Kom, MMSI
Inputkan Alamat Anda    :Kelapa Dua, Depok
Inputkan Nomor Rekening Anda:08080808
Terima Kasih Telah Menggunakan Layanan Kami
=====
Menu Bank Untung Selalu :
=====
[1]. Proses Cek Saldo
[2]. Proses Menabung
[3]. Proses Transfer
[4]. Proses Kliring
[5]. Proses Lainnya
[6]. Keluar
Inputkan Pilihan anda : 5
-----
Anda memilih Proses Lainnya
Inputkan Mana Anda      :Fauziah, S.Kom, MMSI
Inputkan Alamat Anda    :Kelapa Dua, Depok
Inputkan Nomor Rekening Anda:08080808
Terima Kasih Telah Menggunakan Layanan Kami
=====
Menu Bank Untung Selalu :
=====
[1]. Proses Cek Saldo
[2]. Proses Menabung
[3]. Proses Transfer
[4]. Proses Kliring
[5]. Proses Lainnya
[6]. Keluar
Inputkan Pilihan anda : 6
-----
Anda memilih keluar dari Sisten. Terima Kasih Banyak
Press any key to continue..._
```

Gambar 4.50 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel/ Input*

```
C:\Program Files\Xinox Software\CreatorV3LE\GE2001.exe
=====
Menu Bank Untung Selalu :
=====
[1]. Proses Cek Saldo
[2]. Proses Menabung
[3]. Proses Transfer
[4]. Proses Kliring
[5]. Proses Lainnya
[6]. Keluar
Inputkan Pilihan anda : 7
-----
Mohon Maaf Nilai Yang Anda Inputkan Salah. Coba Lagi
=====
```

21 Gambar 4.51 Tampilan *Output* Dengan *Logika Switch* Menggunakan *Variabel/ Input*

## STRUKTUR CONTROL LOGIKA SWITCH

Struktur *control* perulangan pada bahasa pemrograman digunakan untuk mengendalikan proses mengulang suatu nilai sesuai dengan batasan atau kondisi yang telah ditentukan. Struktur *control* perulangan yang sering kita gunakan adalah :

- a. Logika *For*
- b. Logika *While*

### a. Logika Perulangan dengan *For*

Merupakan jenis struktur *control* yang digunakan untuk mengulang suatu proses sesuai dengan harga awal, kondisi dan peubah yang digunakan pada suatu keadaan. Tujuan dari logika perulangan *for* adalah:

1. Mempersingkat penulisan *statement* secara berulang-ulang misalnya kita akan mencetak nilai 1-100, cukup kita tuliskan logika *for* saja, tanpa harus mengetikkannya berulang kali.
2. Menghemat penyimpanan data pada memori, karena

```
For(typedata_hargaawal;Kondisi;Peubah)
```

```
Statement
```

akan menjadi lebih hemat.

### Bentuk Umum Logika Perulangan For



## Keterangan Bentuk Umum For

Dari bentuk umum yang telah dibuat dapat dijelaskan sebagai berikut

:

1. Jenis Data : Menunjukkan jenis data yang digunakan
2. Nama *Variabel* : Nama variable yang kita deklarasikan
3. Kondisi : merupakan batasan perulangan yang kita deklarasikan
4. Peubah: menentukan nilai peubah naik atau peubah turun, untuk peubah naik biasanya kita menggunakan statement ++ [misal  $i++$  artinya nilai  $i=i+1$ ]/ sering dikenal dengan istilah *increment* dan peubah turun biasanya kita menggunakan -- [misal  $i--$  artinya nilai  $i=i-1$ ] sering dikenal dengan istilah *decrement*.

Misalnya contoh logika *for* dideklarasikan dengan nilai sebagai berikut :

### Contoh1

`For(int i=0;i<=5;i++)`, dari statement tersebut dapat dijelaskan :

- Harga awal  $i=0$ , Perulangan dengan kondisi  $i<=5$  artinya perulangan akan terus melakukan *looping* dengan kondisi  $i<=5$ , misalnya [0,1,2,3,4,5], Kemudian nilai  $i++$ , yaitu nilai  $i$  bertambah 1 dan akan berhenti jika nilai batas perulangan sudah mencapai nilai 5.

### Contoh2

`For(int i=0;i<=5;i=i+2)`, dari *statement* tersebut dapat dijelaskan

:

Untuk lebih jelasnya dapat dilihat contoh program dengan logika *For* menggunakan Bahasa C++

## Contoh Program Logika For Dengan C++

//program 4.17

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
void main()
```

```
{
```

```
//menampilan angka 1 s.d 10 dengan for
```

```
cout<<"Menampilkan angka 1 s.d 10 dengan FOR"<<endl;
```

```
for (int i=1;i<=10;i++)
```

```
{
```

```
cout<<i<<endl;
```

```
}
```

```
}
```

## Keterangan Kode Program

Dari Program 4.17 merupakan *statement* yang berisi untuk menampilkan angka 1 sampai 10, dengan harga awal  $i=1$ ; kondisi  $i$  akan terus dicetak jika masih memenuhi kondisi  $i \leq 10$  dan nilai  $i$  akan bertambah 1, sampai kondisi  $i$  terpenuhi.

Untuk lebih jelasnya dapat kita lihat tampilan *output* sebagai berikut ini :



```
(Inactive) DAC_LAT-1\FOR1-10.EXE
Menampilkan 1 s.d 10 dengan FOR
1
2
3
4
5
6
7
8
9
10
```

Contoh Logika *For* yang lainnya dapat dilihat melalui program berikut ini

## Contoh Program Logika For Dengan C++

//program 4.18

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
void main()
```

```
{
```

```
//menampilkan angka 1 s.d 9 dengan for Nilai Bertambah 2
```

```
cout<<"Menampilkan angka 1 s.d 9 dengan FOR Nilai Bertambah  
2"<<endl;
```

```
for (int i=1;i<=10;i=i+2)
```

```
{
```

```
cout<<i<<";
```

```
}
```

```
}
```

## Keterangan Kode Program

Dari Program 4.18 merupakan *statement* yang berisi untuk menampilkan angka 1 sampai 10, dengan harga awal  $i=1$ ; kondisi  $i$  akan terus dicetak jika masih memenuhi kondisi  $i \leq 10$  dan nilai  $i$  akan bertambah 2, sampai kondisi  $i$  terpenuhi. Dan dari *output* akan terlihat nilai  $i$  hanya sampai 9 karena jika  $i$  bertambah lagi maka hasil  $i$  akan menjadi 11, sedangkan batasan kondisi yang kita deklarasikan hanya  $i \leq 10$ .

Untuk lebih jelasnva dapat kita lihat tampilan *output* sebaaai berikut

Menggunakan Kombinasi beberapa Logika *For* Pada Program

## Contoh Program Logika For Dengan C++

//program 4.19

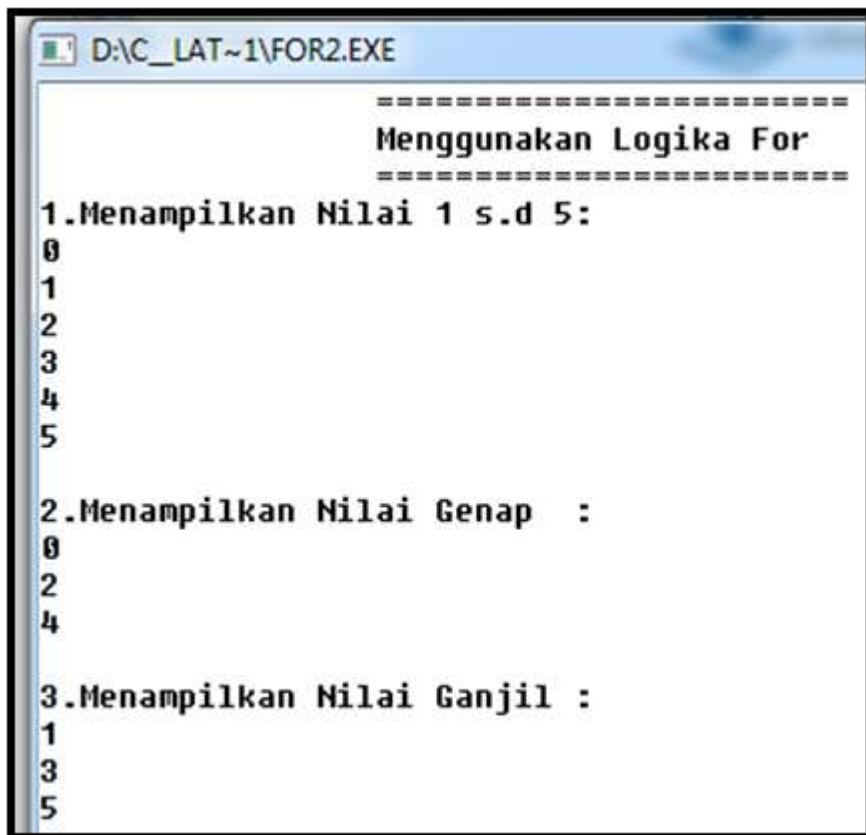
```
#include<iostream.h>
#include<conio.h>
void main()
{
    int x;
    cout<<"\t\t ====="<<endl;
    cout<<"\t\t Menggunakan Logika For "<<endl;
    cout<<"\t\t ====="<<endl;
    cout<<"1.Menampilkan Nilai 1 s.d 5: "<<endl;
    for(x=0;x<=5;x++)
    {
        cout<<x;
        cout<<"\n";
    }
    cout<<endl;
    cout<<"2.Menampilkan Nilai Genap : "<<endl;
    for(x=0;x<=5;x++)
    {
        x%2==0;
        cout<<x;
        x++;
        cout<<"\n";
    }
    cout<<endl;
    cout<<"3.Menampilkan Nilai Ganjil : "<<endl;
```

## Keterangan Kode Program

Dari Program 4.19 merupakan kombinasi logika *for*, yaitu untuk menampilkan nilai 1 sampai 5, nilai ganjil dan nilai genap, dengan batasan kondisi yang telah ditentukan.

1. Untuk Logika *For 1* : `for(x=0;x<=5;x++)`, artinya nilai harga awal yang digunakan adalah  $x=0$ , batasan kondisinya adalah nilai  $x$  akan dicetak sampai dengan 5, karena masih memenuhi kondisi  $[x<=5]$  dan nilai  $x$  terus bertambah  $[x++]$  artinya nilai  $x=x+1$ . Nilai yang tercetak  $[0,1,2,3,4,5]$
2. Untuk Logika *For 2* : `for(x=0;x<=5;x++)`. Perulangan akan terus melakukan proses sampai kondisi  $x<=5$ , namun ada *statement* tambahan yaitu: `"x%2==0 "` dari *statement* tersebut dapat dijelaskan menggunakan logika *modulo* yaitu apakah nilai  $x$  jika dibagi 2 menghasilkan 0, maka akan mencetak bilangan genap, misalnya :  $0 \bmod 2=0$ , maka 0 merupakan nilai genap  
1.  $1 \bmod 2 = 1$ , maka 1 bukan bilangan genap.  $2 \bmod 2=0$ , maka 2 adalah bilangan genap.  $3 \bmod 2=1$ , maka 3 bukan bilangan genap.  $4 \bmod 2 = 0$ , maka 4 adalah bilangan genap.  $5 \bmod 2 =1$ , maka 5 bukan bilangan genap. Namun nilai 5 tidak dicetak pada *output*, karena 5 bukan bilangan genap, dan tidak masuk dalam logika  $[x\%2==0]$ . Nilai yang tercetak  $[0,2,4]$
3. Untuk Logika *For3* : `for(x=0;x<=5;x++)`. Perulangan akan terus melakukan proses sampai kondisi  $x<=5$ , namun ada *statement* tambahan yaitu: `"x%2!=0 "` dari *statement* tersebut dapat

Nilai 0  $\text{mod } 2 = 0$ , artinya nilai 0 bukan bilangan ganjil, nilai 1  $\text{mod } 2 = 1$ , artinya nilai 1 tidak sama dengan 0, maka nilai 1 adalah bilangan ganjil, 2  $\text{mod } 2 = 0$ , artinya nilai 0 sama dengan 0, maka 2 bukan bilangan ganjil, 3  $\text{mod } 2 = 1$ , artinya nilai 1 tidak sama dengan 0, maka nilai 3 adalah bilangan ganjil. 4  $\text{mod } 2 = 0$ , artinya nilai 0 sama dengan 0, maka 4 bukan bilangan ganjil. 5  $\text{mod } 2 = 1$ , artinya nilai 5 adalah bilangan ganjil. Maka hasil yang tercetak adalah :[1,3,5]. Untuk lebih jelasnya maka output yang tercetak seperti terlihat berikut ini :



```
D:\C_LAT~1\FOR2.EXE
=====
Menggunakan Logika For
=====
1.Menampilkan Nilai 1 s.d 5:
0
1
2
3
4
5

2.Menampilkan Nilai Genap :
0
2
4

3.Menampilkan Nilai Ganjil :
1
3
5
```

Gambar 4.54 Tampilan *Output* Dengan *Logika For* Untuk Menampilkan nilai 0-5. Nilai Genap

# Contoh Logika For dengan Decreament

Berikut ini adalah contoh logika for *decrement* yaitu nilai perulangan turun. Untuk lebih jelasnya dapat dilihat pada contoh program 4.20

//Program 4.20

```
#include<iostream.h>
```

```
#include<conio.h>
```

```
void main()
```

```
{
```

```
int x;
```

```
cout<<"\t ====="<<endl;
```

```
cout<<"\t Menggunakan Logika For Decreament "<<endl;
```

```
cout<<"\t ====="<<endl;
```

```
cout<<"1.Menampilkan Nilai 10 s.d 1: "<<endl;
```

```
for(x=10;x>=1;x--)
```

```
{
```

```
cout<<x;
```

```
cout<<"\n";
```

```
}
```

```
int y;
```

```
cout<<endl;
```

```
cout<<"2.Menampilkan Nilai Genap 20 s.d 2 : "<<endl;
```

```
for(y=20;y>=2;y--)
```

```
{
```

```
y%2==0;
```

```
cout<<"3.Menampilkan Nilai Ganjil 20 s.d 1 : "<<endl;
for(z=20;z>=1;z--)
{
z%2!=0;
cout<<z-1;
z--;
cout<<"\n";;
}
getch();
}
```

## Keterangan Kode Program

Dari Program 4.20 merupakan kombinasi logika *for*, yaitu untuk menampilkan nilai 10 sampai 1, nilai ganjil dan nilai genap, dengan batasan kondisi yang telah ditentukan.

1. Untuk Logika *For 1* : `for(x=10;x>=1;x++)`, artinya nilai harga awal yang digunakan adalah  $x=10$ , batasan kondisinya adalah nilai  $x$  akan jika kondisi  $x>=1$ , nilai  $x$  terus berkurang [ $x--$  artinya nilai  $x=x-1$ ]. Nilai yang tercetak [10,9,8,7,6,5,4,3,2,1]
2. Untuk Logika *For 2* : `for(y=20;y>=2;y--)`. Perulangan akan terus melakukan proses dengan kondisi  $y>=2$ , namun ada *statement* tambahan yaitu: "`y%2==0`" dari *statement* tersebut dapat dijelaskan menggunakan logika *modulo* yaitu apakah nilai  $y$  jika dibagi 2 menghasilkan 0, maka akan mencetak bilangan genap, misalnya :  $20 \bmod 2=0$ , maka 20 merupakan nilai genap dan seterusnya sampai kondisi  $y>=2$ . Nilai  $y$  akan terus



dari *statement* tersebut dapat dijelaskan menggunakan logika *modulo* yaitu apakah nilai  $z$  jika dibagi 2 hasilnya adalah tidak sama dengan 0 [ $\neq 0$ ] dibaca [tidak sama dengan nol]. Jika kondisi tersebut terpenuhi, maka nilai akan tercetak pada *output* dan jika tidak terpenuhi, maka hasil tidak tercetak pada *output*. Misalnya saja  $:20 \bmod 2 = 0$ , artinya 20 bukan bilangan ganjil.  $19 \bmod 2 = 1$  artinya hasil modulo adalah  $\neq 0$ , maka 19 adalah bilangan ganjil dan seterusnya.

Untuk lebih jelasnya dapat dijelaskan melalui *output* berikut:



```
g++ D:\C++_LAT-1\DECERE
=====
Menggunakan Logika For Decrement
=====
1. Menampilkan Nilai 10 s.d 1:
10
9
8
7
6
5
4
3
2
1

2. Menampilkan Nilai Genap 20 s.d 2 :
20
18
16
14
12
10
8
6
4

3. Menampilkan Nilai Ganjil 20 s.d 1 :
19
17
15
13
11
9
7
5
3
1
```

**b. Logika Perulangan dengan *While***

Merupakan jenis struktur *control* yang digunakan untuk mengulang suatu proses sesuai dengan kondisi yang telah ditentukan nilainya.

**Bentuk Umum Logika Perulangan *While***

```
While (Condition)
{
    Statement;
}
```

Dari bentuk umum Logika *While* dapat dijelaskan sebagai berikut :

Ketika Kondisi Masih terpenuhi, maka perulangan akan terus melakukan proses pencetakan *statement*.

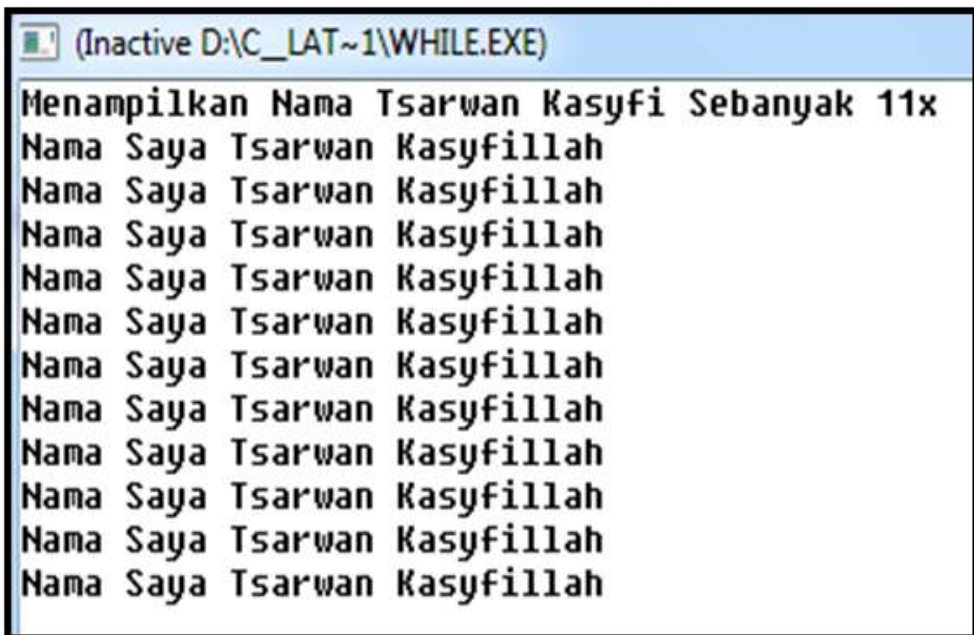
Untuk lebih jelasnya dapat dilihat pada contoh program berikut ini :

**//program 4.21**

```
#include <iostream.h>
#include <conio.h>
void main()
{
//menampilan Output Nama Tsarwan Kasyfillah Sebanyak 11x
int n;
// harga awal n=0
n=0;
// Mendeklarasikan Logika While
```

## Keterangan Kode Program

Dari Program 4.21 merupakan contoh program yang menjelaskan proses pencetakan nama sebanyak 11 x dengan *output*:" Nama Saya Tsarwan Kasyfillah". Untuk *variable* yang digunakan adalah *n* [ merupakan *variable* untuk menentukan banyaknya perulangan dengan harga awal=0], kemudian untuk logika *while* menggunakan *instruksi/statement While*( $n \leq 10$ ), yang artinya nilai *n* akan terus tercetak jika masih memenuhi kondisi yang ada yaitu  $n \leq 10$ . Dan nilai *n* akan terus bertambah 1 ( $n++$  artinya :  $n=n+1$ ). Maka *output* yang dapat ditampilkan adalah:



```
(Inactive D:\C_LAT~1\WHILE.EXE)
Menampilkan Nama Tsarwan Kasyfi Sebanyak 11x
Nama Saya Tsarwan Kasyfillah
Nama Saya Tsarwan Kasyfillah
Nama Saya Tsarwan Kasyfillah
Nama Saya Tsarwan Kasyfillah
Nama Saya Tsarwan Kasyfillah
Nama Saya Tsarwan Kasyfillah
Nama Saya Tsarwan Kasyfillah
Nama Saya Tsarwan Kasyfillah
Nama Saya Tsarwan Kasyfillah
Nama Saya Tsarwan Kasyfillah
Nama Saya Tsarwan Kasyfillah
```

Gambar 4.56 Tampilan *Output* Dengan *Logika For* Untuk Menampilkan nilai 0-5, Nilai Genap dan Nilai Ganjil Dengan Kondisi  $x \leq 5$

## Contoh Program Logika For Dengan C#

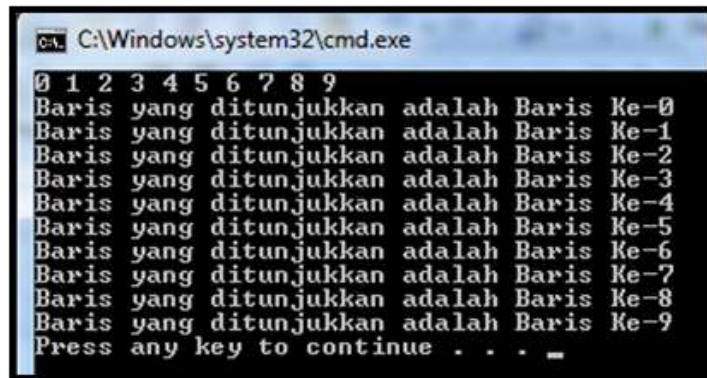
//program 4.22

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace for1
{
    class Program
    {
        static void Main(string[] args)
        {
            //Logika For dengan harga awal =0
            for (int i = 0; i < 10; i++)
            {
                //mencetak nilai 0 sampai 9
                Console.Write(i + " ");
            }
            Console.WriteLine("");
            for (int i = 0; i < 10; i++)
            {
                //mencetak 0 1 sampai 9
                Console.WriteLine("Baris yang ditunjukkan adalah Baris Ke-" + i);
            }
        }
    }
}
```

## Keterangan Kode Program

Pada Program 4.22 merupakan contoh program dengan logika perulangan *for increment* dengan harga awal  $i=0$  dan batasan perulangan  $i<10$ , artinya nilai  $i$  akan terus berulang jika nilai  $i$  mencapai nilai 9,  $i$  terus bertambah dengan  $i=i++$ .



```
C:\Windows\system32\cmd.exe
0 1 2 3 4 5 6 7 8 9
Baris yang ditunjukkan adalah Baris Ke-0
Baris yang ditunjukkan adalah Baris Ke-1
Baris yang ditunjukkan adalah Baris Ke-2
Baris yang ditunjukkan adalah Baris Ke-3
Baris yang ditunjukkan adalah Baris Ke-4
Baris yang ditunjukkan adalah Baris Ke-5
Baris yang ditunjukkan adalah Baris Ke-6
Baris yang ditunjukkan adalah Baris Ke-7
Baris yang ditunjukkan adalah Baris Ke-8
Baris yang ditunjukkan adalah Baris Ke-9
Press any key to continue . . . =
```

Gambar 4.57 Tampilan *Output* Dengan Logika *For* Untuk Menampilkan nilai 0-9

## Contoh Program Logika For Dengan C#

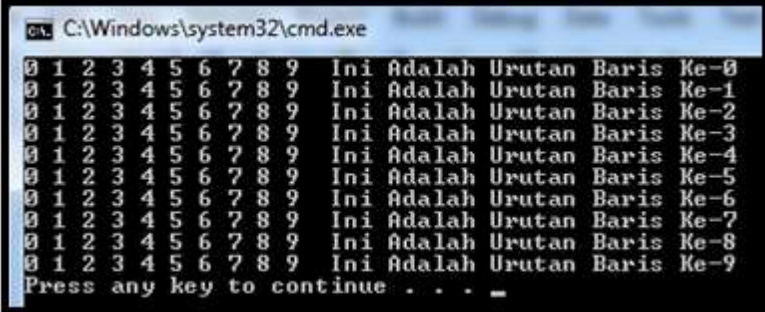
### //program 4.23

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace for2
{
    class Program
    {
        static void Main(string[] args)
        {
            // mendeklarasikan baris dengan harga awal=0
            for (int i = 0; i < 10; i++)
            // mendeklarasikan kolom dengan harga awal=0
            {
                for (int j = 0; j < 10; j++)
                {
                    // mencetak kolom
                    Console.Write(j + " ");
                }
                // mencetak hasil dengan tulisan barisan ke-
                Console.WriteLine(" Ini Adalah Urutan Baris
                Ke-" + i);
            }
        }
    }
}
```

## Keterangan Kode Program

Pada Program 4.23 merupakan contoh program dengan logika perulangan *for increment* dengan harga awal  $i=0$  dan batasan perulangan  $i<10$ , artinya nilai  $i$  akan terus berulang jika nilai  $i$  mencapai nilai 9,  $i$  terus bertambah dengan  $i=i++$ .

Untuk lebih jelasnya dapat dilihat pada output berikut ini :



```
C:\Windows\system32\cmd.exe
0 1 2 3 4 5 6 7 8 9 Ini Adalah Urutan Baris Ke-0
0 1 2 3 4 5 6 7 8 9 Ini Adalah Urutan Baris Ke-1
0 1 2 3 4 5 6 7 8 9 Ini Adalah Urutan Baris Ke-2
0 1 2 3 4 5 6 7 8 9 Ini Adalah Urutan Baris Ke-3
0 1 2 3 4 5 6 7 8 9 Ini Adalah Urutan Baris Ke-4
0 1 2 3 4 5 6 7 8 9 Ini Adalah Urutan Baris Ke-5
0 1 2 3 4 5 6 7 8 9 Ini Adalah Urutan Baris Ke-6
0 1 2 3 4 5 6 7 8 9 Ini Adalah Urutan Baris Ke-7
0 1 2 3 4 5 6 7 8 9 Ini Adalah Urutan Baris Ke-8
0 1 2 3 4 5 6 7 8 9 Ini Adalah Urutan Baris Ke-9
Press any key to continue . . . -
```

Gambar 4.58 Tampilan *Output* Dengan *Logika For* Untuk Menampilkan nilai 0-9

## Contoh Program Logika For Dengan C#

//program 4.24

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

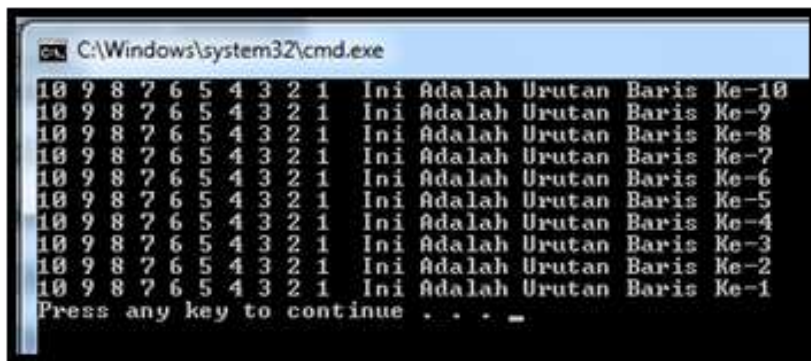
namespace for21
{
    class Program
    {
        static void Main(string[] args)
        {
            // mendeklarasikan baris dengan harga awal=10
            for (int i = 10; i >=1; i--)
            // mendeklarasikan kolom dengan harga awal=10
```

```
{  
// mencetak kolom  
Console.Write(j + " ");  
}  
// mencetak hasil dengan tulisan barisan ke-  
Console.WriteLine(" Ini Adalah Urutan Baris Ke-" + i);  
    }  
}  
}
```

## Keterangan Kode Program

Pada Program 4.24 merupakan contoh program dengan logika perulangan *for Decreament* dengan harga awal  $i=10$  dan batasan perulangan  $i \geq 1$ , artinya nilai  $i$  akan terus berulang jika nilai  $i$  mencapai nilai 1,  $i$  terus berkurang dengan  $i=i--$ .

Untuk lebih jelasnya dapat dilihat pada *output* berikut ini :



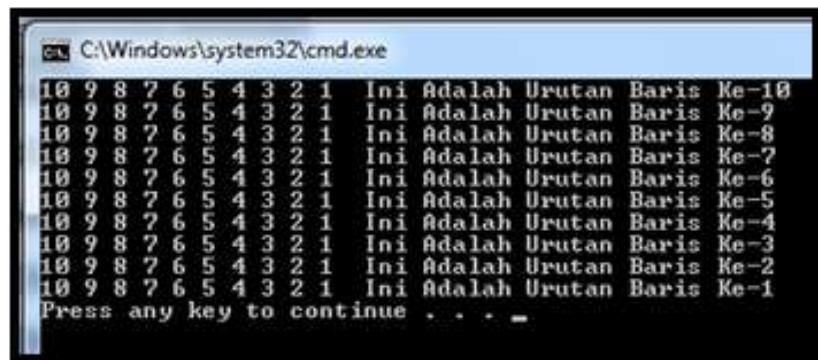
```
C:\Windows\system32\cmd.exe  
10 9 8 7 6 5 4 3 2 1 Ini Adalah Urutan Baris Ke-10  
10 9 8 7 6 5 4 3 2 1 Ini Adalah Urutan Baris Ke-9  
10 9 8 7 6 5 4 3 2 1 Ini Adalah Urutan Baris Ke-8  
10 9 8 7 6 5 4 3 2 1 Ini Adalah Urutan Baris Ke-7  
10 9 8 7 6 5 4 3 2 1 Ini Adalah Urutan Baris Ke-6  
10 9 8 7 6 5 4 3 2 1 Ini Adalah Urutan Baris Ke-5  
10 9 8 7 6 5 4 3 2 1 Ini Adalah Urutan Baris Ke-4  
10 9 8 7 6 5 4 3 2 1 Ini Adalah Urutan Baris Ke-3  
10 9 8 7 6 5 4 3 2 1 Ini Adalah Urutan Baris Ke-2  
10 9 8 7 6 5 4 3 2 1 Ini Adalah Urutan Baris Ke-1  
Press any key to continue . . . _
```

Gambar 4.59 Tampilan *Output* Dengan *Logika For* Untuk Menampilkan nilai 10-1 dengan perulangan *decrement* [perulangan menurun]

## Keterangan Kode Program

Pada Program 4.24 merupakan contoh program dengan logika perulangan *for Decreament* dengan harga awal  $i=10$  dan batasan perulangan  $i \geq 1$ , artinya nilai  $i$  akan terus berulang jika nilai  $i$  mencapai nilai 1,  $i$  terus berkurang dengan  $i=i--$ .

Untuk lebih jelasnya dapat dilihat pada *output* berikut ini :



```
C:\Windows\system32\cmd.exe
10 9 8 7 6 5 4 3 2 1 Ini Adalah Urutan Baris Ke-10
10 9 8 7 6 5 4 3 2 1 Ini Adalah Urutan Baris Ke-9
10 9 8 7 6 5 4 3 2 1 Ini Adalah Urutan Baris Ke-8
10 9 8 7 6 5 4 3 2 1 Ini Adalah Urutan Baris Ke-7
10 9 8 7 6 5 4 3 2 1 Ini Adalah Urutan Baris Ke-6
10 9 8 7 6 5 4 3 2 1 Ini Adalah Urutan Baris Ke-5
10 9 8 7 6 5 4 3 2 1 Ini Adalah Urutan Baris Ke-4
10 9 8 7 6 5 4 3 2 1 Ini Adalah Urutan Baris Ke-3
10 9 8 7 6 5 4 3 2 1 Ini Adalah Urutan Baris Ke-2
10 9 8 7 6 5 4 3 2 1 Ini Adalah Urutan Baris Ke-1
Press any key to continue . . . -
```

Gambar 4.59 Tampilan *Output* Dengan Logika *For* Untuk Menampilkan nilai 10-1 dengan perulangan *decrement* [perulangan menurun]

## Contoh Program Logika For Dengan C#

//program 4.25 *For Decreament*

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace for11
{
    class Program
    {
        static void Main(string[] args)
        {
```

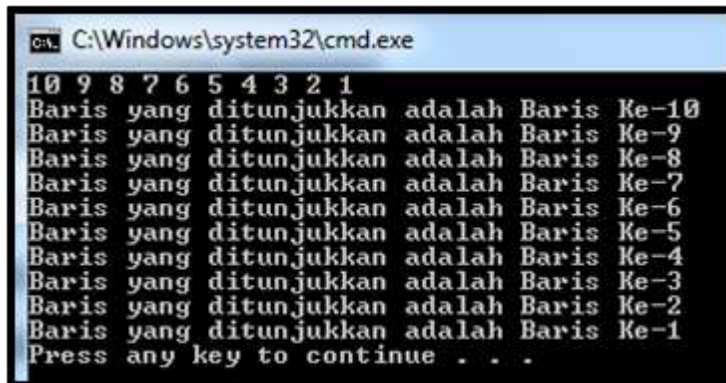


```
//mencetak nilai 10 sampai 1
    Console.Write(i + " ");
}
Console.WriteLine("");
for (int i = 10; i >=1; i--)
{
    //mencetak 10 sampai 1
    Console.WriteLine("Baris yang ditunjukkan adalah Baris Ke-" +
i);
}
}
}
```

## Keterangan Kode Program

Pada Program 4.25 merupakan contoh program dengan logika perulangan *for Decreament* dengan harga awal  $i=10$  dan batasan perulangan  $i \geq 1$ , artinya nilai  $i$  akan terus berulang jika nilai  $i$  mencapai nilai 1,  $i$  terus berkurang dengan  $i=i--$ .

Untuk lebih jelasnya dapat dilihat pada *output* berikut ini :



```
C:\Windows\system32\cmd.exe
10 9 8 7 6 5 4 3 2 1
Baris yang ditunjukkan adalah Baris Ke-10
Baris yang ditunjukkan adalah Baris Ke-9
Baris yang ditunjukkan adalah Baris Ke-8
Baris yang ditunjukkan adalah Baris Ke-7
Baris yang ditunjukkan adalah Baris Ke-6
Baris yang ditunjukkan adalah Baris Ke-5
Baris yang ditunjukkan adalah Baris Ke-4
Baris yang ditunjukkan adalah Baris Ke-3
Baris yang ditunjukkan adalah Baris Ke-2
Baris yang ditunjukkan adalah Baris Ke-1
Press any key to continue . . .
```

Gambar 4.60 Tampilan *Output* Dengan *Logika For* Untuk Menampilkan nilai 10-1 dengan perulangan *decrement* [perulangan menurun]

## Contoh Program Logika For Dengan C#

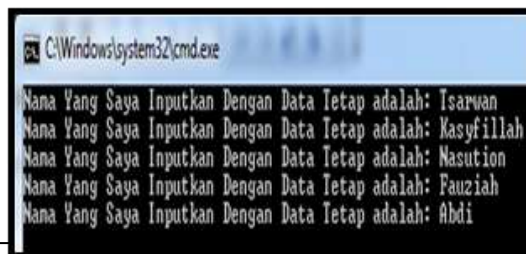
//Program 4.26 Menggunakan Foreach

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Foreach1
{
    class Program
    {
        static void Main(string[] args)
        {
            string[] nama =
            { "Tsarwan", "Kasyfillah", "Nasution", "Fauziah", "Abdi" };
            foreach (string nm in nama)
            Console.WriteLine("Nama Yang Saya Inputkan Dengan Data Tetap
            adalah: " + nm);
            Console.ReadLine();
        }
    }
}
```

## Keterangan Kode Program

Pada Program 4.26 merupakan contoh program dengan logika perulangan *foreach* dengan nilai yang sudah dideklarasikan secara tetap dengan isi nama yaitu :”Tsarwan, Kasyfillah, Nasution, Fauziah dan Abdi.” Tipe data yang digunakan pada program 4.26 adalah *string*, yaitu menghasilkan *output* bukan dalam bentuk angka, namun dalam bentuk gabungan beberapa huruf. Untuk lebih jelasnya dapat dilihat pada *output* berikut



```
C:\Windows\system32\cmd.exe
Nama Yang Saya Inputkan Dengan Data Tetap adalah: Tsarwan
Nama Yang Saya Inputkan Dengan Data Tetap adalah: Kasyfillah
Nama Yang Saya Inputkan Dengan Data Tetap adalah: Nasution
Nama Yang Saya Inputkan Dengan Data Tetap adalah: Fauziah
Nama Yang Saya Inputkan Dengan Data Tetap adalah: Abdi
```

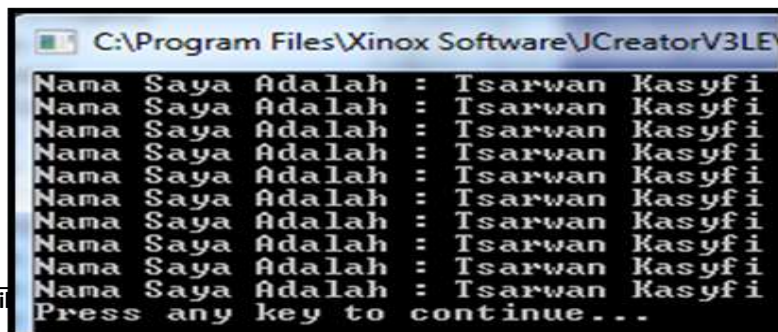
## Contoh Program Logika For Dengan Java

//Program 4.27 Menggunakan Java

```
class for2 {  
    public static void main(String[] args) {  
        for (int i=0; i<10; i++) {  
            System.out.println("Nama Saya Adalah : Tsarwan Kasyfi");  
        }  
    }  
}
```

## Keterangan Kode Program

Pada Program 4.27 merupakan contoh program dengan logika perulangan *for* sebagai berikut:” for (int i=0; i<10; i++)”. Artinya nilai perulangan dimulai dari 0 dan kondisi berakhir jika nilai i<10, proses akan terus berulang jika nilai i masih memenuhi kondisi tersebut[i<10, berarti ada 10 data nama yang akan tercetak, karena dimulai dari nilai indeks 0, sampai 9. Untuk lebih jelasnya dapat dilihat pada *output* berikut:



```
C:\Program Files\Xinox Software\JCreatorV3LE  
Nama Saya Adalah : Tsarwan Kasyfi  
Nama Saya Adalah : Tsarwan Kasyfi  
Nama Saya Adalah : Tsarwan Kasyfi  
Nama Saya Adalah : Tsarwan Kasyfi  
Nama Saya Adalah : Tsarwan Kasyfi  
Nama Saya Adalah : Tsarwan Kasyfi  
Nama Saya Adalah : Tsarwan Kasyfi  
Nama Saya Adalah : Tsarwan Kasyfi  
Nama Saya Adalah : Tsarwan Kasyfi  
Press any key to continue...
```

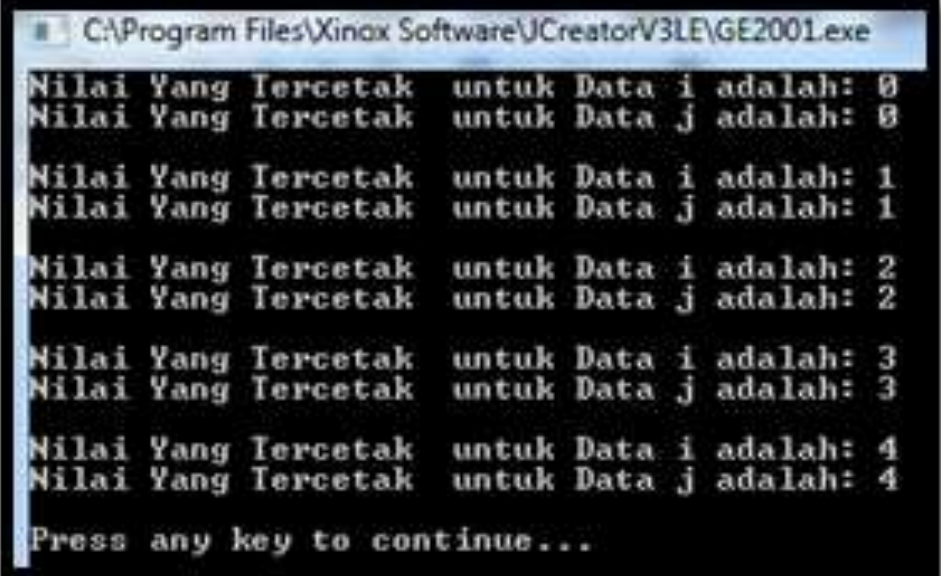
## Contoh Program Logika For Dengan Java

//Program 4.28 Menggunakan Java

```
class for23 {
public static void main(String[] args) {
int i=0;
for (int j=0; j < 5; j++) {
// Mencetak nilai baris
System.out.println("Nilai Yang Tercetak untuk Data i adalah: " + i);
// Mencetak nilai kolom
System.out.println("Nilai Yang Tercetak untuk Data j adalah: " + j);
System.out.println();
    i++;
    }
}
}
```

## Keterangan Kode Program

Pada Program 4.28 merupakan contoh program dengan logika perulangan *for* sebagai berikut:”int i=0; menunjukkan nilai indeks i dimulai dari 0 dan terus bertambah i++[i=i+1] dan nilai i akan terus bertambah jika nilai i<5[artinya data yang tercetak adalah nilai data ke-0 sampai 4[ada 5 elemen data]/untuk nilai baris. for (int j=0; j<5; j++)”. Artinya nilai perulangan kolom dimulai dari 0 dan kondisi berakhir jika nilai j<10, proses akan terus berulang jika nilai j masih



```
C:\Program Files\Xinox Software\JCreatorV3LE\GE2001.exe
Nilai Yang Tercetak untuk Data i adalah: 0
Nilai Yang Tercetak untuk Data j adalah: 0
Nilai Yang Tercetak untuk Data i adalah: 1
Nilai Yang Tercetak untuk Data j adalah: 1
Nilai Yang Tercetak untuk Data i adalah: 2
Nilai Yang Tercetak untuk Data j adalah: 2
Nilai Yang Tercetak untuk Data i adalah: 3
Nilai Yang Tercetak untuk Data j adalah: 3
Nilai Yang Tercetak untuk Data i adalah: 4
Nilai Yang Tercetak untuk Data j adalah: 4
Press any key to continue...
```

Gambar 4.63 Tampilan *Output* Dengan *Logika For* Menggunakan Java Untuk mencetak Nilai Baris dan Nilai Kolom mulai dari indeks ke 0 sampai indeks ke 4[ada 5 elemen data baris dan kolom]

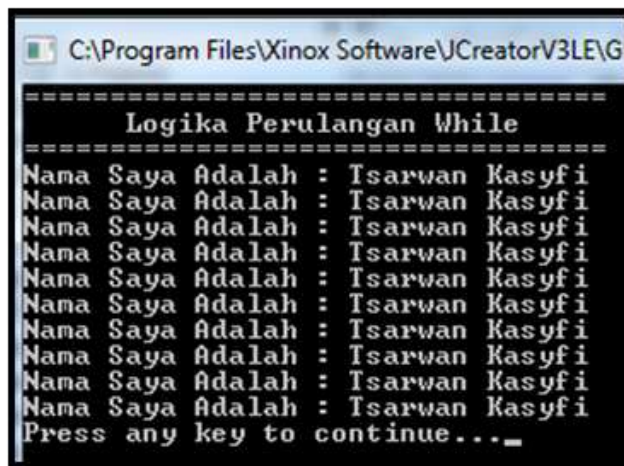
## Contoh Program Logika While Dengan Java

//Program 4.29 Menggunakan Java

```
class while1 {
public static void main(String[] args) {                               int
i=0;
System.out.println("=====");
System.out.println("    Logika Perulangan While");
System.out.println("=====");
while (i<10) {
System.out.println("Nama Saya Adalah : Tsarwan Kasyfi");
```

## Keterangan Kode Program

Pada Program 4.29 merupakan contoh program dengan logika perulangan *While* sebagai berikut: "int i=0; menunjukkan nilai indeks i dimulai dari 0 dan terus bertambah i++[i=i+1] dan nilai i akan terus bertambah jika nilai i<10[artinya data yang tercetak adalah nilai data ke-0 sampai 9[ada 10 elemen data. Untuk lebih jelasnya dapat dilihat pada *output* berikut:



```
C:\Program Files\Xinox Software\JCreatorV3LE\G
=====
Logika Perulangan While
=====
Nama Saya Adalah : Tsarwan Kasyfi
Nama Saya Adalah : Tsarwan Kasyfi
Nama Saya Adalah : Tsarwan Kasyfi
Nama Saya Adalah : Tsarwan Kasyfi
Nama Saya Adalah : Tsarwan Kasyfi
Nama Saya Adalah : Tsarwan Kasyfi
Nama Saya Adalah : Tsarwan Kasyfi
Nama Saya Adalah : Tsarwan Kasyfi
Nama Saya Adalah : Tsarwan Kasyfi
Nama Saya Adalah : Tsarwan Kasyfi
Press any key to continue..._
```

Gambar 4.64 Tampilan *Output* Dengan *Logika While* Menggunakan Java Untuk mencetak Nama Saya Adalah :Tsarwan Kasyfi Sebanyak 10 x

## Contoh Program Logika While Dengan Java

//Program 4.30

```
class while2 {
    public static void main(String[] args) {

        // memberikan nilai konstan untuk menghasilkan nilai faktorial
        int nilai=8;
```

```
}  
int i=nilai;  
do {  
    System.out.print(i);  
    if (i != 1) {  
        System.out.print(" * ");  
    } else {  
        System.out.print(" = ");  
    }  
    //mencetak nilai perkalian faktorial  
    hasil *= i;  
    i--;  
    // menggunakan logika while  
} while (i >= 1);  
System.out.println(hasil);  
}  
}
```

## Keterangan Kode Program

Pada Program 4.30 merupakan contoh program dengan logika perulangan *While* sebagai berikut:”akan mencetak nilai faktorial 8! Yaitu :  $8*7*6*5*4*3*2*1$ , dan nilai hasil diberikan nilai awal =1, dengan kondisi *if*. Untuk lebih jelasnya dapat dilihat *output* sebagai berikut:



```
C:\Program Files\Xinox Software\Creator\3LE\GE2001.exe  
8 ! [Faktorial Adalah] = 8 * 7 * 6 * 5 * 4 * 3 * 2 * 1 = 40320  
Press any key to continue...
```

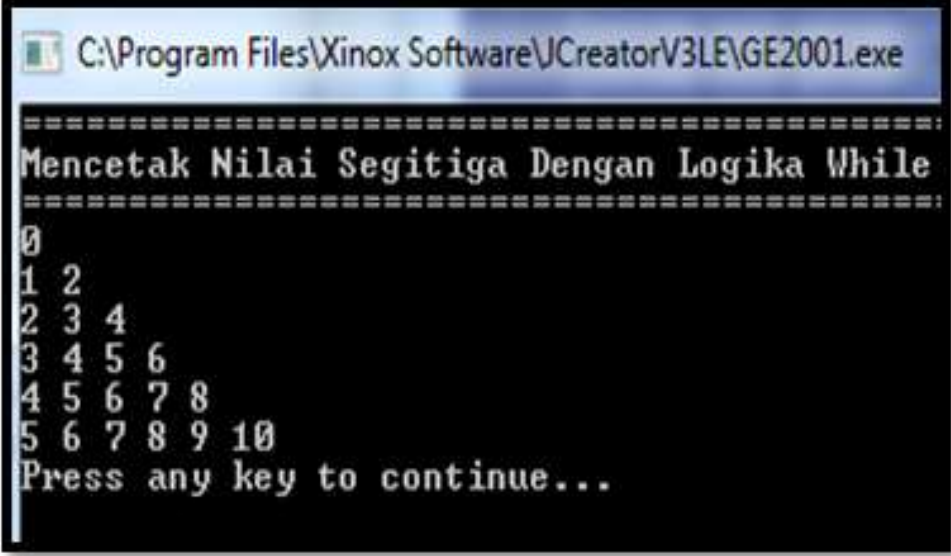
## Contoh Program Logika While Dengan Java

```
//Program 4.31
class while3 {
public static void main(String[] args) {
System.out.println("=====");
System.out.println("Mencetak Nilai Segitiga Dengan Logika While");
System.out.println("=====");
// Mendeklarasikan baris dan Kolom dengan harga awal baris =0
int Brs=0, Kol;
do {
//Memberikan harga awal kolom =0
Kol=0;
do {
//Mencetak isi baris dan kolom
System.out.print(Brs+Kol + " ");
// indeks kolom terus bertambah
Kol++;
//Mendeklarasikan perulangan dengan kondisi elemen kolom <= Baris
} while (Kol<=Brs);
System.out.println();
// indeks baris terus bertambah
Brs++;
// mendeklarasikan perulangan dengan kondisi elemen baris <=5
} while (Brs<=5);
}
}
```

## Keterangan Kode Program

Pada Program 4.31 merupakan contoh program dengan logika





```
C:\Program Files\Xinox Software\JCreatorV3LE\GE2001.exe
=====
Mencetak Nilai Segitiga Dengan Logika While
=====
0
1 2
2 3 4
3 4 5 6
4 5 6 7 8
5 6 7 8 9 10
Press any key to continue...
```

Gambar 4.66 Tampilan *Output* Dengan *Logika While* Bersarang Menggunakan Java Untuk mencetak Nilai Segitiga.

## Contoh Program Logika While Dengan Java

//Program 4.32

```
class while4 {
    public static void main(String[] args) {
        System.out.println("=====");
        System.out.println("Mencetak Nilai Segitiga Dengan Logika While");
        System.out.println("=====");
        int Brs=0, Kol;
        do {
            Kol=0;
            do {
                System.out.print(Brs +Kol + " ");
                Kol++;
            } while (Kol<=Brs);
```

## Keterangan Kode Program

Pada Program 4.32 merupakan contoh program dengan logika perulangan *While* bersarang untuk menghasilkan output segitiga dengan melakukan proses perulangan baris dan kolom dengan harga awal baris dan kolom telah ditentukan, begitu pula dengan batasan yang dideklarasikan dengan logika *while*, yang artinya ketika kondisi masih terpenuhi, maka akan mencetak hasil yang sesuai, untuk lebih jelasnya dapat dilihat pada *output* berikut ini.



```
C:\Program Files\Xinox Software\JCreatorV3LE\GE2001.exe
Mencetak Nilai Segitiga Dengan Logika While
-----
1
1 2
1 2 3 4
1 2 3 4 5 6
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10 11 12
1 2 3 4 5 6 7 8 9 10 11 12 13 14
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Press any key to continue...
```

Gambar 4.67 Tampilan *Output* Dengan *Logika While* Bersarang Menggunakan Java Untuk mencetak Nilai Segitiga

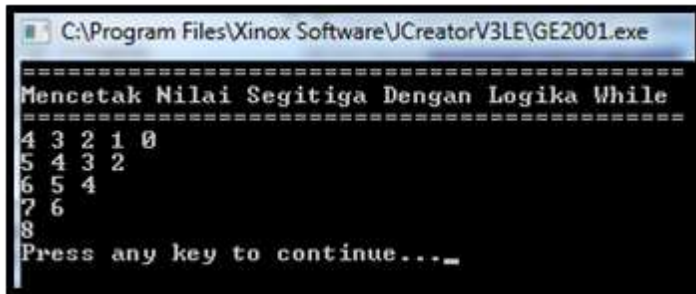
## Contoh Program Logika While Dengan Java

//Program 4.33 Kombinasi *Increment* dan *Decrement*

```
class while5 {
public static void main(String[] args) {
System.out.println("=====");
System.out.println("Mencetak Nilai Segitiga Dengan Logika While");
System.out.println("=====");
int Brs=0, Kol;
do {
Kol=4;
do {
System.out.print(Brs+Kol + " ");
Kol--;
```

## Keterangan Kode Program

Pada Program 4.33 merupakan contoh program dengan logika perulangan *While* bersarang untuk menghasilkan output segitiga dengan melakukan proses perulangan baris dan kolom dengan harga awal baris dan kolom telah ditentukan, begitu pula dengan batasan yang dideklarasikan dengan logika *while*, yang artinya ketika kondisi masih terpenuhi, maka akan mencetak hasil yang sesuai, dan menambahkan logika penambahan/increment dan penurunan/decrement untuk lebih jelasnya dapat dilihat pada *output* berikut ini.



```
C:\Program Files\Xinox Software\JCreatorV3LE\GE2001.exe
=====
Mencetak Nilai Segitiga Dengan Logika While
=====
4 3 2 1 0
5 4 3 2
6 5 4
7 6
8
Press any key to continue...
```

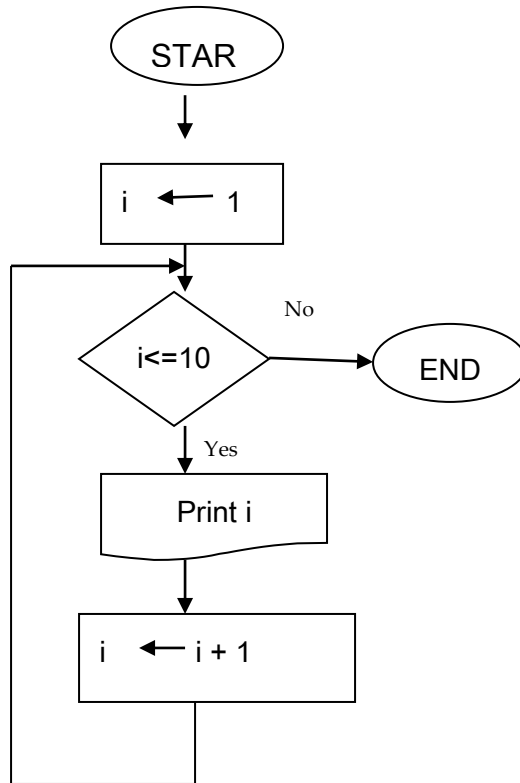
Gambar 4.68 Tampilan *Output* Dengan *Logika While* Bersarang Menggunakan Java Untuk mencetak Nilai Segitiga dengan logika increment dan decrement.

## Info Flowchart Dengan Logika For

Untuk pemrograman logika for, simbol perulangan yang digunakan cukup simbol proses misalnya seperti berikut :

```
For i=1;j<=10;i++
```

## Flowchart Logika For Increment



Flowchart 4.1

Keterangan *Flowchart* 4.1 :

*Flowchart* 4.1 merupakan *flowchart increment* `++` yang digunakan sebagai alur logika perulangan untuk harga awal `i=1`, kemudian melakukan pengecekan kondisi apakah nilai

$i \leq 10$ , maka nilai  $i$  akan dicetak, jika tidak memenuhi kondisi maka proses berakhir, kemudian nilai  $i$  akan terus bertambah sampai kondisi berakhir [ $i \leq 10$ ].

Dari flowchart tersebut terlihat output berikut ini

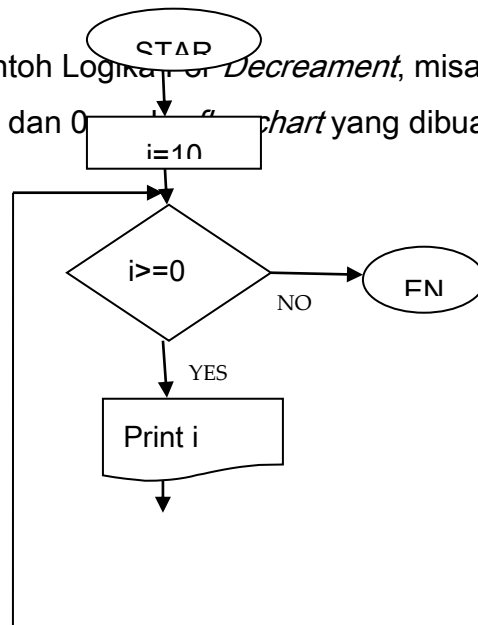


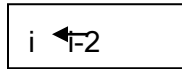
```
(Inactive D:\C\_LAT-1\FOR1-10.EXE)
Menampilkan 1 s.d 10 dengan FOR
1
2
3
4
5
6
7
8
9
10
```

## Flowchart Logika For Decreament

Gambar 4.69 Tampilan Hasil Dengan Logika *For* sesuai Flowchart 4.1

Untuk Contoh Logika *For Decreament*, misalnya akan dicetak nilai 10,8,6,4,2 dan 0





Keterangan *Flowchart* 4.2 : <sup>Flowchart 4.2</sup>

*Flowchart* 4.1 merupakan *flowchartDecreament* [--] yang digunakan sebagai alur logika perulangan untuk harga awal  $i=10$ , kemudian melakukan pengecekan kondisi apakah nilai  $i \geq 0$ , maka nilai  $i$  akan dicetak, jika tidak memenuhi kondisi maka proses berakhir, kemudian nilai  $i$  akan terus berkurang sampai kondisi berakhir di  $[i \geq 0]$ .

Output dari *flowchart* tersebut adalah :

```
(Inactive D:\C_LAT~1\FOR1-10.EXE)
Menampilkan 10 s.d 0 dengan FOR
10
8
6
4
2
0
```

Gambar 4.70 Logika For dengan nilai Decreament [--]

# PENGAYAAN

Jawablah latihan berikut dengan jelas dan benar

1. Mengapa dalam bahasa pemrograman [C++,C# dan Java diciptakan Struktur Perulangan
2. Tuliskan macam dan bentuk umum struktur perulangan dengan lengkap dan jelas.
3. Berikan contoh menggunakan struktur perulangan dalam bahasa C++, C# dan Java menggunakan logika For
4. Berikan contoh menggunakan struktur perulangan dalam bahasa C++, C# dan Java Menggunakan Logika While
5. Gambarkan Pula Flowchart dari program yang telah dibuat

# GLOSARIUM

1. Struktur perulangan pada bahasa pemrograman digunakan untuk memudahkan programmer dalam proses pengetikan program, artinya jika ingin membuat suatu logika berulang programmer cukup membuat satu baris *statement* saja, mempersingkat waktu pengetikan, menghemat penggunaan memori dan kerja *compiler* lebih ringan

- 2 Contoh logika perulangan yang digunakan adalah logika *for*

# GLOSARIUM BAB IV

3. Pada Logika perulangan baik *for* dan *while* menggunakan harga awal, kondisi dan peubah
4. Untuk nilai peubah menggunakan peubah *increment* dan *decrement*
5. Peubah *increment* adalah peubah yang nilainya selalu bertambah, misalnya  $x++$  artinya nilai  $x=x+1$
6. Peubah *Decrement* adalah peubah yang nilainya selalu berkurang, misalnya  $x--$  artinya nilai  $x=x-1$
7. Modulo adalah operator yang digunakan untuk menghasilkan sisa hasil bagi, misalnya  $5 \text{ mod } 2$  hasilnya adalah : 1,  $2 \text{ mod } 2$  hasilnya adalah 0 dan seterusnya.
8. Untuk operator modulo lambang atau symbol yang digunakan adalah [%].
9. Bentuk umum dari logika *for* adalah :  
  
For\_tipe\_data\_ inisialisasi nilai\_awal\_kondisi\_peubah  
  
Misalnya: *For* (int  $i=0;i \leq 5;i++$ ) merupakan contoh *for* dengan nilai awal=0, kondisi  $i \leq 5$ , nilai  $i$  *increment*[bertambah]. *For* (int  $i=5;i >= 0,i--$ ) merupakan contoh *for* dengan nilai awal=5, kondisi  $i >= 0$





*Sumber: defenseindustrydaily.com*

# BAB V

## ARRAY PADA BAHASA PEMROGRAMAN

### **Tujuan Pembelajaran**

Setelah mempelajari bab ini Mahasiswa diharapkan mampu:  
Menjelaskan Konsep umum array dan dapat membuat contoh program menggunakan bahasa pemrograman C++, C# dan Java, dapat membedakan array 1 dimensi dan multidimensi dan dapat mengimplementasikannya ke dalam contoh program

*Array* merupakan kumpulan data yang digunakan pada bahasa pemrograman yang disusun dalam bentuk *indeks* baris dan *indeks* kolom serta memiliki tipe data sejenis. Macam *Array* ada *array 1 dimensi* dan *multidimensi*

Kata Kunci: *Array*, *indeks* baris, *indeks* kolom, *1 dimensi*, *multidimensi*



# 5.1 Pengertian Array & Macamnya

## Pengertian *Array*

*Array* merupakan kumpulan elemen data yang terdiri dari elemen baris dan kolom dengan tipe data sejenis. *Array* yang akan dibahas adalah :

- a. *Array 1 dimensi*
- b. *Array multidimensi*

Keterangan :

- a. *Array 1 Dimensi*

Bentuk Umum dari *array* 1 dimensi adalah :

Tipe Data Nama array elemen arra
----------------------------------

Contoh Deklarasi *Array* :

```
int Angka[10]
```

## 1Dimensi

Pada contoh deklarasi *array* tersebut adalah :

1. Tipe Data yang digunakan adalah : *Integer*
2. Nama Array : Angka
3. Banyak Elemen array adalah : 10 Data Elemen Array

Untuk lebih jelasnya dapat terlihat pada program berikut ini menggunakan C++ dan Java

### 52 Contoh Program Dengan Array

```
// Program 5.1
#include <iostream.h>
#include <conio.h>
void main()
{
clrscr ();
int Angka[10];
int i;
cout<<"=====" ;cout<<"\n";
cout<<"Menggunakan Array 1 Dimensi" ;cout<<"\n";
cout<<"=====" ;cout<<"\n";
```

```
for(i=0;i<10;i++)
{
cout<<"Inputkan Data Array "<<i<<":"; cin>>Angka[i];
}
for(i=0;i<10;i++)
{
cout<<endl<<"array ke-"<<i<<" Adalah :- " <<Angka[i];
cout<<"\n";

}
getch ();
}
```

## Keterangan Kode Program

Dari program 5.1 terlihat menggunakan deklarasi data *array* dengan *int* Angka[10], tipe data *integer*, banyak elemen *array* yang terbentuk adalah 10 buah data. Untuk Logika *For* yang digunakan adalah :

1. Logika *For*: *for*(i=0;i<10;i++) artinya nilai awal i dimulai dari nilai =0 dengan kondisi perulangan data i<10[artinya data yang tercetak adalah 10 data, yaitu dihitung mulai dari elemen ke-0 sampai elemen ke-9, nilai i terus bertambah dengan *increment*[i++ artinya i=i+1]. Logika *For* yang pertama ini berfungsi untuk menampung atau menyimpan masing-masing nilai dengan *indeks* elemen baris dan elemen kolom
2. Logika *For* berikutnya adalah : *for*(i=0;i<10;i++), logika *for* ini

Sesuai sekali dengan logika for yang dituliskan `for(i=0;i<10;i++)`, yaitu dimulai dari indeks ke-0 dan

Untuk lebih jelasnya dari program 5.1 dapat dilihat *output* sebagai berikut :

```
D:\C_LAT-1\ARRAY\EXE
Menggunakan Array 1 Dinens.
-----
Inputkan Data Array 0:9
Inputkan Data Array 1:8
Inputkan Data Array 2:7
Inputkan Data Array 3:6
Inputkan Data Array 4:5
Inputkan Data Array 5:4
Inputkan Data Array 6:3
Inputkan Data Array 7:2
Inputkan Data Array 8:1
Inputkan Data Array 9:0

array ke-0 Adalah :- 9
array ke-1 Adalah :- 8
array ke-2 Adalah :- 7
array ke-3 Adalah :- 6
array ke-4 Adalah :- 5
array ke-5 Adalah :- 4
array ke-6 Adalah :- 3
array ke-7 Adalah :- 2
array ke-8 Adalah :- 1
```

```
{  
  cout<<"Inputkan Karakter/Huruf Ke-"<<i<<":"; cin>>Data[i];  
}  
for(i=0;i<7;i++)  
{  
  cout<<endl<<"Huruf Yang Tercetak Pada Urutan Ke- "<<i<<" Adalah :  
  "<<Data[i];  
  }  getch ();  
}
```

## Keterangan Kode Program

Dari program 5.2 terlihat menggunakan deklarasi data array dengan *char* Data[6], tipe data *Char* [karakter/huruf], banyak elemen array yang terbentuk adalah 6 buah data. Untuk Logika *For* yang digunakan adalah :

# Output Dari Program 5.1

29

```
D:\C_LAT-1\ARRAYCHA.EXE
Inputkan Karakter/Huruf Ke-0:F
Inputkan Karakter/Huruf Ke-1:A
Inputkan Karakter/Huruf Ke-2:U
Inputkan Karakter/Huruf Ke-3:2
Inputkan Karakter/Huruf Ke-4:I
Inputkan Karakter/Huruf Ke-5:A
Inputkan Karakter/Huruf Ke-6:H

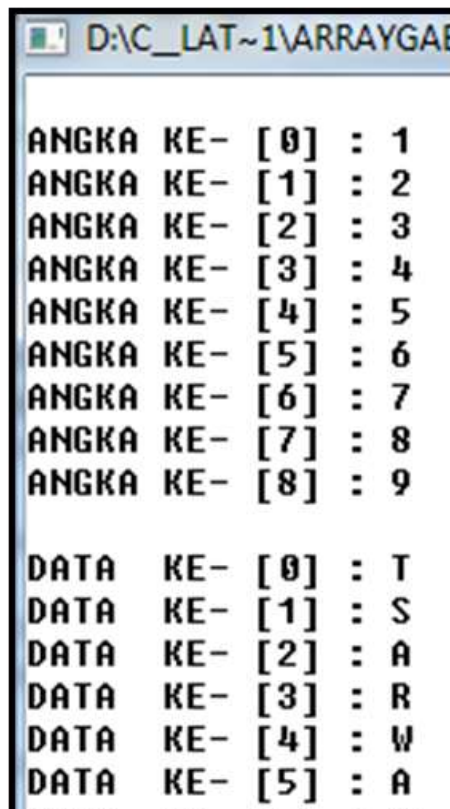
Huruf Yang Tercetak Pada Urutan Ke- 0 Adalah : F
Huruf Yang Tercetak Pada Urutan Ke- 1 Adalah : A
Huruf Yang Tercetak Pada Urutan Ke- 2 Adalah : U
Huruf Yang Tercetak Pada Urutan Ke- 3 Adalah : 2
Huruf Yang Tercetak Pada Urutan Ke- 4 Adalah : I
Huruf Yang Tercetak Pada Urutan Ke- 5 Adalah : A
Huruf Yang Tercetak Pada Urutan Ke- 6 Adalah : H
```



Gambar 5.2 Tampilan Array 1 Dimensi Menggunakan Array Karakter

```
//Program 5.3 Menggunakan Array Gabungan
#include<iostream.h>
#include<conio.h>
main()
{
clrscr();
int x;
int ANGKA[9] = {1,2,3,4,5,6,7,8,9};
char DATA[7] = {'T','S','A','R','W','A','N'};
for(x=0; x<9;++x)
{
cout << "\nANGKA KE- ["<<x<<"] : "<< ANGKA[x];
}
cout<<"\n";
for(x=0; x<7; ++x)
{
cout << "DATA KE- ["<<x<<"] : "<< DATA[x];
}
```

Untuk lebih jelasnya dapat dilihat pada *output* berikut ini:



```
D:\C_LAT~1\ARRAYGAE
ANGKA KE- [0] : 1
ANGKA KE- [1] : 2
ANGKA KE- [2] : 3
ANGKA KE- [3] : 4
ANGKA KE- [4] : 5
ANGKA KE- [5] : 6
ANGKA KE- [6] : 7
ANGKA KE- [7] : 8
ANGKA KE- [8] : 9

DATA KE- [0] : T
DATA KE- [1] : S
DATA KE- [2] : A
DATA KE- [3] : R
DATA KE- [4] : W
DATA KE- [5] : A
```

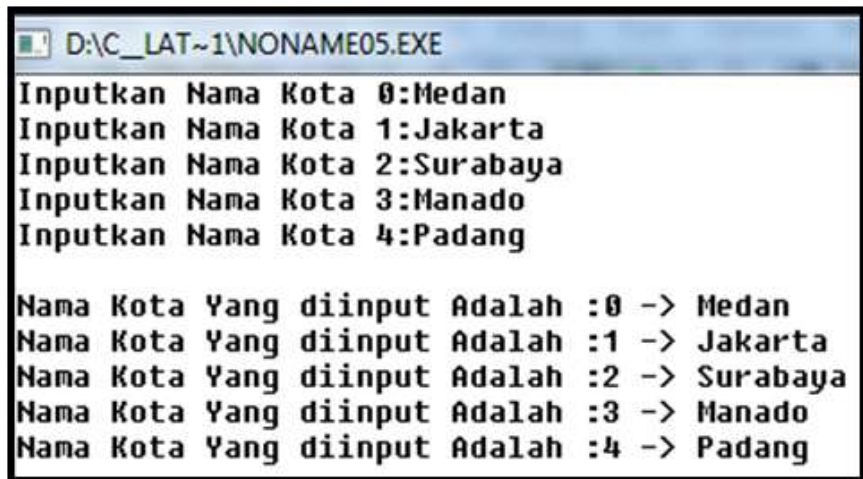
```
}  
  for(i=0;i<5;i++)  
  {  
    cout<<endl<<"Nama Kota Yang diinput Adalah :"<<i<<" -> "<<Kota[i];  
  }  getch () ;  
}
```

## Keterangan Kode Program

Dari program 5.4 terlihat menggunakan deklarasi data *array* karakter.

Untuk Logika *For* yang digunakan adalah :

1. Logika *For* : `for(i=0;i<5;i++)` artinya nilai awal `i` dimulai dari nilai `=0` dengan kondisi perulangan data `1,5`[artinya data yang tercetak adalah 5 data, yaitu dihitung mulai dari elemen ke-0 sampai



```
D:\C_LAT~1\NONAME05.EXE
Inputkan Nama Kota 0:Medan
Inputkan Nama Kota 1:Jakarta
Inputkan Nama Kota 2:Surabaya
Inputkan Nama Kota 3:Manado
Inputkan Nama Kota 4:Padang

Nama Kota Yang diinput Adalah :0 -> Medan
Nama Kota Yang diinput Adalah :1 -> Jakarta
Nama Kota Yang diinput Adalah :2 -> Surabaya
Nama Kota Yang diinput Adalah :3 -> Manado
Nama Kota Yang diinput Adalah :4 -> Padang
```

Gambar 5.4 Tampilan *Array* 1 Dimensi Untuk mencetak nama Kota dengan 5 buah elemen data kota dan panjang maksimum dari masing-masing kota adalah 10 karakter

**CONTOH PROGRAM ARRAY 1 DIMENSI DENGAN BAHASA C#**

```
int i, n = 0;
Console.Write("Inputkan Jumlah Mahasiswa : ");
n = int.Parse(Console.ReadLine());
for (i = 1; i <= n; i++)
{
    Console.WriteLine("Inputkan Mahasiswa ke-" + i);
    Console.Write("\t Inputkan Nama Mahasiswa : ");
    nama[i] = Console.ReadLine();
    Console.Write("\t Inputkan Nilai Quiz : ");
    quiz[i] = int.Parse(Console.ReadLine());
    Console.Write("\t Inputkan Nilai UTS : ");
    uts[i] = int.Parse(Console.ReadLine());
    Console.Write("\t Inputkan Nilai UAS : ");
    uas[i] = int.Parse(Console.ReadLine());
    total[i] = 0.2 * quiz[i] + 0.4 * uts[i] + 0.4 * uas[i];
    Console.WriteLine();
    if (total[i] >= 80 && total[i] <= 100)
    {
        huruf[i] = "A ";
    }
    else if (total[i] >= 60 && total[i] <= 79)
    {
        huruf[i] = "B";
    }
    else if (total[i] >= 40 && total[i] <= 59)
    {
        huruf[i] = "C ";
    }
    else if (total[i] >= 20 && total[i] <= 39)
```

```
Console.WriteLine(i + " " + nama[i] + "\t" + quiz[i] +
"\t" + uts[i] + "\t" + uas[i] + "\t" + total[i] + "\t" +
huruf[i]);
}
Console.WriteLine("=====");
Console.WriteLine("          TERIMA KASIH          ");
Console.Read();
    }
}
}
```

## Keterangan Kode Program

Dari program 5.5 terlihat menggunakan deklarasi data *array* dengan beberapa data, yaitu :

```
string[] nama = new string[20];
int[] quiz = new int[10];
int[] uts = new int[10];
int[] uas = new int[10];
double[] total = new double[10];
string[] huruf = new string[20];
```

4. Untuk Logika *For* menggunakan deklarasi:

`"for (i = 1; i <= n; i++)"`, nilai awal yang digunakan adalah  $i=1$ , nilai  $i$  akan tercetak sesuai nilai  $n$  yang diinputkan dan nilai  $i$  akan terus bertambah  $i++$  yaitu nilai  $i=i+1$ .

5. Untuk `nama[i]`, `quiz[i]` `uts[i]` `uas[i]` adalah penyimpanan data dengan menggunakan indeks ke- $i$ .

6. Untuk Total Menggunakan rumus :

```
total[i] = 0.2 * quiz[1] + 0.4 * uts[i] + 0.4 * uas[i];
```

7. Untuk Logika *If* menggunakan gabungan dengan logika *And*[`&&`].

```
if (total[i] >= 80 && total[i] <= 100)
{
    huruf[i] = "A ";
}
else if (total[i] >= 60 && total[i] <= 79)
{
    huruf[i] = "B";
}
else if (total[i] >= 40 && total[i] <= 59)
```

```
C:\Windows\system32\cmd.exe
Inputkan Jumlah Mahasiswa : 5
Inputkan Mahasiswa ke-1
Inputkan Nama Mahasiswa : Tsarwan Ksyfi
Inputkan Nilai Quiz : 90
Inputkan Nilai UTS : 90
Inputkan Nilai UAS : 90

Inputkan Mahasiswa ke-2
Inputkan Nama Mahasiswa : Fauziah Kasyfi
Inputkan Nilai Quiz : 70
Inputkan Nilai UTS : 70
Inputkan Nilai UAS : 70

Inputkan Mahasiswa ke-3
Inputkan Nama Mahasiswa : Abdi Kasyfillah
Inputkan Nilai Quiz : 40
Inputkan Nilai UTS : 50
Inputkan Nilai UAS : 40

Inputkan Mahasiswa ke-4
Inputkan Nama Mahasiswa : Fath Alghifarii
Inputkan Nilai Quiz : 20
Inputkan Nilai UTS : 30
Inputkan Nilai UAS : 20

Inputkan Mahasiswa ke-5
Inputkan Nama Mahasiswa : Fifi Albanantani
Inputkan Nilai Quiz : 10
Inputkan Nilai UTS : 0
Inputkan Nilai UAS : 0
```



# Contoh Program Array 1 Dimensi Dengan Java

//Program 5.6 Array Dengan Java

```
class ARRAY{  
public static void main(String[] args) {  
3 String s = "TSARWAN KASYFI";  
char[] nama = new char[s.length()];  
nama = s.toCharArray();  
    System.out.println("=====|");  
    System.out.println("|Array 1 Dimensi Data Karakter|");  
    System.out.println("=====|");  
    System.out.println("Karakter pertama      : " + nama[0]);
```

## Keterangan Kode Program

Dari program 5.6 terlihat jelas menggunakan *array* 1 dimensi untuk jenis data *character*, bentuk umum yang digunakan pada program 5.6 adalah `char[] nama = new char[s.length()]`; dan untuk menghasilkan atau memunculkan data menggunakan `statement: System.out.println("Karakter pertama: " + nama[0]);` menunjukkan perintah *output* untuk mencetak karakter [0] dengan nilai 'T' dan seterusnya, Pada Program jenis data yang digunakan adalah : `String s = "TSARWAN KASYFI"`; Untuk lebih jelasnya terlihat *output* berikut ini:

## B. *Array* 2 Dimensi

Bentuk umum *array* 2 dimensi adalah

Tipe data Nama- <i>Array</i> Banyak
-------------------------------------

Contoh:

```
int matrix[2][2]
```

30

Terlihat pada contoh deklarasi tipe data : *Integer*, Nama *Array*:

Matrix, jumlah elemen baris : 2 dan jumlah elemen kolom=2.

Untuk lebih jelasnya dapat kita lihat program 5.7 berikut ini :

**Contoh Program Array 2 Dimensi Pada C++**

// Program 5.7



```
cout<<endl;
}
// mencetak nilai indeks baris dan indeks kolom
for(i=0;i<=1;i++)
{
for(j=0;j<=1;j++)
{
cout<<matrix[i][j]<<" ";
}
cout<<endl;
}
getch();
}
```

## Keterangan Kode Program

Pada Program 5.7 terlihat merupakan contoh program *array* 2 dimensi, dengan mendeklarasikan ordo baris dan kolom masing-masing 2 elemen.

1. Nama *Array* Yang dideklarasikan :  
int matrix[2][2]; yaitu dengan tipe data *integer*, nama array matrik, memiliki 2 buah elemen baris dan 2 buah elemen kolom
2. Logika *For* yang digunakan 1, adalah logika untuk menyimpan nilai indeks baris dan kolom yang diinputkan, yaitu menggunakan logika *for*.

```
for(i=0;i<=1;i++)
```

```
for(j=0;j<=1;j++)
```

3. Logika *For* yang digunakan 2, adalah logika untuk mencetak nilai indeks baris dan kolom yang diinputkan, yaitu menggunakan logika *for*:

```
for(i=0;i<=1;i++)
```

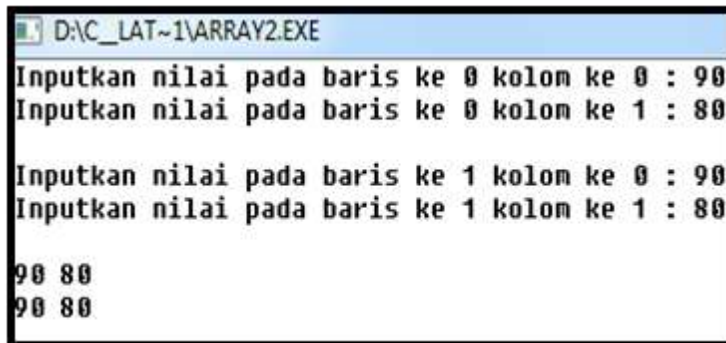
```
for(j=0;j<=1;j++)
```

*For* *i* artinya mendeklarasikan nilai baris dengan nilai awal baris  $i = 0$ , kemudian kondisi  $i \leq 1$ , yaitu [baris ke-0 dan baris ke- 1], nilai baris terus bertambah dengan peubah  $i++[i=i+1]$  dan akan berhenti jika nilai  $i \leq 1$ .

*For* *j* artinya mendeklarasikan nilai kolom dengan nilai awal kolom=0, kemudian kondisi  $j \leq 1$ , yaitu [baris ke-0 dan baris ke-1], nilai kolom terus bertambah dengan peubah  $j++[j=j+1]$  dan akan berhenti jika nilai  $i \leq 1$ .

4. `cin>>matrix[i][j]`, artinya menginputkan nilai baris dan kolom
5. `cout<<matrix[i][j]<<" "` , artinya mengeluarkan *output* nilai baris dan kolom.

Untuk lebih jelasnya dapat dilihat tampilan *output* berikutini:



```
D:\C_LAT~1\ARRAY2.EXE
Inputkan nilai pada baris ke 0 kolom ke 0 : 90
Inputkan nilai pada baris ke 0 kolom ke 1 : 80

Inputkan nilai pada baris ke 1 kolom ke 0 : 90
Inputkan nilai pada baris ke 1 kolom ke 1 : 80

90 80
90 80
```

Gambar 5.8 Tampilan Hasil Array 2 Dimensi Menggunakan Data Matrik dengan ordo 2 baris dan 2 Kolom

Contoh Program *array* 2 dimensi dengan data *character*

```
// Program 5.8
//array dua dimensi
#include<iostream.h>
#include<conio.h>
main()
{
// deklarasi elemen baris dan kolom
char nama[2][2][10];
int i,j;
// perulangan dan proses penyimpanan data baris dan kolom
for(i=0;i<=1;i++)
{
for(j=0;j<=1;j++)
{
cout<<"Inputkan Nama pada baris ke "<<i<<" kolom ke "<<j<<"
: ";
cin>>nama[i][j];
}
cout<<endl;
}
// mencetak nilai indeks baris dan indeks kolom
for(i=0;i<=1;i++)
{
for(j=0;j<=1;j++)
{
cout<<nama[i][j]<<" ";
```

## Keterangan Kode Program

Pada Program 5.8 terlihat merupakan contoh program *array* 2 dimensi, dengan mendeklarasikan ordo baris dan kolom masing-masing 2 elemen.

1. Nama *Array* Yang dideklarasikan :

`char nama[2][2][10];` yaitu dengan tipe data *char*, deklarasi array *nama*, memiliki 2 buah elemen baris dan 2 buah elemen kolom dan panjang *field* *nama* adalah maksimum 10 karakter

2. Logika *For* yang digunakan 1, adalah logika untuk menyimpan nilai indeks baris dan kolom yang diinputkan, yaitu menggunakan logika *for*:

```
for(i=0;i<=1;i++)
```

```
for(j=0;j<=1;j++)
```

*For* *i* artinya mendeklarasikan nilai baris dengan nilai awal baris  $i = 0$ , kemudian kondisi  $i \leq 1$ , yaitu [baris ke-0 dan baris ke- 1], nilai baris terus bertambah dengan peubah  $i++[i=i+1]$  dan akan berhenti jika nilai  $i \leq 1$ .

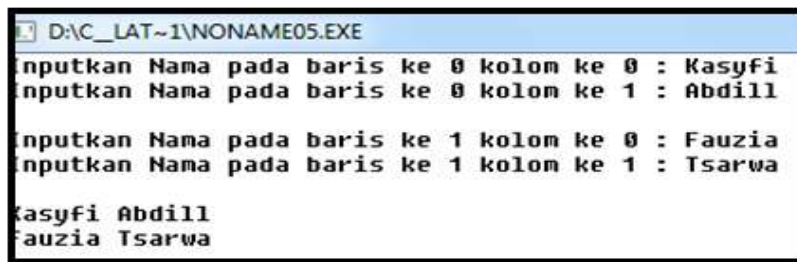
3. *For* *j* artinya mendeklarasikan nilai kolom dengan nilai awal kolom=0, kemudian kondisi  $j \leq 1$ , yaitu [baris ke-0 dan baris ke-1], nilai kolom terus bertambah dengan peubah  $j++[j=j+1]$  dan akan berhenti jika nilai  $i \leq 1$ . Logika *For* yang digunakan 2, adalah logika untuk mencetak nilai indeks baris dan kolom yang diinputkan, yaitu menggunakan logika *for*:



*For j* artinya mendeklarasikan nilai kolom dengan nilai awal kolom=0, kemudian kondisi  $j \leq 1$ , yaitu [baris ke-0 dan baris ke-1], nilai kolom terus bertambah dengan peubah  $j++[j=j+1]$  dan akan berhenti jika nilai  $i \leq 1$ .

4. `cin>>nama[i][j]`, artinya menginputkan nama pada baris dan kolom
5. `cout<<nama[i][j]<<" "` , artinya mengeluarkan *output* nama pada baris dan kolom.

Untuk lebih jelasnya dapat dilihat tampilan *output* berikutini:



```
D:\C_LAT~1\NONAME05.EXE
inputkan Nama pada baris ke 0 kolom ke 0 : Kasyfi
inputkan Nama pada baris ke 0 kolom ke 1 : Abdill

inputkan Nama pada baris ke 1 kolom ke 0 : Fauzia
inputkan Nama pada baris ke 1 kolom ke 1 : Tsarwa

kasyfi Abdill
fauzia Tsarwa
```

Gambar 5.9 Tampilan Hasil Array 2 Dimensi Menggunakan Data Nama dengan Pendeklarasian nama untuk 2 baris dan 2 kolom

6. Untuk penulisan indeks baris dan kolomnya adalah:

Nama[0.0]=Kasyfi  
Nama[0.1]=Abdill  
Nama[1.0]=Fauzia  
Nama[1.1]=Tsarwa

Dari hasil output yang ditampilkan terlihat jelas bahwa nilai indeks baris dan kolom yang dihasilkan sesuai dengan

## Contoh Program Array 2 Dimensi Dengan C#

```
//Program 5.9

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace LatihanArray
{
    class Program
    {
        public static void Main(string[] args)
        {
            // Penentuan array 2 dimensi
            int[,] matrik = new int[11, 11];
            //Indeks array dengan elemen 10 baris dan 10 kolom
            for (int baris = 1; baris < 11; baris++)
            for (int kolom = 1; kolom < 11; kolom++)
            matrik[baris, kolom] = baris * kolom;
            // Output Array dengan indeks baris dan Kolom
            for (int baris = 1; baris < 11; baris++)
            {
                for (int kolom = 1; kolom < 11; kolom++)

                Console.Write(matrik[baris, kolom] + "\t");
                Console.WriteLine();

            }
            Console.WriteLine("Press any key to continue . . .");
            Console.ReadLine();
        }
    }
}
```

## Keterangan Kode Program

Pada Program 5.9 terlihat merupakan contoh program *array* 2 dimensi, dengan mendeklarasikan ordo baris dan kolom masing-masing 10 elemen.

1. Nama *Array* Yang dideklarasikan :

`int[,] matrik = new int[11, 11];` yaitu dengan tipe data *int*, deklarasi array matrik, memiliki 10 buah elemen baris dan 10 buah elemen kolom.

2. Logika *For* yang digunakan 1, adalah logika untuk menyimpan nilai indeks baris dan kolom yang diinputkan, yaitu menggunakan logika *for*.

```
for(baris=1;baris<11;baris++)
```

```
for(kolom=1;kolom<11;kolom++)
```

*For* baris artinya mendeklarasikan nilai baris dengan nilai awal baris =1, kemudian kondisi baris<11, yaitu [baris ke-1 dan ke- 10], nilai baris terus bertambah dengan peubah baris++[baris=baris+1] dan akan berhenti jika nilai baris<11.

3. *For* kolom artinya mendeklarasikan nilai kolom dengan nilai awal kolom=1, kemudian kondisi kolom<11, yaitu [baris ke-1 sampai baris ke-10], nilai kolom terus bertambah dengan peubah kolom++[kolom=kolom+1] dan akan berhenti jika nilai i<11. Logika *For* yang digunakan 2, adalah logika untuk mencetak nilai indeks baris dan kolom yang *diinputkan*, yaitu menggunakan logika *for*.

5. `Console.WriteLine(matrik[baris, kolom],` Merupakan statement yang digunakan untuk menghasilkan *output* atau cetakan untuk hasil akhir matrik dengan ukuran 10 x 10

6. `"\t"`, digunakan untuk mengatur jarak antara tulisan yang satu dengan yang lain fungsinya sama seperti tab

Untuk lebih jelasnya dapat terlihat *output* berikut

```
1      2      3      4      5      6      7      8      9      10
2      4      6      8      10     12     14     16     18     20
3      6      9      12     15     18     21     24     27     30
4      8      12     16     20     24     28     32     36     40
5      10     15     20     25     30     35     40     45     50
6      12     18     24     30     36     42     48     54     60
7      14     21     28     35     42     49     56     63     70
8      16     24     32     40     48     56     64     72     80
9      18     27     36     45     54     63     72     81     90
10     20     30     40     50     60     70     80     90     100
Press any key to continue . . .
```

Gambar 5.10 Tampilan Hasil Array 2 Dimensi Menggunakan Data matriks dengan ordo 10 x 10 dan hasil yang tercetak merupakan perkalian dari isi elemen baris dan kolom

# Contoh Program Array 1 & 2 Dimensi Dengan Java

//Program 5.10

```
public class array3 {  
    public static void main(String args[]) {  
        int nilai[]=new int[5];  
        nilai[0]=70;  
        nilai[1]=80;  
        nilai[2]=90;  
        nilai[3]=95;  
        nilai[4]=88;  
        System.out.println("Menggunakan Array 1 Dimensi Data Konstan ");  
        System.out.println("===== ");  
        for(int i=0; i<5; i++)  
            System.out.println("Matrik Yang tercetak adalah = " + nilai[i]);  
    }  
}
```

## Keterangan Kode Program

1. Dari program 5.10, menggunakan data konstan dengan nilai data sudah dideklarasikan yaitu:

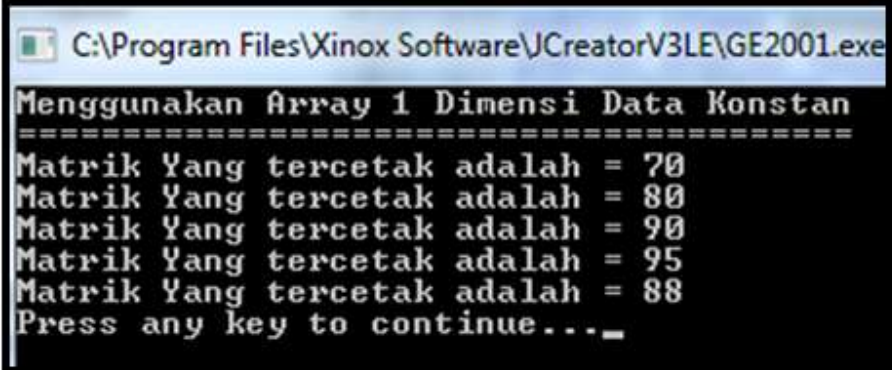
```
nilai[0]=70;
```

```
nilai[1]=80;
```

Nama *array* yang digunakan nilai dengan banyaknya elemen *array* adalah 5 data terdiri dari elemen ke-0 sampai elemen ke-4

3. Untuk logika *for* nya menggunakan bentuk umum sebagai berikut:  
`for(int i=0; i<5; i++)`, yang artinya harga awal dari nilai *i* adalah 0, melakukan proses perulangan dengan batas kondisi `i<5`[ada 5 data/elemen] dan nilai *i* terus bertambah[`i++`, artinya `i=i+1`]

Untuk lebih jelasnya dapat dilihat *output* berikut ini:



```
C:\Program Files\Xinox Software\JCreatorV3LE\GE2001.exe
Menggunakan Array 1 Dimensi Data Konstan
=====
Matrik Yang tercetak adalah = 70
Matrik Yang tercetak adalah = 80
Matrik Yang tercetak adalah = 90
Matrik Yang tercetak adalah = 95
Matrik Yang tercetak adalah = 88
Press any key to continue..._
```

Gambar 5.11 Tampilan Hasil Array 1 Dimensi Menggunakan Data konstan terdiri dari 5 buah elemen data

//Program 5.11

```
class Duadimensiarray
{
public static void main (String[] args)
{
//mendeklarasikan 2 baris dan 2 kolom
int[][] angka = new int[2][2];
angka[0][0]=70;
```

```
System.out.println("=====\n");
System.out.println("Contoh Array 2 Dimensi");
System.out.println("=====\n");
//mendeklarasikan logika for
for (int i=0; i<2; i++)
{
for (int j=0; j<2; j++)
System.out.print(angka[i][j] + " ");
System.out.println();
}
}
}
```

## Keterangan Kode Program

Pada Program 5.11 terlihat merupakan contoh program *array* 2 dimensi, dengan mendeklarasikan ordo baris dan kolom masing-masing 2 elemen.

1. Nama *Array* Yang dideklarasikan :

`int[][] angka = new int[2][2];`; yaitu dengan tipe data *int*, deklarasi array angka, memiliki 2 buah elemen baris dan 2 buah elemen kolom.

2. Logika *For* yang digunakan 1, adalah logika untuk menyimpan nilai indeks baris dan kolom yang diinputkan, yaitu menggunakan logika *for*:

```
for (int i=0; i<2; i++)
```

4. *For* kolom artinya mendeklarasikan nilai kolom dengan nilai awal kolom=0, kemudian kondisi kolom<2, yaitu [kolom ke-0 sampai kolom ke-1], nilai kolom terus bertambah dengan peubah kolom++[kolom=kolom+1] dan akan berhenti jika nilai i<11. Logika *For* yang digunakan 2, adalah logika untuk mencetak nilai indeks baris dan kolom yang *diinputkan*, yaitu menggunakan logika *for*:

```
for(baris=0;baris<2;baris++)
```

```
for(kolom=0;kolom<2;kolom++)
```

5. *For* baris artinya mendeklarasikan nilai baris dengan nilai awal baris baris =0, kemudian kondisi baris<2, yaitu[baris ke-0 sampai baris ke-1], nilai baris terus bertambah dengan peubah baris++[baris=baris+1] dan akan berhenti jika nilai baris<2.

6. Untuk Indeksnya dapat dideklarasikan sebagai berikut:

```
angka[0][0]=70;
```

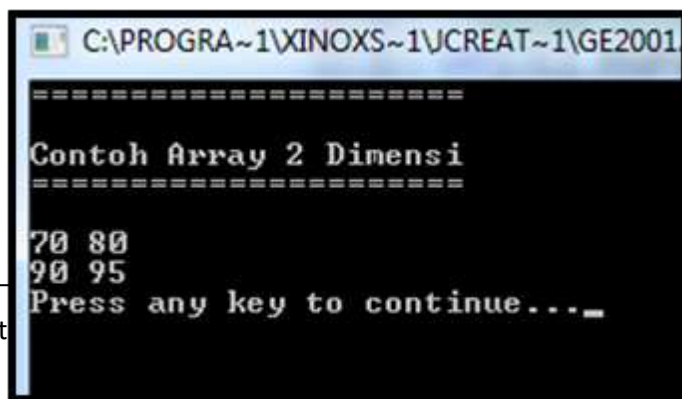
```
angka[0][1]=80;
```

```
angka[1][0]=90;
```

```
angka[1][1]=95;
```

7. `System.out.print(angka[i][j] + " ");` merupakan *output* yang dapat dicetak dengan memanggil *indeks* baris dan kolom yang sudah dideklarasikan.

Untuk lebih jelasnya dapat dilihat *output* berikut ini:



```
C:\PROGRA~1\XINOX~1\CREAT~1\GE2001
=====
Contoh Array 2 Dimensi
=====
70 80
90 95
Press any key to continue..._
```



# PENGAYAAN

Jawablah pertanyaan di bawah ini dengan jelas dan lengkap

1. Jelaskan tentang logika *Array*
2. Apa perbedaan logika *array* 1 dimensi dan 2 dimensi
3. Tuliskan bentuk umum logika *array* 1 dimensi
4. Tuliskan bentuk umum logika *array* 2 dimensi
5. Tuliskan contoh *array* 1 dimensi untuk data karakter
6. Tuliskan contoh *array* 1 dimensi untuk data integer
7. Tuliskan contoh *array* 2 dimensi untuk data *character*

# GLOSARIUM

1. Array merupakan kumpulan data yang memiliki tipe data sejenis
2. Jenis array yang digunakan adalah array 1 dimensi dan multidimensi
3. Pada array biasanya menggunakan baris dan kolom
4. Bentuk umum array yang digunakan adalah tipe data\_nama\_array\_element\_array
5. Logika for yang digunakan adalah logika elemen baris dan logika elemen kolom
6. Untuk Bentuk umum logika array 1 dimensi adalah : tipe\_data\_nama\_array\_banyak\_element
7. Contoh deklarasi `int angka[5];`



Sumber: [defenseindustrydaily.com](http://defenseindustrydaily.com)

# BAB VI

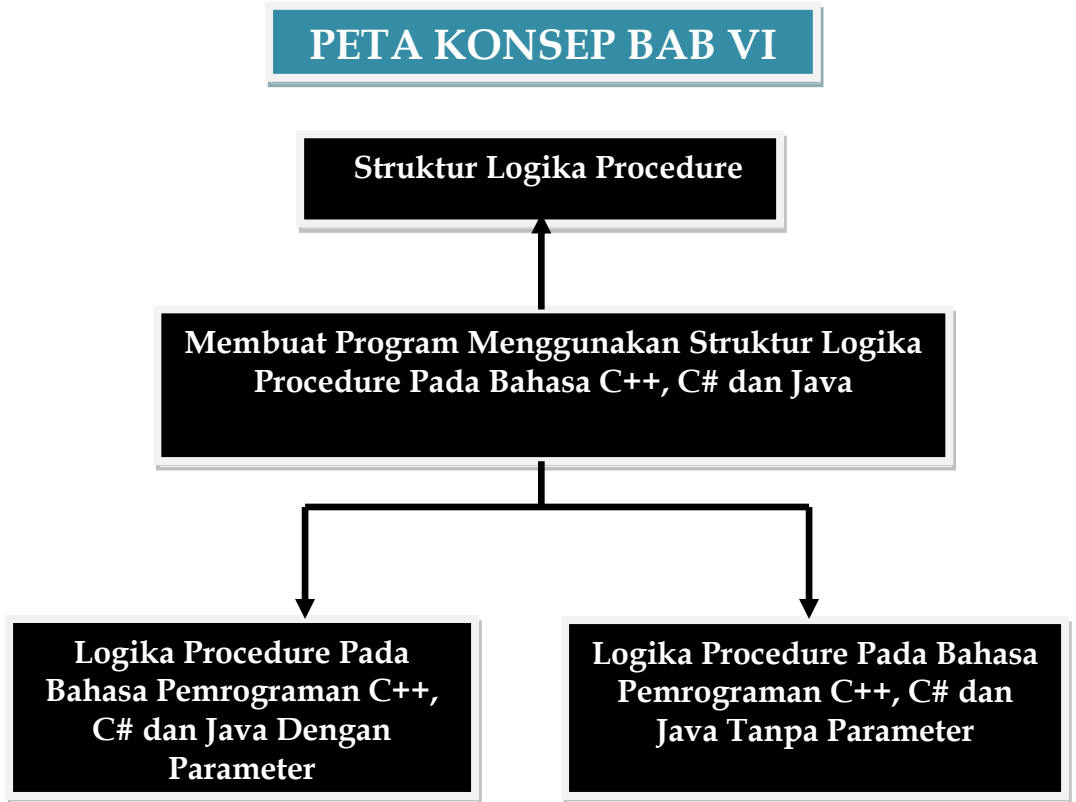
## *PROCEDURE* PADA BAHASA PEMROGRAMAN

### **Tujuan Pembelajaran**

Setelah mempelajari bab ini Mahasiswa diharapkan mampu:  
Menjelaskan Konsep umum *procedure* dan dapat membuat contoh program menggunakan bahasa pemrograman C++, C# dan Java, dapat membedakan pendeklarasian *procedure* yang digunakan dalam bahasa pemrograman

*Procedure* merupakan sub program yang digunakan dalam program utama yang fungsinya adalah untuk mengerjakan suatu proses tertentu tanpa mengembalikan hasil proses yang ada dalam sebuah program, umumnya pada *procedure* dideklarasikan dengan nama paragraph *procedure*

Kata Kunci: *procedure, sub program, program utama, paragraph procedure*



# 6.1 Pengertian Procedure & Bentuk Umum

- Pengertian *Procedure*

Merupakan *statement* atau *instruksi/perintah* yang digunakan dalam sebuah program dan dideklarasikan sebuah nama paragraph tertentu, tujuan dari *procedure* adalah untuk mengerjakan suatu proses tertentu tanpa mengembalikan hasil proses yang ada dalam sebuah

pl `Void nama Paragraph Procedure`

- Bentuk umum dari *paragraph procedure* adalah:

## Contoh Program dengan Procedure Menggunakan C++

```
//Program 6.1
#include<iostream.h>
void Persegi_Panjang()
{
double p, l;
cout<<"|=====|"<<endl
;
cout<<"Menggunakan Procedure Untuk Mencari Luas &
```

```
cout<<"Jadi Luas Persegi Panjang Adalah   : "<<p*l<<endl;
cout<<"Jadi Keliling Persegi Panjang Adalah:
"<<2*(p+l)<<endl;
cout<<"=====
"<<endl;
cout<<" Terimakasih"           <<endl;
cout<<"=====
"<<endl;
}
void Lingkaran()
{
double phi=3.14;
double r;
cout<<"|=====|"<<endl;
cout<<"| Menggunakan Procedure Untuk Mencari Luas &
keliling Lingkaran | "<<endl;
cout<<"|=====|"
"<<endl;
cout<<"Luas & Keliling Lingkaran"<<endl;
cout<<"Inputkan Jari-Jari Lingkaran: ";
cin>>r;
cout<<"Jadi Luas Lingkaran Adalah   : "<<phi*r*r<<endl;
cout<<"Jadi Keliling Lingkaran Adalah   :
"<<2*(phi*r)<<endl;
cout<<"=====
"<<endl;
cout<<" TerimaKasih"           <<endl;
```

```
cin>>a;
cout<<"Inputkan Nilai Tinggi      : ";
cin>>t;
cout<<"Inputkan Nilai Sisi        : ";
cin>>s;
cout<<"Jadi Luas Segitiga Adalah    : "<<0.5*(a*t)<<endl;
cout<<"Jadi Luas Persegi Adalah     : "<<s*s<<endl;
cout<<"=====
"<<endl;
cout<<"                               Terima Kasih"
<<endl;
cout<<"=====
"<<endl;
}
int main()
{
Persegi_Panjang();
Lingkaran();
Segitiga_Persegi();
return 0;
}
```

323 |

## Keterangan Kode Program C++

1. Menggunakan 3 buah paragraph procedure dengan nama
  - a. void Persegi\_Panjang()
  - b. void Lingkaran()
  - c. void Segitiga\_Persegi()

5. Untuk Rumus menggunakan :

- `cout<<"Jadi Luas Persegi Panjang Adalah : "`  
`<<p*l<<endl;`
- `cout<<"Jadi Keliling Persegi Panjang Adalah: "`  
`<<2*(p+l)<<endl;`
- `cout<<"Jadi Luas Lingkaran Adalah : "`  
`<<phi*r*r<<endl;`
- `cout<<"Jadi Keliling Lingkaran Adalah : "`  
`<<2*(phi*r)<<endl;`
- `cout<<"Jadi Luas Segitiga Adalah : "`  
`<<0.5*(a*t)<<endl;`
- `cout<<"Jadi Luas Persegi Adalah : "`  
`<<s*s<<endl;`

Untuk lebih jelasnya dapat dilihat output berikut ini:

```
(Inactive D:\C...LAT-1\PROCEDURE.EXE)
|-----|
|Menggunakan Procedure Untuk Mencari Luas & keliling Persegi Panjang|
|-----|
Luas & Keliling Persegi Panjang
Inputkan Panjang Persegi Panjang      : 8
Inputkan Lebar Persegi Panjang        : 8
Jadi Luas Persegi Panjang Adalah      : 64
Jadi Keliling Persegi Panjang Adalah  : 32
|-----|
|                          Terima Kasih                          |
|-----|
|Menggunakan Procedure Untuk Mencari Luas & keliling Lingkaran |
|-----|
Luas & Keliling Lingkaran
Inputkan Jari-Jari Lingkaran: 8
Jadi Luas Lingkaran Adalah          : 200.96
Jadi Keliling Lingkaran Adalah      : 50.24
|-----|
|                          Terima Kasih                          |
|-----|
|Menggunakan Procedure Untuk Mencari Luas Segitiga & Persegi |
|-----|
Luas Segitiga & Persegi
Inputkan Nilai Alas                  : 8
Inputkan Nilai Tinggi                : 8
Inputkan Nilai Sisi                  : 8
Jadi Luas Segitiga Adalah            : 32
Jadi Luas Persegi Adalah             : 64
|-----|
|                          Terima Kasih                          |
|-----|
```



5. Untuk Rumus menggunakan :

- `cout<<"Jadi Luas Persegi Panjang Adalah : "`  
`<<p*l<<endl;`
- `cout<<"Jadi Keliling Persegi Panjang Adalah: "`  
`<<2*(p+l)<<endl;`
- `cout<<"Jadi Luas Lingkaran Adalah : "`  
`<<phi*r*r<<endl;`
- `cout<<"Jadi Keliling Lingkaran Adalah : "`  
`<<2*(phi*r)<<endl;`
- `cout<<"Jadi Luas Segitiga Adalah : "`  
`<<0.5*(a*t)<<endl;`
- `cout<<"Jadi Luas Persegi Adalah : "`  
`<<s*s<<endl;`

Untuk lebih jelasnya dapat dilihat output berikut ini:

```
(Inactive) D:\C...LAT-1\PROCEDUR.EXE
|-----|
| Menggunakan Procedure Untuk Mencari Luas & keliling Persegi Panjang |
|-----|
Luas & Keliling Persegi Panjang
Inputkan Panjang Persegi Panjang      : 8
Inputkan Lebar Persegi Panjang        : 8
Jadi Luas Persegi Panjang Adalah      : 64
Jadi Keliling Persegi Panjang Adalah  : 32
|-----|
|                               |
|                               |
|-----|
| Menggunakan Procedure Untuk Mencari Luas & keliling Lingkaran |
|-----|
Luas & Keliling Lingkaran
Inputkan Jari-Jari Lingkaran: 8
Jadi Luas Lingkaran Adalah          : 200.96
Jadi Keliling Lingkaran Adalah      : 50.24
|-----|
|                               |
|                               |
|-----|
| Menggunakan Procedure Untuk Mencari Luas Segitiga & Persegi |
|-----|
Luas Segitiga & Persegi
Inputkan Nilai Alas                  : 8
Inputkan Nilai Tinggi                : 8
Inputkan Nilai Sisi                  : 8
Jadi Luas Segitiga Adalah            : 32
Jadi Luas Persegi Adalah             : 64
|-----|
```

## Contoh Program C# Menggunakan Procedure

```
//Program 6.2
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Prosedure
{
    class Perkalian
    {
        private
        int nilai1, nilai2;
        float hasil;
        public void input(int r, int v)
        {
            nilai1 = r;
            nilai2 = v;
            hasil = nilai1 * nilai2;
        }
        public void hasilperkalian()
        {
            Console.WriteLine(hasil);
        }
    }
    class Perkalian1
    {
        private
        int nilai3, nilai4;
        float hasil1;
        public void input(int w, int y)
        {
            nilai3 = w;
            nilai4 = y;
            hasil1 = (nilai3 * nilai4) / 2;
        }
        public void hasilperkalian1()
        {
            Console.WriteLine(hasil1);
        }
    }
}

class Perkalian2
{
```

```
private
    int nilai5, hasil2;
    public void input(int j)
    {
        nilai5 = j;
        hasil2 = nilai5 * 2 * 22 / 7;
    }
    public void hasilperkalian2()
    {
        Console.WriteLine(hasil2);
    }
}
class Perkalian3
{
    private
        int nilai7, nilai8, nilai9;
        float hasil3;
        public void input(int m, int n, int p)
        {
            nilai7 = m;
            nilai8 = n;
            nilai9 = p;
            hasil3 = nilai7 * nilai8 * nilai9;
        }
        public void hasilperkalian3()
        {
            Console.WriteLine(hasil3);
        }
    }
}

namespace Prosedure
{
    class Program
    {
        public static void Main(string[] args)
        {
            int r, v, w, y, j, m, n, p;
            Perkalian b = new Perkalian();
            Perkalian1 c = new Perkalian1();
            Perkalian2 d = new Perkalian2();
            Perkalian3 e = new Perkalian3();
        }
    }
}
```

```
Console.WriteLine("=====");
    Console.WriteLine(" Menggunakan Procedure Pada
Bahasa C#");

Console.WriteLine("=====");
    Console.WriteLine("Menghitung Luas Persegi ");

Console.WriteLine("=====");
    Console.WriteLine("Inputkan nilai panjang :");
    r = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("Inputkan nilai lebar :");
    v = Convert.ToInt32(Console.ReadLine());
    b.input(r, v);
    Console.WriteLine();
    Console.WriteLine("Hasil Luas Persegi adalah :");
    b.hasilperkalian();
    Console.WriteLine();
    Console.WriteLine("Menghitung Luas Segitiga ");

Console.WriteLine("=====");
    Console.WriteLine("Inputkan nilai alas :");
    w = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("Inputkan nilai tinggi :");
    y = Convert.ToInt32(Console.ReadLine());
    c.input(w, y);
    Console.WriteLine();
    Console.WriteLine("Hasil Luas Segitiga adalah :");
    c.hasilperkalian1();
    Console.WriteLine();
    Console.WriteLine("Menghitung Luas Persegi Panjang ");
    Console.WriteLine("=====");
    Console.WriteLine("Inputkan nilai panjang :");
    r = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("Inputkan nilai lebar :");

    v = Convert.ToInt32(Console.ReadLine());
    b.input(r, v);
    Console.WriteLine();
    Console.WriteLine("Hasil Luas Persegi Panjang adalah:");
    b.hasilperkalian();
    Console.WriteLine();

Console.WriteLine(" Menghitung Luas Lingkaran ");

Console.WriteLine("=====");
```

```
Console.Write("Inputkan nilai jari-jari :");
    j = Convert.ToInt32(Console.ReadLine());
    d.input(j);
    Console.WriteLine();
    Console.Write("Hasil Luas Lingkaran adalah :");
    d.hasilperkalian2();
    Console.WriteLine();
    Console.WriteLine("Menghitung Luas Kubus ");
Console.WriteLine("=====");
    Console.Write("Inputkan nilai panjang :");
    m = Convert.ToInt32(Console.ReadLine());
    Console.Write("Inputkan nilai lebar :");
    n = Convert.ToInt32(Console.ReadLine());
    Console.Write("Inputkan nilai tinggi :");
    p = Convert.ToInt32(Console.ReadLine());
    e.input(m, n, p);
    Console.WriteLine();
    Console.Write("Hasil Luas Kubus adalah :");
    e.hasilperkalian3();
    Console.WriteLine();
    Console.ReadKey(true);

}
}
}
```

## Keterangan Kode Program

Dari Program 6.2 Menggunakan

1. Logika procedure pada bahasa C#, yaitu dengan beberapa *paragraph*, misalnya saja dideklarasikan *paragraph procedure* dengan nama[perkalian, perkalian1 sampai perkalian 3, jadi ada 4 buah *paragraph procedure*], ciri utama dari pendeklarasian *paragraph procedure* adalah `class Perkalian`, `class Perkalian1`, `class Perkalian2`, `class Perkalian3`, contoh pendeklarasian program *procedure* yang dijadikan ciri khusus adalah: `namespace` Prosedure

`namespace` Prosedure

2. Untuk program utamanya memiliki ciri khusus yaitu:

```
namespace Prosedure
{
    class Program
    {
        public static void Main(string[] args)
        {
```

3. Untuk nama *variabel* yang dideklarasikan adalah dilihat dari masing- masing *paragraph procedure* misalnya:

```
class Perkalian
{
    private
    int nilai1, nilai2;
    float hasil;
    public void input(int r, int v)
    {
```

4. Untuk perkalian menggunakan rumus, misalnya diambil 1 buah *paragraph* misalnya *paragraph* perkalian

```
nilai1 = r;
nilai2 = v;
hasil = nilai1 * nilai2;
```

5. Untuk variabel program utama menggunakan

```
public static void Main(string[] args)
{
    int r, v, w, y, j, m, n, p;
```

6. Untuk rumus yang digunakan, langkah selanjutnya hanya memanggil nama *paragraph procedure* saja, seperti berikut ini:

```
Perkalian b = new Perkalian();
Perkalian1 c = new Perkalian1();
Perkalian2 d = new Perkalian2();
Perkalian3 e = new Perkalian3();
```

```
C:\Windows\system32\cmd.exe

=====
Menggunakan Procedure Pada Bahasa C#
=====

      Menghitung Luas Persegi
=====
Inputkan nilai panjang :9
Inputkan nilai lebar   :9

Hasil Luas Persegi adalah :81

      Menghitung Luas Segitiga
=====
Inputkan nilai alas    :9
Inputkan nilai tinggi  :9

Hasil Luas Segitiga adalah :40

      Menghitung Luas Persegi Panjang
=====
Inputkan nilai panjang :9
Inputkan nilai lebar   :9

Hasil Luas Persegi Panjang adalah:81

      Menghitung Luas Lingkaran
=====
Inputkan nilai jari-jari :9

Hasil Luas Lingkaran adalah :56

      Menghitung Luas Kubus
=====
Inputkan nilai panjang :9
Inputkan nilai lebar   :9
Inputkan nilai tinggi  :9

Hasil Luas Kubus adalah :729
```

Gambar 6.2 Hasil Output menggunakan Procedure Dengan Beberapa Nama Paragraph Procedure

## Contoh Program Java Menggunakan Procedure

```
class procedure
{
// Paragraph Procedure dengan nama Perkalian
private static int perkalian (int bill, int bil2)
{
return (bill*bil2);
}
private static double perkalian(double bill,double bil2)
{
return(bill*bil2);
}
public static void main(String[] args)
{
    System.out.println("=====
=====");
    System.out.println("    Menggunakan Procedure Pada
Java    ");
    System.out.println("=====
=====");
int x = 90; int y = 90; int z;
double a = 98.2;
double b = 92.1; double s; z=perkalian(x,y); s=perkalian(a,b);
System.out.println("Maka Hasil Dari Perkalian Pertama Adalah: "+z);
System.out.println("Maka Hasil Dari Perkalian Kedua Adalah : "+s);
System.out.println("=====");
}
```



## Keterangan Kode Program

1. Dari program procedure menggunakan java, ada 2 buah paragraph procedure yang dideklarasikan yaitu :

```
private static int perkalian (int bill, int bil2)
{
return (bill*bil2);
}
private static double perkalian(double bill,double bil2)
{
return(bill*bil2);
```

2. Untuk Program utamanya adalah :

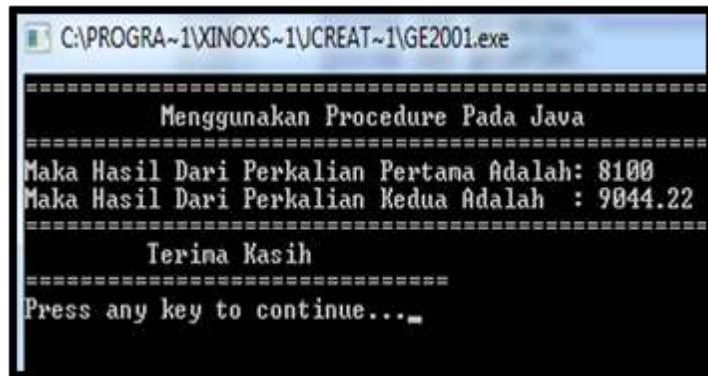
```
public static void main(String[] args)
{
System.out.println("=====");
);
System.out.println("Menggunakan Procedure Pada Java
");
System.out.println("=====");
);
int x = 90; int y = 90; int z;
double a = 98.2;
double b = 92.1; double s; z=perkalian(x,y);
s=perkalian(a,b);
System.out.println("Maka Hasil Dari Perkalian Pertama
Adalah: "+z);
System.out.println("Maka Hasil Dari Perkalian Kedua
```

Yaitu nilai :int x = 90; int y = 90; int z; double a = 98.2;double b = 92.1;

3. Untuk proses pemanggilan statement yang digunakan adalah:

```
double s; z=perkalian(x,y); s=perkalian(a,b);
```

Untuk lebih jelasnya dapat dilihat output berikut ini:



## Pengayaan

1. *Procedure* dalam bahasa pemrograman digunakan untuk
2. Mengapa ada *paragraph procedure* dalam bahasa pemrograman
3. Tuliskan bentuk umum *paragraph procedure*
4. Berikan contoh program menggunakan *paragraph procedure* dengan bahasa C++, C#

# GLOSARIUM

1. Merupakan *statement* atau *instruksi/perintah* yang digunakan dalam sebuah program dan dideklarasikan sebuah nama paragraph tertentu, tujuan dari *procedure* adalah untuk mengerjakan suatu proses tertentu tanpa mengembalikan hasil proses yang ada dalam sebuah program.

2. Bentuk umum dari *paragraph procedure* adalah:

*Void\_nama\_Paragraph\_Procedure*

3. Variabel yang digunakan pada *paragraph procedure* adalah *variable* global, yaitu jenis *variable* yang hanya digunakan pada bagian *procedure* saja

4. Untuk jenis *variable* yang digunakan

## Contoh Program Procedure & Function

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Prosedure
{
    class Perkalian
    {
        private
        int nilai1, nilai2;
        float hasil;
        public void input(int r, int v)
        {
            nilai1 = r;
            nilai2 = v;
            hasil = nilai1 * nilai2;
        }
        public void hasilperkalian()
        {
            Console.WriteLine(hasil);
        }
    }
    class Perkalian1
    {
        private
        int nilai3, nilai4;
        float hasil1;
        public void input(int w, int y)
        {
            nilai3 = w;
            nilai4 = y;
            hasil1 = (nilai3 * nilai4) / 2;
        }
        public void hasilperkalian1()
        {
            Console.WriteLine(hasil1);
        }
    }
}
```

```
}
}
class Perkalian3
{
    private
    int nilai7, nilai8, nilai9;
    float hasil3;
    public void input(int m, int n, int p)
    {
        nilai7 = m;
        nilai8 = n;
        nilai9 = p;
        hasil3 = nilai7 * nilai8 * nilai9;
    }
    public void hasilperkalian3()
    {
        Console.WriteLine(hasil3);
    }
}
}

namespace Prosedure
{
    class Program
    {
        public static void Main(string[] args)
        {
            int r, v, w, y, j, m, n, p;
            Perkalian b = new Perkalian();
            Perkalian1 c = new Perkalian1();
            Perkalian2 d = new Perkalian2();
            Perkalian3 e = new Perkalian3();

            Console.WriteLine("=====");
            Console.WriteLine(" Menggunakan Procedure Pada
            Bahasa C#");

            Console.WriteLine("=====");
            Console.WriteLine("Menghitung Luas
            Persegi ");

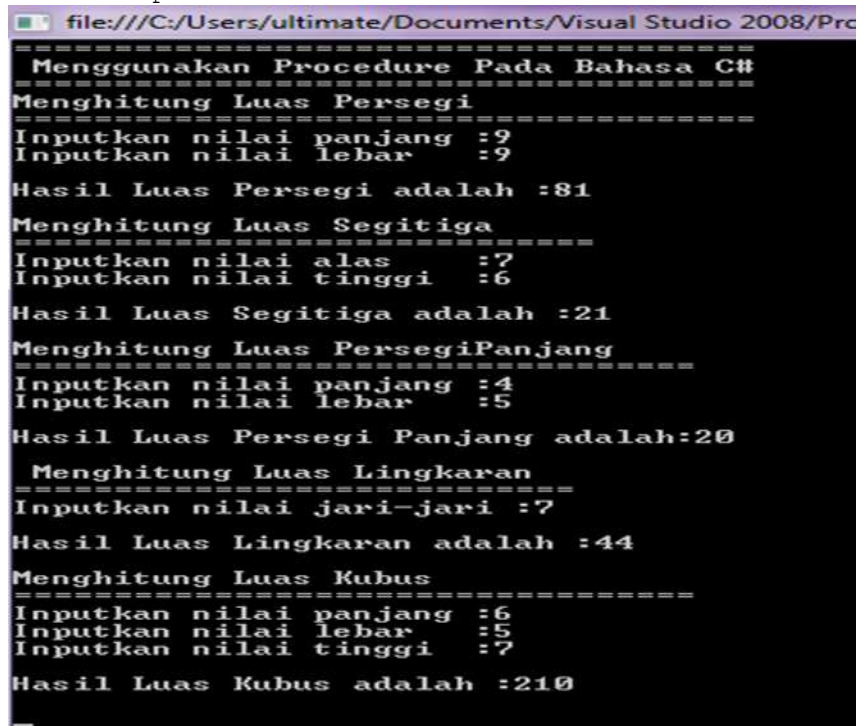
            Console.WriteLine("=====");
            Console.WriteLine("Inputkan nilai panjang
            :");
        }
    }
}
```

```
r = Convert.ToInt32(Console.ReadLine());
Console.Write("Inputkan nilai lebar  :");
v = Convert.ToInt32(Console.ReadLine());
b.input(r, v);
Console.WriteLine();
Console.Write("Hasil Luas Persegi adalah :");
b.hasilperkalian();
Console.WriteLine();
Console.WriteLine("Menghitung Luas Segitiga ");
Console.WriteLine("=====");
Console.Write("Inputkan nilai alas  :");
w = Convert.ToInt32(Console.ReadLine());
Console.Write("Inputkan nilai tinggi  :");
y = Convert.ToInt32(Console.ReadLine());
c.input(w, y);
Console.WriteLine();
Console.Write("Hasil Luas Segitiga adalah :");
c.hasilperkalian1();
Console.WriteLine();
Console.WriteLine("Menghitung Luas Persegi Panjang
");
Console.WriteLine("=====");
Console.Write("Inputkan nilai panjang :");
r = Convert.ToInt32(Console.ReadLine());
Console.Write("Inputkan nilai lebar  :");
v = Convert.ToInt32(Console.ReadLine());
b.input(r, v);
Console.WriteLine();
Console.Write("Hasil Luas Persegi Panjang adalah:");
b.hasilperkalian();
Console.WriteLine();
Console.WriteLine(" Menghitung Luas Lingkaran ");
Console.WriteLine("=====");
Console.Write("Inputkan nilai jari-jari :");
j = Convert.ToInt32(Console.ReadLine());
d.input(j);
Console.WriteLine();
Console.Write("Hasil Luas Lingkaran adalah :");
d.hasilperkalian2();
Console.WriteLine();
```

```
Console.WriteLine("Menghitung Luas Kubus   ");
Console.WriteLine("=====");
Console.Write("Inputkan nilai panjang :");
m = Convert.ToInt32(Console.ReadLine());
Console.Write("Inputkan nilai lebar   :");
n = Convert.ToInt32(Console.ReadLine());
Console.Write("Inputkan nilai tinggi  :");
p = Convert.ToInt32(Console.ReadLine());
e.input(m, n, p);
Console.WriteLine();
Console.Write("Hasil Luas Kubus adalah :");
e.hasilperkalian3();
Console.WriteLine();
Console.ReadKey(true);

    }
}
}
```

Hasil Output :



```
file:///C:/Users/ultimate/Documents/Visual Studio 2008/Pro
=====
Menggunakan Procedure Pada Bahasa C#
=====
Menghitung Luas Persegi
=====
Inputkan nilai panjang :9
Inputkan nilai lebar   :9
Hasil Luas Persegi adalah :81
Menghitung Luas Segitiga
=====
Inputkan nilai alas   :7
Inputkan nilai tinggi :6
Hasil Luas Segitiga adalah :21
Menghitung Luas Persegi Panjang
=====
Inputkan nilai panjang :4
Inputkan nilai lebar   :5
Hasil Luas Persegi Panjang adalah:20
Menghitung Luas Lingkaran
=====
Inputkan nilai jari-jari :7
Hasil Luas Lingkaran adalah :44
Menghitung Luas Kubus
=====
Inputkan nilai panjang :6
Inputkan nilai lebar   :5
Inputkan nilai tinggi  :7
Hasil Luas Kubus adalah :210
=====
```

PENGAYAAN

# contoh-contoh program

## CONTOH PROGRAM Mencari Luas dan Keliling Persegi Panjang

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace PersegiPanjang
{
    class Program
    {
        static void Main(string[] args)
        {
            int p, l, luas, kel;

            Console.WriteLine("=====");
            Console.WriteLine("Hitung Luas dan Keliling Persegi Panjang");
            Console.WriteLine("=====");
            Console.Write("Inputkan Panjang      : ");
            p = Convert.ToInt32(Console.ReadLine());
            Console.Write("Inputkan Lebar      : ");
            l = Convert.ToInt32(Console.ReadLine());
            luas = p * l;
            kel = 2 * (p + l);
            Console.WriteLine("Luas Persegi Panjang Adalah      : " +
                luas);
            Console.WriteLine("Keliling Persegi Panjang Adalah : " + kel);

            Console.WriteLine("=====");
            Console.WriteLine("          Terima Kasih");
            Console.WriteLine("=====");
        }
    }
}
```

Hasil Tampilan saat dijalankan/ Running dengan menekan CTRL+F5



# contoh-contoh program

```
C:\Windows\system32\cmd.exe

=====
Hitung Luas dan Keliling Persegi Panjang
=====
Inputkan Panjang      : 9
Inputkan Lebar        : 9
Luas Persegi Panjang Adalah      : 81
Keliling Persegi Panjang Adalah : 36
=====

                Terima Kasih
=====
Press any key to continue . . .
```

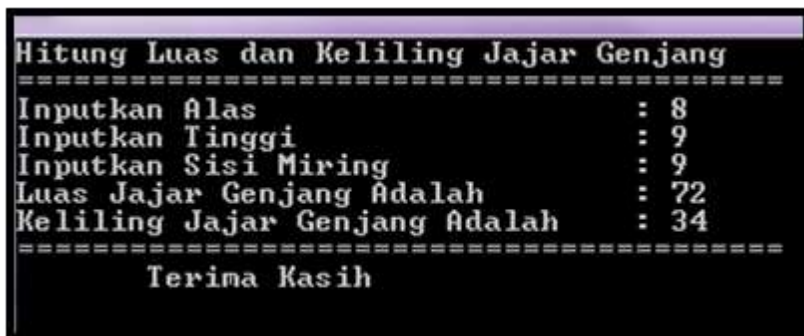
## MENCARI LUAS DAN KELILING JAJARAN GENJANG

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace InputanJajarGenjang
{
    class Program
    {
        static void Main(string[] args)
        {
            int a, t, s, luas, kel;
            Console.WriteLine("Hitung Luas dan Keliling Jajar
Genjang");
            Console.WriteLine("=====");
            Console.Write("Inputkan Alas      : ");
            a = Convert.ToInt32(Console.ReadLine());
            Console.Write("Inputkan Tinggi      : ");
            t = Convert.ToInt32(Console.ReadLine());
            Console.Write("Inputkan Sisi Miring      : ");
            s = Convert.ToInt32(Console.ReadLine());
            luas = a * t;
            kel = 2 * (a + s);
```

```
Console.WriteLine("Luas Jajar Genjang Adalah   : " + luas);  
Console.WriteLine("Keliling Jajar Genjang Adalah   : " +  
kel);  
Console.WriteLine("=====");  
Console.WriteLine("                Terima Kasih");  
Console.ReadKey();  
  
    }  
}  
}
```

Hasil Tampilan saat dijalankan/ Running dengan menekan CTRL+F5



```
Hitung Luas dan Keliling Jajar Genjang  
=====  
Inputkan Alas           : 8  
Inputkan Tinggi         : 9  
Inputkan Sisi Miring    : 9  
Luas Jajar Genjang Adalah : 72  
Keliling Jajar Genjang Adalah : 34  
=====  
                Terima Kasih
```

### CONTOH PROGRAM Mencari Luas Permukaan Volume TABUNG

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
  
namespace ConsoleApplication35  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            double r, t, luas, vol;  
            Console.WriteLine("Hitung Luas dan Volume Tabung");  
            Console.WriteLine("=====");  
            Console.Write("Inputkan Jari-Jari           : ");  
            r = Convert.ToInt32(Console.ReadLine());  
            Console.Write("Inputkan Tinggi           : ");  
            t = Convert.ToInt32(Console.ReadLine());
```

```
luas = 3.14 * r * r;  
vol = luas * t;  
Console.WriteLine("Luas Permukaan Tabung Adalah      : " +  
luas);  
Console.WriteLine("Volume Tabung Adalah              : " +  
vol);  
  
Console.WriteLine("=====");  
Console.WriteLine("                Terima Kasih");  
Console.ReadKey();  
    }  
}
```

Hasil Tampilan saat dijalankan/ Running dengan menekan CTRL+F5



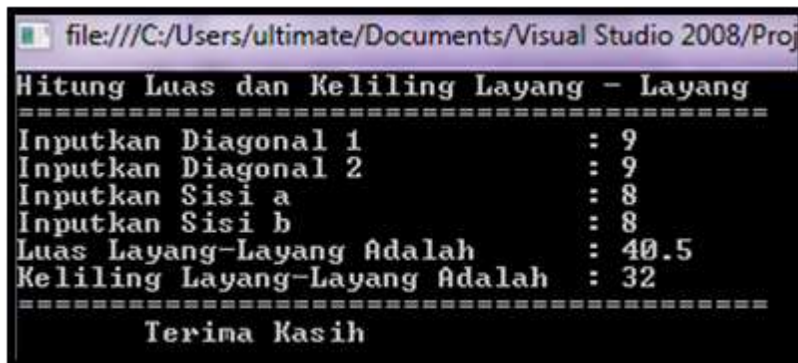
```
file:///C:/Users/ultimate/Documents/Visual Studio 2008/Projects/  
Hitung Luas dan Volume Tabung  
=====  
Inputkan Jari-Jari      : 9  
Inputkan Tinggi        : 9  
Luas Permukaan Tabung Adalah : 254.34  
Volume Tabung Adalah   : 2289.06  
=====  
Terima Kasih
```

### CONTOH PROGRAM Mencari Luas & Keliling LAYANG-LAYANG

```
double d1, d2, sa, sb, luas, kel;  
Console.WriteLine("Hitung Luas dan Keliling Layang - Layang");  
Console.WriteLine("=====");  
Console.Write("Inputkan Diagonal 1      : ");  
d1 = Convert.ToInt32(Console.ReadLine());  
Console.Write("Inputkan Diagonal 2      : ");  
d2 = Convert.ToInt32(Console.ReadLine());  
Console.Write("Inputkan Sisi a      : ");  
sa = Convert.ToInt32(Console.ReadLine());  
Console.Write("Inputkan Sisi b      : ");  
sb = Convert.ToInt32(Console.ReadLine());  
luas = 0.5 * d1 * d2;  
kel = 2 * (sa + sb);  
Console.WriteLine("Luas Layang-Layang Adalah      : " + luas);
```

```
Console.WriteLine("Keliling Layang-Layang Adalah : " + kel);  
Console.WriteLine("=====  
Console.WriteLine("          Terima Kasih");  
Console.ReadKey();  
  
    }  
}  
}
```

Hasil Tampilan saat dijalankan/ Running dengan menekan CTRL+F5



```
file:///C:/Users/ultimate/Documents/Visual Studio 2008/Proj  
Hitung Luas dan Keliling Layang - Layang  
=====  
Inputkan Diagonal 1 : 9  
Inputkan Diagonal 2 : 9  
Inputkan Sisi a : 8  
Inputkan Sisi b : 8  
Luas Layang-Layang Adalah : 40.5  
Keliling Layang-Layang Adalah : 32  
=====  
Terima Kasih
```

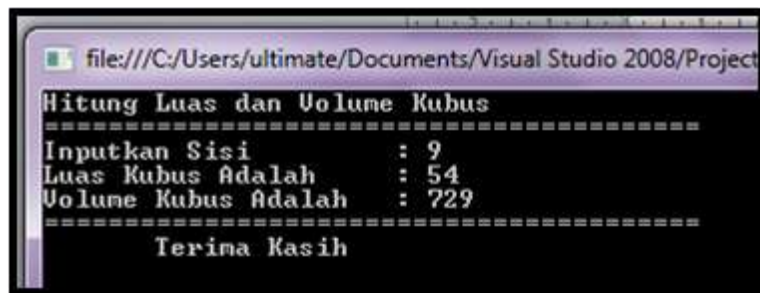
### CONTOH PROGRAM Mencari Luas & Volume Kubus

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
namespace ConsoleApplication37  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            int s, luas, vol;  
            Console.WriteLine("Hitung Luas dan Volume Kubus");  
            Console.WriteLine("=====  
            Console.Write("Inputkan Sisi : ");  
            s = Convert.ToInt32(Console.ReadLine());  
            luas = 6 * s;  
            vol = s * s * s;  
            Console.WriteLine("Luas Kubus Adalah : " + luas);  
            Console.WriteLine("Volume Kubus Adalah : " + vol);  
            Console.WriteLine("=====  
            Console.WriteLine("          Terima Kasih");
```

```
Console.ReadKey();
```

```
    }  
    }  
}
```

Hasil Tampilan saat dijalankan/ Running dengan menekan CTRL+F5



### CONTOH PROGRAM MEMBUAT MENU

```

C:\Windows\system32\cmd.exe
-----
          DAFTAR MENU RESTORAN BUNDA FAUZIAH KELAPA DUA, DEPOK
-----
Nama Pelanggan yang berkunjung :Tsarwan Kasyfillah Nasution
ID Pelanggan yang berkunjung   :01
Alamat Pelanggan berkunjung    :Kelapa Dua, Depok
No.HP Pelanggan berkunjung     :081383955321

          MAKANAN DAN MINUMAN YANG DI SEDIAKAN
-----
          MAKANAN
-----
          JENIS      : HARGA
-----
A. Bakar
1.Ayam Bakar      : Rp.9 .000
2.Bebek Bakar    : Rp.10.000
3.Jagung Bakar   : Rp.7 .000
B. Goreng
1.Ikan Goreng    : Rp.8 .000
2.Ayam Goreng    : Rp.5 .000
3.Nasi Goreng    : RP.12.000
C. Rebus
1.Mie Rebus      : Rp.9 .000
2.Ikan Rebus    : Rp.7 .000
3.Jagung Rebus  : Rp.4. 000
-----
          MINUMAN
-----
          JENIS      : HARGA
-----
A. SOFTDRINK
1.Panta          : Rp.4 .000
2.Coca-Cola     : Rp.4 .000
3.Sprite        : Rp.4 .000
B. MINUMAN HANGAT
1.Kopi          : Rp.5 .000
2.Teh           : Rp.4 .000
3.Susu         : Rp.3 .000
C. JUS
1.Jus Melon     : Rp.8 .000
2.Jus Pepaya   : Rp.7 .000
3.Jus Jeruk    : Rp.8 .000
-----
Inputkan Nama Makanan yang ingin anda pesan      : ikan goreng
Inputkan jumlah Makanan Yang Ingin anda pesan    : 1
-----
Inputkan Nama Minuman yang anda pesan            : teh
Inputkan jumlah Minuman yang anda pesan         : 1
-----
Total Biaya Makanan                             : Rp. 8000,-
Total Biaya Minuman                             : Rp. 4000,-
-----
Jumlah Seluruh pesanan                          : Rp. 12000,-
pph                                              : Rp. 600,-
Jumlah Seluruh pesanan setelah pph              : Rp. 12600,-
-----
Press any key to continue . . . -
    
```

