

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Tuna netra adalah mereka yang sama sekali tidak memiliki kemampuan penglihatan (buta *total/total blind*) dan (kurang awas/ *low vision*) yakni mereka yang masih memiliki kemampuan penglihatan namun tidak dapat membaca tulisan dengan ukuran 12 point dalam cahaya normal dan jarak normal meskipun dengan bantuan kacamata. Berdasarkan data pada Sistem Informasi Management Penyandang Disabilitas Kementerian Sosial RI, pada saat ini diketahui jumlah penyandang disabilitas di Indonesia sebanyak 222.179 jiwa, dari jumlah tersebut terdapat penyandang tuna netra sebanyak 18.173 orang dengan rincian sebanyak 6.811 orang dengan kategori *low vision* dan sebanyak 11.362 orang dalam kategori *total blind* [1]. Sedangkan pada tahun 2020 WHO merilis setidaknya terjadi peningkatan sebesar 42,8% penyandang tunanetra sepanjang tahun 1990 sampai dengan tahun 2020, yang semula sebanyak 34,4 juta orang menjadi 49,1 juta orang.

Perkembangan teknologi yang begitu pesat, dapat menunjang dan membantu dalam berbagai aspek kehidupan, kemajuan teknologi saat ini sudah selayaknya dimanfaatkan untuk membantu penyandang tunanetra agar dapat lebih mandiri dalam menjalani kehidupan sehari-hari. Meskipun presentase jumlah penyandang tunanetra di Indonesia hanya berada pada kisaran 1% dari total jumlah penduduk di Indonesia (estimasi Kementerian Kesehatan RI), namun bukan berarti para penyandang disabilitas netra tersebut tidak berhak mendapat kehidupan yang lebih layak.

Meningkatnya angka penyandang tunanetra itu membuat banyak peneliti yang telah membuat *prototype* alat bantu bagi tunanetra Wakyu Dwiyono dkk [3] menggunakan 3 sensor ultrasonik untuk mendeteksi penghalang bagian depan, kanan, dan kiri dengan *error* sebesar 2,5%. Sementara itu A. Junfithrana dkk [4] selain menggunakan sensor ultrasonik sebagai pendeteksi halangan dan motor dc sebagai output berupa getaran apabila terdeteksi penghalang dengan jarak 50cm, rentang *error* pada pengukuran jarak oleh sensor ultrasonik sebesar 0,2 – 5%. Y. Triarnowo E dkk [5] menggunakan sarung tangan sebagai alat bantu navigasi dengan sensor ultrasonik HC-SR04 dan Ping Parallax, namun alat ini hanya mendeteksi adanya penghalang di sekeliling pengguna, tidak termasuk objek seperti lubang yang juga dapat membahayakan pengguna. Umpan balik dari pembacaan sensor ultrasonik berasal dari motor vibrator yang intensitasnya berbanding terbalik dengan jarak halangan atau objek. V.

Fergiyawan dkk [2] merancang alat bantu agar penyandang tunanetra tidak menabrak orang atau benda disekitarnya. Sistem memberi informasi suara dari *buzzer* serta getaran motor servo sebagai indikator apabila terdeteksi halangan oleh sensor ultrasonik. Tetapi pada penelitian ini hanya membaca penghalang sejajar garis lurus dan belum dapat mendeteksi adanya lubang atau galian di bagian bawah yang dapat membahayakan pengguna. F. Nova [6] membuat sistem berupa jam tangan yang dikontrol menggunakan Arduino Promini328 serta sensor ultrasonik sebagai pendeteksi halangan pada bagian depan dan lubang di bagian bawah. Umpan balik apabila terdeteksi halangan oleh sensor ultrasonik berupa suara yang dikeluarkan oleh raspberry pi.

Berdasarkan latar belakang tersebut, diketahui alat bantu tunanetra yang ada sejauh ini sebatas alat yang dapat mendeteksi penghalang, namun belum ada alat yang dapat memandu pengguna untuk berpindah dari satu tempat ke tempat lain menggunakan *smartphone* android.

## 1.2 Rumusan Masalah

Berdasarkan uraian latar belakang tersebut saat ini baru tersedia sistem yang dapat mendeteksi penghalang bagi tunanetra, namun sistem tersebut belum dilengkapi dengan kemampuan untuk dapat menavigasi tunanetra dari suatu tempat ke tempat tujuannya. Oleh karena itu dibuatlah alat yang dapat mendeteksi penghalang sekaligus menavigasi ke tempat tujuan yang terintegrasi pada *smartphone* android miliknya

## 1.3 Tujuan

Tujuan utama dalam penelitian ini adalah untuk membuat sebuah sistem yang dapat membantu mobilitas penyandang tunanetra. Dengan demikian para penyandang tuna netra dapat berpindah dari satu tempat ke tempat lainnya dengan aman dan mandiri melalui sebuah sistem yang tidak hanya dapat mendeteksi adanya halangan di bagian depan, samping kanan, kiri, dan bagian bawah tetapi juga dapat memberi navigasi suara melalui Google Maps.

## 1.4 Batasan Masalah

Penelitian ini dibatasi pada hal-hal berikut ini :

- a. Navigasi di-*setting* hanya untuk ke tempat-tempat tertentu (rumah, toko, pos polisi, taman) dan disimpan pada *database server*.

- b. Perubahan tempat yang telah di-*setting* pada tombol hanya dapat diubah oleh admin dan wali.
- c. Rancang bangun berbasis mikrokontroller Arduino.
- d. Alat hanya terintegrasi pada *Smartphone* Android.

## 1.5 Metode Penyelesaian Masalah

Penelitian ini menggunakan metode penyelesaian sebagai berikut:

a. Studi Pustaka

Dilakukan dengan mengumpulkan referensi melalui buku, jurnal, artikel, maupun situs yang berhubungan dengan materi tugas akhir. Hal ini dilakukan supaya mendapat pembandingan serta acuan dalam menganalisis rancang bangun alat bantu navigasi tunanetra yang sesuai dengan kondisi lapangan.

b. Perancangan *Hardware*

Merancang sistem dalam bentuk diagram blok yang bertujuan sebagai gambaran cara kerja sistem.

c. Perancangan *Software*

Perancangan sistem dilakukan dengan *software* Arduino IDE untuk mengontrol Arduino, *software* Xampp dan phpMySql untuk membuat *database* dan *website host*, serta Android Studio dalam untuk membuat aplikasi android.

d. Uji Coba dan Pengujian

Langkah uji coba dan pengujian dilakukan bertujuan untuk mengetahui apakah alat telah bekerja sebagaimana mestinya sesuai rancangan.

e. Analisis

Analisis data sehingga mengetahui bagaimana alat bekerja pada kondisi tertentu.

f. Penulisan Laporan

Tahap akhir sebagai bentuk hasil dan pertanggungjawaban dari penelitian yang telah dilakukan.

## **BAB 2**

### **LANDASAN TEORI**

Pada bab ini akan diuraikan konsep dasar yang berkaitan serta menunjang pembahasan judul tugas akhir. Adapun teori-teori dasar pada penelitian ini adalah sebagai berikut:

#### **2.1. Kajian Pustaka**

Beberapa penelitian serupa yang sudah dilakukan akan dijadikan *referensi* dalam penelitian ini. Adapun beberapa penelitian serupa yang telah dilakukan adalah sebagai berikut :

Pada artikel “Alat Bantu Navigasi Penyandang Tuna Netra Menggunakan Sensor Ping dan Buzzer” menyajikan alat bantu navigasi tunanetra, menggunakan mikrokontroler dan sensor jarak, *voice database*, RFID reader, dan RF *wireless link*, yang dapat memberikan informasi suara berupa nama tempat dan jarak. Pada penelitian ini, sistem pengukur jarak berfungsi dengan presentasi eror rata-rata 2,5% dalam mengukur objek yang menghalangi pengguna [3].

Artikel “Rancang Bangun Alat Bantu Jalan Untuk Penyandang Tuna Netra Berbasis Arduino” mengembangkan konsep alat bantu jalan dengan tongkat yang memanfaatkan mikrokontroler, arduino, sensor ultrasonik, dan motor dc. Hasil penelitian menunjukkan alat tersebut mampu mendeteksi jarak maksimum 310 cm dengan margin error 0.2 - 5% [4].

Pada penelitian “Pengembangan Sarung Tangan Sonar Sebagai Alternatif Alat Bantu Navigasi Tunanetra Menggunakan Arduino” merancang sebuah sarung tangan yang menggunakan sensor ultrasonik dengan mikrokontroler Arduino yang dapat mendeteksi objek diam maupun bergerak di sekitar pengguna, dengan jarak efektif 5cm – 1m. *Feedback* berupa getaran yang berasal dari motor vibrator yang intensitasnya berbanding terbalik dengan jarak halangan/objek. Hasil rata-rata akurasi sebesar 97,98% untuk sensor HC SR-04 dan 97.6% untuk sensor PING Parallax [5].

Dalam artikel yang berjudul “Alat Pemandu Jalan Untuk Penyandang Tunanetra Menggunakan Sensor Ultrasonic Berbasis Arduino”, menyebutkan sistemnya dirancang berbasis Arduino dan menggunakan sensor ultrasonik HC-SR04 yang dapat membantu penyandang tunanetra dalam melakukan aktifitas sehari-hari agar tidak menabrak orang atau benda disekitarnya. Sistem ini memberikan informasi suara dari buzzer dan getaran yang

berasal dari motor servo. Dalam penelitiannya sistem berjalan dengan lancar, menggunakan led sebagai indikator, dengan hasil pengujian sistem menunjukkan alat dapat bekerja dengan baik dari variasi 7 jarak sensor dan 7 skenario pengujian [2].

Pada penelitian “Mata Ketiga Untuk Tuna Netra Menggunakan Sensor Ultrasonik dan Arduino Pro mini328” dalam penelitiannya menggunakan sensor ultrasonik sebagai pemancar dan menghitung jarak dengan perbedaan selisih waktu, kemudian data pengolah data yang digunakan yakni mikrokontroler pro mini328 dan node mcu esp8266 dan keluaran berupa suara yang dikeluarkan oleh raspberry pi yang diletakkan pada saku pengguna dan dihubungkan dengan headset untuk memberi peringatan ketika sensor ultrasonik aktif [6].

## 2.2. Landasan Teori

Berikut merupakan beberapa hal yang menjadi landasan teori Rancang Bangun Alat Bantu Navigasi Tunanetra Berbasis Mikrokontroler Arduino Terintegrasi *Smartphone* Android.

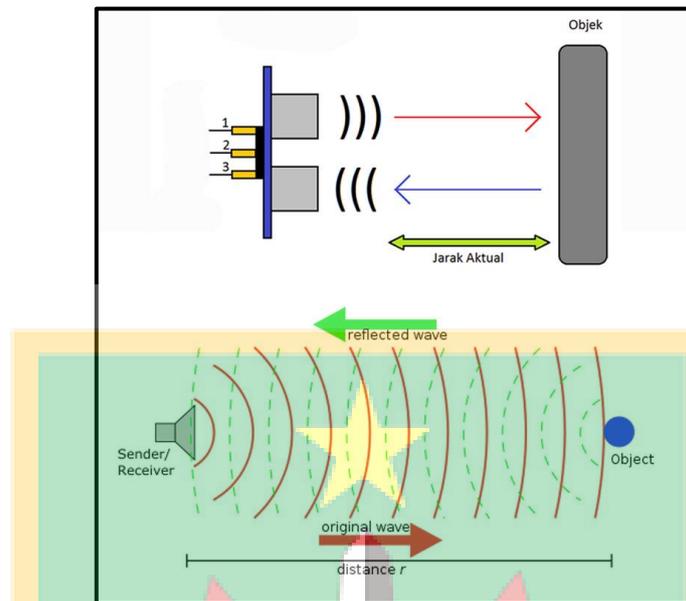
### 2.2.1 Sensor Ultrasonik

Sensor ultrasonik merupakan sensor yang dapat mengkonversi besaran fisis berupa bunyi ke besaran listrik. Prinsip kerjanya adalah dengan memancarkan gelombang ultrasonik atau suara, kemudian memantulkannya kembali apabila terdapat benda atau halangan sehingga dapat ditafsirkan untuk menentukan jarak suatu benda dengan frekuensi tertentu. Perhitungan jarak dilakukan dengan menghitung selisih antara kecepatan gelombang dan waktu perjalanan gelombang. Selain mendeteksi atau menghitung jarak benda, sensor ultrasonik juga dapat mendeteksi jenis, model, maupun keretakan benda yang memantulkan gelombang suara.

Frekuensi gelombang suara yang digunakan oleh sensor ultrasonik untuk menghitung jarak sebuah benda sebesar 20.000 Hz. Frekuensi gelombang ultrasonik yang besar ini tidak dapat didengar oleh telinga manusia. Frekuensi tersebut dapat didengar oleh kelelawar, lumba-lumba, anjing dan kucing. Gelombang ultrasonik dapat merambat pada zat cair, gas dan padat. Berdasarkan prinsip kerjanya yang memancarkan dan memantulkan gelombang ultrasonik atau suara. Sensor ultrasonik terdiri dari *transmitter* dan *receiver*.

Terdapat sebuah alat pada sensor ultrasonik bernama piezoelektrik yang berfungsi untuk membangkitkan gelombang ultrasonik dengan frekuensi tertentu. Piezoelektrik akan menghasilkan gelombang ultrasonik saat diberikan tegangan listrik. Gelombang ultrasonik

yang dihasilkan oleh piezoelektrik ini sekitar 40 kHz, secara bersamaan osilator diterapkan pada benda tersebut. Secara detail cara kerja sensor ultrasonik dapat dilihat pada Gambar 2.1



**Gambar 2.1** Cara Kerja Sensor Ultrasonik

Secara lebih rinci, cara kerja sensor ultrasonik dimulai saat sinyal dengan frekuensi di atas 20 kHz dipancarkan oleh *transmitter* ultrasonik pada frekuensi serta durasi tertentu. Dalam mengukur jarak, umumnya frekuensi yang digunakan sebesar 40 kHz. Sinyal yang dipancarkan oleh *transmitter* merambat pada kecepatan kurang lebih 340 m/s sebagai gelombang bunyi. Ketika gelombang bunyi menabrak suatu benda, sinyal kemudian akan dipantulkan oleh benda tersebut. Gelombang pantulan lalu diterima oleh *receiver* dan diproses perhitungan jarak benda menggunakan rumus sebagai berikut :

$$S = \frac{340 \cdot t}{2} \dots\dots\dots(2.1)$$

Dengan S adalah jarak sensor ultrasonik dengan benda, sedangkan t merupakan selisih waktu pancaran gelombang oleh *transmitter* dan waktu ketika gelombang pantul diterima oleh *receiver*.

### 2.2.2.1 Sensor Ultrasonik HY-SRF05

Sensor Ultrasonik HY-SRF05 merupakan salah satu jenis modul sensor ultrasonik yang bekerja dengan prinsip sistem sonar dan radar sama seperti sensor ultrasonik

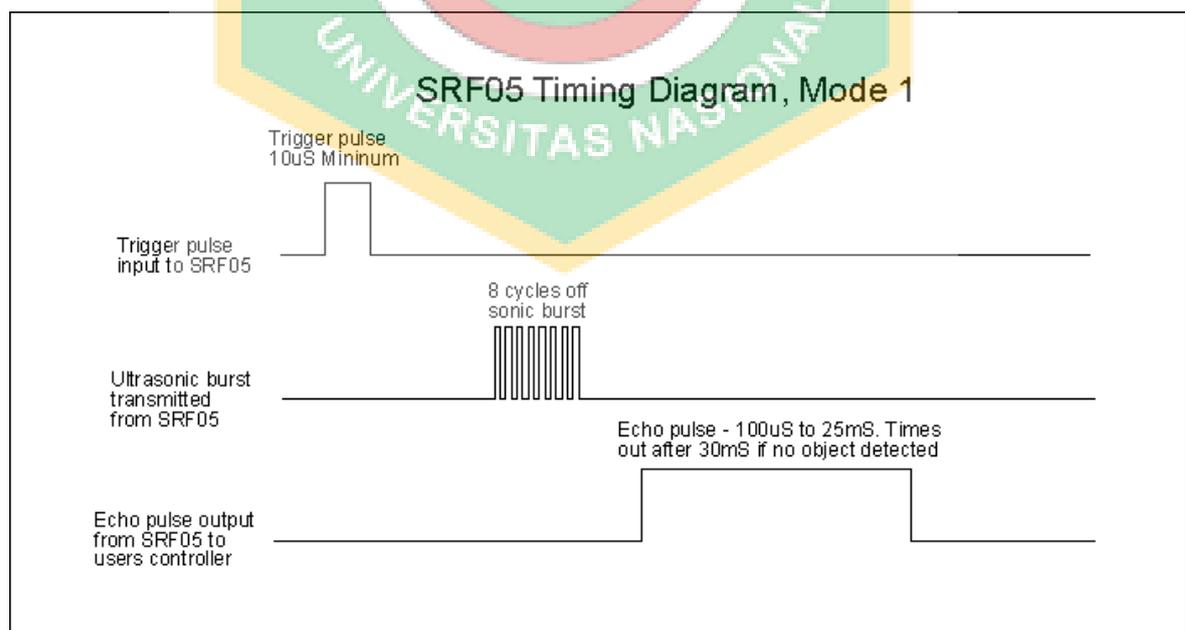
lainnya. Merupakan evolusi dari HC-SR04 dirancang untuk meningkatkan fleksibilitas, meningkatkan jangkauan, serta mengurangi biaya. Rentangnya meningkat dari 3 meter menjadi 4,5 meter. Adapun Spesifikasi dari Modul Sensor Ultrasonik HY-SRF05 dapat dilihat pada Tabel 2.1.

**Tabel 2.1** Spresifikasi Modul HY-SRF05

Keterangan	Spesifikasi
Tegangan Kerja	5V DC
Konsumsi Arus	< 2mA
Tingkat Output	High-5V
Tingkat Output	Akhir 0V
Angle Sensor	≤ 15°
Deteksi Jarak	2 cm - 450cm
Tinggi Presisi	Sampai dengan 0.2cm
Koneksi	VCC, Trig (Trigger/Kontrol Picuan), Echo (menerima end), GND
Tegangan	5V

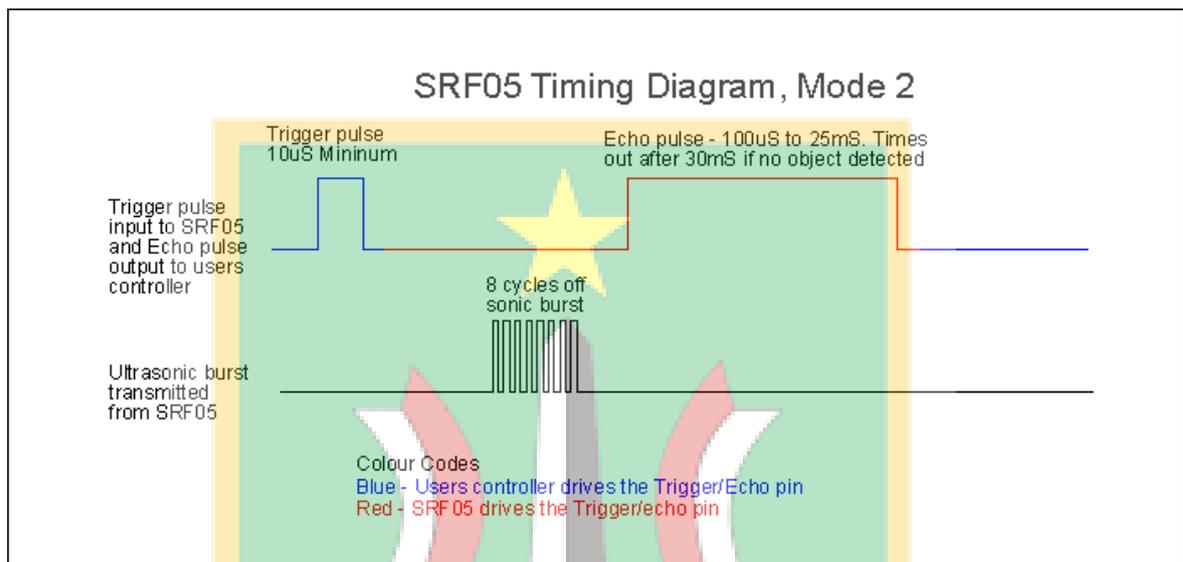
Mode operasi dari sensor ultrasonic HY-SRF05 ini menggunakan mode operasi baru dengan cara mengikat pin mode ke *ground*. Hal ini memungkinkan SRF05 memakai satu pin sebaga pemicu dan gema, sehingga dapat menghemat pin pada mikrokontroler. Saat pin mode tidak terhubung, SRF05 beroperasi dengan pin pemicu dan gema yang terpisah seperti HC-SR04[7].

Berikut ini merupakan Gambar *Timing Diagram* Mode 1 dan Mode 2 pada Modul Sensor Ultrasonik HY-SR05 dapat dilihat pada Gambar 2.2 dan Gambar 2.3.



**Gambar 2.2** *Timing Diagram* HY-SRF05 Mode 1

Pada mode 1 ini kompatibel dengan HC-SRF04 dimana pemicu dan gema terpisah. Mode ini merupakan mode yang paling sederhana untuk digunakan. Semua *library* kode HC-SR04 akan bekerja untuk HY-SRF05 dalam mode ini. Dalam menggunakan mode 1 ini, pin mode dapat dibiarkan tidak terhubung. Pada HY-SRF05 ini memiliki resistor *pull up* internal pada pin ini.



**Gambar 2.3** Timing Diagram HY-SRF05 Mode 2

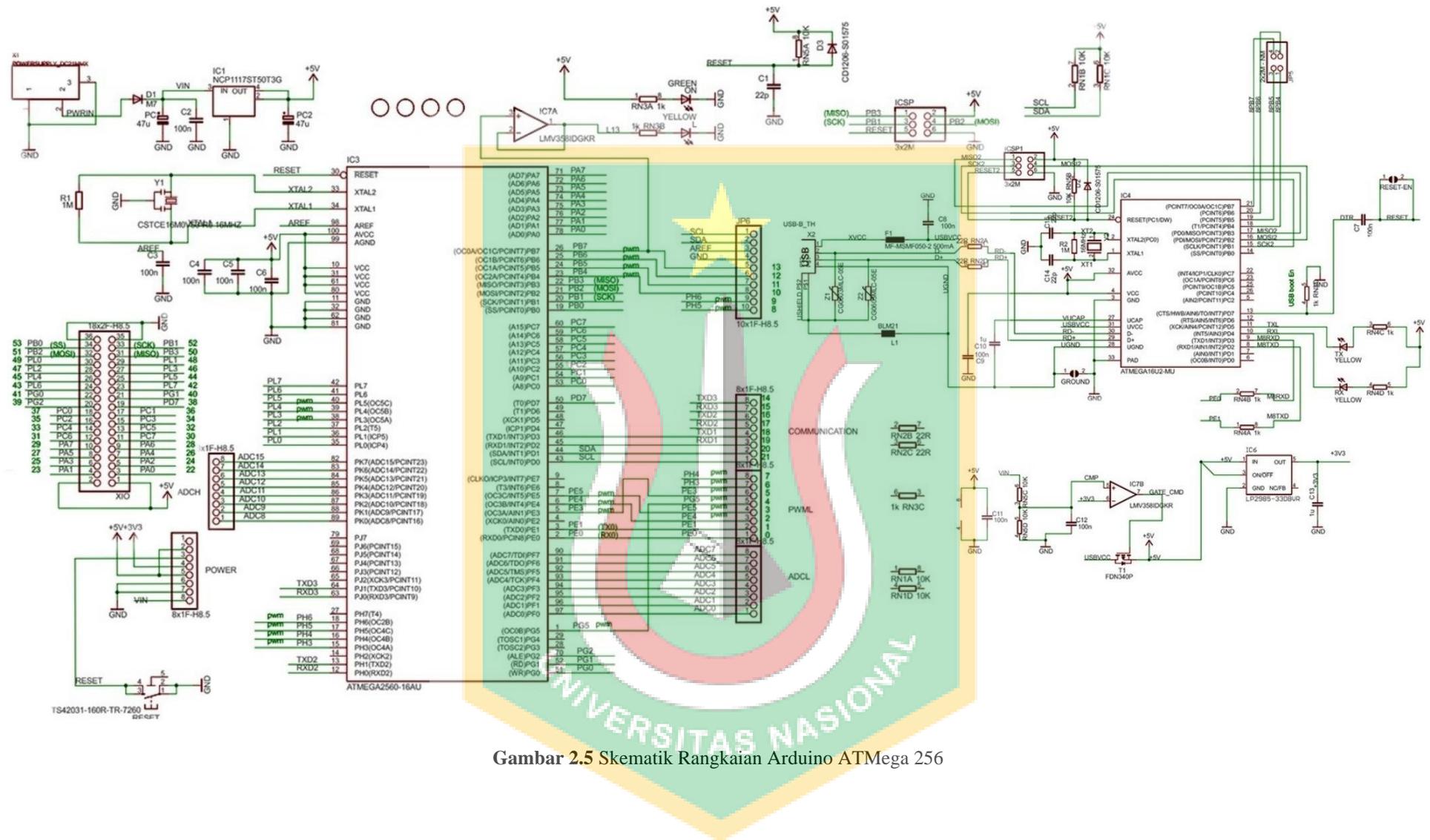
Pada Mode 2 ini, menggunakan satu buah pin untuk sinyal *trigger* dan *echo*. Untuk dapat menggunakan mode ini dapat dengan cara menyambungkan pin mode ke pin *ground* 0V. Sinyal *echo* akan muncul pada pin yang sama dengan sinyal *trigger*. HY-SRF05 tidak akan menaikkan level sinyal *echo* hingga 700 µs setelah sinyal pemicu berakhir. Berikut ini merupakan rangkaian skematik dari sensor ultrasonik HY-SRF05 dapat dilihat pada Gambar 2.4.



(*Pulse with Modulation*, 16 pin lainnya sebagai dapat digunakan sebagai *input* analog, osilator 16 MHz, USB *port*, *header ICSP*, *POWER JACK*, dan tombol reset. Arduino ini diatur melalui sebuah kode, program atau instruksi yang ditulis dalam Bahasa pemrograman pada Arduino IDE. Arduino berbeda dari Bahasa pemrograman lain karena mudah dipelajari, menggunakan bahasan pemrograman C++ dan didukung oleh *library* yang lengkap serta mudah untuk dipelajari sehingga kita tidak hanya dapat melakukan pemrograman tetapi juga melakukan kompilasi program, *debugging* dan proses download ke Arduino *boardnya* hanya dengan sekali klik, maka program yang kita buat dapat langsung *ter-compile* serta terdownload ke mikrokontroler yang ada di *board* Arduino [8].

Berikut ini merupakan gambar rangkaian skematik atau sirkuit diagram dari Arduino ATmega 2560 dapat dilihat pada Gambar 2.5 Sedangkan untuk spesifikasi Arduino ATmega 2560 dapat dilihat pada Tabel 2.2





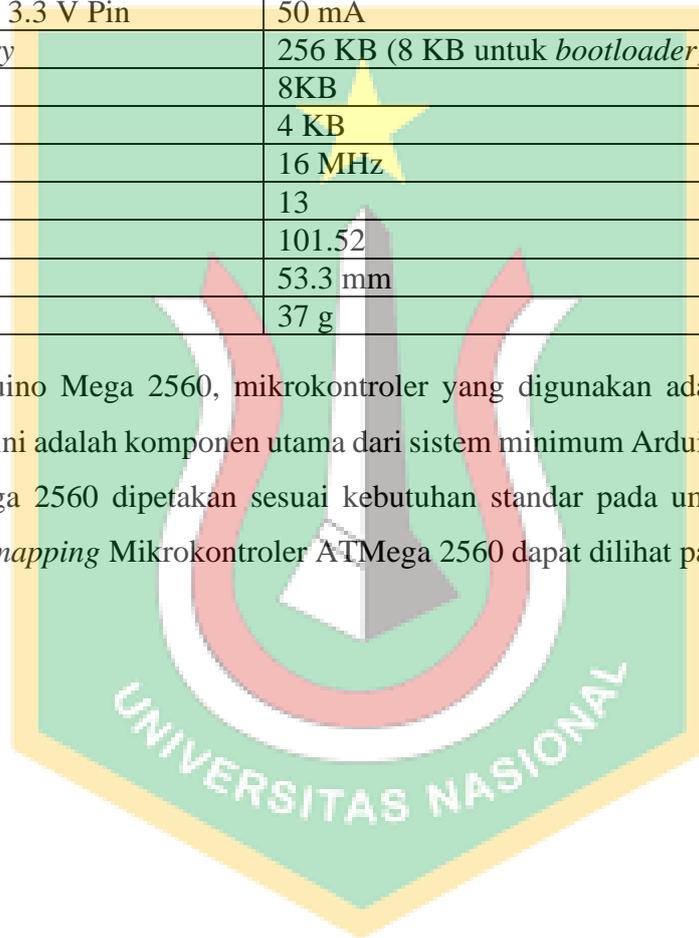
Gambar 2.5 Skematik Rangkaian Arduino ATmega 256

Spesifikasi dari Arduino ATMega 2560 dapat dilihat pada Tabel 2.2.

**Tabel 2. 2** Spesifikasi Arduino ATMega 2560

<b>Keterangan</b>	<b>Spesifikasi</b>
Mikrokontroler	ATMega2560
Tegangan Operasi	5 Volt
Tegangan Input Rekomendasi	7-12 Volt
Batasan Tegangan Input	6-20 Volt
Pin I/O Digital	54
Pin Input Analog	16
Arus DC Per I/O Pin	20 mA
Arus DC untuk 3.3 V Pin	50 mA
<i>Flash Memmory</i>	256 KB (8 KB untuk <i>bootloader</i> )
SRAM	8KB
EEPROM	4 KB
<i>Clock Speed</i>	16 MHz
<i>Led_Builtin</i>	13
Panjang	101.52
Lebar	53.3 mm
Berat	37 g

Pada Arduino Mega 2560, mikrokontroler yang digunakan adalah ATMega 2560. Mikrokontroler ini adalah komponen utama dari sistem minimum Arduino Mega 2560. Pada tiap pin ATMega 2560 dipetakan sesuai kebutuhan standar pada umumnya. Berikut ini merupakan pin *mapping* Mikrokontroler ATMega 2560 dapat dilihat pada Gambar 2.6.





Tabel 2.3 Deskripsi tiap pin ATmega 2560

<i>Pin Number</i>	<i>Pin Name</i>	<i>Mapped Pin Name</i>	<i>Pin Number</i>	<i>Pin Name</i>	<i>Mapped Pin Name</i>
1	PG5 (OC0B)	Digital pin 4 (PWM)	38	PL3 (OC5A)	Digital pin 46 (PWM)
2	PE0 (RXD0/PCINT8)	Digital pin 0 (RX0)	39	PL4 (OC5B)	Digital pin 45 (PWM)
3	PE1 (TXD0)	Digital pin 1 (TX0)	40	PL5 (OC5C)	Digital pin 44 (PWM)
4	PE2 (XCK0/AIN0)		41	PL6	Digital pin 43
5	PE3 (OC3A/AIN1)	Digital pin 5 (PWM)	42	PL7	Digital pin 42
6	PE4 (OC3B/AINT4)	Digital pin 2 (PWM)	43	PD0 (SCL/INT0)	Digital pin 21 (SCL)
7	PE5 (OC3C/INT5)	Digital pin 3 (PWM)	44	PD1 (SDA/INT1)	Digital pin 20 (SDA)
8	PE6 (T3/INT6)		45	PD2 (RXDI/INT2)	Digital pin 19 (RX1)
9	PE7 (CLKO/ICP3/INT7)		46	PD3 (TXDI/INT3)	Digital pin 18 (TX1)
10	VCC	VCC	47	PD4 (ICP1)	
11	GND	GND	48	PD5 (XCK1)	
12	PH0 (RXD2)	Digital pin 17 (RX2)	49	PD6 (T1)	
13	PH1 (TXD2)	Digital pin 16 (TX2)	50	PD7 (T0)	Digital pin 38
14	PH2 (XCK2)		51	PG0 (WR)	Digital pin 41
15	PH3 (OC4A)	Digital pin 6 (PWM)	52	PG1 (RD)	Digital pin 40
16	PH4 (OC4B)	Digital pin 7 (PWM)	53	PC0 (A8)	Digital pin 37
17	PH5 (OC4C)	Digital pin 8 (PWM)	54	PC1 (A9)	Digital pin 36
18	PH6 (OC2B)	Digital pin 9 (PWM)	55	PC2 (A10)	Digital pin 35
19	PB0 (SS/PCINT0)	Digital pin 53 (SS)	56	PC3 (A11)	Digital pin 34
20	PB1 (SCK/PCINT1)	Digital pin 52 (SCK)	57	PC4 (A12)	Digital pin 33
21	PB2 (MOSI/PCINT2)	Digital pin 51 (MOSI)	58	PC5 (A13)	Digital pin 32
22	PB3 (MISO/PCINT4)	Digital pin 50 (MISO)	59	PC6 (A14)	Digital pin 31
23	PB4 (OC2A/PCINT4)	Digital pin 10 (PWM)	60	PC7 (A15)	Digital pin 30
24	PB5 (OCIA/PCINT5)	Digital pin 11 (PWM)	61	VCC	VCC
25	PB6 (OCIB/PCINT6)	Digital pin 12 (PWM)	62	GND	GND
26	PB7 (OC0A/OCIC/PCINT7)	Digital pin 13 (PWM)	63	PJ0 (RXD3/PCINT9)	Digital pin 15 (RX3)
27	PH7 (T4)		64	PJ1 (TXD3/PCINT10)	Digital pin 14 (TX3)
28	PG3 (TOSC2)		65	PJ2 (XCK3/PCINT11)	
29	PG4 (TOSC1)		66	PJ3 (PCINT12)	
30	RESET	RESET	67	PJ4 (PCINT13)	
31	VCC	VCC	68	PJ5 (PCINT14)	
32	GND	GNS	69	PJ6 (PCINT15)	
33	XTAL2	XTAL2	70	PG2 (ALE)	Digital pin 39
34	XTAL1	XTAL1	71	PA7 (AD7)	Digital pin 29
35	PL0 (ICP4)	Digital pin 49	72	PA6 (AD6)	Digital pin 28
36	PL1 (ICP5)	Digital pin 48	73	PA5 (AD5)	Digital pin 27
37	PL2 (T5)	Digital pin 47	74	PA4 (AD4)	Digital pin 26
75	PA3 (AD3)	Digital pin 25	88	PK1 (ADC9/PCINT17)	Analog pin 9
76	PA2 (AD2)	Digital pin 24	89	PK0 (ADC8/PCINT16)	Analog pin 8
77	PA1 (AD1)	Digital pin 23	90	PF7 (ADC7/TDI)	Analog pin 7
78	PA0 (AD0)	Digital pin 22	91	PF6 (ADC6/TDO)	Analog pin 6
79	PJ7		92	PF5 (ADC5/TMS)	Analog pin 5
80	VCC	VCC	93	PF4 (ADC4/TCK)	Analog pin 4
81	GND	GND	94	PF3 (ADC3)	Analog pin 3
82	PK7 (ADC15/PCINT23)	Analog pin 15	95	PF2 (ADC2)	Analog pin 2
83	PK6 (ADC14/PCINT22)	Analog pin 14	96	PF1 (ADC1)	Analog pin 1
84	PK5 (ADC13/PCINT21)	Analog pin 13	97	PF0 (ADC0)	Analog pin 0
85	PK4 (ADC12/PCINT20)	Analog pin 12	98	AREF	Analog Reference
86	PK3 (ADC11/PCINT19)	Analog pin 11	99	GND	GND
87	PK2 (ADC10/PCINT18)	Analog pin 10	100	AVCC	VCC

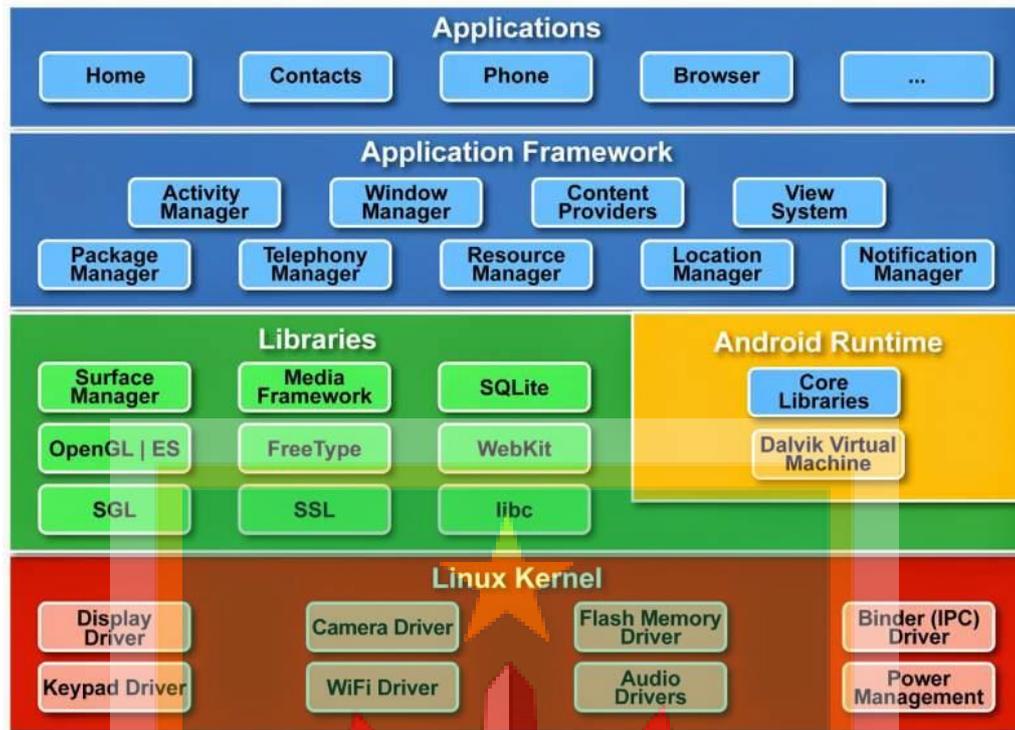
### 2.2.3 Android Operating System

Android merupakan *Operating System* berbasis Linux, dirancang untuk *mobile device* layer sentuh. Pada mulanya Android dikembangkan oleh Android Inc. dengan investor Google. Namun pada tahun 2005 secara resmi Android dibeli oleh Google, dan dirilis secara resmi pada tahun 2007. Android secara resmi dikembangkan oleh Open Handset Alliance, yakni asosiasi yang terdiri dari sebanyak 84 perusahaan yang beberapa diantaranya adalah perusahaan-perusahaan ternama seperti Intel, Sony, Samsung, dan Google. Meski sudah diperkenalkan pada tahun 2007, Android baru benar-benar diluncurkan secara massal pada tahun 2008. Seperti sistem operasi pada umumnya, Android berfungsi untuk mengelola sumber daya perangkat layaknya memori dan *Disk space*, mengeksekusi aplikasi dan *software*, selain itu juga untuk menampilkan *user interface*.

Dalam perkembangannya sampai dengan saat ini android telah beberapa kali mengalami *upgrade*. Generasi terakhir yang dirilis pada Oktober 2021 yakni Android 12, akan kembali disusul oleh Android 13 yang akan dirilis pada April 2022. Adapun kelebihan Android diantaranya :

- a. Sifat Android yang *open source*
- b. Banyak aplikasi yang tersedia pada Google Play Store
- c. Android dapat digunakan pada berbagai brand *smartphone*
- d. Memiliki *user interface* yang *user friendly*
- e. Tersedia berbagai custom *read-only memory* (ROM)

Android dibangun dengan berorientasi terhadap objek, artinya setiap komponen penyusunnya adalah objek yang *reusable* atau dapat digunakan kembali. Secara garis besar arsitektur android terdiri dari empat lapisan komponen. Berikut ini adalah gambar arsitektur android, ditunjukkan oleh Gambar 2.7.



Gambar 2.7 Arsitektur Android

Pada lapisan terbawah terdapat Linux Kernel, yang merupakan jantung dari seluruh sistem Android, meskipun lapisan ini tidak benar-benar berkomunikasi dengan *user* maupun *developer*. Adapun fungsi dari lapisan Linux Kernel sebagai abstraksi *hardware*, program manajemen memori, pengaturan sekuritas, manajemen energi *software* (baterai), *driver* atau program yang mengontrol *hardware* serta *network stack*.

Pada lapisan selanjutnya terdapat lapisan *Library* yaitu lapisan yang membawa sekumpulan perintah yang berfungsi mengarahkan *android device* untuk menangani berbagai macam jenis data. Contohnya pada *recorder* dari berbagai macam format audio dan video yang diatasi oleh Media *Framework Library*. Di bawah ini merupakan fungsi dari *Library* antara lain sebagai berikut :

- a) Surface Manager : Berfungsi mengolah tampilan Windows pada Layar
- b) SGL : Grafik 2 Dimenso
- c) Open GL|ES : Grafik 2D dan 3D
- d) Media Framework : Membantu perekaman dari berbagai macam format gambar, audio, dan video
- e) *Free Type* : *Translator* font
- f) WebKit : *Browser Engine*
- g) Libc (Sistem C *Library*)

- h) SQLite : Database
- i) Open SSL : Keamanan Jaringan

Android *Library* ini menyangkut *Library* berbasis Java , berfungsi sebagai android *developing*. Contohnya pada *Library* yang memberi fasilitas dalam membangun *user interface*, akses database, penggambaran grafik, serta *library* pada *application framework*.

Lapisan berikutnya adalah Android *Runtime* yang berada pada lapisan yang sama dengan *Library*. Pada lapisan ini terdapat Dalvik VM (*Virtual Machine*), sejenis Java *Virtual Machine* yang tidak hanya didesain khusus namun juga dioptimalkan untuk Android yang berguna untuk membuat tiap aplikasi android berjalan dengan prosesnya masing-masing. Dalvik VM memakai karakteristik dasar Linux, seperti multi *threading* dan manajemen memori. Dalvik VM merupakan *Virtual Machine* berbasis register. Dikarenakan kebutuhan memorinya kecil, Dalvik VM dapat bekerja secara optimal, serta didesain untuk dapat menjalankan banyak *Virtual Machine* pada waktu yang bersamaan. Dalvik VM bergantung pada Sistem Operasi yang mendasari proses isolasi, *threading support* dan manajemen memori, selain itu Dalvik VM juga beroperasi pada eksistensi DEX.

*Layer* terluar pada arsitektur android adalah *Application Framework* yang dapat berinteraksi secara langsung dengan aplikasi yang digunakan oleh *user*. Program yang ada pada *Application Framework* mengatur fungsi dasar dari *device* seperti manajemen *resource*, manajemen panggilan, manajemen window dan sebagainya. Lapisan ini merupakan alat dasar yang bias digunakan untuk mengembangkan aplikasi oleh *developer*. Terdapat beberapa program penting pada *Application Framework* diantaranya sebagai berikut :

- a. *Activity Manager* yang berfungsi sebagai pengontrol seluruh aspek dari siklus hidup aplikasi dan *Activity Stack*.
- b. *Content Providers* yang berfungsi untuk memberikan izin aplikasi untuk mempublikasi dan *sharing* data dengan aplikasi lain.
- c. *Resource Manager* yang berfungsi sebagai pemberi akses kepada *resources* yang bukan kode seperti *strings*, pengaturan warna, dan layout *user interface*.
- d. *Notification Manger* berfungsi untuk membuat aplikasi agar bias menampilkan *reminder* dan notifikasi kepada *user*.
- e. *View System* berfungsi untuk membuat *user interface* pada aplikasi.

#### 2.2.4 Router dan Modem

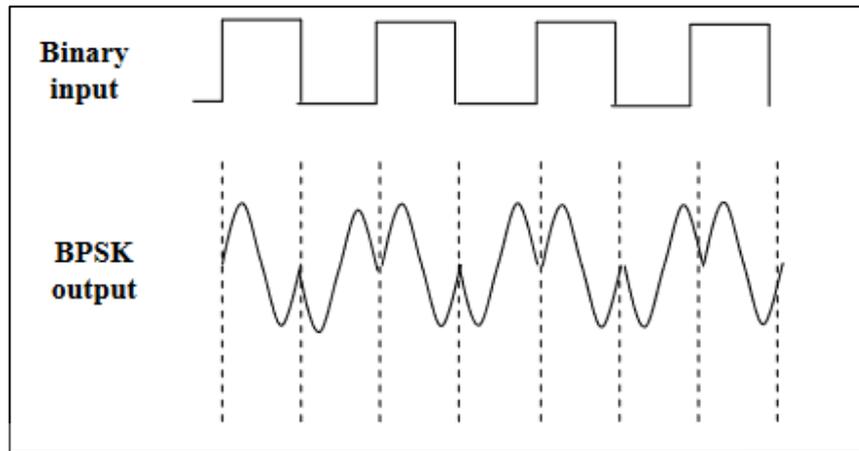
*Router* merupakan sebuah alat yang menghubungkan dua buah jaringan atau lebih yang memiliki alamat jaringan yang berbeda. *Router* menentukan akan diarahkan ke titik jaringan yang mana paket yang ditujukan ke suatu alamat tujuan. *Router* umumnya berfungsi sebagai *gateway* yaitu jalan keluar utama dari suatu jaringan untuk menuju jaringan di luarnya.

Modem merupakan singkatan dari modulator demodulator, yaitu perangkat keras yang digunakan untuk mengubah sinyal digital menjadi sinyal analog sehingga data dari komputer dapat dikirimkan melalui saluran telepon atau saluran lainnya. Data dari komputer yang berbentuk sinyal digital diberikan kepada modem untuk diubah menjadi sinyal analog. Sinyal analog tersebut dapat dikirimkan melalui beberapa media telekomunikasi seperti telepon dan radio. Setibanya di modem tujuan, sinyal analog tersebut diubah menjadi sinyal digital kembali dan dikirimkan kepada komputer.

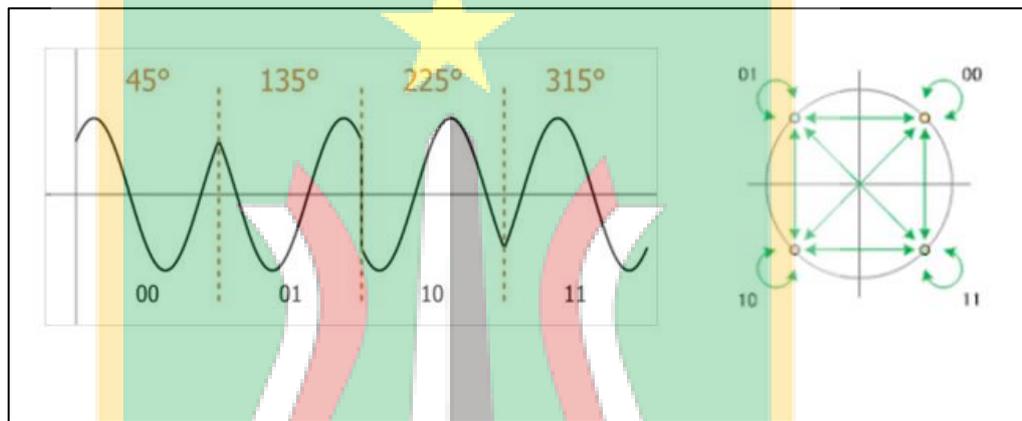
#### 2.2.5 Modulasi Digital

Modulasi Digital adalah proses memasukan sinyal digital (*bitstream*) ke sinyal pembawa. Modulasi digital sebenarnya merupakan proses modifikasi sifat dan karakteristik sifat gelombang pembawa sedemikian rupa sampai bentuk yang dihasilkan (*modulated carrier*) mempunyai sifat dari bit (0 atau 1) yang dimilikinya. Artinya, dengan memperhatikan *modulated carrier* nya, dapat diketahui urutan bit dan juga waktu (*timing*, sinkronisasi). Metode modulasi digital memungkinkan sinyal digital dari tiap level dikirim ke penerima dengan benar. Media transmisi fisik seperti logam atau optik ataupun media transmisi non-fisik juga dapat digunakan untuk transmisi ini [9]. Lebih lanjut dijelaskan beberapa jenis modulasi digital yaitu modulasi BPSK (*Binary Phase Shift Keying*) dan QPSK (*Quadrature Phase Shift Keying*) yang keduanya termasuk dalam jenis Teknik modulasi *Phase Shift Keying*.

Dalam Wifi, modulasi bias QPSK (jika sinyalnya penerimaan kuat) atau BPSK (apabila sinyal penerimaan lemah). Tetapi sinyal digital tersebut diacak terlebih dahulu dengan Teknik DSSS (*Direct Sequence Spread Spectrum*) agar tidak dapat dilacak oleh orang yang tidak berhak.



Gambar 2.8 Sinyal BPSK Dengan Binary Input 1010101



Gambar 2.9 Sinyal QPSK dan Diagram Konstelasi

Gambar 2.21 merupakan contoh untuk sinyal *Binary Phase Shift Keying* (BPSK) dengan masukan 1010101 sedangkan pada gambar 2.22 sinyal *Quadrature phase shift keying* (QPSK) adalah teknik modulasi lain dengan mentransmisikan dua bit per simbol. Dengan kata lain, simbol QPSK tidak mewakili 0 atau 1 — ini mewakili 00, 01, 10, atau 11.

### 2.2.6 Free Space Loss

FSL (*Free Space Loss*) merupakan redaman yang muncul akibat perambatan sinyal dari pemancar ke penerima melalui ruang hampa. *Free Space Loss* dipengaruhi oleh jarak dan frekuensi. Pada ruang di sepanjang ruang antena pemancar dan penerima ini tidak di bolehkan adanya penghalang, karena transmisinya sendiri bersifat *Line of Sight* (LOS). Untuk dapat menentukan nilai dari *Free Space Loss*, dapat menggunakan persamaan sebagai berikut:

$$FSL = 92,45 + 20 \log f + 20 \log D \dots \dots \dots (2.2)$$

Dengan,

FSL = Free Space Loss (dB)

f = Frekuensi (GHz)

D = Panjang lintasan (km)

Apabila frekuensi yang digunakan dalam satuan MHz, maka persamaan yang digunakan untuk mencari *Free Space Loss* (FSL) adalah persamaan sebagai berikut:

$$FSL = 32,45 + 20 \log f + 20 \log D \dots \dots \dots (2.3)$$

FSL = Free Space Loss (dB)

f = Frekuensi (MHz)

D = Panjang lintasan (km)

### 2.2.6 Wifi Modul ESP 01

ESP 01 adalah sebuah perangkat mikrokontroler yang dipakai untuk tersambung ke Wifi. Pada seri ESP 01 ini, tidak terdapat mikrokontroler di dalam *board*-nya sehingga masih dibutuhkan mikrokontroler tambahan agar bisa dipakai. ESP 01 ini adalah sebuah antena penangkap sinyal wifi yang harus terhubung pada mikrokontroler seperti Arduino atau Raspberry. ESP 01 ini terhubung secara serial yang mana pin TX dan RX dari dua *device* dihubungkan, lalu tegangan 3,3 V dan GND pada ESP 01 ini juga dihubungkan ke Arduino. Berikut ini adalah spesifikasi dari Modul Wifi ESP 01 [10].

#### Spesifikasi :

Tegangan operasional 3.3 V

Kompatibel dengan Arduino menggunakan RX-TX UART

*Built-in* regulator tegangan dan komponen manajemen daya

Fungsi *wake up* dan kirimkan paket bias secepat < 2ms

Konsumsi daya ketika *standby* <1.0 mW (DTIM3)

Standard Wifi 802.11 b/g/n

Fungsi Wifi *Direct* (P2P), *soft-AP*

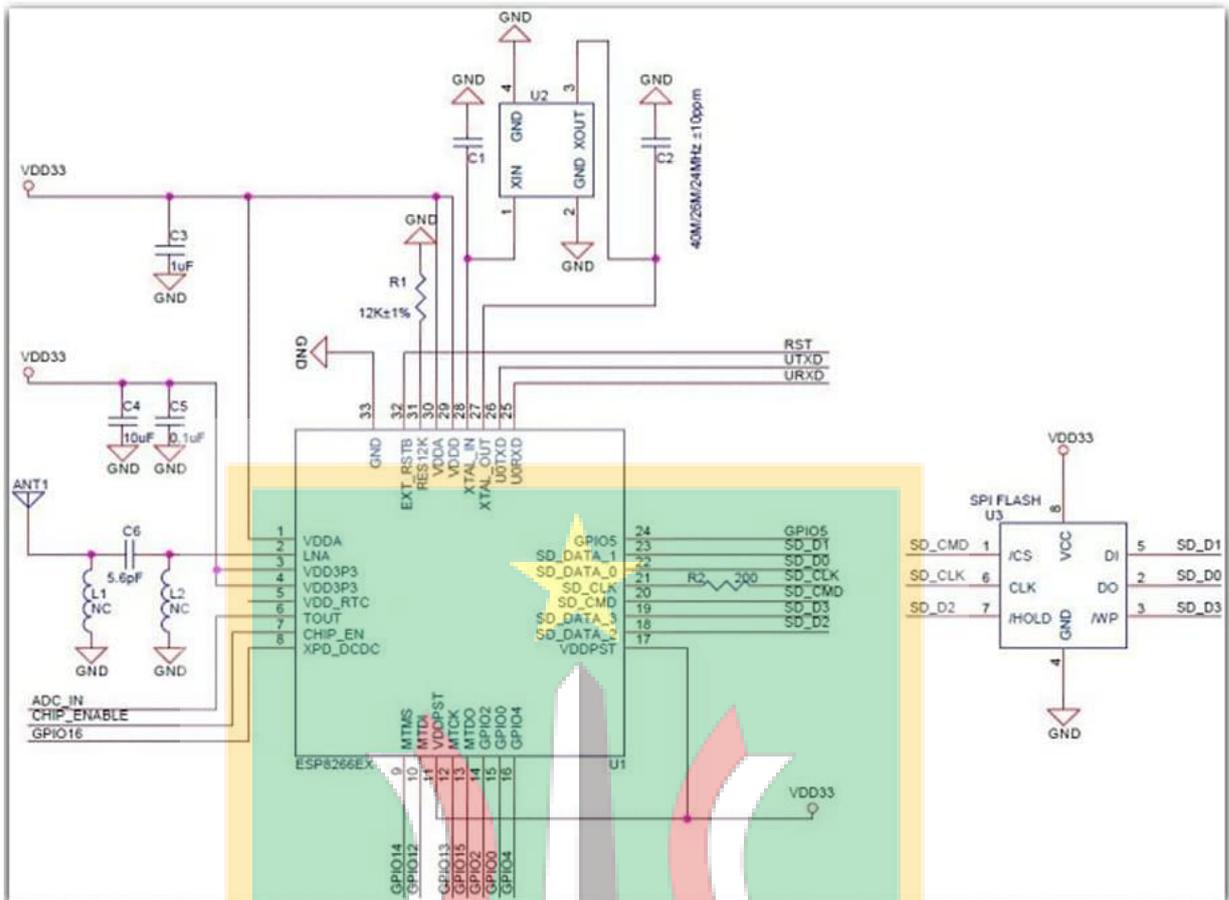
Terintegrasi stack protocol TCP/IP

Output power +19.5dBm pada mode 802.11b

Memori flash 1MB

ESP 01 32-bit

Berikut ini merupakan diagram skematik dari *board* ESP 01 dapat dilihat pada Gambar 2.12



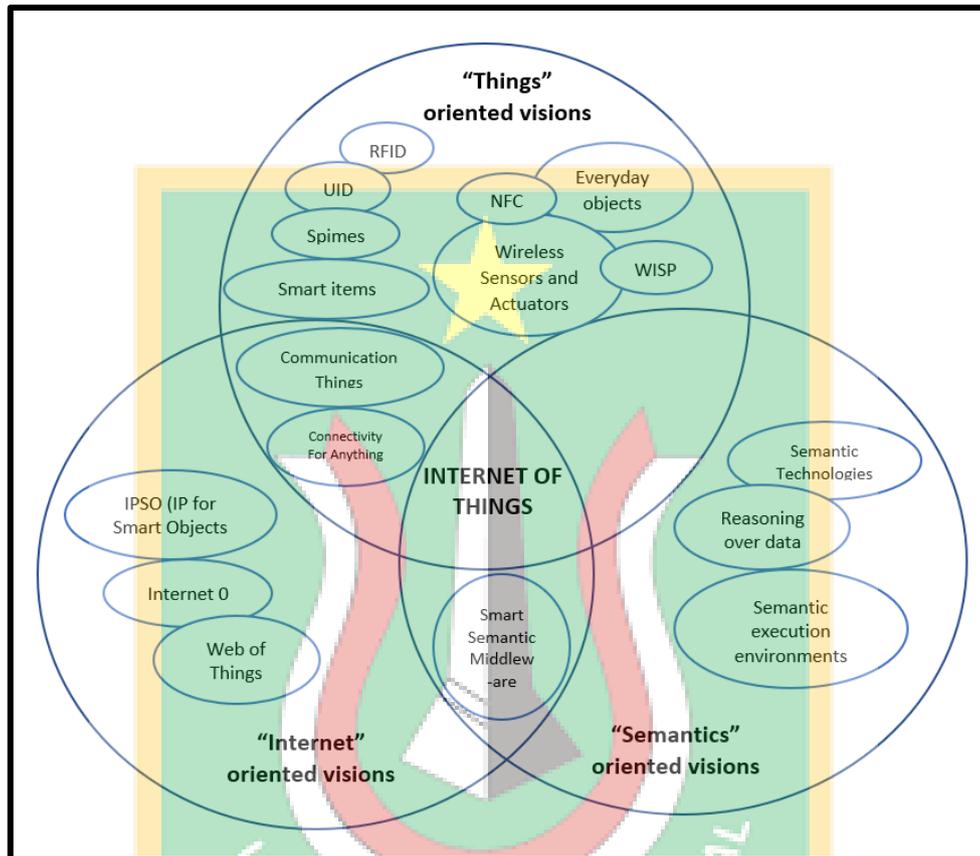
Gambar 2.10 Diagram Skematik ESP 01

## 2.2.7 Internet Of Things (IoT)

*Internet of Things* terdiri dari kata ‘Internet’ yang berarti sebuah jaringan komputer yang memakai protokol-protokol (TCP/IP) yang dipakai untuk komunikasi dan *sharing* informasi dalam suatu lingkup tertentu. Sementara itu kata ‘Things’ didefinisikan sebagai objek dari dunia fisik yang didapat melalui sensor yang dan dikirimkan melalui internet. Menurut Rekomendasi ITU-T Y.2060 *Internet of Things* diartikan sebagai sebuah penemuan yang dapat menyelesaikan masalah melalui kombinasi teknologi serta dampak social. Dilihat dari standarisasi Teknik, IoT digambarkan sebagai infrastruktur global untuk memenuhi kebutuhan informasi masyarakat, yang memungkinkan layanan canggih dengan interkoneksi fisik maupun virtual berdasarkan pada perkembangan informasi dan teknologi komunikasi yang (ICT).

Istilah *Internet of Things* sendiri dicetuskan oleh Kevin Ashton, dalam *e-book* berjudul ‘*Making Sense of IoT*’. Dalam *e-book* tersebut IoT didefinisikan sebagai sensor-sensor yang terkoneksi ke internet dan berperilaku seperti internet dengan membuat koneksi terbuka tiap

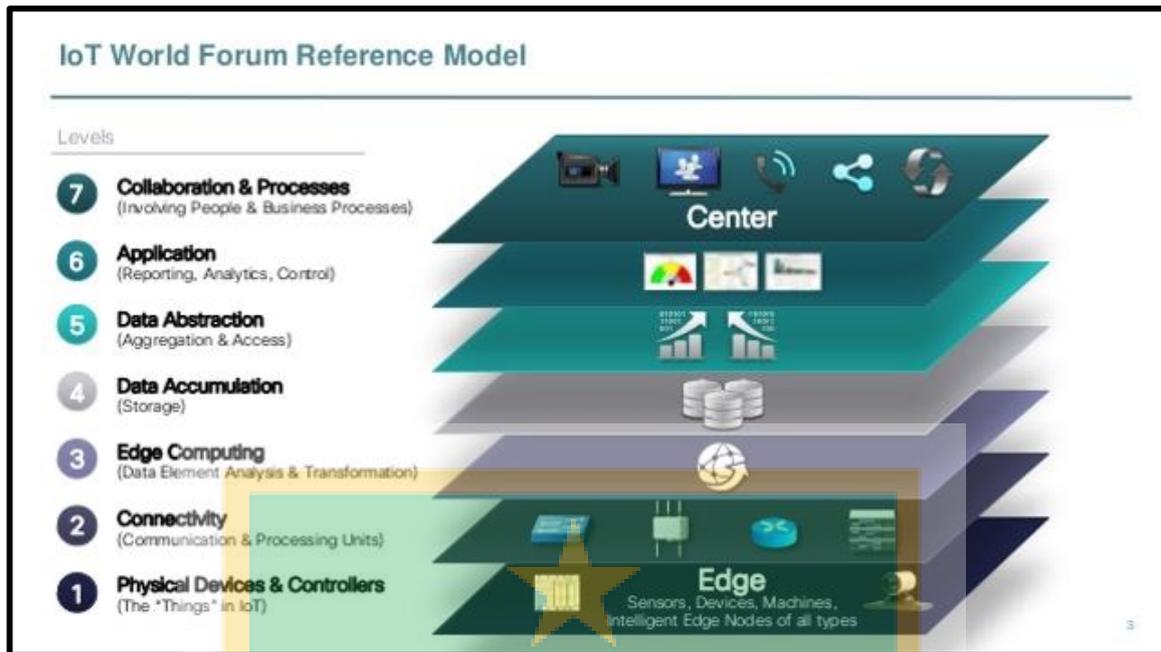
saat, selain itu juga dapat berbagi data secara bebas dan memungkinkan aplikasi-aplikasi yang tak terduga. Hal ini akan membuat komputer dapat memahami dunia di dan menjadi bagian dari hidup manusia [11]. Berikut ini adalah gambaran mengenai konsep utama lahirnya teknologi dan standarisasi dari paradigma *Internet of Things*, ditunjukkan pada gambar 2.13.



**Gambar 2.11** Konsep Utama Lahirnya IoT

### 2.2.9.1 Referensi Model IoT

Berdasarkan IoT *Reference Model*, terdapat 7 (tujuh) lapisan referensi model dalam memahami IoT. Ketujuh lapisan tersebut dapat dilihat pada Gambar 2.14.



Gambar 2.12 Lapisan Referensi Model IoT

Ketujuh lapisan tersebut terdiri dari *Physical Device & Controllers*, *Connectivity*, *Edge Computing*, *Data Accumulation*, *Data Abstraction*, *Application*, *Collaboration & Process*.

**a. *Physical Device & Controllers***

*Physical Device & Controllers* terdiri dari 3 (tiga) bagian diantaranya :

- *Sensor* : Berfungsi dalam mengidentifikasi bagian fisik dari alam. Sensor ini dapat berupa indikator suhu, pengukur jarak dan lainnya.
- *Embedded System* : Sistem minimal atau pusat data yang kecil, lengkao dengan *IO interface*.
- *Gateway* : *Device* untuk komunikasi berfungsi sebagai penghubung perangkat fisik dan internet.

**b. *Connectivity***

Merupakan perangkat komunikasi yang berfungsi sebagai penghubung antara perangkat fisik dan *edge computing*, dapat berupa Wifi, 4G, LoRA dan lainnya.

**c. *Edge Computin***

Lapisan yang fungsinya untuk menangkap data yang dikirim oleh sensor. Pada lapisan ini, data disiapkan agar dapat disimpan dalam database

**d. Data Accumulation**

Pada lapisan ini, data yang masuk kemudian disimpan dalam sebuah *storage*. *Storage* yang bisa digunakan berbasis NoSQL atau SQL

**e. Data Abstraction**

Lapisan ini mengatur aliran data di server atau di *cloud*, di mana data yang masuk kemudian diarahkan ke tempat penyimpanan atau ke tempat lain seperti visualisasi, *machine learning* atau lainnya.

**f. Application**

Lapisan ini berfungsi layaknya kontrol sistem, vertical untuk aplikasi seluler dan juga bisnis intelijen dan analisis. Data tadi, diolah menggunakan *machine learning* sehingga didapatkan klasifikasi data, *clustering*, dan *forecasting*.

**g. Collaboration Process**

Lapisan ini memberi informasi kepada perorangan untuk mengambil tindakan berdasarkan data yang diterima. Proses tersebut dapat dilakukan sebagai umpan balik.

