

## BAB II

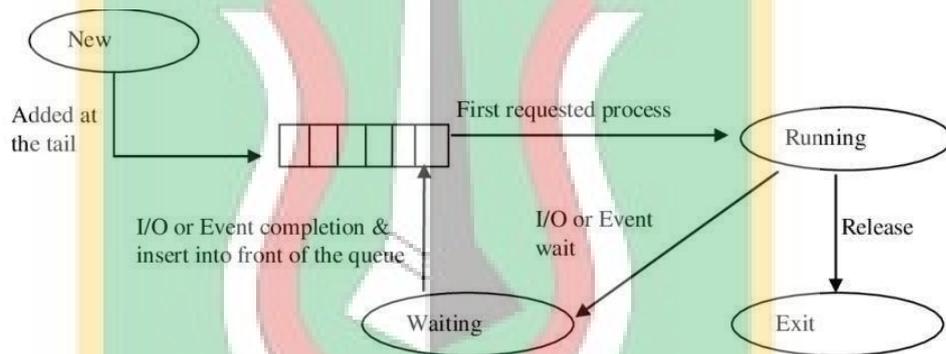
### TINJAUAN PUSTAKA

Bab ini menjelaskan teori untuk penulisan dalam melakukan penerapan metode *First Come First Served* pada website *Wedding Organizer*.

#### 2.1 First Come First Serve

*First Come First Serve* (FCFS) adalah metode penjadwalan produk untuk mementingkan tugas utama yang ada terlebih dahulu. Metode FCFS digunakan karena flow yang sesuai untuk work remaining times yang serupa.

Proses dalam FCFS:



Gambar 2.1 Tahapan FCFS

Pada gambar 2.1 dalam flowchart diatas dijelaskan bahwa metode FCFS (first come first served), new adalah pelanggan baru yang akan memesan paket wedding. Setelah pelanggan baru sudah memesan paket maka proses transaksinya harus menunggu antrian terlebih dahulu baru proses transaksinya bisa di proses oleh admin karena pada dasarnya metode FCFS (first come first served) melayani pelanggan yang pertama dan menyelesaikan transaksi pelanggan yang pertama.

#### 2.2 Wedding Organizer

*Wedding Organizer* merupakan jasa yang menyediakan perencanaan rangkaian pernikahan untuk membantu calon pengantin dan keluarga dalam melangsungkan acara sesuai sesuai jadwal. *Wedding Organizer* juga menyediakan berbagai

macam kebutuhan yang berhubungan dengan acara pernikahan (Purnama, Fauziah and Sholihati, 2022).

### **2.3 Sistem Informasi**

Abdul Qadir mengatakan bahwa istilah sistem informasi seringkali digunakan tanpa aksesoris komputerisasi, yang sebenarnya merupakan bagian terpenting. Ketika datang ke sistem informasi, kita dapat menyimpulkan bahwa sistem informasi adalah sekumpulan proses aktivitas ganda seperti: Penyimpanan, Analisis dan Pemrosesan data akan menghasilkan suatu informasi (Purwanto *et al.*, 2022).

### **2.4 WWW (World Wide Web)**

WWW merupakan ruang informasi sumber daya internet yang diakses oleh browser yang dapat diakses dengan menggunakan berbagai *device* yang dihosting kedalam internet (Saputra and Darusalam, 2022).

### **2.5 Website**

Website atau situs merupakan sekelompok halaman web, yang memiliki suatu nama *domain* atau *subdomain* yang terdapat pada WWW di internet (Agarwal and Jain, 2014).

Website masih sering dipakai hingga saat ini oleh pengguna internet, karena website dapat diakses disegala *device* yang dapat mengakses internet. Website merupakan sebuah aplikasi yang berbasis *Hyper Text Markup Language* (HTML).

### **2.6 PHP**

PHP (*Hypertext Preprocessor*) yang berarti pemrograman dari sisi server untuk membuat website yang dapat berinteraksi langsung dengan database dan ditampilkan pada website (Widodo, 2018).

Netcraft di seluruh dunia banyak situs yang menggunakan PHP bisa disimpulkan bahwa PHP banyak digunakan untuk pengembangan website. Ini mengejutkan para pengembang, termasuk Rasmus sendiri, dan tentu saja, ketika seorang pengembang inti (*core programmer*) mencoba menulis ulang parser PHP-

nya dan mengintegrasikan skrip Zend miliknya dengan mesinnya, pembuatnya Satu dia melebihi harapan Zeev Suraski dan dia Andi Gutsman. Netcraft mengklaim bahwa PHP adalah bahasa web paling populer di dunia. Tentu saja diumumkan bahwa ada. Jutaan domain menggunakan PHP sebagai sisi *server* (Prasetya, 2021).

## 2.7 Javascript

Bahasa pemrograman *prototype* yang berjalan bagian *client* (Ubaya et al., 2018). JavaScript biasanya digunakan sebagai bahasa pemrograman untuk otomatisasi pada suatu halaman pada saat dimuat. Selain untuk otomatisasi, biasanya digunakan sebagai bahasa pemrograman untuk membuat interaksi pada halaman website, sehingga dapat membuat website terlihat lebih hidup. Untuk saat ini, JavaScript tidak hanya dapat digunakan pada website saja, akan tetapi dapat digunakan pada server ataupun perangkat yang memiliki program khusus yaitu JavaScript *engine* (Anam, 2018).

## 2.8 JQuery

jQuery yaitu *library* yang berfokus pada interaksi JavaScript dan HTML. jQuery sangat sederhana dan dapat digunakan untuk memanipulasi dokumen serta menambahkan animasi agar website menjadi lebih hidup. JQuery menjadi salah satu library yang sangat populer diJavaScript (Saputra, Parjito and Wantoro, 2020).

Perpustakaan jQuery memuat fitur berikut:

- Manipulasi HTML Dan CSS
- Manipulasi CSS
- AJAX
- Animasi dan Javascript *Effect*

## 2.9 Database (Basis data)

*Database* yaitu kumpulan entitas data yang memiliki kaitan diantara entitas lainnya sehingga saling berhubungan dan menghasilkan suatu informasi (Abdulrohim *et al.*, 2022).

Dengan adanya *database* dapat meningkatkan efisiensi dari pengelolaan serta pengolahan data. Bentuk dari *database* biasanya berupa *table* yang berisikan atribut-atribut serta nilai tertentu. Untuk pengoperasiannya *database* menggunakan bahasa sql, akan tetapi untuk saat ini *database* dapat dibuat secara mudah dengan fitur yang tersedia di aplikasi *database*. *Database* biasanya digunakan sebagai tempat penyimpanan dari aplikasi baik itu berupa website dan *mobile*. Dalam pembuatannya, *database* memerlukan aplikasi pihak ketiga diantaranya MySQL, MariaDB, PostgreSQL, DLL (Syahputra, Siambaton and Haramaini, 2021).

## 2.10 Mockup

*Mockup* merupakan sebuah rancangan ataupun design model produk yang dibuat sebagai acuan sebelum dapat direalisasikan. *Mockup* merupakan visualisasi untuk memudahkan desainer dalam melakukan evaluasi produk jadi. Umumnya, *mockup* akan ditampilkan ketika proses presentasi, sehingga orang lain bisa memperoleh gambaran produk. Berkaitan dengan hal tersebut, *mockup* adalah kunci yang paling penting untuk dapat menentukan apakah sebuah konsep design bisa diterima atau tidak, terlebih bila berhadapan dengan atasan dipekerjaan. Adanya *mockup* ini dapat memudahkan desainer dalam menerima saran.

*Mockup* terdiri dari dua kategori, yaitu *Low fidelity* sebagai rancangan design sederhana. Sementara *High fidelity* adalah rancangan yang sudah kompleks dalam menampilkan produknya.

## 2.11 CodeIgniter

CodeIgniter merupakan *framework* web *application* bersifat *open source* menggunakan model MVC dengan PHP, sehingga mempercepat pengembang aplikasi web.

## 2.12 Use Case Diagram

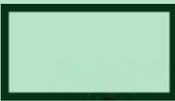
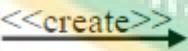
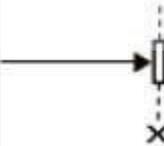
*Use case diagram* direpresentasikan sebagai sekumpulan pesan yang dipertukarkan oleh satu atau lebih unit sistem dengan aktor di luar sistem. Kegunaan praktis dari *use case diagram* adalah untuk mendefinisikan aktifitas interaksi sistem dengan *actor*. *Use case* mencakup interaksi yang ada didalam sistem. Simbol dan deskripsi diagram *use case* ditampilkan dalam *table 2.1*

Ket	Simbol	Deskripsi
<i>Use Case</i>	UseCase	Digunakan sebagai entitas atau entitas yang berinteraksi dengan sitem dan actor
Aktor		Aktor berfungsi untuk mewakili user, dan sistem untuk berinteraksi dengan use case.
Generalisasi		Digunakan sebagi penghubung antar usecase dan actor
<i>Include</i>		Duntuk menunjukan bahwa actor dapat berpartisipasi dengan usecase
<i>Extend</i>		Menunjukkan bahwa adanya tambahna fungsional dari usecase yang lain jika kondisi terpenuhi.

Tabel 2.1.Simbol Use Case Diagram

### 2.13 Sequence Diagram

Diagram ini memvisualisasikan interaksi antar objek dengan sistem secara terperinci dan menampilkan pesan yang akan dikirim pada sistem *Wedding Organizer* berbasis web, terdiri dari garis vertical untuk menggambarkan waktu dan horizontal untuk obejknya. Deskripsi diagram, simbol, dan diagram urutan, seperti yang ditunjukkan pada tabel.2.2.

Keterangan	Simbol	Deskripsi
Objek		Berpartisipasi dalam mengirimkan pesan yang berurut.
Waktu Aktif		Sebagai penerima pesan dan penunjuk objek.
Garis hidup		Ini menunjukkan masa pakai objek untuk seluruh urutan, berakhir pada X saat kelas tidak lagi berinteraksi.
Pesan		Sebagai pengirim pesan ke objek lain.
Create		Arah panah menunjuk ke objek lain dan menciptakan objek lain
Destroy		Mengakhiri objek lain dan panah menunjuk objek yang sudah jadi.

Tabel 2.2.Simbol dan Keterangan Sequence Diagram

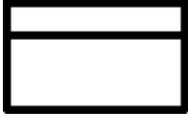
### 2.14 Activity Diagram

Keterangan	Simbol	Deskripsi
Status Awal		Awal dimulainya sistem.
Aktivitas		Aktivitas yang terdaat pada sistem.
Percabangan		Dimana terdiri dari beberapa pilihan aktivitas.
Status akhir		Status terakhir akitivitas diagram
Action		Bagian yang menunjukkan bagaimana setiap kelas antarmuka berinteraksi satu dengan yang lainnya.
Fork node		Aliran yang dapat berubah terhadap aliran tertentu.
Swimlane		Penanggung jawab aktivitas sistem.

Tabel 2.3.Simbol dan KeteranganActivity Diagram

### 2.15 Class Diagram

Keterangan	Simbol	Deskripsi
------------	--------	-----------

<i>Class</i>		Untuk membagi atribut dan operasi yang identic.
<i>Nary Association</i>		Untuk menghindari asosiasi dengan objek lain.
<i>Generalization</i>		Hubungan di mana <i>descendan</i> berbagi struktur data objek di atas objek induknya <i>ancestor</i> .
<i>Realization</i>		Suatu operasi yang dilakukan oleh suatu objek.

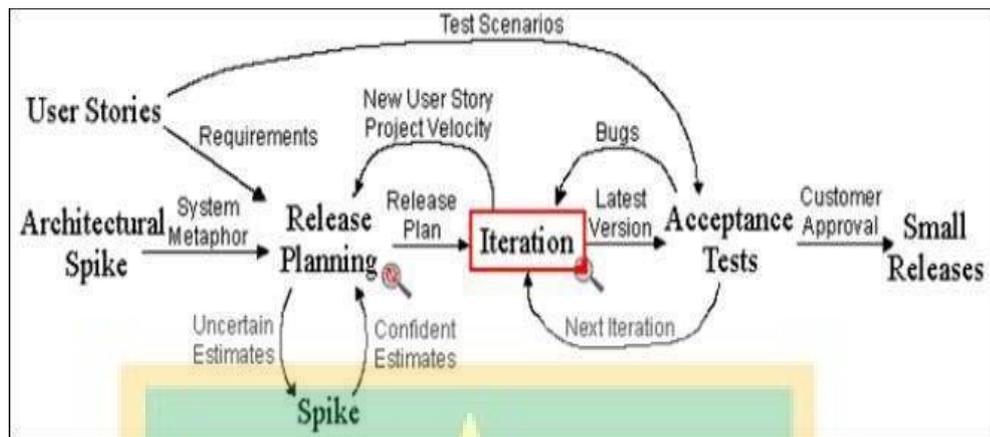
Tabel 2.4 Simbol dan Keterangan *Class Diagram*

*Class Diagram* adalah diagram yang membentuk struktur kelas, atrbut, metode, dan objek. Simbolnya digambarkan dalam bentuk kotak pada tabel 2.4.

## 2.16 Agile Model

Agile merupakan jenis model pengembangan perangkat lunak inkremental (Nurdin, Fauziah and Komalasari, 2022).

Dalam menggunakan metodologi agile, dilakukan secara kolaboratif, terstruktur, dan terorganisir pada setiap pihak yang bekerja sama dalam sebuah tim. Proyek yang menggunakan metodologi agile biasanya dikembangkan dengan jangka waktu yang pendek dan bertahap. Penggunaan metodologi ini juga menuntut kecepatan peneliti untuk mengambil keputusan tanpa mengesampingkan kualitas dari produk. Dalam proses pengembanganya juga dilakukan evaluasi secara berkala untuk memperbaiki hal-hal yang perlu ditingkatkan.



Gambar 2.2 Agile Model (Sarker, 2015)

### 2.17 Alasan Pemilihan Teori, Kerangka Kerja, atau Mekanisme

Sebagai penguat alasan peneliti dalam penentuan penggunaan aplikasi ini diantaranya adalah keuntungan dalam penggunaan metodologi ini. Selain itu juga dilihat berdasarkan perbedaan dengan perbandingan antara metodologi agile dengan metodologi *waterfall*.

Menurut Sarker (2015:68) , terdapat 8 keuntungan dalam menerapkan metodologi ini yaitu :

1. Kepuasan pada pelanggan dengan pengiriman perangkat lunak yang bermanfaat secara terus menerus dan cepat.
2. Interaksi dan orang lebih ditekankan dari pada proses dan alat
3. *Software* memiliki fungsi sering dikirimkan (berminggu-minggu, bukan berbulan-bulan).
4. Interaksi tatap muka merupakan bentuk komunikasi terbaik.
5. Tutup kerjasama harian antara pelaku bisnis dan tim pengembang.
6. Perhatian terus menerus terhadap desain yang baik dan keunggulan teknis.
7. Adaptasi regular terhadap keadaan yang berubah.
8. Keterlambatan perubahan persyaratan dapat dilakukan.

Perbandingan antara metodologi agile dan *waterfall* dinyatakan dalam *table* sebagai berikut: