

BAB II

TEORI PENUNJANG

2.1. Tinjauan Perusahaan

Tinjauan perusahaan adalah sebuah penelitian tentang profil dan struktur organisasi yang ada pada perusahaan yang akan menjadi tempat dibangunnya penelitian berbasis *web* ini nantinya.

2.1.1. Profil Perusahaan

Ummajee merupakan sebuah perusahaan swasta yang didirikan pada Maret 2019 oleh Jihan Al-haddar di Metro Indah Mall (MIM) Blok GF A6 No.25, Jalan Soekarno Hatta, Buah Batu, Kota Bandung. Perusahaan ini bergerak di bidang *fashion*. Ummajee didirikan karena butik selalu identik dengan harga barangnya yang mahal dan hanya terjangkau oleh kalangan menengah keatas, maka dari itu Ummajee hadir dalam konsep butik dengan harga barang yang terjangkau bagi semua kalangan masyarakat namun dengan barang yang berkualitas dan dengan barang yang terbatas.

Ummajee memiliki *brand* tersendiri yaitu *UmmaJee*. Jenis produk yang dijual berupa Gamis, Tunik, *Overalls*, Koko, Celana, Kerudung, dll. Produk yang terdapat di Ummajee tidak hanya untuk kalangan wanita saja, tetapi tersedia juga produk untuk kalangan pria. Namun ukuran yang ditawarkan hanya untuk balita hingga remaja saja. Klasifikasi ukurannya pun bisa dilihat dibawah ini:

- | | | |
|--------------------------------------|---|---------------------|
| a. XS (<i>Extra Small</i>) | : | umur 2 – 3 tahun. |
| b. S (<i>Small</i>) | : | umur 4 – 5 tahun. |
| c. M (<i>Medium</i>) | : | umur 6 – 7 tahun. |
| d. L (<i>Large</i>) | : | umur 8 – 9 tahun. |
| e. XL (<i>Extra Large</i>) | : | umur 10 – 11 tahun. |
| f. XXL (<i>Double Extra Large</i>) | : | umur 12 – 13 tahun. |

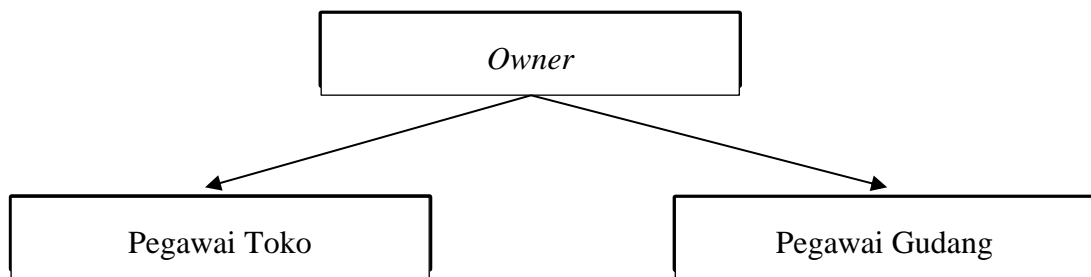
Visi dari Ummajee ialah mejadi butik yang menyediakan pakaian anak-anak yang modis, nyaman, dan trendi seperti hal nya orang dewasa. Sedangkan misi nya adalah ingin menjadikan toko Ummajee sebagai butik yang lebih dikenal oleh kalangan masyarakat luas khususnya masyarakat diluar Bandung.



Gambar II.1 Logo Ummajee

2.1.2. Struktur Organisasi dan Fungsi

Gambar 2.2 berikut ini merupakan struktur organisasi dari Ummajee.



Gambar II.2 Struktur Organisasi

Bagian-bagian deskripsi tugas yang terdapat di Toko Ummajee adalah sebagai berikut :

1. Pemilik / *Owner*

Tugas dan tanggung jawab *Owner* adalah sebagai berikut:

- a. Memimpin perusahaan sekaligus bertanggung jawab penuh atas keberlangsungan perusahaan.

- b. Menentukan jumlah produksi barang pada setiap pengadaan barang.
 - c. Melakukan pemesanan barang.
 - d. Menerima laporan barang masuk, laporan barang keluar harian hingga bulanan dan laporan sisa stok barang.
 - e. *Owner* dapat mengolah data produk, mengolah data kategori, mengolah data barang masuk, mengolah pengguna, membuat daftar pemesanan barang, melakukan peramalan.
2. Pegawai Toko
- Tugas dan tanggung jawab pegawai toko adalah sebagai berikut:
- a. Bertanggung jawab atas kerapihan, kebersihan, dan keindahan toko Ummajee saat jam kerja.
 - b. Melayani kebutuhan konsumen yang datang ke toko.
 - c. Sekaligus memberikan laporan kepada *owner* pengolahan data barang masuk dan keluar di toko.
3. Pegawai Gudang
- Tugas dan tanggung jawab pegawai gudang adalah sebagai berikut:
- a. Bertanggung jawab atas ketersediaan barang di gudang serta pengiriman barang kepada konsumen online.
 - b. Melakukan pengiriman barang ke toko.
 - c. Melakukan pengecekan barang di gudang secara berkala.
 - d. Melapor kepada *owner* bila terjadi stok barang mulai menipis.

2.2. Landasan Teori

Landasan teori ini berisi tentang teori yang dibutuhkan dalam melakukan penelitian sistem aplikasi ini.

2.2.1. Sistem Prediksi

Prediksi adalah suatu proses memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi masa lalu dan sekarang yang dimiliki, agar kesalahannya (selisih antara sesuatu yang terjadi dengan hasil perkiraan) dapat diperkecil. Prediksi tidak harus memberikan jawaban

secara pasti kejadian yang akan terjadi, melainkan berusaha untuk mencari jawaban sedekat mungkin yang akan terjadi [6] [15].

Pengertian Prediksi sama dengan ramalan atau perkiraan. Menurut kamus besar bahasa Indonesia, prediksi adalah hasil dari kegiatan memprediksi atau meramal atau memperkirakan nilai pada masa yang akan datang dengan menggunakan data masa lalu. Prediksi menunjukkan apa yang akan terjadi pada suatu keadaan tertentu dan merupakan input bagi proses perencanaan dan pengambilan keputusan.

Peramalan (*forecasting*) adalah suatu prosedur untuk membuat informasi factual tentang situasi sosial masa depan atas dasar informasi yang telah ada tentang masalah kebijakan. Ramalan mempunyai tiga bentuk utama: proyeksi, prediksi, dan perkiraan.

1. Suatu proyeksi adalah ramalan yang didasarkan pada ekstrapolasi atas kecenderungan masa lalu maupun masa kini ke masa depan. Proyeksi membuat pertanyaan yang tegas berdasarkan argument yang diperoleh dari metode tertentu dan kasus yang paralel.
2. Sebuah prediksi adalah ramalan yang didasarkan pada asumsi teoritik yang tegas. Asumsi ini dapat berbentuk hukum teoretis (misalnya hukum berkurangnya nilai uang), proposisi teoritis (misalnya proposisi bahwa pecahnya masyarakat sipil diakibatkan oleh kesenjangan antara harapan dan kemampuan), atau analogi (misalnya analogi antara pertumbuhan organisasi pemerintah dengan pertumbuhan organisme biologis).
3. Suatu perkiraan (*conjecture*) adalah ramalan yang didasarkan pada penilaian yang *informative* atau penilaian pakar tentang situasi masyarakat masa depan.

Tujuan dari pada diadakannya peramalan kebijakan adalah untuk memperoleh informasi mengenai perubahan dimasa yang akan datang yang akan mempengaruhi terhadap implementasi kebijakan serta konsekuensinya. Oleh

karenanya, sebelum rekomendasi diformulasikan perlu adanya peramalan kebijakan sehingga akan diperoleh hasil rekomendasi yang benar-benar akurat untuk diberlakukan pada masa yang akan datang. Didalam memprediksi kebutuhan yang akan datang dengan berpijak pada masa lalu, dibutuhkan seseorang yang memiliki daya sensitifitas tinggi dan mampu membaca kemungkinan-kemungkinan dimasa yang akan datang. Permalan kebijakan juga diperlukan untuk mengontrol, dalam artian, berusaha merencanakan dan menetapkan kebijakan sehingga dapat memberikan alternatif-alternatif tindakan yang terbaik yang dapat dipilih diantara berbagai kemungkinan yang ditawarkan oleh masa depan. Masa depan juga terkadang banyak dipengaruhi oleh masa lalu. Dengan mengacu pada masa depan analisis kebijakan harus mampu menaksir nilai apa yang bisa atau harus membimbing tindakan di masa depan [6].

2.2.2. Metode *Weight Moving Average* (WMA)

Weighted Moving Average (WMA) adalah rata-rata bergerak yang memiliki bobot. Metode *Weighted Moving Average* merupakan metode yang mempunyai teknik pemberian bobot yang berbeda atas data yang tersedia dengan demikian bahwa data yang paling akhir adalah data yang paling relevan untuk peramaln sehingga diberi bobot yang lebih besar. Bobot di-tentukan sedemikian rupa sehingga jumlah keseluruhan sama dengan satu. Metode *weighted moving average* atau metode rata-rata bergerak tertimbangan terlebih dahulu menejemen atau analisis data menetapkan bobot (*weighted factor*) dari data yang ada. Penetapan bobot dimaksud bersifat subjektif, tergantung pada pengalaman dan opini analisis data. Jumlah keseluruhan bobot sama dengan satu. Rumus metode *weighted moving average* sebagai berikut [7]:

$$F_t = \frac{\sum(\text{bobot pada periode } n) (\text{bobot pada periode } n)}{\sum \text{bobot}} \quad (2.1)$$

Dimana :

F_t = Peramalan permintaan periode berikutnya

Bobot = Penilaian sesuai panjang periode

Untuk analisa perhitungan metode *weighted moving average* dapat diliha pada subbab 3.1.3.2.

2.2.3. Pengukuran Kesalahan Peramalan

Pada dasarnya tidak ada teknik yang dapat menghasilkan ramalan yang sangat akurat, yaitu masa yang akan datang tidak mungkin dapat diramalkan secara tepat dan sempurna. Karena itu keandalan ramalan digunakan untuk melihat seberapa handal atau akuratnya suatu metode peramalan. Untuk menguji keakuratannya ramalan tersebut, peramal dapat menggunakan pengukuran keandalan, yaitu dengan *Mean Squared Error (MSE)* dan *Mean Absolute Percentage Error (MAPE)*. Secara umum semakin rendah nilai MAPE berarti semakin baik dan akurat [8].

Mean squared error adalah rata-rata dari kesalahan peramalan dikuadratkan. *Mean absolute percentage error* adalah persentase kesalahan dari peralaman. MSE dihitung dengan persamaan (2.2) dan MAPE dihitung dengan Persamaan (2.3) [7].

Mean Squared Error (MSE)

$$MSE = \sum |X_t - F_t|^2 / n \quad (2.2)$$

Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{100}{n} \sum |X_t - F_t / X_t| \quad (2.3)$$

Dimana :

X = Data aktual pada periode t

F = Data peramalan pada periode t

n = Jumlah data

Untuk analisa perhitungan kesalahan peramalan dapat dilihat pada subbab 3.1.3.3.

2.2.4. *Safety Stock dan Reorder Point*

Safety Stock adalah adalah persediaan ekstra dilakukan untuk melayani asuransi terhadap fluktuasi permintaan. *Safety stock* berguna untuk melihat berapa stok barang aman di setiap harinya. Rumus yang digunakan dalam melakukan perhitungan jumlah persediaan antisipasi bisa dilihat pada persamaan (2.4) [7] :

$$SS = (Maximum Usage - Average Usage) \times Lead Time \quad (2.4)$$

Dimana :

SS = Jumlah persediaan antisipasi (unit)

Maximum Usage = Penggunaan unit maksimal per hari (unit)

Average Usage = Penggunaan rata-rata unit per hari (unit)

Lead time = Waktu yang dibutuhkan untuk menerima pesanan (hari)

Reorder Point adalah titik waktu ketika sebuah pesanan baru harus diadakan. Setelah didapatkan nilai *safety stock* bisa dilakukan pencarian nilai *reorder point* untuk melihat pada saat kapan kita harus memproduksi baju kembali. Rumus yang digunakan bisa dilihat pada persamaan 2.5 [7]. Jika *stock* barang kurang dari nilai *reorder point* maka status barang tidak aman, dan segera melakukan produksi sesuai dengan nilai peramalan. Jika stok barang lebih dari nilai *reorder point* maka status barang aman, dan menunggu sampai status tidak aman lalu melakukan produksi kembali sesuai nilai peramalan.

$$ROP = (Average Usage \times Lead Time) + Safety Stock \quad (2.5)$$

Dimana:

ROP	= Titik pemesanan kembali bahan baku (unit)
<i>Average Usage</i>	= Penggunaan rata-rata unit per hari (unit)
<i>Lead Time</i>	= Waktu yang dibutuhkan untuk menerima pesanan (hari)
<i>Safety Stock</i>	= Jumlah persediaan antisipasi (unit)

Untuk analisa *safety stock* dan *reorder point* dapat dilihat pada subbab 3.1.3.4.

2.2.5. Website

Website adalah : “*Web* dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar, data animasi, suara, video dan gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*)”[5].

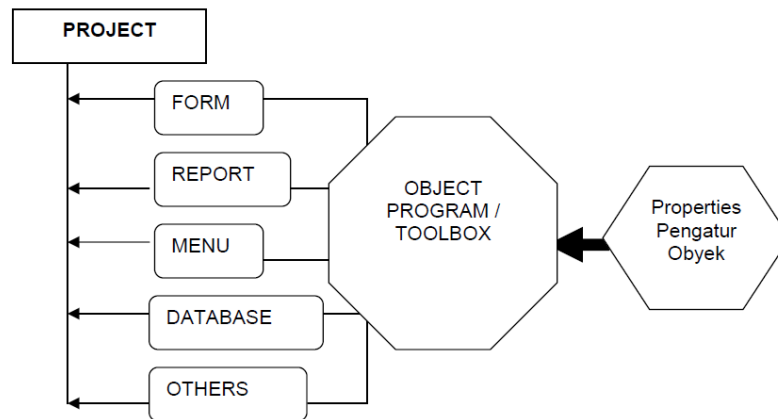
Web adalah : “Sekumpulan halaman yang terdiri dari beberapa halaman yang berisi informasi dalam bentuk data digital baik berupa *text*, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet” [5].

Berdasarkan uraian, bisa disimpulkan bahwa *web* adalah Sebuah software yang berfungsi untuk menampilkan dokumen - dokumen pada suatu *web* yang membuat pengguna dapat mengakses internet melalui software yang terkoneksi dengan internet.

2.2.6. Object Oriented Programming (OOP)

Pemrograman Berbasis Objek atau *Object Oriented Programming* (OOP) adalah sebuah istilah yang diberikan kepada bahasa pemrograman yang menggunakan teknik berorientasi atau berbasis pada sebuah obyek dalam pembangunan program aplikasi, maksudnya bahwa orientasi pembuatan program tidak lagi menggunakan orientasi linear melainkan berorientasi pada objek-objek yang terpisah-pisah. Suatu perintah dalam bahasa ini diwakili oleh sebuah Objek yang didalamnya berisi beberapa perintah-perintah standar sederhana [9]. Objek ini dikumpulkan dalam modul *form* atau *report* atau modul lain dan disusun didalam

sebuah *project*. Gambaran tentang pemrograman ini dapat dilihat pada gambar 2.3 seperti dibawah ini:



Gambar II.3 Model Pemrograman Berbasis Objek

2.2.7. PHP (*Perl Hypertext Preprocessor*)

PHP singkatan dari *Perl Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang berintegrasi dengan HTML dan berada pada *server* (*server side HTML embedded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman *web* dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru/*up to date*. Semua *script* PHP dieksekusi pada *server* dimana *script* tersebut dijalankan.

Pengertian PHP adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah *web* dan bisa digunakan pada HTML. PHP merupakan singkatan dari “PHP : *Perl Hypertext Preprocessor*”, dan merupakan bahasa yang disertakan dalam dokumen HTML sekaligus bekerja di sisi *server* (*server-server HTML-embedded scripting*). Artinya sintaks dan

perintah yang diberikan akan sepenuhnya dijalankan di *server* tetapi disertakan pada halaman HTML biasa, sehingga scriptnya tak tampak disisi *client*.

PHP dirancang untuk dapat bekerja sama dengan *database server* dan dibuat sedemikian rupa sehingga pembuatan dokumen HTML yang dapat mengakses *database* menjadi begitu mudah. Tujuan dari bahasa *scripting* ini adalah untuk membuat aplikasi dimana aplikasi tersebut dibangun oleh PHP pada umumnya akan memberikan hasil kepada *web browser*, tetapi proses keseluruhannya dijalankan di *server*. [10]

Adapun kelebihan-kelebihan dari PHP yaitu:

- a. PHP merupakan sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya. Tidak seperti halnya bahasa pemrograman aplikasi yang lainnya.
- b. PHP dapat berjalan pada *web server* yang dirilis oleh *Microsoft*, seperti IIS atau PWS juga pada apache yang bersifat *open source*.
- c. Karena sifatnya yang *open source*, maka perubahan dan perkembangan interpreter pada PHP lebih cepat dan mudah, karena banyak milis-milis dan *developer* yang siap membantu pengembangannya.
- d. Jika dilihat dari segi pemahaman, PHP memiliki referensi yang begitu banyak sehingga sangat mudah untuk dipahami.
- e. PHP dapat berjalan pada 3 (tiga) *operating system*, yaitu: *Linux*, *unix*, dan *windows*, dan juga dapat dijalankan secara *runtime* pada suatu *console*.
- f. PHP bersifat bebas dipakai (*free*).

2.2.8. Framework Laravel

Laravel adalah sebuah *framework* PHP yang dirilis dibawah lisensi MIT, dibangun dengan konsep MVC (*model view controller*). Laravel sendiri adalah *framework* PHP MVC yang dikembangkan oleh Taylor Otwell pada tahun 2011 dan sekarang telah mencapai versi 7 dan dengan slogan laravel “*PHP THAT DOESN'T HURT. CODE HAPPY & ENJOY THE FRESH AIR.*” kita akan merasa

bahwa pemrograman php itu menyenangkan dan mudah karna banyak sekali fitur-fitur yang akan sangat membantu kita dalam mengkoding dengan menggunakan *framework* laravel ini [11].

Dan berikut adalah beberapa fitur-fitur yang akan membantu anda dalam memprogram *web* anda.

a. *RESTful Routing*

RESTful adalah cara baru dalam mengelola request seperti *GET*, *POST*, *PUT*, *PATCH*, *DELETE*, *STORE*, dll Jika kita biasanya, dalam membuat *controller* yang berisi *method* tertentu misalkan “*UserController* dengan *method showUser*” lalu kita dapat mengakses www.WebYangKamuBuat.com/user, tapi dengan *Laravel* untuk mengakses situs tersebut anda harus mengetikkan kode ini di file “*routes.php*”

```
“Route::get('/user', array('uses' => 'showUser@UserController');”
```

maksudnya adalah *request* dengan *method* “*GET*” dan url user akan diarahkan pada *UserController method showUser*. Dengan menggunakan fitur ini keuntungannya adalah anda bisa apa saja url dan fitur yang ada dalam *website* kita dan kemana larinya (*controllernya*) dan dengan ini akan memudahkan anda dalam proses pengembangan dan perbaikan *error*.

b. *Composer Powered*

Composer adalah *Dependency Management* *PHP* yang membantu kita untuk mencari *library* yang ingin kita pakai dan menginstalnya, semua *library* dari *composer* dihost di packagist.org dan *laravel* kita menginstalnya melalui *Composer*. dan kasus yang sering dan biasa kita temui yaitu menemukan *library* yang kita sukai tetapi ada di *framework* lain menemukan *library* yang sesuai tapi repot karna kita harus menginstallnya kembali setiap akan memulai *project* baru. Dengan *composer* yang ada di *laravel* kita hanya perlu membuat sebuah file

yang bernama `composer.json` yang berisi *library* apa yang kita perlukan contohnya kita memerlukan *library Carbon* untuk manipulasi tanggal, kita hanya perlu mengetikkan

```
{“require”: {“brinanesbit/carbon”: “~1.6”}}
```

Setelah itu anda harus mengetikkan

```
curl -s http://getcomposer.org/installer | php  
php composer.phar install
```

dan terakhir *include* `vendor/autoload` pada awal file `index.php` anda dengan mengetikkan

```
require ‘vendor/autoload.php’;
```

dan *library carbon* sudah dapat anda pakai

c. *Command Line Tools* (Artisan)

Laravel telah dibekali dengan beberapa *command line tools* bernama “Artisan” yang harus diketikkan di CMD dan berikut beberapa fungsi dari artisan *Database Migration* *Serve application* (tanpa perlu menaruhnya di `htdocs`, keren kan?), merubah status aplikasi menjadi *down* dan *up* *Database Seeding* (Memasukkan data awal ke database), *Tail* (Melihat log server secara realtime) dan masih banyak lagi

d. *Test Driven Development Ready*

Test driven development adalah cara baru pengembangan software, dimana kita harus menuliskan *test code* terlebih dahulu sebelum kita menulis kode sebenarnya. Laravel sendiri memakai PHP unit sebagai test frameworknya.

e. *Beautiful Templating Engine*

Templating engine adalah program yang memparse *syntax template engine* tersebut ke HTML. Laravel sendiri memiliki *templating engine* yang *powerful* bernama *blade*, dengan menggunakan laravel

sebenarnya kita bisa memilih untuk menggunakan PHP murni atau *Templating Engine* hanya dengan penamaan filenya dengan menggunakan “*blade*” di nama file nya contoh “namafile.blade.php” dan dengan *Templating Engine* akan membuat pengkodean lebih mudah dan simple sebagai contoh di PHP biasa

```
“if (isset($nama)) {
```

```
Nama anda
```

```
}
```

dan dengan blade

```
@if (isset($nama))
```

```
Nama anda {{ $nama }}
```

```
@endif
```

f. *Elegant Code*

Di Laravel sudah tersedia fitur *design pattern facade*, yang berarti kita tidak perlu berurusan dengan API dari library-library yang rumit, dan kita hanya perlu mengaksesnya dengan cara yang sangat mudah tanpa harus meload *class* tersebut dan menginstansiasi *class* tersebut. Berikut adalah contohnya,

```
Session::set('nama', 'Tedja');
```

```
Route::get('/user', 'uses' => 'index@UserController');
```

```
Form::text('nama');
```

g. *Eloquent ORM*

Di dalam Laravel ORM (*Object Relational Model*) memiliki beberapa kelebihan dibandingkan dengan ORM lainnya. Dengan ORM yang ada di laravel anda bisa memprogram *web* dengan kode yang mudah, singkat, tidak ribet, dan simple dan masih banyak fitur fitur lainnya seperti:

- a. *Queue*
- b. *SSH*
- c. *Authentication*
- d. *Pagination* </li• >
- e. *Session*
- f. *Redis Support*
- g. *Schema Builder*
- h. *Mail*
- i. *Validator*
- j. *Session*

2.2.9. UML

“*Unified Modeling Language (UML)* adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem”

Unified Modeling Language (UML) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software*. Diagram *Unified Modelling Language (UML)* antara lain sebagai berikut:

1. *Use Case Diagram*

Use case menggambarkan *external view* dari sistem yang akan kita buat modelnya. Model *use case* dapat dijabarkan dalam diagram *use case*, tetapi perlu diingat, diagram tidak identik dengan model karena model lebih luas dari diagram. *Use case* harus mampu menggambarkan urutan

aktor yang menghasilkan nilai terukur. Untuk melihat *use case* diagram yang dibuat pada sistem ini dapat dilihat pada subbab 3.1.5.1.

2. *Class Diagram*

Kelas sebagai suatu set objek yang memiliki atribut dan perilaku yang sama, kelas kadang disebut kelas objek. *Class* memiliki tiga area pokok yaitu:

- a. Nama, kelas harus mempunyai sebuah nama.
- b. Atribut, adalah kelengkapan yang melekat pada kelas. Nilai dari suatu kelas hanya bisa diproses sebatas atribut yang dimiliki.
- c. Operasi, adalah suatu proses yang dapat dilakukan oleh sebuah kelas, baik pada kelas itu sendiri ataupun kepada kelas lainnya.

Untuk melihat *class* diagram yang dibuat pada sistem ini dapat dilihat pada subbab 3.1.5.4.

3. *Activity Diagram*

Diagram *activity* menunjukkan aktivitas sistem dalam bentuk kumpulan aksi-aksi, bagaimana masing-masing aksi tersebut dimulai, keputusan yang mungkin terjadi hingga berakhirnya aksi. *Activity* diagram juga dapat menggambarkan proses lebih dari satu aksi selama waktu bersamaan. “Diagram *activity* adalah aktifitas-aktifitas, objek, statem transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas”. Untuk melihat *class* diagram yang dibuat pada sistem ini dapat dilihat pada subbab 3.1.5.2.

4. *Sequance Diagram*

“Secara mudahnya *sequance* diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case* diagram.” [12]. Untuk melihat *class* diagram yang dibuat pada sistem ini dapat dilihat pada subbab 3.1.5.3.

2.2.10. ERD (*Entity Relationship Diagram*)

ERD (*Entity Relationship Diagram*) adalah model teknik pendekatan yang menyatakan atau menggambarkan hubungan suatu model. Didalam hubungan ini tersebut dinyatakan yang utama dari ERD adalah menunjukkan objek data (*Entity*) dan hubungan (*Relationship*), yang ada pada *Entity* berikutnya. [10]

“*Entity Relationship Diagram* (ERD) adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antar entitas”.

Proses memungkinkan analis menghasilkan struktur basis data dapat disimpan dan diambil secara efisien.

Simbol-simbol dalam ERD (*Entity Relationship Diagram*) adalah sebagai berikut:

- a. Entitas: suatu yang nyata atau abstrak yang mempunyai karakteristik dimana kita akan menyimpan data.
- b. Atribut: ciri umum semua atau sebagian besar instansi pada entitas tertentu.
- c. Relasi: hubungan alamiah yang terjadi antara satu atau lebih entitas.
- d. Link: garis penghubung atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi.

Untuk melihat gambar ERD yang dibuat pada sistem ini dapat dilihat pada gambar 3.23.

2.2.11. MYSQL

“MySQL adalah sebuah basis data yang mengandung satu atau jumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau sejumlah tabel”.

MySQL adalah database *server open source* yang cukup populer keberadaannya. Dengan berbagai keunggulan yang dimiliki, membuat software

database ini banyak digunakan oleh praktisi untuk membangun suatu *project*. Adanya fasilitas API (*Application Programming Interface*) yang dimiliki oleh MySQL, memungkinkan bermacam – macam aplikasi komputer yang ditulis dengan berbagai bahasa pemrograman dapat mengakses basis data MySQL.

Tipe data MySQL, “Tipe data MySQL adalah data yang terdapat dalam sebuah tabel berupa *field – field* yang berisi nilai dari data tersebut. Nilai data dalam *field* memiliki tipe sendiri – sendiri”. [13]

2.2.12. XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis. Untuk mendapatkannya dapat mendownload langsung dari *web* resminya. [14]