BAB 2

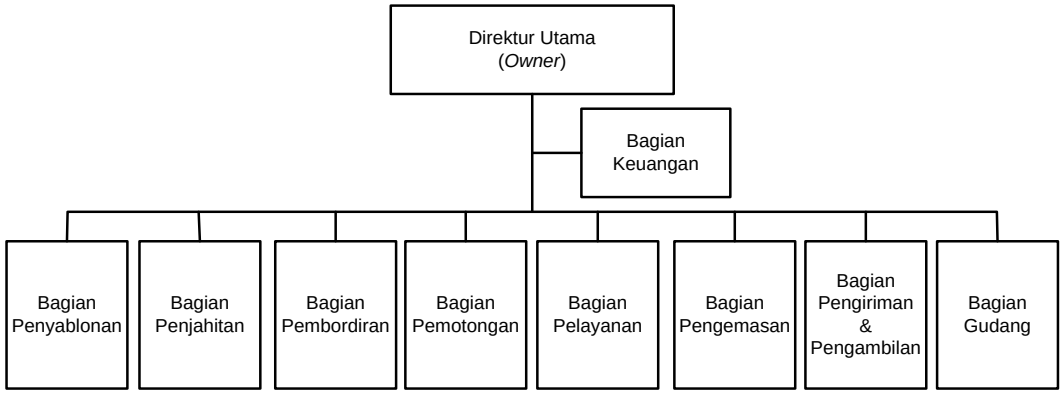
TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

Pabrik baju 12Titan berdiri pada tanggal 12 Januari 2009 dan merupakan sebuah usaha konveksi yang termasuk jenis garment, souvenir dan advertising. Produk yang dihasilkan berupa pakaian meliputi kaos, kemeja, seragam, dll. Usaha ini merupakan Usaha Kecil dan Menengah (UKM) yang beralamat di JL. Sidomukti No. T-5 Rt/Rw 02/11 Kelurahan Sukaluyu Kecamatan Cibeuying Kaler Kota Bandung.

2.1.1 Struktur Organisasi

Struktur organisasi merupakan penjabaran tingkatan hierarki suatu organisasi atau perusahaan, dimana dijelaskan bagian-bagian yang terkait dalam perusahaan tersebut. Struktur organisasi dari Pabrik Baju 12Titan dapat dilihat pada gambar 2.1.



Gambar 2.1 Struktur Organisasi Pabrik Baju 12Titan

2.1.2 Deskripsi Kerja di Pabrik Baju 12Titan

Deskripsi kerja di Pabrik Baju 12Titan adalah sebagai berikut:

1. Direktur Utama (*Owner*) bertugas memimpin, bertanggung jawab dan menetapkan kebijakan-kebijakan perusahaan dengan mengkoordinasikan, mengarahkan dan mengenda1ikan kegiatan-kegiatan perusahaan. Selain itu Direktur bertugas dalam proses penerimaan pesanan dan bertemu konsumen.
2. Bagian Keuangan, mengatur pengelolaan keuangan perusahaan, mencatat setiap pemasukan dan pengeluaran biaya perusahaan.
3. Bagian Penyablonan bertugas melakukan dan mengawasi proses sablon oleh para karyawan terhadap produk berdasarkan pesanan/order.
4. Bagian penjahitan bertugas melakukan proses penjahitan terhadap pesanan yang ada.
5. Bagian pembordiran, melakukan proses bordir pada produk sesuai dengan pesanan.
6. Bagian pemotongan, melakukan proses pemotongan pada bahan baku bersama staf sesuai ukuran pesanan.
7. Bagian Pelayanan, mengurus pesanan dan memberi pelayanan jika ada pertanyaan dari konsumen.
8. Bagian pengemasan, melakukan proses *packing,* pada pesanan yang sudah jadi.
9. Bagian pengiriman dan pengambilan, mengelola pengiriman barang pesanan kepada konsumen.

10. Bagian Gudang, memeriksa dan mencatat barang dan stok bahan di gudang.

2.2 Landasan Teori

Dalam sub bab ini akan dibahas tentang teori-teori yang berhubungan dengan pembangunan *e-commerce* di Pabrik Baju 12Titan.

2.2.1 Pengertian Internet

Internet (*inter-network*) dapat diartikan jaringan komputer luas yang menghubungkan pemakai komputer satu komputer dengan komputer lainnya dan dapat berhubungan dengan komputer dari suatu Negara ke Negara di seluruh dunia, dimana didalamnya terdapat berbagai aneka ragam informasi Fasilitas layanan internet *browsing* atau *surfing* yaitu kegiatan "berselancar" di internet. Kegiatan ini dapat dianalogikan layaknya berjalan–jalan di mal sambil melihat–lihat ke toko-toko tanpa membeli apapun.

Terdapat banyak sekali layanan aplikasi di internet dan masih terus akan berkembang sejalan dengan perkembangan teknologi informasi antara lain *World Wide Web* (WWW), URL, HTTP, E-Mail dan FTP.

2.2.1.1 World Wide Web (WWW)

Awalnya informasi dapat dicari pada internet dengan menggunakan fasilitas *information service* berbasis *archie*, *gopher* dan *WAIS (Wide Area Information System).* Pencarian informasi berdasarkan menu-menu pada sistem tersebut dan output yang dihasilkan berbasis teks. Saat ini dengan teknologi *World Wide Web*, dimungkinkan untuk mengakses informasi secara interaktif, dan bentuk informasinya berupa tampilan grafis maupun teks. Hal ini dimungkinkan dengan adanya *Hypertext*

Transfer Protocol (HTTP) yang digunakan untuk mengakses suatu informasi yang disimpan pada suatu situs web (*website).* Untuk dapat menggunakan sarana ini, dibutuhkan aplikasi *Web Browser.*

2.2.1.2 Universal Resource Locater (URL)

Pada penggunaan *World Wide Web*, penunjukan suatu sumber informasi menggunakan metode *Universal Resource Locater* (URL), yang merupakan konsep penamaan lokais standar suatu file, direktori, komputer, lokasi komputernya dan metode yang digunakan. URL tidak hanya dapat menunjuk ke suatu file tapi dapat juga menunjuk suatu query, dokumen dalam suatu *database*, atau hasil dari perintah atau yang lainnya. Dengan URL ini didefinisikan lokasi dan metode pengaksesan file tersebut.

2.2.1.3 Hyper Text Transfer Protocol (HTTP)

Web merupakan terobosan baru bagi teknologi sistem informasi yang menghubungkan data dari banyak sumber dan layanan yang beragam macamnya di internet. Pengguna tinggal mengklik tombol mousenya pada link-link *hypertext* yang ada untuk melompat ke dokumen diberbagai lokasi di internet. Link-link sendiri bisa mengacu pada dokumen web, *server* FTP *(File Transfer Protocol)*, e-mail ataupun layanan lainnya.

Server dan browser web berkomunikasi satu sama lain dengan protokol yang memang dibuat khusus untuk ini, yaitu HTTP. HTTP bertugas menangani permintaan *(request)* dari browser untuk mengambil dokumen-dokumen web.

2.2.1.4 Electronic mail (E-mail)

Email adalah pengiriman surat elektronik baik berupa teks maupun gabungan dengan gambar yang dikirimkan dari satu alamat *email* ke alamat *email* lainnya di jaringan internet. Seperti layaknya surat biasa pada umumnya, email berfungsi untuk mengirimkan surat atau pesan kepada orang lain. Perbedaan *email* dengan surat biasa adalah *email* sudah tidak lagi membutuhan kertas sebagai media untuk menuliskan pesan. Media yang digunakan adalah data digital melalui jaringan internet.

2.2.1.5 File Transfer Protocol

File Transfer Protocol (FTP) adalah suatu protokol yang berfungsi untuk tukar-menukar file dalam suatu *network* yang menggunakan TCP koneksi bukan UDP. Dua hal yang penting dalam FTP adalah FTP *Server* dan FTP *Client*. FTP *server* adalah suatu *server* yang menjalankan *software* yang berfungsi untuk memberikan layanan tukar menukar *file* dimana *server* tersebut selalu siap memberikan layanan FTP apabila mendapat permintaan *(request)* dari FTP *client*.

FTP *client* adalah komputer yang merequest koneksi ke FTP *server* untuk tujuan tukar menukar *file*. Setelah terhubung dengan FTP *server*, maka *client* dapat men-download, meng-upload, merename, men-delete, dan lain-lain sesuai dengan permission yang diberikan oleh FTP *server*.

Tujuan dari FTP *server* adalah sebagai berikut:

1. *Sharing* data
2. Menyediakan *indirect* atau *implicit remote* computer
3. Menyediakan tempat penyimpanan bagi *Member*
4. Menyediakan transfer data yang *reliable* dan efisien

2.2.2 Sistem Keamanan Jaringan dan Komputer

Setiap melakukan sebuah transaksi atau pertukaran informasi pada jaringan komputer, apalagi informasi tersebut merupakan informasi penting, diperlukan sebuah keamanan yang dapat menjaga informasi tersebut agar tidak sampai hilang atau diakses oleh pihak yang tidak berwenang. Beberapa jenis keamanan jaringan yang dapat digunakan yaitu Seritifikasi digital dan protokol keamanan.

2.2.2.1 IP *Dedicated*

IP *Dedicated* (Internet Protocol) adalah IP *address* yang digunakan secara eksklusif pada satu akun hosting. *Dedicated IP* biasa digunakan untuk jalur transaksi yang menggunakan SSL-enncrypted untuk pengamanan transaksi di internet, biasanya digunakan oleh website yang memiliki *e-commerce* atau penjualan online.

Adapun fungsi dan keuntungan IP *dedicated* adalah :

1. Sebuah website dapat diakses langsung dari IP address tanpa harus mengetikkan Nama Domain.
2. Menggunakan *Dedicated* IP dapat memaksimalkan SERP (Search Engine Result Page).
3. Penggunaan *Dedicated* IP pada banyak website yang Anda miliki dapat digunakan untuk BACKLINK SEO website yang lebih baik di banding dengan menggunakan Shared IP,

4. Penggunaan *Dedicated IP* dapat dipasangkan dengan SSL- enncrypted yang berfungsi menjaga keamanan transaksi melalui internet saat menggunakan Kartu Kredit.
5. Website lebih cepat diakses.

2.2.2.2 Secure Socket Layer (SSL)

SSL adalah suatu protokol yang biasanya digunakan untuk mengamankan komunikasi di internet dan melalui web yang dikembangkan oleh Netscape Communication[5]. SSL dimasukan ke dalam browser dan bekerja di antara protokol komunikasi TCP/IP di internet dan perangkat lunak aplikasi.

Cara komunikasi melalui internet yaitu pesan yang dikirim dilewatkan melalui sebuah soket yang menerjemahkan pesan tersebut dalam TCP/IP. Sebagaimana diketahui, TCP/IP adalah perangkat standar protokol yang digunakan untuk berkomunikasi antara komputer di internet. Jika digunakan SSL, maka soket-soket tersebut akan dipasangi dengan sistem keamanan yang menggunakan kriptografi kunci umum. SSL menggunakan teknologi kunci umum dan sertifikat digital untuk otentikasi *server* dalam suatu transaksi dan melindungi informasi pribadi pada saat melalui satu jaringan ke jaringan lainnya di internet. Meski SSL melindungi informasi pada saat melalui jaringan internet, SSL tidak melindungi informasi pribadi, seperti angka –angka kartu kredit yang tersimpan dalam *server* pedagang. Jika *server* tidak aman maka data-data pribadi tersebut dapat diakses oleh pihak-pihak yang tidak berwenang.

2.2.2.3 HTTPS

HTTPS singkatan dari *HyperText Transport Protocol Secure*, memiliki pengertian sama dengan HTTP tetapi dengan alasan keamanan (*security*), HTTPS memberi tambahan *Secure Socket Layer* (SSL). Umumnya website yang menggunakan HTTPS ini adalah website yang memiliki tingkat kerawanan tinggi yang berhubungan dengan masalah keuangan dan privacy dari pelanggannya seperti website perbankan dan investasi.

Teknologi HTTPS protokol mencegah kemungkinan "dicurinya" informasi penting seperti *credit card* yang dikirimkan selama proses komunikasi berlangsung antara *user* dengan *web server* atau sebaliknya. Secara teknis, website yang menggunakan HTTPS akan melakukan enkripsi terhadap informasi data menggunakan teknik enkripsi SSL. Dengan cara ini meskipun seseorang berhasil "mencuri" data tersebut selama dalam perjalanan user web *server*, orang tersebut tidak akan bisa membacanya karena sudah diubah oleh teknik enkripsi SSL.

Untuk mengenali suatu website menggunakan SSL atau tidak, cukup dengan memperhatikan informasi di depan alamat website pada browser. Apabila alamat website diawali dengan https:// maka bisa dipatikan website tersebut sudah menggunakan teknologi SSL. Cara lain yaitu dengan melihat "icon kunci" di pojok kanan bawah browser, apabila "icon" tersebut terlihat dalam posisi terkunci maka website tersebut aman.

2.2.3 Pengertian *E-commerce*

Electronic Commerce (*E-commerce*) didefinisikan sebagai proses pembelian dan penjualan produk, jasa dan informasi yang dilakukan secara elektronik dengan memanfaatkan jaringan computer yaitu internet.

E-commerce berarti membeli atau menjual secara elektronik, dan kegiatan ini dilakukan pada jaringan internet. *E-commerce* juga dapat berarti pemasangan iklan, penjualan dan dukungan pelayanan yang terbaik menggunakan sebuah webshop 24 jam sehari bagi seluruh pelanggannya[6].

2.2.3.1 Model *E-commerce*

Pada dasarnya *E-commerce* dapat dibagi menjadi dua jenis, yaitu *Business to Business* (B2B) dan *Business to Customer* (B2C, retail). Kedua jenis *E-commerce* ini memiliki karakteristik yang berbeda.

Business to Business secara *online* merupakan bisnis perusahaan dengan perusahaan lainnya. Dengan kata lain, bisnis ini dilakukan dari dan untuk perusahaan. Tujuan dari *Business to Business* adalah membantu bisnis pihak lain untuk memperbaiki kebijakan, prosedur, layanan konsumen, dan operasi umum. Dengan demikian, pihak pemberi jasa berfungsi sebagai konsultan terhadap perusahaan lain. *Business to Business* tidak terbatas pada masalah jasa, tetapi juga dapat berupa produk, misalnya PT. Inti sebagai pembuat perangkat telepon berbisnis dengan PT. Telkom yang merupakan penyelenggara jasa telekomunikasi[5].

Business to Business memiliki karakteristik :

1. *Trading Partners* yang sudah diketahui dan umumnya memiliki hubungan (*relationship*) yang cukup lama. Informasi hanya dipertukarkan dengan partner tersebut . Dikarenakan sudah mengenal lawan komunikasi, maka jenis informasi yang dikirim dapat disusun sesuai dengan kebutuhan dan kepercayaan (*trust*).
2. Pertukaran data (*data exchange*) berlangsung berulang kali dan secara berkala dengan format data yang sudah disepakati bersama. Dengan kata lain, pelayanan yang digunakan sudah khusus. Hal ini memudahkan pertukaran data untuk dua entitas yang menggunakan standar yang sama.
3. Salah satu pelaku dapat melakukan inisiatif untuk mengirimkan data, tidak harus menunggu partnernya.
4. Modal yang umum digunakan adalah *peer to peer*, dimana *processing intelligence* dapat didistribusikan di kedua pelaku bisnis.

Topik yang juga termasuk di dalam *Business to Business* adlah *Electronic/Internet Procurement* dan *Enterprise Resource Planning*. Hal ini adalah implementasi penggunaan teknologi informasi pada perusahaan dan pada manufacturing.

Sedangkan untuk *Business to Customer* memiliki karteristik sebagai berikut:

1. Terbuka untuk umum, dimana informasi disebarkan ke umum.
2. Pelayanan yang diberikan bersifat umum dengan mekanisme yang dapat digunakan oleh khalayak ramai.

3. Pelayanan diberikan berdasarkan permohonan (*on Demand*). Konsumen melakukan inisiatif dan produsen harus siap memeberikan respon sesuai dengan permohonan.
4. Pendekatan *Client/Sever* sering digunakan dimana mengambil asumsi client (konsumen) menggunakan system yang minimal (berbasis web) dan processing business procedure diletakan di sisi *server*.

Business to Customer memiliki permasalahan berbeda. Mekanisme untuk mendekati konsumen pada saat ini menggunakan bermacam-macam pendekatan misalnya dengan "*Electronic Shopping Mall*" atau menggunakan konsep "Portal".

2.2.3.2 Manfaat *E-commerce*

Manfaat yang diperoleh bagi organisasi pemilik *E-commerce*:

1. Memperluas lokasi di mana suatu perusahaan dapat memperoleh informasi *(market place)* hingga ke pasar nasional dan internasional.
2. Dengan biaya yang tidak begitu besar, sebuah perusahaaan dapat dengan mudah menemukan lebih banyak pelanggan, *supplier* yang lebih baik dan partner bisnis yang paling cocok dari seluruh dunia.
3. *E-commerce* menurunkan biaya pembuatan, pemrosesan, pendistribusian, penyimpanan dan pencarian informasi yang menggunakan kertas.
4. *E-commerce* memungkinkan pengurangan *inventory* dan *overhead* dengan menyederhanakan *supply chain management* tipe "*pull*". Dalam *supply chain management* tipe "*pull*", proses dimulai dari pesanan pelanggan serta digunakan *manufacturing just-in-time.*
5. *E-commerce* mengurangi waktu penerimaan produk dan jasa.

6. *E-commerce* memperkecil biaya telekomunikasi. Karena internet lebih murah dibandingkan VAN.
7. Akses informasi menjadi lebih cepat.
8. Biaya transportasi dan fleksibilitas bertambah.

Manfaat bagi konsumen antara lain:

1. *E-commerce* memungkinkan pelanggan untuk berbelanja atau melakukan transaksi selama 24 jam sehari sepanjang tahun dari hampir setiap lokasi.
2. *E-commerce* memberikan lebih banyak pilihan kepada pelanggan, mereka bisa memilih berbagai produk dari banyak *vendor*.
3. *E-commerce* menyediakan produk dan jasa yang tidak mahal kepada pelanggan dengan cara mengunjungi banyak tempat dan melakukan perbandingan secara cepat.
4. Pelanggan bisa menerima informasi yang *relevan* secara detil dalam hitungan detik, bukan lagi hari atau minggu.
5. *E-commerce* memberi tempat kepada para pelanggan untuk berinteraksi dengan pelanggan lain di *electronic community* dan bertukar pikiran serta pengalaman.

2.2.3.3 Fitur-fitur Utama *E-commerce*

Sebuah situs *E-commerce* biasanya memiliki fitur-fitur utama sebagai berikut:

1. *Etalase*/ Halaman *Display* yang memuat nama, deskripsi dan harga barang/ jasa yang ditawarkan.
2. Katalog, yang bisa dieksplorasi isinya dengan mudah untuk mencari produk/jasa yang kita inginkan, umumnya di fasilitasi dengan fungsi *Search*/Cari.
3. S*hopping Cart* (keranjang belanja), yaitu tempat kita menaruh barang-barang atau jasa yang kita inginkan. Memuat Nama barang, *Quantity* dan Harga serta Jumlah bayar, umumnya dilengkapi dengan *automatic calculation* yang dapat memunculkan informasi nominal pembayaran yang harus dilakukan.
4. *Payment Methode*, umumnya menawarkan tiga metode cara bayar, yaitu *Cash/Transfer/Debit,* Kredit dengan Kartu Kredit dan C.O.D (*Cash On Delivery*).
5. Sistem Pengiriman *(delivery methode)*, umumnya dengan pihak ketiga/perusahaan jasa pengiriman barang dengan berbagai layanannya yang dapat dipilih oleh pembeli misalnya layanan kiriman sampai pada hari yang sama, keesokan harinya dan lain-lain.
6. *Contact Link*, berisi alamat dan nomor telepon atau saluran lain (*email*/IM) yang dapat digunakan untuk komunikasi antara pembeli dan penjual, umumnya disertakan pula *Contact Form* yang terdiri dari *field-field* untuk memasukkan identitas pengirim (nama, *email/website*) serta berita/pesan yang ingin disampaikan. Isi pesan yang disampiakan melalui *Contact*

Form umumnya dapat diteruskan kepada penjual/pengelola/pemilik web dalam bentuk email.

2.2.4 Basis Data

Basis data (*database)* merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya yang disimpan sedemikian rupa dan tanpa pengulangan (*redundansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan[1].

*S*istem basis data adalah suatu sistem informasi yang mengintegrasikan dari kumpulan data yang saling berhubungan satu dengan yang lainnya dan membuat tersedia untuk beberapa aplikasi yang bermacam-macam di dalam suatu organisasi.

2.2.4.1 Bahasa basis data

Bahasa basis data terdiri dari :

1. *Data Definition Language* (DDL), merujuk pada kumpulan perintah yang dapat digunakan untuk mendefinisikan objek – objek basis data, seperti membuat sebuah tabel basis data atau indeks primer atau sekunder.
2. *Data Manipulation Language* (DML), mengacu pada kumpulan perintah yang dapat digunakan untuk melakukan manipulasi data, seperti penyimpanan data ke suatu tabel, kemudian mengubahnya dan menghapusnya atau hanya sekedar menampilkannya kembali.

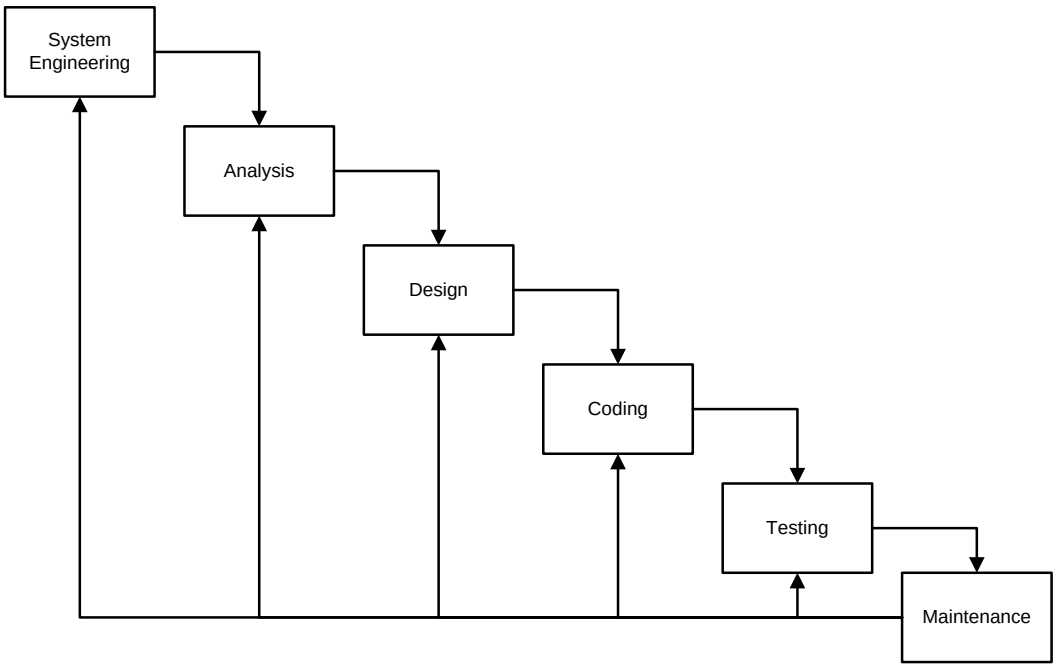
2.2.4.2 SQL (Structured Query Language)

SQL adalah bahasa yang digunakan untuk berkomunikasi dengan *database*. Menurut ANSI (American National Standards Institute), bahasa ini merupakan standar untuk relational *database* management systems (RDBMS).

Pernyataan-pernyataan SQL digunakan untuk melakukan beberapa tugas seperti : *update* data pada *database* atau menampilkan data dari *database*. Hampir semua software *database* mengimplementasikan bahasa SQL sebagai komponen utama dari produknya, salah satunya MySQL.

2.2.5 Metode Pembangunan Sistem

Metode yang digunakan yaitu metode *waterfall*, metode ini sering disebut sebagai pendekatan air terjun (*Waterfall Approach).* Metode *waterfall* terkadang juga disebut model *Classic Life Cycle* (CLC), model ini terhubung secara sistematik memulai teknis pengembangan perangkat lunak yang dimulai sari level sistem dengan tahapan-tahapan selanjutnya yaitu system *engginering*, *analysis*, *design*, *coding*, *testing* dan *maintenance*, dimana konsep dari metode ini adalah bagaimana melihat suatu masalah secara sistematis dan terstruktur[4]. Metode *Waterfall* dapat dilihat pada gambar 2.2.



Gambar 2.2 Metode Waterfall

Penjelasan metodologi *waterfall* :

1. System *engginering*, adalah pembuatan suatu perangkat lunak yang merupakan bagian terbesar dari pengerjaan suatu proyek. Untuk pekerjaan dimulai dengan menempatkan segala hal yang diperlukan dalam pelaksanaan proyek.

2. *Analysis*, merupakan tahapan dimana system engginering menganalisis hal-hal yang diperlihatkan dalam pelaksanaan proyek pembuatan atau pengembangan perangkat lunak yang bertujuan untuk memahami sistem yang ada, mengidentifikasi masalah dan mencari solusinya.

3. *Design*, tahapan ini merupakan tahap penerjemah dari keperluan atau data yang telah dianalisis kedalam bentuk yang mudah dimengerti oleh pemakai (user).

4. *Coding*, yaitu menterjemahkan data atau pemecahan masalah yang dirancang ke dalam bahasa pemrograman yang telah di tentukan.

5. *Testing*, program selesai dibuat, maka berikutnya adalah uji coba terhadap program tersebut.

6. *Maintenance* atau pemeliharaan yaitu penerapan secara keseluruhan disertai pemeliharaan jika terjadi perubahan struktur baik dari segi *software* maupun *hardware*.

2.2.6 Metode Berorientasi Objek

Pada metode pengembangan system ini akan menggunakan *object oriented* (OO) yang merupakan paradigma dalam rekayasa *software* yang didasarkan pada objek dan kelas. *Object-oriented* merupakan metodologi terbaik dalam rekayasa *software* dan mencakup bidang aplikasi yang sangat luas. Karena luasnya cakupan object-oriented maka terdapat beberapa hal yang membingungkan berkenaan dengan istilah dan konsep *object-oriented*. Istilah dan konsep yang berkenaan dengan *object-oriented* ini adalah :

1. *Object-oriented Analysis* adalah metode analisis yang memeriksa requirement (syarat/keperluan yang harus dipenuhi suatu sistem) dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup permasalahan.
2. *Object-oriented Design* adalah metode untuk mengarahkan arsitektur *software* yang didasarkan pada manipulasi objek-objek system atau subsistem.

Object oriented analysis dan *design* (OOAD) merupakan pendekatan yang menekankan pada solusi logis berbasis objek. Konsep dasar dalam OOAD mencakup objek dan kelas.

1. Objek

   Objek adalah "benda", secara fisik atau konseptual dapat kita temui di sekeliling kita. *Hardware, Software*, dokumen, manusia dan bahkan konsep semuanya adalah konsep objek. Sebuah objek memiliki keadaaan sesaat (*state*) dan perilaku (*behaviour*). *State* dari sebuah objek adalah kondisi objek tersebut atau himpunan dari keadaan yang menggambarkan objek tersebut.

2. Kelas

   Kelas adalah definisi umum untuk himpunan objek sejenis. Kelas menetapkan spesifikasi prilaku dan atribut objek-objek tersebut.

Object oriented merupakan metode yang paling baik dalam rekayasa *software* diantaranya *procedure-oriented, object-oriented, data struktur-oriented, data flow-oriented*, dan *constraint-oriented*. Sehingga dengan metode object-oriented ini dapat diaplikasikan dalam seluruh ruang lingkup rekayasa *software*.

Untuk memhami keunggulan OOAD maka kita harus memahami masalah yang dihadapi antara lain :

1. *Software* sulit untuk dimodifikasi bila memerlukan pengembangan.
2. Proses pembuatan *software* memerlukan waktu yang cukup lama sehingga kadang kala melebihi anggaran dalam pembuatannya.

3. Para programmer selalu membuat *software* dari dasar karena tidak adanya kode yang bias digunakan ulang (*reuse*).

Dari beberapa keunggulan diatas, dengan menggunakan object-oriented maka sangat menguntungkan bagi programmer karena programmer dapat mendesign program dalam bentuk objek-objek dan hubungan antara objek-objke tersebut untuk kemudian dimodelkan dalam system nyata. Keuntungan yang lain adalah proses pembuatan *software* dapat dilakukan dengan lebih cepat karena *software* dibangun dalam objek-objek standard, sehingga dapat digunakan secara berulang-ulang.

Bahasa pemograman yang menggunakan object-oriented diantaranya yang terkenal C++, Visual Basic dan Java.

2.2.7 Pengertian Unified Modeling Language (UML)

Unified Modeling Langauge (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek[2].

UML merupakan gabungan dari metode booch, Rumbaugh (OMT) dan Jacobson. Tetapi UML ini akan mencakup lebih luas dari OOA dan OOD. Pada pertengahan dan pengembangan UML dilakukan standarisasi proses dengan OMG (*Object Management Group*) dengan harapan UML akan menjadi bahasa standar pemodelan pada masa yang akan datang.

UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan preoses. Bahasa

pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendisain secara cepat.

2.2.8 Bagian-bagian dari UML

Bagian-bagian utama dari UML adalah *view, diagram, model elemen,* dan *general mechanism*.

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat grafik, tetapi merupakan suatu abstraksi yang berisi sejumlah diagram.

Beberapa jenis *view* dalam UML antara lain : *use case view, logical view komponen view, concurennses view dan deployment view.*

1. *Use Case View*

   Use Case View digunakan untuk memahami dan menggunakan sistem yang kita modelkan. *View* ini melihat pada bagaimana *actor* dan *use case* berinteraksi.

2. *Logical View*

   Logical View mengarah pada persyaratan (*requierments*) fungsional sistem. *View* ini melihat pada kelas-kelas dan hubungan antar kelas-kelas tersebut.

3. *Component View*

   Component View mengarah pada pengaturan software. *View* ini mengandung informasi mengenai komponen-komponen *software*, komponen tereksekusi (*executable*) dan *library* dan sistem yang kita modelkan.

4. *Deployment View*

   Deployment View memperlihatkan pemetaan setiap proses ke dalam hardware. View ini paling bermanfaat ketika kita membuat model suatu sistem yang diterapkan dalam lingkungan arsitektur yang terdistribusi dimana kita menerapkan aplikasi dan *server* pada lokasi yang berbeda.

Diagram berbentuk grafik yang menunjukan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu *view* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu.

Adapun jenis diagram antara lain *use case diagram, class diagram, activity diagram, sequence diagram, package diagram* dan *deployment diagram.*

2.2.8.1 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya.Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam

dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal.

Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

2.2.8.2 *Class* Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

a. Nama (dan *stereotype*)
b. Atribut
c. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan

2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.
3. *Public*, dapat dipanggil oleh siapa saja

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class.* Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*.

2.2.8.3 *Activity* Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk

menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

2.2.8.4 *Sequence* Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. *Sequence* diagram terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

2.2.8.5 *Package* Diagram

Package diagram adalah sebuah bentuk pengelompokkan yang memungkinkan untuk mengambil setiap bentuk di UML dan mengelompokkan elemen-elemennya dalam tingkatan unit yang lebih tinggi[2].

Package diagram sangat berguna pada sistem berskala besar untuk mendapatkan sebuah gambaran ketergantungan antar elemen-elemen besar dalam sistem.

2.2.8.6 *Deployment* Diagram

Deployment/*physical* diagram menggambarkan detail bagaimana komponen di-deploy dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, *server* atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisikal.

Sebuah *node* adalah *server*, workstation, atau piranti keras lain yang digunakan untuk men-deploy komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan requirement dapat juga didefinisikan dalam diagram ini.

2.2.9 Model-View-Controller (MVC)

MVC adalah sebuah software yang memisahkan antara aplikasi logika dengan presentasi pada halamn web sehingga hal ini menyebabkan halaman web akan mengandung kode yang sedikit karena sudah terjadi pemisahan antara tamilan dan pemrograman[7].

1. **Model**

   Model merupakan bagian yang mengatur struktur data. Secara spesifik *class* model akan mengandung fungsi kode yang akan membantu dalam segala proses yang berhubungan dengan *database* seperti memasukan, mengubah, menampilkan dan menghapus data pada *database*.

2. **View**

   View merupakan informasi yang disampaikan ke pengguna. Sebuah view biasanya berupa halaman web, atau dapat berupa fragmen halaman seperti *header* dan *footer*.

3. **Controller**

   Controller berperan sebagai perantara Model dan View yang mengatur semua sumber yang dibutuhkan untuk memproses halaman HTTP dalam membuat halaman web.

2.2.10 *Framework*

Pengertian *framework* secara umum adalah sebuah susunan atau rangkaian kerja yang tetap dan dibuat sedemikian rupa yang kemudian dapat digunakan kembali dalam sebuah aktifitas kerja yang lain tetapi tetap dalam satu area kerja dengan rangkaian kerja yang sebelumnya.

Framework web adalah kumpulan sebuah maupun banyak modul-modul dalam bentuk *class library* yang dapat digunakan kembali untuk membentuk sebuah aplikasi web yan lebih besar dengan memanfaatkan modul-modul *class library* tersebut[7].

2.2.11 Perangkat Lunak Pendukung

2.2.11.1 HTML

HTML (*Hyper Text Markup Language*) adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web dan menampilkan berbagai informasi di dalam sebuah *browser* Internet. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML *(Standard Generalized Markup Language)*, HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat

ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh World *Wide Web Consortium* (W3C).

HTML berupa kode-kode tag yang menginstruksikan browser untuk menghasilkan tampilan sesuai dengan yang diinginkan. Sebuah file yang merupakan file HTML dapat dibuka dengan menggunakan browser web seperti Mozilla Firefox atau Microsoft Internet Explorer. HTML juga dapat dikenali oleh aplikasi pembuka email ataupun dari PDA dan program lain yang memiliki kemampuan browser.

HTML dokumen tersebut mirip dengan dokumen teks biasa, hanya dalam dokumen ini sebuah teks bisa memuat instruksi yang ditandai dengan kode atau lebih dikenal dengan TAG tertentu. Sebagai contoh jika ingin membuat teks ditampilkan menjadi tebal seperti: *CETAK MIRING*, maka penulisannya dilakukan dengan cara: CETAK MIRING. Tanda digunakan untuk mengaktifkan instruksi cetak miring (*italic*), diikuti oleh teks yang ingin dimiringkan, dan diakhiri dengan tanda untuk menonaktifkan cetak miring tersebut.

Secara garis besar, terdapat beberapa jenis elemen dari HTML:

1. *Structural*, tanda yang menentukan level atau tingkatan dari sebuah teks (contoh, <h1>Golf</h1> akan memerintahkan browser untuk menampilkan “Golf” sebagai teks tebal besar yang menunjukkan sebagai Heading 1.
2. *Presentational*, tanda yang menentukan tampilan dari sebuah teks tidak peduli dengan level dari teks tersebut (contoh, boldface akan

menampilkan bold. Tanda presentational saat ini sudah mulai digantikan oleh CSS dan tidak direkomendasikan untuk mengatur tampilan teks.

3. *Hypertext,* tanda yang menunjukkan pranala ke bagian dari dokumen tersebut atau pranala ke dokumen lain (contoh, unikomakan menampilkan tulisan unikom sebagai sebuah hyperlink ke URL tertentu), *Elemen widget* yang membuat objek-objek lain seperti tombol (<button>), list (<li>), dan garis horizontal (<hr>). Selain markup presentational, markup yang lin tidak menentukan bagaimana tampilan dari sebuah teks. Namun untuk saat ini, penggunaan tag HTML untuk menentukan tampilan telah dianjurkan untuk mulai ditinggalkan dan sebagai gantinya digunakan *Cascading Style Sheets.*

Contoh HTML Sederhana :

```
<html>
  <head>
    <title>Unikom</title>
  </head>
  <body>
    <p>Selamat Datang</p>
  </body>
</html>
```

2.2.11.2 PHP

PHP merupakan singkatan dari "*PHP: Hypertext Preprocessor*", adalah sebuah bahasa scripting yang terpasang pada HTML. Sebagian besar sintaks mirip

dengan bahasa C, Java, asp dan Perl, ditambah beberapa fungsi PHP yang spesifik. Tujuan utama bahasa ini adalah untuk memungkinkan perancang web untuk menulis halaman web dinamik dengan cepat[3].

Pada saat ini bahasa PERL dan CGI sudah jauh ketinggalan jaman sehingga sebagian besar designer web banyak beralih ke bahasa *server*-side scripting yang lebih dinamis seperti PHP.

Seluruh aplikasi berbasis web dapat dibuat dengan PHP. Namun kekuatan yang paling utama PHP adalah pada konektivitasnya dengan system *database* di dalam web. Sistem *database* yang dapat didukung oleh PHP adalah :

1. Oracle
2. MySQL
3. Ms. Access
4. Sybase
5. PostgreSQL

PHP dapat berjalan di berbagai system operasi seperti windows 98/NT, UNIX/LINUX, solaris maupun macintosh. Keunggulan lainnya dari PHP adalah PHP juga mendukung komunikasi dengan layanan seperti protocol IMAP, SNMP, NNTP, POP3 bahkan HTTP.

2.2.11.3 MySQL

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial. MySQL

sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Keandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query* MySQL bisa sepuluh kali lebih cepat dari *Postgre*SQL dan lima kali lebih cepat dibandingkan *Interbase*. Selain itu MySQL juga memiliki beberapa keistimewaan, antara lain :

1. *Portability*

   MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga*, dan masih banyak lagi.

2. *Open Source*

   MySQL didistribusikan secara *open source* (gratis), dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.

3. *Multiuser*

   MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. *Performance tuning*

   *My*SQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. *Column types*

   MySQL memiliki tipe kolom yang sangat kompleks, seperti *signed/unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.

6. *Command dan functions*

   MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam *query*.

7. *Security*

   MySQL memiliki beberapa lapisan sekuritas seperti level *subnetmask*, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta *password terenkripsi*.

8. *Scalability dan limits*

   MySQL mampu menangani *database* dalam skala besar, dengan jumlah *records* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. *Connectivity*

   MySQL dapat melakukan koneksi dengan *client* menggunakan *protokol* TCP/IP, *Unix soket* (UNIX), atau *Named Pipes* (NT).

10. *Localization*

MySQL dapat mendeteksi pesan kesalahan pada client dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk didalamnya.

11. *Interface*

    MySQL memiliki *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

12. *Clients dan tools*

    MySQL dilengkapi dengan berbagai *tool* yang dapat digunakan untuk administrasi *database,* dan pada setiap *tool* yang ada disertakan petunjuk online.

13. Struktur tabel

    MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan *database* lainnya semacam *PostgreSQL* ataupun *Oracle*.

2.2.11.4 PHPDesigner

PhpDesigner merupakan aplikasi khusus untuk membangun sebuah aplikasi berbasis web, khususnya PHP. Aplikasi ini dapat mengenali sintaks html, javascript, CSS yang sering digunakan dalam membangun sebuah aplikasi berbasis web. Selain itu, aplikasi ini juga mengenali sintaks bahasa java dan C#.

2.2.11.5 Adobe Photoshop

Adobe photoshop adalah sebuah aplikasi pengolah gambar yang digunakan untuk membuat sebuah gambar atau memanipulasi foto yang sudah ada. Photosop dilengkapi dengan berbagai fitur yang dapat memudahkan pengguna untuk berkreasi dengan desain sesuai keinginan. Selain itu photoshop dapat membaca, menyimpan dan mengubah berbagai format data gambar seperti JPEG, GIF, PNG. Photoshop juga sangat berguna untuk membuat tampilan desain sebuah situs.

2.2.11.6 WAMP*Server*

WAMP merupakan akronim dari Windows, Apache, MySQL, dan PHP. WAMP digunakan untuk membangun sebuah host lokal (*localhost*) untuk membuat simulasi sebuah website yang memiliki *server side script* sebelum dipublikasikan pada *server* yang sebenarnya.

Jaringan yang dibentuk hanya untuk wilayah lokal saja atau *Local Area Network* (LAN) sehingga hanya mampu melakukan hubungan antar komputer yang ada dalam jaringan tersebut dan tidak dapat berhubungan antar dunia karena localhost ini bukan internet. Biasanya *localhost* menggunakan default IP address yaitu 127.0.0.1 atau pada saat kita masukkan IP itu dalam browser maka akan muncul *server* itu dalam bentuk *software server*, seperti jika kita menggunakan WAMP, maka akan muncul adalah halaman WAMP tersebut.

2.2.11.7 Codeigniter

Codeigniter adalah sebuah *framework* untuk web yang dibuat dalam format PHP. Format yang dibuat ini selanjutnya dapat digunakan untuk membuat sistem aplikasi web yang kompleks[7]. Codeigniter dapat mempercepat proses pembuatan web, karena semua *class* dan modul yang dibutuhkan sudah disediakan dan dapat dapat langsung digunakan pada aplikasi web yang akan dibangun.