

BAB II

LANDASAN TEORI

2.1 Digital Menu

Menurut Mahmood (2015:533), restoran memiliki opsi untuk menampilkan menu mereka, mengubah item menu, dan menyorot item khusus dengan menggunakan papan menu digital. mereka juga membantu dalam mengubah harga secara instan atau menawarkan diskon khusus. papan menu digital membantu pemilik restoran untuk menyediakan hiburan bagi pelanggan sementara mereka menunggu pesanan mereka di restoran cepat saji. mereka juga dapat membantu memberikan informasi khusus seperti nilai gizi dari item menu dan dapat digunakan untuk penjualan sugestif

2.2 Pengertian Internet

Menurut Dr. Rusman, M.Pd. (2017:235) *internet*, atau *international networking* di definisikan dua komputer atau lebih yang memiliki konektivitas membentuk jaringan komputer hingga meliputi jutaan computer di dunia secara global (internasional), yang saling berinteraksi dan bertukar informasi.

Pengertian internet juga mencakup perangkat lunak berupa data yang dikirim dan disimpan sewaktu-waktu dapat diakses. Beberapa computer yang saling berhubungan satu sama lain dapat mencipatakan fungsi *sharing* yang secara sederhana hal ini dapat disebut sebagai jaringan (*networking*).

Fungsi *sharing* yang tercipta melalui jaringan (*networking*) tidak hanya mencakup fasilitas yang sangat dan sering dibutuhkan, seperti *printer* atau *modem*, maupun yang berkaitan dengan data atau program aplikasi tertentu. Melihat pengertian-pengertian diatas internet memungkinkan suatu program untuk melakukan *sharing* dengan siapa pun, dimana pun, dan kapan pun termasuk pengimplementasian program *e-learning* model CBI.

Sebuah sistem komputer yang terhubung secara langsung ke jaringan memiliki nama domain dan alamat IP (*Internet Protocol*) dalam bentuk numerik

dengan format tertentu sebagai pengenalan. Internet juga memiliki *gateway* ke jaringan dan layanan yang berbasis protokol lainnya.

2.3 Pengertian Website

Website atau situs web menurut Yuhefizar, S.kom dan Ir. HA Mooduto Rahmat Hidayat, ST. (2009:2) adalah keseluruhan halaman-halaman web yang terdapat dalam sebuah domain yang mengandung informasi. Sebuah website biasanya dibangun atas banyak halaman web yang saling berhubungan. Hubungan antara satu halaman web dengan halaman web yang lainnya disebut dengan hyperlink, sedangkan teks yang dijadikan media penghubung disebut hypertext.

Domain adalah nama unik yang dimiliki oleh sebuah institusi sehingga bisa diakses melalui internet, misalnya lintau.com, yahoo.com, google.com, ephi.web.id dan lain-lain. Untuk mendapat sebuah domain kita harus melakukan register pada register-register yang ditentukan.

Istilah lain yang sering ditemui sehubungan dengan website adalah homepage. Homepage adalah halaman awal sebuah domain. Misalnya, Anda Membuka website lintau.com, halaman pertama yang muncul disebut dengan homepage, jika Anda meng-klik menu-menu yang ada dan meloncat ke lokasi yang lainnya, disebut web page, sedangkan keseluruhan isi/conten domain disebut website.

2.3.1 Pengertian Domain

Menurut Rahmat Hidayat (2010:9) Domain adalah alamat unik di dunia internet yang digunakan untuk mengidentifikasi sebuah website, atau dengan kata lain domain adalah alamat yang digunakan untuk mencari dan menemukan sebuah website pada dunia internet. Nama domain diperjual belikan secara bebas di internet dengan status sewa tahunan. Nama domain sendiri mempunyai identifikasi ekstensi/akhiran sesuai dengan kepentingan dan lokasi keberadaan website tersebut. Apabila dibahas lebih lanjut mengenai pengertian domain, kita perlu sedikit mengetahui tentang bagaimana sebuah host di lingkungan internet diakses.

Internet terdiri dari jutaan computer sebagai host yang tersebar di seluruh dunia yang semuanya saling bergubungan melalui suatu bentuk jaringan dengan hirarki tertentu. Host – host tersebut saling berkomunikasi melalui suatu protocol standar yang disebut TCP/IP (Transmission Control Protocol/Internet Protocol). Agar setiap computer yang membentuk jaringan internet dapat berkomunikasi satu sama lain, masing – masing haruslah memiliki alamat tertentu. Alamat ini haruslah uni, jadi tidak boleh ada dua host yang memiliki alamat yang sama.

Sistem pengalamatan yang digunakan berupa kombinasi 4 deret bilangan antara 0 s/d 255 yang masing – masing dipisahkan oleh tanda titik (.), mulai dari 0.0.0.1 hingga 255.255.255.255. Deretan angka – angka ini dikenal sebagai alamat IP (IP address). Setiap host yang tersambung dalam jaringan internet harus memiliki alamat IP sebagai pengenalan agar dapat berkomunikasi dengan host lain dalam jaringan. Pengalamatan berbasis IP ini memungkinkan internet mengalami lebih dari 4 milyar host. Pada kenyataannya, tidak semua kombinasi alamat IP bisa dipergunakan. Ada beberapa kombinasi khusus yang dicadangkan untuk keperluan tertentu sehingga tidak boleh digunakan untuk keperluan pengalamatan. Contohnya IP 127.0.0.1 yang diperlukan untuk menunjuk (lookup) ke host lokal. Walaupun secara teknis sistem pengalamatan berbasis IP cukup handal, tetapi masih memiliki kelemahan. Otak manusia umumnya tidak mudah menghafal dan mengingat kombinasi angka dalam jumlah besar. Solusinya adalah mengasosiasikan nomor IP dalam kombinasi huruf yang membentuk sebuah nama yang mudah diingat. Nama host sebagai pengenalan jaringan internet inilah yang disebut sebagai domain, sedangkan sistem pengalamatan berbasis domain dikenal sebagai Domain Name Service (DNS).

2.4 HTML

Menurut Rio Jumardi, S. T., M.Eng. (2019:15) *Hypertext Markup Language* (HTML) adalah sebuah bahasa yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web Internet dan pemformatan hiperteks sederhana yang ditulis dalam bekas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata

lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan dalam format ASCII normal sehingga menjadi halaman web dengan perintah-perintah HTML.

Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (*Standard Generalized 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).

2.5 CSS

Menurut Ardhana (2012:108) CSS merupakan salah satu bahasa pemrograman web untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. Pada umumnya CSS dipakai untuk menformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML.

2.6 Bootstrap

Menurut Jubille Enterprise (2016:1) Bootstrap adalah *framework front-end* yang intuitif dan powerful untuk pengembangan aplikasi web yang lebih cepat dan mudah. Bootstrap menggunakan HTML, CSS dan JavaScript. Bootstrap dikembangkan oleh Mark Otto dan Jacob Thornton dari Twitter. *Framework* ini diluncurkan sebagai produk *open source* pada Agustus 2011 di GitHub. Bootstrap memiliki fitur – fitur komponen *interface* yang bagus seperti *Typography*, *Forms*, *Buttons*, *Tables*, *Navigation*, *Dropdowns*, *Alerts*, *Modals*, *Tabs*, *Accordion*, *Carousel* dan lain sebagainya.

2.7 PHP

Menurut Anhar (2010:23) PHP Hypertext Preprocessor (PHP) adalah bahasa pemrograman web berupa script yang dapat diintegrasikan dengan

HTML. Untuk mempelajari PHP, kita harus mempelajari dahulu dasar pemrograman PHP. Beberapa hal yang akan memudahkan kita dalam mempelajari dasar pemrograman PHP.

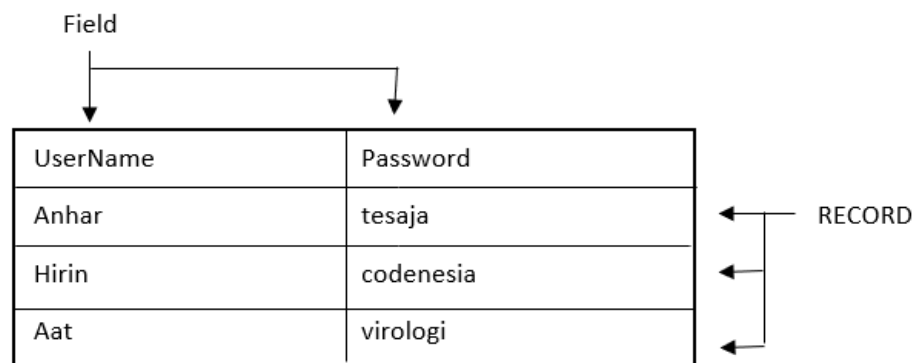
2.8 MySQL

Menurut Anhar (2010:45) My Structure Query Language (MySQL) adalah salah satu DataBase Management System (DBMS) dari sekian banyak DBMS seperti Oracle, MS SQL, Postagre SQL, dan lainnya. MySQL berfungsi untuk mengolah database menggunakan bahasa SQL. MySQL bersifat open source sehingga kita bisa menggunakannya secara gratis. Pemrograman PHP juga sangat mendukung/support dengan database MySQL.

2.9 Database

Menurut Anhar (2010:45) Database adalah sekumpulan tabel-tabel yang berisi data dan merupakan kumpulan dari field atau kolom. Struktur file yang menyusun sebuah database adalah Data Record dan Field.

- Data adalah satu satuan informasi yang akan diolah. Sebelum diolah, data dikumpulkan di dalam suatu file database.
- RECORD adalah data yang isinya merupakan satu kesatuan seperti NamaUser dan Password. Setiap keterangan yang mencakup NamaUser dan Password dinamakan satu record. Setiap record diberi nomor urut yang disebut nomor record (Record Number).
- FIELD adalah sub bagian dari Record. Dari contoh isi record di atas maka terdiri dari 2 field, yaitu: field NamaUser dan Password



Gambar 2.1 Tabel database UserName Password

2.10 Notepad++

Menurut Ardhana (2012:24) notepad++ adalah salah satu program yang digunakan untuk melakukan editor seperti HTML, PHP, Java Scripts, dan lain-lain.

2.11 Unified Modelling Language (UML)

Menurut Gandharba Swain (2010:24) *UML* adalah bahasa kosa kata dan aturannya memfokuskan representasi konseptual dan fisik dari suatu sistem. Sistem intensif perangkat lunak memerlukan bahasa yang membahas berbagai pandangan arsitektur sistem. Kosa kata dan aturan bahasa seperti UML memberi model yang baik, tetapi tidak memberi tahu model apa yang harus anda buat dan kapan harus dibuat. Proses yang terdefinisi dengan baik akan memandu dalam memutuskan artefak apa yang akan diproduksi, kegiatan apa dan apa yang digunakan pekerjaan untuk menciptakannya dan mengelolanya, dan bagaimana menggunakan artefak untuk menggunakan artefak untuk mengukur dan mengendalikan proyek secara keseluruhan.




2.11.1 Use Case Diagram

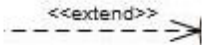
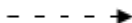
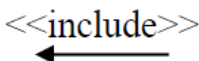
Menurut Gandharba Swain (2010:59) *Use case diagram* adalah sekumpulan urutan aksi yang melakukan hasil yang dapat diamati oleh aktor. Secara grafis *use case* diwakili oleh elips. Setiap *use case* harus memiliki nama yang membedakannya dari *use case* lainnya. Bisa berupa nama atau nama jalur

sederhana. Dalam nama path, nama *use case* harus diawali dengan nama paket itu sendiri.

Simbol-simbol yang digunakan pada use case diagram ditunjukkan pada tabel berikut.

Tabel 2.1 Simbol-simbol pada *Use Case diagram*

Nama	Simbol	Deskripsi
<i>Use Case</i>		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal di frase nama <i>Use Case</i> .
Aktor		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor hanya memberikan informasi ke sistem, aktor hanya menerima informasi dari sistem, aktor memberikan dan menerima informasi ke sistem dan dari sistem.
Asosiasi		Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor. Asosiasi merupakan hubungan statis antar elemen yang menggambarkan elemen yang memiliki atribut berupa elemen lain, atau elemen yang harus mengetahui eksistensi elemen lain.







Ekstensi		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek. Biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan. Misalnya arah panah mengarah pada <i>use case</i> yang ditambahkan, biasanya <i>use case</i> yang menjadi <i>extend</i> -nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.
Generalisasi		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya, misalnya : arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum). Generalisasi merupakan hubungan hirarkis antara elemen. Elemen dapat mewarisi semua atribut dan metode elemen asalnya dan menambah fungsionalitas baru.
Include		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat.

2.11.2 Activity Diagram

Menurut Gandharba Swain (2010:24) *Activity diagram* menunjukkan aliran dari aktivitas ke aktivitas dalam suatu sistem. Suatu aktivitas menunjukkan serangkaian aktivitas, aliran berurutan atau bercabang dari aktivitas ke aktivitas, dan objek yang bertindak dan ditindak lanjuti. *Activity diagram* sangat penting dalam memodelkan fungsi suatu sistem. *Activity diagram* menekankan aliran kontrol diantara objek.

Simbol-simbol yang digunakan pada *activity diagram* ditunjukkan pada tabel dibawah ini.

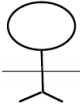

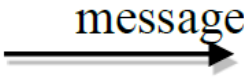

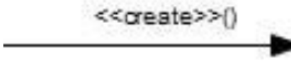
Tabel 2.2 Simbol-simbol pada *activity diagram*

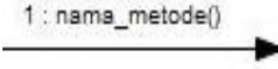
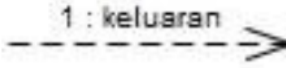
Nama	Simbol	Deskripsi
Status Awal		Status awal aktivitas sistem, sebuah aktivitas memiliki sebuah status awal
Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
percabangan/ <i>decision</i>		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan <i>/ join</i>		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.11.3 Sequence Diagram

Menurut Gandharba Swain (2010:59) *Sequence diagram* adalah diagram yang memuat interaksi yang menekankan urutan waktu pesan. *Sequence diagram* menunjukkan serangkaian objek dan pesan yang dikirim dan diterima oleh objek – objek tersebut. Objek biasanya bernama atau anonim untuk contoh kelas, tetapi juga dapat mewakili contoh dari hal – hal lain, seperti kolaborasi, komponen , dan node, *sequence diagram* digunakan untuk menggambarkan tampilan dinamis suatu sistem.

Tabel 2.3 Simbol-simbol pada *sequence diagram*

Nama	Simbol	Deskripsi
Aktor		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
Lifeline		Menyatakan kehidupan suatu objek, untuk menggambarkan kelas dan objek.
Objek		Menyatakan objek yang berinteraksi (pesan).
Waktu Aktif		Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif.
Pesan tipe <i>create</i>		Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek

		yang dibuat.
Pesan tipe <i>call</i>		Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, sesuai dengan kelas objek yang berinteraksi.
Pesan tipe <i>return</i>		Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.

2.12 Entity Relationship Diagram

Menurut Sutanta (2011:91) “*Entity Relationship Diagram* (ERD) merupakan suatu model data yang dikembangkan berdasarkan objek.” *Entity Relationship Diagram* (ERD) digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pengguna secara logis. *Entity Relationship Diagram* (ERD) didasarkan pada suatu persepsi bahwa *real world* terdiri atas obyek – obyek dasar tersebut. Penggunaan *Entity Relationship Diagram* (ERD) relative mudah dipahami, bahkan oleh para pengguna yang awam. Bagi perancang atau analis sistem, *Entity Relationship Diagram* (ERD) berguna untuk memodelkan sistem yang nantinya, basis data akan dikembangkan. Model ini juga membantu perancangan atau analis sistem pada saat melakukan analis dan perancangan basis data karena model ini dapat menunjukkan macam data yang dibutuhkan dan kerelasian antar data didalamnya.

2.12.1 Komponen *Entity Relationship Diagram* (ERD)

Komponen *Entity Relationship Diagram* (ERD) menurut Sutanta (2011:91) adalah sebagai berikut:

1. Entitas merupakan suatu obyek yang dapat dibedakan dari lainnya yang dapat diwujudkan dalam basis data. Obyek dasar dapat berupa orang, benda atau hal yang keterangannya perlu disimpan di dalam basis data. Untuk menggambarkan sebuah entitas digunakan aturan sebagai berikut:
 - 1) Entitas dinyatakan dengan simbol persegi panjang.
 - 2) Nama entitas dituliskan didalam simbol persegi panjang.
 - 3) Nama entitas berupa kata benda, tunggal.
 - 4) Nama entitas sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.
2. Atribut merupakan keterangan – keterangan yang terkait pada sebuah entitas yang perlu disimpan dalam basis data. Atribut berfungsi sebagai penjelas pada sebuah entitas. Untuk menggambarkan atribut digunakan aturan sebagai berikut:
 - 1) Atribut digambarkan dengan simbol elips.
 - 2) Nama atribut dituliskan didalam simbol elips.
 - 3) Nama atribut merupakan kata benda, tunggal.
 - 4) Nama atribut sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.
3. Relasi merupakan hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Aturan penggambaran relasi adalah sebagai berikut:
 - 1) Relasi dinyatakan dengan simbol belah ketupat.
 - 2) Nama relasi dituliskan didalam simbol belah ketupat.
 - 3) Nama relasi berupa kata kerja aktif.
 - 4) Nama relasi sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

Komponen-komponen yang terlibat dalam *Entity Relationship Diagram* adalah sebagai berikut:

1. Objek Data (*entity*)

Objek data adalah sekumpulan objek atau sesuatu yang dapat di bedakan atau didefinisikan secara unik. Objek data pada ERD disimbolkan dengan bentuk persegi panjang.

1. *Attribute*

Attribute adalah karakteristik dari entitas atau *relationship* yang menyediakan penjelasan detail tentang entitas.

2. *Relationship*

Relationship adalah hubungan yang terjadi antara satu *entity* atau lebih.

Ada tiga jenis *cardinality* dalam *relationship* yaitu:

1. *One to one* (satu ke satu).

Setiap entitas pada himpunan entitas A (pengojek) berhubungan dengan paling banyak satu entitas himpunan B (motor), dan begitu pula sebaliknya.

2. *One to many* (satu ke banyak).

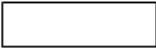
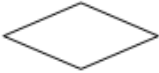

Setiap entitas pada himpunan entitas A (instruktur) berhubungan dengan banyak entitas pada himpunan entitas B (siswa), tetapi tidak sebaliknya. *Relationship* ini digambarkan sebagai berikut:

3. *Many to many* (banyak ke banyak).

Setiap entitas pada himpunan entitas A (siswa) berhubungan dengan banyak entitas pada himpunan entitas B (nilai), begitu pula sebaliknya. *Relationship* ini digambarkan sebagai berikut:

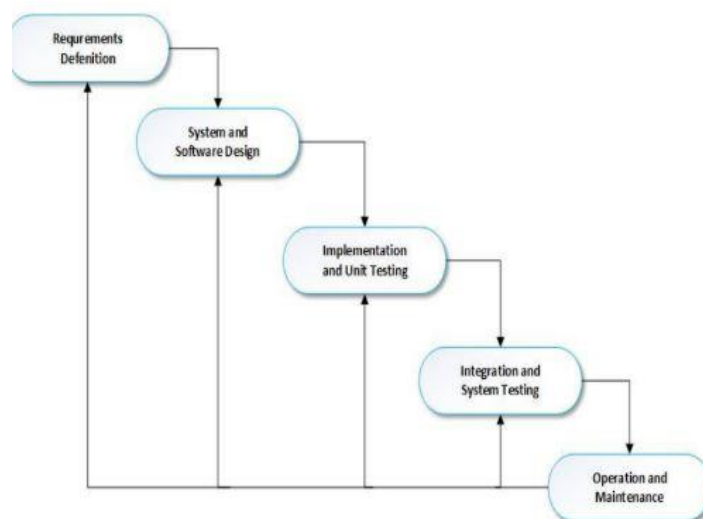
Adapun simbol-simbol yang digunakan pada *Entity Relationship Diagram* dapat dilihat pada table berikut ini.

Tabel 2.4 Simbol *Entity Relationship Diagram (ERD)*

Simbol	Keterangan
	Melambangkan entitas yang mewakili objek sebenarnya di dunia nyata.
	Menunjukkan kerelasian antara dua entitas atau lebih.
	Melambangkan atribut yang dimiliki oleh entitas.

2.13 Konsep Metode Perancangan

Menurut Yurindra (2017:43) *Waterfall* merupakan merupakan model yang membangun perangkat lunak berdasarkan daur hidup perangkat lunak (SDLC), yaitu model yang mempunyai struktur yang dimulai dari perencanaan, analisis, *design* dan implementasi, sehingga tahap pengembangan dalam *waterfall* mempunyai struktur model pengembangan yang disebut dengan *linier* dan *sequential*. Berikut merupakan beberapa step dari metode waterfall:

**Gambar 2.2** Metode Waterfall

1. Analisis (*Analysis*)

Seluruh kebutuhan *software* harus bisa di dapatkan dalam fase ini, termasuk didalamnya kegunaan *software* yang diharapkan pengguna dan batasan *software*. Informasi ini biasanya diperoleh melalui wawancara, survey, atau diskusi. Informasi tersebut dianalisis untuk mendapatkan dokumen kebutuhan pengguna untuk digunakan pada tahap selanjutnya.

2. Desain (*Designing*)

Tahap ini dilakukan sebelum *coding*. Tahap ini bertujuan untuk memberikan gambaran apa yang seharusnya dikerjakan dan bagaimana tampilannya. Tahap ini membantu dalam menspesifikasikan kebutuhan *hardware* dan sistem serta mendefinisikan arsitektur sistem secara keseluruhan.

3. Implementasi (*Implementation*)

Dalam tahap ini dilakukan pemrograman. Pembuatan *software* dipecah menjadi modul-modul kecil yang nantinya di gabungkan pada tahap berikutnya. Selain itu tahap ini juga dilakukan pemeriksaan terhadap modul yang dibuat, apakah sudah memenuhi fungsi yang diinginkan apa belum .

4. Pengujian Sistem (*Testing*)

Di tahap ini dilakukan penggabungan modul – modul yang sudah dibuat dan pengujian ini dilakukan apakah *software* yang dibuat telah sesuai dengan desainnya dan masih terdapat kesalahan atau tidak.

5. Pengembangan (*Maintenance*)

Ini merupakan tahap terakhir dalam metode *waterfall*. *Software* yang sudah jadi dijalankan dan dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak di temukan pada langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.