BAB II
LANDASAN TEORI

2.1 Sistem Informasi

Sesungguhnya yang dimaksud sistem informasi tidak harus melibatkan komputer. Sistem informasi yang menggunakan komputer biasa disebut sistem informasi berbasis komputer (Computer Based Information System atau CBIS). Dalam praktik, istilah sistem informasi lebih sering dipakai tanpa embel-embel berbasis komputer, walaupun dalam kenyataannya komputer merupakan bagian yang penting. Di buku ini, yang dimaksudkan dengan sistem informasi adalah sistem informasi berbasis komputer.

Ada beragam definisi sistem informasi, sebagaimana tercantum pada Tabel 2.1. Berdasarkan berbagai definisi tersebut, dapat disimpulkan bahwa sistem informasi mencangkup sejumlah komponen (manusia, komputer, teknologi informasi dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi) dan dimaksudkan untuk mencapai suatu sasaran atau tujuan (Kadir, 2014).

Tabel 2.1 Definisi Sistem Informasi

Sumber	Definisi
Alter (1992)	Sistem informasi adalah kombinasi antar prosedur kerja, informasi, orang dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.
Bodnar dan Hopwood (1993)	Sistem informasi adalah kumpulan perangkat keras dan peragkat lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna.
Gelinas, Oram dan Wiggins (1990)	Sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan dan mengelola data serta

	menyediakan informasi keluaran kepada para pemakai.
Hall (2001)	Sistem informasi adalah sebuah rangkaian prosedur formal, dimana data dikelompokkan, diproses menjadi informasi dan didistribusikan kepada para pemakai.
Turban, McLean dan Wetherbe (1999)	Sebuah sistem informasi mengumpulkan, memproses, mentimpan, menganalisis dan menyebarkan informasi untuk tujuam yang spesifik.
Wilkinson (1992)	Sistem informasi adalah kerangka kerja yang mengoordinasikan sumber daya (manusia dan komputer) untuk mengubah masukan (*input*) menjadi keluaran (informasi) guna mencapai sasaran-sasaran perusahaan.

2.2 Metode Pengembangan Sistem

Pada awal pengembangan perangkat lunak, para pembuat program (*programmer*) langsung melakukan pengodean perangkat lunak tanpa menggunakan prosedur atau tahapan pengembangan perangkat lunak. Dan ditemuilah kendala-kendala seiring dengan pengembangan skala sistem-sistem perangkat yang semakin besar (Rosa, 2011).

SDLC dimulai dari tahun 1960-an, untuk mengembangkan sistem skala usaha besar secara fungsional untuk para konglomerat pada zaman itu. Sistem-sistem yang di bangun mengelola informasi kegiatan dan rutinitas dari perusahaan-perusahaan yang berpotensi memiliki data yang besar dalam perkembangannya.

SDLC atau *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik).
Tahapan-tahapan yang ada pada SDLC secara global adalah sebagai berikut :

a. Inisiasi (*Initiation*)

Tahap ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak.

b. Pengembangan Konsep Sistem (*System Concept Development*)

   Mendefinisikan linkup konsep termasuk dokumen lingkup sistem, analisis manfaat biaya, manajemen rencana, dan pembelajaran kemudaha sistem.

c. Perencanaan (*Planning*)

   Mengembangkan rencana manajemen proyek dan dokumen perencaan lainnya.

d. Analisis Kebutuhan (*Requirements Analysis*)

   Menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan user dan membuat dokumen kebutuhan fungsional.

e. Desain (*Design*)

   Mentransformasikan kebutuhan detail menjadi kebutuhan yang sudah lengkap, dokumen desain sistem fokus pada bagaimana dapat memenuhi fumgsi-fungsi yang dibutuhkan.

f. Pengembangan (*Development*)

   Mengonversi desain ke sisitem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan, mempersiapkan berkas atau file pengujian, pengodean, pengompilasian, memperbaiki dan membersihkan program.

g. Integrasi dan Pengujian (*Integration and Test*)

   Mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional dengan diarahkan oleh staf penjamin kualitas dan user sehingga menghasilkan laporan analisis pengujian.

h. Implementasi (*Implementation*)

   Implementasi perangkat lunak pada lingkungan produksi (lingkungan pada user) dan menjalankan resolusi dari permasalahan yang teridentifikasi dari fase integrasi dan pengujian.

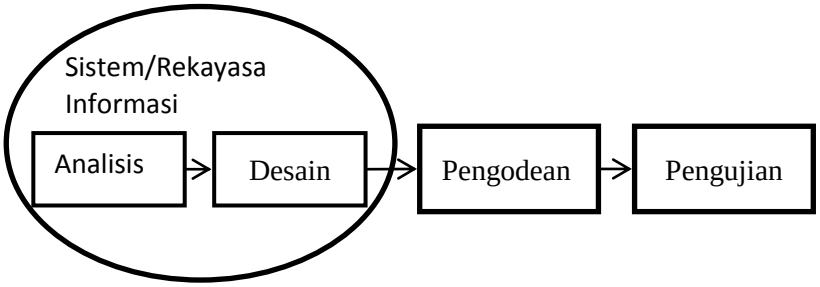
i. Operasi dan Pemeliharaan (*Operations and Maintenance*)

Mendeskripsikan pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan produksi, termasuk implementasi akhir dan masuk pada proses peninjauan.

j. Disposisi (*Disposition*)

Mendeskripsikan aktifitas akhir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktifitas *user*.

SDLC memiliki beberapa model dalam penerapan tahapan prosesnya, diantaranya adalah model *waterfall.* Model air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian dan tahap pendukung (Rosa, 2011) seperti pada Gambar 2.1.



Gambar 2.1 Ilustrasi Model *Waterfall*

a. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

b. Desain

Desain perangkat lunak adalah proses multilangkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan representasi desain agar dapat diimplementasikan menjadi program pada tahap

selanjutnya. Desainperangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

c. Pembuatan Kode Program

   Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

d. Pengujian

   Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian telah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

e. Pendukung

   Tidak menutup kemungkinan sebuah perangkat lunak mengalami perunahan ketika sudah dikirim ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perbuahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

Dari kenyataan yang terjadi sangat jarang model air terjun dapat dilakukan sesuai alurnya karena sebab berikut :

a. Perubahan spesifikasi perangkat lunak terjadi ditengah alur pengembangan.
b. Sangat sulit bagi pelanggan untuk mendefinisikan semua spesifikasi di awal alur. Pelanggan seringkali butuh contoh (*prototype*) untuk menjabarkan spesifikasi kebutuhan sistem lebih lanjut.
c. Pelanggan tidak mungkin bersabar mengakomodasi perubahan yang diperlukan di akhir alur pengembangan.

Dengan berbagai kelemahan yang dimiliki model air terjun tapi model ini telah menjadi dasar dari model-model yang lain dalam melakukan perbaikan model pengembangan perangkat lunak. Model air terjun sangat cocok digunakan

kebutuhan pelanggan sudah sangat dipahami dan kemungkinan terjadinya perubahan kebutuhan selama pengembangan pernagkat lunak kecil. Hal positif dari model air terjun adalah struktur pengembangan sistem jelas, dokumentasi dihasilkan disetiap tahap dan sebuah tahap dijalankan setelah tahap sebelumnya selesai dijalankan (tidak ada tumpang tindih pelaksanaan tahap).

2.3 Alat Bantu Perancangan Sistem

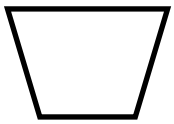
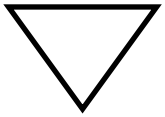
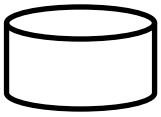
2.3.1 Flowchart

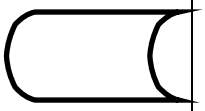
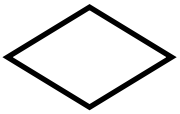
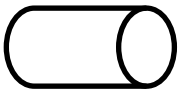
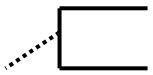
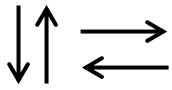
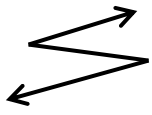
Flowchart adalah bagan alir yang menggambarkan suatu tahap penyelesaian masalah dengan menggunakan simbol-simbol yang standar efektif dan tepat. Ada dua macam *flowchart*, yaitu :

a. *Flowchart* Dokumen

*Flowchart*dokumen merupakan suatu bagan yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya.

Tabel 2.2 Simbol *Flowchart* Dokumen (Susanta, 2004)

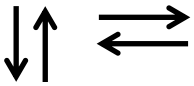
No.	Simbol	Keterangan
1.		Simbol dokumen menunjukkan dokumen input dan output baik proses manual, mekanik atau manual.
2.		Simbol manual menunjukkan kegiatan yang dilakukan secara manual.
3.		*File* bukan komputer yang diarsipkan
4.		Simbol proses menunjukkan kegiatan proses dari program komputer.
5.		Simbol *harddisk* menunjukkan input atau output menggunakan *harddisk.*

6.		Simbol *keyboard* menunjukkan *input* menggunakan *online keybord*.
7.		Simbol *diskette* menunjukkan *input* atau *output* menggunakan *disc*.
8.		Simbol penghubung menunjukkan penghubung ke halaman yang sama atau ke lain halaman.
9.		Simbol *display* menunjukkan *output* yang ditampilkan di monitor.
10.		Simbol terminal yang menunjukkan awal proses dan akhir proses.
11.		Simbol kondisi menunjukkan keadaan/kondisi.
12.		Simbol pita magnetik menunjukkan *input* atau *output* menggunakan pita magnetik.
13.		Simbol pita kertas berlubang menunjukkan *input* atau *output* menggunakan kertas berlubang.
14.		Simbol drum magnetik menunjukkan *input* atau *output* menggunakan drum magnetik.
15.		Simbol penjelasan menunjukkan penjelasan dari suatu proses.
16.		Simbol garis alir menunjukkan arus dari proses.
17.		Simbol penghubung komunikasi menunjukkan proses transmisi data melalui *channel* komunikasi.

b. *Flowchart* Program

Flowchart program yang menggambarkan secara detail atau secara rinci tentang proses atau urutan logika yang terjadi dalam sebuah program.

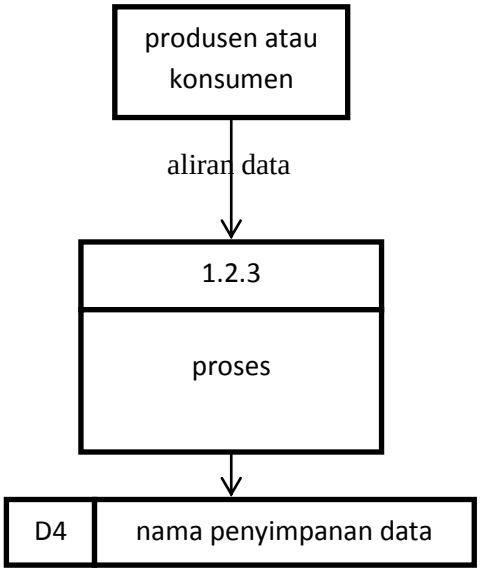
Tabel 2.3 Simbol *Flowchart* Program (Susanta, 2004)

No.	Simbol	Keterangan
1.		Simbol titik terminal digunakan untuk menunjukkan awal dan akhir dari suatu proses.
2.		Simbol *input/output* digunakan untuk mewakili data *input* dan *output.*
3.		Simbol keputusan digunakan untuk mewakili suatu proses pengolahan data.
4.		Simbol arah data digunakan sebagai arah arus logika program.
5.		Simbol proses digunakan untuk penyelesaian kondisi didalam program.
6.		Simbol proses terdefinisi digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan ditempat lain.
7.		Simbol persiapan pemberian nilai awal dari suatu variabel.
8.		Simbol penghubung menunjukkan penghubung ke halaman yang sama atau ke halaman berbeda.

2.3.2 Data Flow Diagram

Data Flow Diagram (DFD) awalnya dikembangkan oleh Chris Gane dan Trish Sarson pada tahun 1979 yang termasuk dalam *Structure System Analysis and Design Methodology* (SSADM) yang ditulis oleh Chris Gane dan Trish Sarson. Sistem yang dikembangkan ini berbasis pada dekomposisi fungsional dari

sebuah sistem. Berikut adalah contoh DFD yang dikembangkan oleh Chris Gane dan Trish Sarson pada Gambar 2.2.



Gambar 2.2 DFD dikembangkan oleh Crish Gane & Trish Sarson

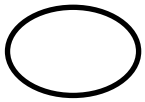
Edward Yourdon dan Tom DeMarco memperkenalkan metode yang lain pada tahun 1980-an dimana mengubah persegi dengan sudut lengkung (pada DFD Crish Gane dan Trish Sarson) dengan lingkaran untuk menotasikan. DFD Edward Yourdon dan Tom DeMarco populer digunakan sebagai model analisis sistem perangkat lunak untuk sistem perangkat lunak untuk sistem perangkat lunak yang akan diimplementasikan dengan pemograman terstruktur. Informasi yang ada di dalam perangkat lunak dimodifikasi dengan beberapa transformasi yang dibutuhkan. *Data Flow Diagram* (DFD)atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*).

DFD dapat digunakan untuk mempresentasikan sebuah sistem atau perangkat lunak pada beberapa level abstraksi. DFD dapat dibagi menjadi beberapa *level* yang lebih detail untuk mempresentasikan aliran informasi atau fungsi yang lebih detail. DFD lebih sesuai digunakan untuk memodelkan fungsi-fungsi perangkat lunak yang akan diimplementasikan menggunakan pemograman

terstruktur, karena pemograman terstruktur membagi-bagi bagiannya dengan fungsi-fungsi dan prosedur-prosedur (Rosa, 2011).

DFD tidak sesuai untuk memodelkan sistem perangkat lunak yang akan dibangun menggunakan pemograman berorientasi objek. Paradigma pemograman terstruktur dan pemograman berorientasi objek merupakan hal yang berbeda. Simbol-simbol pada DFD Edward Yourdon dan Tom DeMarco adalah seperti pada Tabel 2.4.

Tabel 2.4 Simbol *Data Flow Diagram* (DFD)

No.	Simbol	Keterangan
1.		Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program. Nama yang diberikan pada sebuah proses biasanya berupa kata kerja.
2.		*File* atau basis data atau penyimpanan (*storage*); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada basis data (*Entity Relationship Diagram, Conseptual Data Model, Physical Data Model*). Nama yang diberikan pada sebuah penyimpanan biasanya kata benda.
3.		Entitas luar (*external entity*)atau masukan (*input*) atau keluaran (*output*) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan. Nama yang digunakan pada masukan (*input*) atau keluaran (*output*) biasanya berupa kata benda.

4.		Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses atau dari proses ke masukan (*input*) atau keluaran (*output*). Nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya "data siswa" atau tanpa kata data seperti "siswa".

Tahapan-tahapan perancangan dengan menggunakan DFD adalah sebagai berikut:

a. Membuat DFD Level 0 (Diagram Konteks)

   DFD level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DFD level 0 digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.

b. Membuat DFD Level 1

   DFD level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan.

c. Membuat DFD Level 2

   DFD level 2 merupakan turunan dari DFD level 1. DFD level 2 lebih detail tergantung pada tingkat kedetailan modul tersebut.

d. Membuat DFD Level 3 dan seterusnya.

   DFD level 3, 4, 5 dan seterusnya merupakan breakdown dari modul pada DFD level di atasnya. Breakdown pada level 3, 4, 5 dan seterusnya aturannya sama persisdengan DFD level 1 atau level 2.

2.3.3 Basis Data

Basis data (*database*) adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi. Basis data di maksudkan untuk mengatasi problem pada sistem yang memakai pendekatan berbasis berkas.

Untuk mengelola basis data diperlukan perangkat lunak yang disebut *Database Management System* (DBMS). DBMS adalah perangkat lunak sistem yang memungkinkan para pemakai membuat, memelihara, mengontrol, dan

mengakses basis data dengan cara yang praktis dan efisien. DBMS dapat digunakan untuk mengakomodasikan berbagai macam pemakai yang memiliki kebutuhan akses yang berbeda-beda.

Gambar di atas menunjukan bahwa suatu aplikasi dapat berkomunikasi melalui DBMS untuk mengakses basis data dan kemudian membetuk laporan-laporan Selain itu, pemakai juga bisa berinyeraksi secara langsung dengan DBMS untuk mengakses basis data, baik untuk keperluan memintaataupun untuk melakukan perubahan data. Interaksi secara langsung denganbasis data memungkinkan pemakai untuk memperoleh informasi-informasi yang dibutuhkan sewaktu-waktu dan bersifat sementara, tanpa memerlukanbantuan pemrogram.

Umumnya DBMS menyediakan fitur-fitur sebagai berikut :

a. Independensi data program

   Karena basia data ditangani oleh DBMS, program dapat ditulis sehingga tidak tergantung pada stuktur data dalam basis data. Dengan perkataan lain, program tidak akan terpengaruh sekiranya bentuk fisik data diubah.

b. Keamanan

   Keamanan dimaksudkan untuk mencegah pengaksesan data oleh orang yang tidak berwewenang.

c. Integritas

   Hal ini ditujukan untuk menjaga agar data selalu dalam keadaan yang valid dan konsisten.

d. Konkurensi

   Konkurensi memungkinkandata dapat diakses oleh banyak pemakai tanpa menimbulkan masalah.

e. Pemulihan (*recovery*)

   DBMS menyediakan mekanisme untuk mengembalikan basis data ke keadaansemula yang konsisten sekiranya terjadi gangguan perangkat keras atau kegagalan perangkat lunak.

f. Katalog sistem

   Katalog sistem adalah deskripsi tentang data yang terkandung dalam basis data yang dapat diakses oleh pemakai.

g. Perangkat produktivitas

Untuk menyediakan kemudahan bagi pemakai dan meningkatkan produktivitas, DBMS menyediakan sejumlah perangkakat produktivitas seperti pembangkit query dan pembangkit laporan.

Komponen-komponen yang menyusun lingkungan DBMS terdiri atas:

a. Perankat keras

   Perangkat keras digunakan untuk menjalankan DBMS beserta aplikasi-aplikasinya. Perangkat keras berupa komputer dan periferal pendukungnya. Komputer dapat berupa PC, minikomputer, mainframe, dan lain-lain.

b. Perangkat lunak

   Komponen perangkat lunak mencakup DBMS itu sendiri, program aplikasi, serta perangkat lunak pendukung untuk komputer dan jaringan. Program aplikasi dapat dibangun dengan menggunakan bahasa pemrograman seperti C++, Pascal, Delphi, atau Visual BASIC.

c. Data

   Bagi sisi pemakai, komponen terpenting dalam DBMS adalah data karena dari data inilah pemakai dapat memperoleh informasi yang sesuai dengan kebutuhan masing-masing.

d. Prosedur

   Prosedur adalah petujuk tertulis yang berisi cara merancang hingga menggunakan basis data. Beberapa hal yang dimasukkan dalam prosedur:

   1. Cara masuk ke DBMS (login).
   2. Cara memekai fasilitas-fasilitas tertentu dalam DBMS maupun cara menggunakan aplikasi.
   3. Cara mengaktifkan dan menhentikan DBMS.
   4. Cara membuat cadangan basis data dan cara mengembalikan cadangan ke DBMS.

e. Orang

   Komponen orang dapat dibagi menjadi tiga kelompok, yaitu :

   1. Pemakai akhir (end-user).
   2. Pemogram aplikasi.
   3. Administrator basis data.

Tedapat beberapa elemen basis data, yaitu :

a. Database

Database atau basis data adalah kumpilan tabel yang mempunyai kaitan antara suatu tabel dengan tabel lainya sehingga membentuk suatu bangunan data.

b. Tabel

Tabel adalah kumpulan *record-record* yang mempunyai panjang elemen yang sama dan atribut yang sama namun berbeda data *value*nya.

c. Entitas

Entitas adalah sekumpulan objek yang terdefinisikan yang mempunyai karakteristik sama dan bisa dibedakan satu dengan lainnya. Objek dapat berupa barang, orang, tempat atau suatu kejadian.

d. Atribut

Atribut adalah deskripsi data yang bisa mengidentifikasi entitas yang membedakan entitas tersebut dengan entitas yang lain. Seluruh atribut harus cukup untuk menyatakan identitas objek atau dengan kata lain, kumpulan atribut dari setiap entitas dapat mengidentifikasi keunikan suatu individu.

e. *Data Value* (Nilai Data)

Data value adalah data aktual atau informasi yang disimpan pada tiap data, elemen atau atribut. Atribut nama pegawai menunjukan tempat dimana informasi nama karyawan disimpan, nilai datanya misalnya adalah Anjang, Arif, Suryo dan lain-lain yang merupakan isi data nama pegawai tersebut.

f. File

File adalah kumpulan *record* sejenis yang mempunyai panjang elemen yang sama, atribut yang sama namun berbeda nilai datanya.

g. Record/Tuple

Kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu entitas secara lengkap. Satu *record* mewakili satu data atau informasi.

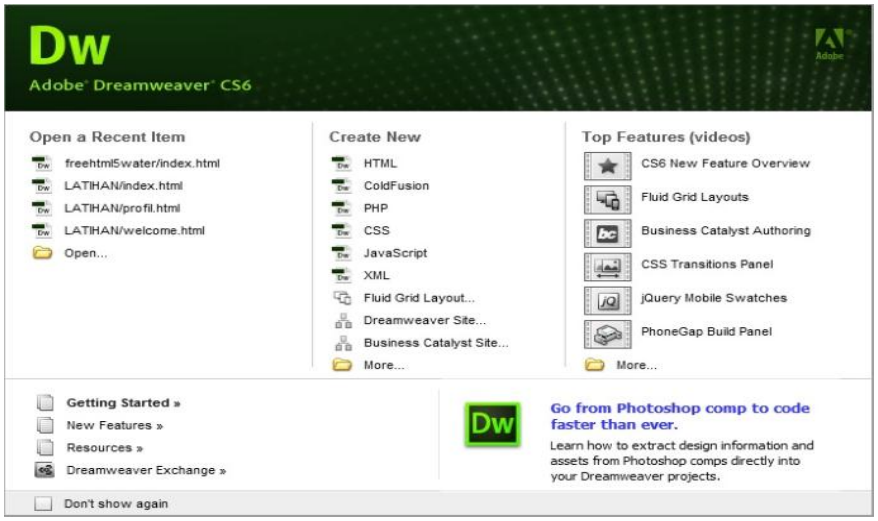
2.4 Kebutuhan Perangkat Lunak

2.4.1 PHP (*Hypertext Prepocessor*)

PHP yaitu bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada *server* (*server side HTML embedded scripting).* PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan di tampilkan dibuat saat halaman itu diminta oleh *client.* Mekanisme ini menyebabkan informasi yang di terima client selalu yang terbaru/*up to date*. Semua *script* PHP dieksekusi pada *server* dimana *script* tersebut dijalankan (Rudyanto, 2011).

2.4.2 Adobe Dreamweaver CS

Adobe Dreamweaver CS merupakan salah satu program aplikasi yang digunakan untuk membangun sebuah *website*, baik secara grafis maupun dengan menulis kode sumber secara langsung. *Adobe Dreamweaver CS* memudahkan pengembangan *website* untuk mengelola halaman-halaman *website* dan aset-asetnya, baik gambar (*image*), animasi *flash*, video, suara dan lain sebagainya. Selain itu, *Adobe Dreamweaver CS* juga menyediakan fasilitas untuk melakukan pemrograman scripting, baik ASP (*Active Server Page*), JSP (*Java Server Page*), PHP (*Hypertext Preprocessor*), JavaScript (*js*), *Cold Fusion*, CSS (*Cascading Style Sheet*), XML (*Extensible Markup Language*) dan lainnya (Wahana Komputer, 2010). Tampilan *start page welcome to Adobe Dereamweaver CS* dapat dilihat pada Gambar 2.3.



Gambar 2.3 Tampilan *Start Page Welcome To Adobe Dereamweaver CS*

2.4.3 MySQL

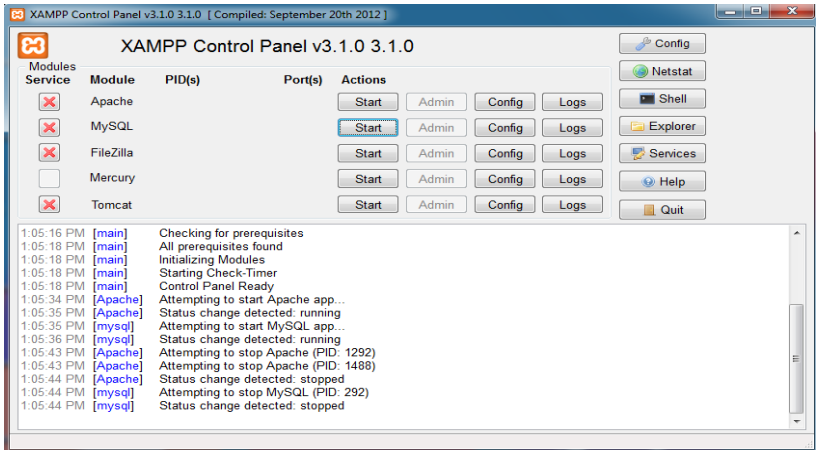
MySQL adalah salah satu jenis *database server* yang terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan *database* sebagai sumber dan pengelolaan data. Kepopuleran *MySQL* antara lain karena *MySQL* menggunakan *SQL* sebagai bahasa dasarnya untuk mengakses *database*nya sehingga mudah untuk digunakan.

MySQL termasuk RDBMS (*Relational Database Management System*). Pada *MySQL*, sebuah d*atabase* mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah kolom dan baris, dimana setiap kolom berisi sekumpulan data, dan baris merupakan sekumpulan data yang saling berkaitan dan membentuk informasi. Kolom biasanya disebut sebagai field dan informasi yang tersimpan dalam setiap baris disebut *record* (Rudyanto, 2011). Tampilan MySQL dapat dilihat pada Gambar 2.4.

Gambar 2.4 Tampilan *MySQL*

2.4.4 Xampp

Xampp adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolahan data *MySQL* di komputer lokal. *Xampp* berperan sebagai *server* web pada kumputer. *Xampp* juga dapat disebut sebuah *Cpanel server virtual*, yang dapat membantu melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses dengan internet (Yogi, 2008). Tampilan *start page welcome to xampp* dapat dilihat pada Gambar 2.5.



Gambar 2.5 Tampilan *Start Page Welcome To Xampp*

2.4.5 CodeIgniter

CodeIgniter adalah aplikasi *open source* yang berupa *framework* dengan model MVC (*Model*, *View*, *Controller*) untuk membangun *website* dinamis dengan menggunakan PHP. *CodeIgniter* memudahkan *developer* untuk membuat aplikasi web dengan cepat dan mudah dibandingkan dengan membuatnya dari awal. *CodeIgniter* dirilis pertama kali pada 28 Februari 2006.

Framework secara sederhana dapat diartikan kumpulan dari fungsi-fungsi atau prosedur-prosedur dan *class-class* untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang pemrograman, tanpa harus membuat fungsi atau *class* dari awal. Ada beberapa alasan mengapa menggunakan *framework*, yaitu :

a. Mempercepat dan mempermudah pembangunan sebuah aplikasi web.
b. Relatif memudahkan dalam proses maintenance karena sudah ada pola tertentu dalam sebuah *framework* (dengan syarat programmer mengikuti pola standar yang ada).
c. Umumnya *framework* menyediakan fasilitas-fasilitas yang umum dipakai sehingga kita tidak perlu membangun dari awal, misalnya validasi, ORM, *pagination*, *multiple database*, *scaffolding*, pengaturan *session*, *error handling* dan lainnya.
d. Lebih bebas dalam pengembangan jika dibandingkan CMS.

Model *View Controller* merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, berawal pada bahasa pemrograman Small Talk, MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, user interface dan bagian yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang membangun suatu MVC pattern dalam suatu aplikasi yaitu :

a. *View*, merupakan bagian yang menangani *presentation logic*. Pada suatu aplikasi web bagian ini biasanya berupa *file template* HTML yang diatur oleh *controller*. *View* berfungsi untuk menerima dan merepresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian model.
b. Model, biasanya berhubungan langsung dengan *database* untuk memanipulasi data (*insert, update, delete, search*), menangani validasi dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.
c. *Controller*, merupakan bagian yang mengatur hubungan antara bagian model dan bagian view. *Controller* berfungsi untuk menerima request dan data dari *user* kemudian menentukan apa yang akan diproses oleh aplikasi.

Dengan menggunakan prinsip MVC suatu aplikasi dapat dikembangkan sesuai dengan kemampuan developernya, yaitu *programmer* yang menangani bagian model dan *controller*, sedangkan *designer* yang menangani bagian *view*,

sehingga penggunaan arsitektur MVC dapat meningkatkan *maintanability* dan organisasi kode. Walaupun demikian dibutuhkan komunikasi yang baik antara *programmer* dan *design*er dalam menangani variabel-variabel yang akan ditampilkan. Ada beberapa kelebihan *CodeIgniter* (CI) dibandingkan dengan *framework* PHP lain, yaitu :

a. Performa sangat cepat

   Salah satu alasan tidak menggunakan *framework* adalah karena eksekusinya yang lebih lambat daripada PHP *from the scracth*, tapi *CodeIgniter* sangat cepat bahkan mungkin bisa dibilang *CodeIgniter* merupakan *framework* yang paling cepat dibanding *framework* yang lain.

b. Konfigurasi yang sangat minim (*nearly zero configuration*)

   Tentu saja untuk menyesuaikan dengan *database* dan keleluasaan *routing* tetap diizinkan melakukan konfigurasi dengan mengubah beberapa *file* konfigurasi seperti *database.php* atau *autoload.php*, namun untuk menggunakan *CodeIgniter* dengan *setting standard*, anda hanya perlu merubah sedikit saja *file* pada *folder config*.

c. Banyak komunitas

   Dengan banyaknya komunitas CI ini, memudahkan kita untuk berinteraksi dengan yang lain, baik itu bertanya atau teknologi terbaru.

d. Dokumentasi yang sangat lengkap

   Setiap paket instalasi codeigniter sudah disertai *user guide* yang sangat bagus dan lengkap untuk dijadikan permulaan, bahasanya pun mudah dipahami (Dewi, 2011).

2.4.6 Bootsrap

Bootrstrap adalah sebuah alat yang membantu pengembang *website* dalam mendesain sebuahtampilan *website*. *Bootstrap* juga bisa disebut sebagai *Framework CSS* pembuatan *website* gratis dari *twitter*. *Bootstrap* menggunakan *Lisensi Apache 2.0*, sebuah lisensi terbuka yang membebaskan kita menggunakannya dengan mudah. Kelebihan yang dimiliki *bootstrap* adalan sebagai berikut :

a. Cepat, memiliki banyak *library* yang menyediakan potongan kode yang siap digunakan.
b. Fleksibel, dapat menyesuaikan sesuai kebutuhan *website*.
c. *The Grid*, menggunakan *grid* menjadikan pekerjaan menjadi lebih mudah. Penataan sudah diselesaikan dalam mode otomatis. *Bootstrap* dapat berjalan sempurna pada *browser-browser* seperti *Opera, Firefox, Safari, Chrome,* dan *Internet Explorer*. *Bootstrap* diciptakan pada Tahun 2011 oleh Mark Otto dan Jacob Thomton yang menjadi *programmer* di twitter.Pertama kali diluncurkan pada Agustus 2011.*Bootstrap* hanya sebuah proyek berbasis CSS, namun telat berevolusi menjadi sebuah *framework* yang berisi*Javascript Plugin, Icon, Forms* dan *Button* (Ariskevin, 2014).

2.5 Penelitian Terkait

Penelitian yang terkait dengan penelitian yang sudah dilakukan sebelumnya adalah sebagai berikut :

a. Menurut Victor dalam penelitiannya menyimpulkan bahwa konsumen dapat memesan produk pada CV. Richness dimana saja yang terhubung dengan jaringan internet, konsumen hanya perlu mendaftar, melakukan *login*, memilih produk dan melakukan pemesanan.
b. Menurut William dalam penelitiannya menyimpulkan bahwaAplikasi ini dapat diakses menggunakan *handphone* untuk memudahkan mekanisme pemesanan pada *foodcourt* sehingga lebih cepat dan efisien dan dapat digunakan untuk membantu keperluan manajemen dan administrasi *foodcourt*.
c. Menurut Gat dalam penelitiannya menyimpulkan bahwa sistem pemesanan makan dan minuman berbasis *mobile* telah berhasil dibangun dengan mengacu kepada kebutuhan bagi pihak restoran. Sistem ini memerlukan 3 (tiga) komputer *desktop*, minimal 1 (satu) *tablet* tergantung dari banyaknya pelayan dan 1 (satu) perangkat *wireless* yang menghubungkan perangkat *tablet* dengan *database*. Pengaplikasian sistem ini sangat membantu dalam mempermudah pengelolaan pesanan konsumen dimana diantara pelayan, bagian dapur dan kasir saling terkoneksi sehingga pada saat pelayan menginputkan data pesanan menggunakan tablet, maka data pesanan tersebut akan langsung diterima oleh bagian dapur.

d. Menurut Antonius dalam penelitiannya menyimpulkan bahwa dengan adanya sistem *delivery order* yang berbasis Android dapat mempermudah jalannya perusahaan untuk mengembangkan usahanya.