

BAB II **KAJIAN TEORI**

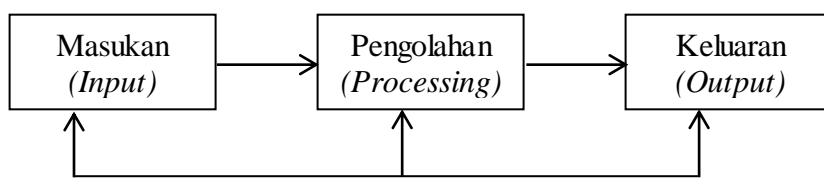
A. Deskripsi Teori

Beberapa teori yang digunakan sebagai dasar dari penelitian ini adalah (1) Sistem Informasi, (2) Model Pengembangan Sistem, (3) Basis Data Relasional, (4) Perangkat Lunak Pengembangan, (5) Jaminan Kualitas Perangkat Lunak, dan (6) Layanan Internet Mahasiswa UNY. Penjelasan lebih lanjut mengenai teori-teori tersebut adalah sebagai berikut:

1. Sistem Informasi

Sistem adalah kumpulan elemen, komponen, atau subsistem yang saling berintegrasi dan berinteraksi untuk mencapai tujuan tertentu (Supriyanto,2005:238).

Menurut Abdul Kadir (2014:61-62), sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan elemen-elemen yang membentuk sebuah sistem yaitu tujuan, masukan, keluaran, proses, mekanisme, pengendalian, dan umpan balik. Selain itu, sistem juga berinteraksi dengan lingkungan dan memiliki batas. Sedangkan menurut Scott (1996), sistem terdiri dari unsur-unsur seperti masukan (*input*), pengolahan (*processing*), serta keluaran (*output*). Model sistem tersebut dapat dilihat pada Gambar 1.



Gambar 1. Model Sistem

Informasi adalah data yang telah diolah menjadi bentuk yang berarti bagi penerimanya dan bermanfaat dalam mengambil keputusan saat ini atau mendatang (Supriyanto,2005:243).

Menurut Henry C. Lucas Jr. (1993:3) sistem informasi adalah sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan dan/atau untuk mengendalikan organisasi.

Menurut Abdul Kadir (2014:71-72) sistem informasi mengandung komponen-komponen sebagai berikut:

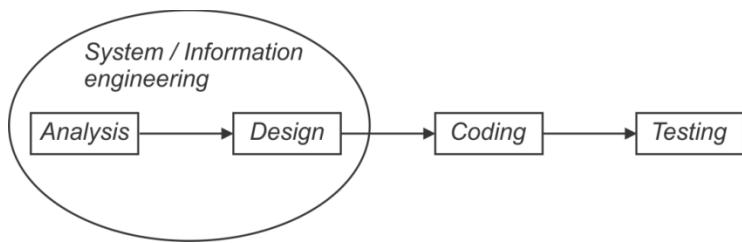
- a. Perangkat keras (*hardware*), yang mencakup peranti-peranti fisik seperti komputer dan printer.
- b. Perangkat lunak (*software*) atau program, yaitu sekumpulan instruksi yang memungkinkan perangkat keras memproses data.
- c. Prosedur, yaitu sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.
- d. Orang, yakni semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan dan penggunaan keluaran sistem informasi.
- e. Basis data (*database*), yaitu kumpulan tabel, hubungan, dan lain-lain yang berkaitan dengan penyimpanan data.
- f. Jaringan komputer dan komunikasi data, yaitu sistem penghubung yang memungkinkan sumber (*resources*) dipakai secara bersama atau diakses oleh sejumlah pemakai.

2. Model Pengembangan Sistem

a. Waterfall Model

Model pengembangan sistem yang akan dipakai adalah metode analisis sistem terstruktur *Waterfall Model*.

Menurut Pressman (2001:28-29) *Waterfall Model* atau disebut juga *classic life cycle* adalah suatu model yang digunakan dalam pengembangan perangkat lunak dengan pendekatan sistematis dan sekuensial. Tahapan dalam *Waterfall Model* adalah *analysis*, *design*, *coding*, *testing*, dan *support*. Tahapan dalam *Waterfall Model* tersebut digambarkan dalam Gambar 2.



Gambar 2. *Waterfall Model*

Berikut penjelasan setiap tahapan dalam *Waterfall Model*:

- 1) *Software requirements analysis*, adalah tahap menganalisa hal-hal yang diperlukan dalam pelaksanaan perancangan sistem.
- 2) *Design*, adalah tahap penerjemah atau tahap perancangan dari keperluan-keperluan yang dianalisis dalam bentuk yang lebih mudah dimengerti oleh pemakai.
- 3) *Code generation*, adalah tahap implementasi dari hasil sistem yang telah dirancang dalam bahasa pemograman yang telah ditentukan dan digunakan dalam pembuatan sistem.

4) *Testing*, adalah tahap pengujian terhadap program yang telah dibuat. Pengujian dilakukan agar fungsi-fungsi dalam sistem bebas dari *error*, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

b. Data Flow Diagram

Data Flow Diagram (DFD) adalah diagram yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, di mana data disimpan, proses apa yang dihasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut (Kristanto, 2008:61).

Al-Bahra bin Ladjamuddin (2005: 67-75), elemen dasar dari *DFD* adalah sebagai berikut:

1) Kesatuan Luar (*External Entity*)

Kesatuan luar adalah sesuatu yang berada di luar sistem tetapi memberikan data ke dalam sistem atau sebaliknya. Kesatuan luar tidak termasuk bagian dari sistem. Pedoman pemberian nama kesatuan luar adalah sebagai berikut:

a) Nama kesatuan luar berupa kata benda.

b) Kesatuan luar tidak boleh memiliki nama yang sama kecuali memang objeknya sama.

2) Proses (*Process*)

Proses merupakan kegiatan atau kerja yang dilakukan oleh sistem. Proses berfungsi mentransformasikan satu atau beberapa data masukan menjadi satu

atau beberapa data keluaran sesuai dengan spesifikasi yang diinginkan. Pedoman pemberian nama proses adalah sebagai berikut:

- a) Nama proses terdiri dari kata kerja dan kata benda yang mencerminkan fungsi proses tersebut.
- b) Kata proses tidak boleh digunakan sebagai bagian dari nama suatu proses.
- c) Tidak boleh ada beberapa proses yang memiliki nama yang sama.
- d) Proses harus diberi nomor.

3) Simpanan Data (*Data Store*)

Simpanan data merupakan tempat penyimpanan data yang ada dalam sistem. Pedoman pemberian nama simpanan data adalah sebagai berikut:

- a) Nama harus mencerminkan simpanan data tersebut.
- b) Bila namanya lebih dari satu kata, maka harus diberi tanda sambung.

4) Arus Data (*Data Flow*)

Arus data merupakan tempat mengalirnya informasi dan digambarkan dengan garis yang menghubungkan komponen dari sistem. Arus data ditunjukkan dengan arah panah dan garis diberi nama atas arus data yang mengalir. Pedoman pemberian nama aliran data adalah sebagai berikut:

- a) Nama aliran data yang terdiri dari beberapa kata dihubungkan dengan garis sambung.
- b) Sedapat mungkin nama aliran data ditulis lengkap.
- c) Tidak boleh ada aliran data dari kesatuan luar ke simpanan data atau sebaliknya, hubungan kesatuan luar dengan simpanan data harus melalui proses.

5) Simbol *DFD*

Simbol *DFD* disajikan pada Tabel 1:

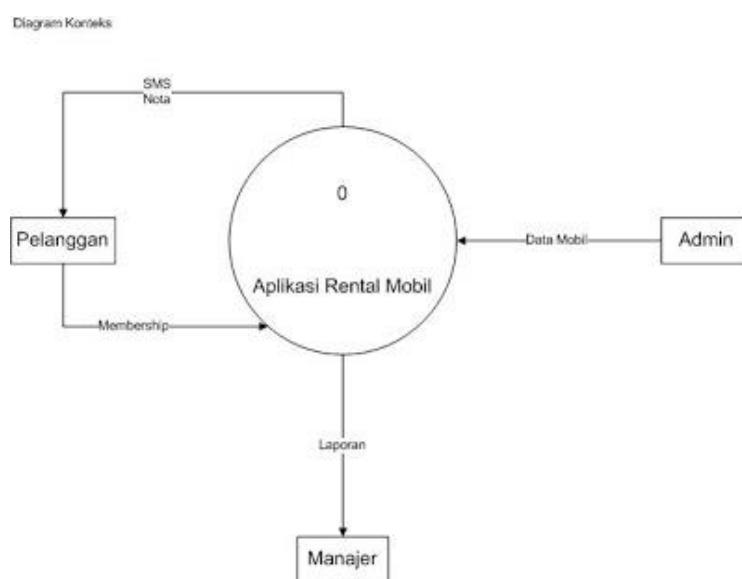
Tabel 1. Simbol Data Flow Diagram

Simbol	Keterangan
	Kesatuan Luar
	Proses
→	Arus Data
	Penyimpanan Data

Menurut Agus Winarno (2007: 2) *DFD* dibagi menjadi 2 jenis, yaitu:

- Diagram Konteks (*Context Diagram*)

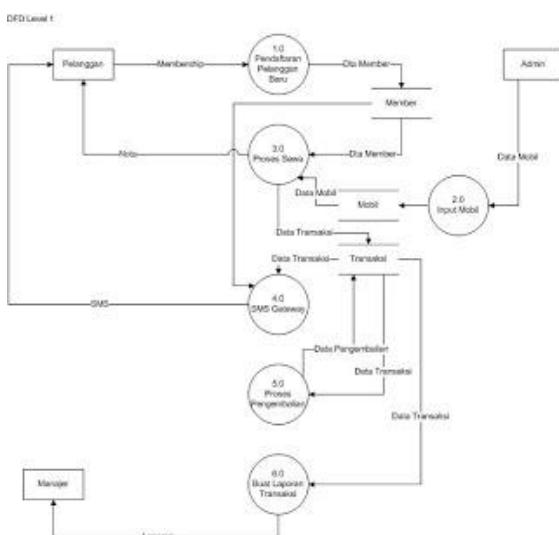
Diagram konteks merupakan *DFD* Level 0, yaitu diagram yang paling sederhana dari sebuah sistem informasi yang menggambarkan aliran data dari kesatuan luar ke dalam sistem dan sebaliknya. Contoh diagram konteks ditampilkan pada Gambar 3.



Gambar 3. Contoh Diagram Konteks

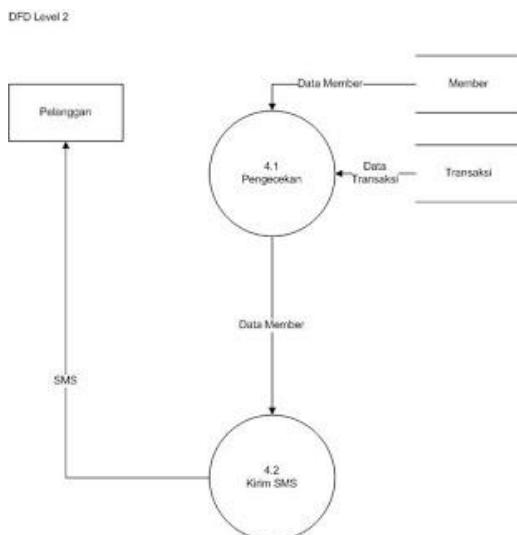
b) *DFD Level n*

DFD Level n merupakan diagram yang digunakan untuk menggambarkan diagram hasil pengembangan dari diagram konteks ke dalam komponen yang lebih detail. Nilai n merupakan banyaknya angka atau digit yang digunakan untuk penomoran proses yang ada. Contoh *DFD* Level 1 disajikan pada Gambar 4.



Gambar 4. Contoh DFD Level 1

Contoh *DFD* Level 2 disajikan pada Gambar 5.



Gambar 5. Contoh DFD Level 2

c. *Flowmap*

Menurut Winarno (2004:102), *flowmap* adalah gambar yang menggunakan lambang-lambang baku untuk menggambarkan sistem atau proses. Pada waktu akan menggambar suatu bagan alir, analisis sistem dapat mengikuti pedoman sebagai berikut :

- 1) Bagan alir sebaiknya digambar dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
- 2) Kegiatan di bagan alir harus ditunjukan dengan jelas darimana kegiatan akan dimulai dan dimana akan berakhir.
- 3) Masing-masing kegiatan di dalam bagan alir sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan.
- 4) Masing-masing kegiatan di dalam bagan alir harus didalam urutan yang semestinya.
- 5) Kegiatan yang terpotong dan akan disambung di tempat lain harus ditunjukkan dengan jelas.
- 6) Gunakanlah simbol-simbol bagan alir yang standar.

Simbol-simbol *flowmap* ditunjukkan pada Tabel 2 berikut:

Tabel 2. Simbol *Flowmap*

Simbol	Keterangan	Simbol	Keterangan
	Mulai/selesai		Kondisi
	Proses		Tempat Penyimpanan
	Dokumen <i>input/ output</i>		Penghubung

3. Basis Data Relasional

a. Basis Data

Menurut Yakub (2008:1-3) basis dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data merupakan representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, konsep, keadaan, dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya. Sehingga, basis data (*database*) merupakan kumpulan data yang saling berhubungan (punya relasi).

Secara konsep, basis data adalah kumpulan dari data-data yang membentuk suatu berkas yang saling berhubungan dengan tata cara yang tertentu untuk membentuk data baru atau informasi, atau merupakan kumpulan dari data yang saling berhubungan antara satu dengan yang lainnya yang diorganisasikan berdasarkan skema atau struktur tertentu (Supriyanto, 2005:190).

Prinsip utama basis data adalah pengaturan data dengan tujuan utama fleksibilitas dan kecepatan dalam pengambilan data kembali. Tujuan basis data diantaranya sebagai efisiensi yang meliputi *speed*, *space*, *accuracy*, menangani data dalam jumlah besar, kebersamaan pemakaian (*shareability*), dan meniadakan duplikasi dan inkonsistensi data.

b. Database Management System

Database Management System (DBMS) merupakan kumpulan program aplikasi yang digunakan untuk membuat dan mengelola basis data. DBMS

merupakan perangkat lunak yang menentukan bagaimana data tersebut diorganisasi, disimpan, diubah, dan diambil kembali (Yakub, 2008:14).

Utami (2008:2) mengemukakan bahwa *DBMS* merupakan perangkat lunak yang dirancang untuk dapat melakukan utilisasi dan mengelola koleksi data dalam jumlah yang besar. *DBMS* juga dirancang untuk dapat melakukan manipulasi data secara lebih mudah. Beberapa contoh *DBMS* adalah *PostgreSQL*, *MySQL*, *DB2*, *Oracle*, *SQL Server*, dan lain-lain.

Supriyanto (2005:196) memaparkan, manfaat penggunaan *DBMS* adalah:

- 1) Untuk mengorganisasikan dan mengelola data dalam jumlah besar.
- 2) Untuk membantu dalam melindungi data dari kerusakan yang disebabkan penggunaan atau pengaksesan yang tidak sah.
- 3) Memudahkan dalam pengambilan data kembali (*data retrieval*)
- 4) Untuk memudahkan dalam penggunaan atau pengaksesan data secara bersamaan dalam suatu jaringan.

Keunggulan *DBMS* adalah:

- 1) Kepraktisan, yaitu penggunaan media penyimpanan yang berukuran kecil namun padat informasi.
- 2) Kecepatan, yaitu mesin dapat mengambil atau mengubah data jauh lebih cepat daripada manusia.
- 3) Mengurangi Kejemuhan, yaitu menghindari pekerjaan yang berulang-ulang dan monoton yang bisa membosankan.
- 4) Kekinian (*up to date*), yaitu informasi yang tersedia pada *DBMS* akan bersifat mutakhir dan akurat setiap saat.

c. Model Basis Data Relasional

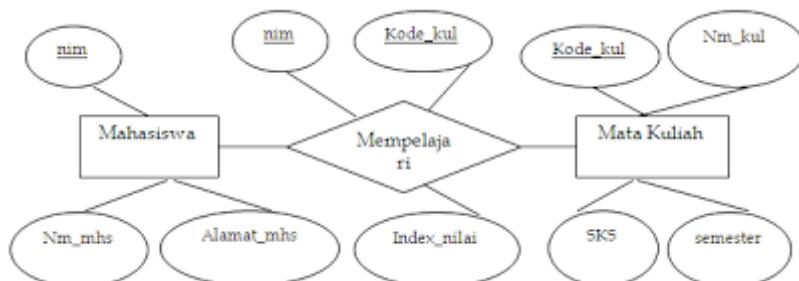
Model basis data relasional dapat dikatakan sebagai kumpulan satu atau lebih relasi dimana setiap relasi merupakan korelasi dari data disajikan dalam bentuk tabel yang terdiri dari baris dan kolom. Relasi merupakan konstruksi diskripsi data yang utama dalam model relasional (Utami, 2008:12).

Untuk merancang model dasar dari struktur data serta hubungan dari setiap data tersebut digunakan *Entity Relationship Diagram (ERD)*. *ERD* adalah sekumpulan cara atau peralatan untuk mendeskripsikan data-data atau objek-objek yang dibuat berdasarkan dan berasal dari dunia nyata yang disebut entitas (*entity*) serta hubungan (*relationship*) antar entitas-entitas tersebut dengan menggunakan beberapa notasi. Komponen-komponen pembentuk *ERD* disajikan pada Tabel 3.

Tabel 3. Komponen Pembentuk Entity Relational Diagram

Simbol	Komponen	Keterangan
[rectangle]	Entitas	Individu yang mewakili suatu objek dan dapat dibedakan dengan objek yang lain.
(oval)	Atribut	Properti yang dimiliki oleh suatu entitas, dimana dapat mendeskripsikan karakteristik dari entitas tersebut.
(diamond)	Relasi	Menunjukkan hubungan diantara sejumlah entitas yang berbeda.
—	Garis penghubung	Menunjukkan adanya hubungan atau relasi

Contoh *Entity Relationship Diagram (ERD)* ditampilkan pada Gambar 6.



Gambar 6. Contoh Entity Relational Diagram

Hal yang perlu diperhatikan dalam model basis data relasional adalah normalisasi. Normalisasi merupakan teknik analisis data yang mengorganisasikan atribut-atribut data dengan cara mengelompokkan sehingga terbentuk entitas yang non-*redundant*, stabil, dan fleksibel.

Aturan-aturan dalam masing-masing bentuk normalisasi tersebut adalah sebagai berikut (Abdul Kadir, 2001: 54):

1) Bentuk tidak normal (UNF)

Bentuk ini merupakan kumpulan data yang disimpan, tidak ada keharusan mengikuti suatu format tertentu, mungkin saja data tidak lengkap atau terduplikasi dan data dikumpulkan apa adanya.

2) Bentuk normal pertama (1NF)

Suatu tabel dikatakan dalam bentuk normal pertama (1NF) bila setiap kolom bernilai tunggal tunggal untuk setiap baris. Ini berarti bahwa nama kolom yang berulang cukup diwakili oleh suatu nama kolom (tidak perlu ada indeks dalam memberi nama kolom).

3) Bentuk normal kedua (2NF)

Suatu tabel berada dalam bentuk normal kedua (2NF) jika tabel berada dalam bentuk normal pertama, semua kolom bukan kunci primer tergantung sepenuhnya terhadap kunci primer. Suatu kolom disebut tergantung sepenuhnya terhadap kunci primer jika nilai pada suatu kolom selalu bernilai sama untuk suatu nilai kunci primer yang sama.

4) Bentuk normal ketiga (3NF)

Suatu tabel berada dalam bentuk normal ketiga (3NF) jika tabel berada dalam bentuk normal kedua, setiap kolom bukan kunci primer yang tidak memiliki ketergantungan secara transitif terhadap kunci primer.

4. Perangkat Lunak Pengembangan

a. *PHP*

PHP (*PHP: Hypertext Preprocessor*) adalah salah satu bahasa pemrograman yang berjalan dalam sebuah *web server* dan berfungsi sebagai pengolah data pada sebuah *server* (Divisi Penelitian dan Pengembangan MADCOMS, 2004:1).

Dengan menggunakan program *PHP*, sebuah *website* akan lebih interaktif dan dinamis. Data yang dikirim oleh pengunjung *website* akan diolah dan disimpan pada *database web server* dan ditampilkan kembali apabila diakses.

PHP memiliki beberapa keunggulan yaitu:

- 1) Memiliki tingkat akses yang lebih cepat
- 2) Memiliki tingkat *lifecycle* yang cepat sehingga selalu mengikuti perkembangan teknologi internet
- 3) Memiliki tingkat keamanan yang tinggi
- 4) Mampu berjalan di beberapa *server* yang ada, misalnya *Apache*, *Microsoft IIS*, *PWS*, *AOLserver*, *phttpd*, *fhttpd*, dan *Xitami*.
- 5) Mampu berjalan di *Linux* sebagai platform sistem operasi utama bagi *PHP*, namun juga dapat berjalan di *FreeBSD*, *Unixm Solaris*, *Windows*, dan yang lain.

- 6) Mendukung akses ke beberapa *database* yang sudah ada, baik yang bersifat gratis ataupun komersial.
- 7) Bersifat gratis.

Menurut Divisi Penelitian dan Pengembangan MADCOMS (2004:15-18) kode-kode bahasa *PHP* dalam penulisannya menyatu dengan tag-tag HTML dalam satu file. Kode *PHP* diletakkan antara tanda <? atau <?php dan diakhiri dengan tanda ?> sebagai bahasa pemrograman *PHP*. Akan tetapi ada beberapa cara lagi untuk memberikan tanda bahwa kode yang ditulis adalah kode *PHP*, antara lain:

- 1) Model *javascript*, kode *PHP* diawali dengan tag <scriptlanguage="php"> dan diakhiri </script>.
- 2) Model ASP, penulisan kode *PHP* diawali dengan tag <% dan diakhiri dengan %>. Tapi, cara ini berfungsi bila *server* diatur lebih dulu sehingga modul *PHP* mengenalinya.

File yang mengandung kode *PHP* ini akan diberi ekstensi .php atau ekstensi lain yang ditetapkan oleh web server sebagai file *PHP*. Contoh penulisan kode *PHP* yang menyatu dengan tag html:

```
<html>
  <head>
    <title>Program PHP</title>
  </head>
  <body>
    <?php Echo "Mari Belajar PHP" ?>
  </body>
</html>
```

Deni Sutaji (2012: 3) memaparkan bahwa *PHP* memiliki variabel yang digunakan sebagai tempat penyimpanan data sementara. Data tersebut akan hilang

setelah program selesai dieksekusi. Aturan penggunaan nama variabel adalah sebagai berikut:

- 1) Diawali dengan karakter \$.
- 2) Bersifat *case sensitive*, jadi \$Var berbeda dengan \$var atau \$VAR.
- 3) Karakter pertama harus berupa huruf atau garis bawah (_).
- 4) Karakter berikutnya boleh huruf, angka atau garis bawah (_).

Berikut adalah beberapa contoh penulisan dan pendeklarasian variabel:

\$namadepan = "Ali Shodikin";

\$harga_barang = 3500;

\$nilai3 = 3.45;

b. MySQL

Menurut Bunafit Nugroho (2008, 91), *MySQL (My Structured Query Language)* adalah sebuah program pembuat dan pengelola basis data atau yang sering disebut *DBMS (Database Management System)*.

Salah satu kelebihan dari *MySQL* adalah merupakan program pengakses basis data yang bersifat jaringan sehingga dapat digunakan untuk aplikasi *multiuser*. Selain itu *MySQL* menggunakan bahasa *query standar SQL (Structured Query Language)*.

Menurut Yakub (2008:99), *SQL (Structured Query Language)* adalah suatu bahasa komputer yang mengikuti standar *ANSI (American National Standard Institute)*, yaitu sebuah bahasa standar yang digunakan untuk mengakses dan melakukan manipulasi sistem basis data.

Yakub (2008:101-116) mengemukakan bahwa secara umum *SQL* dibagi menjadi tiga, yaitu:

1) *Data Definition Language (DDL)*

DDL merupakan suatu perintah yang berfungsi untuk mendefinisikan atribut-atribut basis data, tabel, atribut serta hubungan. Beberapa sintaks yang sering dijumpai adalah sebagai berikut:

- a) CREATE TABLE, bertugas untuk membuat tabel
- b) CREATE INDEX, bertugas untuk membuat suatu indeks dalam tabel
- c) DROP TABLE, bertugas untuk menghapus suatu tabel
- d) DROP INDEX, bertugas untuk menghapus suatu indeks tabel
- e) ALTER TABLE, bertugas untuk mengubah struktur suatu tabel

2) *Data Manipulation Language (DML)*

DML adalah sekumpulan perintah yang digunakan untuk melakukan proses manipulasi data pada suatu basis data. Berikut penjelasan sintaks-sintaks yang digunakan pada *DML*:

- a) INSERT, bertugas untuk menambahkan data ke dalam suatu tabel dalam suatu basis data
- b) UPDATE, bertugas untuk merubah (*update*) data dalam suatu tabel dalam suatu basis data
- c) SELECT, bertugas untuk mengakses data dari suatu tabel dalam basis data
- d) DELETE, bertugas untuk menghapus data dari suatu tabel dalam basis data

3) *Data Control Language (DCL)*

DCL merupakan kelompok perintah yang berfungsi untuk mengendalikan pengaksesan data, *DCL* digunakan untuk menangani masalah keamanan dalam *database server*. Sintaks yang digunakan dalam *DCL* adalah sebagai berikut:

- a) GRANT, digunakan untuk memberikan atau mengijinkan seorang user untuk mengakses tabel dalam basis data tertentu.
- b) REVOKE, digunakan untuk mencabut suatu hak akses dalam basis data tertentu.

c. SMS Gateway

Menurut Sofyan Maulana (2015:7-8), *gateway* dapat diartikan sebagai jembatan penghubung antara satu sistem dengan sistem lain yang berbeda sehingga dapat terjadi suatu pertukaran data antara sistem tersebut. Dengan demikian, *SMS Gateway* adalah suatu penghubung untuk lalu lintas data *SMS*, baik yang dikirim maupun yang diterima.

SMS Gateway adalah suatu teknologi pengolahan *SMS* yang dilakukan secara terkomputerisasi dan memanfaatkan layanan *SMS* itu sendiri untuk berbagai keperluan serta tujuannya masing-masing.

Terdapat banyak aplikasi yang dapat digunakan untuk membangun *SMS Gateway* antara lain *Gammu*, *NowSMS* dan *PlaySMS*. *Gammu* adalah salah satu aplikasi *SMS Gateway* yang cukup populer. Kelebihan *Gammu* dari aplikasi *SMS Gateway* lain antara lain (Priyatna, 2013):

- 1) *Gammu* dapat berjalan di Windows dan Linux.
- 2) *Gammu* kompatibel dengan bermacam-macam *device*.

- 3) *Gammu* kompatibel dengan *device* yang terhubung dengan kabel data *USB* maupun serial.
- 4) *Gammu* bersifat *Open Source* atau gratis.
- 5) *Gammu* dapat menggunakan *MySQL*, *PostgreSQL* dan *ODBC* sebagai basis datanya.
- 6) Dokumentasi *Gammu* lengkap dan jelas.

5. Jaminan Kualitas Perangkat Lunak

Jaminan kualitas perangkat lunak (*Software Quality Assurance/ SQA*) adalah rangkaian kegiatan yang dirancang untuk mengevaluasi proses di mana produk dikembangkan atau dirangkai. Tujuan dari jaminan kualitas adalah untuk memberikan data yang diperlukan oleh manajemen dan menginformasikan masalah kualitas produk, sehingga dapat memberikan kepastian dan konfidensi bahwa kualitas produk dapat memenuhi sasaran, tidak hanya berkualitas menurut pengembang tapi juga berkualitas dan sesuai dengan keinginan pengguna (Galin,2004:26).

a. Faktor-Faktor Penentu Kualitas Perangkat Lunak

Menurut McCall (Pressman, 509-510), faktor-faktor penentu kualitas perangkat lunak adalah sebagai berikut:

- 1) *Correctness*, sejauh mana suatu perangkat lunak memenuhi spesifikasi dan tujuan penggunaan perangkat lunak dari *user*.
- 2) *Reliability*, sejauh mana keakuratan suatu perangkat lunak dalam melaksanakan fungsinya.

- 3) *Efficiency*, banyaknya sumber daya komputasi dan kode program yang dibutuhkan suatu perangkat lunak untuk melakukan fungsinya.
- 4) *Integrity*, sejauh mana akses ke perangkat lunak dan data oleh pihak yang tidak berhak dapat dikendalikan.
- 5) *Usability*, usaha yang diperlukan untuk mempelajari, mengoperasikan, menyiapkan input, dan mengartikan output dari perangkat lunak.
- 6) *Maintainability*, usaha yang diperlukan untuk menetapkan dan memperbaiki kesalahan dalam program.
- 7) *Testability*, usaha yang diperlukan dalam pengujian program untuk memastikan bahwa program melaksanakan fungsi yang ditetapkan.
- 8) *Flexibility*, usaha yang diperlukan untuk memodifikasi program operasional.
- 9) *Portability*, usaha yang diperlukan untuk memindahkan program dari perangkat keras atau lingkungan sistem perangkat lunak tertentu ke yang lainnya.
- 10) *Reusability*, tingkat kemampuan program atau bagian dari program yang dapat dipakai ulang dalam aplikasi lainnya, berkaitan dengan paket dan lingkup dari fungsi yang dilakukan oleh program.
- 11) *Interoperability*, usaha yang diperlukan untuk menggabungkan satu sistem dengan yang lainnya.

b. Spesifikasi Kebutuhan Perangkat Lunak

Secara sederhana, Spesifikasi Kebutuhan Perangkat Lunak (*Software Requirement Specifications/ SRS*) adalah dokumen yang menjelaskan tentang berbagai kebutuhan yang harus dipenuhi oleh suatu software. Dokumen ini dibuat

oleh *developer* (pengembang *software*) setelah menggali informasi dari calon pemakai *software*.

Pembuatannya mengikuti standar yang ada dan paling diakui oleh para praktisi rekayasa *software* di dunia. Oleh karena itu, standar yang akan dibahas disini adalah standar dari *IEEE (Institute of Electrical and Electronics Engineers)* (Wisnu, 2012).

SRS yang baik akan bermanfaat bagi *customer*, *supplier*, atau perorangan. Manfaat-manfaat tersebut antara lain sebagai berikut:

- 1) Sebagai bentuk perjanjian antara *customer* dan *supplier* tentang *software* apa yang akan dibuat.
- 2) Mengurangi beban dalam proses pengembangan *software*.
- 3) Sebagai bahan perkiraan biaya dan rencana penjadwalan.
- 4) Sebagai dasar validasi dan verifikasi *software* di ujung penyelesaian proyek nantinya.
- 5) Memfasilitasi transfer, semisal *software* tersebut ingin ditransfer ke pengguna atau mesin-mesin yang lain. *Customer* akan lebih mudah jika ingin mentransfer *software* ke bagian-bagian lain dalam organisasinya. Bahkan, jika terjadi pergantian personil *developer*, proyek dapat mudah ditransfer ke personil baru dengan memahami *SRS* ini.
- 6) Mendasari perbaikan produk *software* di kemudian hari. Jadi, *SRS* boleh diperbaiki dengan alasan dan mekanisme tertentu serta atas kesepakatan antara *customer* dan *developer*.

Dengan *SRS*, pengguna dapat mencerahkan semua keinginannya terkait *software* dengan jelas dan akurat sehingga *developer* dapat memahami apa yang diinginkan pengguna dengan tepat. Standar ini dapat membantu dalam mengembangkan dokumen *SRS* yang baku untuk perusahaan pribadi, membantu membuat dokumen *SRS* dengan format dan isi yang standar (minimal), serta membantu mengembangkan rincian-rincian pendukung lainnya.

6. Layanan Internet Mahasiswa UNY

Layanan Internet Mahasiswa UNY. (LIMUNY) adalah sebuah unit usaha dibawah naungan Badan Pengelolaan dan Pengembangan Usaha (BPPU) Universitas Negeri Yogyakarta yang bergerak di bidang pelayanan jasa informasi khususnya internet. LIMUNY didirikan pada tahun 2006. Hingga tahun 2016 LIMUNY memiliki perangkat komputer sebanyak 332 unit. Tugas utama LIMUNY adalah dapat memberikan pelayanan berupa penyediaan jasa akses internet kepada mahasiswa UNY dan masyarakat dengan cara yang sangat mudah dan nyaman serta biaya yang sangat terjangkau.

Pada tahun 2016, terdapat 37 operator LIMUNY yang aktif yang terbagi menjadi 6 orang operator *hardware*, 7 orang operator *service center*, dan 24 operator *billing*. Untuk operator *billing*, terdapat lima posisi yaitu *billing 1*, *billing 2*, *billing 3*, *helpdesk 1*, dan *helpdesk 2*. Sistem kerja di LIMUNY berupa *shift* dimana dalam satu hari terdapat empat shift untuk *billing* dan dua *shift* untuk *hardware* dan *service center* dengan pembagian operator yang bertugas seperti pada Tabel 4 dan Tabel 5.

Tabel 4. Pembagian *Shift Billing* dan *Hardware*

Shift	Jam Kerja Billing dan Hardware	Posisi yang Bertugas
Pagi	06.30 - 12.30 WIB	<i>billing 1, billing 2, billing 3, helpdesk 1, helpdesk 2, dan hardware</i>
Siang	12.30 - 17.30 WIB	<i>billing 1, billing 2, billing 3, helpdesk 1, helpdesk 2, dan hardware</i>
Sore	17.30 - 22.30 WIB	<i>billing 1, billing 2, billing 3, helpdesk 1,</i>
Malam	22.30 - 06.30 WIB	<i>billing 1, billing 2, billing 3</i>

Tabel 5. Pembagian *Shift Service Center*

Shift	Jam Kerja Service Center	Posisi yang Bertugas
Pagi	07.00 - 12.30 WIB	<i>Frontliner dan teknisi</i>
Siang	12.30 - 17.30 WIB	<i>Frontliner dan teknisi</i>

B. Kerangka Berpikir

Salah satu masalah yang terjadi pada sistem administrasi operator LIMUNY adalah informasi yang berkaitan dengan operator hanya dapat diakses terbatas di lingkungan LIMUNY. Informasi-informasi tersebut meliputi jadwal *shift*, gaji, poin pelanggaran, dan masalah presensi.

Dalam proses melihat dan menukar jadwal, operator harus datang ke LIMUNY untuk mencoret jadwal otentik ataupun untuk melihat keseluruhan jadwal yang sudah mengalami perubahan. Jumlah operator yang tidak sedikit dan kesibukan operator yang berbeda-beda mengakibatkan pertukaran jadwal sangat sering terjadi sehingga jadwal otentik menjadi semakin susah terbaca sehingga menyulitkan pengecekan gaji, poin, dan permasalahan presensi.

Dalam mengecek gaji dan poin operator juga hanya bisa mengakses informasi tersebut pada mesin presensi di LIMUNY. Gaji operator dihitung berdasarkan jumlah jam kerja operator dalam satu bulan dikurangi potongan jika datang terlambat. Sedangkan poin pelanggaran diperoleh jika operator datang

terlambat, bolos *shift*, dan atau bolos rapat bulanan. Bila poin pelanggaran sudah mencapai batas tertentu, operator akan memperoleh surat peringatan. Selain itu, apabila terjadi permasalahan saat presensi dengan mesin presensi, operator juga harus mengisi formulir permasalahan presensi dan harus diverifikasi oleh saksi dan Koordinator Sumber Daya Manusia.

Keterbatasan dalam memperoleh informasi-informasi ini memunculkan gagasan untuk membangun sebuah sistem informasi berbasis *website* yang dilengkapi dengan *SMS Gateway* guna memfasilitasi operator dalam memperoleh informasi yang berkaitan dengan operator. Diharapkan dengan sistem informasi ini informasi-informasi tersebut dapat diakses setiap saat dan dari mana saja. Selain itu, dengan sebuah sistem informasi pendataan dan pengolahan informasi akan lebih rapi dan efisien karena data disimpan dalam suatu sistem basis data.