

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Dari beberapa pengamatan yang telah dilakukan sudah ada beberapa program aplikasi yang mirip dengan program yang akan di buat, tetapi masih banyak kekurangan, cara pemakaian masih terlalu susah dan sering error. Maka dari itu penulis membangun ulang aplikasi untuk mengatasi masalah tersebut untuk menyelesaikan masalah. Saya sampaikan beberapa aplikasi yang mungkin mirip degan sms gateway :

1. "Perangkat Lunak SMS Gateway untuk Customer Relationship Management" yang di buat oleh : Ema Utami yang menggunakan SMS gateway untuk perusahaan distributor makanan ringan, yang digunakan untuk :
 - mempromosikan produk-produk untuk pelanggan
 - mengingatkan untuk acara meeting perusahaan
 - memberikan laporan kegiatan harian
 - melakukan follow up pada pelanggan

sistem ini sangat membantu dalam merekap dan membantu meringankan beban pegawai perusahaan tersebut.

2. "Sistem Informasi Akademik Berbasis SMS Gateway pada SMK Batik Purworejo" yang dirancang oleh Yuli Astuti sistem ini di rancang untuk

mempermudah guru dalam merekap nilai dan nilai raport yang biasanya melihat nilai butuh waktu 30 menit sampai 1 jam, maka dengan di bentuknya sistem ini, output yang dikeluarkan hanya tidak lebih dari 1 menit.

2.2 Dasar Teori

Aplikasi yang digunakan adalah pengembangan dari aplikasi sebelumnya, aplikasi yang dirancang sekarang lebih mudah di gunakan dan praktis sekali. Dengan begitu user sebagai operator bisa menggunakan dengan nyaman, aplikasi SMS Gateway di gunakan karena mudah penggunaanya dan sudah berbentuk aplikasi yang sangat mudah di mengerti dan sederhana.

2.3 Java

2.3.1 Pengertian dan Sejarah Java

Bahasa pemrograman Java pertama lahir dari The Green Project, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan versi yang dinamakan Oak. Proyek ini dimotori oleh Patrick Naughton, Mike Sheridan, James Gosling dan Bill Joy, beserta sembilan pemrogram lainnya dari Sun Microsystems. Salah satu hasil proyek ini adalah maskot Duke yang dibuat oleh Joe Palrang.

Java adalah bahasa pemrograman berorientasi-objek yang serbaguna. Bahasa ini menyediakan sejumlah perluasan yang didukung pengembangan aplikasi GUI, dan juga pengembangan aplikasi client/server terhadap jaringan local (LAN) atau jaringan luas (WAN). Java merupakan kumpulan bentuk dan

sifat-sifat yang telah dikenal pada bahasa-bahasa pemrograman seperti C, C++, Objective-C, Smalltalk, dan Common Lisp. Dari Objective-C, Java memakai konsep antarmuka dari Smalltalk, Java meminjam model ekstensibilitas sewaktu pengesekuan, manajemen memori dinamis, dan beberapa pengekseskuan secara bersamaan (*multithreading*) / terhadap hal-hal ini, Java menambahkan keamanan, paradigma pemrograman yang sederhana, dan banyak lagi.

Disamping itu source code java bisa di pindah-pindah antar sistem komputer, bahkan hasil kompilasinya bisa di jalankan di berbagai sistem komputer. Hal ini di mungkinkan dengan adanya bytecodes. Setiap program yang di tulis dengan bahasa java. Hasil kompilasinya berupa bytecode, yaitu mesin (Machine Code) tapi tidak spesifik untuk satu jenis proses tertentu.

Umumnya jika kita menulis program (misalnya dalam bahasa C++) hasil kompilasi pada satu sistem komputer, misalnya PC, kompiler akan menterjemahkan source code kedalam bahas mesin yang spesifikasi untuk prosesor yang ada di PC. Jika hendak di jalankan di sistem lain, Macintosh yang bersangkutan dan terkadang harus melakukan pengendalian source program terlebih dahulu.

Alasan lain kepopuleran Java di antara pemrograman adalah bahwa secara arsitekturnya netral. Ia merupakan bahasa penterjemah, dan anda dapat menjalankan program Java pada sebarang platform yang memiliki interpreter dan lingkungan *runtime*. Ini memungkinkan pemrograman menulis kode (dan hanya menulis sekali saja).

Java dapat bekerja pada lingkungan sebagai berikut :

Platform perangkat keras	Sistem Operasi
SPARC	SOLARIS
Intel	Windows XP, Vista, Windows 7
Macintosh	MacOS 7.5 dan lain-lain
Linux	Ubuntu, Kubuntu dan Lain-lain

Tabel 2.1 Platform / sistem operasi yang didukung oleh java

2.3.2 Kompiler Java dan Interpreter Java

Walaupun Java adalah bahasa penerjemah, Dalam proses menjalankan program Java harus dikompilasi terlebih dahulu. **Kompiler** Java dikenal dengan (*javac*) berfungsi untuk mengubah kode sumber program menjadi *bytecode* yang dapat dieksekusi dalam lingkungan *runtime* Java. Sedangkan **Interpreter** Java (*java*) menjalankan *bytecode* terkompilasi pada sistem lokal. Program *javac* dan *java* berada dalam direktori *bin* pada Java Development Kit (JDK). Lokasi file dari kompiler dan intrepeter terdapat pada direktori */local/jdk/bin*.

2.3.3 Versi Awal

Versi awal Java ditahun 1996 sudah merupakan versi release sehingga dinamakan Java Versi 1.0. Java versi ini menyertakan banyak paket standar awal diantaranya :

1. `Java.lang` adalah peruntukan kelas elemen-elemen dasar.
2. `Java.io` adalah peruntukan kelas input dan output.
3. `Java.util` adalah peruntukan kelas pelengkap seperti kelas struktur data dan kelas-kelas penanggalan.
4. `Java.net` adalah peruntukan kelas TCP/IP, yang memungkinkan berkomunikasi dengan komputer lain menggunakan jaringan TCP/IP.
5. `Java.awt / java.swing` adalah kelas dasar untuk aplikasi antarmuka dengan pengguna (GUI).
6. `Java.applet` adalah kelas dasar aplikasi antarmuka untuk diterapkan pada penjelajah web.

2.3.4 Kelebihan Java

Java memiliki banyak kelebihan diantaranya sebagai berikut :

- a) *Multiplatform*, Kelebihan utama dari java ialah dapat dijalankan di beberapa platform / sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan di mana saja.

- b) OOP (*Object Oriented Programming* – Pemrograman Berorientasi Objek) artinya semua aspek yang terdapat di Java adalah Objek. Java merupakan salah satu bahasa pemrograman berbasis obyek secara murni. Semua tipe data diturunkan dari kelas dasar yang disebut *Object*. Hal ini sangat memudahkan pemrogram dalam mendesain, membuat, mengembangkan dan mengalokasikan kesalahan program dengan basis Java.
- c) Perpustakaan Kelas yang lengkap, Java terkenal dengan kelengkapan *library*/perpustakaan (kumpulan program-program yang disertakan dalam pemrograman java) yang membantu pemrograman dalam menyelesaikan programnya.
- d) Bergaya C++, memiliki sintaks seperti bahasa pemrograman [C++] sehingga menarik banyak pemrogram C++ untuk pindah ke Java. Saat ini pengguna Java sangat banyak, sebagian besar adalah pemrogram C++ yang pindah ke Java.
- e) Pengumpulan sampah otomatis, memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas).

2.3.5 Kekurangan Java

Selain mempunyai kelebihan Java juga mempunyai kelemahan, diantaranya :

- a) Tulis sekali, perbaiki dimana saja Masih ada beberapa hal yang tidak kompatibel antara platform satu dengan platform lain. Untuk J2SE, misalnya *SWT-AWT bridge* yang sampai sekarang tidak berfungsi pada Mac OS X.
- b) Mudah didekompilasi, Dekompilasi adalah proses mengembalikan dari kode jadi menjadi kode sumber. Dengan mudahnya proses dikompilasi maka program akan lebih sulit untuk disembunyikan dan mudah untuk dibajak/*dirverse-engineer*.
- c) Penggunaan memori yang banyak. Penggunaan memori untuk program berbasis Java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal.

2.3.6 Contoh Kode Program Sederhana Dalam Java

Contoh program untuk menampilkan hello world menggunakan pemrograman java adalah sebagai berikut :

```

public class Hello
{
    /**
     * My first java program
     */
    public static void main(String[] args) {

        //Menampilkan kata "Hello world" dilayar
        System.out.println("Hello world!");
    }
}

```

Gambar 2.1 contoh sederhana kode java

2.1.1

2.4 SMS (Short Message Service)

SMS adalah merupakan teknologi yang memungkinkan untuk menerima dan mengirim pesan berupa karakter sebanyak 160 karakter antara telepon bergerak (ponsel). Teknologi ini pertama kali di perkenalkan pada tahun 1992 di eropa oleh ETSA (*Europe Telecommunication Standards Institute*) dan pada awalnya menjadi standard untuk telepon wireless yang berbasis GSM (*Global Sistem for Mobile Communication*).

2.4.1 Kelebihan SMS

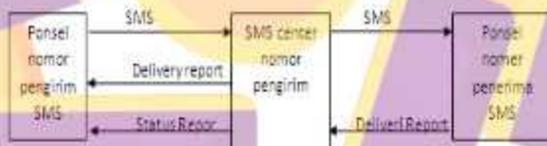
Ada beberapa alasan mengapa orang begitu nyaman menggunakan teknologi SMS ini untuk berkomunikasi satu dengan yang lainnya, yaitu :

- a. SMS dapat dibaca dan dikirim kapanpun di manapun kita berada jika daerah tersebut terjangkau oleh jaringan GSM.

- b. SMS dapat dikirimkan ke nomor tujuan meskipun nomor tujuan itu tidak aktif, kita tidak perlu mengecek apakah nomor tujuan aktif atau tidak.
- c. Untuk saat ini SMS dianggap lebih murah dan praktis.
- d. Saat ini hampir semua orang memiliki handphone yang bisa di gunakan untuk SMS.

2.4.2 Mekanisme Kerja SMS

SMS yang kita kirim dari handphone akan melalui beberapa proses sebelum SMS tersebut sampai ke nomor tujuan. Ketika kita mengirim sebuah pesan, SMS tersebut dikirim dulu ke server yang disebut juga SMS Center yang bertugas mengirim SMS ke nomor tujuan. Berikut adalah gambar mekanisme pengiriman SMS



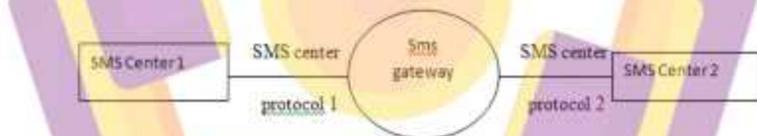
Gambar 2.1 Mekanisme Kerja SMS

SMS yang dikirim oleh nomor pengirim akan di sampaikan ke dalam SMS Center operator ke nomor pengirim, kemudian SMS Center akan mengirimkan SMS ke nomor yang di tuju secara langsung. Nomor penerima akan mengirim sebuah *delivery report* yang menyatakan bahwa SMS telah di terima oleh SMS Center. SMS Center kemudian akan meneruskan report tersebut ke nomor pengirim SMS, di sertai status report dari proses pengiriman SMS tersebut.

2.5 SMS GATEWAY

Istilah gateway, bila dilihat arti secara kamus inggris-indonesia adalah pintu gerbang. Namun pada dunia komputer, gateway bisa diartikan sebagai jembatan penghubung antar satu sistem dengan sistem yang lain yang berbeda, sehingga dapat terjadi pertukaran data antar sistem tersebut. Dengan demikian SMS gateway dapat diartikan sebagai penghubung untuk lalu lintas data SMS, baik yang dikirimkan maupun yang diterima.

Awal-awal pada teknologi SMS digunakan, SMS gateway di manfaatkan untuk menjembatani antar SMS server karena perbedaan bahasa (protocol), sebagai gambarannya adalah seperti gambar di bawah ini:



Gambar 2.3 Mekanisme Kerja SMS Gateway dulu

Seiring dengan perkembangan teknologi komputer, sekarang ini masyarakat lebih mengartikan SMS gateway sebagai jembatan komunikasi yang menghubungkan perangkat komunikasi (handphone) dengan perangkat lunak lainnya yang dibutuhkan. Kemudian pengertian SMS gateway lebih mengarah pada sebuah program yang mengkomunikasikan antara sistim operasi komputer dengan perangkat komunikasi yang terpasang untuk mengirim dan menerima

SMS. Pengertian SMS adalah tersebut diatas dapat digambarkan seperti dibawah ini :



Gambar 2.4 Mekanisme kerja SMS gateway

2.6 Program SMS Gateway Windows

SMS (Short Messaging Services) Data SMS (Short Messaging Services) yang kita kirim atau yang kita terima sebenarnya memiliki format tersendiri untuk dapat diterjemahkan oleh sebuah mobile phone. Format atau mode yang dipakai untuk mengirim dan menerima SMS sebenarnya ada dua yaitu mode text dan mode PDU (Protocol Data Unit). Akan tetapi, sistem mode text tidak didukung oleh semua operator GSM maupun terminal

1. Text Mode Mode ini adalah cara termudah untuk mengirim pesan. Pada mode teks pesan yang kita kirim tidak dilakukan konversi. Teks yang dikirim tetap dalam bentuk aslinya dengan panjang mencapai 160 (7 bit default alphabet) atau 140 (8 bit) karakter. Sesungguhnya, mode teks adalah hasil encode yang direpresentasikan dalam format PDU. Kelemahannya, kita tidak dapat menyisipkan gambar dan nada dering ke dalam pesan yang akan dikirim serta terbatasnya tipe encoding.

2. PDU (Protocol Data Unit) Mode PDU mode adalah format message dalam heksadesimal octet dan semi-desimal octet dengan panjang mencapai 160 (7 bit default alphabet) atau 140 (8 bit) karakter. Kelebihan menggunakan mode PDU adalah kita dapat melakukan encoding sendiri yang tentunya harus pula didukung oleh hardware dan operator GSM, melakukan kompresi data, menambahkan nada dering dan gambar pada pesan yang dikirim. Beberapa tipe encoding yang umum digunakan adalah "PCCP437", "PCDN", "8859-1", "IRA" dan "GSM". Dengan adanya SMSC ini dapat mengetahui status dari pesan SMS yang telah dikirim, apakah telah sampai atau gagal diterima oleh handphone tujuan. Apabila handphone tujuan dalam keadaan aktif dan dapat menerima pesan SMS yang dikirim, ia akan mengirim kembali pesan konfirmasi ke SMSC yang menyatakan bahwa pesan telah diterima. Kemudian SMSC mengirimkannya kembali status tersebut kepada si pengirim. Jika handphone dalam keadaan mati, pesan yang dikirimkan akan disimpan pada SMSC sampai period-validity terpenuhi. Koneksi ke SMSC Untuk dapat mengirim dan menerima pesan, kita harus melakukan koneksi ke SMSC. Ada beberapa cara untuk melakukan koneksi ke SMSC antara lain : Menggunakan terminal baik berupa GSM modem atau handphone. Cara ini adalah yang paling mudah tetapi memiliki kekurangan antara lain jumlah pesan yang dikirim per menit sangat terbatas (sekitar 6-10 pesan per menit). Untuk mengantisipasi hal ini biasanya digunakan lebih dari satu terminal.

Koneksi langsung ke SMSC Dengan melakukan koneksi langsung ke SMSC kita dapat mengirim pesan dalam jumlah banyak, dapat mencapai sekitar 600 SMS per menit bergantung pada kapasitas dari SMSC itu sendiri. Untuk melakukan koneksi ke SMSC diperlukan protocol penghubung. Protocol yang umum digunakan adalah UCP, SMPP, CIMD2, OIS dan TAP. Masing-masing operator GSM menyediakan tipe protocol yang berbeda-beda.

Menggunakan software Bantu Saat ini banyak vendor telekomunikasi menawarkan software bantu untuk melakukan koneksi ke SMSC, dari yang bersifat freeware, open source sampai dengan yang komersial. Pemilihan koneksi ke SMSC biasanya disesuaikan dengan jumlah pesan SMS yang akan dikirim. Gambar dibawah ini menunjukkan skema tipe koneksi ke SMSC.



Gambar 2.5 Mode Pengiriman dan Penerimaan SMS dengan PDU

2.7 NetBeans

NetBeans mengacu pada dua hal, yakni platform untuk pengembangan aplikasi desktop java, dan sebuah Integrated Development Environment (IDE) yang dibangun Menggunakan platform NetBeans. Platform Netbeans memungkinkan aplikasi dibangun dari sekumpulan komponen perangkat lunak

moduler yang disebut "modul". Sebuah modul adalah suatu arsip Java (Java archive) yang memuat kelas-kelas Java untuk berinteraksi dengan NetBeans Open API dan file manifestasi yang mengidentifikasinya sebagai modul. Aplikasi yang dibangun dengan modul-modul dapat dikembangkan dengan menambahkan modul-modul baru. Karena modul dapat dikembangkan secara independen, aplikasi berbasis platform NetBeans dapat dengan mudah dikembangkan oleh pihak ketiga. Secara mudah dan powerful. Netbeans IDE adalah IDE open source yang ditulis sepenuhnya dengan bahasa pemrograman Java menggunakan platform Netbeans. Netbeans IDE mendukung pengembangan semua tipe aplikasi Java(J2SE, web, EJB, dan Aplikasi Mobile). Fitur lainnya adalah sistem proyek berbasis Ant, kontrol versi, dan refactoring.

Versi terbaru saat ini adalah Netbeans IDE 6.7.1 yang dirilis Oktober 2009 mengembangkan fitur-fitur Java EE yang sudah ada(termasuk Java Persistence support, EJB-3 dan JAX-WS). Sementara paket tambahannya, NetBeans Enterprise Pack mendukung pengembangan aplikasi perusahaan Java EE 5, meliputi alat desain visual SOA, skema XML, web service dan pemodelan UML. NetBeans C/C++ Pack

Mendukung proyek C/C++. Modularitas semua fungsi IDE disediakan oleh modul-modul. Tiap modul menyediakan fungsi yang didefinisikan dengan baik, seperti dukungan untuk bahasa pemrograman Java, editing, atau dukungan bagi CVS. NetBeans memuat semua modul yang diperlukan dalam pengembangan Java dalam sekali download, memungkinkan pengguna untuk mulai bekerja sesegera mungkin. Modul-modul juga mengizinkan NetBeans untuk

bisa dikembangkan. Fitur-fitur baru, seperti dukungan untuk bahasa pemrograman lain, dapat ditambahkan dengan menginstal modul tambahan. Sebagai contoh, Sun Studio, Sun Java Studio Enterprise, dan Sun Java Studio Creator dari Sun Microsystems semuanya berbasis NetBeansIDE.

Paket-paket Tambahan Netbeans IDE, yaitu :

- a. Netbeans Enterprise.
- b. Pack Netbeans Profiler.
- c. Netbeans Enterprise Pack.
- d. Netbeans Ruby Pack.
- e. Netbeans JavaScript Editor.

2.8 UML (Unified Modelling Language)

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. UML menggunakan *class* dan *operation* dalam konsep dasarnya. Maka ia lebih cocok digunakan untuk pembuatan aplikasi yang menggunakan pemrograman beorientasi objek. UML mempunyai berbagai jenis diantaranya :

1. Use case, menggambarkan fungsionalitas yang diharapkan dari sebuah sistem.
2. Class diagram, spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek.
3. Statechart diagram, transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima.
4. Activity diagram, menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.
5. Sequence diagram, menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu.
6. Colaboration diagram, menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*.
7. Component diagram, menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya.
8. Deployment diagram, menggambarkan detail bagaimana komponen *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 fisik.

2.9 Konsep Dasar Database

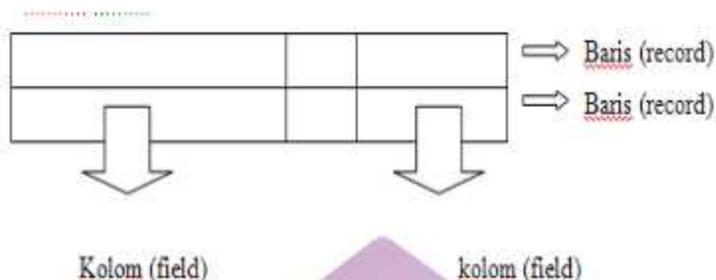
2.9.1 Definisi Database

Database adalah sekumpulan data yang disusun dalam bentuk (beberapa) table yang saling berkaitan maupun berdiri sendiri. Dalam contoh kasus adalah misalnya seorang guru mempunyai data-data siswa yang harus dikelola sedemikian rupa sehingga mudah diolah. Data-data tersebut bila disusun dan di pilih berdasarkan kategori tertentu, akan menjadi sebuah database. Penyusunan data base bisa dilakukan secara sederhana dengan menuliskan data-datanya pada buku khusus. Atau bisa juga dengan cara modern dengan memanfaatkan teknologi komputer. Cara apapun yang digunakan, untuk mempermudah penyusunan dan penampilan data, maka digunakan bentuk table.

2.9.2 Tabel, kolom (field) dan baris (row)

Tabel adalah sebagai alat untuk mengelompokkan data di dalam database yang berisi kolom dan baris agar mudah lihat dan di baca. Table terdiri atas baris dan kolom. Dalam database, baris adalah sebagai record dan kolom sebagai field.

Gambar di bawah ini adalah bentuk umum dari sebuah table :



Gambar 2.6 Bentuk Tabel

2.9.3 Kolom Multivaluad dan Multipart

Multivalued field di dalam database adalah sebuah kolom yang berisi lebih dari satu nilai yang sejenis. Sedangkan multipart field adalah kolom yang berisi lebih dari satu nilai yang berbeda jenis. Sebagai contoh :

Siswa	Alamat	Telepon	Jumlah absen
Joko	Jl. Mawar No.45 Magga Etan Rt/w 02/04 sukoharjo	0819456765678	6
Sitorus	Jl. Sudirman No.55 Rambutan Rt/w 09/06 sukoharjo	0867434325667	3
...

Diagram illustrating the structure of a table with multivalued and multipart fields. The table has four columns: Siswa, Alamat, Telepon, and Jumlah absen. Below the table, arrows point to the 'Alamat' and 'Telepon' columns, which are labeled as 'Multipart' and 'Multivaluade Field' respectively.

Gambar 2.7 Bentuk Kolom Multivaluad dan Multipart

2.9.4 Normalisasi Database

Adalah proses mengubah suatu tabel yang besar dan kompleks menjadi suatu tabel-tabel yang lebih kecil dan sederhana. Tujuan dari normalisasi adalah untuk mengurangi pemasukan data yang berulang (redundant data), duplikasi data, selain itu juga untuk menghindari terjadinya masalah pengisian data, mengubah, maupun menghapus data.

2.9.5 Primary Key dan Foreign Key key Field

Adalah kolom khusus yang memiliki fungsi sebagai pembeda *record* satu dengan yang lainnya. Key field dapat dibagi menjadi 2 (dua) kategori yaitu *primary key* dan *foreign key*. *primary key* adalah suatu kolom (field) yang menjadi titik acuan pada sebuah tabel atau mewakili sebuah tabel, bersifat unik dalam artian tidak ada nilai yang sama dalam satu tabel. *Foreign key* adalah suatu kolom (field) dalam tabel yang digunakan sebagai "kaitan" tabel lainnya sehingga dapat dibuat sebuah hubungan antara tabel tersebut dengan tabel lainnya.

2.9.6 Kriteria Tabel Ideal

Database bisa dikatakan baik jika sudah minimal 80% terhindar dari permasalahan yang muncul dikemudian hari. Berikut adalah kriteria tabel yang baik dan ideal :

- a. Nama tabel harus memberikan gambaran yang tepat dan benar tentang isi tabel tersebut.
- b. Isi tabel hanya memiliki satu subjek database atau topik saja.

- c. Tabel harus memiliki satu *primary key*.
- d. Kolom hanya memiliki satu jenis nilai saja.
- e. Tabel tidak mengandung kolom yang berisi hasil penghitungan suatu proses. Kecuali untuk relasi antara *primary key* dan *foreign key* masih bisa dilakukan.
- f. Tidak ada data kolom yang identik di dalam tabel lainnya, jika tidak bisa dihindarkan, maka minimalkan pengulangan data.

2.9.7 Jenis-jenis tabel berdasarkan fungsinya

a. Tabel Data

Adalah sebuah tabel yang menampung data-data yang digunakan sebagai informasi. Tabel ini bersifat dinamis yaitu sering diubah, ditambah dan dihapus.

b. Tabel Validasi

Adalah sebuah tabel yang menampung data-data yang memiliki kriteria tertentu dan biasanya digunakan oleh data pada tabel lainnya sebagai validasi. Tabel validasi ini disebut juga tabel referensi (*reference table*) dan tabel pencarian (*look-up table*). Sifat tabel ini biasanya statis.

c. Tabel Penghubung

Adalah sebuah tabel yang berfungsi sebagai “jembatan” antara dua tabel yang mengalami relasi many-to-many. Dengan adanya tabel penghubung maka relasi yang terjadi akan berubah dari many-to-many menjadi one-to-many.

2.10 Database MySQL

2.10.1 Teori Database

MySQL merupakan software sistem manajemen database (Database Management Sistem/DBMS) yang sangat populer dikalangan pemrograman web, terutama dikalangan Linux. Tetapi software database ini kini telah tersedia juga pada platform sistem operasi Windows. Software database MySQL kini dilepas sebagai software manajemen database yang Open Source.

2.10.2 DDL (Data Definition Language)

Merupakan kelompok perintah yang digunakan untuk melakukan pendefinisian tabel. Dengan kelompok perintah dalam DDL ini maka kita dapat membuat tabel, mengubah strukturnya, menghapus tabel, membuat indeks untuk tabel.

Adapun contoh dari DDL adalah sebagai berikut :

a. CREATE DATABASE

Perintah create database digunakan untuk membuat database baru
sintaks:

Nama database diawali dengan menggunakan huruf, kemudian di ikuti dengan kombinasi huruf dan karakter, sebaiknya tidak mengandung spasi dan tanda baca. Untuk menggunakan database tersebut di gunakan perintah :

```
USE nama_database;
```

b. CREATE TABLE

Perintah create table digunakan untuk menyimpan data. Cara membuat tabel adalah dengan menggunakan perintah:

```
CREATE TABLE nama_table  
(Field 1 Tipe data 1, field 2, tipe data 2, ...);
```

c. ALTER TABLE

Untuk mengubah field baru pada tabel digunakan perintah berikut :

```
ALTER TABLE nama_tabel ADD fieldbaru,tipenya;
```

d. DROP TABLE

Apabila table yang telah dibuat tidak diperlukan lagi maka dapat dihapus dengan perintah :

```
DROP TABLE nama_tabel;
```

e. DROP DATABASE

Untuk menghapus database yang tidak diperlukan lagi dapat menggunakan perintah :

```
DROP DATABASE nama_table;
```

2.10.3 DML (Data Manipulation Language)

Data manipulation language merupakan bahasa yang digunakan untuk melakukan manipulasi data dalam database (tambah, ubah, hapus, cari). Perintah-perintah yang termasuk dalam DML adalah :

a. INSERT

Perintah insert digunakan untuk memasukan data ke dalam tabel. Perintah insert dapat dituliskan sebagai berikut :

```
INSERT INTO nama_table values (nilai1,nilai2,...)
```

b. UPDATE

Perintah update digunakan untuk memperbaiki data dalam suatu record (baris) dalam suatu tabel. Perbaikan dapat dilakukan untuk satu record, beberapa atau seluruh record. Perintah update dapat langsung dituliskan sebagai berikut :

```
UPDATE nama_tabel SET field=nilai1[,field2=nilai2[,...]]
[where kondisi];
```

c. DELETE

Perintah delete digunakan untuk melakukan penghapusan record dari suatu table yang memiliki kondisi yang dinyatakan dalam pernyataan kondisi.

Perintah delete dapat di tulis sebagai berikut :

```
DELETE from nama-table
[WHERE kondisi];
```

d. SELECT

Perintah select digunakan untuk menampilkan isi dari suatu table. Perintah select ditulis sebagai berikut :

```
SELECT {*field1[,field2[,...]]}from nama_tabel
[WHERE kondisi];
```

2.11 Konsep Dasar Pemodelan Arsitektur Sistem

Pemodelan dalam suatu rekayasa perangkat lunak merupakan suatu hal yang dilakukan di tahap awal. Di dalam suatu rekayasa perangkat pemodelan sebaiknya di lakukan di bagian awal dari rekayasa, dan pemodelan ini memperbaharui pekerjaan-pekerjaan dalam rekayasa perangkat lunak tersebut.

2.11.1 Model Pendekatan Waterfall

Model ini telah diperoleh dari proses engineering lainnya. Model ini menawarkan cara pembuatan perangkat lunak secara lebih nyata. Langkah-langkah yang penting dalam model ini adalah :

a. Penentuan dan Analisis Spesifikasi

Jasa, kendala dan tujuan dihasilkan dari konsultasi dengan penggunaan sistem. Kemudian semuanya itu dibuat dalam bentuk yang dapat dimengerti oleh pengguna dan staf pengembang.

b. Desain Sistem dan Perangkat Lunak

Proses desain sistem membagi kebutuhan-kebutuhan menjadi sistem perangkat lunak atau perangkat keras. Proses tersebut menghasilkan sebuah arsitektur sistem keseluruhan. Desain perangkat lunak tersebut menghasilkan fungsi sistem perangkat lunak dalam bentuk yang mungkin ditransformasikan ke dalam satu atau lebih program yang dapat dijalankan.

c. Integrasi dan Pengujian Sistem

Unit program di integrasikan dan diuji menjadi sistem yang lengkap untuk menyakinkan bahwa persyaratan perangkat lunak telah dipenuhi. Setelah di uji coba sistem disampaikan kepada pengguna.

d. Operasi dan Pemeliharaan

Pemeliharaan termasuk pembetulan kesalahan yang tidak dapat ditentukan pada langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru ditentukan.

