BAB II

LANDASAN TEORI

2.1 Pengertian Program Aplikasi

Dalam menentukan standar aplikasi perlu diketahui pengertian dasar dari program aplikasi.

2.1.1 Program

Program adalah ekspresi, pernyataan atau kombinasi yang disusun dan dirangkai menjadi satu kesatuan prosedur yang berupa urutan langkah untuk menyelesaikan masalah yang diimplemtasikan dengan menggunakan bahasa pemprograman, sehingga dapat dieksekusi oleh komputer.

2.1.2 Aplikasi

Aplikasi merupakan penerapan, penyimpan suatu data dan penyelesaian masalah kedalam suatu sarana atau media yang digunakan untuk menerapkan atau mengimplementasikan permasalahan tersebut sehingga berubah menjadi suatu bentuk yang baru tanpa menghilangkan nilai-nilai dasar dari suatu data. Permasalahan atau pekerjaan dalam hal ini hanya bentuk dari tampilan data yang berubah, sedangkan isi yang termuat dalam data tersebut tidak mengalami perubahan.

Maka program aplikasi adalah sederetan kode yang digunakan untuk mengatur komputer supaya dapat melakukan pekerjaan sesuai dengan keinginan *programer* atau *user*.

2.2 Pengertian Angka kredit

Angka kredit, adalah satuan nilai dari tiap butir kegiatan dan atau akumulasi nilai butir-butir kegiatan yang diberikan/ditetapkan berdasarkan penilaian atas prestasi yang telah dicapai oleh seorang guru dan yang dipergunakan sebagai salah satu syarat dalam rangka pembinaan karier dalam jabatan fungsional/kepangkatan guru.

2.3 Unsur Kegiatan yang di Nilai

Unsur kegiatan yang dinilai dalam memberikan angka kredit terdiri atas:

a. Unsur utama

b. Unsur penunjang.

a. Unsur utama terdiri atas:

a. Pendidikan, meliputi:

1. Mengikuti pendidikan sekolah dan memperoleh gelar/sebutan
2. Mengikuti pendidikan sekolah dan memperoleh gelar/sebutan tambahan yang setingkat atau lebih tinggi di luar bidang ilmunya.
3. Memberikan pendidikan dan pelatihan fungsional Guru dan memperoleh Surat Tanda Tamat Pendidikan dan Pelatihan (STTPP) termasuk yang berbentuk kegiatan magang Guru yunior.

b. Tridarma Perguruan Tinggi, meliputi :

1. Melaksanakan pendidikan dan pengajaran, meliputi :

f. membina kegiatan siswa di bidang akademik dan kesiswaan;

g. mengembangkan program sekolah;

h. mengembangkan bahan pengajaran;

i. menyampaikan orasi ilmiah;

j. membimbing Guru yang lebih rendah jabatannya; dan atau

k. melaksanakan kegiatan *detasering* dan pencangkokan Guru

2. Melaksanakan penelitian dan pengembangan serta menghasilkan karya ilmiah, karya teknologi, karya seni monumental/seni pertunjukan, dan karya sastra meliputi:

a menghasilkan karya penelitian, karya ilmiah;

b menterjemahkan/menyadur buku ilmiah;

c mengedit/menyunting karya ilmiah;

d membuat rancangan dan karya teknologi; dan atau

e membuat rancangan dan karya seni.

Karya ilmiah yang tidak atau kurang berbobot tidak begitu saja dapat dialihkan ke butir Pendidikan dan Pengajaran atau Pengabdian kepada Masyarakat. Pengalihan harus mempertimbangkan substansi dan manfaatnya.

3. melaksanakan Pengabdian kepada Masyarakat, meliputi:

a. menduduki jabatan pimpinan dalam lembaga pemerintahan/pejabat negara sehingga harus dibebaskan dari jabatan organiknya;

b. melaksanakan pengembangan hasil pendidikan dan penelitian yang dapat dimanfaatkan oleh masyarakat;

c. memberi latihan/penyuluhan/penataran/pada masyarakat;

d. memberi pelayanan kepada masyarakat atau kegiatan lain yang menunjang pelaksanaan tugas umum pemerintahan dan pembangunan; dan atau

e. membuat/menulis karya pengabdian kepada masyarakat.

b. Unsur penunjang adalah kegiatan yang mendukung pelaksanaan tugas pokok Guru, meliputi:

a. menjadi anggota dalam suatu panitia/badan pada perguruan tinggi;

b. menjadi anggota panitia/badan pada lembaga pemerintah;

c. menjadi anggota organisasi profesi;

d. mewakili perguruan tinggi/lembaga pemerintah duduk dalam panitia antar lembaga.

e. menjadi anggota delegasi nasional ke pertemuan internasional;

f. berperan serta aktif dalam pertemuan ilmiah;

g. mendapat tanda jasa/penghargaan;

h. menulis buku pelajaran SLTA ke bawah; dan atau

i. mempunyai prestasi di bidang olahraga/kesenian/sosial.

2.4.1 Penilaian perhitungan angka kredit

(1) Tim Penilai Angka kredit Pusat bertugas membantu Sekretaris Jenderal Departemen Pendidikan Nasional dalam melakukan penilaian terhadap usul Penetapan angka kredit guru serta melaksanakan tugas-tugas lain yang diberikan oleh Menteri Pendidikan Nasional yang berhubungan dengan penetapan angka kredit guru.

(2) Untuk melaksanakan tugas sebagaimana dimaksud pada ayat (1) Tim Penilai Angka kredit Pusat mempunyai fungsi sebagai berikut:

a. meneliti persyaratan dan bukti-bukti yang dipersyaratkan bagi setiap usul penetapan angka kredit yang diajukan;

b. melakukan penilaian terhadap angka-angka kredit yang diajukan pada setiap usul penetapan angka kredit guru yang menjadi wewenangnya;

c. menyampaikan hasil penilaiannya kepada Menteri Pendidikan Nasional selaku yang berwenang menetapkan angka kredit; dan

d. melaksanakan tugas- tugas lain yang berhubungan dengan Penetapan Angka Kredit

(3) Tim Penilai Angka Kredit Pusat dibentuk dengan Keputusan Sekretaris Jenderal Departemen Pendidikan Nasional.

(4) Dalam melaksanakan tugasnya, Tim Penilai Angka Kredit Pusat bertanggung jawab kepada Sekretaris Jenderal Departemen Pendidikan Nasional.

Dinas Pendidikan Kecamatan Subang



Jalan Darmodiharjo No.5 Kelurahan Sukamelang,
Kecamatan Subang, Kabupaten Subang Telp 0260-413450

Visi :

- **Kecamatan Subang unggul dalam mutu, optimal dalam pelayanan dan responsif terhadap inovasi**

Misi :

- **Menuntaskan wajar Diknas 9 tahun**
- **Peningkatan mutu pendidikan**
- **Peningkatan efisiensi pendidikan**
- **Peningkatan relevansi pendidikan**
- **Pemasyarakatan budaya gemar baca**

SASARAN

1. Menuntaskan wajar Diknas 9 tahun

1.1 Tercapainya angka parisipasi sebesar 100% lebih

1.2 Tercapainya angka partisipasi murni sebesar 100%

2. Meningkatkan mutu pendidikan

2.1 Tercapainya mutu prestasi akademik rata-rata 7.0

2.2 Tercapainya peningkatan mutu non akademik/imtaq

2.3 Terselenggaranya pengelolaan pendidikan yang demokratis, pasilitatif, dan inovatif

3. Meningkatkan efisiensi pendidikan

3.1 Tertekannya angka DO/mengulang test

3.2 Terpenuhinya penambatan/pemerataan guru, termasuk guru kontrak/GBS

3.3 Terenovasinya sekolah-sekolah yang rusak berat dan ringan oleh pemerintah dan atau melaui partisipasi masyarakat

3.4 Terealisasinya dana perawatan sekolah melalui programACIS_PAMIARSA (aku cinta indahnya sekolah dengan cara perpatungan miara sakola)

3.5 Terpenuhinya kebutuhan buku pelajaran (ratio 1:1)

3.6 Terwujudnya program akselerasi

4. Meningkatkan relevansi pendidikan

4.1 Tersusunnya materi muatan lokal yang dapat menyentuh kebutuhan pembangunan daerah/masyarakat sekitar sekolah

4.2 Terciptanya managemen partisipatif dari dewan sekolah melalui managemen berbasis sekolah(MBS)

4.3 Terimplementasinya konsep sekolah masyarakat (society school)

5. Memasyarakatkan budaya gemar baca

5.1 Tercapainya peningkatan minat baca siswa sehingga membaca siswa sehingga

membaca merupakan suatu kebutuhan untuk belajar, bukan belajar untuk membaca

5.2 Terselenggaranya program gerakan budaya gemar pustaka (gebyar pustaka)

5.3 Terimplementasinya konsep perpustakaan kelas

2.3 Pengembangan Sistem

Alat pengembangan sistem adalah unsur yang sangat penting dalam melakukan langkah–langkah metodologi pengembangan sistem yang terstruktur. Alat–alat yang digunakan umumnya berupa gambar, diagram atau grafik serta kamus data .

2.5.1 Metode RekayasaPerangkat Lunak

Ada beberapa metode yang dapat digunakan dalam pengembangan sebuah perangkat lunak, salah satu diantaranya yaitu metode waterfall. Pengembangan sistem dengan menggunakan Metode *waterfall* ini terbagi menjadi beberapa tahap diantaranya tahap rekayasa sistem, analisis, perancangan, pengkodean, pengujian dan pemeliharaan, tahapan-tahapan tersebut saling terkait dan saling mempengaruhi.

a. Rekayasa Sistem

Tahap awal yang dilakukan adalah rekayasa sistem yaitu perumusan sistem yang akan kita buat. Tujuan dilakukan tahap ini adalah agar pengembang benar-benar mengerti tentang sistem yang akan dibuat dan langkah-langkah

serta kebijaksanaan apa saja yang berkaitan dengan pengembangan sistem tersebut.

b. Analisis

Tujuan dilakukan tahapan ini adalah untuk memahami sistem yang ada pada saat ini agar dapat mendefinisikan permasalahan sistem sehingga selanjutnya dapat menentukan kebutuhan sistem secara garis besar sebagai persiapan ke tahap perancangan

c. Perancangan

Pada tahap perancangan diberikan gambaran umum yang jelas kepada pengguna dan rancang bangun yang lengkap tentang sistem yang akan dikembangkan kepada pihak-pihak yang terlibat dalam pengembangan sistem. Tahap perancangan ini dilakukan sebagai persiapan untuk tahap implementasi.

d. Implementasi/Coding/Pemrograman

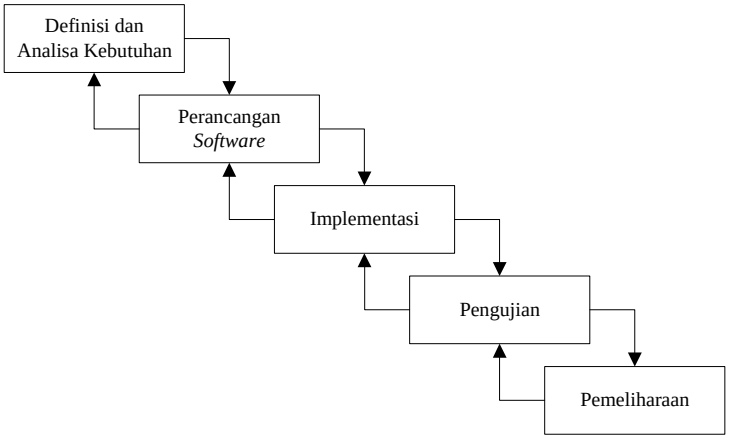
Setelah tahap perancangan sistem, selanjutnya dilakukan konversi rancangan sistem ke dalam kode-kode bahasa pemrograman yang diinginkan. Pada tahap ini dilakukan pembuatan komponen-komponen sistem yang meliputi modul program, antarmuka dan basis data.

e. Pengujian

Tahap pengujian ini dilakukan untuk mendapatkan serta memastikan bahwa perangkat lunak yang dihasilkan adalah valid dan sesuai dengan kebutuhan yang telah dideskripsikan.

f. Pemeliharaan

Pada tahap pemeliharaan ini perangkat lunak sudah diserahkan kepada pengguna. Pada tahap ini dilakukan evaluasi terhadap sistem yang baru untuk mengetahui apakah sistem telah memenuhi tujuan yang ingin dicapai. Dari hasil evaluasi ini dimungkinkan untuk melakukan perubahan-perubahan yang diperlukan terhadap sistem agar sistem senantiasa dapat digunakan dengan baik.



Gambar 2.8 Metode Waterfall

2.5.2 Diagram Konteks

Diagram konteks adalah arus data yang berfungsi untuk menggambarkan hubungan aliran-aliran data antara sistem dengan bagian-bagian luar yang berkaitan dengan sistem tersebut sebagai suatu proses dengan data masukan dan keluaran digambarkan sebagai panah yang masuk dan keluar. Diagram konteks biasa disebut dengan model sistem pokok (*fundamental system model*) mewakili keseluruhan elemen *software* dengan masukan (*input*) dan keluaran (*output*) yang diindikasi dengan anak panah masuk dan keluar memperlihatkan suatu hubungan antara sistem dengan lingkungannya yang menjadi sumber data.

2.5.3 Data Flow Diagram

Untuk menganalisa sistem yang dibuat dilakukan pemodelan dengan menggunakan *Data Flow Diagram* (DFD). DFD adalah suatu teknik grafis yang menggambarkan aliran informasi dan perubahan yang digunakan sebagai perpindahan data dari masukan (*input*) ke keluaran (*output*) .

DFD dapat dibagi menjadi beberapa level yang menggambarkan penambahan aliran informasi dan fungsionalitas yang lebih rinci. DFD level 0 disebut dengan diagram konteks. Selanjutnya pada level yang lebih tinggi, proses tersebut dipecah-pecah untuk memperoleh aliran data dan proses yang lebih rinci.

2.5.4 Flow Map

Flow Map merupakan data berbentuk dokumen di dalam suatu sistem informasi yang merupakan suatu aktifitas yang saling terkait dalam hubungannya dengan kebutuhan data dan informasi dengan cara mendifinisikan hubungan antara pelaku proses, proses, dan aliran data.

2.4 Konsep Dasar Basis Data

Basis data terdiri atas dua kata yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data adalah representasi dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

Basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti:

a. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan yang tidak perlu, untuk memenuhi berbagai kebutuhan.
c. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

2.6.1 *Entity Relation Diagram*

Entity Relationship Diagram (ERD) merupakan suatu diagram yang digunakan untuk menghubungkan antar elemen (*relational Condition*), dimana pada tahap selanjutnya dapat diimplementasikan ke dalam bentuk tabel relasi.

Beberapa macam hubungan antar relasi, antara lain :

1. Satu Ke Satu (*One to One*)

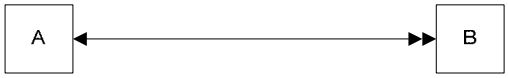
Bentuk relasi antara satu entitas dengan jumlah satu ke entitas dengan jumlah yang sama.



Gambar 2.9 Hubungan relasi *One to One*

2. Satu Ke Banyak (*One to Many*)

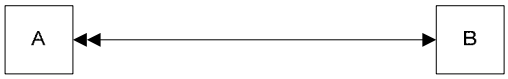
Bentuk relasi dari entitas dengan jumlah satu ke entitas lain yang berjumlah lebih dari satu (Entitas dengan banyak alternatif tujuan).



Gambar 2.10 Hubungan relasi *One to Many*

3. Banyak ke Satu (*Many to One*)

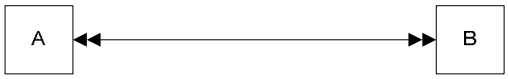
Bentuk relasi yang mendefinisikan hubungan antara entitas yang berjumlah lebih dari satu dengan entitas yang berjumlah satu.



Gambar 2.11 Hubungan relasi *Many to One*

4. Banyak ke Banyak (*Many to Many*)

Bentuk relasi yang mendeskripsikan permasalahan yang komplek yaitu hubungan antara entitas yang berjumlah lebih dari satu dengan entitas dengan jumlah yang sama.



Gambar 2.12 Hubungan relasi *Many to Many*

2.6.2 Normalisasi

Normalisasi adalah proses memecah suatu file database yang mengandung permasalahan. Tujuan dari normalisasi adalah untuk mengurangi penyimpangan informasi dalam proses :

1. Penyimpangan penyisipan, yaitu tidak ada tempat untuk menyisipkan informasi dalam suatu entiti sampai dibangun suatu relasi dengan entiti yang lainnya.
2. Penyimpangan penghapusan, yaitu penghapusan suatu record akan menghilangkan semua informasi dari himpunan data dalam satu entiti.

3. Penyimpangan perubahan, yaitu perubahan terhadap suatu nilai atribut yang harus dilakukan di beberapa tempat.

Pada proses normalisasi selalu diuji beberapa kondisi, relasi tersebut dipecahkan dalam beberapa tabel, yaitu :

1. Normal Kesatu (1NF/*First Normal Form*)

   Suatu relasi dikatakan dalam bentuk normal pertama jika dan hanya jika memenuhi syarat bahwa relasi tersebut memenuhi nilai-nilai *atomik* (tidak ada yang berulang-ulang atau bernilai ganda)

2. Normal Kedua (2NF/*Second Normal Form*)

   Suatu relasi berada dalam bentuk normal kedua jika dan hanya jika berada pada bentuk normal pertama dan semua atribut bukan kunci memiliki dependensi sepenuhnya terhadap kunci primer.

3. Normal Ketiga (3NF/*Third Normal Form*)

   Relasi harus dalam bentuk normal kedua dari semua atribut tabel, kunci harus tergantung hanya pada *primary key.*

4. Bentuk Normal *Boyce-Codd* (BCNF)

Definisi bentuk *Boyce-Codd* adalah suatu relasi disebut memenuhi bentuk normal *Boyce-Codd* jika dan hanya jika semua penentu (*determinan*) adalah kunci kandidat (atribut yang bersifat unik). BCNF merupakan bentuk normal sebagai perbaikan terhadap 3NF, tetapi tidak untuk sebaliknya karena bentuk normal ketigapun mungkin masih mengandung *anomali* sehingga masih perlu dinormalisasi lebih lanjut.

2.6.3 Kamus Data

Kamus data adalah daftar teroganisir dari semua elemen data yang ada pada suatu sistem dengan definisi yang jelas/tepat, sehingga user dan analisis sistem mendapat kesepahaman dari input, output dan komponen dari penyimpanan dan kalkulasi "*intermediate*" yang ada.

2.6.4 Operasi Dasar Basis Data

Pengoperasian *database* meliputi:

1. Pembuatan database baru (*create database*) yaitu menciptakan database baru.
2. Penghapusan database (*drop database*) yaitu penghapusan suatu database.
3. Pembuatan file/tabel baru ke dalam suatu database yaitu penambahan file/tabel ke dalam database.
4. Penambahan/pengisian data baru ke sebuah file/tabel di sebuah database (*insert*)
5. Pengambilan data dari sebuah file/tabel (*retrieve/search*) yaitu pencarian data pada suatu file.
6. Pengubahan data dari sebuah file/tabel (*update*) yaitu perbaikan data pada suatu file.
7. Penghapusan data dari sebuah file/tabel (*delete*).

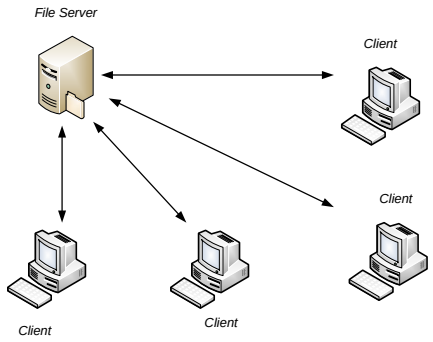
2.5 Konsep Dasar Client/Server

Peranan utama suatu program *server* adalah melayani *client* yang berjumlah banyak yang memiliki tujuan untuk menggunakan secara bersama sumber daya yang dimiliki oleh *server* tersebut [8]. Berikut ini adalah karasteristik suatu yang biasanya dimiliki oleh suatu program server.

1. Menanti permintaan *client.* Program server menghabiskan sebagian besar waktu kerjanya secara pasif menanti permintaan *client*. Biasanya permintaan ini datang dalam bentuk *message* melalui sessi komunikasi. Beberapa *server* menggunakan suatu sesi khusus untuk setiap client. *Server* yang lainnya menggunakan *session* yang digunakan secara dinamis. Ada juga yang menggunakan gabungan kedua teknik ini (*dedicated* dan dinamis). Untuk dapat bekerja dengan baik, *server* harus tetap dapat bekerja ketika terjadi permintaan yang banyak *(rush hour traffic)*.
2. Melaksanakan banyak permintaan pelayanan pada saat yang bersamaan. *Server* harus sesegera mungkin melaksanakan pelayanan yang diminta oleh *client*. Jelas ini berarti, bahwa *client* tak boleh bergantung pada proses server yang hanya memiliki thread tunggal. *Server* harus dapat secara konkuren menyediakan pelayanan dengan tetap menjaga integritas sumber dayanya.
3. Mendahulukan client yang memiliki prioritas lebih tinggi (VIP). *Server* harus menyediakan beberapa tingkatan prioritas untuk clientnya. Misal untuk suatu pekerjaan batch dilakukan pada tingkatan prioritas yang rendah, sedangkan untuk pekerjaan yang berkaitan dengan *On Line Transaction Processing (OLTP)* dilakukan dengan prioritas tinggi.

4. Memulai dan melaksanakan aktifitas pekerjaan di *background*. *Server* harus dapat menjalankan program di *background*, misal melakukan *download record* dari *database* utama selama waktu tidak sibuk. Inisiatif ini harus dapat dilakukan secara otomatis oleh *server*.
5. Tetap menjaga agar sistem tetap selalu bekerja. Program *server* biasanya tergolong *mission-critical application.* Akan terjadi kerugian bila *server* tak bekerja melayani *client.* Dengan demikian program server dan environmentnya harus dapat bekerja secara *robust* (tahan terhadap gangguan).
6. Bertambah besar. Biasanya program *server* membutuhkan memori dan prosesor yang besar. *Environment* dari *server* haruslah dapat di *upgrade* dan memiliki skalabilitas yang baik.

Model hubungan *Client/Server* memungkinkan jaringan untuk mensentralisasi fungsi dan aplikasi kepada satu atau dua *dedicated file server*. Sebuah *file server* menjadi jantung dari keseluruhan sistem, memungkinkan untuk mengakses sumber daya dan menyediakan keamanan. *Workstation* yang berdiri sendiri dapat mengambil sumber daya yang ada pada *file server*. Model hubungan ini, menyediakan mekanisme untuk mengintegrasikan seluruh komponen yang ada di jaringan dan memungkinkan banyak penggunaan secara bersama-sama memakai sumber daya pada *file server*.



Gambar 2.13 Model Hubungan *Client/server*

Kelebihan model hubungan *Client/server*:

1. Terpusat (sumber daya dan keamanan data dikontrol melalui server)
2. Skalabilitas
3. Fleksibel
4. Teknologi baru dengan mudah terintegrasi ke dalam system
5. Keseluruhan komponen (*Client, network, server*) dapat bekerja bersama.

Kekurangan model hubungan *client/server*:

1. Mahal
2. Membutuhkan investasi untuk dedicated *file server*
3. Perbaikan (Jaringan besar membutuhkan seorang staff untuk mengatur agar sistem berjalan secara efisien)
4. Berketergantugan.
5. Ketika *server* jatuh mengakibatkan keseluruhan operasi pada *network* akan jatuh pula.

2.6 Software Pendukung

Software pendukung digunakan sebagai fasilitator dalam memenuhi kriteria dalam membuat suatu aplikasi yang sempurna

Software pendukung ini disesuaikan dengan kemampuan seorang programmer dalam memadukan antara bahasa pemograman dengan database yang digunakan

2.8.1 Visual Basic 6.0

Visual Basic 6.0 adalah bahasa pemrograman berbasis Windows. Sebagai bahasa pemrograman yang mutakhir, visual basic didesain untuk dapat memanfaatkan fasilitas yang tersedia dalam Microsoft Windows [6].

Visual Basic juga merupakan bahasa pemrograman *Object Oriented Programming* (OOP), yaitu pemrograman yang berorientasi objek. Visual Basic 6.0 menyediakan objek-objek yang sangat kuat, berguna dan mudah dipakai.

Visual Basic 6.0 adalah salah satu bahasa pemrograman yang telah menerapkan konsep RAD (*Rapid Application Development*) atau pengembangan aplikasi secara cepat, berbasis visual dan familiar bagi pengguna. Sebagai software DBMS, kita dapat merancang, membuat dan membangun program aplikasi database maupun program-program lainnya [6].

2.8.2 MySQL

MySQL merupakan salah satu dari sekian banyak Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*GNU General Public Licence*) [3]. Sehingga setiap orang mudah untuk

mendapatkan dan bebas untuk menggunakan MySQL, dengan batasan tidak boleh dijadikan sebuah produk turunan yang bersifat *closed source* atau dikomersialkan. Meski sebelumnya dalam masalah lisensi ini, MySQL pernah mengalami perubahan, namun hingga saat ini lisensi tersebut sudah erat pada MySQL.

MySQL merupakan turunan dari salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL sendiri adalah sebuah konsep pengoperasian *database,* terutama untuk pemilihan/seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Hal ini akan memudahkan pengguna *database* ketika ingin menggunakan kembali *database* yang ada.

MySQL merupakan terobosan solusi yang tepat dalam aplikasi database. Sifatnya yang *open source* serta dukungan oleh ribuan bahkan jutaan komunitas pengguna di internet, menjadikan MySQL sebagai software database yang cukup banyak digunakan. Tak hanya itu, kemampuannya yang bisa digunakan pada berbagai sistem operasi juga menjadikan MySQL sebagai *software database* pilihan.

Pada dasarnya, keandalan suatu sistem *database* (DBMS) dapat diketahui dengan melihat cara kerja optimizer-nya dalam melakukan proses-proses perintah SQL. Baik itu yang dibuat oleh user maupun program-program aplikasi lainnya.

MySQL merupakan database server yang dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam query data.

Kelebihan dari MySql adalah sebagai berikut :

1. *Portability*

MySql dapat digunakan dengan stabil tanpa kendala, berarti pada berbagai sistem operasi diantaranya seperti Windows, Linux, FreeBSD, OS X Server, Solaris, Amiga, HP-UX dan masih banyak lagi.

2. *Open Source*

   Sebagaimana telah diuraikan sebelumnya, MySql didistribusikan secara open source(gratis), di bawah lisensi GPL, sehingga kita dapat memperoleh dan menggunakannya secara cuma-cuma tanpa dipungut biaya sepeser pun.

3. *Multiuser*

   MySql dapat digunakan untuk menangani beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik. Hal ini akan memungkinkan sebuah *database server MySql* dapat diakses *client* secara bersamaan dalam waktu yang bersamaan pula.

4. *Perfomance Tuning*

   Dalam hal ini perfomansi, MySql memiliki kecepatan yang cukup menakjubkan dalam menangani query-query sederhana, serta mampu memproses lebih banyak SQL persatuan waktu.

5. *Column Type*

   MySql mendukung Type kolom (tipe kolom) yang sangat komplek, seperti *signed/unsigned integer, float, double, char, varchar, text, blob, date, time, datetime, timestamp, year, set serta enum*. Bukan tidak mungkin tipe-tipe data tersebut akan terus dikembangkan oleh pengembang MySql.

6. *Command dan Functions*

MySql memiliki operator dan fungsi secara penuh yang mendukung perintah *select* dan *where* dalam query.

7. *Security*

   Sekuritas sistem merupakan hal yang sangat diperhatikan pada MySql. Terbukti adanya beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta *password* yang terenkripsi.

8. *Scability dan Limits*

   Dalam hal batas kemampuan, MySql terbukti mampu menangani *database* dalam skala cukup besar, dengan jumlah records lebih dari 50 juta 60 ribu tabel serta 5 miliar baris. Selain itu, batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. *Connectivity*

   MySql mampu melakukankoneksi dengan *client* menggunakan protocol TCP/IP, Unix soket (Unix), atau Named Pipes(NT).

10. *Localisation*

    Localisation merupakan salah satu kemampuan MySql dalam mendeteksi pesan kesalahan (*error code*) pada *client* dengan menggunakan lebih dari dua puluh bahasa. Dengan demikian, pengguna cukup menyesuaikan bahasa yang digunakan.

11. *Interface*

Sama halnya dengan *software-software database* lainnya, MySql memiliki *interface*(antarmuka) terhadap berbagai aplikasi dan bahasa pemprograman dengan menggunakan fungsi API (*Aplication Programing Interface*).

12. *Client dan Tools*

MySql dilengkapi dengan berbagai *tool* yang dapat digunakan untuk administrasi *database*, dan pada setiap *tool* yang ada disertakan petunjuk *online*. Hal ini akan semakin memudahkan dalm penggunaan MySql.

13. *Struktur Tabel*

Struktur tabel MySQL cukup baik, serta cukup fleksibel, misalnya ketika menangani antar tabel dibandingkan *database* lainnya semacam PostgreSql ataupun Oracle.

2.8.3 MyODBC

ODBC adalah sebuah standar yang dikembangkan microsoft untuk mendukung driver database yang bersifat *indefendent* terhadap produk[9], dengan adanya *ODBC* kita dapat menggunakan database yang tidak didukung oleh *BDE* (*Borland Database Engine*). *Driver ODBC* dapat diperoleh dari produsen *database* yang kita pakai. *Driver ODBC* ada yang 16 bit dan ada yang 32 bit. *Delphi* memakai *driver* yang 32 bit. *Microsoft* memproduksi *driver ODBC* untuk *Access, Excell, SQL Server, MySql*, dan lain lain.