BAB 2

TINJAUAN PUSTAKA

2.1 Profil Instansi

2.1.1 Sejarah Instansi

Pada awal berdirinya ditahun 1989 sekolah ini hanya bersifat temporer dengan status terbuka dan belum terakr*edit*asi, beroperasi di atas bangunan yang hanya seluas 5,6 X 6 M^2 dan berjumlah murid tidak lebih dari 15 orang.

Seiring berjalannya waktu, jumlah murid dari sekolah ini pun makin bertambah, dan akhirnya pengelola pun merasa membutuhkan gedung yang lebih luas dan layak untuk tempat belajar para muridnya. Dengan bantuan dari warga komplek perumahan sekitar, maka dibangunlah sebuah gedung tambahan yang dibangun di dekat area sekolah tersebut seluas 14 X 28 M^2.

Karena prestasi dan kualitasnya yang sudah cukup baik, maka pada tahun 1992 sekolah ini diresmikan oleh Departemen Agama dengan nama RA. Nurul Iman. Dengan jumlah murid pada saat itu adalah 50 orang yang terdiri dari 3 kelas TK utama.

Berikut adalah para pendiri sekaligus pembina sekolah RA. Nurul Iman :

1. H. Lukman Hakim → Penasehat
2. H. Thabrani → Ketua Yayasan
3. H. Mardiono, S.H. → Ketua II
4. H. A. Suyuti → Sekretaris II
5. H. Djamsari Kamal → Sekretaris I
6. H. Bahrudin Akbar → Bendahara I

7. T. Abu Bakar Sabi → Bendahara II
8. Ibu Sulastri → Kepala Sekolah mencakup guru (sejak periode awal sekolah)
9. Ibu Yuli → Staff Guru (periode awal)
10. Ibu Tutu → Staff Guru (periode awal)

Hingga saat ini luas bangunan RA. Nurul Iman adalah 485 M^2, dengan jumlah kelas 3 Lokal TK, 4 Lokal Madrasah Diniyah, dan 2 Lokal TPA. Sekolah ini beralamatkan di Komplek Green Garden Blok A kelurahan Rorotan Kecamatan Cilincing Jakarta Utara, dengan status gedung "Milik Sendiri", nomor statistik 101231720036 dan izin pendirian WJ/6B/PP004/3987/92.

Di sekolah ini juga menyediakan beberapa ekstrakurikuler bagi para muridnya, yaitu menari, renang, dan *drumband*, yang masing-masing dari kegiatan ekskul tersebut telah meraih prestasi yang cukup baik hingga saat ini.

2.1.2 Deskripsi Pekerjaan Instansi

Deskripsi pekerjaan (*Job description*) adalah uraian dari setiap tugas atau tanggung jawab yang diemban oleh setiap anggota dari suatu organisasi, agar fungsi dari masing-masing anggota menjadi jelas ruang lingkupnya.

Berikut adalah *jobdesc* yang ada di RA. Nurul Iman – Jakarta :

1. Kepala Sekolah :

a. Monitoring kegiatan belajar mengajar (KBM)
b. Pengambilan keputusan
c. Menghadiri segala kegiatan yang berhubungan dengan prosedur KBM-nya.

2. Guru :

a. Mengajar

b. Membuat laporan perencanaan pembelajaran

c. Monitoring siswa didiknya

d. Membuat laporan akhir siswa

3. Tata Usaha :

a. Pembukaan surat masuk dan surat keluar

b. Mengurus seluruh proses administrasi sekolah, seperti ; PSB, iuran bulanan, dan kegiatan-kegiatan lainnya yang membutuhkan biaya.

2.2 Landasan Teori

2.2.1 Pengertian Sekolah Taman Kanak-kanak

Taman Kanak-Kanak (TK) termasuk dalam jenjang Pendidikan Anak Usia Dini (PAUD), yakni usia 6 tahun atau di bawahnya yang dikemas dalam bentuk pendidikan formal. Program pembelajaran TK lebih ditekankan kepada pemberian rangsangan pendidikan guna membantu pertumbuhan dan perkembangan jasmani dan rohani agar anak memiliki kesiapan dalam memasuki pendidikan lebih lanjut.

Masa belajar seorang murid di TIK tergantung pada tingkat kecerdasannya yang dinilai dari rapor per semesternya. Untuk lulus dari tingkat program di TK selama dua (2) tahun, yakni :

1. TK Nol Kecil selama 1 tahun
2. TK Nol Besar selama 1 tahun

Usia rata-rata minimal anak untuk dapat memulai pendidikan di TK berkisar 4-5 tahun, sedangkan usia rata-rata untuk lulus dari TK berkisar 6-7 tahun. Setelah lulus dari TK, murid akan melanjutkan program pendidikan ke tingkat Sekolah Dasar atau yang biasa disingkat SD.

2.2.1.1 Pengertian *Raudhatul Athfal*

Raudatul Athfal (disingkat RA) merupakan jenjang pendidikan anak usia dini (yakni usia 6 tahun atau di bawahnya) dalam bentuk pendidikan formal, di bawah pengelolaan Kementerian Agama

RA setara dengan taman kanak-kanak (TK), di mana kurikulumnya ditekankan pada pemberian rangsangan pendidikan untuk membantu pertumbuhan dan perkembangan jasmani dan rohani agar anak memiliki kesiapan dalam memasuki pendidikan lebih lanjut.

2.2.1.2 Teknik Pembelajaran Siswa Sekolah Taman Kanak-kanak

Teknik pembelajaran di TK yaitu dengan memberi kesempatan anak-anak belajar sesuai dengan usia tiap tingkatannya, antara lain :

1. Bernyanyi
2. Membaca
3. Berhitung
4. Budi bahasa
5. Agama
6. Dan berbagai macam keterampilan lainnya.

Tujuan dari pembelajaran tersebut diatas adalah untuk menciptakan daya cipta kanak-kanak serta memacunya untuk belajar mengenai berbagai ilmu pengetahuan yang dirancang sebagai upaya untuk menumbuhkembangkan daya pikir dan peranan anak kecil, yang dikemas dalam bentuk belajar sambil bermain.

2.2.2 Pengertian Sistem Informasi

Menurut Jogiyanto dalam bukunya yang berjudul "Analisis dan Disain Sistem Informasi", pengertian sistem informasi itu sendiri adalah;

> *"Suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasasi, dan menyediakan pihak luar tertentu dengan laporan - laporan yang diperlukan".*

Sistem informasi dapat dikatakan sebagai suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam suatu sistem organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi. Kegiatan sistem informasi mencakup:

1. *Input*, menggambarkan kegiatan untuk menyediakan data untuk diproses.
2. *Proses*, menggambarkan bagaimana suatu data diproses untuk menghasilkan suatu informasi yang bernilai tambah.
3. *Output*, suatu kegiatan untuk menghasilkan laporan proses diatas tesebut.
4. Penyimpanan, suatu kegiatan untuk memelihara dan menyajikan data.
5. Kontrol, suatu aktifitas untuk menjamin bahwa sistem informasi tersebut sesuai yang diharapkan.

2.2.2.1 Komponen Sistem Informasi

Sistem informasi dapat terdiri dari komponen-komponen, yaitu:

1. Blok Masukan

   Input mewakili data yang masuk ke dalam sistem informasi. *Input* disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukan, yang dapat berupa dokumen-dokumen dasar.

2. Blok Model

   Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data *input* dan data yang tersimpan di dasar data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan

3. Blok Keluaran

   Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi

   Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, mengasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem keseluruhan.

5. Blok Basis Data

   Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanpulasinya. Data perlu disimpan di dalam dasar data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa, supaya informasi yang dihasilkan berkualitas.

6. Blok Kendali

   Untuk supaya sistem informasi dapat berjalan sesuai dengan yang diingikan, maka perlu diterapkan pengendalian-pengendalian. Beberpa pengendali perlu dirancang dan diterapkan untuk menyakinkan bahwa hal-hal yang dapat

merusak sistem dapat dicegah atapun bila terlanjur kesalahan-kesalahan dapat langsung cepat diatasi.

2.2.3 Sistem Pendukung Keputusan

SPK sebagai sebuah sistem berbasis komputer yang membantu dalam proses pengambilan keputusan. SPK sebagai sistem informasi berbasis komputer yang adaptif, interaktif, fleksibel, yang secara khusus dikembangkan untuk mendukung solusi dari pemasalahan manajemen yang tidak terstruktur untuk meningkatkan kualitas pengambilan keputusan. Dengan demikian dapat ditarik satu definisi tentang SPK yaitu sebuah sistem berbasis komputer yang adaptif, fleksibel, dan interaktif yang digunakan untuk memecahkan masalah-masalah tidak terstruktur sehingga meningkatkan nilai keputusan yang diambil.

Menurut Moore and Chang, SPK dapat digambarkan sebagai sistem yang berkemampuan mendukung analisis *ad hoc* data, dan pemodelan keputusan, berorientasi keputusan, orientasi perencanaan masa depan, dan digunakan pada saat-saat yang tidak biasa.

Tahapan SPK:

1. Definisi masalah
2. Pengumpulan data atau elemen informasi yang relevan
3. pengolahan data menjadi informasi baik dalam bentuk laporan grafik maupun tulisan
4. menentukan alternatif-alternatif solusi (bisa dalam persentase)

Tujuan dari SPK:

1. Membantu menyelesaikan masalah semi-terstruktur

2. Mendukung manajer dalam mengambil keputusan
3. Meningkatkan efektifitas bukan efisiensi pengambilan keputusan

Dalam pemrosesannya, SPK dapat menggunakan bantuan dari sistem lain seperti *Artificial Intelligence*, *Expert Systems*, *Fuzzy Logic*, dll.

2.2.4 Metode Pembangunan Perangkat Lunak

Dalam proses pembangunan aplikasi pada tugas akhir kali ini, metode penelitian dan pengembangan yang digunakan adalah:

2.2.4.1 Metode *Waterfall*

Metode ini biasa juga disebut dengan *classic life cycle*. Metode ini membutuhkan pendekatan sistematis dan sekuensial dalam pengembangan perangkat lunak, dimulai dari tingkat sistem dan kemajuan melalui analisis, desain, pengkodean, pengetesan dan pemeliharaan. Pemodelan ini menyangkut aktivitas berikut:

Berikut ini beberapa kelebihan yang dimiliki metode *Waterfall*, diantaranya adalah:

1. Metode ini masih lebih baik digunakan walaupun sudah tergolong “kuno”, daripada menggunakan pendekatan asal-asalan.
2. Selain itu, metode ini juga masih masuk akal jika kebutuhan sudah diketahui dengan baik.

Berikut ini beberapa kekurangan yang dimiliki metode *Waterfall*, diantaranya adalah:

1. Pada kenyataannya, jarang mengikuti urutan sekuensial seperti pada teori. Iterasi sering terjadi menyebabkan masalah baru.

2. Sulit bagi pelanggan untuk menentukan semua kebutuhan secara eksplisit.
3. Pelanggan harus sabar, karena pembuatan perangkat lunak akan dimulai ketika tahap desain sudah selesai. Sedangkan pada tahap sebelum desain bisa memakan waktu yang lama.
4. Kesalahan di awal tahap berakibat sangat fatal pada tahap berikutnya.

2.2.4.2 Perancangan Sistem

Proses perancangan diperlukan untuk menghasilkan suatu rancangan sistem yang baik, karena dengan adanya rancangan yang tepat akan menghasilkan sistem yang stabil dan mudah dikembangkan di masa mendatang. Perancangan yang kurang baik akan mengakibatkan sistem yang dibangun harus dirombak total atau sistem yang dibangun akan sangat berlebihan dari kebutuhan yang diperlukan. Tahap perancangan disebut juga tahap pemecahan masalah, yaitu dengan menyusun suatu alur sistem, masukan, prosedur proses, keluaran dan *database*.

A. Bagan Alir Dokumen (*Flowmap*)

Digunakan untuk mendefinisikan hubungan antara bagian (pelaku proses), proses (manual atau berbasis komputer) dan aliran data dalam bentuk dokumen keluaran dan masukan. Adapun simbol-simbol yang digunakan dalam *flowmap*, dapat dilihat dalam daftar simbol.

B. Bagan Alir (*Flowchart*)

Flowchart adalah alat untuk menganalisa proses yang merupakan gambaran dalam bentuk diagram alir dari algoritma-algoritma dalam suatu program, yang menyatakan arah alur program tersebut. Hal tersebut

memungkinkan untuk memecah proses menjadi kejadian-kejadian individual atau aktifitas untuk menunjukan secara singkat hubungan diantaranya. Konstruksi *flowchart* memungkinkan pengertian lebih baik kepada proses dan pengertian yang lebih baik terhadap proses akan membawa kepada perbaikan pengembangan suatu sistem. Adapun simbol-simbol yang digunakan dalam *flowchart*, dapat dilihat dalam daftar simbol.

C. Bagan Alir Data Sistem Terstruktur

Analisis sistem terstruktur yaitu pendekatan proses analisis terhadap fakta atau data yang diperoleh dari proses sebelumnya dengan menggunakan prosedur dan peralatan analisis sistem terstruktur. Alat-alat analisis terstruktur, yaitu:

1. *Data Flow Diagram* (DFD)

DFD adalah diagram yang digunakan untuk menggambarkan arus data dalam sistem informasi dengan menggunakan simbol-simbol atau notasi-notasi tertentu.

Dalam mengembangkan suatu aliran data atau proses yang terjadi di dalam sistem *data flow diagram* menggunakan simbol-simbol yang memiliki arti tersendiri dalam menerangkan:

a. *Eksternal Entity*

Eksternal entity dapat merupakan kesatuan (*entity*) dilingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya, yang memberikan *input-output* dari sistem.

b. *Data Flow*

Arus data ini mengatur diantara proses, simpan data, dan kesatuan luar. Arus data ini menujukkan arus data yang dapat berupa masukan sistem atau hasil proses sistem.

c. Proses

Untuk *physical data flow diagram* (PDFD), data dilakukan oleh orang, mesin atau komputer. Sedangkan untuk *logical data flow diagram* (LDFD), suatu proses hanya menujukkan proses dari komputer.

d. Penyimpanan Data

Simpanan data (*data store*) merupakan tempat penyimpanan data. Simpanan data dari DFD disimbolkan dengan sepasang garis horizontal paralel.

Konsep dasar DFD dapat dilakukan dengan analisa *Top Down*, yaitu pemecahan sistem yang besar menjadi beberapa sub-sub sistem yang lebih kecil DFD terdiri dari:

a. *Context Diagram*

Diagram konteks yaitu diagram yang menunjukkan batas dan jangkauan dari sistem informasi yang dibuat. Merupakan gambaran sistem secara garis besar dan hanya memperlihatkan kelompok data *input* dan *output*. Konteks diagram merupakan *level* teratas dari diagram arus data.

b. *Middle Level*

Merupakan pemecahan dari tiap–tiap proses yang mempunyai fungsi sama. Pada *middle level* diagaram 0 dipecah menjadi diagram 1,2,3 dan seterusnya yang merupakan penguraian dari diagram konteks.

c. *Lowest Level* (DFD Level Terendah),

Diagram yang menunjukkan proses yang lebih detail dari level sebelumnya.Merupakan pemecahan dari data *flow* yang ada pada *middle level*.Pemecahan tersebut masih tetap mempunyai fungsi yang sama dari *level* sebelumnya. Untuk *Lowest Level*, pemberian nomor diagram terdiri dari bagian *middle level.*

2. *Data Dictionary* (Kamus Data)

Kamus data adalah pusat penyimpanan atau dokumentasi manual dari semua informasi yang terdapat pada sistem informasi atau program aplikasi yang digunakan sebagai sumber informasi pada proses *systems design*. Kamus data dapat terdiri dari:

1. KD elemen data/*Field*
2. KD alur data
3. KD penyimpanan data/tabel
4. KD proses pengolahan data
5. KD *record*
6. KD laporan

2.2.5 Konsep Basis Data

Basis data (*database*) merupakan kumpulan data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. *Database* merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi bagi para pemakai. Sistem basis data adalah suatu sistem

informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lainnya. Beberapa istilah dalam *database*, yaitu:

1. *Entity*
2. *Atribut*
3. *Data Value*
4. *Record*

2.2.5.1 Pengertian Basis Data

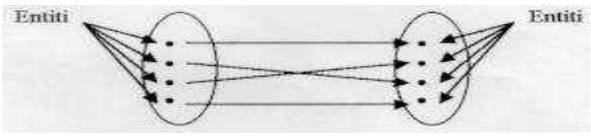
Basis data adalah kumpulan *file-file* yang mempunyai kaitan antara satu *file* lain dengan *file* lain sehingga membentuk suatu bangunan data untuk menginformasikan suatu perusahaan/instansi dalam batasan tertentu basis data merupakan salah satu komponen penting dalam sistem informasi karena basis data adalah dasar untuk menyediakan informasi bagi para pemakai.

2.2.5.2 ERD (*Entity Relationship Diagram*)

Entity Relationship Diagram yang berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan attribute-attribut yang merepresentasikan seluruh fakta dari dunia nyata yang kita tinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan Diagram *Entity Relationship* (Diagram E-R). Antar entitas selalu ada tiga hubungan biner, yaitu:

1. Satu ke Satu (*One To One Relationship*)

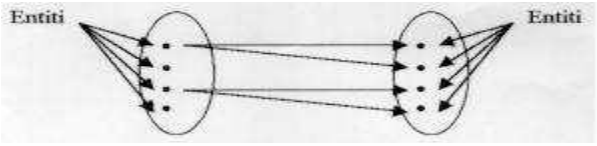
Hubungan antara *entity* pertama dengan *entity* kedua adalah satu berbanding satu.



Gambar 2.1 Relasi satu ke satu

2. Satu ke Banyak (*One To Many*)

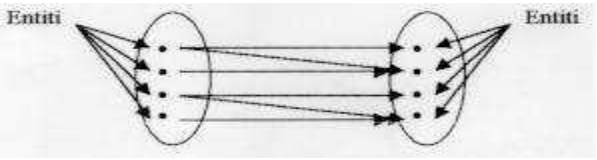
Himpunan *entity* pertama dapat berhubungan satu atau lebih *entity* pada himpunan kedua, tetapi tidak sebaliknya.



Gambar 2.2 Relasi satu ke banyak

3. Banyak ke Banyak (*Many To Many*)

Setiap *entity* pada himpunan *entity* pertama dapat berhubungan dengan banyak *entity* pada himpunan *entity* kedua, begitu pula sebaliknya.



Gambar 2.3 Relasi banyak ke banyak

2.2.5.3 Relasi Tabel

Di dalam sebuah *database*, setiap tabel memiliki sebuah field yang memiliki nilai unik untuk setiap baris. Field ini ditandai dengan icon bergambar kunci di depan namanya. Baris-baris yang berhubungan pada tabel mengulangi kunci primer (primary key) dari baris yang dihubungkannya pada tabel lain.

Salinan dari kunci primer di dalam tabel-tabel yang lain disebut dengan kunci asing (foreign key). Kunci asing ini tidak perlu bersifat unik, dan semua field bisa menjadi kunci asing. Yang membuat sebuah field merupakan kunci asing adalah jika dia sesuai dengan kunci primer pada tabel lain.

2.2.6 *Borland Delphi*

Borland Delphi adalah sebuah alat pengembang aplikasi-aplikasi untuk sistem operasi *Microsoft Windows*. *Delphi* sangat berguna dan mudah digunakan

untuk membuat suatu program berbasis GUI (*Graphical user interface*) atau *console* (mode teks).

Borland Delphi mempunyai "saudara" bernama *Borland Kylix* yaitu versi *Delphi* yang digunakan untuk membuat aplikasi pada sistem operasi *Linux*. Dengan dipasangkannya *Borland Delphi* dengan *Borland Kylix* maka pengembang *software* dapat membuat aplikasi berbasis *Windows* yang dapat dengan mudah dikompilasi ulang pada *Linux*.

Delphi merupakan bahasa pemrograman pertama yang memecahkan batasan antara bahasa tingkat tinggi, pengembangan aplikasi dengan cepat (*Rapid Application Development*/RAD). Ketika membuat aplikasi GUI dengan *Delphi*, pengembang perangkat lunak akan mendapatkan bahasa pemrograman (dalam hal ini *Object Pascal*) yang dibungkus dalam lingkungan RAD. Semua *user interface* seperti *form*, tombol (*button*), dan *object list* telah disertakan dalam *Delphi* dalam bentuk komponen atau *control*. Pengembang dapat dengan mudah menempatkan komponen-komponen tersebut ke dalam *form*. Pengembang dapat juga menempatkan *control ActiveX* pada form untuk membuat program-program khusus seperti *Browser Web* dalam waktu yang cepat. *Delphi* memungkinkan pengembang untuk merancang keseluruhan *interface* secara *visual*, dan dengan cepat dapat diimplementasikan sebuah kode perintah berbasis *event* (*event driven*) dengan mengklik *mouse*.

Dengan *Delphi*, pengembang perangkat lunak dapat membuat program *Windows* dengan lebih cepat dan lebih mudah dari sebelumnya.

2.2.6.1 Tipe Aplikasi *Database* Delphi

Aplikasi *database* di delphi mempunyai banyak tipe, yaitu:

1. Sistem Tunggal atau *Stand-Alone*
2. Sistem *File-Share* atau Sentralisasi
3. Sistem *Client-Server*
4. Sistem *Multi-Tier*

Penjelasan diatas adalah sebagai berikut;

a. Sistem Tunggal atau *Stand-Alone*

Pada arsitektur ini *database* dan aplikasi *database* ditempatkan pada mesin (komputer) yang sama. Arsitektur ini adalah arsitektur yang paling sederhana dan dirancang untuk *single user*. *Database* yang digunakan adalah *database* lokal.

b. Sistem *File-Share*

Arsitektur ini sama dengan aplikasi *database* stand-alone kecuali *database* dapat diakses dari beberapa *client* yang terhubung dalam jaringan. Jadi aplikasi ditempatkan ditempatkan di masing-masing *client* untuk untuk mengakses *database* yang sama yang diletakan di suatu komputer dalam jaringan.

Aplikasi *database* stand-alone atau *file-share* termasuk dalam aplikasi *single-tiered application*, karena aplikasi dan *database* berbagai *file system* yang sama.

c. Sistem *Client-Server*

Arsitektur ini merupkan dari du komponen utama yaitu *client* dan *server*. Apliksi *database* berada pada mesin *client*. Sedangkan pada *server* terdapat

remote *database* sever dan *database* yang akan diakses. Aplikasi jenis ini juga sering diebut sebagai *two-tiered application*. Jika aplikasi juga ditempatkan di mesin yang sama dengan *server* yang berisi *remote database server*, aplikasi ini juga tetap disebut *two-tiered application*, karena aplikasi dan *database server* beroperasi pada dua *sistem* independen yang berbeda.

d. Sistem *Multi-Tier*

Arsitektur ini merupakan perluasan dari sitem *client-server*. Arsitektur ini terdiri dari remote *database server*, aplikasi *server* yang mengakses remote *server* dan aplikasi *client* yang mengakses aplikasi *server*. Jumlah tier dalam aplikasi dapat ditambah sesuai kebutuhan.

2.2.6.2 Kelebihan Menggunakan *Borland Delphi*

Kelebihan-kelebihan yang dapat diambil ketika seorang pengembang perangkat lunak menggunakan *Borland Delphi* adalah:

1. *Delphi* mendukung Pemrograman Berorientasi Objek (*Object Oriented Programming*/OOP).
2. Pengembangan aplikasi secara cepat (*Rapid Application Development*/RAD)
3. Menggunakan bahasa tingkat tinggi yang mudah dipahami.
4. Hasil dari proses kompilasi berupa sebuah *file* yang dapat dieksekusi (*executable file*) sehingga mempermudah dalam pendistribusian program dan mengurangi banyaknya *file* pendukung DLL.
5. *Delphi* menyediakan banyak sekali komponen yang dapat digunakan. Selain itu banyak juga komponen yang bersumber dari pihak ketiga yang biasanya

disertai dengan dokumentasi, *source code* dan lain-lain. Komponen dari pihak ketiga bisa yang komersil atau *free*.

6. Mendukung banyak *database server* (*MySQL, SQL Server, Interbase, Oracle* dll) sehingga dapat mempermudah dalam membuat aplikasi *database*.

2.2.7 RDBMS MySQL

Kebanyakan dari *database* tergantung pada *Database Management System* (DBMS) untuk mengelola data yang tersimpan dalam sistem *database* dan menyiapkan data agar tersedia bagi pengguna yang ingin mengakses informasi tertentu. Sebuah DBMS terdiri atas satu perangkat *server* dan *client* yang komprehensif (meliputi banyak hal) yang mendukung berbagai macam tugas-tugas administratif dan yang berhubungan dengan data. Beberapa perangkat DBMS menyediakan beberapa tipe perangkat *client*, yang mengijinkan anda untuk berinteraksi secara langsung dengan data yang tersimpan dalam *database*.

Minimal sekali, sebuah DBMS harus dapat menyimpan data dan mengijinkan data tersebut dapat diambil kembali dan dimodifikasi, sekaligus melindungi data terhadap suatu operasi yang dapat merusak atau menyebabkan ketidakkonsistenan (inkonsistensi) data. Bagaimanapun, kebanyakan sistem menyediakan lebih banyak kemampuan. Secara umum, beberapa DBMS saat ini mendukung tipe-tipe fungsional berikut:

a. *Managing storage*
b. *Maintaining security*
c. *Maintaining metadata*
d. *Managing transactions*

e. *Supporting connectivity*

f. *Optimizing performance*

g. *Providing back-up and recovery mechanisms*

h. *Processing requests for data retrieval and modification.*

2.2.7.1 Jenis-jenis RDBMS

RDBMS (*Relational Database Management System*). Beberapa RDBMS: MySQL, *Oracle*, DB2, SQL *Server*, dan PostgreSQL.

Produk-produk ini sebagaimana perangkat DBMS, mengijinkan Anda untuk mengakses dan mengelola kebutuhan metadata untuk menentukan data yang disimpan. Perbedaan utama anta DBMS dan RDBMS adalah bahwa RDBMS lebih spesifik ke *database* relasional. Tidak hanya mendukung penyimpanan data dalam struktur seperti tabel, namun juga sebuah *relationship* (keterhubungan) di antara tabel-tabel tersebut.

2.2.7.2 Fitur-fitur MySQL

MySQL termasuk salah satu "pemain" besar dalam pasar RDBMS, hal itu dikarenakan oleh lengkapnya fitur-fitur yang dimiliki oleh MySQL yang dapat memenuhi kesluruhan fungsi-fungsi bagi sebuah *database.*

Berikut adalah beberapa fitur-fitur yang membuat MySQL unggul beseta penjelasannya;

1. *Scalability*

   MySQL saat ini dapat menangani *database* yang cukup besar. Beberapa organisasi atau perusahaan yang telah menerapkannya antara lain: Yahoo!, Cox Communications, Google, Cisco, Texas Instruments, UPS, Sabre Holdings, HP, The Associated Press, dan lain-lain. Bahkan NASA dan Biro

Sensus US telah mengimplementasikan MySQL *Solutions*. Menurut dokumentasi produk MySQL, beberapa *database* yang digunakan oleh MySQL AB, perusahaan yang membuat MySQL, berisi 50 juta rekod, dan beberapa pengguna MySQL melaporkan bahwa *database* mereka berisi 60.000 tabel dan 5 milyar baris.

2. *Portability*

   Beberapa sistem operasi yang dapat menjalankan MySQL: Unix, Linux, Windows, OS/2, Solaris, dan MacOS. MySQL juga dapat berjalan pada arsitektur yang berbeda-beda, mulai dari PC pada level bawah sampai level tinggi seperti mainframe.

3. *Connectivity*

   MySQL mendukung jaringan secara penuh dan *socket* TCP/IP, *socket* Unix, dan *named pipes* (penamaan). Di lain hal, MysQL dapat diakses dari manapun pada *internet*, dan *multiple* (banyak) pengguna dapat mengakses *database* MySQL secara *simultan* (bersamaan). MySQL juga menyediakan API (*Application Programming Interface*) untuk mendukung konektivitas dengan beberapa aplikasi yang ditulis dengan menggunakan C, C++, Pearl, PHP, Java, dan Python.

4. *Security*

   MySQL meliputi sistem yang handal untuk mengontrol akses ke data. Sistem menggunakan sebuah *host* dan struktur berbasis *client* yang mengontrol siapa saja yang dapat mengakses informasi tertentu dan tingkatan akses ke informasi tersebut. MySQL juga mendukung protokol SSL (*Secure Sockets Layer*) untuk membuat suatu koneksi yang terenkripsi.

5. *Speed*

   Jumlah waktu yang diperlukan sebuah *database* MySQL untuk merespon *request data* sama cepatnya bahkan lebih cepat ketimbang RDBMS komersil lainnya. *Website* MySQL (www.mysql.com) menyediakan hasil-hasil tes *benchmark* yang menunjukkan hasil kecepatan dalam penerapan MySQL.

6. *Ease of Use*

   MySQL mudah untuk di-instal dan diterapkan. Pengguna dapat mendapatkannya dan menjalankannya dalam beberapa menit setelah men-downloadnya. Pada tingkat administratif, MySQL secara relatif mudah dioptimalkan, terutama jika dibandingkan dengan produk RDBMS lainnya.

7. *Open Source Code*

   MYSQL AB menyediakan *source code* MYSQL bagi siapa saja untuk men-*download* dan menggunakannya. Filosofi dari *open source* adalah mengijinkan *audiens* umum untuk berpartisipasi.

2.2.8 Perancangan Aplikasi (*Design Application*)

Perancangan aplikasi dimaksudkan untuk memudahkan pemrogram komputer dalam melaksanakan tugasnya. Pada langkah ini terdapat empat bagian, yaitu perancangan *input/output*, pengkodean, struktur menu dan kebutuhan sistem.

2.2.8.1 Perancangan *Input - Output*

a. Perancangan Masukan (*Input*)

Perancangan *input* merupakan awal dimulainya proses informasi. Pada perancangan *input* ini dilakukan tahapan perancangan *interface* dari program, yaitu bagimana merancang bentuk dari program sistem informasi yang mudah

untuk diakses, cepat dan memperhitungkan efisiensi waktu dan hardware komputer pengguna.

b. Perancangan Keluaran (*Output*)

Keluaran (*output*) merupakan produk dari sistem informasi yang dapat dilihat. dilakukan perancangan *output* sistem, bagaimana mendesain sebuah *output* yang didapat dari beberapa proses peng*edit*an data yang ada. *Output* ini berupa tampilan dalam bentuk *paper*/*printer output*. Contoh *output* adalah faktur, check, tanda terima pembayaran dan sebagainya.

2.2.8.2 Struktur Menu

Menu banyak digunakan dalam dialog karena merupakan jalur pemakai (*user interface*) yang mudah dipahami dan mudah digunakan. Menu berisi dengan beberapa alternatif atau pilihan yang disajikan kepada *user*. *User* dapat memilih pilihan di menu dengan cara menekan tombol angka atau huruf yang dihubungkan dengan pilihan tersebut. Jika pilihan dari menu terlalu banyak, maka dapat diorganisasikan secara berjenjang.

2.2.8.3 Kebutuhan Sistem

Kebutuhan-kebutuhan sistem (*system requirements*) yang harus di perhatikan dalam merancang sistem informasi antara lain :

1. Kebutuhan Perangkat Lunak (*Software Requirements*)

   Sistem harus disesuaikan dengan ketersediaan perangkat lunak yang digunakan oleh perusahaan dan dapat mengikuti perubahan (*upgrade*) perangkat lunak yang dilakukan oleh perusahaan.

2. Kebutuhan Perangkat Keras (*Hardware Requirements*)

Sistem harus disesuaikan dengan ketersediaan perangkat keras yang digunakan oleh perusahaan dan dapat mengikuti perubahan (*upgrade*) perangkat keras yang dilakukan oleh perusahaan.