

BAB II

LANDASAN TEORI

2.1 Pengertian Parkir

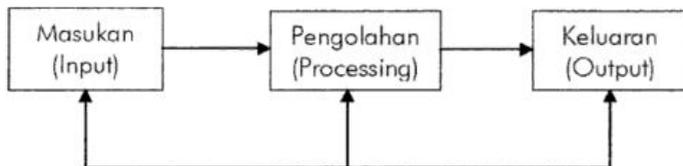
Parkir adalah keadaan tidak bergerak suatu kendaraan yang bersifat sementara karena ditinggalkan oleh pengemudinya. Fasilitas parkir dibangun bersama-sama dengan kebanyakan gedung, untuk memfasilitasi kendaraan pemakai gedung. Termasuk dalam pengertian parkir adalah setiap kendaraan yang berhenti pada tempat-tempat tertentu baik yang dinyatakan dengan rambu lalu lintas ataupun tidak, serta tidak semata-mata untuk kepentingan menaikkan dan/atau menurunkan orang dan/atau barang (<https://id.wikipedia.org/wiki/Parkir>).

2.2 Pengertian Sistem

Asal kata Sistem berasal dari bahasa Latin *systema* dan bahasa Yunani *sustema*. Pengertian sistem adalah suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan.

Secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung sama lain. *Murdick dan Ross (1993)* mendefinisikan sistem sebagai seperangkat elemen yang digunakan satu dengan lainnya untuk suatu tujuan bersama. Sementara, definisi sistem dalam kamus *Webster's Unbringed* adalah elemen-elemen yang saling berhubungan dan membentuk satu kesatuan atau organisasi.

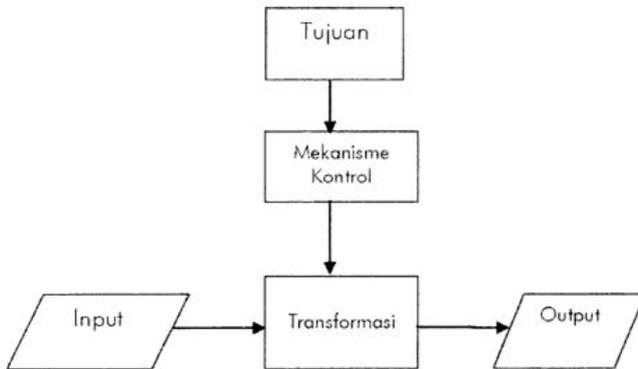
Menurut *Scott (1996)*, sistem terdiri dari unsur-unsur seperti masukan (*input*), pengolahan (*processing*), serta keluaran (*output*). Ciri pokok sistem menurut Gaspert ada empat, yaitu sistem itu beroperasi dalam suatu lingkungan terdiri atas unsur-unsur, ditandai dengan saling berhubungan, dan mempunyai satu fungsi atau tujuan utama.



Gambar 2.1. Model Sistem

Gambar di atas menunjukkan bahwa sistem atau pendekatan sistem minimal harus mempunyai empat komponen, yakni masukan, pengolahan, keluaran, dan balikan atau control.

Sementara *Mc. Leod (1995)* mendefinisikan sistem sebagai sekelompok elemen-elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. Sumber daya mengalir dari elemen *output* dan untuk menjamin prosesnya berjalan dengan baik maka dihubungkan dengan mekanisme control. Untuk lebih jelasnya elemen sistem tersebut dapat digambarkan dengan model sebagai berikut:



Gambar 2.2. Model Hubungan Elemen-Elemen Sistem

Banyak ahli mengajukan konsep sistem dengan deskripsi yang berbeda, tetapi pada prinsipnya hampir sama dengan konsep dasar sistem umumnya. *Schröderberg (1971)* dalam *Suradinata (1996)* secara ringkas menjelaskan bahwa sistem adalah:

1. Komponen-komponen yang saling berhubungan satu sama lain.
2. Suatu keseluruhan tanpa memisahkan komponen pembentuknya.
3. Bersama-sama dalam mencapai tujuan.
4. Memiliki input dan output yang dibutuhkan oleh sistem lainnya.
5. Terdapat proses yang mengubah input dan output.
6. Menunjukkan adanya entropi.
7. Memiliki aturan.
8. Memiliki subsistem yang lebih kecil.
9. Memiliki deferensiasi antar subsistem.
10. Memiliki tujuan yang sama meskipun mulainya berbeda.

2.2.1. Karakteristik Sistem

Untuk memahami atau mengembangkan suatu sistem, maka perlu membedakan unsur-unsur dari sistem yang membentuknya. Berikut adalah karakteristik sistem yang dapat membedakan suatu sistem dengan sistem lainnya:

1. Batasan (*boundary*), penggambaran dari suatu elemen atau unsur mana yang termasuk di dalam sistem dan mana yang di luar sistem.
2. Lingkungan (*environment*), segala sesuatu di luar sistem, lingkungan yang menyediakan asumsi, kendala, dan input terhadap suatu sistem.
3. Masukan (*input*), sumber daya (data, bahan baku, peralatan, energy) dari lingkungan yang dikonsumsi dan dimanipulasi oleh suatu sistem.
4. Keluaran (*output*), sumber daya atau produk (informasi, laporan, dokumen, tampilan layar computer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.
5. Komponen (*component*), kegiatan-kegiatan atau proses dalam suatu sistem yang mentransformasikan input menjadi bentuk setengah jadi (output), komponen ini bisa merupakan subsistem dari sebuah sistem.
6. Penghubung (*interface*), tempat dimana komponen atau sistem dan lingkungannya bertemu atau berinteraksi.
7. Penyimpanan (*storage*), area yang dikuasai dan digunakan untuk penyimpanan sementara dan tetap dari informasi, energy, bahan baku, dan sebagainya. Penyimpanan merupakan suatu media penyangga di antara komponen tersebut bekerja dengan berbagai tingkatan yang ada dan memungkinkan komponen yang berbeda dari berbagai data yang sama.

2.3 Pengertian Bahasa Pemrograman

Bahasa pemrograman merupakan kumpulan aturan yang disusun sedemikian rupa sehingga memungkinkan pengguna komputer membuat program yang dapat dijalankan dengan aturan tersebut. Bahasa pemrograman dapat dikelompokkan dalam berbagai macam sudut pandang. Salah satu pengelompokan bahasa pemrograman adalah pendekatan dari notasi bahasa pemrograman tersebut, apakah lebih dekat ke bahasa mesin

atau manusia. Dengan cara ini, bahasa pemrograman dapat dikelompokkan menjadi tiga yakni *high level* (misal: Pascal dan Basic, *middle level* (misal: Bahasa C), dan *low level* (misal: Bahasa Assembly).

2.3.1. Klasifikasi Bahasa Pemrograman

1. Berdasarkan Tingkat

Berdasarkan tingkat kedekatannya dengan bahasa mesin dan bahasa manusia, bahasa pemrograman diklasifikasikan menjadi:

- a. Bahasa tingkat rendah atau disebut juga bahasa assembly, memetakan setiap instruksi dari kode sumber ke dalam satu instruksi bahasa mesin. Untuk dapat memrogram dalam bahasa assembly, diperlukan pengetahuan yang benar tentang perangkat keras, representasi data, cara kerja pemrosesan, dan instruksi-instruksi mesin komputer.
- b. Bahasa tingkat tinggi, memiliki kedekatan dengan bahasa dan cara berpikir manusia. Contoh: Bahasa Pascal.
- c. Bahasa tingkat menengah, antara kedua tingkat di atas. Contoh: Bahasa C.

2. Berdasarkan Paradigma

Berdasarkan paradig, pemrograman computer dapat diklasifikasikan sebagai berikut:

- a. *Pemrograman Prosedural*. Program tersusun atas prosedur-prosedur algoritma. Pemrograman procedural dicirikan dengan adanya *sequence* (urutan), *branch* (percabangan), dan *loop* (pengulangan). Urutan, seperti bahasa tersebut memiliki urutan pemrosesan tertentu; biasanya dari atas ke bawah dan dari kiri ke kanan. Percabangan, merupakan alternative alur program berdasarkan evaluasi terhadap kondisi tertentu.
- b. Pengulangan terhadap alur program berdasarkan evaluasi terhadap kondisi tertentu.
- c. Contoh bahasa pemrograman procedural adalah bahasa pascal, C, dan Basic.
- d. *Pemrograman Fungsional*. Program tersusun atas fungsi-fungsi, baik yang sudah terdefinisi maupun yang dibuat sendiri oleh pengguna. Contoh LISP.

- e. *Pemrograman Deklaratif*. Tersusun atas deklarasi-deklarasi logika berupa fakta, aturan dan sasaran. Bahasa pemrograman deklaratif memiliki mekanisme inferensi, yang memungkinkan computer ‘menalar’ atau menelusuri berbagai fakta dan aturan yang diberikan untuk mendapatkan hasil/ jawaban dari pertanyaan tertentu.
- f. *Pemrograman Berorientasi Objek*. Dalam pemrograman berorientasi objek, persoalan dan solusinya dimodelkan sebagai objek.

Perbedaan dasar bahasa-bahasa pemrograman terletak pada tipe data yang digunakan, jenis operasi yang tersedia, dan mekanisme untuk mengendalikan operasi-operasi yang dikerjakan pada data.

2.3.2. Tipe Data

Tipe data adalah sebuah kelas dari objek-objek data bersama-sama dengan himpunan operasi untuk membuat dan memanipulasinya. Walaupun sebuah program hanya berurusan dengan objek data tertentu seperti array A, variabel integer X datau file F, sebuah bahasa pemrograman perlu biasa berurusan dengan tipe-tipe data seperti kelas array-array, integer-integer, ataupun file-file.

Setiap bahasa pemrograman mempunyai sekumpulan tipe data primitive yang menjadi bawaan bahasa tersebut (*built in*) dan ada juga yang menyediakan fasilitas bagi programmer untuk mendefinisikan tipe data baru. Sebuah tipe data dalam bahasa pemrograman dapat dipelajari dari dua tingkatan yang berbeda, yaitu spesifikasi dan implementasi.

Elemen-elemen dasar spesifikasi suatu tipe data adalah:

1. Atribut-atribut yang membedakan objek-objek data dengan tipe tersebut.
2. Nilai-nilai yang mungkin dimiliki oleh objek data dengan tipe tersebut.
3. Operasi-operasi yang dimungkinkan untuk memanipulasi objek data dengan tipe tersebut.

Sedangkan elemen dasar implementasi sebuah tipe data adalah:

1. Representasi penyimpanan yang digunakan untuk mewakili objek-objek data dengan tipe tersebut pada penyimpanan computer selama program dijalankan.
2. Alat (manner) di mana operasi-operasi yang didefinisikan untuk tipe data itu direpresentasikan dengan algoritma-algoritma dan prosedur-prosedur yang memanipulasi representasi penyimpanan yang dipilih untuk objek data tersebut.

Objek data yang hanya mempunyai sebuah nilai tunggal dan hanya dimanipulasi sebagai sebuah satuan disebut objek data elementer. Setiap bahasa pemrograman biasanya mengenal tipe-tipe data seperti integer, Boolean, char, dan enumerasi (terbilang) sebagai tipe data elementernya.

2.4 Pengenalan dan Pengertian Delphi

Delphi adalah suatu bahasa pemrograman (*development language*) yang digunakan untuk merancang suatu aplikasi program. Delphi termasuk dalam pemrograman bahasa tingkat tinggi (*high level language*). Maksud dari bahasa tingkat tinggi yaitu perintah-perintah programnya menggunakan bahasa yang mudah dipahami oleh manusia. Bahasa pemrograman Delphi disebut bahasa prosedural artinya mengikuti urutan tertentu. Dalam membuat aplikasi perintah-perintah, Delphi menggunakan lingkungan pemrograman visual.

Delphi merupakan generasi penerus dari Turbo Pascal. Pemrograman Delphi dirancang untuk beroperasi dibawah sistem operasi Windows. Program ini mempunyai beberapa keunggulan, yaitu produktivitas, kualitas, pengembangan perangkat lunak, kecepatan kompilasi, pola desain yang menarik serta diperkuat dengan bahasa pemrograman yang terstruktur dalam struktur bahasa pemrograman *Object Pascal*.

Sebagian besar pengembang Delphi menuliskan dan mengkompilasi kode program di dalam lingkungan pengembang aplikasi atau *Integrated Development Environment* (IDE). Lingkungan kerja IDE ini menyediakan sarana yang diperlukan untuk merancang, membangun, mencoba, mencari atau melacak kesalahan, serta mendistribusikan aplikasi. Sarana-sarana inilah yang memungkinkan pembuatan prototipe aplikasi menjadi lebih mudah dan waktu yang diperlukan untuk mengembangkan aplikasi menjadi lebih singkat.

2.4.1. File-file Penyusun Project

Sepintas sebuah program aplikasi yang dapat dibuat dengan menggunakan Delphi hanya terdiri dari file project dan sebuah unit. Namun kenyataannya terdapat beberapa file yang dibentuk pada saat membangun sebuah program aplikasi. Berikut ini merupakan file-file penyusun projek yang terdapat pada program Delphi, yaitu :

1. File Project (.Dpr) dan file Unit (.Pas)

Sebuah program Delphi terbangun dari modul-modul source code yang disebut unit. Delphi menggunakan sebuah file projek (.Dpr) untuk menyimpan program utama. File sumber untuk unit biasanya berisi sebagian besar kode di dalam aplikasi, file ini ditandai dengan ekstensi (.Pas). Setiap aplikasi atau projek terdiri atas file projek tunggal atau lebih dalam file unit.

2. File Form (.Dfm)

File *form* adalah file biner yang dibuat oleh Delphi untuk menyimpan informasi yang berkaitan dengan *form*.

3. File Resource (.Res)

File *resource* merupakan file biner yang berisi sebuah ikon yang digunakan oleh *project*. File ini secara terus menerus di-update atau diubah oleh Delphi sehingga file ini tidak bisa diubah oleh pemakai. Dengan menambahkan file *resource* pada aplikasi dan menghubungkan dengan file project dapat menggunakan *editor resource*, misalnya editor untuk membuat file *resource*.

4. File Project Options (.Dof) dan File Desktop Settings (.Dsk)

File *project options* merupakan file yang berisi *options-options* dari suatu *project* yang dinyatakan melalui perintah *Options* dari menu *Project*. Sedang file *desktop setting* berisi *option-option* yang dinyatakan melalui perintah *Environment Options* dari menu *Tools*. Perbedaan di antara kedua jenis file tersebut adalah bahwa file *project options* dimiliki oleh setiap *project* sedangkan file *desktop setting* dipakai untuk lingkungan Delphi.

Apabila ada kerusakan pada kedua jenis file tersebut dapat mengganggu proses kompilasi. Prosedur yang dapat kita tempuh untuk menangani gangguan tersebut adalah dengan menghapus kedua jenis file tersebut yaitu **. Do f** dan **. Dsk** karena kedua file tersebut akan terbentuk secara otomatis pada saat menyimpan *project*.

5. **File Backup (.~d p, .~d f, .~p a)**

File-file dengan ekstensi di atas merupakan file *backup* dari suatu *project*, *form* dan unit. Ketiga jenis file tersebut akan terbentuk pada saat proses penyimpanan untuk yang kedua kalinya. Kerena ketiga file tersebut berjenis *backup* (cadangan) maka ketiga jenis file tersebut berisi salinan terakhir dari file-file utama sebelum disimpan lebih lanjut.

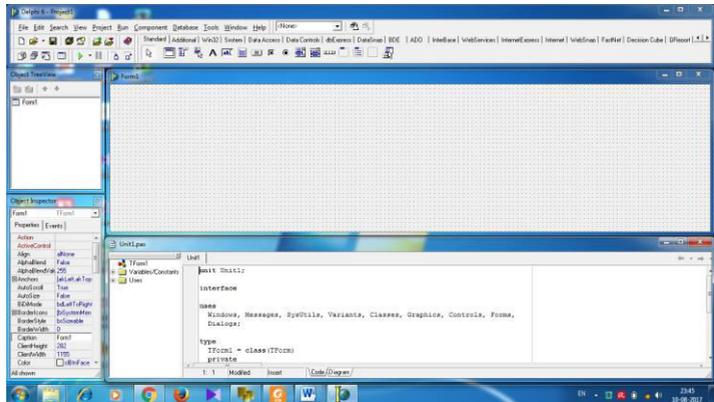
6. **File jenis lain**

File-file dengan ekstensi lain yang dapat ditemukan dalam folder tempat penyimpanan program aplikasi selain yang memiliki ekstensi yang telah disebutkan pada umumnya adalah file-file yang dibentuk oleh *compiler* dan beberapa file Windows yang digunakan Delphi. File-file tersebut adalah:

- a. File *Executable* (**. Exe**). File ini dibentuk oleh compiler dan merupakan file eksekusi (*executable*) dari program aplikasi. File ini berdiri sendiri dan hanya memerlukan file library di **D LL**, **VB X** dan lain-lain.
- b. File unit *Object* (**. Dcu**). File ini merupakan file unit (**. Pa s**) yang telah dikompilasi oleh compiler yang akan dihubungkan dengan file eksekusi.
- c. File *Dinamic Link Library* (**.D ll**). File ini dibentuk oleh **compiler** apabila kita merancang **. DLL** sendiri.
- d. File *Help*. File ini merupakan file Windows dan merupakan file help standar yang dapat dipakai diprogram aplikasi Delphi.
- e. File *Image* (**. Wm f**, **. Bmp**, **.Ico**). File-file ini merupakan file Windows dari aplikasi selain Delphi yang dapat digunakan untuk mendukung program aplikasi yang kita rancang tampak lebih menarik.

2.4.2. Tampilan Delphi

Berikut adalah gambar dari keseluruhan tampilan Delphi:



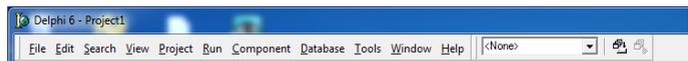
Gambar 2.3. Tampilan Delphi

2.4.3. IDE (*Integrated Development Environment*)

Kemudian, hal yang paling pertama dalam mengenal Delphi adalah harus mengetahui IDE. IDE (*Integrated Development Environment*) merupakan lingkungan/ wilayah dimana seluruh tools atau komponen-komponen yang dibutuhkan untuk merancang atau membangun aplikasi program. Secara umum IDE Delphi di kelompokkan kepada 8 bagian yaitu :

1. Main Menu

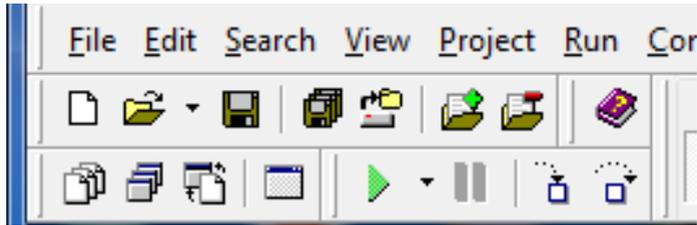
Merupakan penunjuk ke seluruh fasilitas yang disediakan aplikasi Delphi.



Gambar 2.4. Menu Pemrograman Delphi

2. *Toolbar/ Speedbar*

Merupakan Icon (*Sortcut*) yang dirancang untuk lebih memudahkan menjangkau fasilitas yang ada pada Delphi.



Gambar 2.5. Toolbar Pemrograman Delphi

3. Component Palette

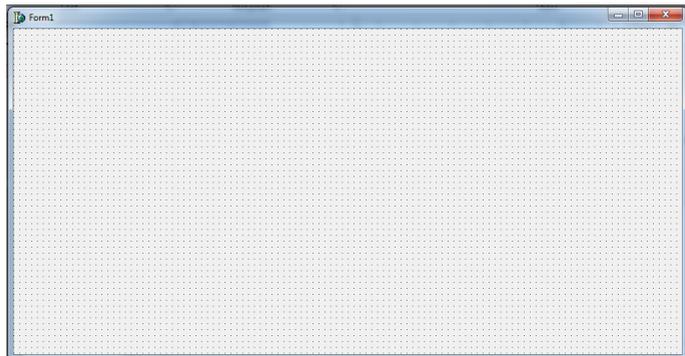
Merupakan komponen-komponen VCL (*Visual Component Library*) yang dikelompokkan kedalam Tab-tab, komponen komponen inilah yang akan digunakan untuk merancang *interface* atau antar muka aplikasi.



Gambar 2.6. Component Palette Pemrograman Delphi

4. Form Designer

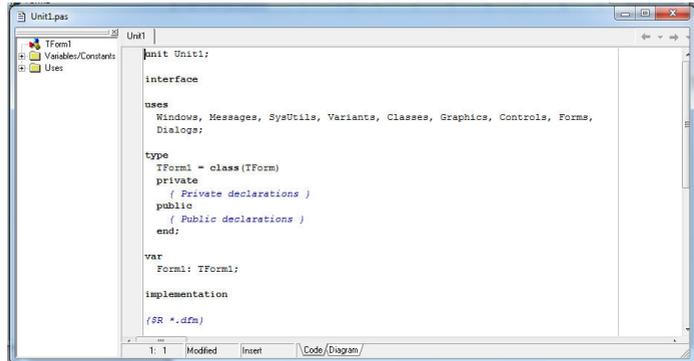
Merupakan *interface* (antar muka) apalikasi yang akan dibangun, Form akan menampung seluruh komponen yang akandigunakan dalam proses perancangan sebuah aplikasi dengan Delphi.



Gambar 2.7. Form Designer Pemrograman Delphi

5. Code editor

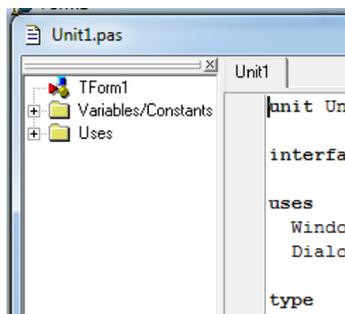
Code editor merupakan tempat untuk menuliskan kode program menggunakan bahasa object Pascal. Kode program tidak perlu di tulis secara keseluruhan karena Delphi sudah menyediakan blok atau kerangka untuk menulis kode program.



Gambar 2.8. Code Editor Pemrograman Delphi

6. Code Explorer

Digunakan untuk memudahkan berpindah antar file unit di dalam jendela *code editor*. *Code explorer* berisi daftar yang menampilkan semua tipe, class, properti, method, variabel global, rutin global yang telah didefinisikan di dalam unit. Saat memilih sebuah item dalam code explorer, kursor akan berpindah menuju implementasi dari item yang dipilih di dalam code editor.



Gambar 2.9. Code Explorer Pemrograman Delphi

7. Object inspector

Object inspector digunakan untuk mengubah properti atau karakteristik dari suatu komponen. Terdiri dari 2 tab yaitu :

a. Properties

Digunakan untuk menentukan seting suatu objek. Satu objek memiliki beberapa properti yang dapat diatur langsung dari object inspector maupun melalui kode program. Seting ini mempengaruhi cara kerja objek tersebut saat aplikasi dijalankan.

b. Event

Merupakan bagian yang dapat diisi dengan kode program tertentu yang berfungsi untuk menangani *event-event* (berupa sebuah *procedure*) yang dapat di respon oleh sebuah komponen.

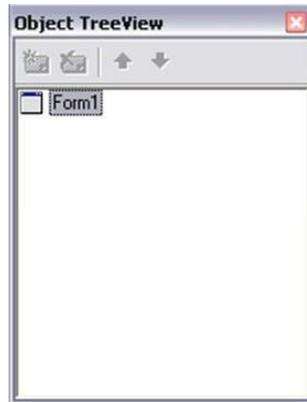
Event adalah peristiwa atau kejadian yang diterima oleh suatu objek, misal: klik, drag, dan lain-lain. *Event* yang diterima objek akan memicu Delphi menjalankan kode program yang ada didalamnya. Misalnya ingin sesuatu dikerjakan pada saat *form* ditutup, maka untuk menyatakan tindakan tersebut (berupa sebuah *procedure*) menggunakan **OnClose**.



Gambar 2.10. Object Inspector Pemrograman Delphi

8. Object Tree View

Object tree view berisi daftar komponen yang sudah diletakkan di form designer.



Gambar 2.11. Object Tree View Pemrograman Delphi

2.4.4. Pengertian Zeos

Delphi menyediakan berbagai macam koneksi database mulai dari BDE (Borland Database Engine), dbExpress, ADO, Interbase dan lain-lain. Koneksi menggunakan BDE sebenarnya sudah lebih dari cukup, BDE bisa mengakses hampir semua tipe database. dbExpress bisa digunakan untuk database MySQL, Interbase dan lain-lain. ADO bisa digunakan untuk mengakses database Microsoft Access, MySQL atau database yang bisa diakses melalui ODBC Windows. Sedangkan Interbase biasa digunakan untuk mengakses database Interbase dan turunannya seperti Firebird.

Zeos adalah salah satu komponen library delphi untuk mengakses database mysql, Zeos tidak hanya untuk MySQL, tapi juga Interbase, Firebird, SQLite dan lainnya.

Zeos adalah komponen alternatif pengganti koneksi ADO yang dapat dengan mudah digunakan untuk mengakses database MySQL, PostGre, Interbase atau Firebird. ZeosDBO versi 6.6.6 stable dapat digunakan di Delphi mulai dari Delphi 5 ke atas, sedangkan ZeosDBO versi 7.0.0 alpha hanya untuk Delphi 2009, 2010 dan 2011.

2.5 Pengertian Database

Basis data (atau database) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut (<http://id.wikipedia.org/wiki/Database>). Database digunakan untuk menyimpan informasi atau data yang terintegrasi dengan baik di dalam komputer.

Untuk mengelola database diperlukan suatu perangkat lunak yang disebut DBMS (Database Management System). DBMS merupakan suatu system perangkat lunak yang memungkinkan user (pengguna) untuk membuat, memelihara, mengontrol, dan mengakses database secara praktis dan efisien. Dengan DBMS, user akan lebih mudah mengontrol dan memanipulasi data yang ada.

Sedangkan RDBMS atau Relationship Database Management System merupakan salah satu jenis DBMS yang mendukung adanya relationship atau hubungan antar tabel. Di samping RDBMS, terdapat jenis DBMS lain, misalnya Hierarchy DBMS, Object Oriented DBMS, dsb. Beberapa software atau perangkat lunak DBMS yang sering digunakan dalam aplikasi program antara lain :

- a. DB2 - <http://www-306.ibm.com/software/data/db2/>
- b. Microsoft SQL Server - <http://www.microsoft.com/sql/>
- c. Oracle - <http://www.oracle.com>
- d. Sybase - <http://www.sybase.com/>
- e. Interbase - <http://www.borland.com/interbase>
- f. Teradata - <http://www.teradata.com/>
- g. Firebird - <http://www.firebirdsql.org/>
- h. MySQL – <http://www.mysql.com>
- i. PostgreSQL - <http://www.postgresql.org/>

2.5.1. Beberapa Istilah Database:

1. Tabel

Sebuah tabel merupakan kumpulan data (nilai) yang diorganisasikan ke dalam baris (*record*) dan kolom (*field*). Masing-masing kolom memiliki nama yang spesifik dan unik.

2. Field

Field merupakan kolom dari sebuah tabel. *Field* memiliki ukuran tipe data tertentu yang menentukan bagaimana data nantinya tersimpan.

3. Record

Field merupakan sebuah kumpulan nilai yang saling terkait.

4. Key

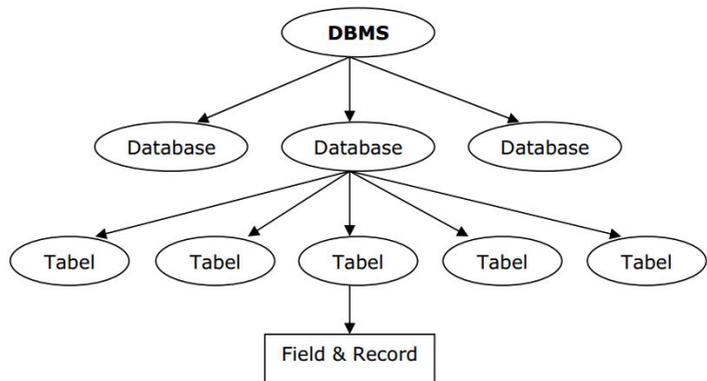
Key merupakan suatu *field* yang dapat dijadikan kunci dalam operasi tabel. Dalam konsep database, *key* memiliki banyak jenis diantaranya *Primary Key*, *Foreign Key*, *Composite Key*, dll.

5. SQL

SQL atau *Structured Query Language* merupakan suatu bahasa (*language*) yang digunakan untuk mengakses *database*. SQL sering disebut juga sebagai *query*.

2.5.2. Hierarki Database

Dalam konsep database, urutan atau hierarki database sangatlah penting. Urutan atau hierarki database digambarkan dalam gambar sebagai berikut :



Gambar 2.12. Hierarki Database

2.6 Pengertian MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *Database Management System*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak seperti PHP atau Apache yang merupakan *software* yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber

dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu MySQL AB. MySQL AB memegang penuh hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius. MySQL dapat didownload di situs resminya, <http://www.mysql.com>.

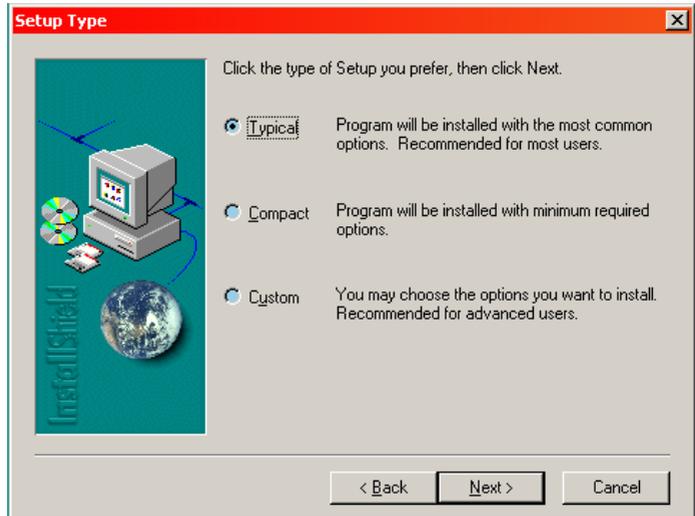
2.6.1. Fitur-fitur MySQL antara lain :

1. *Relational Database System*. Seperti halnya *software database* lain yang ada di pasaran, MySQL termasuk RDBMS.
2. Arsitektur *Client-Server*. MySQL memiliki arsitektur *client-server* dimana *server database* MySQL terinstal di *server*. Client MySQL dapat berada di komputer yang sama dengan server, dan dapat juga di komputer lain yang berkomunikasi dengan *server* melalui jaringan bahkan internet.
3. Mengenal perintah SQL standar. SQL (*Structured Query Language*) merupakan suatu bahasa standar yang berlaku di hampir semua *software database*. MySQL mendukung SQL versi SQL:2003.
4. Mendukung *Sub Select*. Mulai versi 4.1 MySQL telah mendukung *select* dalam *select (sub select)*.
5. Mendukung *Views*. MySQL mendukung *views* sejak versi 5.0
6. Mendukung *Stored Prosedured (SP)*. MySQL mendukung SP sejak versi 5.0
7. Mendukung *Triggers*. MySQL mendukung *trigger* pada versi 5.0 namun masih terbatas. Pengembang MySQL berjanji akan meningkatkan kemampuan *trigger* pada versi 5.1.
8. Mendukung *replication*.
9. Mendukung transaksi.
10. Mendukung *foreign key*.
11. Tersedia fungsi GIS.
12. *Free* (bebas didownload)
13. Stabil dan tangguh
14. Fleksibel dengan berbagai pemrograman
15. *Security* yang baik
16. Dukungan dari banyak komunitas
17. *Perkembangan software* yang cukup cepat.

2.6.2. Instalasi MySQL 3.23.52-Max

Step by Step Instalasi MySql sebagai server Database :

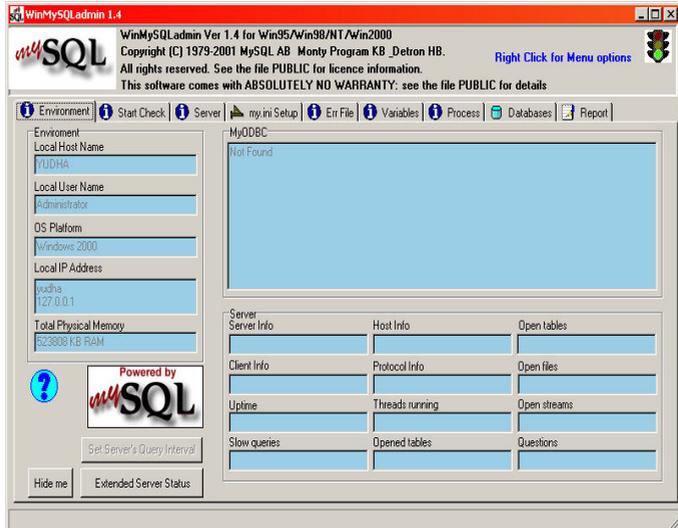
1. Jalankan program Setup.
2. Untuk penempatan directory atas hasil Instalasi MySql, Dapat anda tentukan letaknya sesuai dengan keinginan anda sendiri. Secara Default MySql akan memberikan tawaran untuk menempati C:\MySql sebagai default alternatif peletakkannya.
3. Pilihlah jenis Typical untuk Instalasinya. Seperti tampak dalam gambar. [Gambar 2.13]



Gambar 2.13. Typical Installasi

4. Jika anda dimintai mengisi user, isilah root sebagai usernya, dan masukkan manager untuk passwordnya. Tetapi jika tidak biarkan saja. (biasanya setup tidak meminta untuk memasukkan inputan user & password jika sebelumnya anda sudah pernah menginstall MySql di Computer anda)
5. Setelah proses Instalasi MySql sudah selesai, selanjutnya untuk menjalankan service MySql, anda harus masuk ke directory bin. Sebagai contoh jika anda menempatkan instalasi mysql tadi ada di directory c:/mysql maka, anda harus masuk ke c:/mysql/bin

6. Selanjutnya anda cari file dengan nama winmysqldadmin dan jalankan / execute file tersebut.
7. Setelah anda menjalankan file winmysqldadmin, biasanya muncul pula pertanyaan untuk menentukan user dan password untuk rootnya. Sebagai root anda tentukan isilah usernya root dan passwordnya adalah manager. Setelah menjalankan winmysqldadmin, maka service MySQL akan berjalan dengan ditandai munculnya tampilan seperti tampak pada gambar. [Gambar 2.14]



Gambar 2.14. MySQL running

8. Selanjutnya, tekan tombol Hide me tampak pada posisi pojok kanan-bawah dari tampilan gambar [Gambar 2.14] maka, tampilan akan hilang dan ditandai dengan munculnya icon service MySQL di posisi tray windows (pojok kanan bawah windows anda, bersebelahan dengan penunjuk waktu/tanggal). Service MySQL ini aktif jika icon di tray tersebut menunjukkan lampu-hijaunya seperti tampak pada gambar [Gambar 2.15]



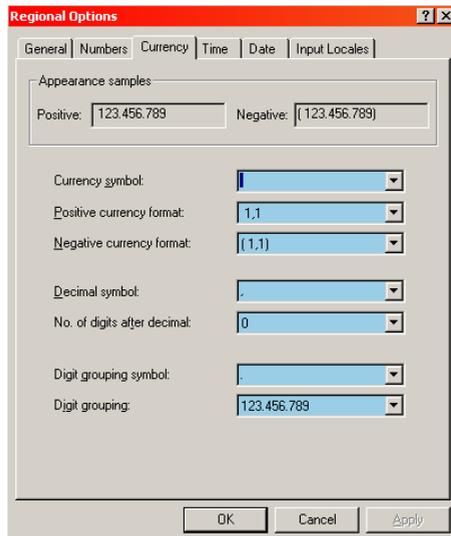
Perhatikan tanda Lampu Hijau sebagai indikator Service MySQL sudah Activ

Gambar 2.15. MySQL sudah aktif

9. Jika semua tahap (1-8 sudah terlaksana) berarti proses instalasi terhadap database MySQL sudah selesai. Selanjutnya tinggal anda melakukan modifikasi setting untuk Regional Setting Microsoft windows.

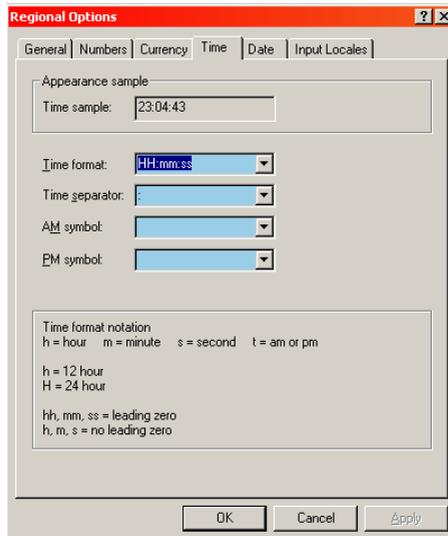
2.6.3. Penyesuaian Regional Setting Microsoft Windows :

1. Masuk ke Control Panel, dan Click icon Regional Setting, selanjutnya pada Tab-Sheet General anda rubah ke posisi Indonesia.
2. Pada posisi Currency, rubahlah tanda Rp dgn tanda blank 1 character. Ini berguna untuk mengurangi space dalam penulisan angka di program retail nantinya. Tampak pada Gambar [Gambar 2.16.]



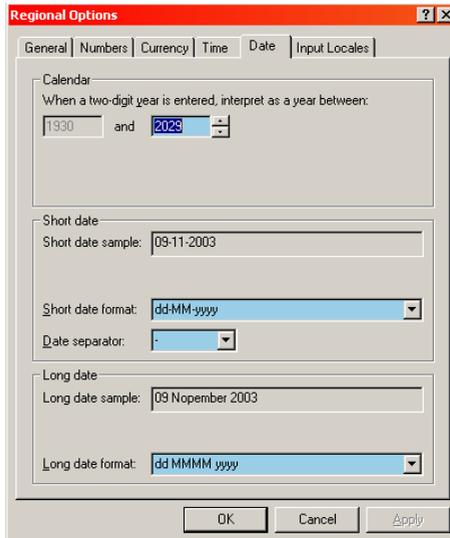
Gambar 2.16. Regional Option

3. Lanjutkan dengan Tab-Sheet Time. Rubahlah Format Jam seperti tampak pada tampilan Gambar [Gambar 2.17]



Gambar 2.17. Tab Sheet Time pada Regional Option

4. Terakhir, rubah pula untuk format Tanggal seperti tampak pada gambar [Gambar 2.18]



Gambar 2.18. Tab Sheet Time pada Regional Option