BAB II

LANDASAN TEORI

2.1 Konsep Sistem Informasi

Telah diketahui bahwa informasi merupakan hal yang sangat penting dan dikatakan bernilai apabila manfaat dari informasi tersebut lebih efektif dibandingkan dengan biaya untuk mendapatkannya di dalam pengambilan keputusan. Informasi itu sendiri dapat diperoleh dari sistem informasi atau disebut juga *Processing System.* Menurut **JOG [1]** sistem informasi adalah :

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

2.1.1 Konsep Dasar Sistem

Istilah "Sistem" sekarang ini banyak dipakai, konsep-konsep yang berhubungan dengan sistem telah diterapkan dalam berbagai rancangan sistem terhadap pemecahan masalah dan manajemen. Sebuah sistem terdiri atas bagian-bagian yang bergabung untuk satu tujuan. Model dasarnya adalah masukan, pengolahan dan keluaran.

2.1.2 Pengertian Sistem

Sistem dapat didefinisikan dengan dua pendekatan, yaitu sistem yang menekankan pada prosedur dan sistem yang menekankan pada elemen atau komponennya. Menurut **JOG [1]:**

"*Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan berkumpul bersama-sama untuk melakukan suatu kegiatan atau penyelesaian suatu sasaran tertentu*".

Suatu sistem mempunyai elemen-elemen penyusun, antara lain :

1. Tujuan

   Sasaran atau tujuan dari sistem sangat menentukan masukan yang dibutuhkan sistem.

2. Batasan

   Dalam mencapai tujuan dari sistem dibutuhkan batasan-batasan suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan suatu sistem menunjukan ruang lingkup dari sistem tersebut.

3. Kontrol

   Kontrol merupakan pengawasan dari pelaksanaan pencapaian tujuan sistem yang dapat berupa asal masukan, frekuensi masukan data, dan jenis masukan.

4. Input

   Input merupakan bagian dari sistem yang bertugas untuk menerima data masukan dimana data dapat berupa asal masukan, frekuensi masukan data dan jenis masukan data.

5. Proses

   Proses merupakan bagian yang mengolah data menjadi informasi sesuai dengan keinginan pemakai.

6. Output

   Output merupakan keluaran atau tujuan dari sistem yang didapat dari input dan proses yang dilakuakan.

7. Umpan Balik

   Dalam suatu sistem yang baik dibutuhkan adanya umpan balik yang tujuannya sebagai perbaikan dan pemeliharaan.

2.1.3 Analisis Sistem

Analisis sistem dapat didefinisikan sebagai sebagai penguraian dari suatu sistem informasi yang utuh kebagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahannya, kesempatan-kesempatan dan hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikan.

2.1.4 Konsep Dasar Informasi

Menurut **JOG [1]:**

" Informasi adalah data yang telah diolah menjadi suatu bentuk yang berarti bagi penerima dan bermanfaat dalam mengambil keputusan saat ini atau mendatang".

Kualitas dari suatu informasi tergantung dari tiga hal, yaitu :

1. Akurat

   Informasi harus bebas dari kesalahan-kesalahan yang biasanya terjasi dan selain itu harus jelas maksud dan tujuannya, sehingga *output* (keluaran) bisa dipertanggungjawabkan.

2. Tepat Waktu

   Informasi pada saat diperlukan tidak boleh terlambat karena informasi yang terlambat tidak akan mempunyai nilai lagi dalam pengambilan suatu keputusan.

3. Relevan

Informasi harus bermanfaat dan sesuai dengan apa yang dibutuhkan pemakai.

2.1.5 Konsep Dasar Sistem Informasi

Sebuah sistem informasi manajemen adalah sebuah sistem informasi yang selain melakukan semua pengolahan transaksi yang perlu untuk sebuah organisasi yang memberi dukungan informasi dan pengolahan untuk fungsi manajemen dan pengambilan keputusan.

Organisasi selalu membutuhkan sistem-sistem untuk mengumpulkan, mengolah, menyimpan, melihat kembali dan menyalurkan informasi. Komputer merupakan sebuah teknologi baru pada sistem informasi, sehingga sebuah sistem informasi berdasarkan komputer akan memberikan efisiensi waktu dan memudahkan dalam penyimpanan data.

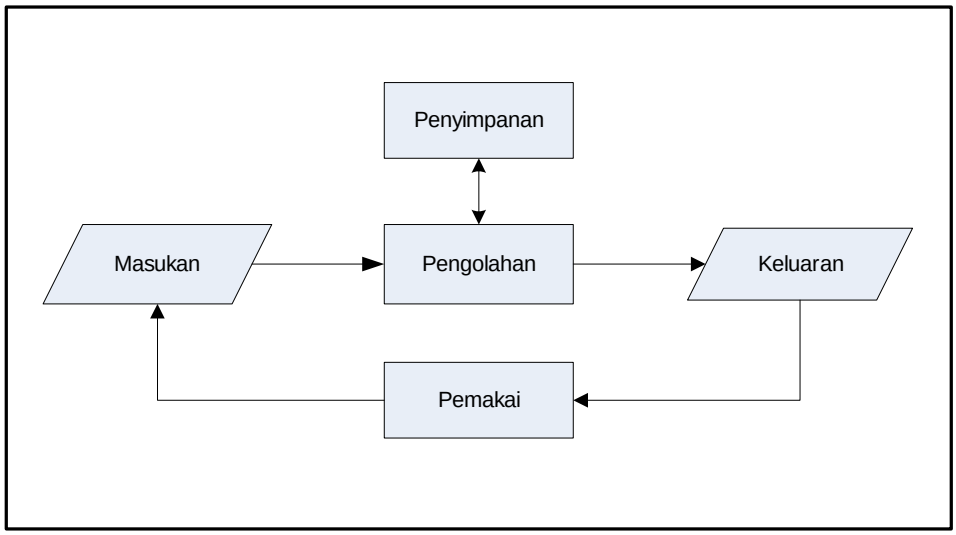
2.1.6 Pengertian Sistem Informasi

Menurut **JOG [1]:**

"suatu jaringan kerja yang merupakan kumpulan dari elemem-elemen yang saling berinteraksi untuk mencapai tujuan tertentu yaitu berupa informasi yang diperlukan dalam pengambilan keputusan baik untuk waktu sekarang ataupun diwaktu yang akan datang".

Suatu sistem informasi berisi himpunan terintegrasi dari komponen manual dan komponen terkomputerisasi yang bertujuan untuk menghasilkan informasi untuk pemakai.

Untuk lebih jelasnya mengenai komponen-komponen fungsional tersebut, maka dapat dilihat pada gambar dibawah ini :



Gambar 2.1 Komponen-Komponen Fungsional
(Sumber : Sistem Informasi Manajemen, Azhar Susanto, 2000, Lingga Jaya ; Bandung)

Sistem Informasi mempunyai beberapa komponen, antara lain :

1. Perangkat Keras (*Hardware*)

   Merupakan komponen fisik berupa peralatan input, peralatan proses, dan peralatan output.

2. Perangkat Lunak (*Software*)

   Merupakan intruksi-intruksi yang membuat komputer malakukan pekerjaan tertentu.

3. SDM (*Brainware*)

   Sebagai pengoperasi sistem.

4. Data

   Fakta-fakta, perkiraan-perkiraan, dan pendapat pandapat yang belum memiliki arti guna.

5. Prosedur

   Intruksi-intruksi yang digunakan dalam mengoperasikan sistem.

2.2 Alat-Alat Pemodelan Sistem Informasi

Alat-alat pemodelan sistem informasi adalah alat-alat yang digunakan dalam perancangan sistem informasi.

2.2.1 Perancangan Proses

Pada perancangan proses alat-alat pemodelan sistem informasi yang digunakan ada tiga jenis, yaitu :

1. Diagram alir Dokumen / *Flowchart*

   Diagram alir data atau *Flowchart* merupakan penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponen dengan maksud untuk mengidentifikasi serta dapat mengevaluasi suatu permasalahan yang diharapkan dapat diusulkan perbaikan-perbaikan.

2. Diagram konteks (*Context Diagram)*

   Context Diagram adalah bagian dari DFD yang berfungsi memetakan model lingkungan, yang dipresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem. *Context Diagram* menyoroti sejumlah karakteristik sistem yaitu :

   a. Kelompok Pemakai, organisasi atau sistem lain dimana sistem melakukan komunikasi (sebagai terminator)
   b. Data Masuk, yaitu data yang diterima system dari lingkungan dan harus diproses dengan cara tertentu.
   c. Data Keluar, yaitu data yang dihasilkan system dan diberikan ke dunia luar.
   d. Penyimpanan data (*Storage*), yaitu digunakan secara bersamaan antara sistem dengan terminator. Data ini dapat dibuat oleh sistem dan digunakan

oleh atau sebaliknya dibuat oleh lingkungan dan digunakan oleh sistem. Hal ini berarti pembuatan simbol penyimpanan dalam *Context Diagram* di benarkan dengan syarat simbol tersebut merupakan bagian dari dunia diluar system.

e. Batasan, antara system dan lingkungan

Simbol yang digunakan dalam *Context Diagram* (CD) antara lain :

1. Persegi Panjang (*terminator*)

   Untuk berkomunikasi langsung dengan system melalui aliran data

2. Lingkaran

   Untuk menunjukan adanya kegiatan proses dalam system.

3. *Data Flow Diagram* (DFD)

DFD sering digunakan untuk menggambarkan suatu system yang telah ada atau sistem baru yang dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau lingkungan fisik di mana data tersebut akan disimpan. DFD merupakan alat yang digunakan pada metodologi pengembangan system yang terstruktur. Sedangkan arti dari DFD tersebut adalah sebuah teknik yang mengambarkan aliran data dan transformasi yang digunakan sebagai perjalanan data dari masukan menuju keluaran.

Diagram ini menjelaskan bagaimana data masukan diubah menjadi keluaran dimana setiap bagian pada diagram menjelaskan proses transformasi yang berbeda. DFD dapat digunakan untuk menggambarkan system atau perangkat lunak pada banyak tingkatan dari suatu abstraksi.

DFD dapat dibagi kedalam tingkatan-tingkatan yang menggambarkan pertambahan atau peningkatan aliran data dan rincian fungsional. Level-0 pada DFD, menyatakan prosedur-prosedur dan keseluruhan system yang digambarkan pada model konteks, sedangkan model konteks merupakan model system pokok yang menggambarkan keseluruhan lemen perangkat lunak sebagai sebuah lingkaran dengan data masukan dan data keluaran dinyatakan dengan anak panah masuk atau keluar, berturut-turut dengan kata lain model konteks menjelaskan bagaimana gambaran system dan hubungan dengan pihak luar.

Setiap proses dan jalur-jalur aliran data di gambarkan pada tingkat yang lebih tinggi (misalnya level-1) dan dinyatakan proses-proses dari keseluruhan proses dari keseluruhan prosedur yang digambarkan pada level-0.

2.2.2 Perancangan Basis Data

DataBase adalah kumpulan file-file yang mempunyai kaitan antara satu file dengan file yang lain sehingga membentuk satu bangunan data untuk menginformasikan satu perusahaan, instansi dalam batasan tertentu. Apabila terdapat file yang tidak dapat dipadukan atau dihubugkan dengan file yang lain berarti file tersebut bukanlah kelompok dari satu database maka ia akan membentuk satu *database* baru.

Perancangan basis data diperlukan agar bisa diperoleh dari data yang komplek dan efisien dalam penggunaan ruang penyimpanan, cepat dalam pengaksesan dan mudah dalam pemanipulasian (tambah, ubah, hapus) data

Dalam merancang basis data, dapat dilakukan dengan :

1. Menerapkan normalisasi terhadap struktur table yang telah diketahui
2. Membuat table model *Entity Relationship*

Perancangan basis data dibutuhkan agar didapat sistem yang lengkap dan efisien melalui beberapa tahap, yaitu :

a. *Entity Relantioship Diagram* (ERD)

ERD adalah sebuah diagram yang menggambarkan model relasi antara rancangan data tersipan atau file, model relasi ini diperluikan untuk menggambarkan struktur data dari relasi antar data serta digunakan pula untuk menentukan hak pemakai (*user*) serta pemilikan data

Model ERD diagram dibentuk dari komponen dasar yaitu :

1 Entitas

Entitas adalah segala sesuatu yang ada dapat dibedakan entitas dapat berupa orang, benda, peristiwa atau konsep yang bisa memberikan atau mengandung informasi.

2 Atribut

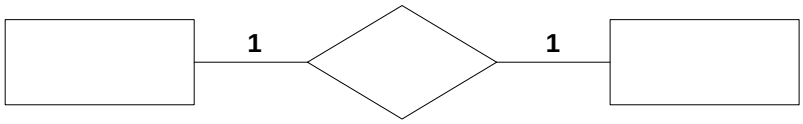
Setiap entitas mempunyai atribut atau elemen data yang mencirikan entitas tersebut.

3 Relasi

Hubungan antara entitas atau beberapa entitas.

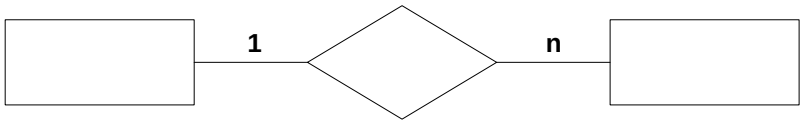
Jenis-jenis relasi yang terjadi diantara dua himpunan entitas dapat berupa

1. Satu ke Satu (One to One)



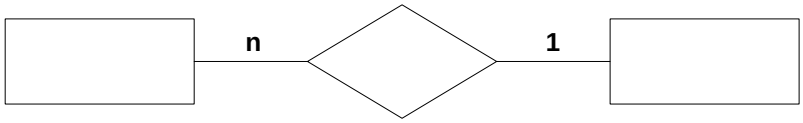
Relasi satu ke satu

2. Satu ke banyak (One to Many)



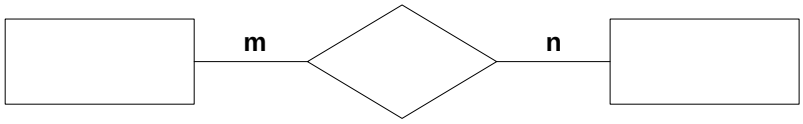
Relasi Satu ke banyak

3. Banyak ke Satu (Many to One)



Relasi banyak ke satu

4. Banyak ke Banyak (Many to Many)



Relasi banyak ke banyak

b. Teknik Normalisasi

Proses normalisasi yaitu proses pengelompokan data elemen menjadi table-table yang menunjukan Entitas dan Relasinya

1. Bentuk Normal Pertama (1NF)

   Bentuk normal tahap pertama (1NF) terpenuhi jika sebuah tabel tidak memiliki atribut bernilai banyak (*Multivalued Attribute*) atau lebih dari satu atribut dengan domain nilai yang sama.

2. Bentuk Normal Kedua (2NF)

   Bentuk normal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk normal kesatu. Atribut bukan kunci haruslah bergantung secara fungsi pada kunci utama / *primary key*

3. Bentuk Normal Ketiga (3NF)

   Bentuk normal ketiga (3NF) terpenuhi jika sebuah tabel, semua atribut yang tidak termasuk *primary key* memiliki ketergantungan pada kunci penentu selain *primary key.*

4. *Boyce Codd*

   Untuk menjadi BCNF (*Boyce Codd Noral Form)* relasi harus dala bentuk normal kesatu (1NF) dan setiap atribut harus memiliki ketergantungan fungsional pada atribut *Super Key*.

c. Kamus Data

Kamus data atau *system data dictionary* adalah katalog fakta tentang data kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan kamus data analisis sistem dapat mendefinisikan data yang mengalir kedalam sistem dengan lengkap pada tahap analisis sistem kamus data digunakan sebagai alat

komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir kedalam sistem. Pada tahap perancangan sistem kamus data di gunakan untuk merancang input, merancang laporan-laporan dan *database*.

2.3 Konfigurasi Sistem Client-Server

Database terdistribusi dapat dijalankan dengan menggunakan beberapa konfigurasi, salah satu yang cukup terkenal saat ini adalah dengan metode *Client-Server*. *Server* adalah komputer yang mempunyai kemampuan yang lebih dari komputer *client* dan menyimpan program dan data-data yang akan didistribusikan. Sistem *Client-Server* mempunyai dua komponen utama, yaitu komponen yang melakukan pemrosesan terbanyak untuk memenuhi permintaan-permintaan dari komputer *Client* dan bertindak sebagai *Server database* yang menyimpan data. Sementara *Client* adalah komputer atau *workstation* dengan kemampuan standar yang digunakan sebagai tampilan untuk user.

Sistem *Client-Server* merupakan suatu sistem komputer yang melibatkan proses-proses *Client* yang meminta suatu pelayanan data kepada komputer *Server* yang menyediakan layanan data tersebut. Sehingga baik *Client* maupun *Server* sama-sama melakukan suatu pekerjaan. Dengan adanya kombinasi *Client* (*Front-End*) dan *Server* (*Back-End*) ini, maka kumpulan dari modul-modul program tidak dieksekusi dalam memori yang sama namun terbagi dalam komputer *Client-Server*.

Dengan arsitektur seperti ini, modul yang memanggil menjadi *Client* yang meminta suatu layanan data dan modul yang dipanggil menjadi *Client* yang meminta suatu layanan data dan modul yang dipanggil menjadi *Server* yang menyediakan layanan data.

Sistem *Client-Server* berjalan, setidaknya pada dua komputer yang berbeda. Biasanya sebuah *Server* melayani beberapa komputer *Client*, walaupun mungkin ada juga yang hanya melayani satu *Client* saja. *Cilent* merupakan sebuah komputer dekstop yang terhubung ke jaringan.

Proses Server berperan sebagai aplikasi yng mengelola sumber daya milik bersama (*Shared Resources*) seperti *database*, printer, atau jalur komunikasi selain menjalankan sebagai *Back-End* sistem *Client-Server* yaitu pusat pemrosesan data. Sedangkan proses *Client* meliputi program-program untuk mengirimkan pesan permintaan pada Server serta melakukan pengaksesan data seperti mengedit, menghapus atau menambah data. Karena itu program pada *Client* adalah aplikasi *Front-End* yang digunakan sebagai antar muka (*interface*) bagi pemakai untuk berinteraksi dengan Server. Selain itu juga pemakaian sumber daya lokal seperti monitor, keyboard dan perangkat lokal lainnya.

Dengan cara kerja seperti ini, yang harus benar-benar dipikirkan adalah bagaimana membuat sistem database pada komputer Server dengan aplikasi sebagai antar muka bagi pemakai pada komputer *Client* dengan bahasa pemrograman yang dipilih kemudian menghubungkan aplikasi tersebut dengan database jarak jauh pada Server. Sering pemakaian tidak sadar bahwa data yang sedang diakses sebenarnya berada pada lokasi yang berjauhan.

2.4 Sistem Perparkiran

Kantor Pelayanan pajak bandung – Tegallega, sistem kinerja kerja perusahaannya tersebut masih menggunakan sistem manual. Yang prosedur kerjanya yaitu petugas parkir hanya mencatat nomor plat kendaraan yang masuk kelokasi parkir saja,apabila kendaraan tersebut keluar petugas parkir mengambil

karcis kendaraan yang keluar. Apabila pengguna parkir tersebut mengalami kehilangan karcis kendaraan, pengguna parkir harus menunjukkan Surat Tanda Nomor Kendaraan (STNK) kepada petugas parkir disertai denda yang sesuai peraturan yang telah ditentukan.

2.5 Perangkat Lunak Pendukung

2.5.1 SQL-Server

Menurut **MAG [2]** *SQL-Server* merupakan salah satu database server dimana database server adalah perangkat lunak yang mampu mengelola data dengan baik sehingga data yang tersimpan dapat digunakan kembali. Ketangguhan serta stabilitas *SQL-Server* adalah mampu berjalan pada berbagai platform sistem operasi. *SQL-Server* termasuk jenis RDBMS (*Relational Database Management System*).

2.5.2 Borland Delphi

2.5.2.1 Pengenalan Borland Delphi

Ide munculnya Borland Delphi sebenarnya berasal dari pemrograman yang sudah dikenal, yaitu Bahasa Pascal yang diciptakan pada tahun 1971 oleh seorang ilmuwan yang berasal dari Swiss yaitu Niclaus Wirth. Nama Pascal diambil dari ilmuwan matematika dan filsafat dari Perancis yaitu *Blaise Pascal*.

Sejak itu muncul beberapa versi Pascal diantaranya Turbo Pascal yang diliris oleh Borland International Incoporation pada tahun 1982. Semula Turbo Pascal hanya dapat dijalankan pada sistem operasi DOS, namun Borland Incoporation mengembangkan Turbo Pascal yang dapat dijalankan pada Windows 3x, yang diberi nama *Turbo Pascal for Windows*.

Borland telah berpengalaman memproduksi perangkat pengembang handal, seperti kelompok Turbo Basic, Turbo C, Turbo Prolog, Turbo Assembler/Turbo Pascal, Visual Basic, Borland C++, Borland Pascal maupun Borland C++ Builder.

Tahun 2001 Borland meluncurkan kembali versi terbaru dari bahasa pemrograman yang sangat diidamkan pemrograman professional, yaitu Borland Delphi. Berbagai jenis aplikasi dapat dibuat dengan Delphi, termasuk aplikasi untuk mengolah text, grafik, angka, database dan aplikasi Web.

Delphi merupakan perangkat pengembangan aplikasi yang sangat terkenal di lingkungan Window. Dengan menggunakan perangkat lunak ini juga dapat membangun aplikasi Windows dengan cepat dan mudah. Dengan pendekatan visual, kita dapat menciptakan aplikasi yang canggih tanpa banyak menuliskan kode.

Delphi menggunakan bahasa Objek Pascal sebagai dasar. Jika telah menguasai pascal, maka akan lebih mudah memahami program delphi. Untuk mempermudah pemrograman dalam membuat program aplikasi, delphi menyediakan fasilitas pemrograman yang secara lengkap. Khusus untuk pemrograman database, delphi menyediakan objek yang sangat kuat, canggih dan lengkap, sehingga memudahkan pemrograman dalam merancang, membuat dan menyelesaikan aplikasi database yang diinginkan. Selain itu delphi juga dapat menangani data dalam berbagai format database, misalnya MS. Accses, SyBase, Oracle, FoxPro, Informix, Interbase, SQL Server dan lain-lain. Format database yang dianggap asli dari Delphi adah Paradox dan dBase.

Keunggulan yang ditawarkan Delphi 7.0 adalah IDAS (Multi User Distribution Applikation Server) yang memungkinkan sebuah aplikasi untuk berkomunikasi dan berbagi data secara mendunia melalui jaringan komunikasi. Delphi sebelunya dipasarkan dengan harga berbeda yang terdiri dari Delphi standard, Delphi professional, Delphi client/server suite dan Delphi interprise. Selain itu pemograman Delphi 7.0 juga memiliki fasilitas-fasilitas yang menjadikanya memiliki struktur dan format yang lebih efisien dan efektif.

2.5.2.2 IDE Delphi

Pada dasarnya IDE (*Integrated Development* Environment) Delphi dibagi menjadi tujuh bagian utama yaitu *menu, speed bar, Component Palette, Form Designer, Code Explorer, Object TreeView,dan Object Inspector.*

1. **Menu**

   Menu pada delphi memiliki kegunaan seperti menu menu pada aplikasi windows lainnya. Segala sesuatu yang berhubungan dengan *IDE delphi* dapat dilakukan melalui menu seperti memanggil program, menyimpan program, menjalankan program.

2. **Speed Bar**

   Speed bar atau sering juga disebut *toolbar* berisi kumpulan tombol yang tidak lain adalah pengganti beberapa item menu yang sering digunakan. Dengan kata lain, setiap tombol pada *speed bar* menggantikan salah satu item menu.



Gambar 2.5 *Speed bar* pada IDE Delphi
(Sumber : Aplikasi SMS Untuk berbagai keperluan, Zakaria Teddy Marcus, 2006, Informatika, Bandung)

3. Component Palette



Gambar 2.6 *Component Palette* pada IDE Delphi
(Sumber : Aplikasi SMS Untuk berbagai keperluan, Zakaria Teddy Marcus, 2006, Informatika, Bandung)

Component palette adalah pustaka dari komponen-komponen yang digambarkan berupa *icon-icon.* Komponen-komponen ini dibagi menjadi beberapa kelompok. Pada dasarnya, komponen ini dikelompokkan berdasarkan fungsinya. Pengelompokkan ini dinyatakan dengan nama tab/pages. Page asal yang disediakan sebagai berikut :

1. Tab Standart

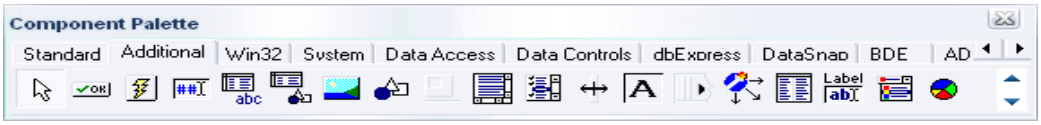


Gambar 2.7 *Tab Standar* IDE Delphi
(Sumber : Aplikasi SMS Untuk berbagai keperluan, Zakaria Teddy Marcus, 2006, Informatika, Bandung)

Nomor Icon	Jenis	Komponen
Frame	Non Visual	Digunakan untuk menampung komponen-komponen. Misalnya beberapa komponen disimpan pada sebuah *frame*. Bila *frame* ini dipanggil kedalam sebuah *form*, maka komponen yang ada di frame tersebut akan terbawa.
MainMenu	Non Visual	Digunakan untuk membuat menu bar dan menu *drop-down*
PopUpMenu	Non Visual	Untuk membuat menu *PopUpmenu* yang akan muncul jika user mengklik tombol kanan *mouse*.
Label	Visual	Digunakan untuk menempatkan teks didalam form.
Edit	Visual	Digunakan untuk menerima masukan satu baris teks dan dapat juga dipakai untuk menampilkan teks.
Memo	Visual	Digunakan untuk menerima masukan atau menampilkan beberapa teks.
Button	Visual	Digunakan untuk membuat tombol.
CheckBox	Visual	Untuk memilih atau membatalkan pilihan, caranya dengan mengklikj pada checkbox.
RadioButton	Visual	Untuk memberikan sekumpulan pilihan dan hanya satu yang dapat dipilih.

ListBox	Visual	Untuk membuat sebuah daftar itemdan user dapat memilih salah satu diantaranya.
ComboBox	Visual	Mirip seperti listBox tetapi memiliki unsur komponen edit. User dapat memilih item dan dapat mengetikkan text dalam kotak.
ScrollBar	Visual	Untuk menggulung form.
GroupBox	Visual	Merupakan sebuah kontainer, yang dipakai untuk mengelompokkan kontrol-kontrol dan komponen radioButton, checkbox dan lainnya yang berhubungan dengan komponen visual.
RadioGroup	Visual	Merupakan kombinasi dari GroupBox dan RadioButton.
Panel	Visual	Merupakan kontainer untuk mengelompokkan komponen-komponen, misalnya edit, label, Status bar.
ActionList	NonVisual	Komponen yang berisi daftar aksi yang digunakan bersama-sama dengan komponen dan kontrol seperti item menu dan button.

2. Tab Additional



Gambar 2.8 *Tab additional* pada Delphi 7.0
(Sumber : Aplikasi SMS Untuk berbagai keperluan, Zakaria Teddy Marcus, 2006, Informatika, Bandung)

Komponen	Jenis	Fungsi
BitBtn	Visual	Mirip dengan komponen button, hanya saja *BitBtn* dilengkapi dengan grafik *bitmap*.
SpeedButton	Visual	Button yang dipakai bersama-sama dengan komponen panel yang digunakan untuk membuat *toolbar* dan *button-button* khusus.
MaskEdit	Visual	untuk memformat data masukan.
StringGrid	Visual	Untuk menampilkan data string dalam baris dan kolom.
Draw Grid	Visual	Digunakan untuk menampilkan informasi yang bukan text dalam baris dan kolom.
Image	Visual	Digunakan untuk menampilkan grafik seperti seperti ikon, *bitmap* dan *metafile*.
Bevel	Visual	Untuk menggambar segiempat dengan tampilan *insert* atau menonjol.
ScrollBox	Visual	Untuk membuat area tampilan yang bisa menggulung.
CheckListBox	Visual	Pengembangan dari *ListBox*, digunakan untuk *MultiSelect*.
StaticText	Visual	Mirip dengan label, tetapi dengan kelebihan untuk pengontrolan windows.
ContolBar	Visual	Untuk mengatur tata letak dari komponen toolbar.

ValueList Editor	Visual	Mirip dengan ListBox, tetapi dengan kemampuan dua panel terdiri atas daftar (*key list*) dan nilai (*value list*) yang dapat diedit.
LabelEdit	Visual	Gabungan komponen antara *Tlabel* dan *Tedit*.
Chart	Visual	Digunakan untuk membuat *chart* atau grafik.

3. Tab Win32



Gambar 2.9 *Tab Win32* pada Delphi 7.0
(Sumber : Aplikasi SMS Untuk berbagai keperluan, Zakaria Teddy Marcus, 2006, Informatika, Bandung)

Komponen	Jenis	Fungsi
TabControl	Visual	Digunakan untuk menambah tab pada sebuah *form*.
PageControl	Visual	Untuk membuat page-page pada sebuah *form*.
ImageList	Non Visual	Untuk mendaftarkan gambar-gambar. Setiap gambar didaftarkan dan diberi nomor indek mulai dari nomor 0.
RichEdit	Visual	Edit box dengan beberapa warna, huruf dan lain-lain.
Trackbar	Visual	Untuk membuat trackbar pada *form*
ProgressBar	Visual	Dipakai untuk mengetahi *progress* dari jalannya prosedur.
DateTimePicker	Visual	Untuk Memasukkan data jam dan tanggal.
MonthCalender	Visual	Dipakai untuk memilih sebuah tanggal atau *range* tanggal.
StatusBar	Visual	Untuk menampilkan informasi status di beberapa panel.

4. Tab System



Gambar 2.10 *Tab system* pada Delphi 7.0
(Sumber : Aplikasi SMS Untuk berbagai keperluan, Zakaria Teddy Marcus, 2006, Informatika, Bandung)

Komponen	Jenis	Fungsi
Timer	Non Visual	Untuk mengaktifkan *procedur*, *fungsi*, dan *event* pada interval waktu tertentu.
PaintBox	Visual	Untuk membuat area yang dapat dicat.
MediaPlayer	Visual	Untuk mengontrol panel sehingga terlihat seperti *VCR*, dipakai untuk mengontrol *file-file* suara dan video.
OleContainer	Visual	Dipakai untuk membuat OLE Client.
DateTimePicker	Visual	Untuk Memasukkan data jam dan tanggal.
MonthCalender	Visual	Dipakai untuk memilih sebuah tanggal atau range tanggal.
StatusBar	Visual	Untuk menampilkan informasi status di beberapa panel.

5. Tab Data Access



Gambar 2.11 *Tab Data Access* pada Delphi 7.0
(Sumber : Aplikasi SMS Untuk berbagai keperluan, Zakaria Teddy Marcus, 2006, Informatika, Bandung)

Komponen	Jenis	Fungsi
DataSource	Non Visual	Bertindak sebagai penghubung antara komponen pengakses data dengan dataset (Tabel Atau Query).
Table	Non Visual	Komponen yang mengakses *record-record* dari suatu tabel.
Query	Non Visual	Komponen yang mengakses *record-record* dari suatu tabel.
StoreProc	Non Visual	Komponen yang mengakses *record-record* dari suatu tabel berdasarkan perintah *store procdur* euntuk database server.
Database	Non visual	Digunakan menciptakan koneksi *database* antara *client* dan *server*.
Session	Non Visual	Menciptakan sebuah *session* dalam aplikasi *multi-threaded dataset.*
Shape	Visual	Untuk menampilkan bentuk-bentuk dasar, seperti segiempat, segitiga dan lingkaran.
Bevel	Visual	Untuk menggambar segiempat dengan tampilan *insert* atau menonjol.
ScrollBox	Visual	Untuk membuat area tampilan yang bisa menggulung.
CheckListBox	Visual	Pengembangan dari *ListBox*, digunakan untuk *MultiSelect*.
StaticText	Visual	Mirip dengan label, tetapi dengan kelebihan untuk pengontrolan windows.
ContolBar	Visual	Untuk mengatur tata letak dari komponen toolbar.

ValueList Editor	Visual	Mirip dengan ListBox, tetapi dengan kemampuan dua panel terdiri atas daftar (*key list*) dan nilai (*value list*) yang dapat diedit.
LabelEdit	Visual	Gabungan komponen antara *Tlabel* dan *Tedit*.
Chart	Visual	Digunakan untuk membuat *chart* atau grafik.

6. Tab Data Access



Gambar 2.12 *Tab Data control* pada Delphi 7.0
(Sumber : Aplikasi SMS Untuk berbagai keperluan, Zakaria Teddy Marcus, 2006, Informatika, Bandung)

Komponen	Jenis	Fungsi
DBGrid	Visual	Tampilan dan edit *dataset* dalam format tabular (baris dan kolom).
DBNavigator	Visual	Untuk mengontrol/menggerakkan posisi record, seperti *Previous, Next, First, Last Record* dan mengubah statusnya seperti *open, close, edit, post* (simpan), *Delete, Cancel Edit.*
DbText	Visual	Menampilkan sebuah *field* dalam bentuk label
DBEdit	Visual	Menampilkan dan Mengedit sebuah *field* dalam kotak edit 1 baris.
DBMemo	Visual	Menampilkan dan mengedit sebuah *field* dalam kotak edit beberapa baris. Biasanya *field* memo.
DBImage	Visual	Menampilkan dan mengedit sebuah *field image* atau *binary* blok data.
DBLookupListBox	Visual	Menampilkan sejumlah pilihan untuk mengisi sejumlah field, pilihan tersebut diambil dari dataset lain.
DBLookupComboBox	Visual	Menampilkan sebuah edit-box dan drop-list untuk mengisi sebuah field, pilihan dari drop-list diambil dari dataset lain.

7. Tab QReport



Gambar 2.13 *Tab Qreport* pada Delphi 7.0
(Sumber : Aplikasi SMS Untuk berbagai keperluan, Zakaria Teddy Marcus, 2006, Informatika, Bandung)

Komponen	Jenis	Fungsi
QuickRep	Visual	Merupakan *form* dasar untuk membuat laporan, pada *form* ini ditampatkan komponen yang berhubungan dengan laporan yang akan dicetak .
QRSubdetail	Visual	Digunakan untuk mencetak laporan dengan format *master-detail* dan dicetak setelah detail band. *Band* ini memiliki dataset sendiri, record yang akan dicetak milai dari *record* awal sampai *record* akhir.
QRBand	Visual	Merupakan komponen dari semua *band* yang ada, komponen ini dapat diubah jenisnya menjadi *band* apa saja yang ada.
QRChildBand	Visual	*Band* ini digunakan apabila mencetak laporan yang lebar dan tidak cukup, maka komponen tersebut harus dipindahkan kekomponen ini.
QRGroup	Visual	Digunakan untuk mencetak laporan yang dibagi menurut group. *Band* ini memiliki rumus yang menentukan kapan *band* ini dicetak.
QRLabel	Visual	Digunakan untuk mencetak teks dilaporan.
QRDBtext	Visual	Digunakan untuk mencetak *field* dari *dataset*. *Field* kalkulasi, string dan berbagai tipe numerik, tanggal dapat dicetak menggunakan komponen ini.
QRExpr	Visual	Digunakan untuk mencetak hasil hitungan rumus.
QRSysData	Visual	Untuk mencetak informasi sistem seperti judul laporan, nomor, halaman, tanggal dan jam.
QRMemo	Visual	Digunakan untuk mencetak komponen memo.
Komponen	**Jenis**	**Fungsi**
ADOConnection	Non Visual	Komponen yang berfungsi untuk menghubungkan aplikasi dengan basisdata.
ADOCommand	Non Visual	Perintah-perintah SQL untuk teknik ADO, tanpa mengembalikan sebuah nilai.
ADODataSet	Visual	Perwakilan data dari satu atau lebih tabel yang terdapat dalam sebuah database ADO.
ADOTable	Non Visual	Perwakilan data dari sebuah tabel yang diakses lewat ADO setelah terhubung dengan *datasource*.
ADOQuery	Non Visual	Perintah SQL untuk memanggil data fisik (tabel) didalam sebuah basisdata.
ADOStoredProc	Non visual	Digunakan untuk aplikasi yang mengakses *StoreProcedure* milik *server* dengan teknik ADO.
RDSConnection	Non Visual	Mengelola sekumpulan *record* yang dilewatkan dari sebuah proses atau mesin ke mesin lainnya. Komponen ini digunakan untuk membangun aplikasi *Multi-Tier*.