BAB II
TINJAUAN PUSTAKA

2.1 SEJARAH INSTANSI

Awal kelistrikan di Bumi Parahyangan sudah ada semenjak Pemerintah Kolonial Belanda masih bercokol di tataran tanah Sunda. Di tahun 1905, di Jawa Barat khususnya kota Bandung, berdiri perusahaan yang mengelola penyediaan tenaga listrik bagi kepentingan publik. Nama perusahaan itu ***Bandungsche Electriciteit Maatschaappij*** **(BEM)**.Dalam perjalanannya, BEM pada tanggal 1 Januari 1920 berubah menjadi Perusahaan Perseroan menjadi ***Gemeenschapplijk Electriciteit Bedrijf Voor Bandoeng*** **(GEBEO)** yang pendiriannya dikukuhkan melalui akte notaris Mr. Andriaan Hendrik Van Ophuisen dengan Nomor: 213 pada tanggal 31 Desember 1949.

Setelah kekuasaan penjajahan beralih ke tangan Pemerintah Jepang, di antara rentah waktu 1942 - 1945, pendistribusian tenaga listrik dilaksanakan oleh ***Djawa Denki Djigyo Sha Bandoeng Shi Sha*** dengan wilayah kerja di seluruh Pulau Jawa.Setelah Indonesia merdeka, tahun 1957 menjadi awal penguasaan pengelolaan penyediaan tenaga listrik di seluruh tanah air yang ditangani langsung oleh Pemerintah Indonesia, 27 Desember 1957, GEBEO diambil alih oleh Pemerintah Indonesia yang kemudian dikukuhkan lewat Peraturan Pemerintah No. 86 Tahun 1958 j.o. Peraturan Pemerintah No. 18 Tahun 1959.

Selanjutnya, di tahun 1961 melalui Peraturan Pemerintah No. 67 dibentuk **Badan Pimpinan Umum Perusahaan Listrik Negara (BPU-PLN)** sebagai wadah kesatuan pimpinan PLN. Sejalan dengan itu, PLN Bandung pun berubah menjadi **PLN Exploitasi XI** sebagai kesatuan BPU-PLN di Jawa Barat, di luar DKI Jaya dan Tangerang.Pada tahun 1970-an dikeluarkan Peraturan Pemerintah No. 18 Tahun 1972 tentang Perusahaan Umum Listrik Negara yang menyebutkan status PLN menjadi **Perusahaan Umum Listrik Negara**. Kemudian, berdasarkan Pengumuman PLN Exploitasi XI No. 05/DIII/Sek/1975 tanggal 14 Juli 1975, PLN Exploitasi XI diubah namanya menjadi **Perusahaan Umum Listrik Negara Distribusi Jawa Barat**.

Memasuki era 1990-an, dengan adanya Peraturan Pemerintah Republik Indonesia No. 23 Tahun 1994 pada tanggal 16 Juni 1994, Perusahaan Umum Listrik Negara Distribusi Jawa Barat diubah lagi menjadi Perusahaan Perseroan (Persero) dengan nama **PT PLN (Persero) Distribusi Jawa Barat** sejak tanggal 30 Juli 1994.Untuk memenuhi tuntutan perubahan dan perkembangan kelistrikan yang dari tahun ke tahun cenderung mengalami peningkatan, maka keluarlah Keputusan Direksi PT PLN (Persero) No. 28.K/010/DIR/2001 tanggal 20 Februari 2001 yang menjadi landasan hukum perubahan nama PT PLN (Persero) Distribusi Jawa Barat menjadi **PT PLN (Persero) Unit Bisnis Distribusi Jawa Barat.**Pada akhirnya, dengan mengacu pada Keputusan Direksi PT PLN (Persero) No. 120.K/010/DIR/2002 tanggal 27 Agustus 2002, PT PLN (Persero) Unit Bisnis Distribusi Jawa Barat berubah lagi namanya menjadi **PT PLN (Persero) Distribusi Jawa Barat dan Banten**, di mana **wilayah kerjanya meliputi Propinsi Jawa Barat dan Propinsi Banten**, hingga saat ini.

2.2 VISI DAN MISI INSTANSI

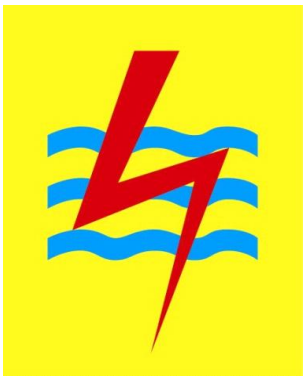
2.2.1 Visi :

- Terwujudnya sumber pelistrikan yang efisien bagi masyarakat khususnya masyarakat untuk wilayah jawabarat dan banten.

2.2.2 Misi :

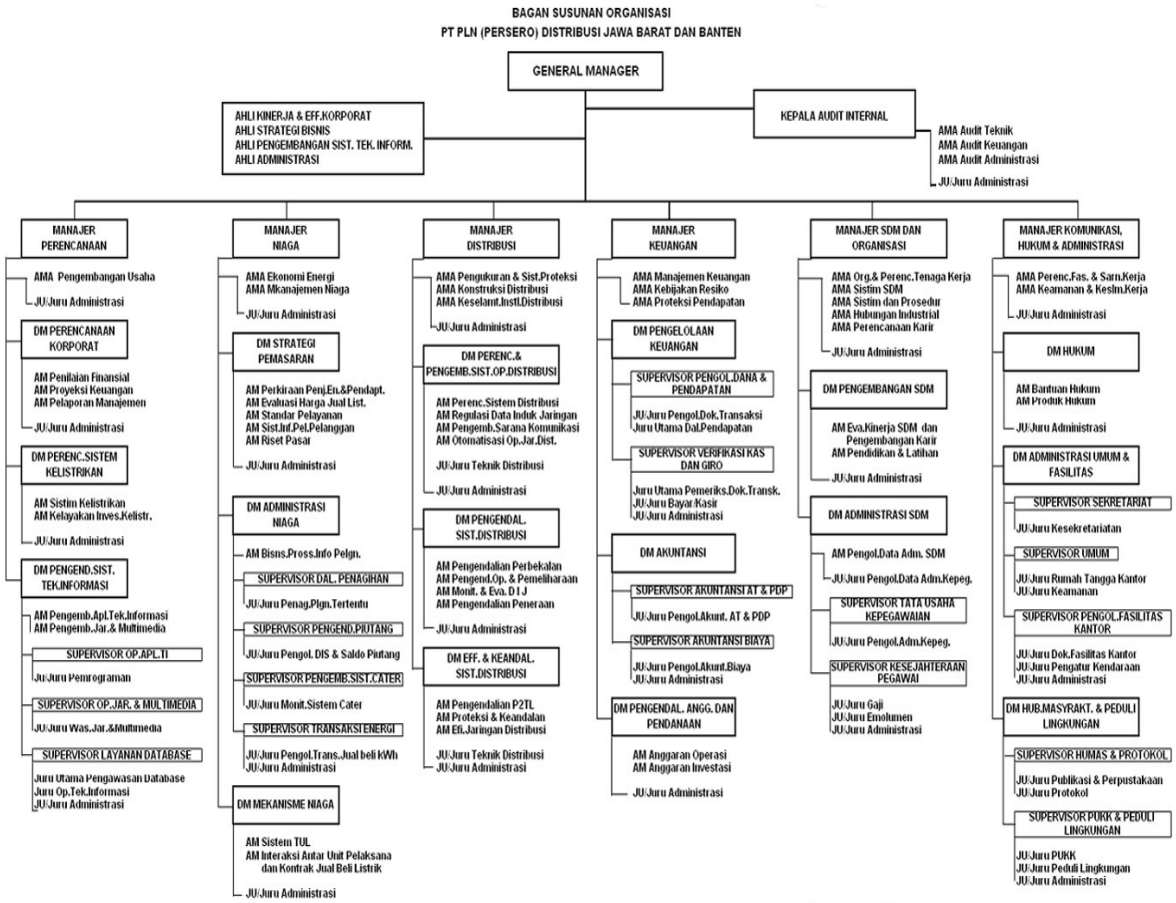
- Melayani dan memelihara pelistrikan bagi masyarakat sekitar.

2.3 LOGO INSTANSI



Gambar 2.1 Logo PT.PLN (Persero)

2.4 STRUKTUR ORGANISASI



Gambar 2.2 Struktur Organisasi

2.5 WILAYAH KERJA & FRONT LINER

Luas wilayah kerja PT PLN (Persero) Distribusi Jawa Barat dan Banten (PLN DJBB) menjangkau lebih dari 42.196 km² yang meliputi Propinsi Jawa Barat dan Propinsi Banten, kecuali Tangerang.

Jumlah konsumen yang mencapai lebih dari 8,474 juta pelanggan, atau 22 % dari jumlah pelanggan PLN secara nasional, menjadikan PLN DJBB merupakan Unit PLN terbesar di Indonesia.

Wilayah dan beban kerja yang sedemikian besarnya, dikelola oleh Unit-unit Pelaksana Area Pelayanan dan Jaringan (APJ) dan Area Pengatur Distribusi (APD), dengan komposisi sebagai berikut:

- Area Pelayanan Jaringan (APJ) : 16 Unit
- Area Pengatur Distribusi (APD) : 1 Unit
- Unit Pelayanan dan Jaringan (UPJ) : 100 Unit
- Unit Pelayanan dan Jaringan Prima (UPJ Prima) : 7 Unit
- Kantor Pelayanan (KP) : 191 KP

Adapun Unit-unit Pelaksana tersebut adalah :

- APD Bandung
- APJ Bandung
- APJ Banten Utara
- APJ Banten Selatan
- APJ Bekasi
- APJ Bogor
- APJ Cianjur
- APJ Cimahi
- APJ Cirebon
- APJ Depok
- APJ Garut
- APJ Gunung Putri
- APJ Karawang
- APJ Majalaya
- APJ Purwakarta
- APJ Sukabumi
- APJ Sumedang
- APJ Tasikmalaya

2.6 POTENSI DAN TANTANGAN KE DEPAN

2.6.1 Potensi

Di masa depan, perkembangan kelistrikan di Jawa Barat dan Banten diperkirakan masih akan terus mengalami pertumbuhan yang pesat. Hal ini disebabkan besarnya potensi yang ada di Jawa Barat dan Banten khususnya, maupun potensi karena lingkungan nasional yang lebih kondusif, diantaranya :

- Rasio Elektrifikasi yang relatif masih rendah (58%)
- Daftar tunggu calon pelanggan PLN yang relatif masih cukup besar
- Pertumbuhan ekonomi dan produk ekspor relatif semakin prospektif

- Posisi Jawa Barat dan Banten sebagai penyangga ibukota negara RI
- Potensi alam sebagai modal pertumbuhan pariwisata daerah
- Potensi sumber energi terbarukan yang sangat besar, khususnya panas bumi dan air.
- Berkembangnya sektor Jasa dan Agrobisnis
- Perkembangan daerah sejalan dengan kebijakan Otonomi Daerah
- Minat Pemerintah Daerah untuk memberikan kontribusi di bidang Kelistrikan
- Perkembangan Teknologi, khususnya Teknologi Informasi dan komunikasi
- Dibangunnya beberapa unit pembangkit baru di Jawa Barat & banten
- Potensi pengembangan jaringan distribusi melalui jaringan 70 kV
- Potensi excess power dari pelanggan besar yang memiliki pembangkit sendiri
- Potensi-potensi bisnis baru yang timbul dari telah adanya Jalan Tol Cipularang
- Peluang sinergi dengan BUMN lain (Bank, PT Pos, dll)
- Peluang sinergi dengan Anak Perusahaan (Icon+, JP, dll)

2.6.2 Tantangan

Selain adanya potensi yang memberikan peluang pertumbuhan yang cukup optimis di masa depan, PLN DJBB menghadapi pula beberapa hal yang menjadi tantangan untuk diselesaikan atau bahkan untuk menciptakan peluang usaha baru bagi PLN DJBB. Adapun beberapa tantangan yang ada antara lain :

- Adanya Visi 75 / 100 dan Visi Caang 2010, menuntut segenap jajaran PLN DJBB untuk bekerja lebih keras lagi untuk merealisasikannya di tengah keterbatasan finansial investasinya.
- Kenyataan bahwa harga jual tenaga listrik lebih rendah dibandingkan dengan biaya pokok penyediaan tenaga listrik, menjadi tantangan tersendiri bagi PLN DJBB untuk secara signifikan meningkatkan efisiensi
- Pertumbuhan konsumsi energi listrik yang masih relatif tinggi di Jawa Barat & Banten menuntuk PLN DJBB untuk lebih antisipatif dan sistematis dalam penyiapan infrastruktur dan layanannya.
- Adanya tren pelanggan besar untuk membangun pembangkit listrik sendiri, sehingga dalam jangka pendek mereka meminta untuk turun daya
- Masih besarnya penduduk yang belum menikmati listrik, sedangkan pada umumnya diperlukan investasi yang sangat besar untuk membangun infrastruktur kelistrikannya

- Keterbatasan investasi untuk penyediaan listrik PLN, telah memunculkan keinginan beberapa pihak untuk membangun kawasan eksklusif penyediaan listriknya (seperti kasus Cikarang Listrindo)

- Belum adanya Gardu Induk di kawasan Selatan Jawa Barat dan Banten yang mengakibatkan tegangan suplai ke pelanggan sangat drop.

- Sangat besarnya beban kerja beberapa APJ (misal APJ Bekasi dan APJ Bogor), sehingga diperlukan pembentukan APJ-APJ baru.

- Masih rendahnya rasio-rasio distribusi (misal ratio kVA trafo distribusi terhadap kVA trafo GI) yang menyebabkan timbulnya *bottle-neck* maupun rendahnya tingkat keandalan.

- Kondisi eksternal yang sangat berpengaruh, misalnya kenaikan harga minyak dunia, instabilitas situasi politik daerah karena Pilkada atau lainnya, tingkat inflasi, kebijakan UMP, gejolak kurs valuta asing, masyarakat yang semakin kritis, dan sebagainya.

2.7 STRATEGI DAN KEBIJAKAN YANG INOVATIF

2.7.1 Payment Point Online Bank (PPOB)

Pada tahun 2007 di-launch sebuah kebijakan yang inovatif, yaitu *Payment Point On-line* Bank (PPOB). PPOB adalah sebuah layanan pelunasan rekening listrik secara *online* yang memanfaatkan infrastruktur milik Bank. Dengan demikian pelanggan dapat membayar di ATM maupun *payment point* mana pun dan kapanpun. Model ini berhasil pula memunculkan peluang usaha baru bagi masyarakat berupa usaha *payment point*. Di sisi PLN selain mendapat nilai tambah berupa *revenue protection*, namun juga sebuah efisiensi signifikan karena tidak perlu membangun infrastruktur *online* sendiri.

2.7.2 Layanan Penyambungan Baru / Perubahan Daya Melalui Jasa PT Pos

Sebuah kebijakan strategis inovatif lainnya adalah Sinergi dengan PT Pos dalam jasa layanan sambungan baru (PB) maupun perubahan daya (PD). Masyarakat kini dapat mengajukan permohonan PB / PD di kantor Pos secara transparan, mudah dan cepat. Kebijakan ini menuntut segenap jajaran PLN untuk lebih antisipatif dan responsif dalam layanan PB / PD.

2.7.3 Tusbung *Online*

Pelaksanaan pemutusan sementara kepada penunggak listrik secara *online* dan *mobile*. Pelanggan yang menunggak akan didatangi oleh petugas, dan dilakukan pemutusan, namun pelanggan tersebut dapat langsung membayar di lokasi serta langsung disambung kembali. Model ini lebih ditujukan pada pembudayaan membayar listrik tepat waktu.

2.7.4 Layanan *Prepaid* (=Prabayar, Masih dalam Tahap *Pilot Project*)

Fenomena prepaid begitu meluas pada waktu penggunaan komunikasi seluler memasyarakat. Dan PLN DJBB tidak mau ketinggalan dengan model itu. Dengan layanan *prepaid* (prabayar), maka pelanggan akan membayar dulu 'pulsa' listriknya (di sini

diistilahkan dengan *token*). Model ini cukup prospektif mengingat bahwa sistem ini akan membantu masyarakat kecil dalam pembayaran listriknya secara 'eceran'. Di sisi lain, sistem ini akan merangsang masyarakat dalam pembudayaan penghematan energi.

2.7.5 *Lending Working-Capital*

Sebuah peluang bagi pelanggan besar PLN DJBB untuk menjamin ketepatan waktu pelunasan tagihan listrik, tanpa dihantui denda keterlambatan. Yaitu dengan skema kerjasama segitiga : Pelanggan-Bank-PLN DJBB. Pihak Bank akan membantu menyediakan dana talangan untuk melinasi tagihan listrik pelanggan, dengan tingkat bunga yang lebih rendah dibandingkan denda keterlambatan PLN.

2.7.6 Implementasi Teknologi Baru

Mulai 2008 PLN DJBB akan banyak mengimplementasikan penerapan Teknologi Baru dalam operasionalnya. Teknologi baru tersebut dapat berupa teknologi informasi, otomasi, telekomunikasi, pemanfaatan GIS *(Geographical Information System)*, pemulihan gangguan dan sebagainya.

2.8 PROGRAM KEMITRAAN DAN BINA LINGKUNGAN (PKBL)

Sebagai tanggungjawab moral dan sosial Perusahaan kepada lingkungan, yang didorong oleh rasa kepedulian dan keinginan untuk terlibat aktif dalam membangun potensi sosial kemasyarakatan, maka PT PLN DJBB mengembangkan Program Kemitraan dan Bina Lingkungan (PKBL).

2.8.1 Program Kemitraan

Program ini merupakan perwujudan nyata PLN DJBB untuk turut terlibat dalam pengembangan ekonomi rakyat, khususnya sektor usaha kecil dan menengah.Sejak digulirkan pada tahun 1992, PT PLN DJBB telah membina 2.460 Mitra Binaan Usaha Kecil dan Koperasi yang tersebar di seluruh Jawa Barat dan Banten, dengan total nilai Rp 15,71 milyar. Program Kemitraan ini berbentuk pinjaman lunak modal kerja dan investasi, pelatihan, pengembangan manajemen, promosi dan pemasaran.

2.8.2 Program Bina Lingkungan (Community Development)

Program ini sepenuhnya merupakan tanggungjawab moral dan sosial PLN DJBB dalam hal kepedulian sosial terhadap lingkungan dan masyarakatnya, sebagai wujud *corporate social responsibility* (CSR). Program ini diwujudkan dalam bentuk bantuan sosial untuk korban bencana alam, beasiswa untuk siswa / mahasiswa berprestasi tapi tidak mampu, bantuan untuk pembangunan fasilitas sosial-keagamaan, bantuan kepada anak cacat, pengobatan massal bagi keluarga kurang mampu, dan bentuk kepedulian sosial lain

2.9 LANDASAN TEORI

2.9.1 Pengertian Borland Delphi 7

Borland Delphi 7 Borland Delphi merupakan suatu bahasa pemrograman yang memberikan berbagai fasilitas pembuatan aplikasi untuk mengolah teks, grafik, angka, database dan aplikasi web. Program ini mempunyai kemampuan luas yang terletak pada produktifitas, kualitas, pengembangan perangkat lunak, kecepatan kompilasi, pola desain yang menarik serta bahasa pemrogramannya terstruktur dan lengkap. Fasilitas pemrograman

dibagi dalam dua kelompok yaitu object dan bahasa pemrograman. Object adalah suatu komponen yang mempunyai bentuk fisik dan biasanya dapat dilihat. Object biasanya dipakai untuk melakukan tugas tertentu dan mempunyai batasan-batasan tertentu.

Sedangkan bahasa pemrograman dapat disebut sekumpulan teks yang mempunyai arti tertentu dan disusun dengan aturan tertentu untuk menjalankan tugas tertentu. Gabungan antara object dengan bahasa pemrograman sering disebut bahasa pemrograman berorientasi object.

IDE Delphi Merupakan lingkungan pemrograman terpadu yang terdapat dalam Delphi.Dengan IDE semua yang diperlukan dalam pengembangan, dalam kondisi normal, semuanya telah tersedia. Adapun bagian-bagian IDE Delphi yang biasa ditampilkan yaitu :

1) Jendela Utama

   Di dalam jendela utama Delphi terdapat menu-menu sebagaimana menu aplikasi Windows umumnya, toolbar yang merupakan langkah cepat dari beberapa menu, dan component palette yaitu gudang komponen yang akan digunakan untuk membuat aplikasi

2) Object Treeview

   Fasilitas ini berguna untuk menampilkan daftar komponen yang digunakan dalam pengembangan aplikasi sesuai dengan penempatannya.

3) Object Inspector

   Object ini digunakan untuk mengatur properti dan event suatu komponen. Akan tetapi tidak dapat mengubah langsung properti-properti yang tidak ditampilkan kecuali melalui penulisan kode program.

4) Form Designer

   Form adalah komponen utama dalam pengembangan aplikasi. Form designer adalah tempat melekatnya komponen yang lain, dengan arti lain tempat komponen-komponen lain diletakkan.

5) Code Editor, Explorer dan Component Diagram

   Code Editor adalah tempat kode program yang diperlukan untuk mengatur tugas aplikasi ditulis. Code Explorer adalah fasilitas yang membantu penjelajahan kode program menjadi lebih mudah. Component Diagram adalah fasilitas yang dapat digunakan untuk membuat diagram komponen-komponen yang digunakan dalam aplikasi.

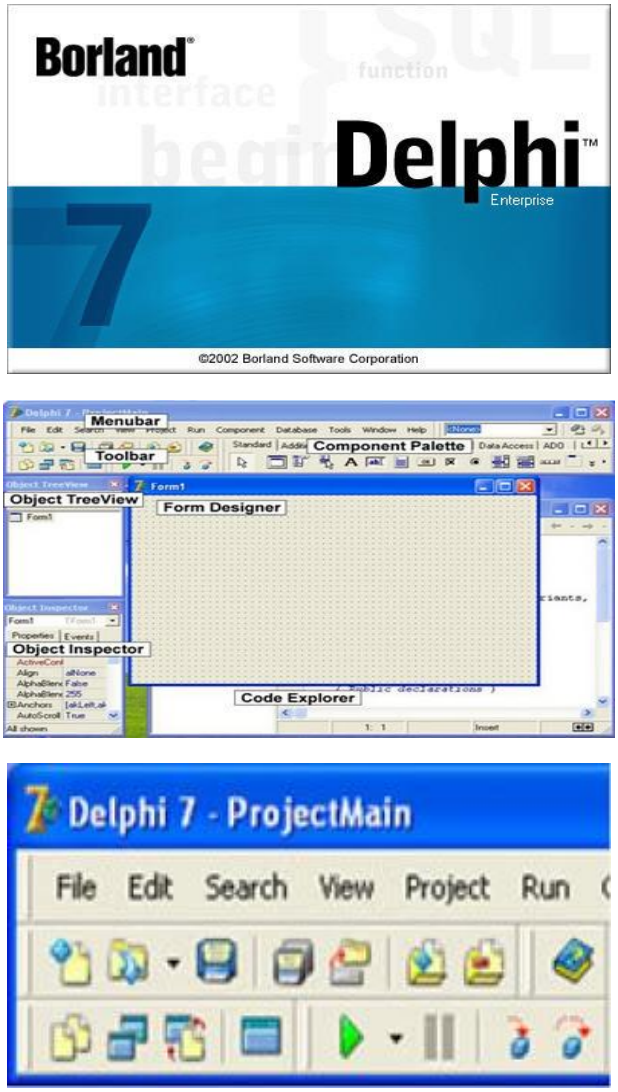
2.9.2 Kelebihan Borland Delphi 7

Berikut ini sebagian kecil dari banyak kelebihan Borland Delphi 7 :

Berbasis Object Oriented Programming (OOP). Setiap bagian yang ada pada programdipandang sebagai suatu object yang mempunyai sifat-sifat yang dapat diubah dan

diatur.Satu file .exe. Setelah program dirancang dalam IDE (Intergrated Development Environment) Delphi, Delphi akan mengkompilasinya menjadi sebuah file executable tunggal. Program yang dibuat dapat langsung didistribusikan dan dijalankan pada komputer lain tanpa perlu menyertakan file DLL dari luar.Ini merupakan sebuah kelebihan yang sangat berarti.Borland Delphi 7 hadir bersama Borland Kylix 3 yang berbasiskan Linux, sehingga memungkinkan programmer untuk membuat aplikasi multi-platform.

2.9.3 Tampilan Aplikasi Borlan Delphi 7



Gambar 2.3 Menubar dan Toolbar Delphi 7

Menubar dan Toolbar merupakan dua bagian yang biasanya terdapat pada aplikasi-aplikasi windows lain. Fungsi dari Menubar dan Toolbar ini relatif hampir sama dengan aplikasi window lain, sehingga secara explisit tidak akan dibahas dalam buku ini, melainkan hanya menubar yang biasa digunakan dalam pembuatan program.



Gambar 2.4 Component Palette Delphi 7

Component Palette berisi kumpulan VCL (Visual Component Library) yang berguna dalam desain aplikasi.VCL merupakan pustaka untuk komponen visual, dimana dalam component palette dilambangkan dengan ikon yang merepresentasikan komponen tersebut.

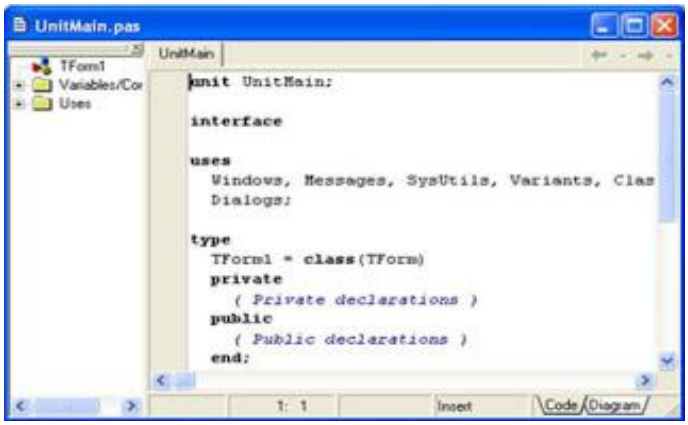
Komponen-komponen VCL pada component palette dikelompokkan ke dalam tab-tab, sesuai dengan fungsinya, dengan maksud untuk memudahkan programmer dalam memilih komponen yang diinginkannya.



Gambar 2.5 Form Designer Delpi 7

Setiap aplikasi biasanya memiliki jendela atau background interface, yang dalam bahasa pemrograman Delphi atau bahkan dalam bahasa pemrograman lain yang berbasis visual, biasa disebut dengan Form. Form Designer berfungsi sebagai tempat untuk mendesain form untuk aplikasi yang akan kita buat, dan juga sebagai tempat untuk meletakkan komponen-komponen yang kita ambil dari component palette.

Form sendiri sebenarnya merupakan komponen di mana property-nya bisa diatur melalui object inspector layaknya komponen-komponen lain. Form bisa jadi merupakan bagian terpenting dari sebuah aplikasi, namun meskipun begitu, beberapa aplikasi ada yang tidak membutuhkan form sama sekali, seperti aplikasi untuk memonitor ketukan keyboard.



Gambar 2.6 Code Explorer Delphi 7

Code Explorer merupakan area di mana kita menuliskan kode program, posisinya secara default terletak dibelakang form. Untuk menampilkan code explorer di depan form, Anda bisa menggunakan tombol F12 pada keyboard.

Pada code explorer tersebut, Anda akan melihat kode-kode dalam bahasa pemrograman delphi yang secara otomatis digenerate oleh Delphi, hal ini jelas akan memudahkan atau mempercepat kita dalam menulis program. Pada code explorer Delphi, ada

sebuah fitur yang disebut dengan code completion. Sesuai dengan namanya, code completion berfungsi melengkapi kode yang kita tulis dalam bentuk pilihan/list dari code-code yang bisa kita gunakan, hal ini akan sangat membantu apabila kita lupa terhadap kode tertentu.

Code completion ini secara otomatis akan muncul ketika Anda menekan tombol titik pada keyboard, selain itu untuk menampilkan code completion ini, Anda juga bisa melakukannya dengan menekan kombinasi tombol Ctrl + Space secara bersamaan.

Disamping code completion, code explorer pada delphi juga dilengkapi dengan hint (layer berwarna kuning yang muncul ketika mouse didekatkan pada komponen tertentu). Hint ini, muncul ketika kita menggunakan procedure atau fungsi dan menekan tombol (pada keyboard, pada hint tersebut akan muncul nama dan tipe data dari parameter yang digunakan atau informasi lain yang diperlukan.



Gambar 2.7 Object TreeView Delphi 7

Object TreeView adalah bagian yang berisi daftar komponen yang digunakan dalam designer. Dengan model tampilan bercabang, akan memudahkan kita dalam menunjuk komponen tertentu terutama jika aplikasi yang dibuat menggunakan banyak komponen.



Gambar 2.8 Object Inspector Delphi 7

Object Inspector adalah bagian yang digunakan untuk memanipulasi sifat atau karakteristik dan event dari komponen yang kita gunakan dalam form designer. Jendela object inspector terbagi menjadi dua bagian tab, yaitu tab property dan tab event. Tab property digunakan untuk memanipulasi properti yang dimiliki oleh komponen tertentu, misalkan ukuran, warna dan caption dari komponen. Sedangkan Tab Event digunakan untuk

menangani pemasukan kode pada kejadian tertentu dari suatu komponen, misalnya : kejadian ketika komponen button (tombol) diklik atau onClick.

Sebagaimana yang telah dijelaskan di atas, salah satu fungsi dari object inspector ini adalah untuk mengatur property dari komponen, pengaturan tersebut tidak akan disimpan pada file unit, melainkan akan disimpan pada file form.

2.9.4 Pengertian Database

Database adalah suatu kumpulan data-data yang disusun sedemikian rupa sehingga membentuk informasi yang sangat berguna. Database terbentuk dari sekelompok data-data yang memiliki jenis/sifat sama. Ambil contoh, data-data berupa nama-nama, kelas-kelas, alamat-alamat. Semua data tersebut dikumpulkan menjadi satu menjadi kelompok data baru, sebut saja sebagai data-data mahasiswa. Demikian juga, kumpulan dari data-data mahasiswa, data-data dosen, data-data keuangan dan lainnya dapat dikumpulkan lagi menjadi kelompok besar, misalkan data-data politeknik elektronika. Bahkan dalam perkembangannya, data-data tersebut dapat berbentuk berbagai macam data, misalkan dapat berupa program, lembaran-lembaran untuk entry (memasukkan) data, laporan-laporan. Kesemuanya itu dapat dikumpulkan menjadi satu yang disebut dengan database

2.9.5 Beberapa Jenis Database

Meskipun sebenarnya tujuan dari database tersebut sama, yaitu lebih mempermudah dalam pengolahan data, namun caranya ada berbagai macam. Macam dari database tersebut dapat dilihat dari bentuk konfigurasi sistemnya atau dari bentuk/isi dari database tersebut.

Ada beberapa jenis dari database, mulai dari yang menggunakan text biasa, menggunakan exel, lotus, foxpro, dbase, paradoc, access, oracle, SQL dan banyak lagi. Masing-masing dapat berbeda dari sisi format datanya, fasilitas yang disediakan dan teknik pengolah databasenya (database engine).

2.9.6 Bentuk Umum Database

Seperti pada uraian-uraian sebelumnya, database terdiri dari kumpulan sekelompok data, dan biasanya dinyatakan dalam bentuk tabel. Data-data tersebut tersimpan dalam suatu file.

2.9.7 Konfigurasi Database

Selain ada beberapa jenis perbedaan database dilihat dari file-file database itu sendiri, database juga dibedakan dari susunan/konfigurasi dari sistem database. Yang terbanyak dapat dibagi menjadi tiga bagian.

2.9.8 Pengertian Microsoft Access

Microsft Access adalah *program aplikasi pengolahan basis data yang merupakan bagian dari Microsoft Office*. Paket program ini merupakan sistem manajemen database relasional untuk Microsoft Windows yang memiliki kemampuan untuk menyortir, mengatur serta melaporkan informasi penting yang dibutuhkan. Dengan adanya versi yang terus diperbaharuhi ini Access semakin dilengkapi dengan fasilitas-fasilitas penting seperti interaksi dengan Internet. Database Access disimpan ke dalam sebuah file yang berekstensi **mdb**. Sebuah file database terdiri dari bagian-bagian yaitu :

- Tables : kumpulan data tentang subjek/topik tertentu yang disusun dalam bentuk kolom dan baris yang merupakan komponen utama dari sebuah database. Pada bidang perlombaan siswa misalnya : data lomba, informasi peserta, daftar perlombaan, jadwal lomba.
- Queries : digunakan untuk mencari dan menampilkan data yg memenuhi syarat tertentu dari satu tabel / lebih
- Forms : digunakan untuk menampilkan data, mengisi data dan mengubah data yg ada dalam tabel.
- Reports : digunakn untuk menampilkan laporan hasil analisis data.
- Pages : digunakan untuk membuat halaman web
- Macros : mengotomatisasi perintah2 yg sering digunakan dalam mengolah data.
- Modulus : digunakan untuk perancangan berbagai modul aplikasi pengolahan database.

2.9.9 Tipe Data Pada Access

- **Text**

  Pada tipe data ini jenis data yang disimpan atau diterima dapat berupa teks, angka, spasi dan tanda baca. Panjang maksimal *type field* adalah 255 karakter yang merupakan *type default.*

- **Memo**

  Pada tipe data ini, jenis data yang disimpan adalah karakter. Panjang maksimal *type field* adalah 65.535 karakter.

- **Number**

  Merupakan tipe data yang digunakan untuk menampung *type data* angka.

- **Date/time**

  Jenis data yang disimpan adalah data tanggal dan waktu dengan besar memory 8 *byte*.

- **Currency**

  Merupakan tipe data yang digunakan untuk menyimpan angka dalam format mata uang. Besarnya memori penyimpanan adalah 4 byte.

- **Auto Number**

  Tipe data ini digunakan untuk memberikan penomoran secara otomatis (penambahan angka otomatis)

- **Yes/No**

  Tipe data ini berisikan data Yes atau No, Benar atau Salah, Ya atau Tidak.

- **OLE Object**

  Tipe data ini dapat memuat gambar, grafis, video dan suara dengan ukuran maksimal 1 GB (batas atas Harddisk)

- **Hyperlink**

  Tipe data yang berisikan alamat hyperlink URL dengan panjang maksimal 64.000 karakter.

- **Lookup Wizard**

  Tipe data yang digunakan untuk menampilkan data dari tabel lain. Besar memori penyimpanan umumnya 4 byte.

2.9.10 Beberapa Istilah Yang Terdapat Dalam Pengoperasian Access

- **DATABASE**

  Merupakan kumpulan file-file yang saling berelasi sehingga membentuk bangunan data untuk menginformasikan data tertentu, relasi tersebut biasa ditunjukkan dengan kunci dari tiap file yang ada.

- **FILE**

  Adalah kumpulan record sejenis yang mempunyai panjang elemen yang sama, attribute yang sama, namun berbeda data isi atau valuenya.

- **FIELD / ENTITY**

  Fields merupakan tempat dimana data atau informasi dalam kelompok yang sama atau sejenis dimasukkan. Field pada umumnya tersimpan dalam bentuk kolom dari sebuah tabel. Pada bidang perlombaan Misalnya field meliputi kode lomba, jenis lomba, no registrasi, kode lomba, tgl registrasi dst

- **RECORD**

  Kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu entity secara lengkap. Satu record mewakili satu data atau informasi tentang seseorang misalnya, nomor induk siswa, nama siswa, kelas

2.9.11 Prosedur Pengoperasian Aplikasi Microsoft Access

Untuk memulai microsoft Access pada dasarnya hampir sama antara Task bar Windows 95 – 2000 yaitu dimulai dengan tombol **Start – Program – Pilih- Program MS-Access.**

Sebagai ilustrasi dapat dilihat pada gambar berikut :



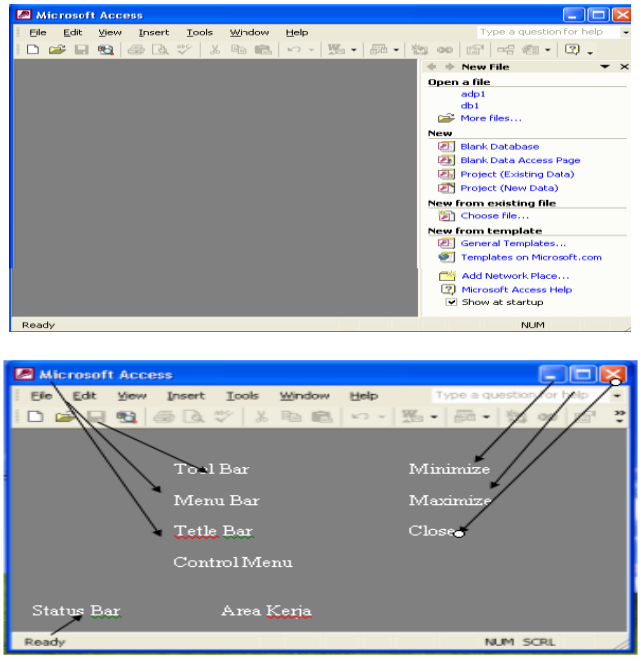
Gambar 2.9 Tampilan untuk memilih Program Ms-Access.

2.9.12 Lembar Kerja Ms- Access

Pada saat anda pertama kali menjalankan microsoft Access maka akan tampil kotak dialog dimana memungkinkan Anda dapat membuat data base baru dengan tiga cara :

1) Blank Access Database
2) Access database Wizards. Pages end Projects dan adalagi pilihan yang disediakan
3) Open Existing File.

Jika Anda tidak menginginkan tampilan Kotak Dialog pada saat pertama kali dijalankan program MS Access Anda dapat memilih perintah option pada menu tools dan matikan Check Box, Starup Dialog, pada tab Sheet View.



Gambar 2.10 Tampilan Program Ms–Access

Pilihan	Keterangan
Blank Access DataBase	Digunakan untuk membuat data base baru yg masih kosong dan Anda dapat menambahkan Obyek tabel, query, Form, Report, dan Obyel lainnya
Blank Data Access Page	Dipakai untuk membuat data base baru yang diberikan oleh MS Access secara bertahap
Project Existing Data	Diapaki untuk membuat data base baru berdasarkan tuntunan yang diberikan oleh microsoft access
Open end existing file	Dipakai untuk membuka file database yang telah dibuat sebelumnya

2.9.13 Pemodelan Terstruktur

Metode ini diperkenalkan pada tahun 1970, yang merupakan hasil turunan dari pemrograman terstruktur. Metode pengembangan dengan metode terstruktur ini terus diperbaiki sampai akhirnya dapat digunakan dalam dunia nyata.

Perancangan ini bertujuan untuk membuat model SOLUSI terhadap PROBLEM yang sudah dimodelkan secara lengkap pada tahap analisis terstruktur. Ada empat kegiatan perancangan yang harus dilakukan, yaitu:

- Perancangan arsitektural; kita merancang struktur modul P/L dengam mengacu pada model analisis yang sesuai (DFD). Langkahnya adalah: mengidentifikasi jenis aliran (transform flowatau transaction flow), menemukan batas-batas aliran (incoming flow dan outgoing flow), kemudian memetakannya menjadi striktur hirarki modul. Selanjutnya, kita alokasikan fungsi-fungsi yang harus ada pada modul-modul yang tepat.

- Perancangan data; kita merancang struktur data yang dibutuhkan, serta merancang skema basisdata dengan mengacu pada model analisis yang sesuai (ERD).

- Perancangan antarmuka; kita merancang antarmuka P/L dengan pengguna, antarmuka dengan sistem lain, dan antarmuka antar-modul.

- Perancangan prosedural; kita merancang detil dari setiap fungsi pada modul. Notasi yang digunakan bisa berupa flow chart, algoritma, dan lain-lain. Pastikan bahwa model perancangan yang dibuat sudah mengakomodasi kebutuhan non fungsional.

2.9.14 Kelebihan Pemodelan Tersruktur

- ✓ Milestone diperlihatkan dengan jelas yang memudahkan dalam manajemen proyek
- ✓ SSAD merupakan pendekatan visual, ini membuat metode ini mudah dimengerti oleh pengguna atau programmer.
- ✓ Penggunaan analisis grafis dan tool seperti DFD menjadikan SSAD menjadikan bagus untuk digunakan.
- ✓ SSAD merupakan metode yang diketahui secara umum pada berbagai industry.
- ✓ SSAD sudah diterapkan begitu lama sehingga metode ini sudah matang dan layak untuk digunakan.
- ✓ SSAD memungkinkan untuk melakukan validasi antara berbagai kebutuhan
- ✓ SSAD relatif simpel dan mudah dimengerti.

2.9.15 Kekurangan Pemodelan Tersruktur

- ✓ SSAD berorientasi utama pada proses, sehingga mengabaikan kebutuhan non-fungsional.
- ✓ Sedikit sekali manajemen langsung terkait dengan SSAD
- ✓ Prinsip dasar SSAD merupakan pengembangan non-iterative (waterfall), akan tetapi kebutuhan akan berubah pada setiap proses.
- ✓ Interaksi antara analisis atau pengguna tidak komprehensif, karena sistem telah didefinisikan dari awal, sehingga tidak adaptif terhadap perubahan (kebutuhan-kebutuhan baru).
- ✓ Selain dengan menggunakan desain logic dan DFD, tidak cukup tool yang digunakan untuk mengkomunikasikan dengan pengguna, sehingga sangat sulit bagi pengguna untuk melakukan evaluasi.
- ✓ Pada SAAD sulit sekali untuk memutuskan ketika ingin menghentikan dekomposisi dan mliai membuat sistem.
- ✓ SSAD tidak selalu memenuhi kebutuhan pengguna.
- ✓ SSAD tidak dapat memenuhi kebutuhan terkait bahasa pemrograman berorientasi obyek, karena metode ini memang didesain untuk mendukung bahasa pemrograman terstruktur, tidak berorientasi pada obyek (Jadalowen, 2002).

2.9.16 Pengertian Analisis Sistem Terstruktur

Definisi analisis sistem terstruktur adalah proses menguraikan sebuah program secara detail sehingga lebih mudah dipahami dan dapat di temukan sebuah solusi untuk memecahkan masalah suatu program.

Analisis Sistem Terstruktur adalah salah satu bentuk pendekatan formal yang pertama dalam sistem informasi. Analisis ini terfokus pada aliran data, proses bisnis dan perangkat lunak.

Analisis terstruktur adalah metode pemodelan klasik. Dimana analisis terstruktur ini merupakan aktifitas pembangunan model.

Ada beberapa pemodelan dalam analisis sistem terstruktur, diantaranya :

- Pemodelan Fungsional (DFD)

  Data Flow Diagram (DFD) adalah representasi grafik dari sebuah sistem. DFD menggambarkan komponen – komponen sebuah sistem, aliran-aliran data di mana komponen – komponen tersebut, dan asal, tujuan, dan penyimpanan dari data tersebut. DFD dapat digambarkan dalam Diagram Context dan Level n. Huruf n dapat menggambarkan level dan proses di setiap lingkaran.

- Pemodelan Data (ERD)

  ERD adalah suatu bentuk model yang menjelaskan hubungan antar data yang berada didalam suatu basis data berdasarkan hubungan relasi antar objek – objek dasar. ERD digunakan untuk memodelkan suatu struktur data dan menghubungkan antar data.

2.9.17 Ciri-Ciri Utama Yang Mendukung Pendekatan Terstruktur

Ciri-ciri utama yang mendukung pendekatan terstruktur :

- ✓ memanfaatkan alat-alat pemodelan
- ✓ menggunakan model untuk menjelaskan berbagai
- ✓ sistem, sub sistem untuk ditelaah dan dievaluasi
- ✓ oleh pelanggan dan pengembang (sebagai alat
- ✓ komunikasi, eksperimentasi atau prediksi)
- ✓ merancang berdasar modul
- ✓ modularisasi adalah proses yang membagi suatu
- ✓ sistem menjadi beberapa modul yang dapat
- ✓ beroperasi secara independent
- ✓ bekerja dengan pendekatan top-down
- ✓ dimulai dari level atas (secara global) kemudian

- ✓ diuraikan sampai ke tingkat modul (rinci)
- ✓ dilakukan secara iterasi
- ✓ dengan iterasi akan didapat hasil yang lebih baik, terlalu
- ✓ banyak iterasi juga akan menurunkan hasilnya dan
- ✓ menunjukkan bahwa tahap sebelumnya tidak dilakukan dengan baik
- ✓ kegiatan dilakukan secara paralel
- ✓ pengembangan subsistem-subsistem dapat dilakukan
- ✓ secara paralel, sehingga akan memperpendek waktu
- ✓ pengembangan sistem
- ✓ menggunakan CASE (Perangkat Lunak Pendukung Proses Pengembangan)
- ✓ dengan CASE (*computer aided software engineering*)
- ✓ memungkinkan analis dapat membangun sistem dan menghasilkan executable secara otomatis

2.9.18 Alat dan Teknik Pengembangan Sistem

Alat yang digunakan dalam suatu metodologi umumnya berupa gambar atau diagram atau grafik agar lebih mudah dimengerti. Selain berbentuk gambar, alat yang digunakan juga tidak berupa gambar misalnya kamus data, struktur inggris, pseudocode atau formulir formulir untuk mencatat atau menyajikan data

2.9.19 Alat-Alat Pengembangan Sistem Yang Berbentuk Diagram

- ✓ Diagram HIPO (Hierarchy plus Input-Proces-Output)
- ✓ Diagram aliran data (DFD/Data Flow diagram)
- ✓ Diagram keterhubungan entitas (ERD/Entity Relationship Diagram)
- ✓ Diagram Perubahan status (STD/State Transition Diagram)
- ✓ Structured Chart
- ✓ Diagram SADT (Structured Analysis and Design Techniques)
- ✓ Diagram Warnier/Orr
- ✓ Diagram Jakson’s
- ✓ Diagram UML

2.9.20 Tujuan Pemodelan Terstruktur

Tujuan pemodelan terstruktur adalah agar pada akhir pengembangan perangkat lunak dapat memenuhi kebutuhan user, dilakukan tepat waktu, tidak melampaui anggaran biaya, mudah dipergunakan, mudah dipahami dan mudah dirawat.

2.9.21 DFD

Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama Bubble chart, Bubble diagram, model proses, diagram alur kerja, atau model fungsi.

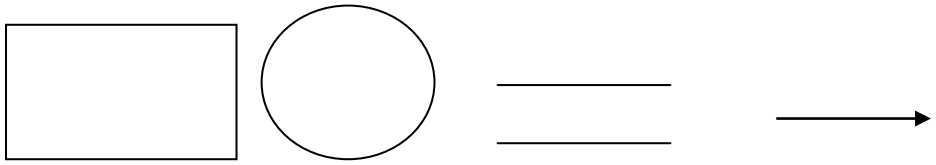
DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem.

DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

2.9.22 Komponen Data Flow Diagram

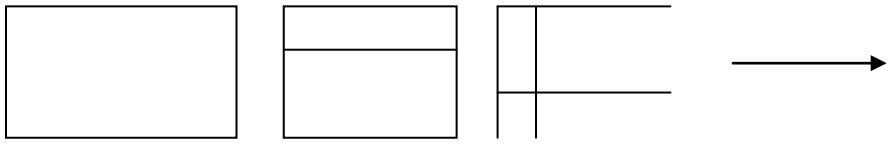
Menurut Yourdan dan DeMarco

.



Terminator **Proses** **Data Store** **Alur Data**

Menurut Gene dan Serson



Data Store **Alur Data**

2.9.23 Komponen Terminator / Entitas Luar

Terminator mewakili entitas eksternal yang berkomunikasi dengan sistem yang sedang dikembangkan. Biasanya terminator dikenal dengan nama entitas luar (*external entity*).

Terdapat dua jenis terminator :

1. Terminator Sumber (*source*) : merupakan terminator yang menjadi sumber.
2. Terminator Tujuan (*sink*) : merupakan terminator yang menjadi tujuan data / informasi sistem

Terminator dapat berupa orang, sekelompok orang, organisasi, departemen di dalam organisasi, atau perusahaan yang sama tetapi di luar kendali sistem yang sedang dibuat modelnya.

Terminator dapat juga berupa departemen, divisi atau sistem di luar sistem yang berkomunikasi dengan sistem yang sedang dikembangkan.

Komponen terminator ini perlu **diberi nama** sesuai dengan dunia luar yang berkomunikasi dengan sistem yang sedang dibuat modelnya, dan biasanya menggunakan **kata benda**, misalnya ***Bagian Penjualan, Dosen, Mahasiswa***.

Ada tiga hal penting yang harus diingat tentang terminator :

1) Terminator merupakan bagian/lingkungan luar sistem. Alur data yang menghubungkan terminator dengan berbagai proses sistem, menunjukkan hubungan sistem dengan dunia luar.

2) Profesinal sistem tidak dapat mengubah isi atau cara kerja, organisasi, atau prosedur yang berkaitan dengan terminator.

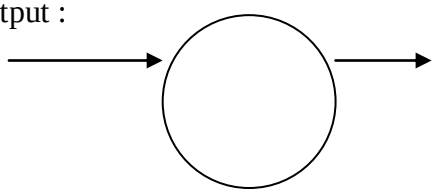
3) Hubungan yang ada antar terminator yang satu dengan yang lain tidak digambarkan pada DFD.

2.9.24 Komponen Proses

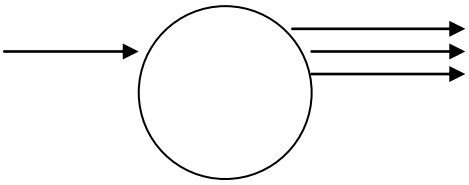
Komponen proses menggambarkan bagian dari sistem yang mentransformasikan input menjadi output.

Proses diberi nama untuk menjelaskan proses/kegiatan apa yang sedang/akan dilaksanakan. Pemberian nama proses dilakukan dengan menggunakan **kata kerja transitif** (kata kerja yang membutuhkan obyek), seperti ***Menghitung Gaji, Mencetak KRS, Menghitung Jumlah SKS***.

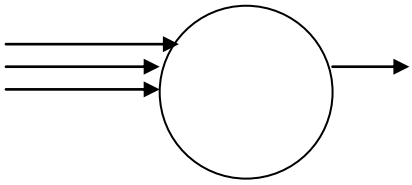
Ada empat kemungkinan yang dapat terjadi dalam proses sehubungan dengan input dan output :



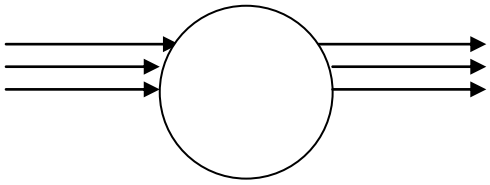
1 Input & 1 Output



1 Input & Banyak Output



Banyak Input & 1 Output



Banyak Input & Banyak Output

Ada beberapa hal yang perlu diperhatikan tentang proses :

- Proses harus memiliki input dan output.
- Proses dapat dihubungkan dengan komponen terminator, data store atau proses melalui alur data.

- Sistem/bagian/divisi/departemen yang sedang dianalisis oleh profesional sistem digambarkan dengan komponen proses.

2.9.25 Komponen Data Store

Komponen ini digunakan untuk membuat model sekumpulan paket data dan **diberi nama** dengan **kata benda jamak**, misalnya ***Mahasiswa***.

Data store ini biasanya berkaitan dengan penyimpanan-penyimpanan, seperti file atau database yang berkaitan dengan penyimpanan secara komputerisasi, misalnya file disket, file harddisk, file pita magnetik. Data store juga berkaitan dengan penyimpanan secara manual seperti buku alamat, file folder, dan agenda.

Suatu data store dihubungkan dengan alur data **hanya pada komponen proses**, tidak dengan komponen DFD lainnya. Alur data yang menghubungkan data store dengan suatu proses mempunyai pengertian sebagai berikut :

- **Alur data dari data store** yang berarti sebagai pembacaan atau pengaksesan satu paket tunggal data, lebih dari satu paket data, sebagian dari satu paket tunggal data, atau sebagian dari lebih dari satu paket data untuk suatu proses

- **Alur data ke data store** yang berarti sebagai pengupdatean data, seperti menambah satu paket data baru atau lebih, menghapus satu paket atau lebih, atau mengubah/memodifikasi satu paket data atau lebih

2.9.26 Bentuk Data Flow Diagram

Terdapat dua bentuk DFD, yaitu **Diagram Alur Data Fisik**, dan **Diagram Alur data Logika**. Diagram alur data fisik lebih menekankan pada bagaimana proses dari sistem diterapkan, sedangkan diagram alur data logika lebih menekankan proses-proses apa yang terdapat di sistem.

1. **Diagram Alur Data Fisik (DADF)**

   DADF lebih tepat digunakan untuk menggambarkan sistem yang ada (sistem yang lama). Penekanan dari DADF adalah bagaimana proses-proses dari sistem diterapkan (dengan cara apa, oleh siapa dan dimana), termasuk proses-proses manual.

   Untuk memperoleh gambaran bagaimana sistem yang ada diterapkan, DADF harus memuat :

   - Proses-proses manual juga digambarkan.

   - Nama dari alur data harus memuat keterangan yang cukup terinci untuk menunjukkan bagaimana pemakai sistem memahami kerja sistem.

   - Simpanan data dapat menunjukkan simpanan non komputer.

   - Nama dari simpanan data harus menunjukkan tipe penerapannya apakah secara manual atau komputerisasi. Secara manual misalnya dapat menunjukkan buku catatat, meja pekerja. Sedang cara komputerisasi misalnya menunjukkan file urut, file database.

- Proses harus menunjukkan nama dari pemroses, yaitu orang, departemen, sistem komputer, atau nama program komputer yang mengakses proses tersebut.

2. Diagram Alur Data Logika (DADL)

DADL lebih tepat digunakan untuk menggambarkan sistem yang akan diusulkan (sistem yang baru). Untuk sistem komputerisasi, penggambaran DADL hanya menunjukkan kebutuhan proses dari sistem yang diusulkan secara logika, biasanya proses-proses yang digambarkan hanya merupakan proses-proses secara komputer saja.

2.9.27 Syarat-Syarat Pembuatan Data Flow Diagram

Syarat pembuatan DFD ini akan menolong profesional sistem untuk menghindari pembentukkan DFD yang salah atau DFD yang tidak lengkap atau tidak konsisten secara logika. Beberapa syarat pembutan DFD dapat menolong profesional sistem untuk membentuk DFD yang benar, menyenangkan untuk dilihat dan mudah dibaca oleh pemakai.

Syarat-syarat pembuatan DFD ini adalah :

- Pemberian nama untuk tiap komponen DFD
- Pemberian nomor pada komponen proses
- Penggambaran DFD sesering mungkin agar enak dilihat
- Penghindaran penggambaran DFD yang rumit
- Pemastian DFD yang dibentuk itu konsiten secara logika

2.9.28 Penggambaran Dfd

Tidak ada aturan baku untuk menggambarkan DFD. Tapi dari berbagai referensi yang ada, secara garis besar langkah untuk membuat DFD adalah :

1) Identifikasi terlebih dahulu semua entitas luar yang terlibat di sistem.

2) Identifikasi semua input dan output yang terlibat dengan entitas luar.

2.9.29 Pengertian Flowmap

Flowmap adalah penggambaran secara grafik darilangkah-langkah dan urut-urutanprosedur dari suatu program

2.9.30 Jenis-Jenis Flowmap

1) Flowmap Sistem
2) Flowmap Paperwork atau Flowmap Dokumen
3) Flowmap Skematik
4) Flowmap Program
5) Flowmap Proses

2.9.31 Aturan Membuat Flowmap

Untuk membuat sebuah analisis menggunakan flowmap seorang analis dan programer memerlukan beberapa tahapan, diantarnya :

1) Flowmap digambarkan dari halaman atas ke bawah dan dari kiri ke kanan.
2) Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat dimengerti oleh pembacanya.
3) Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas.
4) Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja, misalkan MENGHITUNG PAJAK PENJUALAN.
5) Setiap langkah dari aktivitas harus berada pada urutan yang benar.
6) Lingkup dan range dari aktifitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktivitas yang sedang digambarkan tidak perlu digambarkan pada flowchart yang sama. Simbol konektor harus digunakan dan percabangannya diletakan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berkaitan dengan system.
7) Gunakan simbol-simbol flowchart yang standar.