BAB II
LANDASAN TEORI

2.1 Pengertian Sistem

Pengertian sistem didefinisikan oleh **Jogiyanto Hartono** (1981) sebagai berikut :*"Sekumpulan subsistem-subsistem saling berinteraksi dan saling berhubungan membentuk satu kesatuan sehingga tujuan atau sasaran sistem tersebut dapat tercapai".*

Sebagai contoh, sistem komputer dapat terdiri dari subsistem perangkat keras dan subsistem perangkat lunak.

"Sistem adalah suatu kumpulan-kumpulan atau subsistem-subsistem yang dirancang dan disatukan untuk mencapai satu tujuan tertentu." (Burch and Strater. 1, hal II-5)

"Sistem diartikan sebagai suatu kesatuan yang terdiri atas koponen-komponen (subsistem-subsistem) yang saling berhubungan erat menurut rencana tertentu dalam upaya mencapai satu atau beberapa tujuan tertentu." (Award, System Analysis and Design. 1, hal II-5)

"Sistem adalah suatu susunan elemen-elemen yang berinteraksi dan membentuk satu kesatuan yang integral untuk mencapai tujuan bersama dengan mengoperasikan informasi atau energi." (Murdick and Ross. Information System for Modern Management. 10, halI-5)

Dari definisi-definisi tesebut diatas, maka dapat ditarik kesimpulan bahwa Sistem adalah suatu unit yang terdiri dari komponen-komponen yang bersifat

sederhana hingga yang bersifat kompleks dan memiliki sifat saling ketergantungan antara satu dengan yang lainnya sehingga komponen-komponen tersebut akan didefinisikan sebagai satu kesatuan, baik secara fungsional maupun secara logik.

2.1.1 Karakteristik Sistem

Menurut Jogiyanto HM dalam bukunya "*Analisis dan Disain*" sistem mempunyai karakteristik atau sifat-sifat tertentu yaitu mempunyai komponen-komponen (*components*), batas sistem (*boundary*), lingkungan luar sistem (*environments*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolahan (*process*), dan sasaran (*objectives*) atau tujuan (*goal*).

1. Komponen-komponen (*components*), suatu sistem terdiri dari sejumlah komponen yang saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa sub sistem atau bagian-bagian dari sistem.
2. Batas sistem (*boundary*), merupakan daerah yang membatasi antara suatu sistem dengan sistem lainnya.
3. Lingkungan luar sistem (*environments)*, dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem.
4. Penghubung (*interface)*, merupakan media penghubung antara suatu sub sistem dengan sub sistem lainnya.
5. Masukan (*input*), adalah energi yang dimasukan ke dalam sistem.
6. Keluaran (*output)*, adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

Keluaran dapat merupakan masukkan untuk sub sistem yag lain atau kepada supra sistem.

7. Pengolahan (*process*), suatu sistem dapat mempunyai suatu bagian pengolahan yang akan merubah masukan menjadi keluaran.
8. Sasaran (*objectives*) atau tujuan (*goal*), suatu sistem pasti mempunyai sasaran atau tujuan, jika sistem tidak mempunyai sasaran dan tujuan maka opersai tidak akan ada gunanya. Sasaran sistem sangat menentukan sekali masukan sistem dan keluaran yang dihasilkan sistem.

2.1.2 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari berbagai sudut pandang, menurut Jogiyanto HM klasifikasi tersebut diantaranya adalah sebagai berikut :

1. Sistem sebagai sistem abstrak (*abstrac system*) dan sistem fisik (*Physical system*). Sistem abstrak adalah sistem yang berupa pemikiran-pemikiran atau ide-ide yang tidak tampak secara fisik. Sedangkan sistem fisik adalah sistem yang ada secara fisik.
2. Sistem sebagai sistem ilmiah (*natural system*) dan sistem buatan manusia (*human made system*). Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Sedangkan sistem buatan manusia adalah sistem yang dirancang dan diciptakan oleh manusia.
3. Sistem sebagai sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*). Sistem tertentu adalah sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi diantara bagian-bagiannya sehingga bentuk keluaran dapat diramalkan. Sedangkan

sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

4. Sistem sebagai sistem tertutup (*closed system*) dan sistem terbuka (*open system*). Sistem tertutup merupakan sistem yang tidak terpengaruh dan tidak berhubungan dengan lingkungan luarnya. Sistem ini secara otomatis berjalan tanpa ada campur tangan dari pihak luar. Sedangkan sistem terbuka adalah sistem yang berhubungan dan terpengaruh oleh lingkungan luarnya. Sistem ni menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem-subsistem lainnya.

2.2 Pengertian Informasi

Informasi Didefinisikan oleh **Jugiyanto Hartono** adalah sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian (*event*) yang nyata (*fact*) yang digunakan untuk pengambilan keputusan.

Informasi *(information*) adalah hasil dari kegiatan pengolahan data yang memberikan bentuk yang lebih berarti dari suatu kejadian. Informasi merupakan hasil suatu proses. Proses itu sendiri terdiri dari kegiatan-kegiatan mulai dari pengumpulan data, menyusun serta menghubungkan, meringkas, mengambil intisarinya dan menginterpretasikannya sesuai dengan persepsi sistem informasi penerima agar menjadi sebuah informasi yang berguna.

Menurut *Gordon B.Davis*, definisi informasi sebagai berikut:

> *"Informasi data yang telah diolah menjadi suatu bentuk yang penting bagi si penerima dan mempunyai nilai yang nyata atau yang dapat dirasakan dalam keputusan-keputusa yang sekarang atau keputusan-keputusan yang akan datang".*

Menurut *Lucas*, definisi informasi sebagai berikut:

> *"Informasi adalah suatu kesatuan yang tampak ataupun yang tidak tampak yang fungsinya untuk mengurangi ketidakpastian atau peristiwa di masa yang akan datang".*

Informasi yang baik haruslah mencakup hal-hal berikut:

a. *Relevancy*

   Berarti informasi harus memberikan manfaat bagi pemakainya. Relevansi informasi untuk tiap-tiap orang yang satu dengan yang lainnya berbeda, misalnya informasi mengenai sebab-musabab kerusakan mesin produksi kepada akuntan perusahaan adalah kurang relevan dan akan lebih relevan kepada ahli teknik perusahaan.

b. *Accurasi*

   Informasi harus bebas dari kesalahan-kesalahan, tidak menyesatkan, dan harus jelas mencerminkan maksudnya. Ketidak akuratan dapat terjadi karena sumber informasi (data) mengalami gangguan atau kesengajaan sehingga merusak/merubah data-data asli tersebut.

c. *Timelines*

Informasi yang dihasilkan/dibutuhkan haruslah tepat waktu, tidak boleh terlambat. Informasi yang tidak tepat waktu akan berakibat fatal dalam keputusan dan tindakan apabila digunakan sebagai dasar untuk mengambil keputusan.

d. *Efficiency*

Informasi yang diberikan haruslah efisien.

e. *Reliability*

Informasi tersebut harus dapat dipercaya.

2.3 Pengertian Sistem Informasi

Semua organisasi pasti memiliki sistem informasi. Sistem informasi adalah *entity* (kesatuan) formal yang terdiri dari berbagai sumberdaya fisik maupun logik. Dari organisasi ke organisasi, sumberdaya-sumberdaya ini disusun atau distrukturkan dengan beberapa cara yang berlainan karena organisasi dan sistem informasi merupakan sumberdaya-sumberdaya yang bersifat dinamis. Menurut sumber yang mendefinisikan sistem informasi.

"Sistem Informasi adalah suatu sistem manusia-mesin yang terpadu untuk menyajikan informasi guna mendukung fungsi operasi, manajemen dan pengambilan keputusan dalam organisasi". [Budihar01]

Menurut Jeffrey L. Whitten, Lonnie D. Bentley, Kevin C. Dittman, sistem informasi adalah :

“ Pengaturan orang, data, proses dan teknologi informasi yang berinteraksi untuk mengumpulkan, memproses, menyimpan, dan menyediakan sebagai output informasi yang diperlukan untuk mendukung sebuah organisasi ”.

Sistem Informasi dapat didefinisikan sebagai suatu sistem di dalam organisasi yang merupakan kombinasi orang-orang, fasilitas, teknologi, media, prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi yang penting, memproses tipe transaksi dan memberi sinyal kepada manajemen serta yang lainnya yang di dalamnya terdapat kejadian-kejadian eksternal dan internal yang penting serta menyediakan suatu dasar informasi untuk pengambilan keputusan.

Sistem informasi dapat terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*), sebagai suatu sistem blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya.

1. Blok Masukan

   Input mewakili data yang masuk ke dalam sistem informasi. Input diisi termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan yang dapat berupa dokumen-dokumen dasar.

2. Blok Model

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

3. Blok Keluaran

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

4. Blok Teknologi

   Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem keseluruhan. Teknologi terdiri dari 3 bagian utama yaitu teknisi (*humanware* atau *brainware*), perangkat lunak (*software*) dan perangkat keras (*hardware*).

5. Blok Dasar Data

   Dasar data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

6. Blok Kendali

   Supaya sistem informasi dapat berjalan sesuai dengan yang diinginkan, maka perlu diterapkan pengendalian-pengendalian di dalamnya.

Kegiatan yang terdapat di dalam sistem informasi mencakup antara lain :

1. Input, menggambarkan suatu kegiatan untuk menyediakan data untuk diproses.
2. Proses, menggambarkan bagaimana suatu data diproses untuk menghasilkan suatu informasi yang bernilai tambah.

3. Output, suatu kegiatan untuk menghasilkan laporan dari proses di atas tersebut.
4. Penyimpanan, suatu kegiatan untuk memelihara dan menyimpan data.
5. Kontrol, suatu aktivitas untuk menjamin bahwa sistem informasi tersebut berjalan sesuai dengan yang diharapkan.

2.4 Pengertian Data

Data adalah suatu objek dari perilaku kejadian yang berupa keterangan atau *file* yang menerangkan sesuatu. Dapat dibagi atas beberapa, sebagai berikut:

a. Pengolahan Data.

Pengolahan data dapat diartikan sebagai suatu proses untuk menghasilkan bentuk yang lebih berguna dan berarti dari suatu kejadian berupa informasi yang diperlukan.

b. Siklus Pengolahan Data

Suatu proses pengolahan data terdiri dari 3 tahapan dasar, yang disebut dengan siklus pengolahan data (*Data Processing Cycle*). Yaitu *input*, *Processing* dan *output*.(**Jogiyanto Hartono**)



Gambar 2.1 Siklus pengolahan data

c. Sistem Pengolahan Data

Sistem pengolahan data adalah suatu kesatuan yang saling berhubungan yang terdiri dari peralatan, tenaga pelaksana, dan prosedur-prosedur yang akan menghasilkan suatu informasi.

Langkah-langkah Pengolahan Data

Proses pengolahan data biasanya terdiri dari sejumlah operasi pengolahan data yang dilaksanakan dalam beberapa langkah sebagai berikut :

1. Pencatatan

   Prosedur pengolahan data dari suatu kejadian dituliskan ke dalam bentuk formulir atau dokumen yang berguna untuk masukkan dalam pengolahan data.

2. Pemeriksaan Data

   Proses pemeriksaan terhadap data masukkan yang bertujuan agar data yang di dalam dokumen dapat di pindahkan ke dalam komputer secara tepat.

3. Pengelompokkan Data

   Proses pengelompokkan data berdasarkan kategori tertentu dan bertujuan untuk memudahkan atau memberi arti tertentu bagi pengolahan data

4. Penyusunan atau pemilahan

   Proses penyusunan data berdasarkan peringkat tertentu dengan memberikan item penyortirannya yang disebut *key* (kunci).

5. *Merging*

   Dalam proses merging dilakukan pencampuran dua atau lebih kumpulan data, semua kumpulan tersebut diurutkan berdasarkan *key* yang sama, dan meletakkan kumpulan tersebut bersama-sama menjadi bentuk kumpulan data tunggal yang telah diurutkan.

6. Perhitungan.

   Dalam pengolahan data sering diperlukan perhitungan tertentu terhadap data, agar dapat diperoleh hasil akhir yang diperlukan sebagai laporan.

7. Penyimpanan Data

   Proses ini dilakukan untuk menempatkan data ke dalam suatu media penyimpanan, misalnya kertas, *disk* dan sebagainya. Penyimpanan data ini berguna untuk memudahkan dalam pencarian data yang sebelumnya apabila diperlukan.

2.5 Sistem Basis Data

Basis data dapat diartikan sebagai kumpulan data tentang suatu benda atau kejadian yang saling berhubungan satu sama lain. Sedangkan data merupakan fakta yang mewakili suatu obyek seperti manusia, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang dapat dicatat dan mempunyai arti yang implisit. Data dicatat atau direkam dalam bentuk angka, huruf, simbol, gambar, bunyi, atau kombinasinya. Sebagai contoh, terdapat daftar nama, nomor telepon, dan alamat orang-orang yang menjadi anggota suatu organisasi. [Waljiyanto00]

Konsep mengenai basis data dapat dipandang dari beberapa sudut. Dari sisi sistem, basis data merupakan kumpulan tabel-tabel atau files yang saling berelasi. Sementara dari sisi manajemen, basis data dapat dipandang sebagai kumpulan data yang memodelkan aktivitas-aktivitas yang terdapat di dalam *enterprise*-nya. Selain itu, basis data juga mengandung pengertian kumpulan data *non-redundant* yang dapat digunakan bersama (share) oleh sistem-sistem aplikasi yang berbeda.

Penggunaan istilah basis data menurut Elmasri R. (1994) lebih dibatasi pada arti implisit yang khusus, yaitu :

1. Basis data merupakan penyajian suatu aspek dari dunia nyata (*"real word"* atau *"miniword"*).
2. Basis data merupakan kumpulan data dari berbagai sumber yang secara logika mempunyai arti implisit. Sehingga data yang terkumpuk secara acak dan tanpa mempunyai arti, tidak dapat disebut basis data.
3. Basis data perlu dirancang, dibangun, dan data dikumpulkan dalam suatu tujuan. Basis data dapat digunakan oleh beberapa pemakai dan beberapa aplikasi sesuai dengan kepentingan pemakai.

Dari keterangan diatas dapat disimpulkan bahwa basis data mempunyai berbagai sumber data dalam pengumpulan data, bervariasi derajad interaksi kejadian dari dunia nyata, dirancang.

Berikut adalah beberapa pengertian atau definisi lain dari basis data yang dikembangkan atau dasar sudut pandang yang berbeda dan diambil dari pustaka [Fathan02] :

1. Himpunan kelompok data (file/arsip) yang saling berhubungan dan saling berorganisasi sedemikian rupa agar dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan dan disimpan bersama sedemikian rupa tanpa pengulangan yang tidak perlu (redudancy) untuk memenuhi berbagai kebutuhan.
3. Kumpulan file/arsip/tabel yang saling berhubungan yang disimpan di dalam media penyimpanan elektronik.

Sistem basis data merupakan perpaduan antara basis data dan sistem manajemen basis data *(DBMS)* Komponen-komponen sistem basis data meliputi :

1. Perangkat Keras *(Hardware)* sebagai pendukung operasi pengolahan data.
2. Sistem Operasi *(Operating System)* atau perangkat lunak untuk mengelola basis data.
3. Basis data *(Database)* sebagai inti dari sistem basis data.
4. Sistem Manajemen Basis Data (*DBMS*).
5. Pemakai *(User).*
6. Aplikasi lain.

Sistem manajemen basisdata (*DBMS*) adalah suatu program komputer yang memasukkan, mengubah, menghapus, memanipulasi, dan memperoleh data dan informasi dengan praktis dan efisien. *DBMS* merupakan suatu paket perangkat lunak (tanpa basisdata) *general purpose* yang digunakan untuk membangun sistem basisdata tertentu.

DBMS akan menentukan bagaimana data diorganisasikan, disimpan, diubah dan dipanggil. DBMS juga yang menerapkan mekanisme pengamanan data, penggunaan data bersama *(data sharing),* konsistensi data dan sebagainya. Perangkat lunak yang termasuk DBMS adalah keluarga *DBASE* (II, III+, IV, dll) atau *XBase, FoxBase, FoxPro, Paradox, Clipper, Access, Oracle, Sybase, DB2,* dan sebagainya.

Didalam DBMS terdapat beberapa model basisdata yang digunakan. Model ini menyatakan hubungan antara *record-record* yang ada dalam basisdatanya. Model basisdata tersebut adalah :

1. *Flat file* (tabular) -- Data terletak dalam tabel tunggal (tidak terdapat kaitan antara tabel yang satu dengan table-tabel lainnya).
2. *Hierarchical* -- Pada model ini data diorganisasikan kedalam struktur hubungan hirarki bapak-anak (*Parent-Child*) sebagai satu *tree*. Setiap *parent* dapat memiliki *child* lebih dari satu, sementara setiap *child* hanya memiliki satu *parent*.
3. *Network* -- Model ini sangat mirip dengan model *hierarchical*, tetapi pada model ini setiap *child* dapat memiliki lebih dari satu *parent.*
4. *Relational* – Model ini terdiri dari tabel-tabel data (data direpresentasikan dalam tabel yang terdiri dari baris-baris dan kolom-kolom) ternormalisasi dengan *field-field* kunci sebagai penghubung relasional antar tabel.

2.6 Konsep Dasar Rekayasa Perangkat Lunak

Komputer merupakan mesin yang memproses fakta atau data menjadi informasi. Komputer digunakan orang untuk meningkatkan hasil kerja dan memecahkan berbagai masalah. Yang menjadi pemroses data atau pemecah masalah itu adalah peragkat lunak.

Bentuk terkecil dari perangkat lunak adalah aritmatik dan logika.dari operasi dasar ini di susun program atau perangkat lunak. Tingkat pemrosesan yang di kerjakan perangkat lunak pun dari *machine-like*, mulai berubah menjadi *human-like*. Dalam teori informasi, di susun hirarki informasi, mulai dari data/fakta, kemudian setelah proses seleksi dan pengurutan menjadi sesuatu yang berguna menjadi informasi. Informasi yang di susun secara sistematis dengan suatu alur logika tertentu menjadi *knowledge*. Dan pada akhirnya gabungan *knowledge* yang digabung dari berbagai sisi guna membangun *windows*.

Data yang di proses pun telah banyak berubah, yang semula hanya berupa data bilangan dan karakter merambah ke *audio-visual* (bunyi, suara, gambar, film). Sejauh perkembangan hingga saat ini, seluruh proses menggunakan format data digital dengan satuan bit (*binary digit*).

Pemilih perangkat lunak ditentukan oleh kebutuhan pemakai. Perangkat lunak yang berkualitas bukanlah perangkat lunak yang canggih dengan versi terbaru. Perangkat lunak yang berkualitas adalah perangkat lunak yang memenuhi dan sesuai kebutuhan pemakai.

Dalam pengembangan perangkat lunak terdapat proses di dalam pengembanganya, mungkin kita bertanya apakah sebenarnya *perangkat lunak* itu? *Dalam bukunya, Roger S. Pressman, Ph.D mendefinisikan sebagai berikut :*

"(1) perintah (proses computer) yang bila dieksekusi memberikan fungsi dan unjuk kerja yang diinginkan, (2) struktur data yang memungkinkan program memanipulasi informasi secara propesional, dan (3) Dokumen yang menggambarkan operasi dan keggunaan program. Proses perangkat lunak adalah sebuah kerangka kerja untuk tugas – tugas yang dibutuhkan untuk membangun perangkat lunak dengan kualitas yang tinggi"

2.6.1 Rekayasa Perangkat lunak

Definisi perangkat lunak yang diusulkan oleh Fritz Bauer adalah :

"Rekayasa perangkat lunak adalah pengembangan dan penggunaan prinsip pengembangan untuk memperoleh perangkat lunak secara ekonomis yang realiabel dan bekerja secara efisien pada mesin nyata."

International (IEEE) telah mengembangkan definisi yang lebih komperehensif sebagai berikut :

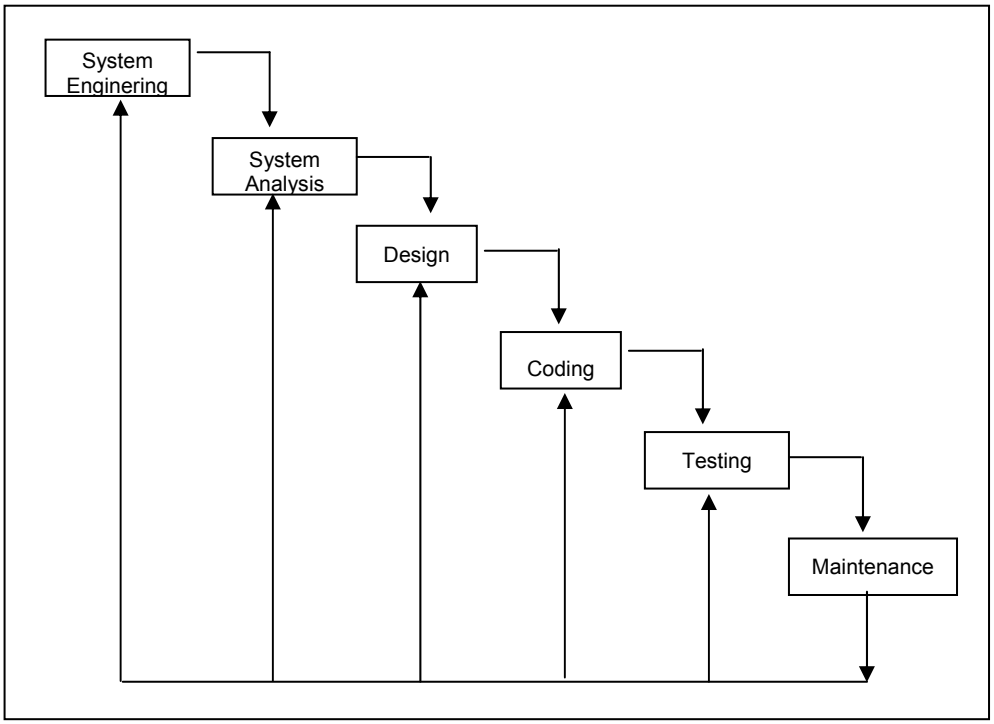
"Rekayasa perangkat lunak : (1) Aplikasi dari sebuah Pendekatan kuantifiabel, disiplin, dan sistematis kepada pengembangan, operasi, dan pemeliharaan perangkat lunak; yaitu aplikasi dari rekayasa perangkat lunak.(2) Studi tentang Pendekatan-pendekatan seperti pada butir "

2.6.2 Model Proses Rekayasa Perangkat Lunak

Model proses untuk rekayasa perangkat lunak dipilih berdasarkan sifat aplikasi dan proyeknya, metode dan alat-alat Bantu yang akan dipakai, dan control serta penyampaian yang dibutuhkan. Model-model proses tersebut adalah :

1. Model Sekuensial Linier atau model air terjun .
2. Model prototype.
3. Model RAD (*Rapid Application Development*).
4. Model Pertambahan.
5. Model Spiral.
6. Model Rakitan Komponen.
7. Model Perkembangan Konkuren.
8. Model Formal.
9. Teknik Generasi Keempat.

Dalam menyelesaikan tugas akhir ini digunakan metode paradigma *waterfall* (*classic life cycle*). Metode ini mudah digunakan dan jika terjadi kesalahan tidak perlu diperbaiki dari awal, adapun diagram dari paradigma *waterfall* adalah sebagai berikut :



Gambar 2.2 Proses Waterfall *(classic life cycle)*

Model sekuensial linier yang sering disebut juga dengan *classic life cycle* atau *waterfall* mempunyai beberapa tahap, sebagai berikut :

1. Rekayasa sistem *(System Engineering)*

Sebelum sistem dikembangkan, terlebih dahulu melakukan pengumpulan data yang berhubungan dengan pembuatan sistem. Tahapan pertama ini ditujukan untuk memperoleh data mengenai permasalahan yang dihadapi. Teknik pengumpulan data yang digunakan adalah : Studi pustaka, wawancara dan studi dokumentasi.

2. Analisa Sistem *(System Analysis)*

Merinci lebih jauh mengenai sistem yang dikembangkan dengan menggunakan DAD (Diagram Alir Dokumen) beserta flowmap. Sehingga diperoleh spesifikasi dari sistem yang akan dibangun.

3. Perancangan *(Design)*

Mempelajari terlebih dahulu sistem aplikasi yang akan dipakai, kemudian menentukan kerangka dasar serta komponen penunjang lainnya dengan pembuatan DFD (Data Flow Diagram) dan juga diagram hubungan antar entitas (Entity Relationship-Diagram) yang memudahkan desain implementasi sistem.

4. Pengkodean *(Coding)*

Adalah kegiatan penulisan kode program yang akan dieksekusi oleh computer berdasarkan hasil dari desain rinci yang akan menghasilkan program yang sesuai dengan kebutuhan pemakai. Adapun software developer yang dipakai untuk pembuatan kode adalah Borland Delphi 7 Enterprise Edition.

5. Pengujian *(Testing)*

Adalah tahapan-tahapan pengujian antar modul dan pengujian integrasi, untuk mengantisipasi kesalahan yang mungkin terjadi ketika aplikasi siap dilepas ke pasar, agar mendapatkan hasil yang diinginkan. Metoda yang digunakan dalam testing ini adalah alpha dan beta-test, artinya pengujian dilakukan oleh pembuat program dan user yang bersangkutan.

6. Implementasi dan Pemeliharaan *(Implementation and Maintenance)*

Adalah tahapan terakhir dimana perangkat lunak yang sudah selesai dan dapat mengalami perubahan-perubahan dan penambahan sesuai dengan kebutuhan.

2.6.3 Data Flow Diagram (DFD)

Keuntungan dari DFD ialah memungkinkan untuk menggambarkan sistem dari level yang paling tinggi dan diturunkan menjadi level yang lebih rendah (dekomposisi), sedangkan kekurangan DFD ialah tidak menunjukan prooses perulangan (*looping*), proses keputusan dan proses perhitungan.

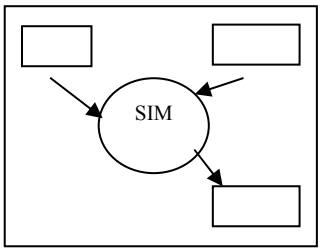
2.6.4 Level DFD

DFD dapat digambarkan dalam diagram konteks dan level n. huruf n dapat menggambarkan level dan proses di setiap lingkaran.

1. Diagram Konteks

   Diagram konteks menggambarkan sistem dalam satu lingkaran dan hubungan dengan entitas luar. Lingkaran tersebut menggambarkan keseluruhan proses dalam sistem tersebut.

   Beberapa hal yang harus diperhatikan dalam menggambar diagram konteks ialah: jangan memberikan nomor pada diagram konteks, hanya menggunakan satu lingkaran tersebut sesuai dengan fungsi sistem tersebut, gambarkan arus data dari dan ke entitas luar.

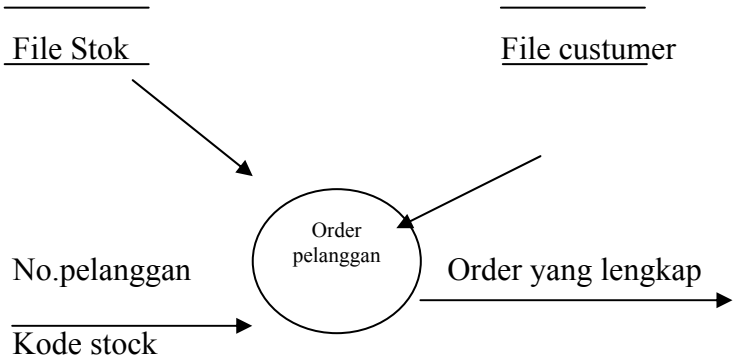


Gambar 2.3 Contoh diagram konteks

2. Diagram Level n

Dalam diagram level n DFD dapat digunakan untuk menggambarkan diagram fisik maupun diagram logis. Beberapa hal yang harus diperhatikan dalam membuat DFD ialah:

a. Jangan menghubungkan langsung antara satu tempat penyimpanan dengan penyimpanan lainnya.
b. Jangan menghubungkan langsung tempat penyimpanan data dengan entitas luar
c. Jangan menghubungakan langsung entitas dengan tempat penyimpanan data
d. Jangan menghubungkan langsung entitas luar dengan entitas luar
e. Jangan membuat suatu proses menerima input tetapi tidak pernah mengeluarkan output yang disebut dengan istilah ”*black hole*”
f. Jangan membuat suatu tempat penyimpanan menerima input tetapi tidak pernah digunakan untuk suatu proses
g. Jangan membuat suatu hasil proses yang lengkap dengan data yang terbatas yang disebut dengan istilah ”*magic process*”



Gambar 2.4 Contoh diagram level n

3. Data Flow Diagram Fisik

Dalam model fisik DFD lebih menunjukan di mana, bagaimana dan oleh siapa proses-proses dalam sistem tersebut dilakukan. Umumnya diagram fisik digunakan untuk menggambarkan sistem yang sedang berjalan pada saat analisa dengan pertimbangan diagram ini lebih mudah dipahami oleh user karena proses tersebut menggambarkan objek-objek yang melakukan proses dalam suatu sistem.

4. Data Flow Diagram Logis

Dalam model logis, DFD lebih menekankan proses-proses dan aliran data dari dan keluar dari proses tersebut.

2.6.5 Komponen DFD

Penjelasan komponen DFD :

1. Menggambarkan Kesatuan Luar
2. Proses
3. Arus data
4. Tempat penyimpanan data
5. Kesatuan Luar

   Kesatuan luar menggambarkan kesatuan-kesatuan diluar sistem yang kita gambarkan. Kesatuan ini menyediakan data untuk *input* ke sistem dan menerima data *output* dari sistem. Setiap kesatuan luar siberi nama sesuai dengan elemennya.

6. Proses

   Proses ialah kegiatan yang mentransformasi dari *input* menjadi *output*. Proses dapat digambarkan dengan lingkaran atu persegi empat bundar (*upright rectangle*). Penulisan lebel diproses dapat menggunakan kata benda untuk menggambarkan DFD model fisik dan kata kerja untuk menggambarkan DFD model logis.

7. Arus data

   Tanda panah digunakan untuk menggambarkan arus data yang mengalir diantara proses, tempat penyimpanan data dan kesatuan luar. Selain itu tanda panah juga dapat mewakili fisik seperti mengalirnya stok/persediaan barang dagangan.

8. Tempat penyimpanan data

   Tempat penyimpanan data digunakan untuk menyimpan data hasil proses maupun menyediakan data untuk diproses.

2.6.6 *Conceptual Data Model* (CDM)

Conceptual Data Model yang berisi komponen-komponen Himpunan Entitas dan Himpunan Relasi yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan seluruh fakta dari '*dunia nyata*' yang kita tinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan diagram *Conceptual Data Model*, penjelasan notasi-notasi di dalam *Conceptual Data Model* yang digunakan adalah:

1. Persegi Empat, menyatakan entitas dan dibawahnya menyatakan atribut (atribut yang berfungsi sebagai *key* digaris bawahi).
2. Garis, menyatakan Relasi sebagai penghubung antara Himpunan Realsi dengan Himpunan Entitas dan Himpunan Entitas dengan Atributnya.
3. Kardinalitas Relasi dapat dinyatakan dengan banyaknya garis cabang pada garis relasi, dan pada garis Relasi dapat langsung di tuliskan nama Relasi yang menghubungkan antar Entitas.

Conceptual Data Model adalah diagram grafikal yang mengambarkan keseluruhan struktur logik dari sebuah basis data. Pada model ini semua data yang ada pada dunia nyata diterjemahkan atau ditransformasikan dengan memanfaatkan sejumlah perangkat konseptual menjadi sebuah diagram data.

Untuk melambangkan fungsi di atas maka dipergunakanlah simbol-simbol yang bisa dilihat pada daftar simbol. Sedangkan arti dari masing-masing simbol adalah sebagai berikut :

a. Data Entitas

   Entitas merupakan individu yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Data entitas adalah segala sesuatu baik yang nyata maupun abstrak yang datanya akan direkam.

b. Atribut

   Atribut mendefinisikan karakteristik *(property)* dari entitas tersebut.

c. Relasi Data

   Relasi menunjukkan adanya hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda.

d. Kardinalitas

   Kardinalitas adalah penentuan jumlah kejadian dari satu entitas yang dapat terhubung dengan sejumlah kejadian pada entitas lain, juga menujukkan jumlah maksimum entitas yang dapat berelasi terhadap himpunan entitas yang lain. Terdapat empat kemungkinan kombinasi kardinalitas :

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

      Setiap entitas yang ada pada himpunan entitas A hanya boleh berhubungan satu entitas yang ada di himpunan entitas B, begitu pula sebaliknya.

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

   Setiap entitas yang ada pada himpunan entitas A boleh berhubungan ke banyak entitas yang ada di himpunan entitas B. Sedangkan untuk setiap entitas yang ada pada himpunan entitas B hanya boleh berhubungan satu entitas yang ada di himpunan entitas A.

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

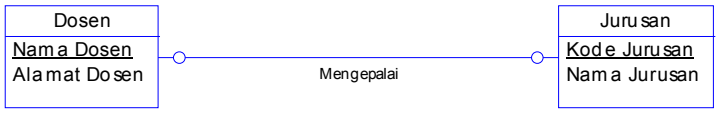
   Setiap entitas yang ada pada himpunan entitas A hanya boleh berhubungan ke satu entitas yang ada di himpunan entitas B. Sedangkan untuk setiap entitas yang ada pada himpunan entitas B boleh berhubungan ke banyak entitas yang ada di himpunan entitas A.

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

   Setiap entitas yang ada pada himpunan entitas A boleh berhubungan ke banyak entitas yang ada pada himpunan entitas B, begitu pula sebaliknya.

Berikut adalah contoh penggambaran relasi antara himpunan entitas lengkap dengan kardinalitas relasi dan atribut-atributnya:

1. Relasi **satu-ke-satu**



Gambar 2.5 Contoh relasi satu ke Satu

2. Relasi **satu-ke-banyak**



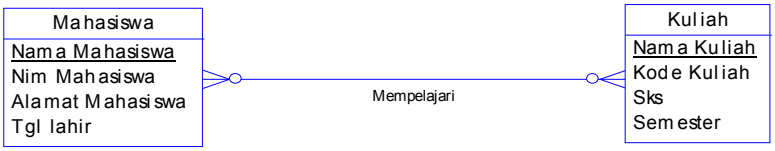
Gambar 2.6 Contoh relasi satu ke Banyak

3. Relasi **banyak-ke-satu**



Gambar 2.7 Contoh relasi Banyak ke-Satu

4. Relasi **banyak-ke-banyak**



Gambar 2.8 Contoh relasi Banyak ke-Banyak

2.6.7 Physical Database Design

Proses desain yang dilakukan sebelumnya terlepas dari konfigurasi system basis data yang akan digunakan. Pada tahapan ini, harus dipertimbangkan kemampuan system yang akan dipakai, jika perlu lakukan perubahan sehingga sesuai dengan performasi sistem. Seperti, jika system basis data yang akan dipakai misalnya dBASEIII, yang hanya memungkinkan membuka dua *file* sekaligus, maka diupayakan agar membuat maksimum dua *file* yang saling berhubungan. Dalam tahap ini juga harus diteliti ulang, apakah memang benar terjadi anomali

seperti penyimpangan, penyisipan penghapusan, dan perubahan. Jika ternyata penyimpangan harus digabung kembali, langkah ini biasa dikenal dengan istilah denormalisasi.

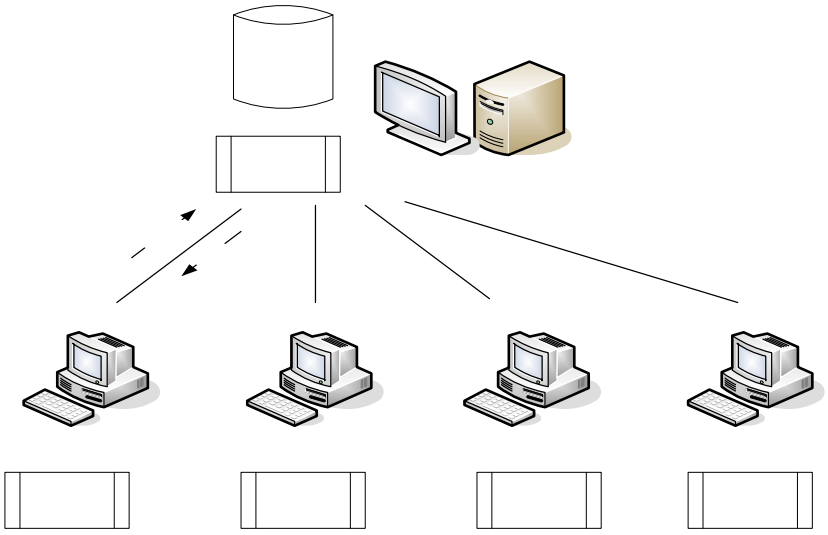
Pada hal ini yang dilakukan adalah mempertimbangkan pembuatan indeks. *Field-field* diberi indeks jika memang benar-benar diperlukan, karena jika dengan pemberian indeks yang berlebihan akan merepotkan dalam pemeliharaan *file* indeks tersebut.

2.7 Sistem *Client/Server*

Definisi yang banyak dipergunakan untuk menjelaskan sistem berbasis *client*/*server* adalah "sistem yang memisahkan antara tugas-tugas komputasi antara proses-proses *client* dan *server*." Dengan sistem *client/server*, kekuatan pemrosesan bisa disebarkan (didistribusikan) ke banyak mesin *client* dan mesin *server* yang terpisah secara fisik (itu sebabnya disebut *distributed system*).

Client/server mengacu kepada proses yang melibatkan setidaknya dua entitas yang independen, satu sebagai *client*, dan yang lainnya sebagai server. *Client* me-*request* ke *server* dan *server* melayani *request* dari *client*. Sebuah *request* dapat merupakan bentuk dari sebuah *query* SQL yang dikirimkan melalui *database engine* SQL. *Database engine* merubah proses permintaan (*request*) dan mengembalikkan ke client dalam bentuk hasil yang diinginkan. Dalam contoh ini, dua proses yang independen bekerja secara bersamaan untuk menyelesaikan tugas yang sama. *Windows printing* dan *print manager* adalah salah satu contoh sederhana hubungan sistem *client/server*. Aplikasi Windows, seperti Ms Excel dan Ms Word, menyajikan dokumen dan disaat dokumen tersebut akan dicetak,

maka dikirimkan perintah kepada *print manager*. *Print Manager* menyediakan servis untuk mengantrikan permintaan dan mengirimkan ke permintaan tersebut ke *printer*, memonitor pekerjaan yang sedang dilakukan, dan kemudian memberi tahu kepada aplikasi bila pekerjaan sudah selesai. Dalam contoh kasus ini *print manager* dianggap sebagai *server*, menyediakan servis untuk mengantrikan dan memproses pekerjaan pencetakan dokumen. Aplikasi Ms Word dan Ms Excel disebut sebagai *client*. Contoh ini menggambarkan bagaimana hubungan *client/server* dapat terjadi antar aplikasi yang mungkin tidak berhubungan dengan database.



Gambar 2.9 Arsitektur *Client/Server*

2.8 Sistem Penjualan

2.8.1 Pengertian penjualan

Penjualan barang dagangan merupakan sumber utama pendapatan dari suatu perusahaan dagang. Dalam usaha melakukan kegiatan tersebut, perusahaan harus menyediakan barang yang akan dijual belikan. Kadang-kadang orang salah mengerti tentang istilah penjualan yang dianggap sama dengan istilah pemasaran. Kedua istilah tersebut mempunyai ruang lingkup yang berbeda. Pemasaran meliputi kegiatan yang luas termasuk didalamnya penjualan. Sedangkan penjualan hanyalah merupakan satu kegiatan saja dalam pemasaran (penjualan merupakan subsistem dari pemasaran). Beberapa ahli menyebut penjualan sebagai ilmu dan beberapa ahli yang lain menyebutnya sebagai seni.

Pada pokoknya penjualan dapat diartikan sebagai berikut:

"Menjual adalah ilmu dan seni yang mempengaruhi pribadi yang dilakukan oleh penjual untuk mengajak orang lain agar bersedia membeli barang atau jasa yang ditawarkannya" (Drs.Alex S,1981:13)

Definisi lain tentang penjualan yang dilakukan oleh William.G.Nickles (1992:35) yang menyebutnya dalam istilah penjualan tatap muka (personal selling)

["Penjualan tatap muka adalah interaksi antar individu, saling bertemu muka yang ditujukan untuk menciptakan, memperbaiki, menguasai dan mempertahankan hubungan pertukaran yang saling menguntungkan dengan pihak lain"]

2.8.2 Tujuan penjualan

Tujuan penjualan pada hakekatnya adalah untuk mendapatkan uang atau laba dengan suatu proses pertukaran barang atau jasa kepada konsumen. Didalam perekonomian kita (ekonomi uang), seseorang yang menjual sesuatu akan mendapatkan imbalan berupa uang. Dengan alat penukar uang, orang akan lebih mudah memenuhi segala keinginannya.

2.8.3 Klasifikasi penjualan

Penjualan dapat diklasifikasikan ke dalam:

1. Penjualan secara tunai

   Penjualan yang dilakukan pada saat itu juga diterima pembayarannya dan dari pihak produsen menyerahkan barangnya. Pada penjualan tunai bagian-bagian yang terlibat antara lain bagian sales, order, billing, keuangan.

2. Penjualan secara kredit

   Penjualan yang dilakukan oleh produsen dengan menerima pembayaran dari konsumen tidak secara langsung. Pihak produsen menyerahkan barangnya kepada pihak konsumen dan pembayarannya dilakukan secara berangsur. Untuk pembelian secara kredit ini konsumen harus memenuhi persyaratan-persyaratan yang telah ditentukan oleh perusahaan. Bagian-bagian yang terlibat dalam penjualan kredit yaitu bagian sales order, billing, keuangan bagian kredit.

   Jadi, dengan adanya klasifikasi penjualan di atas, ini mempermudah pihak pembeli untuk melakukan transaksi dalam hal jual beli.

2.9 Bahasa Pemrograman Borland Delphi

Borland Delphi merupakan suatu bahasa pemrograman yang memberikan berbagai fasilitas pembuatan aplikasi visual. Keunggulan bahasa pemrograman ini terletak pada produktivitas, kualitas, pengembangan perangkat lunak, kecepatan kompilasi, pola desain yang menarik serta diperkuat dengan pemrogramannya yang terstruktur. Keunggulan lain dari Delphi adalah dapat digunakan untuk merancang program aplikasi yang memiliki tampilan yang berbasis Windows .

Lingkungan pengembangan terpadu dalam program Delphi terbagi menjadi beberapa bagian utama, yaitu Main Window, Main Menu, ToolBar, Component Palatte, Form Designer, Code Editor, Object Inspector, Explorer dan Object Tree View.

1. Main Window

   Jendela utama ini adalah bagian dari IDE yang mempunyai fungsi yang sama dengan semua fungsi utama dari program aplikasi Windows lainnya. Jendela utama Delphi terbagi menjadi tiga bagian, berupa Main Menu, Toolbar dan Component Palette.

2. Main Menu

   Menu utama pada Delphi memiliki kegunaan yang sama seperti program aplikasi windows lainnya yaitu untuk mengatur semua fasilitas yang ada dalam linkungan kerja Delphi, Juga dipakai untuk mendesain aplikasi serta memberikan fasilitas penunjuk segera. Sari menu ini, kita dapat memanggil atau menyimpan program, menjalankan program dan sebagainya.

3. Toolbar

   Delphi memiliki beberapa toolbar yang masing-masing memiliki perbedaan fungsi dan setiap tombol pada bagian toolbar berfungsi sebagai pengganti suatu menu perintah yang sering digunakan. Toolbar sering sering disebut juga dengan Speedbar.

   Delphi memiliki enam bagian toobar, antara lain: Standard, View, Debug, Desktops, Custom dan Component Palette.

4. Component Palette

   Component Palette berisi kumpulan-kumpulan icon yang melambangkan komponen-komponen pada VCL (Visual Component Library) atau CLX (Component Library for Cross Platform).

   Pada Component Palette Terdapat beberapa tab, antara lain:

   a. *Standard*, berisi komponen-komponen yang paling sering digunakan dalam semua program windows.
   b. *Additional*, Berisi sekelompok komponen-komponen unit.
   c. *System*, berisi visual dan non visual, seperti komponen timer, srive, akses *file*, DDE (*Dynamic Data Exchange*), Serta komponen-komponen OLE (*Object Linking Embeding*).
   d. *Data Access*, Dipakai untuk menghubungi dan mengambil data dalam *file*.
   e. *Data Controls*, Berisi komponen-komponen yang dipakai untuk menampilkan data-data secara visual yang didapatkan dari komponen-komponen data akses.

f. *Dialogs*, Pada bagian ini sering ditemukan beberapa dialog yang digunakan untuk *setup printer*, pencarian teks, membuka *file* dan sebagainya.

5. Form designer

Merupakan tempat yang dipakai untuk merancang jendela aplikasi baru yang sedang dibuat. Perancangan form dilakukan dengan meletakkan komponen-komponen yang diambil dari Component Palette.

6. Code Editor

Code Editor Merupakan tempat di mana kita dapat menuliskan kode program. Pada bagian ini kita dapat menuliskan pernyataan-pernyataan dalam object pascal. Keuntungan bagi pemakai Delphi adalah bahwa kita tidak perlu menuliskan kode-kode sumber, karena Delphi telah menyediakan kerangka penulisan sebuah program.

7. Object Inspector

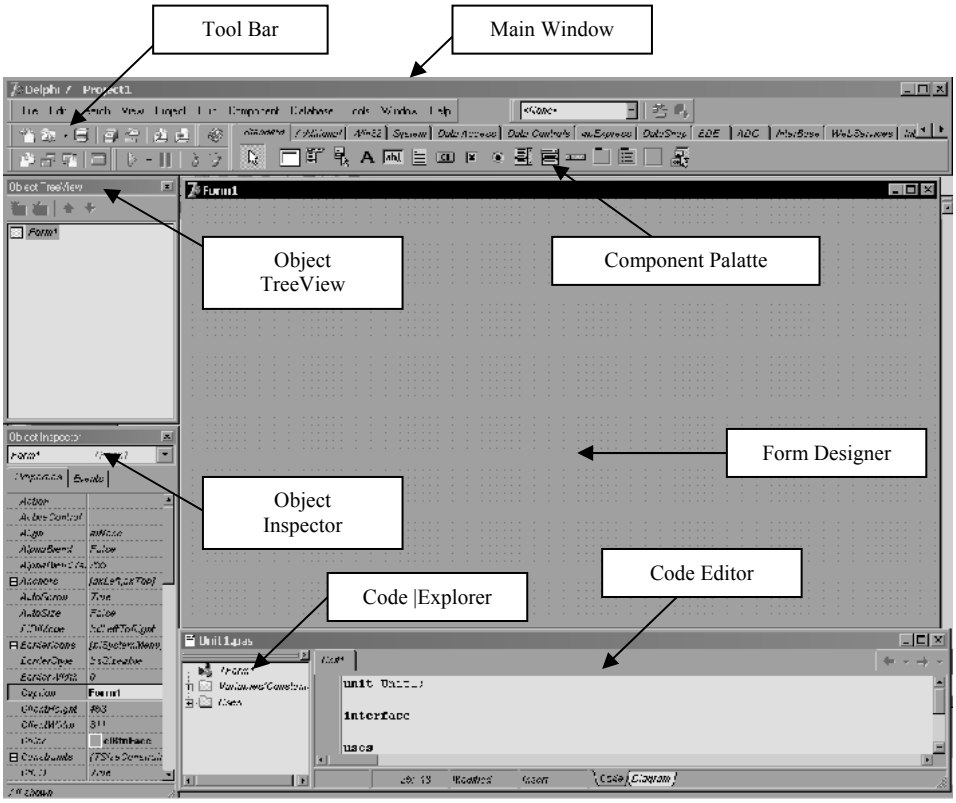
Object Inspector digunakan untuk mengubah karakteristik sebuah komponen. Pada Object Inspector terdapat dua tab yaitu: *Properties*, yang menampilkan daftar properties beserta nilainya untuk object yang tertulis dan *Events*, yang menampilkan daftar event objek yang sedang terpilih untuk mengalihkan prosedur-prosedur penanganan events. Salah satu tab ini dapat diaktifkan dengan cara meng-klik properties dan Event.

8. Code Explorer

Code Explorer digunakan untuk memudahkan pemakai berpindah antar *file* unit yang terdapat didalam jendela Code Editor.

9. Object Tree View

Object Tree View menampilkan diagram pohon dari komponen-komponen yang bersifat visual maupun nonvisual yang telah terdapat dalam form, data module, atau frame. Object Tree View juga menampilkan hubungan logika antar komponen.

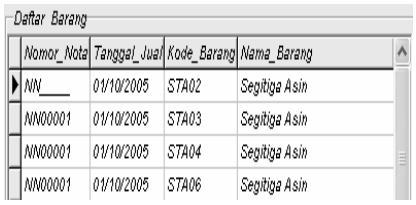
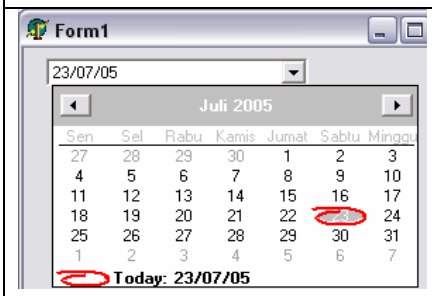


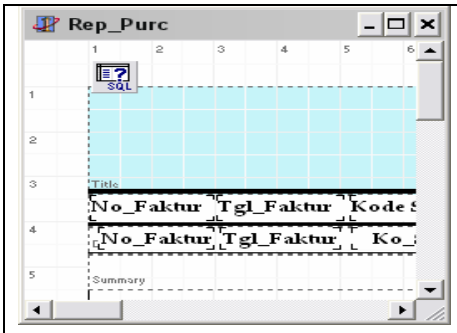
Gambar 2.10 Lembar Kerja Borland Delphi

Adapun beberapa contoh penerapan komponen yang digunakan dalam penulisan ini adalah :

Tabel 2.1 komponen-komponen pada borland delphi

Nama dan bentuk icon	Contoh coding
 Komponen Edit	Menonaktifkan komponen edit1 dan merubah warna latar belakang menjadi tidak aktif **Edit1.Enabled:=False;** **Edit1.Color:=clInAvticeBorder;**
 Komponen Tombol (Button)	Mengaktifkan komponen Button1 dan memfocuskan posisi pointer ke komponen button1 **Button1.Enabled:=True;** **Button1.Setfocus;**
 Table	Penyimpanan data ke dalam database. Diasumsikan koneksi yang dibutuhkan komponen telah diatur. **With Table1 do** **Begin** **Append;**

	Table1[‘Ko_Supplier’]:=Ko_Supplier.Text; **Table1[‘N_Supplier’]:=N_Supplier.Text;** **Post;** **End;**
Nama dan bentuk icon	**Contoh coding**
  Daftar Barang Nomor_Nota | Tanggal_Jual | Kode_Barang | Nama_Barang NN___ | 01/10/2005 | STA02 | Segitiga Asin NN00001 | 01/10/2005 | STA03 | Segitiga Asin NN00001 | 01/10/2005 | STA04 | Segitiga Asin NN00001 | 01/10/2005 | STA06 | Segitiga Asin DbGrid,Datasource,table	Menampilkan data dari dalam database Diasumsikan koneksi yang dibutuhkan komponen telah diatur. **Dbgrid1.Datasource:=dsDatasource1**
 Form1 23/07/05 Juli 2005 Sen Sel Rabu Kamis Jumat Sabtu Minggu 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7 Today: 23/07/05 DateTimePicker1	Menmpilkan data tanggal sesuai dengan tanggal sekarang **DateTimePicker1.DateTime:=Now;**

 Query dan Report	Menampilkan data dalam bentuk laporan Diasumsikan koneksi yang dibutuhkan komponen telah diatur dan pengguna menekan tombol laporan. **Query1.SQL.Clear;** **Query1.SQL.Add('Select * From Pembelian);** **Query1.Open;** **Rep_Purc.Preview;**

2.10 Desain Database

Database merupakan suatu hal yang sangat penting dalam pembuatan program, karena nantinya akan memudahkan pengaksesan *file* didalam program itu sendiri. Kesulitan utama dalam desain database adalah bagaimana agar *database* dapat memuaskan keperluan saat ini dan masa mendatang. Dalam perancangan konseptual akan menunjukan *entity* dan relasinya berdasarkan proses yang diinginkan oleh organisasi.

Definisi *database* adalah sebagai berikut : "*Database* adalah kumpulan *file-file* yang saling berelasi, relasi tersebut biasa ditunjukan dengan kunci dari setiap *file* yang ada" **(Ir. Harianto Kristanto, 1998:1)**.

Suatu *database* menunjukan satu kumpulan data yang dipakai dalam suatu lingkup perusahaan, lembaga atau instansi. Penerapan *database* dalam sistem pengolahan data disebut Sistem *Database*.

1. *Database Management System* (DBMS)

merupakan perangkat lunak atau Program computer yang dirancang secara khusus untuk memudahkan pengolahan database. Salah satu macam DBMS yang popular dewasa ini berupa RDBMS (*Relational Database Management System*), yang merupakan model basis data relasional data dalam bentuk tabel-tabel yang saling terhubungkan. beberapa istilah dalam RDBMS :

2. Baris dan Kolom

Sebuah tabel berisi sejumlah baris data dan sejumlah kolom. Kolom(*field*) dapat didefinisikan sebagai satuan data terkecil dalam sebuah tabel. Baris(*record*) adalah kumpulan kolom yang menyatakan suatu data yang salin terkait

3. Kunci Primer dan Kunci Tamu

Secara konsep Setiap tabel harus memiliki kunci primer. Kunci primer berperan sebagai uindentitas yang unik (tak kembar) untuk masing-masing baris data. Kunci tamu adalah sebuah kolom dalam sebuah tabel yang menjadi penghubung dengan kunci primer pada tabel lain.

4. Indeks

Indeks merupakan suatu mekanisme dalam databse yang memungkinkan pencarian data dapat dilakukan dengan cepat. Sebagai contoh, jika Anda sering mengambil data menurut nama pegawai, maka bila nama pegawa Anda

diindeks, penmgaksessan data terhadap nama akan dilakukan dengan cepat. Namun walaupun indeks bermanfaat dalam mempercepat pencarian data, indeks sebenarnya membuat system penambahan, peremajaan, dan penghapusan data menjadi lambat, karena setiap kali terjadi proses-proses tersebut maka indeks akan diperbaharui. Oleh karena itu penggunaan indeks perlu dibatasi pada kolom-kolom tertentu.

Database Programming Language

Untuk mengakses database yang ada pada *Back end/server* diperlukan aplikasi database yang dijalankan pada *front end/client*. Aplikasi database tersebut pada dasarnya adalah program komputer yang menyediakan fasilitas kepada pengguna untuk mengakses, menambah, mengubah dan menghapus data dari database yang ada di *Back end/server*.

Salah satu contoh dari *database Programming language* adalah *Structure Query Language* (SQL). SQL adalah bahasa standar yang meliputi perintah-perintah untuk menyimpan, menerima, memelihara dan mengatur akses-akses ke basis data, serta digunakan untuk memanipulasi dan menampilkan data dari RDBMS.

Beberapa sintak yang sering digunakan dalam SQL, yaitu :

1. *Select* : select digunakan untuk mengambil data dari suatu tabel.
2. *Insert* : perintah ini digunakan untuk menyisipkan data ke dalam tabel.
3. *Delete* : perintah *delete* digunakan untuk menghapus record pada tabel.

4. *Update* : perintah *update* digunakan untuk memperbaharui nilai suatu data.

SQL merupakan bahasa generasi keempat yang mudah dipahami. SQL merupakan bahasa standar untuk pengolahan database, mulai dikembangkan pada akhir 1970 di laboratorium *IBM, San Jose, California.*

Bahasa SQL terdiri dari :

1. *Data definition language* (DDL), digunakan untuk memproses data baik secara *logical* maupun *physical.*

   CREATE, DROP, ALTER

   Contoh :

   a. Membuat data base.
   b. Memberikan informasi skema dan subskema dari data.
   c. Memberikan informasi semua *record* dan *field*.
   d. Memberikan batasan akses pada user.

2. *Data manipulation language* (DML), digunakan untuk perawatan basis data yang di dalamnya termasuk kegiatan *updating, inserting,* serta *deleting record* dari basis data.Bahasa ini dikembangkan khusus untuk mengakses *Relational Database Management System*. SQL dapat disebut sebagai *Non-Procedural Language* karena bahasa ini tidak memiliki fasilitas untuk pengulangan dan kondisi sebagaimana layaknya bahasa pemrograman lainnya.

   Contoh penggunaan bahasa SQL :

   *Select * from karyawan.db*
   Select nama_karyawan, nomor_karyawan