UNIT PELAKSANA TEKNIS PERPUSTAKAAN UNIKOM

BAB II

DASAR TEORI

2.1. Pengertian Sistem Informasi

Sistem merupakan kumpulan bagian atau komponen yang saling berkaitan, berhubungan, berketergantungan, dan saling mendukung yang secara keseluruhan membentuk satu kesatuan dalam mencapai suatu tujuan tertentu secara efisien dan efektif. Sistem informasi manajemen mulai digunakan pada tahun 1960 dan terus berkembang pesat, hingga saat ini telah diterapkannya sistem pakar, sistem informasi eksekutif, dan strategi sistem informasi. Sedangkan untuk kebutuhan informasi komunikasi dan analisis didukung oleh AI (*Artificial Intelegence*).

2.1.1. Pengertian sistem informasi manajemen

Sistem informasi manajemen merupakan bidang kajian yang sangat penting bagi perkembangan suatu perusahaan, seluruh proses yang terjadi akan berjalan baik apabila didukung oleh sistem informasi yang handal. Jadi Sistem informasi manajemen adalah ilmu yang mempelajari cara-cara mengelola pekerjaan pengolahan informasi dengan menggunakan pendekatan sistem yang berdasarkan pada prinsip manajemen. Sehingga bentuk aplikasi ditiap perusahaan akan berbeda jadinya walaupun mereka menggunakan dasar manajemen yang mirip atau sama, karena tiap manajer memiliki alat pengambil keputusan yang berbeda pula untuk menetapkan suatu keputusan yang tepat bagi perusahaannya.

2.1.2. Kegunaan sistem informasi manajemen

Pengguna sekaligus sebagai pengembang sistem informasi manajemen dalam perusahaan adalah bagian personalia. Hal ini dikarenakan fungsi manajemen informasi tersebut sebagai alat untuk meningkatkan *produktivitas* secara personal maupun organisasi melalui pengambilan keputusan dan komunikasi antar bagian dalam perusahaan dengan membudidayakan kemampuan komputer. Sehingga diharuskan bahwa staff personalia memiliki sistem informasi manajemen yang baik dan handal.

2.2. Konsep Perancangan Sistem Informasi

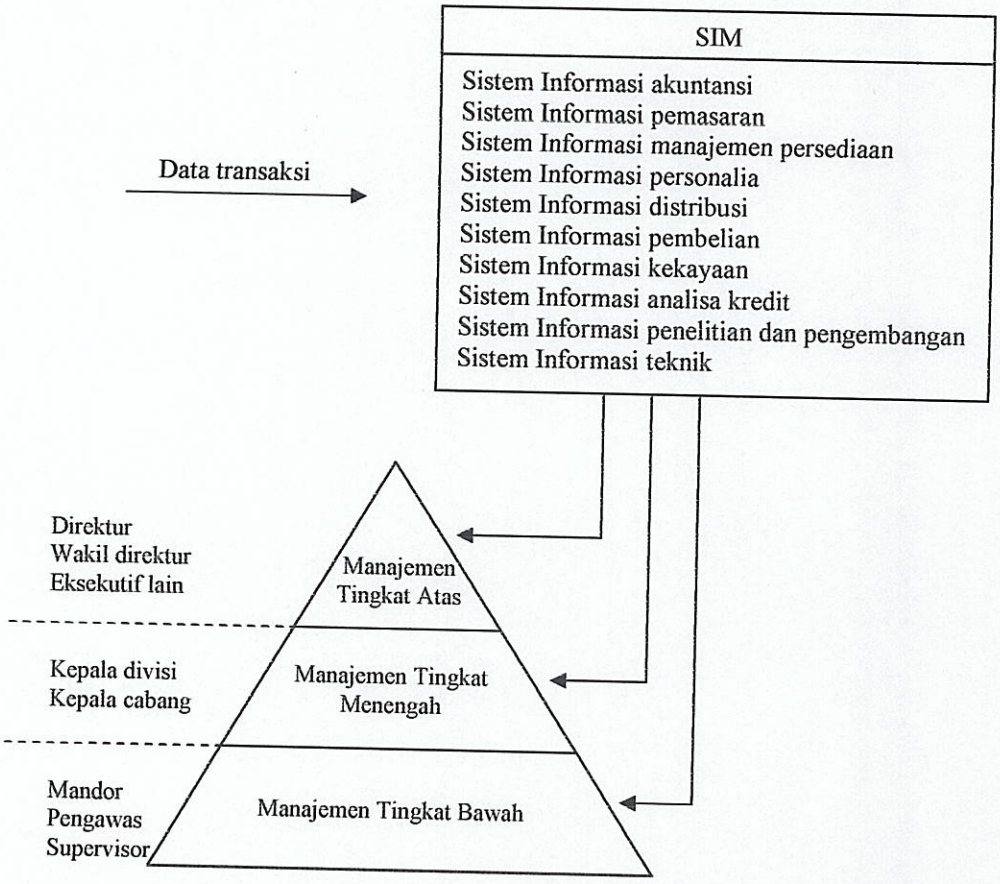
Dari sebuah organisasi banyak sekali unsur atau entitas yang terdapat didalamnya, baik secara perangkat keras maupun lunak. Perangkat keras dapat dilihat dari teknologi yang digunakan, para karyawan dan pelaku bisnis lainnya, gedung atau lokasi, dan produk yang dihasilkannya. Sedangkan dari segi perangkat lunak dapat dilihat dari berbagai penerapan ilmu seperti halnya struktur organisasi, bagan alur, sistem keamanan, penempatan karyawan, siklus keluar-masuk barang, sistem pelaporan, maupun sitem informasi yang dipergunakan.

Untuk membangun suatu sistem informasi diperlukan beberapa langkah, diantaranya sebagai berikut:

1. Pemahaman tentang konsep dasar sistem informasi yang akan diambil, meliputi:
    a. Komponen sistem informasi yang ingin dibuat
    b. Sistem informasi manajemen

c. Sistem informasi akuntansi

2. Peranan sistem informasi bagi manajemen



Gambar. 2.1. Informasi manajemen di semua *level* manajemen

3. Perencanaan sistem yang ingin dibangun, dilihat dari aspek :

   a. Latar belakang organisasi

   b. Proyek-proyek sistem informasi direncanakan

   c. Sasaran pengembangan sistem informasi

   d. Kendala-kendala pengembangan sistem informasi

   e. Prioritas proyek-proyek sistem informasi

f. Rincian proyek-proyek sistem informasi direncanakan

4. Analisis sistem, terdiri dari beberapa langkah dasar yang harus dilakukan oleh para analis sistem, sebagai berikut :

   a. *Identify*, yaitu mengidentifikasi masalah

   b. *Understand*, yaitu memahami kerja dari sistem yang ada

   c. *Analyze*, yaitu menganalisis sistem

   d. *Report*, yaitu membuat laporan hasil analisis

5. Desain sistem, didefinisikan sebagai berikut :

   a. Tahap setelah analisis dari siklus pengembangan sistem

   b. Pendefinisian dari kebutuhan-kebutuhan fungsional

   c. Persiapan untuk rancang bangun implementasi

   d. Menggambarkan bagaimana suatu sistem dibentuk

   e. Berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi

   f. Konfigurasi komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem

6. Seleksi sistem

   Merupakan tahap untuk memilih perangkat keras dan perangkat lunak untuk sistem informasi.

7. Implementasi sistem, terdiri dari :

   a. Menerapkan rencana implementasi

   b. Melakukan kegiatan implementasi

   c. Tindak lanjut implementasi

8. Pengendalian / pengawasan, yang terbagi atas dua tingkatan yakni :

   a. Pengendalian administratif (*administrative control*)

      Yaitu pengawasan atau pengendalian mengenai sistem yang telah ditetapkan dalam rangka tujuan organisasi dan dalam rangka kebijaksanaan mengenai sikap, kelakuan dan berpikir.

   b. Pengendalian operasional (*operative control*)

      Yaitu pengawasan atau pengendalian mengenai perbuatan atau cara bekerja berdasarkan sistem yang ada.

2.3. Komponen Sistem Informasi

Untuk membangun sebuah sistem informasi diperlukan komponen-komponen sebagai bahan pembentuk sistem tersebut. Oleh John Burch dan Gary Grudnitski mengistilahkan komponen tersebut dengan istilah **blok bangunan** (*building block*), yaitu :

1. Blok masukan (*input block*)

   Blok *input* merupakan data-data yang masuk ke dalam sistem informasi, yang dapat berupa dokumen-dokumen dasar yang dapat diolah menjadi suatu informasi tertentu..

2. Blok model (*model block*)

   Blok ini terdiri dari kombinasi prosedur, *logika* dan model matematik yang akan mengolah data *input* (baik berasal dari luar maupun yang tersimpan di *database*) untuk menghasilkan suatu informasi yang dibutuhkan.

3. Blok keluaran (*output block*)

   Merupakan informasi yang dihasilkan bisa berupa kumpulan data yang nantinya akan disimpan atau bisa berupa data cetak laporan.

4. Blok teknologi (*technologi block*)

   Teknologi informasi merupakan penunjang utama dalam berlangsungnya sistem informasi. Teknologi komputer terdiri dari beberapa komponen, diantaranya alat untuk memasukkan data (*input device*), untuk melakukan pengolahan data (*processing device*), alat untuk menyimpan dan mengakses data (*storage device*), alat untuk menghasilkan dan mengirimkan keluaran (*output device*) dan alat untuk membantu pengendalian dari sistem secara keseluruhan (*controll device*). Teknologi informasi terdiri dari 3 (tiga) bagian utama, yaitu teknisi (*humanware* atau *brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

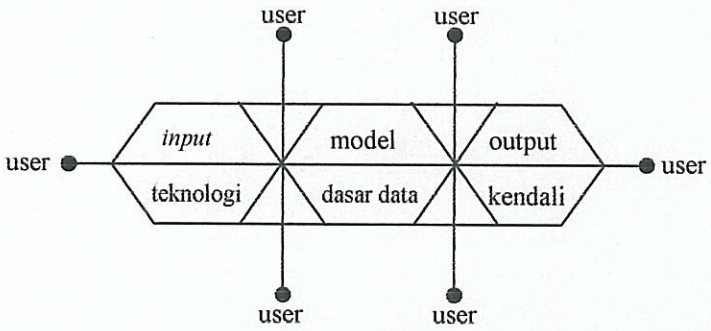
5. Blok Basis Data (*database block*)

   Basis data merupakan sekumpulan data yang saling berhubungan satu sama lainnya, tersimpan di alat penyimpanan komputer (*storage device*). Data tersebut dapat dipanggil keberadaanya untuk proses pengolahan informasi yang dibutuhkan. Supaya informasi yang dihasilkan berkualitas, maka *database* yang ada perlu sistem keamanan dan pengendali yang baik. Untuk

mengakses atau mengolah data dari *database*, seorang user dapat menggunakan software (paket perangkat lunak) yang disebut dengan DBMS (*Database Management System*)

6. Blok kendali (*controls block*)

   Pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun sistem dapat diperbaiki kembali ke keadaan semula (*system recovery*).



Gambar. 2.2. Interaksi komponen sistem informasi

2.4. Basis Data

Basis data merupakan sekumpulan data tertentu yang secara bersama-sama dapat digunakan untuk aplikasi-aplikasi yang berbeda (D.R Howe, "Data Analysis For Data Base Design", 1999).

Penggunaan basis data dapat diartikan sebagai sebuah *locker* atau brankas yang menyimpan berbagai *item* didalamnya. Sedangkan keadaan didalam brankas tersebut perlu juga dilakukan pemisahan atau pengaturan. Dari sini dapat kita simpulkan bahwa dalam sistem basis data diperlukan juga adanya pengaturan atau sering kita istilahkan sebagai suatu sistem, yaitu sistem basis data atau "**Database Management System** (DBMS)".

DBMS merupakan perangkat lunak yang memberikan fasilitas yang tersedia uintuk fungsi pengaturan, pengawasan, pengendalian atau kontrol, pengolahan dan koordinasi terhadap semua proses atau operasi yang terjadi pada sistem basis data.

DBMS terbagi dalam 2 (dua) bagian besar yaitu *global schema* dan *local schema*. Skema global lebih mengacu pada bagaimana data tersebut dapat menjadi suatu basis data dan bagaimana data tersebut dapat disimpan dan diolah menjadi suatu informasi. Sedangkan pada skema lokal terdiri dari 3 (tiga) bagian dasar, yaitu :

1. *Conceptual schemas*

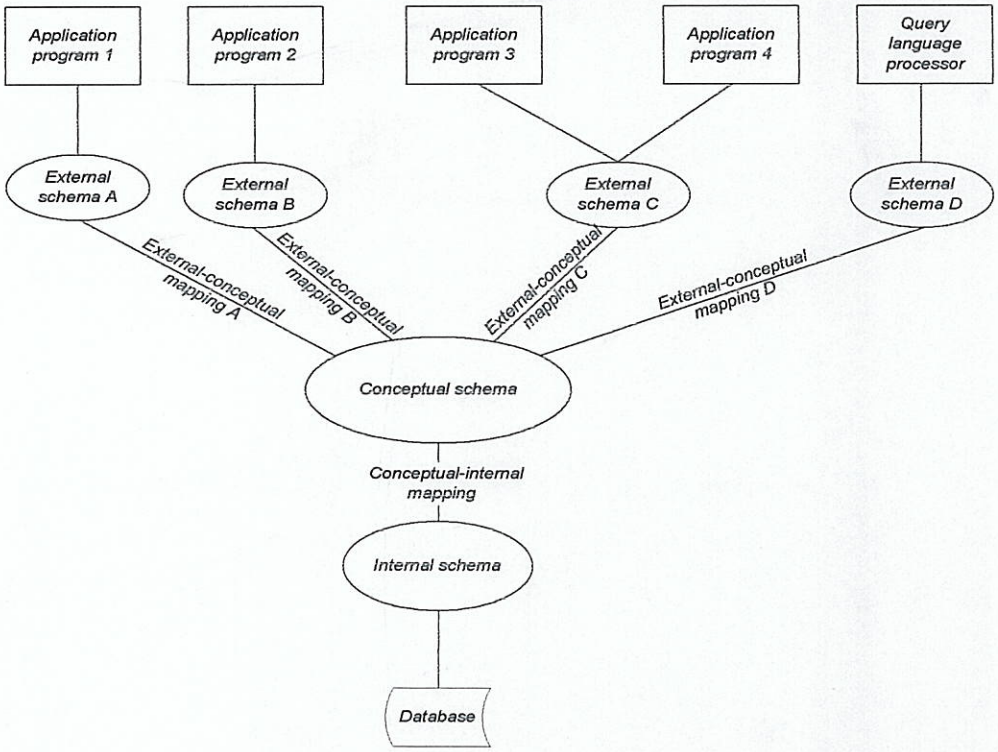
   Merupakan seluruh data-data yang diibutuhkan oleh perusahaan yang kemudian disimpan ke dalam *database*. Mulai dari pembentukkan struktur basis data, pembuatan tampilan data masukkan, dan bagaimana data-data tersebut saling terhubung didalam sistem basis data.

2. *Internal schemas*

   Adalah bagaimana data yang telah tersimpan di *database* dirancang untuk kebutuhan seluruh aplikasi-aplikasi, seperti pembuatan tampilan laporan, pencetakan laporan, penyimpanan kembali setiap hasil aplikasi, pemberian indeks, pembuatan algoritma, *pointer*, *blocksizes*, dan sebagainya.

3. *External schemas*

   Adalah pendefinisian tiap data yang mengalir ke suatu program aplikasi atau ke dalam beberapa aplikasi secara bersamaan.



Gambar. 2.3. Interaksi komponen sistem informasi

Bahasa basis data yang digunakan ada dua (2) buah, yakni Bahasa Definisi Data dan Bahasa Manipulasi Data.

2.4.1. Bahasa Definisi Data (*Data Definition Language* / DDL)

DDL merupakan perintah yang umum digunakan oleh administrator basis data (DBA) untuk mendefinisikan skema dan subskema ke dalam DBMS.

Skema adalah deskripsi lengkap tentang struktur medan (*field*), rekaman (*record*) dan hubungan data pada basis data. Tugas utama skema adalah menjabarkan struktur DBMS. Penjabaran pada DBMS mencakup :

a. Nama basis data.

b. Nama seluruh berkas pada basis data.

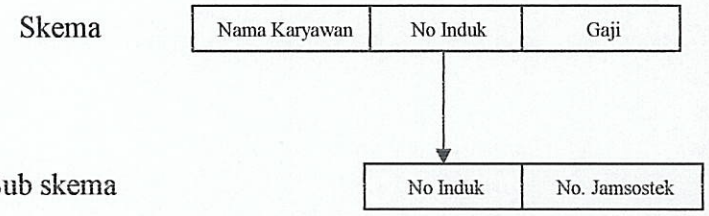
c. Nama rekaman (*record*) dan medan (*field*).

d. Deskripsi berkas, rekaman dan medan.

e. Nama medan kunci.

f. Nama indeks dan medan yang menjadi indeks.

Sub skema adalah pandangan (*view*) yang dibuat untuk pengguna basis data, dan merupakan himpunan bagian dari skema.



Gambar. 2.4. Skema dan Sub-skema dalam basis data

Hasil dari kompilasi perintah DDL adalah kumpulan tabel yang tersimpan dalam file khusus yang disebut kamus data (*data dictionary*). Kamus data merupakan meta data (super data) yaitu data yang dideskripsikan dari data sesungguhnya, dalam suatu sistem operasi basis data sebelum file sesungguhnya diakses.

2.4.2. Bahasa Manipulasi Data (*Data Manipulation Language*/ DML)

DML merupakan bentuk bahasa basis data yang digunakan melakukan manipulasi dan penggambilan data dari suatu basis data.

Proses dalam Bahasa Manipulasi Data berupa :

a. Penyisipan atau penambahan data baru ke suatu basis data.

b. Penghapusan data dari suatu basis data.

c. Pengubahan data disuatu basis data.

Jenis metode pemakaian Bahasa Manipulasi Data terbagi dua (2) macam, yaitu :

1. *Procedural*

Penggunaannya menuntut pihak pengguna untuk menentukan data apa saja yang diperlukan beserta bagaimana cara untuk mendapatkannya. Metode ini dipergunakan apabila si pengguna memerlukan ketepatan dan keakuratan data / informasi yang tinggi.

2. *Non - Procedural*

Sebaliknya metode ini menuntut pihak pengguna untuk menentukan data apa saja yang diperlukan tanpa perlu mencantumkan bagaimana cara mendapatkannya. Sehingga metode ini hanya melakukan search data secara otomatis dari keseluruhan basis data.

2.5. Kamus Data (*Data Dictionary*)

Kamus data yaitu representasi fakta dari data dan kebutuhan-kebutuhan sistem informasi, kamus data biasanya berbentuk sebuah katalog atau perincian secara berurut. Didefinisikan sebagai berikut :

" *Kamus data merupakan sebuah daftar yang terorganisasi dari elemen data yang berhubungan dengan sistem, dengan definisi yang tegar dan teliti sehingga pemakai dan analis sistem akan memiliki pemahaman yang umum mengenai input, output, komponen penyimpanan, dan bahkan kalkulasi inter-*

mediate. " (Roger S. Pressman, Ph.D. , "Rekayasa Perangkat Lunak", 2002, p388)."

Kamus data digunakan pada tahapan analisis sebgai alat komunikasi antara sistem analis dengan pemakai sistem tentang data-data yang dipakai dan informasi yang dihasilkan. Sedangkan pada tahapan desain, kamus data digunakan untuk merancang *input*, *output* (berupa laporan-laporan), dan merancang *database*.

2.6. Rekayasa Sistem Berbasis Komputer

2.6.1. Definisi sistem berbasis komputer

Definisi sistem berbasis komputer menurut Webster's adalah :

"Serangkaian atau tatanan elemen-elemen yang diatur untuk mencapai tujuan yang ditentukan sebelumnya melalui pemrosesan informasi" (Roger S. Pressman, Ph.D. , "Rekayasa Perangkat Lunak", 2002, p276).

Sistem berbasis komputer ini bertujuan untuk mendukung berbagai fungsi bisnis atau untuk pengembangan produk perusahaqan dalam meningkatkan keuntungan bisnis. Untuk mencapai tujuan tersebut, sistem berbasis komputer menggunakan berbagai elemen sistem :

1. Perangkat lunak (*Software*)

   Program komputer, struktur data, dan dokumen yang berhubungan dan berfungsi untuk mempengaruhi metode logis, prosedur, dan kontrol yang dibutuhkan.

2. Perangkat keras (*Hardware*)

   Merupakan perangkat elektronik yang memberikan kemampuan penghitungan, dan perangkat elektromekanik (misalnya sensor, rotor, pompa) yang memberikan fungsi dunia eksternal.

3. Manusia

   Berfungsi pemakai dan operator perangkat keras dan perangkat lunak.

4. *Database*

   Kumpulan informasi yang besar dan terorganisasi yang diakses melalui perangkat lunak melalui *input device*.

5. Dokumentasi

   Berbentuk laporqan-laporan *manual*, formulir, dan informasi deskriptif lainnya yang menggambarkan penggunaan dan atau pengoperasian sistem.

6. Prosedur

   Langkah-langkah yang menentukan penggunaan khusus dari masing-masing elemen sistem atau konteks prosedural dimana sistem berada.

2.6.2. Pemodelan sistem

Untuk memberikan fokus pada pemodelan sistem, maka perekayasa menciptakan model dengan batasan mengikuti dalam buku karangan Motamarri,S., "Systems Modeling and Description", *Software Engineering Notes*, Vol.17, No.2, April 1992, h.57-63, sebagai berikut :

1. Membatasi proses yang melayani kebutuhan sistem global.

2. Menggambarkan tingkah laku proses dan asumsi di mana tingkah laku itu didasarkan.
3. Secara eksplisit membatasi *input exogeneous* dan *endogeneous* kepada model.
4. Mempresentasikan semua hubungan termasuk *output* yang memungkinkan perekayasa memahami pandangan tersebut dengan lebih baik.

Input exogeneous menghubungan satu konstituen dari pandangan yang diberikan dengan konstituen yang lain pada ttingkat yang sama atau tingkat yang lain. Sedangkan ***input endogeneous*** adalah menghubungkan komponen individual dari *konstituen* pada sebuah pandangan khusus.

2.6.3. Rekayasa informasi sebagai suatu kajian

Tujuan dari rekayasa informasi (*information engineering*) adalah untuk membuat arsitektur yang memungkinkan suatu bisnis menggunakan informasi secara efektif. IE juga digunakan untuk mengimplementasikan arsitektur-arsitektur tersebut ke dalam sistem informasi.

Ada 3 (tiga) model arsitektur yang harus diperhatikan dalam IE, yaitu :

1. Arsitektur data

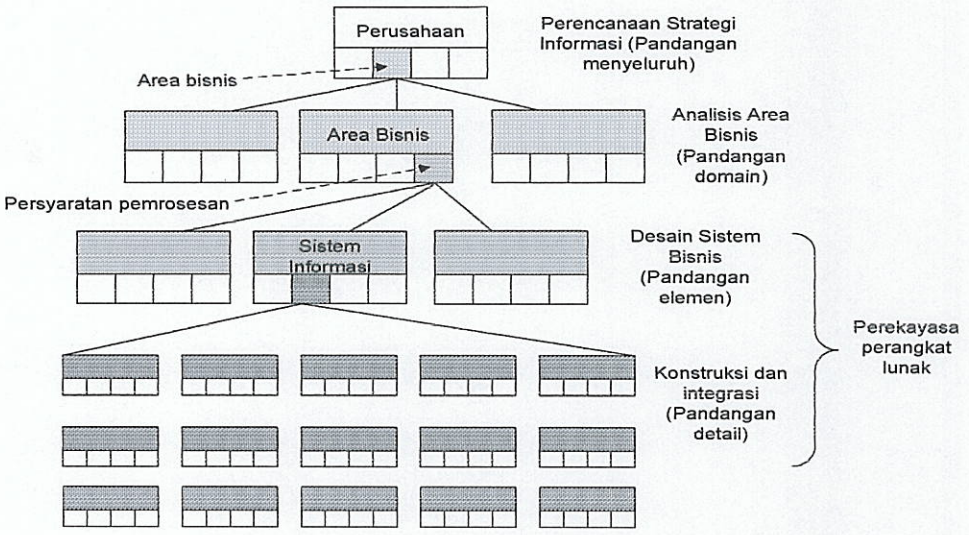
   Kerangka kerja untuk kebutuhan informasi dari bisnis atau fungsi bisnis. Blok bangunan individual-nya berupa objek data yang digunakan dalam *database* yang kemudian diolah untuk memberikan informasi keperluan bisnis.

2. Arsitektur aplikasi

   Elemen-elemen dari suatu sistem yang men-transformasi objek ke dalam arsitektur data untuk keperluan bisnis. Dapat digambarkan sebagai sistem perangkat lunak dalam melakukan transformasi tersebut.

3. Arsitektur teknologi

   Menjadi dasar berlangsungnya kegiatan transformasi dari arsitektur sebelumnya. Mencakup perangkat keras dan perangkat lunak yang mendukung aplikasi dan data.



Gambar 2.5. Hirarki rekayasa informasi

2.7. Desain dan Rekayasa Perangkat Lunak

Desain perangkat lunak merupakan inti dari semua tahapan analisis yang telah dilakukan. Apabila persyaratan perangkat lunak telah mulai dianalisis dan ditentukan, maka desain perangkat lunak menjadi yang pertama dari 3 (tiga) aktivitas teknik, *design – codding – testing*.

Desain meliputi 4 (empat) hal yang menjadi konsep utama atau langkah pembuatan perangkat lunak. Yaitu :

1. Desain data

   Mentransformasi model dasar yang dibuat selama analisis ke dalam struktur data. Objek dan hubungan antar entitas (*Entity Relationship Diagram* / ERD). Dan mengambil basis dari kamus data yang telah dibuat.

2. Desain arsitektur

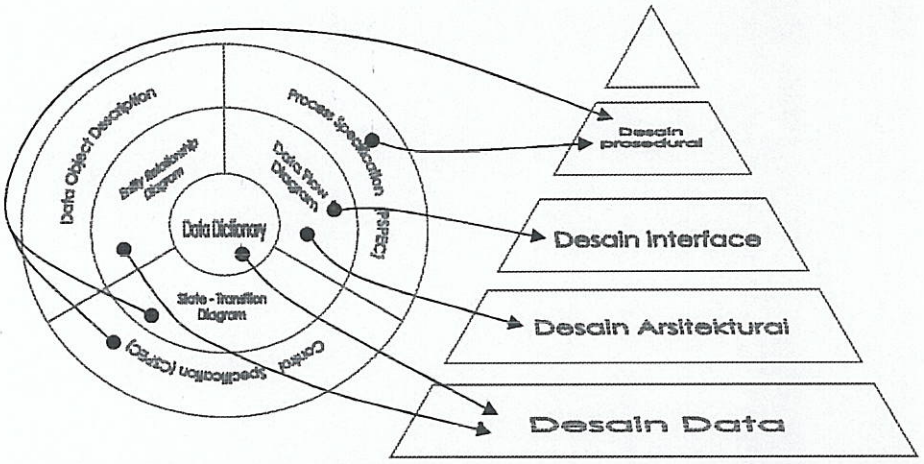
   Menentukan hubungan diantara elemen-elemen struktural utama dari program. Representasi desain tersebut – kerangka kerja modular dari sebuah program komputer – dapat diperoleh dari model-model analisis dan interaksi subsitem yang ditentukan dalam model analisis.

3. Desain *interface*

   Menggambarkan bagaimana perangkat lunak berkomunikasi dalam dirinya sendiri, dengan sistem yang berinter-operasi dengannya, dan dengan manusia yang menggunakannya. *Interface* mengimplikasikan aliran informasi (misal data atau kontrol) dengan demikian, data dan diagram aliran kontrol memberikan informasi yag dibutuhkan bagi desain *interface*.

4. Desain prosedural

   Mentransformasi elemen-elemen struktural dari arsitektur program ke dalam suatu deskripsi prosedural dari komponen-komponen perangkat lunak. Informasi yang diperoleh dari PSPEC, CSPEC, dan STD berfungsi sebagai dasar bagi desain prosedural.



Gambar 2.6. Model analisis dalam suatu desain perangkat lunak

2.8. Bahasa Pemrograman C++ Builder

2.8.1. Ilustrasi dasar

C++ Builder merupakan aplikasi yang digunakan untuk pengembangan sistem informasi dengan memanfaatkan keistimewaan konsep-konsep antar muka grafis dalam Microsoft Windows. Aplikasi dalam C++ Builder beroperasi di bawah lingkup Sistem Operasi Windows. C++ Builder merupakan pengembangan bahasa C atau pemrograman C++ yang merek dagangnya dipegang oleh Borland International Inc.

C++ Builder adalah pemrograman dengan IDE (*Integrated Development Environment)* yang mirip dengan Delphi, namun bahasa dasarnya adalah C++. Keunggulan C++ Builder dengan bahasa pemrograman yang mengadopsi konsep *Event Driven Progamming* seperti Visual Basic dan Borland Delphi adalah:

1. Aplikasi bisa dikompilasi berdasarkan kriteria-kriteria kecepatan eksekusinya atau ukuran memori yang optimal digunakan saat ia dieksekusi.

2. C++ Builder lebih memungkinkan para pemrogram dalam mengelola memori yang digunakan oleh aplikasi yang dikembangkannya, melalui implementasi *pointer* (keunggulan bahasa C++).

C++ Builder menerapkan beberapa sistem pemrograman yang dapat membantu pengguna dalam menggunakan dan mengelola hasil dari pemrograman, seperti :

1. RAD (*Rapid Application Development*)

   Merupakan perangkat pemrograman yang mampu mebuat aplikasi dengan cepat dan mudah.

2. *Event Driven*

   Merupakan pengembangan dari pemrograman *top-down* (pembuatan kode yang menangani banyak pemrosesan secara satu kesatuan). Dengan *event-driven*, pemakai atau *programmer* tidak kesulitan dalam menuliskan proses pada aplikasi tertentu. *Programmer* hanya menuliskan kode program pada event tertentu yang telah dikategorikan oleh C++ Builder secara otomatis. Hal ini juga mempermudah *programmer* dalam mengenali kesalahan pengkodean, dan cepat dalam memperbaikinya. *Event-driven* bukan mencoba menggantikan pemrograman prosedural, namun melengkapi dengan kerangka yang dapat membedakan antara user interface dan proses tertentu dalam aplikasi.

3. *Case Sensitive*

   Program akan mengecek penggunaan huruf besar maupun huruf kecil.

4. *Relational Database*

Dikenalnya terminologi yang merupakan unsur-unsur pembangun konsep *database*, mulai dari Tabel, *Record*, *Field*, *Domain*, *Key* sampai dengan *Integrity*.

2.8.2. Penggunaan *database* dalam C++ Builder

Dua perihal yang harus dipertimbangkan dalam pembuatan *database*, yaitu keamanan dan otoritas. Keamanan menyangkut segala potensi *database* dalam menjaga data dari pihak-pihak yang tidak diinginkan. Sedangkan otoritas berhubungan dengan wewenang para pemakai dalam mengedit dan menggunakan program. Hal ini untuk menghindari terjadinya *concurrent* (pertentangan pengertian yang terjadi dari banyak pengguna).

Selain 2 (dua) hal diatas, aspek lainnya adalah proses *scalling*, yaitu penentuan jenis database yang paling sesuai digunakan.

Secara fisik, database terdiri atas 2 (dua) bagian utama, yaitu *Back-End* dan *Front-End*. *Back-End* merupakan engine yang mampu menghasilkan dan mengedit struktur tabel dan membentuk sistem database, dalam C++ Builder telah disediakan engine berupa BDE (*Borland Database Engine*) dan sarana aplikasinya adalah *Database Desktop*. Sedangkan *Front-End* merupakan sarana komunikasi dengan pemakai dalam membuat tampilan atau menyusun form maupun laporan yang dapat disesuaikan dengan kebutuhan dan keinginan pengguna. Dalam C++ Builder telah disediakan mulai dari komponen-komponen *Data*

Access, komponen *Data Control*, *Report Smith*, *Quick Report*, sampai dengan *Database Form Expert*.

Edisi dalam Borland C++ Builder :

1. *Personal*, untuk pemula
2. *Professional*, untuk pengembangan aplikasi yang professional, berisi semua kemampuan dasar, pemrograman database (termasuk ADO), *server web* (*VisiBroker*) dan beberapa alat bantu lainnya.
3. *Enterprise*, untuk pengembangan aplikasi *enterprise*, berisi fasilitas XML, arsitektur *three-tier* dan fasilitas-fasilitas lain.

Ada 4 (empat) data yang harus diisikan untuk membuat suatu *field* dalam *database desktop* Borland C++ Builder, yaitu :

1. *Field Name*, beberapa aturan dalam menentukan sebuah nama *field*, sebagai berikut :
   a. Panjang tidak boleh lebih dari 10 karakter.
   b. Tidak boleh ada nama *field* yang sama dalam sebuah tabel.
   c. Nama *field* tidak boleh diawali dengan spasi.
   d. Tidak boleh menggunakan karakter khusus.
   e. Hindari pemakaian *keyword* SQL, misalnya *SELECT – INSERT – DELETE* dan lain-lain.
2. *Type*, dengan hanya menekan *SpaceBar* pada *keyboard*, maka BDB akan menampilkan semua tipe yang tersedia. Cara ini mempermudah pengguna dalam menentukan tipe data yang akan diinginkan.

3. *Size*, pengguna menetukan ukuran atau panjang suatu *field*. Panjang ini berpengaruh bukan hanya pada panjang karaktek yang dimasukkan tetapi juga pada penampilan data pada DBD.

4. *Key*, menentukan kunci *field* sebagai indeks penghubung dengan tabel lain.

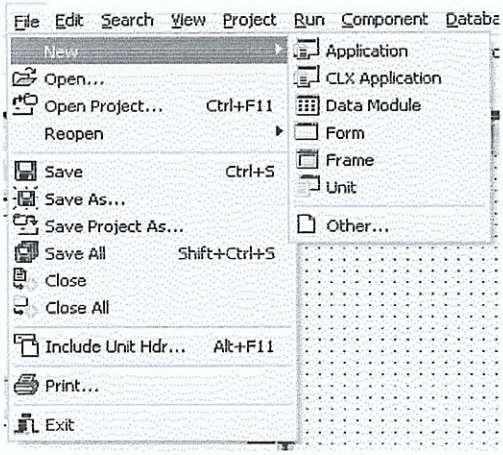
Simbol	Ukuran	Type	Keterangan
A	1 s.d 255	Alpha	Data String
N	-	Number	Bilangan
$	-	Money	Bilangan dengan dua desimal dibelakang koma
S	-128 s.d 128	Short Integer	Bilangan bulat
I	-2747483638 s.d 2747483638	Long Integer	Bilangan bulat
#	0 s.d 32	BCD	*Floating Point* presisi tinggi
D	-	Date	Tanggal
T	-	Time	Waktu
@	-	TimeStamp	Kombinasi tanggal dan waktu
M	1 s.d 240	Memo	Data teks yang panjang
F	0 s.d 240	Formatted Memo	Teks terformat
G	0 s.d 240	Graphic	Citra gambar
O	0 s.d 240	OLE	Objek OLE
L	-	Logika	Boolean
+	-	Auto Increment	Pencacah Unik
B	0 s.d 240	Binary	Data BLBO
Y	1 s.d 255	Byte	Data BLBO tidak terformat

Tabel 2.1. Tabel daftar tipe field pada tabel Paradox

2.8.3. Borland C++ Builder 6.0 Enterprise Suite

Dalam bahasa pemrograman C++ Builder versi 6.0 terdapat beberapa fasilitas baru, diantaranya :

1. Menu *File* | *New*, menu ini akan menampilkan enam tipe *project*, pilihan terakhir yaitu *Other* akan menampilkan kotak dialog *New Items*.



Gambar 2.7. Menu *File* | *New*

2. Jendela *Object TreeView*, adalah sebuah diagram pohon yang menggambarkan hubungan logis antara komponen-komponen *visual* dan *non-visual* pada *form*, data modul atau *frame*.

3. Halaman baru pada *componen palette* diantaranya *WebSnap*, *Com++*, *Indy Client*, *Indy Servers*, *Indy Misc*.

4. halaman *Additional* mempunyai beberapa komponen baru yang berguna untuk menangani *action list* yang dipakai pada menu dan *toolbar*. Komponen barunya adalah *TactionManager*, *TactionMainMenuBar*, *TactionToolBar*, dan *TcustomizeDlg*, serta *TlabeledEdit* (gabungan antara komponen *Label* dan *Edit*).

5. Menu baru pada halaman *Environment Options*, yaitu *Designer*, *Environment Variables*, *Object Inspector*, dan Internet. Penambahan ini untuk mempermudah pembuatan aplikasi *Web*.

2.8.4. Unsur pembangun *project* dalam C++ Builder

Untuk menyusun suatu proses diperlukan unsur pembentuknya, yaitu operator, dari sudut pandang ini, dari sudut pandang ini, suatu bahan proses seperti data, konstanta dan variabel akan disebut dengan *operand.* Operator akan menangani *operand* agar terbentuk operasi atau proses.

Untuk penulisan prosedur dan fungsi dalam C++ Builder disimpan dalam *runtime library* C++ Builder. *Programmer* tinggal memanggil dan mengambil yang sesuai sehingga hal ini berguna untuk menghindari penulisan kode program yang panjang dan salah.

Operator	Nama	Contoh Penggunaan
+	Penambahan	a=b+c
-	Pengurangan	a=b-c
*	Perkalian	a=b*c
/	Pembagian	a=b/c
%	Sisa pembagian	a=b%c

Tabel 2.2. Tabel operator matematika

Operator	Nama	Contoh Penggunaan
=	Penugasan	A=1999
+=	Penugasan dan penambahan	A+=100
-=	Penugasan dan pengurangan	A-=9
*=	Penugasan dan perkalian	A*=10
\=	Penugasan dan pembagian	A\=2000
&=	Penugasan AND	A&=0*0.9
|=	Penugasan OR	A|=0*99

Tabel 2.3. Tabel operator penugasan

Operator	Nama
&&	AND
!	NOT
||	OR

Tabel 2.4. Tabel operator logika

Operator	Nama
= =	Sama dengan
! =	Tidak sama dengan
<	Lebih kecil
>	Lenih besar
< =	Lebih kecil atau sama dengan
> =	Lebih besar atau sama dengan

Tabel 2.5. Tabel operator relasional

Operator	Nama	Contoh Penggunaan
*	Indirection	Char b = * c
&	Address	Char * b = & c
~	Bitwise NOT	Ada & = ~ 0 * 0.5
!	Logika NOT	If (! Ada) { ... }
+ +	Penambahan	Panjang + +
- -	Pengurangan	Lebar - -

Tabel 2.6. Tabel operator *unary*

Nama	Kegunaan
StrTod	Mengkonversi dari *String* ke *Double*
StrToInt	Mengkonversi dari *String* ke *Integer*
StrToL	Mengkonversi dari *String* ke *Long*
Copy	Menyalin sebuah *String*
DateTimeToStr	Mengkonversi dari *Date* ke *String*

Tabel 2.7. Tabel prosedur penulisan konversi