BAB II

TINJAUAN PUSTAKA

2.1 Profil Tempat Kerja Praktek

PT. Dirgantara Indonesia

PT. Dirgantara Indonesia merupakan Badan Usaha Milik Negara Strategis, yang bergerak pada bidang dirgantara dan pertahanan, PT. DI didirikan pada 24 Agustus 2000, berubah menjadi PT Dirgantara Indonesia di Bandung, Kantor Pusatnya berada di Bandung, Indonesia.

Produk-produk yang dihasilkan oleh PT. DI diantaranya Pesawat komersial, Pesawat militer, Komponen pesawat, Servis pesawat, Pertahanan, Teknik *(engineering*), PT. DI mempunyai jumlah karyawan kurang lebih 3.720 (2004).Website resmi PT. DI adalah www.indonesian-aerospace.com

PT. Dirgantara Indonesia (DI) (nama bahasa Inggris: *Indonesian Aerospace Inc*.) adalah industri pesawat terbang yang pertama dan satu-satunya di Indonesia dan di wilayah Asia Tenggara. Perusahaan ini dimiliki oleh Pemerintah Indonesia. DI didirikan pada 26 April 1976 dengan nama PT. Industri Pesawat Terbang Nurtanio dan BJ Habibie sebagai Presiden Direktur. **Industri Pesawat Terbang Nurtanio** kemudian berganti nama menjadi **Industri Pesawat Terbang Nusantara** (**IPTN**) pada 11 Oktober 1985. Seteleah direstrukturisasi, IPTN kemudian berubah nama menjadi Dirgantara Indonesia pada 24 Agustus 2000.

2.1.1 Sejarah Instansi

Industri Pesawat Terbang Nusantara (1976-2000)

PT. Dirgantara Indonesia merupakan Badan Usaha Milik Negara Strategis, yang bergerak pada bidang dirgantara dan pertahanan23 Agustus 1976, berdasar

akte notaris 15 pada 26 April 1976 di Jakarta, berubah menjadi PT Dirgantara Indonesia di Bandung, Kantor Pusatnya berada di Bandung, Indonesia.

Produk-produk yang dihasilkan oleh PT. DI diantaranya Pesawat komersial, Pesawat militer, Komponen pesawat, Servis pesawat, Pertahanan, Teknik (*engineering*), PT. DI mempunyai jumlah karyawan kurang lebih 16.000.

Dirgantara Indonesia tidak hanya memproduksi berbagai pesawat tetapi juga helikopter, senjata, menyediakan pelatihan dan jasa pemeliharaan *(maintenance service*) untuk mesin-mesin pesawat. Dirgantara Indonesia juga menjadi sub-kontraktor untuk industri-industri pesawat terbang besar di dunia seperti *Boeing*, *General Dynamic*, *Fokker* dan lain sebagainya. Dirgantara Indonesia pernah mempunyai karyawan sampai 16 ribu orang. Karena krisis ekonomi banyak karyawan yang dikeluarkan dan karyawannya kemudian menjadi berjumlah sekitar 4000 orang.

Pada awal hingga pertengahan tahun 2000-an Dirgantara Indonesia mulai menunjukkan kebangkitannya kembali, banyak pesanan dari luar negeri seperti Thailand, Malaysia, Brunei, Korea, Filipina dan lain-lain. Meskipun begitu, karena dinilai tidak mampu membayar utang berupa kompensasi dan manfaat pensiun dan jaminan hari tua kepada mantan karyawannya, DI dinyatakan pailit oleh Pengadilan Niaga pada Pengadilan Negeri Jakarta Pusat pada 4 September 2007. Namun pada tanggal 24 Oktober 2007 keputusan pailit tersebut dibatalkan.

NC Programming adalah suatu fungsi yang menghasilkan product program NC , yang hasilnya dipersyaratkan sesuai dengan *design*.

1. NC system and support

Bagian dari *NC Programming* yang mempunyai fungsi :

Record , update dan *filing* data .

a. Membuat ,menyiapkan *system* dan *aplication* yang diperlukan

b. Membackup data

c. Memelihara *hardware , software* dan *system* yang ada.

2. NC Part Programming

Bagian dari *NC Programming* yang mempunyai fungsi :

a. Membuat *machining plan* program NC (step operasi)

b. Membuat program NC (Catia)

c. Mensimulasi program NC (vericut)

d. Membuat MCD dan NCOD

NC (Numerical Control)

MCD (Machine Code Data).

NCOD (Numerical Control Operator Document)

2.1.2 Logo Instansi



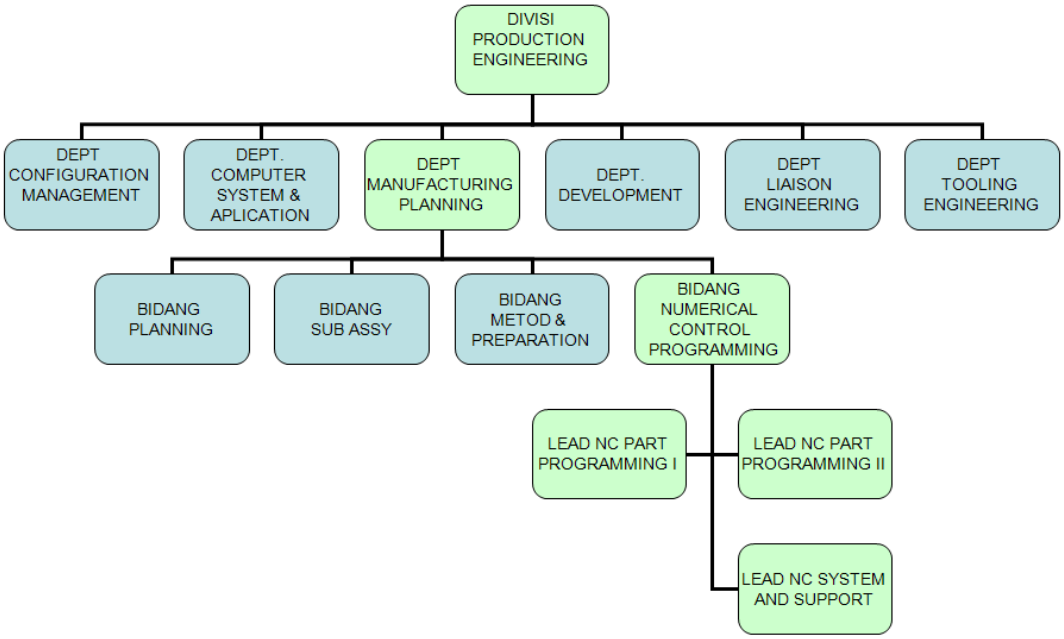
Gambar 2.1 Logo Instansi

2.1.3 Badan Hukum Instansi

PT. Dirgantara Indonesia merupakan salah satu perusahaan BUMN milik pemerintah yang bergerak di bidang Industri pesawat terbang.

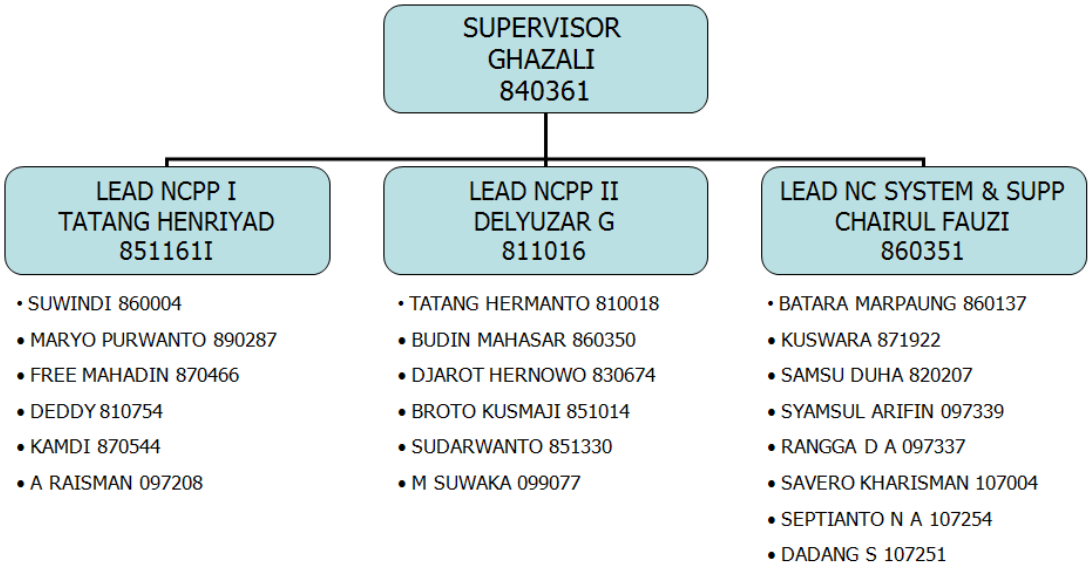
2.1.4 Struktur Organisasi dan *Job Description*

STRUCTURE ORGANISASI DIVISI *PRODUCTION ENGINEERING ARCHITECTURE*



Gambar 2.2 Struktur Organisasi Divisi Produksi

STRUCTURE ORGANISASI *NC PROGRAMMING*



Gambar 2.3 Struktur Organisasi *NC programming*

2.2 Landasan Teori

2.2.1 Pengertian Sistem

Suatu sistem pada dasarnya marupakan suatu susunan yang teratur dari kegiatan yang berhubungan satu sama lain dan prosedur-prosedur yang berkaitan yang melaksanakan dan mempermudah pelaksanaan kegiatan utama dari suatu organisasi.

> *"Sistem adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul, bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu"*. (Andi Kristanto, 2007 : 1)

Sistem itu sendiri mempunyai tujuan yang sama untuk menghasilkan sesuatu yang lebih bermanfaat ada yang menyebutkan mencapai suatu tujuan (*goal*) dan ada yang menyebutkan untuk mencapai sasaran (*object*). Jadi dapat disimpulkan pengertian sistem adalah kumpulan dari berbagai macam elemen yang berbeda namun mempunyai tujuan yang sama untuk mencapai sasaran atau *objek.*

2.2.1.1 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu. Berikut ini adalah karakteristik sistem tersebut :

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem.

2. Batas Sistem

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas suatu sistem menujukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan harus tetap dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan menggangu kelangsungan hidup dari sistem.

4. Penghubung Sistem

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu *subsistem* ke *subsistem* yang lainnya. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

5. Masukan Sistem

Masukan adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran Sistem

Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem.

7. Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran Sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Sasaran dari sistem sangat menetukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

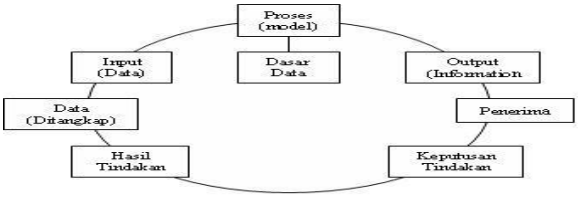
2.2.2 Pengertian Informasi

Informasi merupakan sesuatu yang harus dimiliki oleh suatu perusahaan atau organisasi untuk bisa tetap eksis dan bertahan, karena kurangnya informasi yang dimiliki akan mengakibatkan perusahaan atau organisasi akan hancur dan berakhir.

"Informasi adalah rangkaian data yang mempunyai sifat sementara, tergantung dengan waktu, mampu memberi kejutan atau surprise pada yang menerimanya. Informasi dapat juga dikatakan sebagai data yang telah diproses, yang mempunyai nilai tentang tindakan atau keputusan". (Witarto, 2004:9).

2.2.2.1 Siklus Informasi

Data merupakan bentuk yang masih mentah sehingga perlu diolah melalui suatu model untuk dihasilkan informasi. Data yang diolah melalui suatu model menjadi informasi, penerima akan menerima informasi, membuat suatu keputusan dan mengambil tindakan yang menghasilkan suatu tindakan yang membuat sejumlah data kembali. Data tersebut ditangkap sebagai input dan akan diproses kembali lewat suatu model dan seterusnya membuntuk suatu siklus.



Gambar 2.4 *Siklus Informasi*

2.2.2.2 Kualitas Informasi

Kualitas dari suatu informasi tergantung pada tiga hal, yaitu :

1. Akurat, berarti informasi harus bebas dari kesalahan dan tidak bias atau menyesatkan.
2. Tepat pada waktunya, berarti informasi yang dating pada penerima tidak boleh terlambat. Informasi yang usah tidak mempunyai nilai lagi.
3. Relevan, berarti informasi tersebut harus mempunyai mafaat untuk pemakainya.

2.2.2.3 Pengertian Sistem Informasi

"Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasasi, dan menyediakan pihak luar tertentu dengan laporan - laporan yang diperlukan". (Jogiyanto, 2005 :11)

Sistem informasi dapat dikatakan sebagai suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam suatu sistem organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi. Kegiatan sistem informasi mencakup :

1. *Input*, menggambarkan 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 proses diatas tesebut.
4. Penyimpanan, suatu kegiatan untuk memelihara dan menyajikan data.
5. Kontrol, suatu aktifitas untuk menjamin bahwa sistem informasi tersebut sesuai yang diharapkan.

2.2.2.4 Komponen Sistem Informasi

Sistem informasi dapat terdiri dari komponen-komponen, yaitu :

1. Blok Masukan

Input mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukan, 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, mengasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem keseluruhan.

5. Blok Basis Data

Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanpulasinya. Data perlu disimpan di dalam dasar data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa, supaya informasi yang dihasilkan berkualitas.

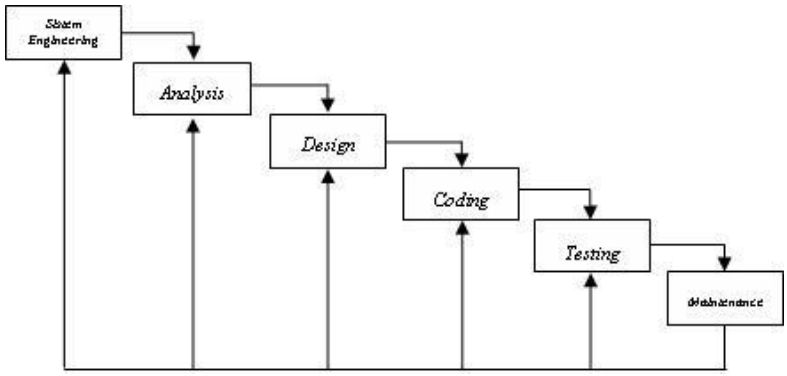
6. Blok Kendali

Untuk supaya sistem informasi dapat berjalan sesuai dengan yang diingikan, maka perlu diterapkan pengendalian-pengendalian. Beberpa pengendali perlu dirancang dan diterapkan untuk menyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah atapun bila terlanjur kesalahan-kesalahan dapat langsung cepat diatasi.

2.2.3 Metodologi Pengembangan Perangkat Lunak

Berikut ini beberapa metodologi pengembangan perangkat lunak yang umum digunakan, yaitu sebagai berikut :

2.2.3.1 Metode Waterfall



Gambar 2.5 Metode Waterfall

Metode ini bisa juga disebut dengan *classic life cycle*. Metode ini membutuhkan pendekatan sistematis dan sekuensial dalam pengembangan perangkat lunak, dimulai dari tingkat sistem dan kemajuan melalui analisis, desain, pengkodean, pengetesan dan pemeliharaan. Pemodelan ini menyangkut aktivitas berikut:

a. Rekayasa dan Pemodelan Sistem/Informasi (*System/Information Engineering and Modeling*). Karena perangkat lunak adalah bagian dari sistem yang lebih besar, pekerjaan dimulai dari pembentukan kebutuhan-kebutuhan dari semua elemen sistem dan mengalokasikan suatu subset ke dalam pembentukan perangkat lunak. Hal ini penting, ketika perangkat lunak harus berkomunikasi dengan *hardware,* orang dan basis data. Rekayasa dan pemodelan sistem menekankan pada pengumpulan kebutuhan pada level sistem dengan sedikit perancangan dan analisis. Tahap ini juga kadang disebut dengan *Project Definition.*
b. Analisis Kebutuhan Perangkat Lunak (*Software Requirements Analysis*). Proses pengumpulan kebutuhan diintensifkan ke perangkat lunak. Harus dapat dibentuk domain informasi, fungsi yang dibutuhkan, performansi dan antarmuka. Hasilnya harus didokumentasikan dan di-*review* ke pelanggan.

c. Desain (*Design*). Proses desain mengubah kebutuhan-kebutuhan menjadi bentuk karakteristik yang dimengerti perangkat lunak sebelum dimulai penulisan program. Desain ini harus didokumentasikan dengan baik dan menjadi bagian konfigurasi perangkat lunak.
d. Penulisan Program (*Coding*). Desain tadi harus diubah menjadi bentuk yang dimengerti mesin (komputer). Maka dilakukan langkah penulisan program. Jika desain-nya detil, maka *coding* dapat dicapai secara mekanis.
e. *Testing*. Setelah kode program selesai dibuat, dan program dapat berjalan, testing dapat dimulai. *Testing* difokuskan pada logika internal dari perangkat lunak, fungsi eksternal, dan mencari segala kemungkinan kesalahan. Dan memeriksa apakah sesuai dengan hasil yang diinginkan.
f. *Support/Maintenance*. Perangkat lunak setelah diberikan pada pelanggan, mungkin dapat ditemui error ketika dijalankan dilingkungan pelanggan. Atau mungkin pelanggan meminta penambahan fungsi, hal ini menyebabkan faktor maintenance (pemeliharaan) ini menjadi penting dalam penggunaan metode ini. Pemeliharaan ini dapat berpengaruh pada semua langkah yang dilakukan sebelumnya.

Berikut ini beberapa kelebihan yang dimiliki metode *Waterfall*, diantaranya adalah :

1. Metode ini masih lebih baik digunakan walaupun sudah tergolong “kuno”, daripada menggunakan pendekatan asal-asalan.
2. Selain itu, metode ini juga masih masuk akal jika kebutuhan sudah diketahui dengan baik.

Berikut ini beberapa kekurangan yang dimiliki metode *Waterfall*, diantaranya adalah :

1. Pada kenyataannya, jarang mengikuti urutan sekuensial seperti pada teori. Iterasi sering terjadi menyebabkan masalah baru.
2. Sulit bagi pelanggan untuk menentukan semua kebutuhan secara eksplisit.

3. Pelanggan harus sabar, karena pembuatan perangkat lunak akan dimulai ketika tahap desain sudah selesai. Sedangkan pada tahap sebelum desain bisa memakan waktu yang lama.
4. Kesalahan di awal tahap berakibat sangat fatal pada tahap berikutnya.

2.2.5 Alat Perancangan Perangkat Lunak

Merupakan alat yang digunakan dalam perancangan sutu perangkat lunak, yang gunanya untuk menyusun rancangan suatu perangkat lunak yang akan dibangun.

2.2.5.1 *Flowmap*

Digunakan untuk mendefinisikan hubungan antara bagian (pelaku proses), proses (manual atau berbasis komputer) dan aliran data dalam bentuk dokumen keluaran dan masukan. Adapun simbol-simbol yang digunakan dalam *flowmap*, dapat dilihat dalam daftar simbol.

2.2.5.2 *Flowchart*

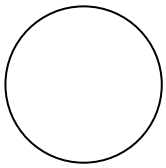
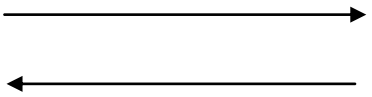
Flowchart adalah alat untuk menganalisa proses yang merupakan gambaran dalam bentuk diagram alir dari algoritma-algoritma dalam suatu program, yang menyatakan arah alur program tersebut. Hal tersebut memungkinkan untuk memecah proses menjadi kejadian-kejadian individual atau aktifitas untuk menunjukan secara singkat hubungan diantaranya. Konstruksi *flowchart* memungkinkan pengertian lebih baik kepada proses dan pengertian yang lebih baik terhadap proses akan membawa kepada perbaikan pengembangan suatu sistem. Adapun simbol-simbol yang digunakan dalam *flowchart*, dapat dilihat dalam daftar simbol.

2.2.5.3 *DFD (Data Flow Diagram)*

Pemodelan sistem informasi memerlukan alat bantu untuk merekomendasikannya, salah satunya adalah dengan diagram konteks. *Diagram konteks* adalah diagram alir data tingkat tinggi yang menggambarkan hubungan antara sistem dengan lingkungan luarnya dengan tujuan untuk menggambarkan

sebuah sistem secara umum, mendefinisikan sistem dari awal sampai akhir baik yang masuk ke dalam sistem maupun yang keluar dari sistem tanpa menggambarkan proses dan simpanan data secara detail.

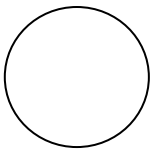
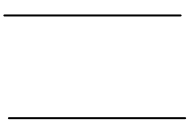
Tabel 2.1 Simbol-simbol Diagram Konteks

Simbol	**Keterangan**
	Kesatuan luar (*external entity*) yang menggambarkan sumber atau tujuan.
	Kegiatan yang dilakukan orang atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk menghasilkan informasi yang keluar dari proses.
	Menunjukkan arus data yang berupa masukan untuk sistem atau dari sistem.

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 merupakan 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.

Tabel 2.2 Simbol-simbol DFD

Simbol	**Keterangan**
 Entitas	Kesatuan luar dilingkungan luar. Sistem dapat berupa uang, organisasi atau sistem lainnya yang berada dilingkungan luarnya yang akan memberikan *input* atau menerima *output* dari suatu sistem.
 Arus Data (flow)	Menunjukkan arus dari data yang dapat berupa masukkan untuk system atau hasil dari proses system.
 Proses	Proses ini mengubah 1 atau lebih input menjadi output. Nama proses dituliskan dengan suatu kata, singkatan atau kalimat sederhana
 Simpanan Data	Simpanan data dapat berupa file, *database*, arsip, tabel

Syarat dan Fungsi *DFD*

Syarat-syarat pembuatan sebuah *DFD,* antara lain :

1. Pemberian nama untuk tiap komponen *DFD*
2. Pemberian nomor pada komponen proses
3. Penggambaran *DFD* sesering mungkin agar nyaman dilihat
4. Penghindaran penggambaran *DFD* yang rumit
5. Pemastian *DFD* yang dibentuk itu konsiten secara logika

Adapun fungsi dari *DFD* adalah sebagai berikut :

1. Membantu para analis sitem meringkas informasi tentang sistem, mengetahui hubungan antar *sub-sub* sistem, dan membantu perkembangan aplikasi secara efektif.
2. *DFD* berfungsi sebagai alat komunikasi yang baik antara pemakai dan analis sistem.
3. Menggambarkan sejumlah batasan otomasi untuk pengembangan alternatif sistem fisik.

Dalam mengembangkan suatu aliran data atau proses yang terjadi di dalam sistem *data flow diagram* menggunakan simbol-simbol yang memiliki arti tersendiri dalam menerangkan :

a. Eksternal Entity

Eksternal entity dapat merupakan kesatuan (*entity*) dilingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya, yang memberikan *input-output* dari sistem.

b. Data Flow

Arus data ini mengatur diantara proses, simpan data, dan kesatuan luar. Arus data ini menujukkan arus data yang dapat berupa masukan sistem atau hasil proses sistem.

c. Proses

Untuk *physical data flow diagram* (*PDFD*), data dilakukan oleh orang, mesin atau komputer. Sedangkan untuk *logical data flow diagram* (*LDFD*), suatu proses hanya menujukkan proses dari komputer.

d. Penyimpanan Data

Simpanan data (*data store*) merupakan tempat penyimpanan data. Simpanan data dari DFD disimbolkan dengan sepasang garis horizontal paralel.

Konsep dasar *DFD* dapat dilakukan dengan analisa *Top Down*, yaitu pemecahan sistem yang besar menjadi beberapa sub-sub sistem yang lebih kecil *DFD* terdiri dari :

a. *Context Diagram*

Diagram konteks yaitu diagram yang menunjukkan batas dan jangkauan dari sistem informasi yang dibuat. Merupakan gambaran sistem secara garis besar dan hanya memperlihatkan kelompok data *input* dan *output*. Konteks diagram merupakan *level* teratas dari diagram arus data .

b. Middle Level

Merupakan pemecahan dari tiap–tiap proses yang mempunyai fungsi sama. Pada *middle level* diagaram 0 dipecah menjadi diagram 1,2,3 dan seterusnya yang merupakan penguraian dari *diagram konteks*.

b. *Lowest Level* (DFD Level Terendah)

Diagram yang menunjukkan proses yang lebih detail dari level sebelumnya.Merupakan pemecahan dari *data flow* yang ada pada *middle level*.Pemecahan tersebut masih tetap mempunyai fungsi yang sama dari *level*

sebelumnya. Untuk *Lowest Level*, pemberian nomor diagram terdiri dari bagian *middle level.*

2. *Data Dictionary* (Kamus Data)

Kamus data adalah pusat penyimpanan atau dokumentasi manual dari semua informasi yang terdapat pada sistem informasi atau program aplikasi yang digunakan sebagai sumber informasi pada proses *systems design*. Kamus data dapat terdiri dari :

a. KD elemen data / *Field*
b. KD alur data
c. KD penyimpanan data/ *table*
d. KD proses pengolahan data
e. KD *record*
f. KD laporan

2.2.5.4 Konsep Basis Data

Basis Data (*Database*) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut.(*http://id.wikipedia.org/wiki/Basis_data :10/12/2010).*

Database merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi bagi para pemakai. Sistem basis data adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lainnya. Beberapa istilah dalam *database*, yaitu :

a. Entity
b. Atribut
c. Data Value
d. Record

Entity Relationship Diagram

Diagram Entity Relationship yang berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan *attribut-attribut* yang merepresentasikan seluruh fakta dari dunia nyata yang kita tinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan *Diagram Entity Relationship (Diagram E-R).* Antar *entitas* selalu ada tiga hubungan biner, yaitu :

1. Satu ke Satu (*One To One Relationship*)

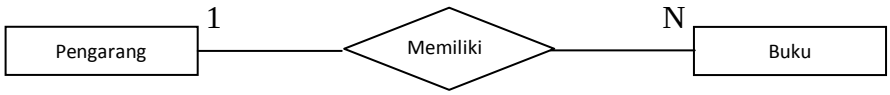
Hubungan antara entity pertama dengan *entity* kedua adalah satu berbanding satu.



Gambar 2.6 ERD One to One

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

Himpunan *entity* pertama dapat berhubungan satu atau lebih *entity* pada himpunan kedua, tetapi tidak sebaliknya.



Gambar 2.7 ERD One to Many

3. Banyak ke Banyak (*Many To Many*)

Setiap entity pada himpunan *entity* pertama dapat berhubungan dengan banyak *entity* pada himpunan *entity* kedua, begitu pula sebaliknya.



Gambar 2.8 ERD Many to Many

2.2.5.5 VB (VISUAL BASIC)

Visual Basic 6.0 merupakan sebuah pengembangan terakhir dari bahasa *BASIC*. *BASIC* dirancang tahun 1950-an dan ditujukan untuk dapat digunakan oleh para programmer pemula. Sedangkan *Visual Basic* itu sendiri memiliki pengertian kata*Visual* dalam nama pemrograman ini mewakili pada metode untuk membuat *Graphical User Interface (GUI)*. Dengan hanya mengatur letak dari elemen-elemen sebuah interface tanpa menuliskan baris kode yang banyak. Kata *BASIC* sendiri merupakan kependekan dari *Beginners All-Purpose Symbolic Instruction Code*. Di dalam bahasa *Visual Basic* telah dilengkapi dengan beberapa ratus pernyataan, fungsi dan kata kunci, banyak di antaranya berkaitan langsung dengan *GUI* dari windows, selain itu *Visual Basic 6,0* masih memiliki keistimewaan utama dari*Visual Basic 6,0* di antaranya seperti :

a. Memiliki *compiler* andal yang dapat menghasilkan *file executable* yang lebih cepat dan lebih efisien dari sebelumnya.

b. Memiliki beberapa tambahan sarana *wizard* yang baru.

Wizard adalah sarana yang mempermudah dalam pembuatan aplikasi dengan mengotomatisasi tugas-tugas tertentu.

c. Teknologi *ActiveX*^{Tm} memungkinkan kita untuk memakai fungsi-fungsi yang disediakan dari aplikasi lain, seperti pengolahan kata *Microsoft Word, Microsoft Excel Spreadsheet,* dan aplikasi windows lainya..

d. *Visual Basic* tersedia dalam tiga edisi setiap peralatan disiapkan untuk memenuhi kebutuhan spesifik dalam setiap pembuatan aplikasi. Adapun ketiga edisi dari *Visual Basic* tersebut yaitu *Visual Basic Learning Edition, Visual Basic Profesional Edition, dan Visual Basic Entreprise Edition*.

Operator Dalam *Visual Basic*

Operator yang biasa digunakan dalam pemrograman*Visual Basic* diantaranya adalah :

a. Operator Matematika

Penggunaan *Operator Matematika* lebih ditujukan untuk pembuatan rumus atau formula. Rumus atau *formula* adalah pernyataan yang menggabungkan angka, *Variable, Operator*, dan kata kunci untuk membuat suatu nilai baru.

b. Operator Perbandingan

Operator perbandingan digunakan untuk membandingkan dua *variable* atau objek.

c. Operator Logika

Operator logika digunakan untuk membandingkan dua *expresi*.

Fungsi Dalam *Visual Basic*

Fungsi yang digunakan dalam pemrograman antara lain adalah fungsi waktu dan *string*.

a. Fungsi Waktu

Visual basic menggunakan fungsi-fungsi *internal* untuk mengolah waktu. Fungsi-fungsi ini digunakan di antaranya untuk menampilkan tanggal dan jam saat ini, selain itu dapat untuk menghitung selisih waktu dan tanggal.

b. Fungsi *String*

Fungsi *String* adalah fungsi-fungsi yang digunakan untuk penanganan dan manipulasi *string*. Fungsi-fungsi ini diantaranya adalah untuk menghitung jumlah paragraf dalam sebuah *string*, mengambil nilai dari sebagian *string*, dan sebagainya.

2.2.5.6 Microsoft Acces (Digunakan sebagai *database*)

Microsoft Access 2007 merupakan salah satu pengolahan *DataBase* yang cukup canggih dengan berbagai kemudahan yang ada seperti pengaturan data, pembuatan form, pembuatan laporan, menyaring data dan lain-lain.

a. Membuat *DataBas*e di Microsoft Acces

Langkah- langkah Cara Membuat DataBase di Microsoft Access 2007 adalah pertama kali membuka Microsoft Access 2007, Kemudian Klik ikon ***Blank DataBase***, Kemudian beri nama *DataBase* yang dibuat sesuai dengan nama file yang diinginkan.

b. Cara Membuat Form Membuat *DataBase* di Microsoft Acces.

Setelah DataBase dibuat, kemudian klik ikon ***Design View,*** Setelah ikon ***Design View*** ditekan maka akan muncul kolom ***Save As***, kolom Save As ini dapat diberi nama sesuai yang diinginkan, lalu tekan **Ok,** Kemudian setelah langkah-

langkah di atas dilakukan maka akan muncul tabel dan masukan nama Field dan Tipe datanya Setelah Data dibuat, lalu klik ikon ***View,*** Setelah mengklik ikon ***View*** maka akan muncul Tabel dan masukan data yang diinginkan ke dalam Tabel tersebut, Lalu untuk membuat ***Form***, tekan ***Create*** lalu klik **Form**.