BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

Pada landasan teori ini diuraikan sejumlah teori untuk membantu dan memecahkan permasalahan yang ada. Beberapa landasan teori tersebut meliputi konsep dasar dan definisi-definisi yang berkaitan dengan perangkat lunak yang digunakan serta faktor-faktor pendukung dalam pelaksanaan perancangannya.

2.1.1 Konsep Dasar Sistem

Sistem definisikan sebagai suatu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang berinteraksi untuk mencapai tujuan [4]. Suatu sistem dapat terdiri dari sistem bagian (*subsystems*). Sebagai misal, sistem komputer dapat terdiri dari subsistem yang lebih kecil lagi atau terdiri dari komponen-komponen. Subsistem perangkat keras (*hardware*) dapat terdiri dari alat masukkan, alat pemroses, alat keluaran dan simpanan luar. Subsistem-subsistem saling berinteraksi dan saling berhubungan membentuk satu kesatuan hingga tujuan/sasaran sistem tersebut dapat tercapai. Interaksi dari subsistem-subsistem sedemikian rupa, sehingga dicapai suatu kesatuan yang terpadu atau terintegrasi (*integrated*). Dapat dibayangkan bagaimana seandainya suatu sistem komputer masing-masing komponennya saling bekerja sendiri-sendiri tidak terintegrasi, maka tujuan dari sistem komputer tersebut tidak akan tercapai.

2.1.1.1 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu[4] :

a) Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponem sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem tidak perduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu. Dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai suatu sistem yang lebih besar yang disebut dengan *supra system*.

b) Batas Sistem

Batas sistem (*boundary*) 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 menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

c) Lingkungan Luar Sistem

Lingkungan luar (*environmant*) 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 dengan demikian harus tetap dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan

dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

d) Penghubung Sistem

Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya yang mengalir dari suatu subsistem ke subsistem yang lainnya. Keluaran (*output*) dari suatu subsistem akan menjadi masukan (*input*) untuk subsistem yang lainnya dengan melalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

e) Masukan Sistem

Masukan (*input*) adalah energi yang dimasukkan kedalam sistem. Masukan dapat berupa masalah perawatan (*maintenance input*) dan masukkan sinyal (*signal input*). *maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran.

f) Keluaran Sistem

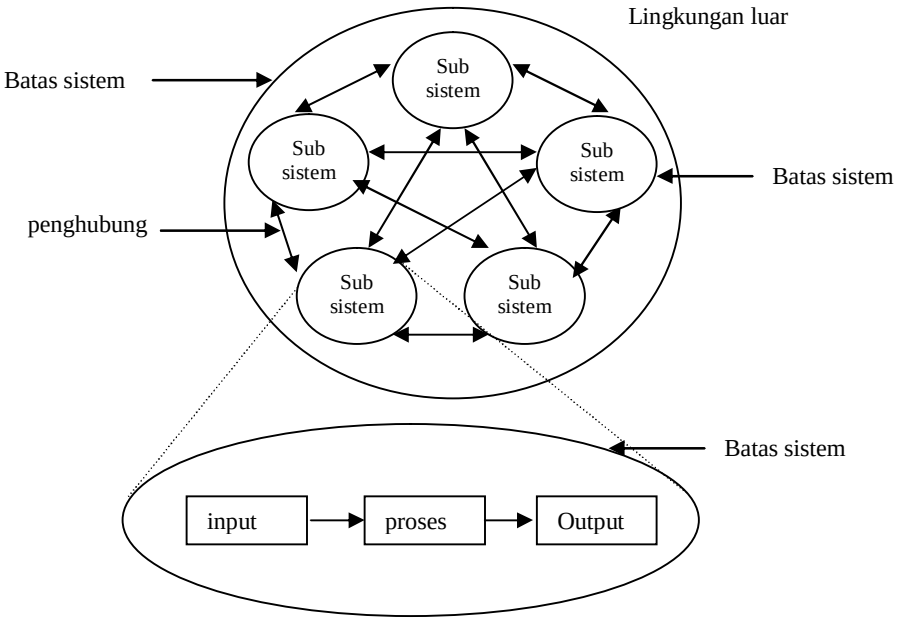
Keluaran (*output*) adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukkan untuk subsistem yang lain atau kepada supra sistem. Misalnya untuk sistem komputer, panas yang dihasilkan adalah keluaran yang tidak berguna dan merupakan hasil sisa pembuangan, sedang informasi adalah keluaran yang dibutuhkan.

g) Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukkan menjadi keluaran. Suatu sistem produksi akan mengolah masukkan berupa bahan baku dan bahan-bahan yang lain menjadi keluaran berupa barang jadi. Sistem akuntansi akan mengolah data-data transaksi menjadi laporan-laporan lain yang dibutuhkan oleh manajemen.

h) Sasaran Sistem

Suatu sistem pasti mempunyai tujuan (g*oal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukkan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.



Gambar II.1 Karakteristik Suatu Sistem

2.1.1.2 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandangan, diantaranya adalah sebagai berikut[4]:

1. Sistem diklasifikasikan sebagai sistem abstrak (*abstrack system*) dan sistem phisik (*physical system*). Sistem abstrak adalah sistem yang berupa pemikiran/ide-ide yang tidak tampak secara fisik. Sistem fisik merupakan sistem yang ada secara fisik.
2. Sistem diklasifikasikan sebagai sistem alamiah (*natural system*) sistem buatan manusia (*human made system*). Sistem alamiah adalah sistem yang terjadi melaui proses alam, tidak dibuat manusia. Sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan

interaksi antara manusia dengan mesin disebut dengan *human-machine system* atau ada yang menyebutnya dengan *man machine system.*

3. Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*). Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi antara bagiannya sudah dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilistik.
4. Sistem diklasifikasikan sebagai sistem tertutup (*closed system*) dan sistem terbuka (*open system*). Sistem tertutup merupakan sistem yang tidak berhubungan dengan lingkungan luarnya. Secara teoritis sistem tertutup ini ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup). Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukkan dan menghasilkan keluaran untuk lingkungan luar/subsistem yang lainnya.

2.1.2 Konsep Dasar Informasi

Informasi ibarat darah yang mengalir didalam tubuh suatu organisasi, sehingga informasi ini sangat penting di dalam suatu organisasi. Suatu sistem yang kurang mendapatkan informasi akan luruh, kerdil dan akhirnya berakhir.

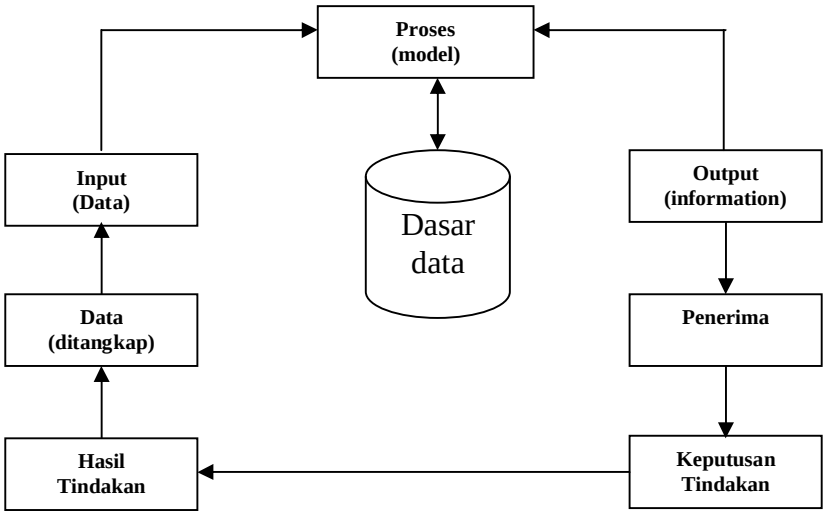
Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya [4]. Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal datum atau data-item. Data adalah

kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejdian-kejadian (*event*) adalah sesuatu yang terjadi pada saat tertentu. Didalam dunia bisnis, kejadian-kejadian nyata yang sering terjadi adalah perbuatan dati suatu nilai yang disebut dengan transaksi.

2.1.2.1 Siklus Informasi

Data merupakan bentuk yang masih mentah yang belum dapat berceritera banyak, sehingga perlu diolah lebih lanjut. Data diolah melalui suatu model untuk dihasilkan informasi[1].

Data yang diolah melalui suatu model menjadi suatu informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus.



Gambar II.2 Siklus Informasi

2.1.2.2 Kualitas Informasi

Kualitas dari suatu informasi (*quality of information*) tergantung dari tiga hal, yaitu informasi harus akurat (*accurate*), tepat waktunya (*timeliness*) dan relevan (*relevance*).

1. Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan (*noise*) yang dapat merubah atau merusak informasi tersebut.
2. Tepat pada waktunya, berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Bila pengambilan keputusan terlambat, maka dapat berakibat fatal untuk organisasi. Dewasa ini mahalnya nilai informasi disebabkan harus cepatnya nilai informasi tersebut didapat, sehingga diperlukan teknologi-teknologi mutakhir untuk mendapatkan, mengolah dan mengirimkannya.
3. Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya, dengan cara mengurangi ketidakpastian, menaikkan kemampuan untuk memprediksi, atau menegaskan ekspektasi semula.

2.1.2.3 Nilai Informasi

Nilai dari informasi (*Value of Information)* ditentukan dari dua hal yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Akan tetapi perlu diperhatikan bahwa informasi yang digunakan didalam suatu sistem

informasi umumnya digunakan untuk beberapa kegunaan. Sebagian informasi tidak dapat ditaksir keuntungannya dengan suatu nilai tetapi dapat ditaksir nilai keefektifannya.

2.1.3 Konsep Dasar Sistem Informasi

Informasi merupakan hal yang sangat penting didalam mengambil keputusan. Informasi tersebut didapatkan dari sistem informasi (*information system*) atau disebut juga dengan *processing systems* atau *information processing systems* atau *information generating systems*[4].

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan [4].

2.1.3.1 Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebutnya dengan istilah blok bangunan (*building block*), yaitu blok masukan (*input block*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technology block*), blok basis data (*database block*), dan blok kendali (*control block*)[4]. Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan utuk mencapai sasarannya.

a) Blok Masukan

Input mewakili data yang masuk kedalam sistem informasi, input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

b) Blok Model

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

c) 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.

d) Blok Teknologi

Teknologi merupakan “kotak alat” (*tool-box*) dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari tiga bagian utama, yaitu teknisi (*humanware* atau *brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*). Teknisi dapat berupa orang-orang yang mengetahui komputer, pemrogram, operator pengolah kata, spesialis telekomunikasi, analis sistem, penyimpan data dan lain sebagainya.

e) Blok Basis Data

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

f) Blok Kendali

Banyak hal yang dapat merusak sistem informasi, seperti misalnya bencana alam, api, temperatur, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri, kesalahan-kesalahan, ketidak efisienan, sabotase dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

2.1.4 Basis Data

Basis data terdiri dari dua kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

Basis data didefinisikan dari sejumlah sudut pandang seperti: [2]

a. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sesemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
c. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Prinsip utama Basis data adalah pengaturan data/arsip. Dan tujuannya adalah kemudahan dan kecepatan dalam pengambilan kembali data/arsip.

2.1.5 *Database Management System* (DBMS)

Manajemen Sistem Basis Data (*Database Managgement System* / DBMS) adalah perangkat lunak yang didesain untuk membantu dalam hal pemeliharaan dan utilitas kumpulan data dalam jumlah besar[2]. DBMS dapat menjadi alternatif penggunaan secara khusus untuk aplikasi, semisal penyimpanan data dalam file dan menulis kode aplikasi yang spesifik untuk pengaturannya.

Penggunaan DBMS untuk suatu aplikasi tergantung pada kemampuan dan dukungan DBMS yang beroperasi secara efisien. Sehingga agar bisa menggunakan DBMS dengan baik, perlu diketahui cara kerja dari DBMS tersebut. Pendekatan yang dilakukan untuk menggunakan DBMS secara baik meliputi : implementasi DBMS dan arsitektur secara mendetail untuk dapat memahami desain dari suatu basis data

2.1.6 Normalisasi

Normalisasi merupakan tahapan perancangan dalam membangun basis data relasional yang tidak secara langsung berkaitan dengan model data, tetapi dengan menerapkan sejumlah aturan dan kriteria standar untuk menghasilkan struktur tabel yang normal. Adapun langkah-langkah untuk menghasilkan struktur tabel yang normal, diantaranya yaitu[2] :

1. Bentuk *Unnormallized*

   Bentuk *unnormal* adalah bentuk tabel dengan mencantumkan semua *field* data yang ada.

2. Bentuk Normal Pertama / 1 NF (*First Normal Form*)

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

3. Bentuk Normal Kedua / 2 NF (*Second Normal Form*)

   Bentuk normal kedua terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam *key primer* memiliki ketergantungan fungsional pada *key primer* secara utuh.

4. Bentuk Normal Ketiga / 3 NF (*Third Normal Form*)

   Bentuk normal ketiga terpenuhi jika dan hanya jika semua atribut bukan kunci memiliki *dependensi* transitif terhadap kunci primer.

5. *Boyce-Codd Normal Form* (BCNF)

Suatu tabel dikatakan berada pada BCNF jika dan hanya jika semua ketergantungan fungsional dengan notasi X → Y, maka X harus merupakan *superkey* pada tabel tersebut.

6. Bentuk Normal Keempat / 4 NF (*Fourth Normal Form*)

   Suatu tabel dikatakan berada pada normal keempat jila tidak mengandung dua atribut atau lebih yang bernilai banyak.

7. Bentuk Normal Kelima / 5 NF (*Fifth Normal Form*)

   Bentuk normal kelima berkaitan dengan ketergantungan relasi antar tabel (*Join Dependency*).

2.1.7 Tools Analisis dan Perancangan Sistem

Dalam tahap perancangan suatu sistem diperlukan adanya teknik-teknik penyusunan sistem untuk menganalisa dan mendokumentasikan data yang mengalir didalam sistem tersebut.

2.1.7.1 Bagan Alir Dokumen *(Document Flowmap)*

Bagan alir dokumen menggambarkan aliran dokumen dan informasi antar *area* pertanggungjawaban didalam sebuah organisasi. Bagan alir ini menelusuri sebuah dokumen dari asalnya sampai tujuannya. Secara rinci bagan alir ini menunjukkan dari mana dokumen tersebut berasal, distribusinya, tujuan digunakan-nya dokumen tersebut dan lain-lain. Bagan alir ini bermanfaat untuk menganalisis kecukupan prosedur pengawasan dalam sebuah sistem. Bagan alir dokumen disebut juga bagan alir formulir yang merupakan yang menunjukkan arus dari laporan dan formulir termasuk tembusannya.

2.1.7.2 *Data Flow Diagram* (DFD)

Model analisis harus dapat mencapai tiga sasaran utama, yaitu menggambarkan apa yang dibutuhkan oleh pelanggan, membangun dasar bagi pembuatan disain perangkat lunak, membatasi serangkaian persyaratan yang dpat divalidasi begitu perangkat lunak dibangun.

Diagram aliran data merupakan model dari sistem untuk menggambarkan pembagian sistem ke modul yang lebih kecil. Salah satu keuntungan menggunakan diagram aliran data adalah memudahkan pemakai atau user yang kurang menguasai bidang komputer untuk mengerti sistem yang akan dikerjakan.

Elemen dasar dari *data flow diagram* adalah[5]:

a. Entitas Luar (*External Entity*)

   Sesuatu yang berada diluar sistem, tetapi ia memberikan data kedalam sistem atau memberikan data dari sistem, disimbolkan dengan suatu kotak notasi. *External Entity* tidak termasuk bagian dari sistem. Bila sistem informasi dirancang untuk satu bagian maka bagian lain yang masih terkait menjadi *external entity*.

b. Arus Data (*Data Flow*)

   Arus data merupakan tempat mengalirnya informasi dan digambarkan dengan garis yang menghubungkan komponen dari sistem. Arus data ditunjukan dengan arah panah dan garis diberi nama atas arus data yang mengalir. Arus data ini mengalir diantara proses, *data store* dan menunjukan arus data dari data yang berupa masukan untuk sistem atau hasil proses sistem.

c. Proses (*Process*)

Proses merupakan apa yang dikerjakan oleh sistem. Proses dapat mengolah data atau aliran data masuk menjadi aliran data keluar. Proses berfungsi mentransformasikan satu atau beberapa data keluaran sesuai dengan spesifikasi yang diinginkan. Setiap proses memiliki satu atau beberapa masukan serta menghasilkan satu atau beberapa data keluaran. Proses sering juga disebut *bubble*.

d. Simpanan Data (*Data Store*)

Simpanan data merupakan tempat penyimpanan data yang ada dalam sistem. Data *store* dapat disimbolkan dengan dua garis sejajar atau dua garis dengan salah satu sisi samping terbuka. Proses dapat mengambil data dari atau memberikan data ke simpanan data (*database*).

1. Diagram Konteks

Diagram konteks adalah diagram yang terdiri dari sustu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh input ke sistem atau output dari sistem. Ia akan memberi gambaran tentang keeluruhan sistem. Sistem dibatasi oleh *boundary* (dapat digambarkan dengan garis putus). Dalam diagram konteks hanya ada satu proses. Tidak boleh ada *store* dalam diagram konteks.

2. Diagram Nol/Zero (*Ovetview Diagram*)

Diagram nol adalah diagram yang menggambarkan proses dari dataflow diagram. Diagram nol memberikan pandangan secara menyeluruh mengenai sistem yang ditangani, menunjukkan tentang fungsi-fungsi utama atau proses yang

ada, aliran data, dan eksternal entity. Pada level ini sudah dimungkinkan adanya/digambarkannya level selanjutnya, simbol '*' atau 'P' (*functional primitive*) dapat ditambahkan pada akhir nomor proses. Keseimbangan input dan output (*balancing*) antara diagram nol dengan diagram konteks harus terpelihara.

3. Diagam Rinci (*Level Diagram*)

Diagram rinci adalah diagram yang menguraikan proses apa yang ada dalam diagram zero atau diagram level diatasnya.

4. Penomoran Level pada DFD

Didalam satu level seyogyanya tidak terdapat lebih dari 7 buah proses dan maksimal 9, bila lebih maka harus dilakukan dekomposisi.

Tabel II.1. Tabel Penomoran Level Pada DFD

Nama Level	Nama Diagram	Nomor Process
0	Context	
1	Diagram 0	1.0,2.0,3.0,…
2	Diagram 1.0	1.1,1.2,1.3,…
2	Diagram 2.0	2.1,2.2,2.3,…
2	Diagram 3.0	3.1,3.2,3.3,…
3	Diagram 1.1	1.1.1,1.1.2,…
3	Diagram 1.2	1.2.1,1.2.2,…
3	Diagram 1.3	1.3.1,1.3.2,…
dst		

2.1.7.3 *Entity Relationship Diagram* (ERD)

ERD adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. Jadi, jelaslah bahwa ERD ini berbeda dengan DFD yang merupakan suatu model jaringan fungsi yang akan dilaksanakan oleh sistem, sedangkan ERD merupakan model jaringan data yang

menekankan pada struktur-struktur dan *relationship* data. ERD memperlihatkan hubungan antar data store pada DFD. Hubungan ini tidak terlihat pada DFD, karena DFD hanya memusatkan perhatian pada fungsi-fungsi sistem bukan pada data yang dibutuhkan.

Diagram hubungan entitas atau yang lebih dikenal dengan sebutan E-R diagram, adalah notasi grafik dari sebuah model data atau sebuah model jaringan yang menjelaskan tentang data yang tersimpan (*storage data*) dalam sistem secara abstrak. Diagram hubungan entitas tidak menyatakan bagaimana memanfaatkan data, membuat data, mengubah data dan menghapus data.

Terdapat tiga macam kardinalitas relasi, yaitu :

1. Relasi satu-ke-satu (*one-to-one*)

   Tingkat hubungan satu-ke-satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.

2. Relasi satu-ke-banyak (*one-to many)* atau banyak-ke-satu (*many-to-one)*

   Tingkat hubungan satu-ke-banyak adalah sama dengan banyak-ke-satu. Tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya satu kejadian pada entitas yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.

3. Relasi banyak-ke-banyak (*many-to-many)*

Tingkat hubungan banyak-ke-banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya. Baik dilihat dari sisi entitas yang pertama, maupun dilihat dari sisi yang kedua.

2.1.7.4 Kamus Data (*Data Dictionary*)

Salah satu komponen kunci dalam sistem manajemen *database* adalah *file* khusus yang disebut kamus data (*data dictionary*). Kamus data merupakan katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan kamus data, analisis sistem dapat mendefinisikan data yang mengalir berisi informasi tentang struktur *database*. Untuk setiap elemen data yang disimpan dalam *database* seperti nomor pokok pegawai, diuraikan secara lengkap mulai dari nama, tempat penyimpanan, program komputer yang berhubungan dan lain-lain.

Kamus data biasanya dipelihara secara otomatis oleh sistem manajemen *database*

Cara mendefinisikan kamus data yaitu[2] :

1. Menggambarkan arti aliran data atau penyimpanan yang ditunjuk dalam DFD.
2. Menggabungkan komponen dari kumpulan data yang mengalir yaitu kumpulan komponen yang mungkin bisa dipecah lagi menjadi data elementer.
3. Menggambarkan data yang tersimpan.
4. Menentukan nilai dibagian elementer dari informasi yang relevan di DFD dan data *store*nya

2.1.8 Sistem Komputer

Sistem komputer adalah sistem yang berbasis komputer yang berguna dalam mengelola data untuk menghasilkan suatu informasi yang dibutuhkan. Elemen-elemen yang mendukung dalam sistem komputer yaitu :

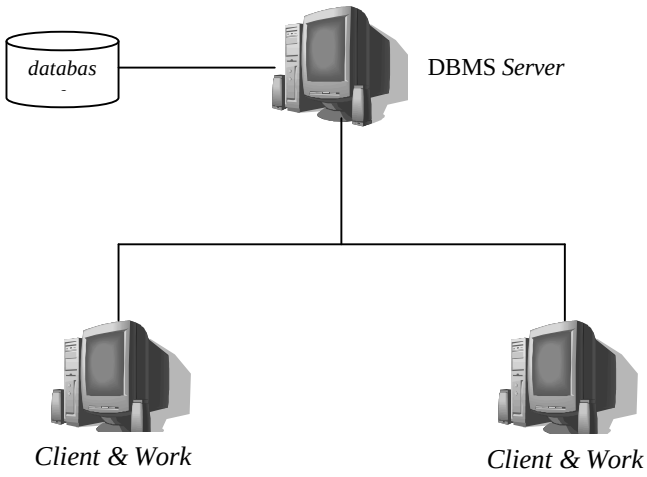
1. *Hardware* (perangkat keras), yaitu peralatan yang ada pada sistem komputer secara fisik yang terdiri dari CPU (*central processor unit)*, piranti masukan dan piranti keluaran.
2. *Software* (perangkat lunak), yaitu program yang berisi perintah-perintah untuk melakukan pengolahan data, dimana program dibuat sesederhana mungkin untuk memudahkan seorang operator komputer.
3. *Brainware*, yaitu manusia yang terlatih didalam mengoperasikan serta mengatur sistem komputer, bisa juga disebut sebagai seorang yang menguasai teknik pemrograman, mulai dari mengetahui masalah, perancangan sampai mampu membuat paket aplikasi yang berguna bagi suatu bidang.

2.1.9 Sistem *Client-Server*

Sistem *Client Server* atau disebut juga sistem tersentralisasi diterapkan pada sebuah sistem jaringan. Sistem *Client Server* ini ditujukan untuk mengatasi kelemahan-kelemahan yang terdapat pada sistem sebelumnya. Sistem *Client Server* terdiri dari dua komponen utama yaitu *client* dan *server*. *Client* berisi aplikasi basis data dan *server* berisi DBMS dan basis data. Setiap aktivitas yang dikehendaki para pemakai akan lebih dulu ditangani oleh *client*. *Client* selanjutnya mengupayakan agar semua proses “sebisa mungkin” ditangani sendiri.

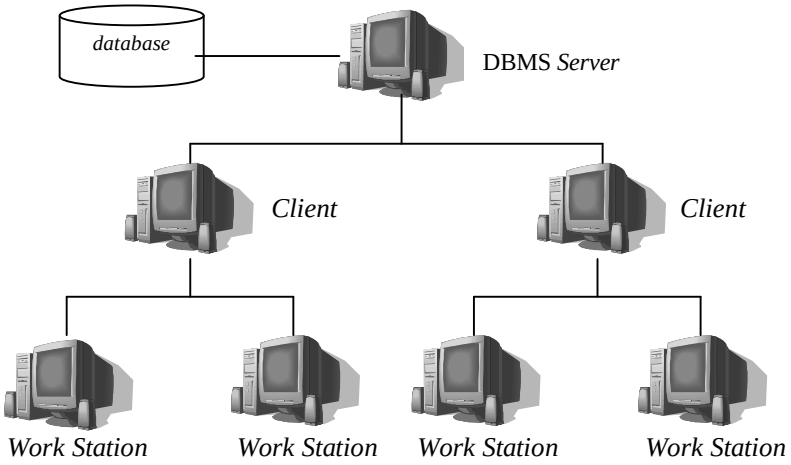
Bila ada proses yang harus melibatkan data yang tersimpan pada basis data barulah *client* mengadakan hubungan dengan *server*.

Pada sistem *Client Server* untuk memenuhi kebutuhan, *client* akan mengirimkan *message* (perintah) *query* pengambilan data. Selanjutnya, *server* yang menerima *message* tersebut akan menjalankan *query* tersebut dan hasilnya akan dikirimkan kembali ke *client*. Dengan begitu transfer datanya jauh lebih efisien. Adapun bentuk dari Sistem *Client Server* yang sederhan adalah sebagai berikut :



Gambar II.3 Sistem *Client Server* Sederhana

Disamping bentuk *client server* sederhana terdapat pula bentuk *client server* yang lebih komplek yang digambarkan sebagai berikut :



Gambar II.4 Sistem *Client Server* Kompleks

Dari kedua gambar diatas, dapat dilihat adanya dua macam implementasi sistem *client server*. Bentuk yang sederhana dapat diterapkan pada sebuah jaringan komputer lokal (LAN) dimana fungsi *client* (untuk menangani sebagian besar proses pengolahan data seperti perhitungan, perulangan, pembandingan, dan lain-lain.) dan fungsi *work station* (untuk menangani interaksi dengan pemakai, menerima data masukan dan menayangkan hasil pengolahan) disatukan.

Adanya pemisahan fungsi *client* dan fungsi *server*, disamping meningkatkan kompleksitas tersendiri dalam pembangunan aplikasi secara keseluruhan, juga menimbulkan kelemahan lain, yaitu aktivitas pemasangan aplikasi yang tidak praktis. Bila terdapat perubahan / perbaikan aplikasi basis data maka harus mengulangi pekerjaan instalasi disemua mesin *client* yang digunakan. Karena itu pekerjaan ini sangat cocok diterapkan pada sistem jaringan yang lebar (WAN). Sedangkan pada varian sistem *client server* yang lebih kompleks, aplikasi basis data tidak ditempatkan disetiap *work station*, tetapi dipasang pada setiap *client* yang jumlahnya jauh lebih sedikit. Jadi setiap *client* dan sejumlah *work*

station membentuk sebuah LAN tersendiri. Karena *client-client* ini merupakan basis tempat aplikasi basis data disimpan dan turut menangani proses-proses dalam aplikasi, maka bagi *work station, client* ini dapat dipandang sebagai *server* aplikasi. Tidak bagaimana *work station* yang diaktifkan dan dinonaktifkan oleh para pemakai, *client-client* tersebut (sebagaimana juga DBMS *server)* harus selalu dalam keadaan aktif dan terkoneksi dalam sebuah jaringan yang lebih besar (WAN). Dengan begitu tahap instalasi aplikasi dapat dilakukan secara jarak jauh (*remote)* dari lokasi lain, sehingga kelemahan dari sisi instalasi dapat diatasi

2.1.10 Software Pendukung

Sistem informasi akademik ini menggunakan Borland Delphi sebagai antar muka dan SQL Server 2000 sebagai databasenya.

2.1.10.1 Borland Delphi

Borland Delphi atau yang biasa disebut Delphi saja, merupakan sarana aplikasi visual. Bahasa pemrograman yang digunakan adalah bahasa pemrograman Pascal atau yang kemudian disebut juga bahasa pemrograman Delphi[6]. Delphi merupakan generasi penerus dari Turbo Pascal. Turbo Pascal yang diluncurkan pada tahun 1983 dirancang untuk dijalankan pada sistem operasi DOS (yang merupakan sistem operasi yang paling banyak digunakan pada saat itu). Sedangkan Delphi yang diluncurkan pertama kali tahun 1995 dirancang untuk beroperasi dibawah sistem operasi windows.

Kelebihan yang dimiliki Borland Delphi diantaranya[6] :

1. Form dan komponen-komponennya dapat dipakai ulang dan dikembangkan.
2. Mampu mengakses VBX.

3. Tersedia template aplikasi dan template form.
4. Meliliki lingkungan pengembangan visual yang dapat diatur sesuai kebutuhan.
5. Menghasilkan file terkompilasi yang berjalan lebih cepat.
6. Kemampuan mengakses data dari bermacam-macam format.

2.1.10.2 Microsoft SQL Server

Microsoft SQL Server adalah sistem manajemen bisnis data yang memakai perintah-perintah *Transact-SQL* untuk mengirimkan perintah dari komputer *client* ke komputer *server*. *Transact-SQL* adalah bahasa SQL yang dikembangkan oleh Microsoft dengan menambahkan dialek-dialek tertentu. Microsoft SQL Server berisi database, mesin database, dan aplikasi yang diperlukan untuk mengelola data dan komponen-komponennya[3].

1. Perintah-Perintah SQL

a. Menambah Data (*Insert*)

Perintah SQL yang digunakan untuk menambah data ke database adalah perintah *INSERT* . Perintah ini mempunyai bentuk umum sebagai berikut:

INSERT [INTO] nama_tabel [(daftar field/kolom)] VALUES (daftar_data)

b. Memilih/Mengakses Data (*Select*)

Perintah SQL yang digunakan untuk memilih data adalah perintah *SELECT*. Bentuk umum perintah ini adalah sebagai berikut:

SELECT daftar_select FROM daftar_tabel

[WHERE kondisi_pencarian]

[GROUP BY daftar_group_by]

[HAVING kondisi_pencarian]

[ORDER BY daftar_order[ASC │DESC]]

c. Mengedit/Mengubah Data (*Update*)

Perintah SQL yang digunakan untuk mengedit data adalah perintah *UPDATE SET*. Perintah ini mempunyai bentuk umum sebagai berikut:

UPDATE nama_table SET field1=databaru1 [data2=databaru2]

[WHERE kondisi_update]

d. Menghapus Data (*Delete*)

Perintah SQL yang digunakan untuk menghapus data adalah perintah *DELETE FROM*. Perintah ini mempunyai bentuk umum sebagai berikut:

DELETE FROM tabel_hapus WHERE kondisi_hapus