

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem Informasi

Sistem merupakan jaringan kerja dari beberapa prosedur yang saling berhubungan, berkumpul bersama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu. Karakteristik sistem tersebut yaitu:

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 atau bagian-bagian dari sistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

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 menunjukkan ruang lingkup dari sistem tersebut.

3. Lingkungan luar sistem

Lingkungan luar dari suatu sistem adalah apapun di luar batas 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 tersebut sehingga harus tetap dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup 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. Keluaran dari subsistem akan menjadi masukan untuk subsistem yang lainnya dengan melalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lain membentuk satu kesatuan.

5. Masukan sistem

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

6. Keluaran sistem

Keluaran (*output*) adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat berupa masukan untuk subsistem yang lain.

7. Pengolah sistem

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran. Di dalam sistem informasi, data-data diolah menjadi informasi.

8. Sasaran sistem

Suatu sistem pasti mempunyai sasaran atau tujuan. Kalau suatu sistem tidak mempunyai suatu sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan.

Informasi dapat diperoleh dapat diperoleh sistem informasi (*information system*) atau disebut juga dengan *processing system* atau *information processing system* atau *information generreting system*. Sumber informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (*event*) adalah sesuatu yang terjadi pada saat yang tertentu. Kesatuan nyata (*fact and entity*) adalah suatu objek nyata seperti tempat, benda, dan orang yang benar ada dan terjadi.

Suatu informasi mempunyai kualitas, kualitas informasi tergantung dari tiga hal yaitu:

1. Akurat

Informasi harus bebas dari kesalahan-kesalahan dan tidak bisa atau menyesatkan. Akurat juga harus mencerminkan maksudnya.

2. Tepat pada waktunya

Informasi yang datang pada penerimanya tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi, karena informasi merupakan landasan di dalam pengambilan keputusan.

3. Relevan

Informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda.

Sistem informasi merupakan suatu sistem di dalam 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 yang diperlukan (Jogiarto,1995:789).

Sistem informasi dapat terdiri dari blok-blok bangunan, blok-blok tersebut adalah:

1. Masukan

Masukan (*input*) mewakili data yang masuk ke dalam sistem informasi. Masukan di sini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Model

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

3. Keluaran

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

4. Teknologi

Teknologi merupakan kotak alat (*tool box*) dalam sistem informasi. Teknologi digunakan untuk menerima masukan, 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 (*brainware*), perangkat lunak (*software*), dan perangkat keras, (*hardware*).

5. Basis Data

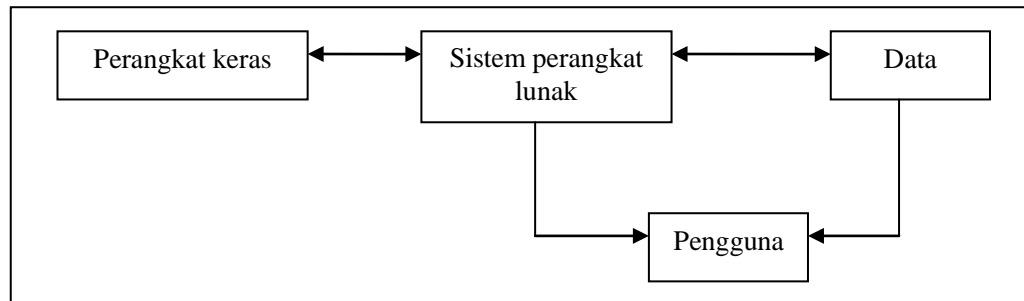
Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

6. Kendali

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

Sistem informasi sendiri memiliki sejumlah komponen tertentu. Sistem informasi terdiri dari beberapa komponen yang berbeda yaitu manusia (*user*), data,

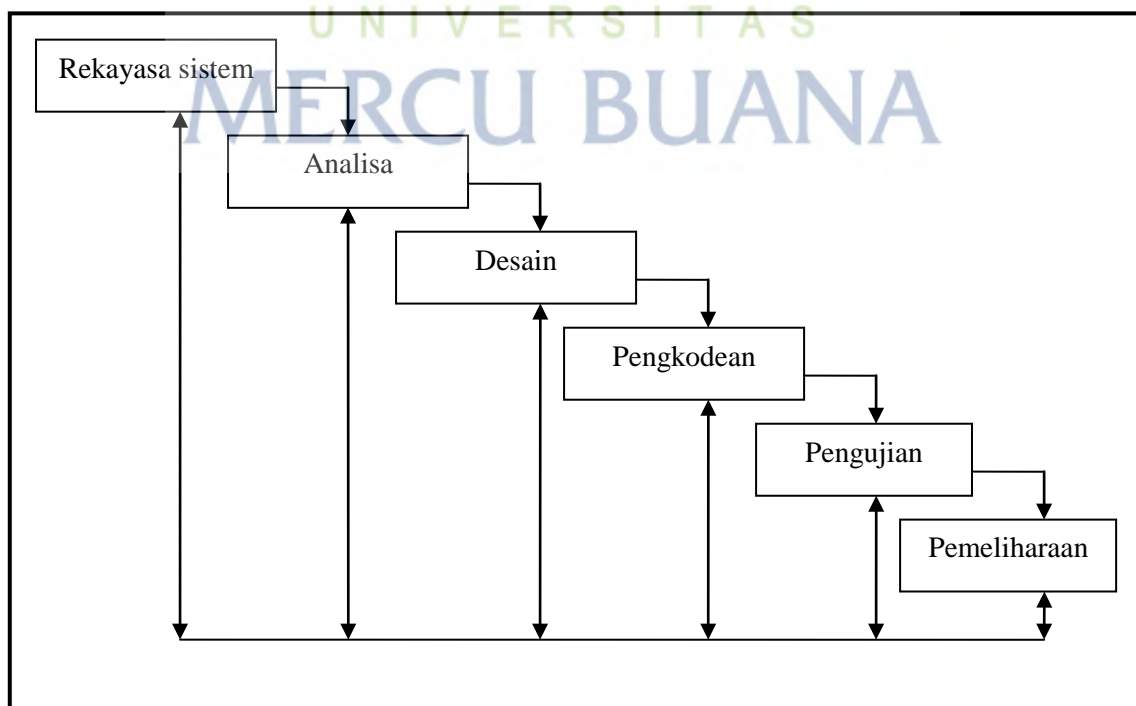
perangkat keras (*hardware*), dan perangkat lunak (*software*). Sebagai suatu sistem, setiap komponen tersebut berinteraksi satu dengan lainnya membentuk suatu kesatuan untuk mencapai sasarnya. Komponen sistem informasi ini dapat dilihat pada Gambar 2.1.



Gambar 2.1 komponen Sistem Informasi

2.2 Metodologi Rekayasa Perangkat Lunak

Model air terjun membutuhkan pendekatan yang sistematis dan sekuensial dalam pengembangan software yang dimulai pada level sistem dan prosesnya melalui analisis, desain, kode, pengujian, dan pemeliharaan seperti yang ditunjukkan oleh Gambar 2.2.



Gambar 2.2 Model Air Terjun

Model Air Terjun (*waterfall*) menawarkan cara pembuatan perangkat lunak secara lebih nyata. Pemodelan dalam perangkat lunak merupakan suatu yang harus dikerjakan di bagian awal dari rekayasa, dan pemodelan ini akan mempengaruhi pekerjaan-pekerjaan dalam rekayasa perangkat lunak tersebut. Model proses perangkat lunak masih menjadi objek penelitian, namun pada saat ini terdapat banyak model umum atau paradigma yang berbeda dari pengembangan perangkat lunak. Salah satu model yang digunakan dalam pengembangan rekayasa lunak adalah model *Waterfall*. Model *waterfall* melingkupi aktifitas-aktifitas sebagai berikut:

1. Rekayasa dan Pemodelan sistem/informasi. Rekayasa dan analisis sistem menyangkut pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis serta desain tingkat puncak. Rekayasa informasi mencakup juga pengumpulan kebutuhan pada tingkat strategi dan tingkat area.
2. Analisis kebutuhan perangkat lunak. Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada perangkat lunak.
3. Proses desain menerjemahkan syarat atau kebutuhan ke dalam sebuah representasi perangkat lunak yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode.
4. Desain harus diterjemahkan ke dalam nebtuk mesin yang bisa dibaca. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis.
5. Proses pengujian berfokus pada logika internal perangkat lunak, memastikan bahwa semua pernyataan sudah diuji dengan mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

6. Perangkat lunak akan mengalami perubahan setelah disampaikan kepada pelanggan. Perubahan akan terjadi karena kesalahan-kesalahan ditentukan, karena perangkat lunak harus disesuaikan untuk mengakomodasi perubahan-perubahan di dalam lingkungan eksternalnya atau karena pelanggan membutuhkan perkembangan fungsional. Pemeliharaan perangkat lunak mengaplikasikan lagi setiap fase program sebelumnya dan tidak membuat yang baru lagi.

2.3 Personal Home Page (PHP)

PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP bernama FI (*Form Interpreted*). Pada saat tersebut PHP adalah sekumpulan script yang digunakan untuk mengolah data form dari web. Perkembangan selanjutnya adalah Rasmus melepaskan kode sumber tersebut dan menamakannya PHP/FI, pada saat tersebut kepanjangan dari PHP/FI adalah *Personal Home Page/Form Interpreter*. Dengan pelepasan kode sumber ini menjadi *open source*, maka banyak programmer yang tertarik untuk ikut mengembangkan PHP. Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini *interpreter* sudah di implementasikan dalam C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend, menulis ulang *interpreter* PHP menjadi lebih bersih, lebih baik dan lebih cepat. Kemudian pada Juni 1998 perusahaan tersebut merilis *interpreter* baru untuk PHP dan meresmikan nama rilis tersebut menjadi PHP 3.0. Pada pertengahan tahun 1999, Zend merilis *interpreter* PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai. Versi ini banyak dipakai sebab versi ini mampu dipakai untuk

membangun aplikasi web kompleks tetapi tetap memiliki kecepatan proses dan stabilitas yang tinggi.

Pada juni 2004, Zend merilis PHP 5.0. Versi ini adalah versi mutakhir dari PHP. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Dalam versi ini juga dikenalkan model pemrograman berorientasi objek baru untuk menjawab perkembangan bahas pemrograman kearah pemrograman berorientasi objek.

PHP adalah bahasa pemrograman scripting sisi server, bahasa pemrograman yang digunakan oleh sebuah web server dan berfungsi sebagai pengolah data pada sebuah server. PHP merupakan interpreter yang dapat dieksekusi sebagai CGI untuk server web atau dijadikan modul dari server web. Portabilitas aplikasi yang dikembangkan dengan menggunakan PHP lebih mudah dan tidak membutuhkan perubahan pada source code aplikasi, salinkan langsung kedalam server tujuan maka maka aplikasi dengan PHP dapat langsung dijalankan.

Kelebihan PHP dari bahasa pemrograman lain:

1. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. PHP mampu berjalan di beberapa server yang ada, misalnya Apache, Microsoft IIS, PWS, dan Xitami.
3. PHP mampu berjalan di Linux sebagai platform system operasi utama bagi PHP, namun juga dapat berjalan di FreeBSD, Unix, Solaris, Windows, dan yang lainnya.
4. PHP juga mendukung akses ke beberapa database yang sudah ada. Database itu antara lain MySQL, PosgreSQL, dan MicrosoftSQL server.

5. PHP adalah termasuk bahasa yang *embedded* (bisa ditempel atau ditempatkan dalam tag HTML).

2.3.1 Aplikasi PHP dengan MySQL

PHP dan MySQL merupakan kombinasi pasangan piranti yang banyak dan umum digunakan dalam pembangunan dan pengembangan situs web yang menggunakan basis data. PHP dan MySQL dapat dikatakan merupakan pasangan software yang serasi untuk pengembangan aplikasi web basis data. Dalam melakukan pemrograman pada situs web dinamis, minimal dibutuhkan PHP, Apache sebagai web server, dan MySQL sebagai basis data. Konsep kerja PHP adalah ketika seseorang mengetikkan alamat di *web browser*, maka browser akan mengirimkan perintah tersebut ke web server. Jika yang diminta adalah berkas (*file*) yang mengandung program *server-side* maka *web server* akan menjalankan terlebih dahulu program tersebut dan mengirimkan hasilnya ke *browser*. Dan jika yang diminta adalah file HTML maka *web browser* akan langsung mengirimkan ke *browser* apa adanya. Server web secara default akan memberikan dokumen HTML pada setiap permintaan yang ditujukan kepadanya. Dokumen HTML dalam pengertian umum merupakan dokumen yang berisi teks yang diformat dengan tag-tag HTML yang dibuat dengan menggunakan editor teks atau editor HTML, informasi yang disajikan pada setiap dokumen bersifat statis, jika akan ada perubahan maka dokument tersebut harus diedit dengan menggunakan editor HTML kemudian diupload kembali kedalam server. MySQL dapat digunakan sebagai sumber basis data untuk informasi yang akan ditampilkan pada suatu halaman web. Komponen pemformatan informasi dengan menggunakan HTML dan menggunakan informasi yang akan disajikan dapat dipisahkan sehingga apabila terjadi perubahan pada data atau informasi, maka

perubahan dapat dilakukan pada bagian data atau informasi saja, jika terjadi perubahan pada bagian pemformatan, maka perubahan dapat dilokalisasi pada bagian pemformatan ini.

Dalam situs web dinamis data-data disimpan pada basis data dan diakses dari basis data, dalam hal ini menggunakan MySQL sebagai basis data. Secara umum, akses ke basis data melalui tiga tahapan yaitu:

1. Koneksi ke basis data,
2. Query/permintaan data (operasi manipulasi data),
3. Pemutusan koneksi.

Untuk melakukan hubungan antara kode program PHP dengan MySQL dapat dilakukan dengan sintaks di bawah ini:

```
mysql_connect("server_name","user_name","password")
mysql_db_query("database_name","query")
```

Pada baris pertama PHP akan melakukan koneksi ke basis data berdasarkan nama server yang digunakan (*server_name*), pengguna yang diberi hak untuk mengakses basis data (*user_name*), serta kata kunci yang digunakan (*password*).

Sedangkan baris kedua digunakan untuk melakukan manipulasi terhadap data dengan memilih basis data (*database_name*) dan manipulasi yang diinginkan (*query*).

Beberapa operasi manipulasi data didalam basis data yang dapat dilakukan adalah sebagai berikut:

1. Memasukan data.

Sintaks:

```
INSERT INTO nama_tabel (field1, field2...) VALUES ('data1', 'data2', ...)
```

Contoh:

```
mysql_db_query(web, "INSERT INTO kelas(nim,nama) VALUES ('$nim', '$nama')");
```

Pada contoh diatas PHP memberikan perintah untuk memasukkan variable \$nim, \$nama ke dalam field nim dan nama yang ada pada tabel kelas.

2. Menampilkan data.

Sintaks:

```
SELECT field1, field2, ... FROM nama_tabel WHERE kondisi1, kondisi2, ...
ORDER BY nama_field
```

Contoh:

```
$query01 = MYSQL_DB_QUERY (web, "SELECT nim,nama,jurusan FROM mhs
WHERE id = '$nim' ORDER BY id ASC");
```

Pada contoh diatas PHP memberikan perintah untuk mencari isi field nim,nama,jurusan dari tabel mhs dengan kondisi field id sesuai dengan variable \$nim dan diurutkan secara urutan naik (*ascending*) pada basis data web.

3. Mengubah data.

Sintaks:

```
UPDATE nama_tabel SET field1='nilai_baru', field2='nilai_baru', ...
WHERE kondisi 1, kondisi 2, ... .
```

Contoh:

```
Mysql_db_query (web, "UPDATE mhs SET nim = '$nim', nama = '$nama',
jurusan='$jurusan' WHERE id = '$nim'");
```

Pada contoh diatas PHP memberikan perintah untuk merubah isi tabel mhs dengan field nim, nama, jurusan dirubah dengan variable \$nim pada basis data web.

3. Menghapus data.

Sintaks:

```
DELETE FROM nama_tabel WHERE syarat1, syarat2, ... .
```

Contoh:

```
mysql_db_query (web, "DELETE FROM hs WHERE kode = '$nim'");
```

Pada contoh tersebut PHP memberikan perintah untuk menghapus isi mhs dengan kondisi `kode` sesuai dengan variable `$nim` pada basis data `web`.

2.4 Basis Data

Basis data (*database*) dapat dianggap sebagai tempat untuk sekumpulan berkas data terkomputerisasi. Sistem basis data pada dasarnya adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara informasi dan membuat informasi tersebut tersedia saat dibutuhkan.

Tujuan dalam basis data adalah mempermudah penciptaan struktur data dan membebaskan program dari masalah penyusunan file yang tidak terstruktur, sehingga mempermudah dalam mendesain dan menyiapkan suatu basis data yang dapat digunakan oleh sejumlah program aplikasi yang berlainan.

Dalam membangun situs yang dinamis diperlukan suatu manajemen basis data agar situs dapat dikelola dengan baik, dalam hal ini diperlukan suatu basis data sebagai media penyimpanan informasi-informasi yang dapat diakses saat diperlukan.

2.4.1 Proses Perancangan Basis Data

Proses perancangan basis data, terlepas dari masalah yang ditangani dibagi menjadi tiga tahapan, yaitu:

1. Perancangan basis data secara konseptual.
2. Perancangan basis data secara logis.
3. Perancangan basis data secara fisis.

Perancangan basis data secara konseptual merupakan upaya untuk membuat model yang masih bersifat konsep.

Perancangan basis data secara logis merupakan tahapan untuk memetakan model konseptual ke model basis data yang akan dipakai, seperti model relasional, hirarki, atau jaringan. Namun sebagaimana halnya perancangan basis data secara konseptual, perancangan ini tidak tergantung pada DBMS (*DataBase Management System*) yang akan dipakai. Itulah sebabnya perancangan basis data secara logis terkadang disebut pemetaan model data.

Perancangan basis data secara fisis merupakan tahapan untuk menuangkan perancangan basis data yang bersifat logis menjadi basis data fisis yang tersimpan pada media penyimpanan eksternal.

2.5 Diagram Alir Data

Diagram Alir Data atau *Data-Flow Diagram*, selanjutnya disebut dengan DAD, menggambarkan sistem sebagai jaringan kerja antar fungsi yang berhubungan satu sama lain dengan aliran dan penyimpanan data (Pohan dan Bahri, 1997:16).

DAD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau lingkungan fisik dimana data itu akan disimpan (Jogiyanto, 2005:700). Ada empat komponen dalam model DAD, yaitu:

1. Proses.

Proses menunjukkan transformasi dari masukan menjadi keluaran, dalam hal ini sejumlah masukan dapat menjadi hanya satu keluaran atau sebaliknya.

Proses direpresentasikan dalam bentuk lingkaran.

2. Aliran.

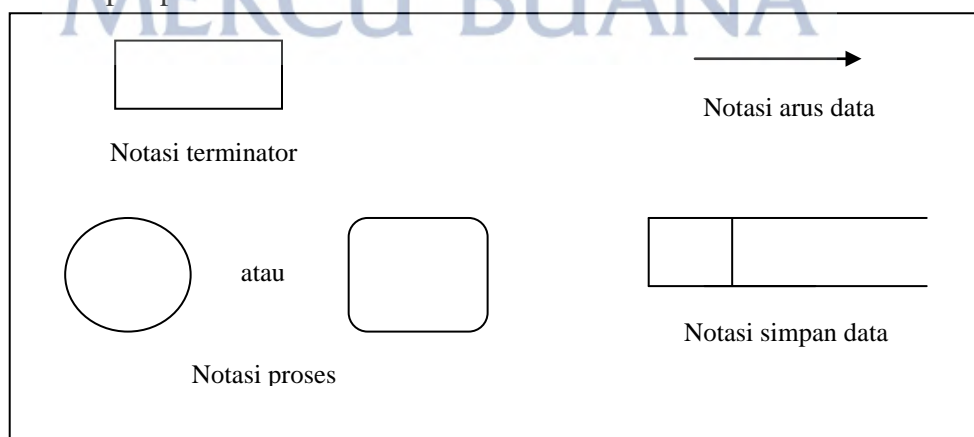
Aliran digunakan untuk menggambarkan gerakan paket data atau informasi dari satu bagian ke bagian lain dari sistem. Nama aliran berfungsi untuk mendefinisikan arti dari aliran itu, dan ditulis untuk mengidentifikasi aliran tersebut. Komponen ini dipresentasikan dengan menggunakan panah yang menuju ke atau dari proses.

3. Penyimpanan.

Komponen ini digunakan untuk memodelkan kumpulan data atau paket data. Notasi yang digunakan adalah garis dua sejajar.

4. Terminator.

Terminator mewakili entiti luar dimana sistem berkomunikasi. Umumnya mengidentifikasi terminator tidak sulit. Kadangkala terminator berupa pemakai sistem itu sendiri. Pada kasus lain pemakai berpikir bahwa dia merupakan bagian dari sistem sehingga mempermudah pemodelan untuk mengidentifikasi terminator yang relevan. Komponen ini dipresentasikan menggunakan persegi panjang. Gambar notasi komponen-komponen dalam DAD tampak pada Gambar 2.3.



Gambar 2.3 Notasi DAD

Sejumlah petunjuk yang perlu dilakukan untuk membuat DAD yang jelas dan enak dibaca yaitu:

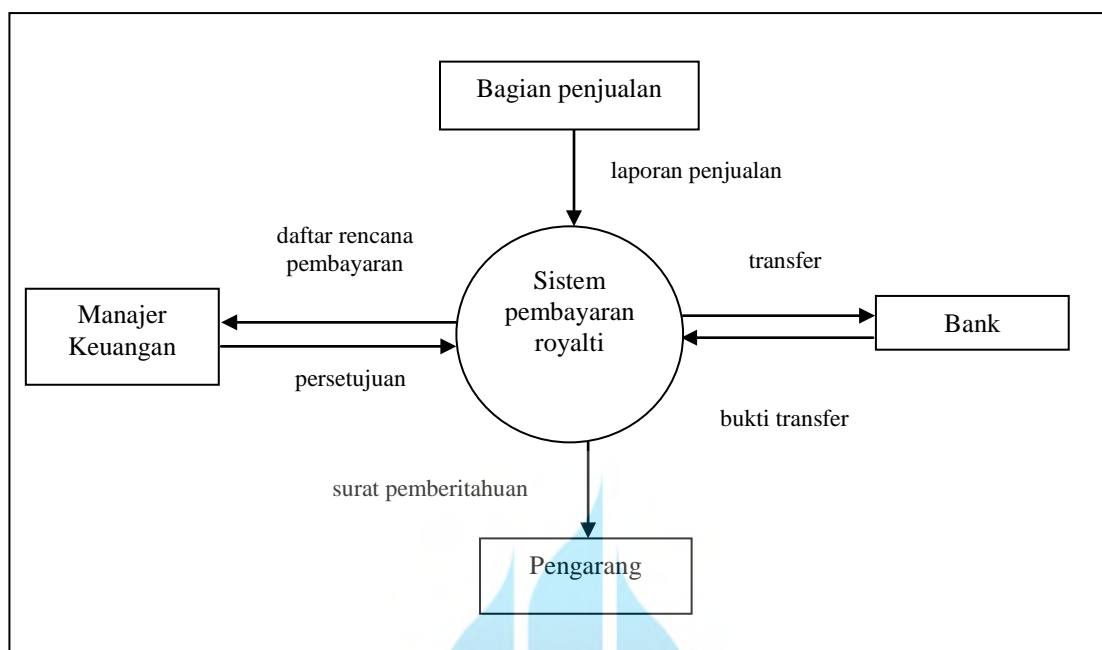
1. Pilih nama yang jelas maksudnya (bagi proses, aliran, penyimpanan, dan terminator).
2. Menomori proses untuk memperjelas sistematika.
3. Menggambar kembali DAD hingga beberapa kali, sehingga cukup estetik.
4. Mencegah DAD yang terlalu kompleks dan tidak perlu.
5. Menjamin konsistensi DAD tersebut secara intern ataupun model lain yang berkaitan dengan DAD.

DAD yang pertama kali digambar, dikenal dengan diagram konteks, yaitu DAD level teratas (Jogiyanto, 2005:714). Diagram konteks adalah diagram yang memperlihatkan sistem sebagai sebuah proses (Abdul Kadir, 2003:40). Tujuannya adalah memberikan pandangan umum sistem. Diagram konteks memperlihatkan sebuah proses yang berinteraksi dengan dengan lingkungannya. Ada pihak luar yang atau lingkungan yang memberi masukan dan ada pihak yang menerima keluaran sistem. Dalam hal ini pihak luar (sering disebut terminator) dapat berupa sistem lain, suatu perangkat keras, orang atau lain sebagainya.

Dari diagram konteks ini kemudian akan digambarkan lagi dengan lebih terinci lagi yang disebut dengan *overview* diagram (level 0). Tiap-tiap proses tersebut kemudian digambarkan lagi lebih terinci, dan seterusnya sampai tiap-tiap proses tidak dapat digambar lebih terinci lagi.

Suatu diagram konteks selalu mengandung satu dan hanya satu proses saja. Proses ini mewakili proses dari seluruh sistem. Diagram konteks ini menggambarkan hubungan *input/output* antara sistem dengan dunia luarnya. Berikut contoh diagram

konteks pada sistem yang dipakai untuk menangani pembayaran royalti buku yang tampak pada Gambar 2.4.



Gambar 2.4 Diagram konteks sistem penanganan royalti buku

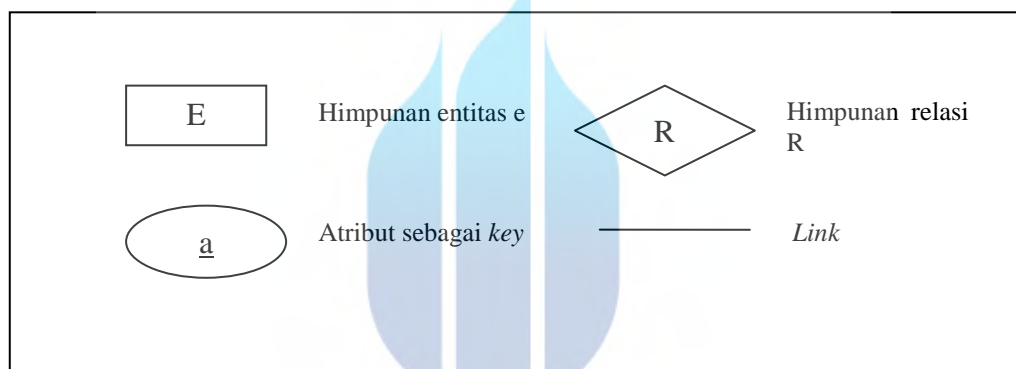
Diagram Keterhubungan Entitas atau *Entity-Relationship Diagram*(ERD) adalah model konseptual yang mendeskripsikan hubungan antar penyimpanan (dalam DAD). ERD digunakan untuk memodelkan struktur data dan hubungan antar data. Model *entity-relationship* yang berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang merepresentasikan seluruh fakta dari *real world* (dunia nyata) yang ditinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan ERD (Fathansyah, 2001:70).

ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data. Notasi-notasi simbolik di dalam ERD yang dapat digunakan adalah:

1. Persegi panjang, menyatakan himpunan entitas.

2. Lingkaran atau elips, menyatakan atribut (atribut yang berfungsi sebagai key digarisbawahi).
3. Belah ketupat, menyatakan himpunan relasi.
4. Garis (*link*), sebagai penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya.
5. Kardinalitas relasi dapat dinyatakan dengan banyaknya garis cabang atau dengan pemakaian angka (1 dan 1 untuk relasi satu-ke-satu, 1 dan N untuk relasi satu-ke-banyak atau N dan N untuk relasi banyak-ke-banyak).

Notasi ERD dapat dilihat pada Gambar 2.5.



Gambar 2.5 Notasi ERD

Entitas merupakan individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain. Sekelompok entitas yang sejenis dan berada dalam lingkup yang sama membentuk sebuah himpunan entitas.

Atribut mendeskripsikan karakteristik (sifat-sifat) dari entitas. Ada atribut yang dijadikan kunci (*key*) dan yang lainnya disebut atribut deskriptif. Atribut deskriptif adalah atribut-atribut yang tidak menjadi *primary-key* atau merupakan anggota dari *primary-key*.

Pada dasarnya, kunci adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data dalam tabel secara unik atau tidak boleh sama

(Fathansyah, 2001:39). Ada tiga macam kunci yang dapat diterapkan pada suatu tabel, yaitu:

1. Kunci Super (*Superkey*).

Superkey merupakan satu atau lebih atribut (kumpulan atribut) yang dapat membedakan setiap baris data dalam sebuah tabel secara unik.

2. Kunci Kandidat (*Candidate-key*).

Candidate-key merupakan kumpulan atribut minimal yang dapat membedakan setiap baris data dalam sebuah tabel secara unik.

3. Kunci Primer (*Primary-key*).

Primary-key dipilih dari sejumlah *candidate-key*, umumnya didasari oleh:

- a. *Key* tersebut lebih sering (lebih natural) untuk dijadikan sebagai acuan.
- b. *Key* tersebut lebih ringkas.
- c. Jaminan keunikan *key* tersebut lebih baik.

Berikut contoh dari satu tabel entitas yang tampak pada Tabel 2.5.

Tabel 2.1 Contoh tabel entitas mahasiswa

<u>Nim</u>	nama_mhs	Jurusan	tgllhr_mhs
01503-080	Aditia	Teknik Informatika	06-06-1986
01503-094	Deddy Mahendra	Teknik Informatika	16-06-1986
01503-101	Angga	Teknik Informatika	26-06-1986

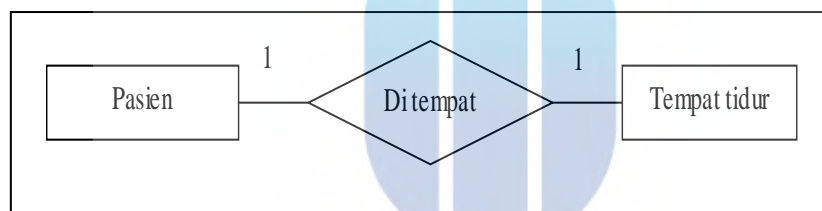
Dari Tabel 2.1 diketahui bahwa himpunan entitasnya adalah mahasiswa, terdiri atas empat kolom (jumlah kolom menyatakan jumlah atribut) dan tiga baris (jumlah baris menyatakan jumlah entitas). Atribut dari tabel diatas adalah nim, nama_mhs, jurusan, dan tgllhr_mhs. Atribut yang berfungsi sebagai *primary-key* adalah nim, dan atribut yang berfungsi sebagai atribut deskriptif adalah nama_mhs, jurusan, dan tgllhr_mhs.

Relasi menunjukkan adanya hubungan diantara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Kumpulan semua relasi di antara entitas-entitas yang terdapat pada himpunan entitas-himpunan entitas membentuk himpunan relasi (*relationship sets*).

Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Kardinalitas relasi yang terjadi diantara dua himpunan entitas dapat berupa:

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

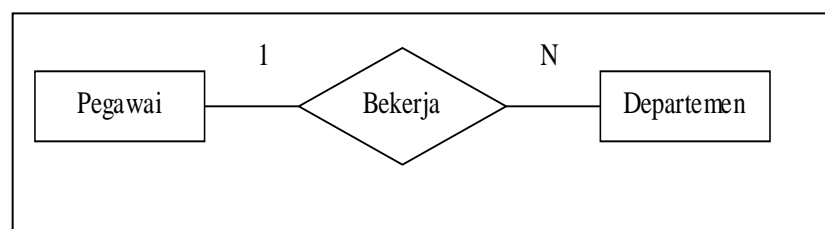
Setiap entitas pada himpunan entitas pertama berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas kedua, begitu juga sebaliknya. Contoh kardinalitas satu ke satu dapat dilihat pada Gambar 2.6.



Gambar 2.6 Contoh kardinalitas satu ke satu

2. Satu ke banyak (*One to many*).

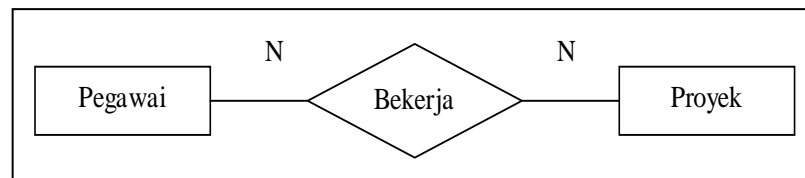
Setiap entitas pada himpunan entitas pertama dapat berhubungan dengan banyak entitas pada himpunan entitas kedua, dan setiap entitas pada himpunan entitas kedua berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas pertama. Contoh kardinalitas satu ke banyak dapat dilihat pada Gambar 2.7.



Gambar 2.7 Contoh kardinalitas satu ke banyak

3. Banyak ke banyak (*Many to many*).

Setiap entitas pada himpunan entitas pertama dapat berhubungan dengan banyak entitas pada himpunan entitas kedua, dan demikian juga sebaliknya. Contoh kardinalitas banyak ke banyak dapat dilihat pada Gambar 2.8.



Gambar 2.8 Contoh kardinalitas banyak ke banyak

Kardinalitas relasi satu ke banyak dan banyak ke satu dapat dianggap sama, karena tinjauan kardinalitas relasi selalu dilihat dari dua sisi (dari himpunan entitas pertama ke himpunan entitas kedua dan dari himpunan entitas kedua ke himpunan entitas pertama).

Dua kelompok pentahapan yang biasa ditempuh dalam pembuatan ERD, yaitu:

1. Tahap pembuatan ERD awal (*preliminary design*).

Mendapatkan sebuah rancangan basis data minimal yang dapat mengakomodasi kebutuhan penyimpanan data terhadap sistem yang sedang ditinjau.

2. Tahap optimasi ERD (*final design*)

Melakukan koreksi terhadap hasil tahap pertama, berupa pendekomposisian himpunan entitas, penggabungan himpunan entitas, pengubahan derajat relasi, penambahan relasi baru hingga perubahan (penambahan dan pengurangan) atribut-atribut untuk masing-masing entitas dan relasi.

Langkah-langkah teknis yang dapat dilakukan untuk menghasilkan ERD awal adalah:

1. Mengidentifikasi dan menetapkan seluruh himpunan entitas yang terlibat.

2. Menentukan atribut-atribut *key* dari masing-masing himpunan entitas.
3. Mengidentifikasi dan menetapkan seluruh himpunan relasi di antara himpunan entitas-himpunan entitas yang ada beserta kunci asingnya (*foreign-key*).
4. Menentukan derajat atau kardinalitas relasi untuk setiap himpunan relasi.
5. Melengkapi himpunan entitas dan himpunan relasi dengan atribut-atribut deskriptif (*non key*).

2.6 Kamus Data

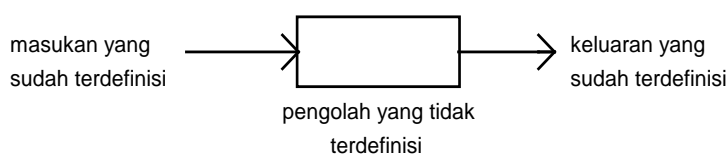
Kamus data merupakan sebuah model yang berfungsi membantu pelaku sistem untuk mengerti aplikasi secara detail (Pohan dan Bahri, 1997:21). Setelah pemodelan dalam bentuk DAD didefinisikan, selanjutnya digambarkan makna atau kandungan dari setiap aliran data dan media penyimpanan data yang ada pada DAD tersebut dalam bentuk kamus data.

2.7 Metode Black Box

Suatu sistem dimana input dan outputnya dapat didefinisikan tetapi prosesnya tidak diketahui atau tidak terdefinisi. Metode ini hanya dapat dimengerti oleh pihak dalam (yang menangani) sedangkan pihak luar hanya mengetahui masukan dan hasilnya. Sistem ini terdapat pada subsistem tingkat terendah.

Contoh :

Bagian pencetakan uang, proses pencernaan.





UNIVERSITAS
MERCU BUANA