

## BAB II

### LANDASAN TEORI

#### 2.1 Konsep Dasar Sistem

##### 2.1.1 E-Commerce (Electronic Commerce)

Istilah *Electronic Commerce* merupakan suatu konsep yang muncul untuk menjelaskan proses dari pembelian, penjualan, pertukaran produk, servis dan informasi melalui jaringan komputer, termasuk *Internet*. Atau dengan kata lain *Electronic Commerce* juga merupakan salah satu cara dalam melakukan bisnis secara elektronik, melalui jaringan *Internet*.

Kalakota dan Whinston (1997) mendefinisikan *E-Commerce* menurut beberapa perspektif, antara lain:

a. Dari perspektif komunikasi

*E-Commerce* merupakan pengiriman barang, servis, informasi, atau pembayaran melalui jaringan computer atau dengan alat elektronik lainnya.

b. Dari perspektif proses bisnis

*E-Commerce* merupakan suatu teknologi aplikasi untuk otomatisasi dari transaksi bisnis dan arus kerja.

c. Dari perspektif servis

*E-Commerce* merupakan suatu alat yang menunjukkan keinginan dari perusahaan, konsumen dan manajemen untuk mengurangi biaya servis sementara meningkatkan kualitas barang dan kecepatan layanan pengiriman.

d. Dari perspektif online

*E-Commerce* menyediakan kemampuan untuk melakukan pembelian dan penjualan produk dan informasi di *Internet* atau layanan *online* lainnya.

Ada tiga tipe utama dari *E-Commerce*, yaitu:

a. B2C (*Business-to-Consumer*)

Di dalam transaksi B2C, transaksi *online* dibuat antara bisnis dan konsumen.

Transaksi ini meliputi transaksi penjualan dengan pembeli-pembeli individu.

b. B2B (*Business-to-Business*)

Di dalam B2B, transaksi *online* dilakukan antara bisnis dan bisnis yang lain.

Partisipannya meliputi bisnis atau organisasi lainnya.

c. B2E (*Business-to-Employee*)

Di dalam B2E, transaksi *online* dilakukan antara bisnis dan pegawai atau karyawan. Transaksi yang ada hanya berupa informasi dan servis untuk kepentingan bisnis perusahaan.

*E-Commerce* berdasarkan pada derajat digitalisasi dari produk, proses, dan pengiriman mempunyai dua macam bentuk, yaitu:

a. *Pure E-Commerce*, yaitu semua dimensinya *digital*.

b. *Partial E-Commerce*, yaitu semua kemungkinan-kemungkinan lainnya, termasuk campuran dari dimensi fisik dan *digital*.

Sebagai contoh membeli buku di *Amazon.com* tidak termasuk *pure E-Commerce* karena buku tersebut dikirimkan melalui FedEx. Namun kalau membeli *e-book* (buku elektronik) dari *Amazon.com* atau membeli *software* dari *Egghead.com* termasuk *pure E-Commerce* karena pengiriman, pembayaran, dan perantara semuanya *digital*.

Keuntungan-keuntungan dari *E-Commerce*, antara lain:

a. Keuntungan bagi organisasi

- Dapat memperluas daerah pemasaran untuk pasar nasional dan internasional.

- Dapat mengurangi biaya pembuatan, pemrosesan, pendistribusian, dan penyimpanan.
  - Biaya telekomunikasi yang lebih murah (*internet* lebih murah dari *Value Added Networks* atau VANS).
- b. Keuntungan bagi konsumen
- Konsumen dapat berbelanja atau melakukan transaksi lainnya selama 24 jam setiap hari.
  - Pilihan produk yang lebih murah bagi konsumen dengan mengizinkan konsumen untuk melakukan perbandingan antara produk yang satu dengan produk yang lain di banyak tempat.
  - Konsumen dapat memperoleh diskon yang banyak karena adanya persaingan.
- c. Keuntungan bagi masyarakat
- Konsumen tidak perlu pergi atau keluar rumah untuk berbelanja sehingga dapat mengurangi kemacetan lalu lintas dan mengurangi polusi udara.
  - Masyarakat yang berada di dunia ketiga dapat menikmati produk dan servis yang tidak tersedia bagi mereka sebelumnya.

Keterbatasan-keterbatasan dari *E-Commerce*, antara lain:

- a. Batasan teknis
- Kurangnya sistem keamanan, kepercayaan, standar, dan protokol komunikasi yang memadai
  - Kurangnya *bandwidth* telekomunikasi.
  - Sulitnya untuk mengintegrasikan *internet* dan aplikasi *E-Commerce* dengan beberapa aplikasi dan database yang telah ada.

- Kemungkinan adanya ketidakcocokan antara aplikasi *E-Commerce* dengan beberapa hardware atau komponen lainnya.

b. Batasan non-teknis

Biaya untuk mengembangkan *E-Commerce* sangat tinggi dan kurangnya pengalaman dapat menghasilkan keterlambatan. Industri *E-Commerce* mempunyai tugas yang sulit untuk meyakinkan konsumen bahwa transaksi secara online pada kenyataannya sangat aman. Kurangnya kepercayaan dari konsumen karena transaksi terjadi tidak secara *face-to-face* (bertatap muka) dan pembayarannya menggunakan uang elektronik.

### 2.1.2 Internet

*Internet* merupakan singkatan dari *International Network*, yaitu suatu jaringan yang memiliki jangkauan mencakup seluruh dunia sehingga memungkinkan sebuah komputer untuk terhubung dengan komputer lainnya yang jaraknya beribu-ribu kilometer.

*Internet* merupakan gabungan dari *LAN (Local Area Network)* dan *WAN (Wide Area Network)*. *LAN* atau *Local Area Network* merupakan suatu jaringan dari sejumlah komputer yang terhubung antara yang satu dengan yang lainnya. Seperti *LAN*, tetapi mempunyai jangkauan area yang lebih luas. Dari terhubungnya antara *LAN* satu dengan *LAN* yang lainnya dan terhubung dengan *WAN* lainnya maka akhirnya terbentuklah *Internet*.

## 2.2 Peralatan Pendukung ( *Toll System* )

### 2.2.1 PHP Hypertext Preprocessor

PHP (*Personal Home Page tools*) merupakan bahasa *scripting* (pemrograman) yang bersifat *server-side* dan dapat diintegrasikan dengan HTML sehingga HTML tidak lagi bersifat statis, namun menjadi bersifat

dinamis. Yang dimaksud dengan sifat *server-side* adalah pengerjaan *script* tersebut sepenuhnya dikerjakan di *server* dan kemudian hasil dari proses *script* tersebut dikirimkan kepada *web browser* dalam bentuk HTML.

Seluruh aplikasi yang berbasis *web* dapat dibuat dengan menggunakan PHP. Salah satu kelebihan utama PHP dibandingkan dengan bahasa pemrograman *web* lainnya adalah mempermudah aplikasi *web* untuk melakukan koneksi dengan *database*.

Kode program PHP menyatu dengan *tag-tag* HTML dalam satu file. Perintah dari PHP selalu dimulai dengan tanda '`<?>`' dan diakhiri dengan tanda '`?>`'. Untuk mengakhiri setiap akhir perintah yang dilaksanakan atau diproses diberi tanda ':'. File yang berisi tag HTML dan kode PHP ini diberi ekstensi `.php` atau ekstensi lainnya yang ditetapkan pada *Apache web server*. Berdasarkan *ekstensi* ini, pada saat *file* diakses, server tahu bahwa file ini mengandung kode PHP. Server menerjemahkan kode ini dan menghasilkan *output* dalam bentuk *tag* HTML yang dikirim ke browser client yang mengakses file tersebut. Pada PHP tidak ada deklarasi variabel yang dibutuhkan. Semua variabel dapat diakses kapan saja dengan nama yang harus berlainan antara variabel yang satu dengan variabel lainnya. Penanda variabel dari PHP adalah tanda '\$' pada bagian awal suatu kata seperti '\$query' ataupun '\$hasil'. Sedangkan untuk struktur bahasa pemrograman PHP mirip dengan struktur bahasa pemrograman C, contohnya : `Printf`, `echo`, `for`, `while` dan masih banyak lagi lainnya.

Kelebihan-kelebihan dari PHP, antara lain :

a. Koneksi ke *database*

PHP mendukung banyak sekali database dalam mode native dan melalui ODBC. Sistem database yang telah didukung oleh PHP hingga saat ini adalah:

*Oracle, Sybase, mSQL, MySQL, Solid, Generic, ODBC, PostgreSQL, Adabas D, FilePro, Velocis, dBase, Unix dbm* dan semua database dengan *interface ODBC*.

b. HTTP *Cookie*

PHP secara transparan mendukung HTTP cookie. Cookie adalah mekanisme untuk menyimpan data pada browser sehingga administrator dapat melacak atau mengidentifikasi user. Administrator dapat menyet cookies menggunakan

*Function getcookie ( )*. Cookie adalah bagian dari HTTP header, sehingga *Function getcookie ( )* harus dipanggil sebelum ada output yang dikirimkan ke browser. Hal ini sama dengan saat menggunakan *function header ( )*. Setiap cookie yang dikirimkan ke administrator dari client secara otomatis dianggap sebagai variabel PHP seperti halnya variable dari metode **GET** dan **POST**. Jika hendak mengisi banya data untuk satu cookie. Hanya dengan menambahkan tanda [ ] pada nama *cookie*.

c. HTTP *Authentication* dengan PHP

Kelebihan HTTP *Authentication* pada PHP hanya tersedia sejak PHP berjalan sebagai modul pada *web sever Apache*. Pada script PHP, *administrator* dapat mengirimkan sebuah pesan "*Authentication*

*Required*' ke *browser client* yang menyebabkan *browser* mengeluarkan input window untuk meminta username dan password. Setelah pemakai mengisi *username* dan *password*. URL yang berisi script PHP dipanggil lagi dengan variable `$PHP_AUTH_USER`, `$PHP_AUTH_PW` dan `$PHP_AUTH_TYPE` yang masing-masing isinya adalah username, password, dan *authentication type*.

d. *Regular Expression*

*Regular Expression* digunakan untuk manipulasi *string* yang kompleks pada PHP. Fungsi *regular expression* adalah: `ereg( )`, `ereg_replce( )`, `eregi( )`, `eregi_replace( )` dan `split( )`. Fungsi-fungsi ini mengambil *regular expression string* dari argument pertamanya. PHP menggunakan *Posix extended regular expression* yang didefinisikan pada Posix 1003.2.

e. File Upload

PHP mampu menerima file yang di-*upload* dari setiap *browser* yang RFC-1867 compliant. Kemampuan *upload* ini memungkinkan pemakai untuk meng-*upload file* teks atau *binary*. Dengan fungsi-fungsi *authentication* dan *logic* PHP, dimiliki control untuk menentukan siapa yang boleh meng-*upload* dan apa yang harus dikerjakan terhadap file tersebut setelah file itu selesai di-*upload*.

f. Pengaksesan File Gambar dengan PHP

PHP tidak terbatas hanya untuk menghasilkan output halaman HTML. PHP juga dapat digunakan untuk menghasilkan file gambar dengan format antara lain: GIF, JPEG, PNG, dan WBMP. Bahkan agar lebih elegan gambar dapat bersifat *stream* dari *server* ke *browser*. Namun, PHP terlebih

dahulu harus di-*compile* dengan GD library yang mengandung fungsi-fungsi manipulasi gambar.

g. *Error Handling*

Seluruh PHP *expression* dapat dipanggil dengan prefix “&”, yang mematikan *error reporting* untuk *expression* itu. Jika terjadi *error* pada *expression* itu dan *feature track errors* diaktifkan. Untuk mendapatkan pesan *error* dapat menggunakan *variable global* `$php_errormsg`.

h. *Server side*

*Script* PHP tidak memerlukan adanya *kompatibilitas browser* atau harus menggunakan *browser* tertentu. *Server* yang mengerjakan *script* tersebut dan hasil yang dikirimkan kembali ke *browser* biasanya dalam bentuk teks ataupun gambar sehingga dapat dikenali oleh *browser* apapun.

i. *Open source*

*Source code* dari PHP dapat digunakan, diganti atau diubah tanpa dikenai biaya.

j. *Multi platform*

*Script* PHP selain dapat dijalankan pada *platform* LINUX, juga dapat dijalankan pada *platform* Windows dengan menggunakan *software* PHP untuk *Windows*, dengan *web server* IIS pada *Windows* 2000/XP ataupun PWS pada *Windows* 98.

k. *Script* asli PHP tidak dapat dilihat oleh *browser* sehingga keamanan lebih terjamin.



### 2.2.2 Apache Web Server

*Apache* adalah salah satu *web server* yang dapat dijalankan pada beberapa macam sistem operasi. Ada beberapa keuntungan dengan menggunakan *web server Apache*, yaitu:

- a. *Apache* mudah dikonfigurasi dan digabungkan dengan beberapa modul tambahan lainnya. Contoh: modul PHP dan modul *header*.
- b. *Apache* merupakan *web server* yang gratis dan *open source*, sehingga untuk pemakaiannya tidak perlu melakukan registrasi atau membayar ijin (*license*) dan *user* diperbolehkan mengubah kode asli dari program *web server* ini.
- c. *Apache* dapat dijalankan pada beberapa sistem operasi, yaitu: Windows NT/9x, Netware 5.x, OS/2, dan UNIX/Linux.

Meskipun beberapa *web server* lain yang merupakan *web server* komersial telah menyatakan bahwa kecepatannya melebihi *Apache*, tetapi kecepatan *Apache* masih dapat diandalkan. Selain itu yang lebih menarik adalah *Apache* merupakan *web server* yang gratis dan *Apache* telah digunakan pada beberapa situs yang memiliki jumlah pengunjung jutaan per hari.

### 2.2.3 MySQL

MySQL adalah sebuah aplikasi RDBMS (*Relational Database Management Server*) yang sangat cepat dan kokoh serta bersifat *open source*. MySQL merupakan sebuah *server* basis data yang banyak digunakan di *internet* karena keandalannya dan juga karena sifatnya yang *shareware*. Bahasa yang digunakan dalam MySQL sama dengan bahasa yang dipakai oleh *server* basis data lainnya, yaitu bahasa SQL. SQL (*Structured Query*

*Languange*) adalah bahasa standar yang digunakan untuk mengakses server basis data.

Dalam konteks bahasa SQL, pada umumnya informasi tersimpan dalam table-tabel yang secara logik merupakan struktur dua dimensi yang terdiri atas baris-baris data (*row* atau *record*) yang berada dalam satu atau lebih kolom (*column*). Baris pada *table* sering disebut sebagai *attributes* atau *field*. Keseluruhan table itu dihimpun dalam satu kesatuan yang disebut basis data.

Sebagai pertimbangan, MySQL mempunyai banyak keistimewaan yang ditawarkan, antara lain:

- a. *Open source*, yaitu MySQL didistribusikan secara *open source* (gratis), di bawah lisensi GPL (*General Public License*) sehingga anda dapat menggunakannya secara cuma-cuma tanpa dipungut biaya sepeser pun.
- b. *Speed* (kecepatan), yaitu MySQL bekerja dengan cepat. Para pengembang berpendapat bahwa MySQL adalah basis data yang tercepat yang pernah didapat.
- c. *Ease of use* (kemudahan dalam penggunaan), yaitu MySQL adalah system basis data yang sederhana dengan performa tinggi dan sangat mudah untuk di-*setup* dan sangat mudah dalam pengaturan, dibandingkan dengan sistem basis data yang lebih besar.
- d. *Query languange support* (mendukung bahasa *query*), yaitu MySQL memahami SQL (*Structured Query Languange*), bahasa umum yang digunakan pada hampir semua basis data modern. MySQL juga dapat diakses menggunakan aplikasi yang mendukung ODBC (*Open Database Connectivity*), yaitu sebuah *protocol* untuk komunikasi basis data yang dikembangkan oleh Microsoft.

- e. *Capability* (kemampuan), yaitu banyak klien dapat melakukan koneksi ke *server* pada saat bersamaan. Klien dapat melakukan koneksi ke *server* pada saat bersamaan. Terdapat beberapa macam *interface* yang dapat digunakan untuk melakukan koneksi ke basis data dan melakukan *queries* serta menampilkan hasil. Sebagai tambahan bermacam-macam *programming interface* yang tersedia adalah C, Perl, Java, PHP, dan Python.
- f. *Connectivity and Security* (konektifitas dan keamanan), yaitu MySQL terhubung dengan jaringan secara keseluruhan dan basis data dapat diakses dari manapun melalui *internet*, sehingga dapat membagi data dimana saja. Dan MySQL memiliki *control akses (privilege)* sehingga siapa saja yang tidak diperbolehkan melihat data tidak dapat melihat data.
- g. *Portability* (dapat dipindahkan), yaitu MySQL dapat berjalan di banyak varian dari UNIX dan juga dapat berjalan pada system selain system UNIX, seperti Windows dan OS/2. MySQL dapat berjalan pada PC rumah (home PC) sampai dengan PC server dengan teknologi tinggi.
- h. *Open distribution* (distribusi yang terbuka), yaitu MySQL mudah untuk didapatkan, hanya dengan menggunakan *web browser* sudah bisa mendapatkan source dari MySQL.

Pada semua distribusi MySQL sudah terdapat beberapa *tools* (perlengkapan), antara lain:

- a. *SQL server*, yaitu program utama (*engine*) dari MySQL yang menyediakan akses ke basis data.
- b. *Client Program*, yaitu program interaktif klien yang digunakan untuk melakukan berbagai perintah pada basis data MySQL, antara lain:

- Program untuk melakukan *queries* secara langsung pada basis data dan menampilkan hasilnya secara langsung.
  - Program administrasi dan beberapa program kecil tambahan yang dapat digunakan untuk menjalankan situs.
  - Program untuk mengatur akses ke basis data.
  - Program untuk melakukan impor dan ekspor basis data.
- c. *Client library, library* yang disediakan untuk mempermudah pemrograman, terutama dengan menggunakan bahasa C karena *library* ini menggunakan bahasa C. Tetapi *library* ini juga dapat digunakan untuk bahasa pemrograman yang lain.

#### 2.2.4 Basis Data

Basis data terdiri dari 2 kata, yaitu basis dan data. Basis dapat diartikan sebagai markas, gudang atau tempat berkumpul. Sedangkan data dapat diartikan sebagai representasi fakta dunia nyata seperti manusia, hewan, konsep, keadaan dan sebagainya, yang direkam dalam bentuk angka, huruf, symbol, teks, gambar, bunyi, atau kombinasinya.

Basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang, seperti:

- a. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian 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 (*redundansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan.

c. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Prinsip kerja dari basis data yaitu pengaturan data atau arsip. Sedangkan tujuannya yaitu kemudahan dan kecepatan dalam pengambilan kembali data atau arsip yang diinginkan.

Basis data menggunakan media penyimpanan elektronik seperti disk (disket atau hardisk). Basis data dikelola atau ditangani melalui perantara alat atau mesin pintar elektronik (yang dikenal sebagai komputer).

### 2.2.5 Konsep Basis Data

Basis data merupakan kumpulan dari data komputer yang diintegrasikan, diorganisir dan disimpan dengan cara yang mudah untuk diakses. Untuk itu digunakan *direct access storage*. Integrasi secara logika dari *record* di beberapa file disebut konsep basis data. Dalam pembuatan basis data setiap unsur yang membangunnya harus diperhatikan dengan seksama karena setiap basis data dibuat untuk dapat berjalan dalam waktu yang lama serta dapat menampung data yang banyak tanpa perlu kehilangan *performance*, sehingga sejak awal dalam pembuatan basis data tersebut haruslah dibuat secepat mungkin tanpa harus kehilangan karakteristik dari basis data itu sendiri.

Ketika mulai menyusun perencanaan dan desain model basis data, hal yang perlu menjadi perhatian adalah pengelompokan atau pengumpulan data-data yang memiliki karakteristik yang sama ke dalam satu kelas. Dengan adanya kompleksitas data yang sangat tinggi maka dapat dipastikan kita mendapatkan banyak sekali pengelompokan kelas-kelas data. Langkah selanjutnya adalah membentuk beberapa file yang menampung kelas-kelas

sejenis, dimana *file-file* tersebut berisi informasi-informasi yang beraneka ragam yang juga merupakan satu kesatuan dengan kelas-kelas tertentu. Pada masing-masing *file*, disiapkan suatu informasi yang berbeda dengan *file-file* lainnya dimana dapat mewakili unsur informasi *file* basis data tersebut.

Tujuan dari pengelompokan data yang ada ke dalam beberapa *file* adalah untuk mengantisipasi terjadinya pengulangan data, menghemat kapasitas tempat penyimpanan data, dan membangun suatu struktur basis data yang baik dan teratur sehingga memudahkan dalam proses pencarian data dan efisiensi waktu.

### 2.2.6 Data Flow Diagram

Data Flow Diagram adalah diagram yang menggunakan notasi-notasi (simbol-simbol) yang digunakan untuk menggambarkan arus dari data sistem dan untuk membantu didalam komunikasi dengan pemakai sistem secara logika. (Jogiyanto,700)

Simbol-simbol dari Data Flow Diagram terdiri atas :

#### 1. Kesatuan Luar (External Entity)

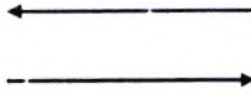
Merupakan kesatuan dilingkungan luar sistem yang dapat berupa orang, organisasi, atau sistem lainnya yang akan memberikan input ataupun menerima output dari sistem.



Gambar 2.1. Simbol External Entity

## 2. Arus Data (Data Flow)

Arus data menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau pun hasil dari proses sistem.



Gambar 2.2. Simbol Arus Data.

## 3. Proses (Process)

Proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses



Gambar 2.3. Simbol Proses

## 4. Simpanan Data (Data Store)

Simpanan data merupakan simpanan dari data yang dapat berupa suatu file atau database, arsip atau catatan manual kotak tempat penyimpanan, tabel dan agenda.



Gambar 2.4. Simbol Simpanan Data

### 2.2.7 Kamus Data (*Data Dictionary*)

Kamus data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data dapat dibuat dan digunakan dengan baik pada tahap analisa sistem maupun pada tahap perancangan sistem. Pada tahap analisa sistem, kamus data digunakan sebagai

alat komunikasi antara analis sistem dan user tentang data yang mengalir pada sistem tersebut serta informasi yang dibutuhkan oleh pemakai sistem (*user*). Sedangkan pada tahap perancangan sistem, kamus data digunakan untuk merancang input, merancang laporan dan database.

#### A. Parameter Kamus Data

Kamus data dibuat berdasarkan arus data yang ada pada Diagram Alir Data (DAD). Arus data di DAD sifatnya adalah global hanya ditunjukkan nama arus datanya saja, keterangan lebih lanjut dari suatu arus data di DAD secara lebih terperinci dapat dilihat pada kamus data. Kamus data harus dapat mencerminkan keterangan yang jelas tentang data yang dicatatnya. Untuk maksud keperluan ini maka kamus data harus memuat hal-hal berikut ini:

##### 1) Nama Arus Data

data Untuk mempermudah mereka yang membaca Diagram Alir Data dan yang memerlukan penjelasan lebih lanjut tentang suatu arus.

##### 2) Alias

Alias perlu ditulis karena data yang sama mempunyai nama yang berbeda untuk orang lain atau departemen satu dengan yang lainnya.

##### 3) Bentuk Data

Telah diketahui bahwa arus data dapat mengalir dari hasil suatu proses ke proses yang lainnya. Data yang mengalir ini biasanya dalam bentuk laporan serta dokumen hasil cetakan komputer. Bentuk data yang mengalir dapat berupa dokumen dasar atau formulir, dokumen hasil cetakan komputer, laporan tercetak, tampilan layar dimonitor, variabel, parameter dan field-field, semua itu dapat digunakan untuk pengelompokan kamus data sewaktu perancangan sistem.



#### 4) Arus Data

Arus Data menunjukkan dari mana data yang mengalir dan ke mana data akan menuju.

#### 5) Penjelasan

Untuk lebih menjelaskan lagi tentang makna arus data yang dicatat dikamus data.

#### 6) Periode

Digunakan untuk mengidentifikasi kapan input data harus dimasukkan ke dalam sistem, kapan proses program harus dilakukan dan kapan laporan harus dihasilkan.

#### 7) Struktur Data

Menunjukkan arus data yang dicatat pada kamus data yang terdiri dari item-item atau elemen-elemen data.

Kamus data juga menjelaskan tanda-tanda khusus yang dipergunakan pada kamus data. Ditunjukkan agar para pembaca bisa mengerti maksud dan nama-nama alias maupun tanda-tanda yang telah dibuat.

Notasi atau simbol yang digunakan dibagi menjadi dua macam yaitu sebagai berikut:

##### 1. Notasi tipe data

Notasi ini digunakan untuk membuat spesifikasi format input maupun output suatu data. Notasi yang umum digunakan antara lain adalah :

Tabel 2.1 Notasi tipe data.

Notasi	Keterangan
X	Setiap karakter
9	Angka numeric
A	Karakter Alphabet
Z	Angka nol ditampilkan sebagai spasi kosong
.	Titik, sebagai pemisah ribuan
,	Koma, sebagai pemisah pecahan
-	Hypen, sebagai tanda penghubung ( contoh : 021-8457677)
/	Slash, sebagai tanda pembagi ( contoh : 25/07/2007 )

## 2. Notasi struktur data

Notasi keterangan ini dipergunakan untuk membuat spesifikasi element data. Dimana notasi yang umum digunakan adalah sebagai berikut :

★ Tabel 2.2 Notasi struktur data ★

Notasi	Keterangan
=	Terdiri dari
+	And (dan)
()	Pilihan (boleh Ya atau Tidak)
{ }	Iterasi atau perulangan proses
[ ]	Pilihan salah satu pilihan
	Pemisah pilihan didalam tanda [ ]
*	Keterangan atau catatan
@	Petunjuk (key filed)

### 2.2.8 Normalisasi

Ketika kita merancang suatu basis adalah untuk suatu sistem relasional, prioritas utama dalam mengembangkan model data logikal adalah dengan merancang suatu perpresentasi data yang tepat bagi relationship dan constrainnya (batasanya). Kita harus mengidentifikasi suatu set relasi yang cocok, demi mencapai tujuan diatas. Teknik yang dapat kita gunakan untuk membantu mengidentifikasi relasi-relasi tersebut dinamakan Normalisasi.

Normalisasi itu adalah proses pengelompokan elemen data kedalam bentuk tabel yang menyatakan entitas-entitas hubungan mereka sehingga terwujud satu bentuk yang memudahkan adanya perubahan dengan dampak sekecil mungkin.

Salah satu alasan mengadakan normalisasi adalah untuk memastikan bahwa model data relational dapat berjalan dengan baik. Ini tidak berarti bahwa struktur yang tidak normal, tidak akan berjalan tetapi akan menyebabkan beberapa masalah ketika program aplikasi berusaha untuk mengubah sistem landas data. Bila ada kesulitan pada pengujian tersebut maka relasi tipe tersebut dipecah pada beberapa tabel lagi atau dengan kata lain perancangan belum mendapatkan database yang optimal.

#### a. Konsep Dasar Normalisasi

Sebelum mengenal tentang normalisasi lebih jauh ada beberapa konsep yang harus dilakukan terlebih dahulu yaitu:

##### 1) Pengertian Ketergantungan Fungsional (*Functionally Dependent*)

Ketergantungan fungsional menggambarkan hubungan antara atribut-atribut dengan relasi. Sebagai contoh: Jika A dan B adalah atribut-atribut dari relasi R&B dikatakan functionally dependent (ketergantungan

fungsional) terhadap A (dinotasikan dengan A ke B), jika masing-masing nilai dari A dalam relasi R berpasangan secara tepat dengan satu nilai dari B dalam relasi R

**2) Pengertian *Fully Functionally Dependent* (FFD)**

Suatu rinci dikatakan *Fully Functionally Dependent* pada suatu kombinasi rincian data jika *functional dependet* pada kombinasi rinci data dan tidak *functional dependent* pada bagian lain dari kombinasi rinci data. Yang dimaksud sebagai bagian lain dari suatu kombinasi rinci data disini adalah kombinasi rinci data selain set *funtional dependent* dan set kosong.

**3) Ketergantungan *Parsial***

Maksudnya sebagian dari kunci itu dapat digunakan sebagai kunci utama.

**4) Ketergantungan *Transitif***

Maksudnya menjadi atribut biasa pada suatu relasi tetapi menjadi kunci pada relasi lain.

**5) Determinan**

Adalah suatu atribut (*field*) atau gabungan atribut dimana beberapa atribut lain bergantung sepenuhnya pada atribut tersebut.

**b. Tahapan Normalisasi**

Atribut Kunci (Key Attribute)

Kunci atribut adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data (*row*) dalam table secara unik. Artinya, jika suatu atribut dijadikan kunci (*key*), maka tidak boleh ada dua atau lebih baris data dengan nilai yang sama untuk atribut tersebut.

Ada beberapa jenis kunci (*key*) dalam database, diantaranya :

### 1) Superkey

Superkey adalah merupakan satu atau lebih atribut (kumpulan atribut) yang dapat membedakan setiap baris data dalam sebuah table secara unik, bisa terjadi ada lebih dari satu kumpulan atribut yang bersifat superkey pada sebuah tabel.

### 2) Kunci Calon (Candidate Key)

Kunci Calon adalah merupakan kumpulan atribut minimal yang dapat membedakan setiap baris dalam tabel secara unik.

### 3) Kunci utama (Primary Key)

Kunci Utama adalah merupakan satu atribut atau kumpulan atribut minimal yang tidak hanya mengidentifikasi secara unik suatu kejadian yang spesifik, tetapi juga mewakili setiap kejadian dari suatu eriti.

### 4) Kunci Alternatif (Alternatif Key)

Kunci Alternatif adalah kunci kandidat yang tidak dipakai sebagai primary key tetapi sering dipakai sebagai kunci pengurutan.

### 5) Kunci Tamu (Foreign Key)

Kunci Tamu adalah satu atribut atau kumpulan atribut yang melengkapi satu relationship hubungan yang menunjukkan keinduknya (Parent).

## c. Langkah- langkah normalisasi

Didalam pembentukan Normalisasi terdapat pula beberapa langkah-langkah pembentukan dalam normalisasi yaitu:

### 1) Bentuk Tidak Normal (*Unnormalized Form*)

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja data tidak lengkap atau terduplikasi.

## 2) First Normal Form (*1NF / first normal format form*)

Pada tahap ini dilakukan penghilangan beberapa group elemen yang berulang agar menjadi satu harga tunggal yang berinteraksi diantara setiap baris pada suatu tabel, dan setiap atribut harus mempunyai nilai data yang atomic (bersifat atomic value). Atom adalah zat terkecil yang masih memiliki sifat induknya, bila dipecah lagi maka ia tidak memiliki sifat induknya.

## 3) Second Normal Form (*2NF / second normal form*)

Yaitu mempunyai ciri yang telah memenuhi kriteria bentuk normal kesatu dan atribut bukan kunci harus bergantung secara fungsi pada kunci utama. atau *Primary key*.

## 4) Third Normal Form (*3NF / third normal form*)

Yaitu relasi harus dalam bentuk normal kedua dan semua atribut bukan primer tidak punya hubungan yang transitif.

## 5) Boyce Codd Normal Form (BCNF)

Suatu relasi dikatakan telah memenuhi kriteria BCNF, jika dan hanya setiap determinan adalah suatu candidate key. Boyce-Codd Normal Form(BCNF) tidak mengharuskan suatu relasi harus sudah dalam bentuk normal ketiga (3-NF), baru bisa dibuatkan kedalam BCNF. Oleh karena itu untuk melakukan uji BCNF kita hanya mengidentifikasi seluruh determinan yang ada pada suatu relasi, lalu pastikan determinan-determinan tersebut adalah candidate key. Sehingga bisa dikatakan bahwa BCNF lebih baik dari bentuk normal ketiga (3-NF), dengan demikian setiap relasi didalam BCNF juga merupakan relasi dalam 3NF belum tentu merupakan relasi didalam BCNF.