**BAB II**

**TINJAUAN PUSTAKA**

## Sistem

Definisi sistem berkembang sesuai dengan konteks dimana pengertian sistem tersebut digunakan. Pada bab ini akan dijelaskan mengenai pengertian sistem, karakteristik sistem dan juga klasifikasi sistem.

### Pengertian Sistem

Sistem berasal dari bahasa Latin (*systēma*) dan bahasa Yunani (*sustēma*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan. Secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur-unsur variable-variabel yang saling terorganisasi, saling berintraksi dan saling tergantung menurut murdick dan ross (1993). Sistem juga merupakan kesatuan bagian-bagian yang saling berhubungan yang berada dalam suatu wilayah serta memiliki item-item penggerak, contoh umum misalnya seperti negara. Negara merupakan suatu kumpulan dari beberapa elemen kesatuan lain seperti provinsi yang saling berhubungan sehingga membentuk suatu negara dimana yang berperan sebagai penggeraknya yaitu rakyat yang berada dinegara tersebut.

Kata "sistem" banyak sekali digunakan dalam percakapan sehari-hari dalam forum diskusi maupun dokumen ilmiah. Kata ini digunakan untuk banyak hal dan pada banyak bidang, sehingga makna yang dimiliki oleh kata “sistem” itu sendiri menjadi beragam. Dalam pengertian yang paling umum, sebuah sistem adalah sekumpulan benda yang memiliki hubungan di antara mereka [2].

### Karakteristik Sistem

Karakteristik sistem dapat dibagi menjadi beberapa jenis, yaitu sebagai berikut [2]:

1. Batasan (*Boundary*)

Penggambaran dari suatu elemen atau unsur yang mana termasuk di dalam sistem dan yang mana di luar sistem.

1. Lingkungan (*Environment*)

Segala sesuatu di luar sistem, lngkungan yang menyediakan asumsi, kendala, dan input terhadp suatu sistem.

1. Masukan (*Input*)

Sumber daya (data, bahan baku, peralatan, energi) dari lngkungan yang dikonsumsi dan dimanipulasi oleh suatu sistem.

1. Keluaran (*Output*)

Sumber daya atau produk (informasi, laporan, dokumen, tampilan layar komputer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.

1. Komponen (*Component*)

Kegiatan-kegiatan atau proses dalam suatu sistem yang mentransformasikan input menjadi bentuk setengah jadi (output). Komponen ini bisa merupakan subsistem dari sebuah sistem.

1. Penghubung (*Interface*)

Tempat dimana komponen atau sistem dan lingkungannya bertemu atau berinteraksi.

1. Penyimpanan (*Storage*)

Area yang dikuasai dan digunakan untuk penyimpanan sementara dan tetap dari informasi, energi, bahan baku, dan sebagainya. Penyimpanan merupakan suatu media penyangga di antara komponen tersebut bekerja dengan berbagai tingkatan yang ada dan memungkinkan komponen yang berbeda dari berbagai data yang sama.

### Klasifikasi Sistem

1. Sebagai sistem abstrak (*abstract system*)
2. Sebagai sistem alamiah (*natural system*)
3. Sebagai sistem tertentu (*determinate system*) dan sistem tak tentu (*probabilistic system*)
4. Sebagai sistem tertutup (*closed system*) dan sistem terbuka (*open system*)
   1. **Pengelolaan Arsip**

Secara umum terdapat banyak tipe mengenai pengelolaan. Pada Bab ini akan dijelaskan secara khusus mengenai pengelolaan arsip yaitu tentang pengertian pengelolaan arsip, peranan kearsipan, dan model pengelolaan arsip.

* + 1. **Pengertian Pengelolaan Arsip**

Pengelolaan arsip adalah suatu proses pengawasan, penyimpanan, dan pengamanan dokumen serta arsip, baik dalam bentuk kertas maupun media elektronik. Suatu lembaga yang mempunyai pengelolaan arsip dengan baik pasti memudahkan lembaga tersebut dalam menemukan rekam jejak lembaga/ intitusi tersebut [3].

* + 1. **Peranan Kearsipan**

Kearsipan mempunyai peranan sebagai pusat ingatan, sebagai sumber informasi dan sebagai alat pengawasan yang sangat diperlukan dalam setiap organisasi dalam rangkaian kegiatan perencanaan, penganalisaan, pengembangan, perumusan kebijaksanaan, pemngambil keputusan, pembuatan laporan, pertanggungjawaban, penilaian dan pengendalian setepat-tepatnya. Setiap kegiatan tersebut, baik dalam organisasi pemerintahan maupun swasta selalu ada kaitannya dengan masalah arsip [3].

* + 1. **Model Pengelolaan Arsip**

Ada dua model dalam mengelola arsip, yaitu [3]:

1. *Life Cycle Model* (Model Siklus Hidup)

Siklus hidup arsip merupakan suatu cara melihat bagaimanan arsip diciptakan dan digunakan. Sebuah siklus kehidupan adalah kumpulan dari beberapa fase daur hidup sebelum disusutkan/ dimusnahkan. Model siklus hidup ini memiliki fungsi untuk mengelola dokumen secara manual.

1. *Records Continuum Model* (Model Arsip Berkelanjutan)

Model manajemen arsip kontinyu adalah model yang bisa diterapkan pada arsip elektronis. Yang dimaksud kontinyu disini adalah bersambung atau menghubungkan antara masa lalu dengan masa sekarang, dan sekarang dengan masa yang akan datang. Manajemen arsip elektronis diperlukan karena dokumen sebuah perusahaan atau negara tidak berupa data fisik tetapi juga berupa elektronik.

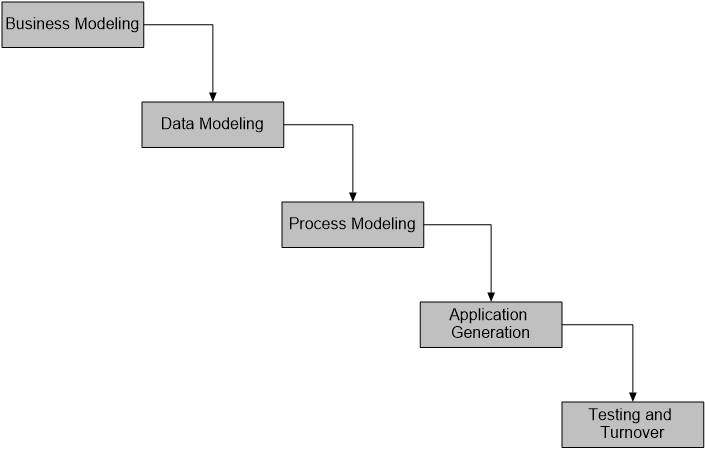
* 1. **Pengertian Sasaran Mutu**

Sasaran mutu merupakan target dari suatu organinsasi dalam melakukan proses yang ingin dicapai dalam jangka waktu tertentu. Sasaran mutu ini dilakukan, guna mengetahui seberapa target yang sudah direncanakan bisa tercapai.

Berdasarkan SNI 19-9000: 2001, sasaran mutu adalah sesuatu yang dicari atau dituju berkaitan dengan mutu. Dengan demikian manajemen puncak hendaknya menetapkan sasaran mutu yang menuju perbaikan kinerja organisasi [4].

* 1. **Metode RAD (*Rapid Application Development*)**

Metode RAD (*Rapid Application Development*) merupakan model proses pembangunan perangkat lunak yang tergolong dalam teknik incremental (bertingkat). RAD menekankan pada siklus pembangunan pendek, singkat, dan cepat. Adapun tahap-tahap yang digunakan dalam penerapan metode ini adalah [5]:



Gambar 2.1 Metode RAD (*Rapid Application Development*)

* 1. **Aplikasi Web**

Aplikasi web adalah sebuah sistem informasi yang mendukung interaksi pengguna melalui antarmuka berbasis web. Aplikasi web merupakan bagian dari *client-side* yang dapat dijalankan oleh browser web. Interaksi web dibagi menjadi 3 langkah, yaitu [6]:

1. Permintaan

Pengguna mengirimkan permintaan ke *server* web, biasanya via halaman web yang ditampilkan pada *browser* web.

1. Pemrosesan

*Server* web menerima permintaan yang dikirimkan oleh pengguna, kemudian memproses permintaan tersebut.

1. Jawaban

*Browser* menampilkan hasil dari permintaan pada jendela *browser*.

Ada beberapa sifat pemrograman web adalah sebagai berikut:

1. *Stateless*

Dengan arsitektur yang cenderung bebas, maka sebuah aplikasi web hampir tidak memiliki batasan bagi para pemakainya.

1. *Multi platform*

Sebuah aplikasi web harus mampu menjalankan dalam sebuah *browser* di lintas *platform* atau lintas operasi.

1. *Navigation sentris*

Sebuah aplikasi web lebih kearah *navigation sentries*. Hal ini secara umum akan membebaskan pengguna untuk mengerjakan sebuah proses tanpa harus melalui prosedur semestinya.

1. *Base on page*

Dengan konsep ini, maka sebuah aplikasi web juga mengandalkan nilai estetika selain nilai dari proses yang dilakukan itu sendiri. Berbeda dengan aplikasi *dekstop* yang lebih banyak menampilkan desain dari aplikasi itu sendiri, maka aplikasi web juga bergantung pada keindahan desain demi mendukung proses yang ada di dalamnya.

1. *Just browse*

Secara umum, sebuah aplikasi web hanya membutuhkan sebuah *browser* tanpa perlu insta lasi *runtime* ataupun aplikasi lain. Meski banyak pula aplikasi web yang membutuhkan *plug-in* tertentu dalam *browser,* tetapi hampir tiap *browser* yang ada telah memiliki fasilitas yang sama.

1. *Less secure*

Salah satu hal yang menjadi isu klasik dalam pemrograman web adalah masalah keamanan yang jauh lebih rentan dibanding aplikasi dekstop. Terlebih lagi jika aplikasi web tersebut oleh pengguna secara jamak di internet.

* 1. **XAMPP**

XAMPP adalah suatu bundel web server yang populer digunakan untuk coba-coba di windows karena kemudahan instalasinya. Bundel program open source tersebut berisi antara lain server web Apache, interpreter PHP, dan basis data MySQL. Setelah menginstall XAMPP kita bisa memulai pemrograman PHP di komputer sendiri maupun mencoba menginstall aplikasi-aplikasi web [6].

* 1. **PHP**

PHP adalah singkatan dari “*PHP Hypertext Processor*”, yang merupakan sebuah bahasa scripting yang terpasang pada HTML. PHP merupakan bahasa pemrograman yang paling sering digunakan oleh programming web karena merupakan bahasa pemrograman open source, sehingga para programming tidak perlu membeli lisensi untuk membuat web.

Rasmus Lerdorf merupakan orang yang membuat PHP pada tahun 1995. Pada waktu itu nama PHP adalah FI (*Form Interpreted*) yang merupakan sekumpulan *script*, digunakan untuk mengolah data form dari web. Pada perkembangan berikutnya, Rasmus akhirnya melepas kode sumber tersebut dan diberi nama PHP (*Personal Home Page*). PHP pertama yaitu 2.0 dirilis pertama pada bulan Nopember 1997. Dalam versi ini sudah ditambahkan modul-modul ekstensi yang menyebabkan kemampuan PHP menjadi lebih baik. Pada bulan Juni tahun 1998 perusahaan tersebut merilis versi baru PHP dengan nama PHP 3.0 dan mengubah kepanjangan PHP menjadi *Hypretext Preprocessor*. Pertengahan tahun 1999, Zend merilis versi baru dari PHP dan diberi nama PHP 4.0. PHP 4.0 merupakan versi PHP yang paling banyak dipakai oleh programming karena bisa untuk membuat aplikasi web yang lebih kompleks. Pada bulan Juni 2014 Zend akhirnya merilis versi baru lagi yang bernama PHP 5.0, ini merupakan versi mutakhir dari PHP dan sudah menerapkan model pemrograman berorientasi objek yang merupakan model yang banyak dikembangkan disemua bahasa pemrograman. PHP group kini sudah merilis versi baru dari PHP, yaitu versi 5.1.6 [7].

* 1. **CSS (*Cascading Style Sheet*)**

CSS adalah singkatan dari *Cascading Style Sheet* yang merupakan kumpulan perintah yang dibentuk dari berbagai sumber yang disususn menurut urutan tertentu sehingga mampu mengatasi konflik style. CSS atau disebut *Cascading Style Sheet* yaitu salah satu bahasa pemrograman web yang mengatur komponen dalam suatu web supaya web lebih terstruktur dan lebih seragam. CSS seperti halnya style pada aplikasi pengolahan kata Microsoft Word yang bisa mengatur beberapa style, misalnya subbab, heading, bodytext, footer, images dan style lainnya untuk dapat digunakan berbsama-sama dalam beberapa file. CSS biaanya digunakan untuk memformat tampilan halaman web yaang dibuat dengan bahasa HTML dan XHTML. CSS mampu mengatur warna body text, ukuran gambar, ukuran border, warna mouse over, warna tabel, warna hyperlink, margin kiri atau kanan atau atas atau bawah, spasi antar paragraf, spasi antar text dan parameter lainnya [8].

* 1. **Javascript**

Javascript adalah nama implementasi Netscape Communication Corporation untuk ECMAScript standar, suatu bahasa skrip yang didasarkan pada konsep pemrograman berbasis *prototipe.* Bahasa ini terutama terkenal karena penggunaanya di situs web (sebagai Javascript sisi klient) yang juga digunakan untuk menyediakan akses skrip untuk object yang dibenamkan (*embedded)* diaplikasi lain.

Walaupun memiliki nama serupa, Javascript hanya sedikit berhubungan dengan bahasa pemrograman java, dengan kesamaan utamanya adalah dengan menggunakan sintak C. Secara semantik, Javascript memiliki lebih banyak kesamaan dengan bahasa pemrograman self. Skrip Javascript yang dimasukkan di dalam berkas HTML ataupun XHTML harus dimasukkan diantara tag <script>..</script> [9].

* 1. **JQuery**

JQuery merupakan suatu *framework* Javascript yang menekankan bagaimana interaksi antara Javascript dan HTML. JQuery pertama kali dirilis pada tahun 2006 oleh John Resig. *Fitur* utama dari JQuery diantaranya [10]:

1. Dapat mengakses elemen dalam dokumen

Javascript khusus, untuk mengakses suatu bagian tertentu dari halaman, harus mengikuti aturan *document object model* dan pengaksesan harus secara spesifik menyesuaikan dengan struktur HTML.

1. Mengubah tampilan halaman website

CSS (*Cascading Style Sheet*) menawarkan metode yang cukup handal dalam mengatur dan mempercantik halaman web

1. Mengubah isi dari dokumen

Tidak hanya memberikan “kosmetik” pada halaman web, JQuery juga memberikan fasilitas untuk mengubah isi dari dokumen hanya dengan beberapa baris perintah.

1. Merespon interaksi *user*

Javascript sendiri memiliki beberapa event-handling seperti onclick untuk menangani *event* saat terjadi *click*.

1. Animasi pada dokumen

Animasi seringkali disertakan dalam suatu halaman web untuk menambah kecantikannya.

1. Mengambil informasi dari server tanpa harus me-*refresh* halaman Mengambil informasi dari server tanpa refresh halaman merupakan salah satu konsep dasar yang dikenal dengan nama AJAX (Asynchronous Javascript and XML).
2. Menyederhanakan penulisan sintaks Javascript

Semboyan JQuery adalah “*Write less, do more*” atau dengan kata lain kesederhanaan dalam penulisan *code*, tetapi menghasilnya tampilan yang lebih.

* 1. **Codeigniter**

Codeigniter adalah sebuah *framework* PHP yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi web berbasis PHP dibanding jika menulis semua kode program dari awal. Codeigniter pertama kali dibuat oleh Rick Ellis, CEO Ellislab, Inc. sebuah perusahaan yang memproduksi CMS (*Content Management System*) yang cukup handal, yaitu *Expression Engine*. Saat ini, CodeIgniter dikembangkan dan dimaintain *oleh Expression Engine Development Team*. Adapun beberapa keuntungan menggunakan Codeigniter, diantaranya [10]:

1. Gratis

Codeigniter berlisensi dibawah Apache/BSD opensorce.

1. Ditulis Menggunakan PHP 4

Meskipun Codeigniter dapat berjalan di PHP 5, namun sampai saat ini kode program Codeigniter masih dibuat dengan menggunakan PHP 4.

1. Berukuran Kecil

Ukuran Codeigniter yang kecil merupakan keunggulan tersendiri. Dibanding dengan framework lain yang berukuran besar.

1. Menggunakan Konsep MVC

Codeigniter menggunakan konsep MVC yang memungkinkan pemisahan layer application-logic dan presentation.

1. URL yang Sederhana

Secara default, URL yang dihasilkan Codeigniter sangat bersih dan Serach Engine Friendly (SEF).

* 1. **MySql**

MySql (*My Structure Query Language*) adalah salah satu database management system (DBMS) dari sekian banyak DBMS seperti oracle, MS SQL, Postagre SQL, dan lainnya. MySql berfungsi untuk mengolah database menggunakan bahasa SQL. MySql bersifat *open source* sehingga kita bisa menggunakannya secara gratis. MySql menggunakan perintah bahasa SQL, antara lain [11]:

1. *SELECT*

Perintah ini digunakan untuk mengambil data dari suatu tabel. Sintak penulisannya adalah:

SELECT{\*namafield} FROM namatabel [WHERE kondisi]

1. *INSERT*

Perintah ini digunakan untuk menyisipkan data ke dalam tabel. Sintak penulisannya adalah:

INSERT INTO nama tabel [(field1[,field2,…])

VALUE (ekpresi1[,ekpresi2,…])

1. *DELETE*

Perintah ini digunakan untuk menghapus record dari suatu tabel. Sintak penulisannya adalah:

DELETE FROM namatabel WHERE kondisi.

1. *UPDATE*

Perintah ini digunakan untuk memperbaharui nilai suatu data pada tabel. Sintak penulisaanya adalah:

UPDATE namatabel SET kriteria WHERE kondisi.

* 1. **Perancangan Terstruktur**

Perancangan terstruktur merupakan aktivitas mentransformasikan hasil analisis kedalam suatu perencanaan untuk dapat diimplementasikan. Perancangan terstruktur dilengkapi dengan alat-alat (*tools*), pada bab ini akan dibahas mengenai tools yang digunakan pada perancangan terstuktur dalam sistem yaitu *flowchart* dan *data flow diagram*.

### *Flowchart*

Diagram alur yang menggambarkan logika dari proses dan bentuk simbol tertentu sehingga memudahkan analisis sistem dalam membuat program. Bagan alir menggunakan serangkaian simbol standar untuk menguraikan prosedur pengolahan transaksi yang digunakan oleh sebuah perusahaan, sekaligus menguraikan aliran data dalam sebuah sistem. Bagian alir sistem digambarkan dengan menggunakan simbol yang tampak pada tabel 2.1 dibawah ini [12]:

Tabel 2. 1 Simbol-simbol *Flowchart*

|  |  |  |
| --- | --- | --- |
| **Nama** | **Lambang** | **Keterangan** |
| *Terminator* |  | Menunjukkan awal dan akhir dari alur suatu program |
| Proses |  | Menunjukkan proses perhitungan aritmatik |
| *Read/Write* |  | Menunjukkan sumber data yang akan diproses/dicetak |
| Operasi Manual |  | Menunjukkan suatu pekerjaan manual |
| Dokumen |  | Menunjukkan dokumen input/output hasil proses yang berjumlah satu dokumen saja |
| *Decision* |  | Menunjukkan proses evakuasi pemeriksaan |
| *Diskette* |  | Menunjukkan penyimpanan data ke dalam sebuah media direct accesss seperti disket |
| *Input* |  | Menunjukkan input data |
| Arsip |  | Menunjukkan file yang diarsip |
| *Arrow* |  | Menunjukkan arus dari suatu proses |
| Simbol Penghubung |  | Menunjukkan penghubung ke halaman yang masih sama atau ke halaman lain |

### 

### *Data Flow Diagram* (DFD)

*Data Flow Diagram* merupakan suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan serta proses yang dikenakan pada data tersebut. *Data Flow Diagram* dapat dibagi menjadi dua, yaitu [13]:

* 1. Diagram Konteks

Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh input ke sistem atau output dari sistem. Diagram ini akan memberi gambaran tentang keseluruhan sistem. Dalam diagram konteks hanya ada satu proses. Tidak boleh ada *store* dalam diagram konteks. Diagram konteks berisi gambaran umum (secara garis besar) sistem yang akan dibuat. Yang dibutuhkan pada diagram konteks adalah (1) Siapa saja pihak yang akan memberikan data ke sistem, (2) Data apa saja yang diberikannya ke sistem, (3) kepada siapa sistem harus memberi informasi atau laporan, dan (4) apa saja isi / jenis laporan yang harus dihasilkan sistem.

* 1. Diagram *Leveled*

Model ini menggambarkan sistem sebagai jaringan kerja antar fungsi yang berhubungan satu dengan yang lain dengan aliran dan penyimpanan data. Sebagai alat bantu perancangan sistem, model ini hanya memodelkan sistem dari sudut pandang fungsi. Dalam DFD Leveled ini akan terjadi penurunan level / *breakdown* dimana dalam penurunan level yang lebih rendah harus mampu merepresentasikan proses tersebut ke dalam spesifikasi yang jelas. Pada DFD *Leveled* terdapat dua jenis notasi untuk melakukan perancangannya, yaitu Notasi *Yourdon / DeMarco* dan Notasi *Gane & Sarson*. Namun pada laporan ini penulis hanya akan menggunakan *Notasi Yourdon / DeMarco* untuk melakukan analisa sistem.

Adapun penjelasan dari symbol-simbol yang digunakan pada DFD yaitu sebagai berikut :

1. *Terminator (External Entity)*

Simbol *terminator* digunakan untuk menggambarkan suatu entity luar yang dapat mengirim data atau menerima data dari suatu sistem. *Entity* luar di sebut juga sumber atau tujuan data, dan dianggap *external* terhadap sistem yang sedang di gambarkan. Setiap entitas diberi label dengan nama yang sesuai. Adapun simbol dari *terminator* dapat dilihat pada gambar 2.1 dibawah ini:

Gambar 2. 2 *Terminator*

Setiap sistem pasti memiliki batas sistem yang memisahkan suatu sistem dengan lingkungan luarnya. Kesatuan luar (*external entity*) merupakan kesatuan di lingkungan luar sistem yang berupa orang, organisasi atau sistem lainnya yang ada di lingkungan luarnya yang akan memberikan *input* atau menerima *output* dari sistem.*Terminator* dapat berupa orang, sekelompok orang, organisasi, departemen di dalam organisasi, maupun perusahaan yang sama namun di luar kendali sistem yang sedang dibuat modelnya.

1. *Data Flow* (Arus Data)

Arus Data pada *Data Flow Diagram* diberi simbol suatu panah. Arus data ini mengalir di antara proses, data *store*, dan *external entity*. Arus data ini menunjukkan arus data yang dapat berupa masukan untuk sistem maupun hasil dari proses sistem. Adapun simbol dari *data flow* dapat dilihat pada gambar 2.2 dibawah ini:

Gambar 2. 3 *Data Flow*

1. Proses

Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dan hasil suatu arus data yang masuk ke dalam proses untuk dilakukan arus data yang akan keluar dari proses. Suatu proses dapat ditunjukkan dengan simbol lingkaran maupun simbol persegi panjang tegak dengan sudut-sudut tumpul. Adapun simbol dari prosesdapat dilihat pada gambar 2.3 dibawah ini:



Gambar 2. 4 Proses

1. Simpanan Data (*Data Store*)

Simpanan data atau Data *Store* merupakan simpanan dari data yang dapat berupa file atau database di sistem komputer, arsip maupun buku. Data *Store* pada *Data Flow Diagram* disimbolkan dengan sepasang garis horizontal paralel. Adapun simbol dari *data store* dapat dilihat pada gambar 2.4 dibawah ini:



Gambar 2. 5 *Data Store*

### Perancangan *Database*

Perancangan Database adalah proses untuk menentukan isi dan pengaturan data yang dibutuhkan untuk mendukung berbagai rancangan sistem. Pada bab ini dibahas mengenai perancangan database dalam sistem yaitu *entity relationship diagram* dan konseptual database.

### *Entity Relationship Diagram*

*Entity Relationship Diagram* adalah suatu pemodelan dari basisdata relasional yang didasarkan atas persepsi di dalam dunia nyata, dunia ini senantiasa terdiri dari sekumpulan objek yang saling berhubungan antara satu dengan yang lainnya. Suatu objek disebut *entity* dan hubungan yang dimilikinya disebut *relationship.* Suatu entity bersifat unik dan memiliki atribut sebagai pembeda dengan entity lainnya [14]. ERDmengidentifikasikan empat jenis objek antara lain:

1. *Entity* merupakan objek nyata yang dapat dibedakan satu sama lain dan tidak saling bergantung.
2. *Attribute* menerangkan suatu *entity* dan mempunyai harga.
3. *Key* merupakan *identifier* yang unik dari suatu *entity* karena nilai dari *key* itu akan berbeda-beda untuk mesing-masing *entity.*
4. *Relationship* merupakan hubungan antar *entity*. Suatu *entity* juga dapat mempunyai suatu *atribute*.

Notasi-notasi yang digunakan pada *Entity Relationship Diagram* diantaranya dapat dilihat pada tabel 2.2 dibawah ini:

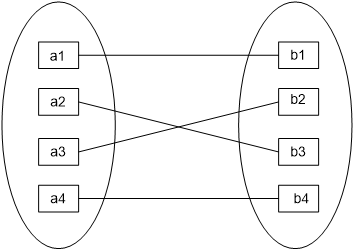
Tabel 2. 2 Simbol-simbol *Entity Relationship Diagram*

|  |  |
| --- | --- |
|  | *Entity* |
|  | *Relationship* |
|  | *Accosiative Entity* |
|  | *Attribute* |

* + 1. **Konseptual Database**

Konseptual Database bisa dianggap sebagai bentuk lengkap ERD, yaitu menggambarkan relasi yang terjadi antar entitas, dimana masing-masing entitas dilengkapi dengan atributnya serta menggambarkan artibut mana yang merelasikan entitas tersebut. Berdasarkan *cardinality ratio constraint*, menjelaskan jenis hubungan antara dua tipe entitas dinyatakan dengan istilah hubungan *one-to-one*, *one-to-many*, *many-to-one* dan *many-to-many*. Bila relasi *one-to-one* maka *foreign key* diletakkan pada salah satu dari dua entitas yang ada atau menyatukan ke dua entitas tersebut. Bila relasi *one-to-many* maka *foreign key* diletakkan di entitas yang many. Bila relasi *many-to-many* maka dibuat “file konektor” yang berisi dua *foreign key* yang berasal dari kedua entitas. Penjelasan masing-masing hubungan tersebut adalah sebagai berikut [14]:

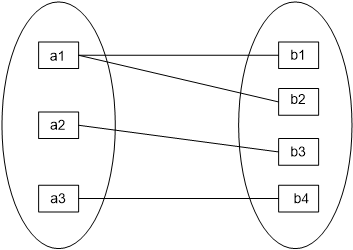
1. *One-to-one*



Gambar 2.6 *one-to-one*

Gambar 2.5 diatas menjelaskan bahwa setiap entitas pada tipe entitas A paling banyak berpasangan dengan satu entitas pada tipe entitas B. Begitu pula sebaliknya.

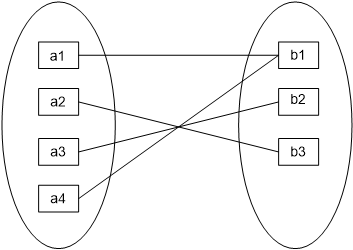
1. *One-to-many*



Gambar 2.7 *one-to-many*

Pada gambar 2.6 diatas menjelaskan bahwa setiap entitas pada tipe entitas A bisa berpasangan dengan banyak entitas pada tipe entitas B, sedangkan setiap entitas pada B hanya bisa berpasangan dengan satu entitas pada tipe entitas B.

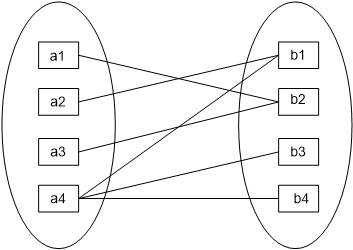
1. *Many-to-one*



Gambar 2.8 *many-to-one*

Pada gambar 2.7 diatas menjelaskan bahwa setiap entitas pada tipe entitas A paling banyak berpasangan dengan satu entitas pada tipe entitas B dan setiap entitas pada tipe entitas B bisa berpasangan dengan banyak entitas pada tipe entitas A.

1. *Many-to-many*



Gambar 2.9 *many-to-many*

Pada gambar 2.8 diatas menjelaskan bahwa setiap entitas pada suatu tipe entitas A bisa berpasangan dengan banyak entitas pada tipe entitas B dan begitu pula sebaliknya.