

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Pada jurnal pertama yang berjudul “Rancang Bangun Sistem Absensi Perkuliahan Auto ID Berbasis RFID” oleh Ahyar Jadid, dkk dari Jurusan Teknik Elektro Universitas Syiah Kuala, Banda Aceh. Di jurnal ini penulis membahas perancangan sistem absensi yang dapat melakukan perekaman data absensi dengan RFID yang terintegrasi dengan basis data berbasis *web*.. Data *id tag* dari hasil *reader* pada perangkat, kemudian akan dikirimkan ke basisdata dan menerima respon balasan berupa identitas dari pengguna *id tag* yang akan ditampilkan pada aplikasi *web*. Proses tersebut akan dilakukan apabila perangkat telat terkoneksi dengan jaringan lokal. Kelemahannya adalah dosen tidak dapat melihat data absensi dari luar kampus apabila dosen berhalangan untuk hadir dikarenakan data absensi hanya dapat diakses di kampus saja.

Pada jurnal kedua yang berjudul “Rancang Bangun Peminjaman Loker Menggunakan RFID” oleh Moh. Reza Syahrial, dkk dari Jurusan Sistem Komputer Institut Bisnis dan Informatika Stikom, Surabaya. Pada jurnal ini penulis menjelaskan bahwa penggunaan kartu RFID untuk peminjaman dan pengembalian loker. Kelemahan pada sistem ini adalah tidak tersedianya informasi ataupun data dari yang mengakses loker dikarenakan *database* yang digunakan tidak bisa diakses oleh pihak lainnya hanya berupa *fixed database*. Artinya data terlebih dahulu dimasukkan, kemudian sistem hanya akan bekerja otomatis dengan data yang tersimpan. Aplikasi *web* dari peminjaman loker ini juga tidak dapat menampilkan riwayat dari peminjaman, hanya menampilkan data peminjaman dan pengembalian saja.

Dari cara kerja kedua jurnal diatas, maka akan dibuat suatu sistem pengabsensian mahasiswa yang dapat diakses melalui *web*. Data absensi akan dimasukkan kedalam *database* yang dapat disimpan dalam PC maupun diakses melalui *smartphone*. Dimana di dalam aplikasi *web* absensi ini akan menampilkan *report* absensi, waktu absensi serta perhitungan kompensasi.

2.2 Pengertian Absensi

Absensi merupakan suatu aktivitas pelaporan dan pendataan kehadiran yang ada dalam sebuah institusi (Setiawan dan Kurniawan, 2015). Berdasarkan cara penggunaannya, sistem absensi dapat dikelompokkan menjadi manual dan digital. Sistem absensi manual yang dilakukan di perguruan tinggi mengharuskan mahasiswa mengisi formulir absensi dengan tandatangan pada saat perkuliahan (Yusuf, 2016). Sistem ini memiliki beberapa kekurangan seperti adanya penitipan tandatangan, memerlukan banyak kertas dan tinta, dan membutuhkan ruang yang banyak sebagai tempat penyimpanan. Kekurangan-kekurangan tersebut dapat diatasi dengan memanfaatkan teknologi yang dipadukan dengan sistem RFID. Sistem ini terdiri dari RFID *reader* dan *tag* yang dapat dikembangkan sebagai mesin absensi mahasiswa pada saat perkuliahan (Hamdani, 2014).

2.3 Pengertian Website

Website atau situs dapat diartikan sebagai kumpulan halaman yang digunakan untuk menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman *web* yang disebut *Hyperlink*, sedangkan teks yang dijadikan media penghubung disebut *Hypertext*.

Web terdiri dari page atau halaman, dan kumpulan halaman yang dinamakan *homepage*. *Homepage* berada pada posisi teratas, dengan halaman-halaman terkait berada di bawahnya. Biasanya setiap halaman di bawah *homepage* disebut *child page*, yang berisi *hyperlink* ke halaman lain dalam *web* (Agung, 2000:30).

Berikut merupakan jenis-jenis *website* yaitu

2.2.1. Website Statis

Website statis adalah suatu *website* yang memiliki halaman yang tidak berubah. Artinya, untuk melakukan sebuah perubahan pada suatu halaman hanya dapat dilakukan secara manual, yaitu dengan mengedit kode-kode yang menjadi struktur dari *website* itu sendiri.

Beberapa contoh *website* statis:

- a. *Website* perusahaan (*Company Profile*)
- b. Search engine (Google, Bing)

2.2.2 Website Dinamis

Website dinamis adalah suatu *website* yang secara strukturnya diperuntukkan untuk *update* sesering mungkin. *Website* ini selain utamanya untuk diakses oleh para pengguna juga telah disediakan halaman backend yaitu untuk mengedit kode dari *website* tersebut.

Beberapa contoh *website* dinamis:

- a. *Blog/ Website* Pribadi
- b. Katalog *Online*
- c. Situs *E-Commerce*
- d. Situs Berita
- e. *Website* Portal

2.2.3 Website Interaktif

Website interaktif adalah suatu *website* yang diperuntukkan untuk berinteraksi dengan orang lain secara online. Pengguna *website* jenis ini biasanya komunitas atau pengguna internet aktif dan dapat berinteraksi serta beradu argumen tentang apa yang sedang mereka pikirkan. Contohnya situs media sosial, situs forum *online*, dan *blog*.

Berikut merupakan manfaat dari *website* yaitu

- a. *Website* dapat menjadi sarana hiburan yang murah.
- b. *Website* menjadi media pengenalan perusahaan atau lembaga.
- c. *Website* dapat menjadi sarana informasi yang cepat dan mudah.
- d. *Website* dapat menjadi sarana edukasi masyarakat.
- e. *Website* dapat menjadi sarana berkomunikasi masyarakat diseluruh dunia.
- f. *Website* dapat menjadi sarana pemasaran dan promosi bisnis yang efektif dengan jangkauan terluas.

(Sumber : gurupendidikan.co.id)

2.4 Pengertian Database

Database merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di simpanan luar komputer serta digunakan perangkat lunak tertentu untuk memanipulasinya. Adalah salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan data bagi para pemakai.

Basis data (*database*) adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Pendefinisian basis data meliputi spesifikasi berupa tipe data, struktur data dan juga batasan-batasan pada data yang akan disimpan. Merupakan aspek yang sangat penting dalam sistem informasi karena berfungsi sebagai gudang penyimpanan data yang akan diolah lebih lanjut. Hal ini menjadi penting karena dapat mengorganisasi, menghindari duplikasi, dan mengantisipasi hubungan antar data yang tidak jelas dan juga update yang rumit (Hartono , 2005 : 17)

2.4.1 Pengertian MySQL

MySQL merupakan software yang tergolong *database server* dan bersifat *Open Source*. *Open Source* menyatakan bahwa *software* ini di lengkapi oleh *source code* (kode yang di pakai untuk membuat MySQL), selain tentu saja bentuk *executable*-nya atau kode dapat di jalankan secara langsung di dalam sistem operasi, dan bisa diperoleh dengan cara mengunduh di internet secara gratis. Hal lainnya adalah MySQL juga bersifat *multiplatform*. MySQL dapat dijalankan pada berbagai sistem operasi (Abdul Kadir, 2009:15).

Beberapa keistimewaan dari MySQL, antara lain :

1. Portabilitas, artinya stabil pada berbagai sistem operasi seperti Windows, Linux, Mac OS.
2. Didistribusikan secara *open source* atau bebas di bawah lisensi dari GPL.
3. *Multiuser*. Dapat digunakan oleh beberapa user dalam waktu secara bersamaan tanpa mengalami masalah.

4. *Performance tuning*. Mempunyai kecepatan dalam pemrosesan query dibandingkan mesin *database* lainnya.
5. Konektivitas. Melakukan koneksi dengan klien dengan menggunakan protokol TCP/IP, Unix socket, serta *Named Pipes*.
6. Dapat menangani perubahan struktur tabel dengan perintah ALTER TABEL dibandingkan PostgreSQL maupun Oracle.
7. Mendukung penggunaan tipe kolom yang kompleks seperti *signed/unsigned integer, float, double, char, date, timestamp*, dan lain sebagainya.
8. Mempunyai beberapa lapisan keamanan seperti *level subnetmask*, nama *host*, izin akses *user* dengan sistem perijinan yang mendetail serta terenkripsi.
9. Mempunyai *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*). Hal ini akan memudahkan pengguna atau *programmer* dalam melakukan pengolahan data.

2.4.2 Pengertian XAMPP

Pengertian XAMPP adalah perangkat lunak bebas (*free software*), yang mendukung untuk banyak sistem operasi. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*). Nama XAMPP merupakan singkatan dari X (untuk sistem operasi), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU (*General Public License*) dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis (Arief, 2011).

2.4.3 Pengertian PHPMyAdmin

PHPMyAdmin adalah suatu program *open source* yang berbasis *web* yang dibuat menggunakan aplikasi PHP. Program ini digunakan untuk mengakses *database* MySQL untuk mempermudah dan mempersingkat pekerjaan yang kita lakukan. Dengan kelebihanannya, para pengguna awam tidak harus paham *syntax-syntax* SQL dalam pembuatan *database* dan tabel.

Adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi MySQL melalui Jejaring Jagat Jembar (*World Wide Web*). PHPMyAdmin mendukung berbagai operasi MySQL, diantaranya (mengelola basis data, tabel-tabel, bidang (*fields*), relasi (*relations*), indeks, pengguna (*users*), perijinan (*permissions*), dan lain-lain. Pada dasarnya, mengelola basis data dengan MySQL harus dilakukan dengan cara mengetikkan baris-baris perintah yang sesuai (*command line*) untuk setiap maksud tertentu (Bunafit , 2004:64).

2.5 Pengertian HTML

Menurut Anhar (2010 : 40) menjelaskan bahwa HTML (*HyperText Markup Language*) adalah sekumpulan simbol-simbol atau *tag-tag* yang dituliskan dalam sebuah *file* yang digunakan untuk menampilkan halaman pada *web browser*. Dapat juga digunakan sebagai *link-link* antara file-file dalam situs atau dalam komputer dengan menggunakan *localhost*, atau *link* yang menghubungkan antar situs dalam dunia internet. Bisa disimpulkan HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman *web*.

2.6 Pengertian PHP

Menurut Anhar (2010 : 3) PHP merupakan sebuah bahasa pemrograman *server side scripting* yang bersifat *open source*. Merupakan sebuah *script* yang terintegrasi dengan HTML dan mampu menyajikan informasi yang dinamis

Beberapa kelebihan PHP sebagai berikut :

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan kompilasi dalam penggunaannya.
2. Bersifat *Open Source* yang berarti dapat digunakan oleh siapa saja secara gratis.
3. Banyak *web server* yang mendukung bahasa PHP seperti: Apache, Lighttpd, IIS dan lain-lain.

2.7 Pengertian Java

Menurut Hariyanto (2003), Java adalah bahasa pemrograman yang dapat dijalankan diberbagai komputer, termasuk telepon genggam. Bahasa Java memiliki fitur keamanan *built-in* yang membuat tugas pemrograman lanjut seperti pemrograman jaringan, pemrograman tersebar, koneksifitas *database*. Dan juga memiliki fitur *multithreading* sehingga mempermudah dalam pemrograman aplikasi multithreading, pemrograman menjadi lebih aman.

2.8 Pengertian Bootstrap

Bootstrap merupakan salah satu *framework* HTML, CSS, dan JS yang cukup populer, serta banyak digunakan oleh para pengembang *web* saat ini. *Framework* ini banyak digunakan untuk membuat *website* yang bersifat responsif. Artinya bisa menyesuaikan tampilan *layout* berdasarkan ukuran *viewport* dari *device* pengaksesnya, mulai dari *smartphone*, *tablet* atau layar PC. Hal ini didasarkan pada fakta di lapangan yang menunjukkan bahwa perkembangan penggunaan *handheld* seperti *smartphone* dan *tablet* semakin meningkat. Dengan *website responsif* tersebut akan memudahkan pengguna melakukan *resizing*, *scrolling*, dan *panning* dengan lebih optimal (Otto, 2011)

Ada beberapa alasan mengapa menggunakan *framework* Bootstrap, antarlain sebagai berikut (Otto, 2011) :

1. Bersifat *open source*. Bebas untuk dikembangkan oleh siapapun, didukung dengan banyaknya referensi dari pengembang lainnya.
2. *Support* terhadap berbagai macam *browser* yang banyak digunakan saat ini (*cross browser support*). Bootstrap menggunakan *normalize* CSS untuk mereset *browser*, membuat semua elemen dalam *website* melakukan render lebih tepat dan seragam pada *browser*, sehingga dapat dibuka dengan baik pada semua *browser*.
3. Waktu pembuatan yang lebih cepat Hal ini karena elemen-elemen yang biasa digunakan dalam sebuah *website* sudah disediakan *class-class* nya.
4. Pilihan desain yang *responsive* dan *non responsive* Kita bisa memilih opsi untuk mengaktifkan desain responsif atau tidak.

5. *Template* yang lebih rapi Hal ini karena Bootstrap sudah terstruktur sehingga untuk pengurangan/penambahan *class* dapat mengikuti kerapian kode yang telah dibuat.

2.9 Pengertian CSS

Menurut Jayan (2010 : 2) CSS adalah singkatan dari *Cascading Style Sheets*. Kegunaannya adalah untuk mengatur tampilan dokumen HTML, contohnya seperti pengaturan jarak antar baris, teks, warna dan format border bahkan penampilan file gambar. CSS berisi rangkaian instruksi yang menentukan bagaimana suatu *text* akan tertampil di halaman *web*. Perancangan desain *text* dapat dilakukan dengan mendefinisikan *fonts* (huruf) , *colors* (warna), *margins* (ukuran), latar belakang (*background*), ukuran font (font *sizes*) dan lain-lain. Elemen-elemen seperti *colors* (warna) , *fonts* (huruf), *sizes* (ukuran) dan *spacing* (jarak) disebut juga “*styles*”.

Cascading Style Sheets juga bisa berarti meletakkan *styles* yang berbeda pada *layers* (lapisan) yang berbeda. CSS terdiri dari *style sheet* yang memberitahukan *browser* bagaimana suatu dokumen akan disajikan. Fitur-fitur baru pada halaman *web* lama dapat ditambahkan dengan bantuan *style sheet*. Saat menggunakan CSS, Anda tidak perlu menulis *font*, *color* atau *size* pada setiap paragraf, atau pada setiap dokumen. Setelah anda membuat sebuah *style sheet*, anda dapat menyimpan kode tersebut sekali saja dan dapat kembali menggunakannya bila diperlukan.

2.10 Pengertian Sublime

Sublime Text 3 adalah sebuah *software* yang dikembangkan oleh Jon Skinner. Beliau merupakan seorang *programmer* dari Australia. Merupakan aplikasi *text editor* untuk menulis kode. Sejumlah bahasa program yang ada pada aplikasi ini. Diantaranya PHP, CSS, C, C++, HTML, ASP, Java, dan sebagainya. Tentu saja, *software* ini bisa lebih memudahkan pekerjaan pengguna saat membuat sebuah program (Priyanto, 2014).

2.11 Arduino

Menurut Sulaiman (2012:1), Arduino merupakan *platform* yang terdiri dari *software* dan *hardware*. *Hardware* Arduino sama dengan *microcontroller* pada umumnya hanya pada Arduino ditambahkan penamaan pin agar mudah diingat. *Software* Arduino merupakan *software open source* sehingga dapat di download secara gratis. *Software* ini digunakan untuk membuat dan memasukkan program ke dalam Arduino.

Menurut Sulaiman (2012:1) Arduino adalah *kit* elektronik atau papan rangkaian elektronik *open source* yang di dalamnya terdapat komponen utama yaitu sebuah *chip* mikrokontroler dengan jenis AVR dari perusahaan Atmel.

Berdasarkan dua definisi yang dikemukakan diatas dapat disimpulkan bahwa Arduino merupakan *kit* elektronik atau papan rangkaian elektronik yang didalamnya terdapat komponen utama yaitu sebuah *chip* mikrokontroler dengan jenis AVR dari perusahaan Atmel serta *software* pemrograman yang berlisensi *open source*.

2.12 Pengertian Aplikasi

Menurut Sri Widiyanti (2000:142) dalam bukunya dengan judul Pengantar Basis Data, menjelaskan tentang aplikasi dan DBMS, beliau menerangkan bahwa: tidak sedikit orang berpendapat bahwa perangkat lunak semacam Foxpro dan MS Access tidak dikategorikan sebagai DBMS, yang berpendapat seperti ini pasti orang beranggapan bahwa sebuah DBMS haruslah sebuah sistem (perangkat lunak) yang berdiri sendiri dan terpisah dengan aplikasi basis data. Basis data dapat dianggap sebagai objek yang hidup, karena sesungguhnya aplikasi tidak pernah berinteraksi langsung sebagai basis data, tetapi selalu melalui DBMS sebagai perantara, bahkan DBMS sendiri bisa melakukan aktifitas sendiri yang bisa ditangkap oleh aplikasi. Jadi interaksi aplikasi, basis data dan DBMS merupakan interaksi dua arah. Hal ini dapat disimpulkan bahwa aplikasi adalah sebuah perangkat lunak yang menjadi bagian akhir dalam sebuah sistem yang digunakan untuk mengolah data menjadi suatu informasi yang berguna bagi pengguna dan sistem yang bersangkutan.

2.13 Pengertian Bagan Alir (*Flowchart*)

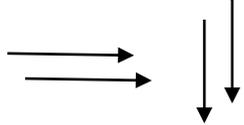
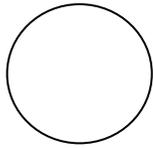
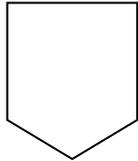
Menurut Hartono (2005 : 795), *flowchart* atau diagram alir merupakan bagan(*chart*) yang menunjukkan hasil (*flow*) dalam program atau prosedur sistem secara logika. Digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. Diagram ini bisa memberi solusi selangkah demi selangkah untuk penyelesaian masalah yang ada di dalam proses atau algoritma tersebut.

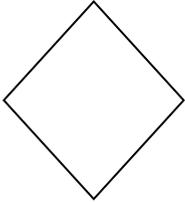
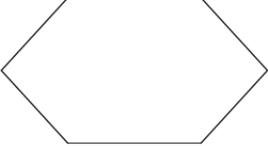
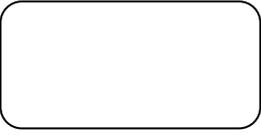
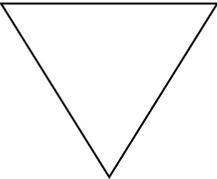
Simbol-simbol yang di pakai dalam *flowchart* dibagi menjadi 3 kelompok:

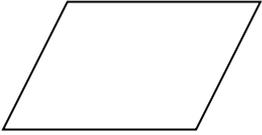
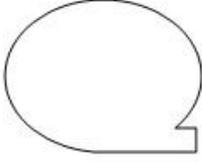
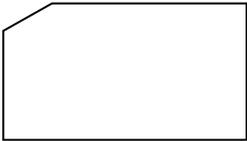
- 1) *Flow direction symbols*. Digunakan untuk menghubungkan simbol satu dengan yang lain, Disebut juga *connecting line*.
- 2) *Processing symbols*. Menunjukkan jenis operasi pengolahan dalam suatu proses/prosedur.
- 3) *Input / Output symbols*. Menampilkan jenis peralatan yang digunakan sebagai media *input* atau *output*

Berikut dibawah ini adalah simbol-simbol yang digunakan dalam *flowchart* disertai dengan keterangan fungsinya :

Table 2.1 Tabel *Flow Symbols*.

SIMBOL	KETERANGAN
	Simbol arus / <i>flow</i> , menyatakan jalannya arus suatu proses.
	Simbol <i>connector</i> , menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.
	Simbol <i>offline connector</i> , menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.

	<p>Simbol proses, yaitu menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.</p>
	<p>Simbol <i>manual</i>, menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer.</p>
	<p>Simbol <i>decision</i>, yaitu menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya / tidak</p>
	<p>Simbol <i>terminal</i>, menyatakan permulaan atau akhir suatu program.</p>
	<p>Simbol <i>predefined process</i>, menyatakan persediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.</p>
	<p>Simbol <i>keying operation</i>, menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai <i>keyboard</i>.</p>
	<p>Simbol <i>offline-storage</i>, menunjukkan bahwa data dalam simbol ini akan disimpan ke dalam suatu media tertentu.</p>
	<p>Simbol <i>manual input</i>, menyatakan data secara manual dengan menggunakan <i>online keyboard</i>.</p>

	<p>Simbol <i>input / output</i>, menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis peralatannya.</p>
	<p>Simbol <i>magnetic tape</i>, menyatakan input berasal dari pita magnetis atau output tersimpan ke dalam pita magnetis.</p>
	<p>Simbol <i>disk storage</i>, menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> tersimpan ke dalam <i>disk</i>.</p>
	<p>Simbol <i>document</i>, mencetak keluaran dalam bentuk dokumen (memulai <i>printer</i>).</p>
	<p>Simbol <i>punched card</i>, menyatakan input berasal dari kartu atau output ditulis ke kartu.</p>

(Sumber : Indrajani, 2015)

2.14 Pengertian *Data Flow Diagram* (DFD)

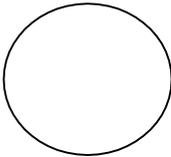
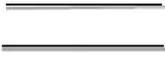
Data Flow Diagram merupakan suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut (Kristanto, 2008).

Data Flow Diagram atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengatur dari

masukan (input) dan keluaran (output). DFD tidak sesuai untuk memodelkan sistem yang menggunakan pemrograman berorientasi objek (Sukamto, 2014).

Notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco) adalah sebagai berikut:

Tabel 2.2 Simbol-simbol Data *Flow* Diagram (DFD)

NOTASI	KETERANGAN
	<p>Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program.</p>
	<p><i>File</i> atau basis data atau penyimpanan (<i>storage</i>) pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel basis data (<i>Entity Relationship Diagram (ERD)</i>, <i>Conceptual Data Model (CMD)</i>, <i>Physical Data Model (PDM)</i>).</p>
	<p>Entitas luar (<i>external entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang memakai atau berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan.</p>
	<p>Aliran data merupakan data yang di kirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>).</p>

(Sumber :Sukamto, 2014).

Sukamto (2014:72) Berikut ini adalah tahapan-tahapan perancangan dengan menggunakan Data Flow Diagram (DFD) :

1. Data Flow Diagram (DFD) Level 0

DFD *Level 0* menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DFD *Level 0* digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.

2. Data Flow Diagram (DFD) Level 1

DFD *Level 1* digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan. DFD *Level 1* merupakan hasil *breakdown* DFD *Level 0* yang sebelumnya sudah dibuat.

3. Data Flow Diagram (DFD) Level 2

Modul-modul pada DFD *Level 1* dapat di *breakdown* menjadi DFD *Level 2*. Modul mana saja yang harus di *breakdown* lebih detail tergantung pada tingkat kedetilan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu untuk di *breakdown* lagi. Untuk sebuah sistem, jumlah DFD *Level 2* sama dengan jumlah modul pada DFD *Level 1* yang di *breakdown*.

4. Data Flow Diagram (DFD) Level 3 dan seterusnya.

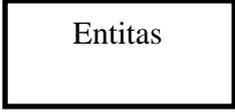
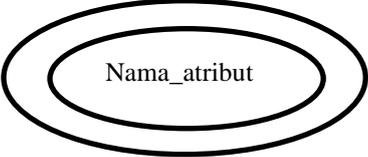
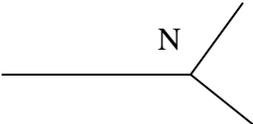
DFD *Level 3*, 4, 5 dan seterusnya merupakan *breakdown* dari modul pada DFD *Level* di atasnya. *Breakdown* pada *level 3*, 4 dan 5 dan seterusnya aturannya sama persis dengan DFD *Level 1* atau *Level 2*.

2.15 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah pemodelan awal basis data yang akan dikembangkan berdasarkan teori himpunan dalam bidang matematika untuk pemodelan basis data relasional. ERD memiliki beberapa aliran notasi seperti notasi *Chen* (dikembangkan oleh Peter Chen). Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi *Crow's Foot*, dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi *Chen* (Sukamto, 2014).

Berikut adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen :

Tabel 2.3 Simbol-simbol *Entity Relationship Diagram* (ERD)

SIMBOL	DESKRIPSI
	<p>Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.</p>
	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.</p>
	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).</p>
<p>Atribut multivalui/<i>multivalue</i></p> 	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.</p>
	<p>Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja.</p>
<p>Asosiasi/<i>association</i></p> 	<p>Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian, keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas A dan entitas B.</p>

(Sumber :Sukamto, 2014).