BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Teknologi Informasi

Warsita, Bambang (2008:135) teknologi informasi adalah sarana dan prasarana (hardware, software, useware) sistem dan metode untuk memperoleh, mengirimkan, mengolah, menafsirkan, menyimpan, mengorganisasikan, dan menggunakan data secara bermakna.

2.1.2 Pengertian Internet

Kamus Besar Bahasa Indonesia Pusat Bahasa (2013:543), internet adalah jaringan komunikasi elektronik yang menghubungkan jaringan komputer dan fasilitas komputer yang terorganisasi di seluruh dunia melalui telepon atau satelit.

Internet (*interconnected network*) adalah jaringan komputer yang menghubungkan antar jaringan secara global, internet dapat juga disebut jaringan dalam suatu jaringan yang luas (Sibero, 2014: 10).

2.1.3 Pengertian *World Wide Web*

Kadir, Abdul (2003: 460), *world wide web* atau *web* merupakan sumber daya internet yang sangat populer dan dapat digunakan untuk memperoleh informasi atau bahkan melakukan transaksi pembelian barang. Web menggunakan protokol yang disebut HTTP (*Hypertext Transfer Protocol*) yang berjalan pada TCP/IP (*Transfer Control Protocol/Internet Protocol*).

2.2 Teori Judul

2.2.1 Aplikasi

Menurut Kamus Kamus Besar Bahasa Indonesia (1998 : 52), “Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atauketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna”.

Anisyah (2000:30), "Aplikasi adalah penerapan, penggunaan atau penambahan. Dari pengertian diatas, dapat disimpulkan bahwa aplikasi adalah *software* yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data."

2.2.2 Absensi

Menurut Erna Simonna (2009), "Absensi adalah suatu pendataan kehadiran, bagian dari pelaporan aktifitas suatu institusi, atau komponen institusi itu sendiri yang berisi data-data kehadiran yang disusun dan diatur sedemikian rupa sehingga mudah untuk dicari dan dipergunakan apabila sewaktu-waktu diperlukan oleh pihak yang berkepentingan. Yang membedakan jenis-jenis absensi dapat dikelompokkan menjadi dua, yaitu:

1. Absensi manual, adalah cara pengentrian kehadiran dengan cara menggunakan pena (tanda tangan).
2. Absensi non manual (dengan menggunakan alat), adalah suatu cara pengentrian kehadiran dengan menggunakan sistem terkomputerisasi, bisa menggunakan kartu dengan *barcode*, *finger print* ataupun dengan mengentrikan nip dan sebagainya".

2.2.3 *IP Address*

2.2.3.1 Pengertian *IP Address*

Menurut Wikipedia, "Alamat *IP* (*Internet Protocol Address* atau sering disingkat *IP*) adalah deretan angka biner antar 32-bit sampai 128-bit yang dipakai sebagai alamat identifikasi untuk tiap komputer host dalam jaringan Internet. Panjang dari angka ini adalah 32-bit (untuk IPv4 atau *IP* versi 4), dan 128-bit (untuk IPv6 atau *IP* versi 6) yang menunjukkan alamat dari komputer tersebut pada jaringan Internet berbasis *TCP/IP*. Pada pembahasan selanjutnya (dalam artikel ini) yang disebut sebagai *IP Address* adalah *IP Address* versi 4. *IP Address* sebetulnya terdiri dari dua bagian yaitu bagian *Network Identifier* (*NetID*) yang berperan dalam identifikasi suatu network dari network yang lain dan bagian *Host Identifier* (*HostID*) yang menentukan alamat *host* atau komputer dalam suatu *network*. Jadi seluruh host yang tersambung dalam jaringan yang sama akan memiliki bit *network* (*NetID*) yang sama. Analoginya adalah seperti alamat rumah yang terdiri dari nama jalan dan nomor rumah". IP address merupakan bilangan

biner 32 bit yang terbagi menjadi empat kelompok, sehingga masing-masing kelompok terdiri dari bilangan biner 8 bit. Ini merupakan implementasi alamat IP yang disebut IPv4 (Wagito, 2005).

2.2.3.2 Jenis-jenis alamat IP

1. Alamat IP Versi 4

Alamat IP versi 4 umumnya diekspresikan dalam notasi desimal bertitik (dotted-decimal notation), yang dibagi ke dalam empat buah oktet berukuran 8 bit. Dalam beberapa buku referensi, format bentuknya adalah w.x.y.z. Karena setiap oktet berukuran 8 bit, maka nilainya berkisar antara 0 hingga 255 (meskipun begitu, terdapat beberapa pengecualian nilai).

Alamat IP yang dimiliki oleh sebuah host dapat dibagi dengan menggunakan *subnet mask* jaringan ke dalam dua buah bagian, yakni:

- *Network Identifier*/NetID atau *Network Address* (alamat jaringan) yang digunakan khusus untuk mengidentifikasikan alamat jaringan di mana host berada.

  Semua sistem di dalam sebuah jaringan fisik yang sama harus memiliki alamat *network identifier* yang sama. *Network identifier* juga harus bersifat unik dalam sebuah *Internetwork*. Alamat *network identifier* tidak boleh bernilai 0 atau 255.
- *Host Identifier*/HostID atau *Host address* (alamat host) yang digunakan khusus untuk mengidentifikasikan alamat host di dalam jaringan. Nilai *host identifier* tidak boleh bernilai 0 atau 255 dan harus bersifat unik di dalam *network identifier* di mana ia berada.

a. Alamat unicast ip v4

Dalam RFC 791, alamat Unicast IP versi 4 dibagi ke dalam beberapa kelas, dilihat dari oktet pertamanya, seperti terlihat pada tabel. Sebenarnya yang menjadi pembeda kelas IP versi 4 adalah pola biner yang terdapat dalam oktet pertama (utamanya adalah bit bit awal/*high-order bit*), tapi untuk lebih mudah mengingatnya, akan lebih cepat diingat dengan menggunakan representasi desimal.

Kelas Alamat IP	Oktet pertama (desimal)	Oktet pertama (biner)	Digunakan oleh
Kelas A	1-126	0xxx xxxx	Alamat unicast untuk skala besar
Kelas B	128-191	1xxx xxxx	Alamat unicast untuk jaringan skala menengah hingga skala besar
Kelas C	192-223	110x xxxx	Alamat unicast untuk skala kecil
Kelas D	224-239	1110 xxxx	Alamat multicast (bukan alamat unicast)
Kelas E	240-255	1111 xxxx	Direservasikan umumnya digunakan sebagai alamat percobaan bukan alamat unicast

Kelas A

Alamat-alamat unicast kelas A diberikan untuk jaringan skala besar. Nomor urut bit tertinggi di dalam alamat IP kelas A selalu diset dengan nilai **0** (nol). Tujuh bit berikutnya—untuk melengkapi oktet pertama—akan membuat sebuah *network identifier*. 24 bit sisanya (atau tiga oktet terakhir) merepresentasikan *host identifier*. Ini mengizinkan kelas A memiliki hingga 126 jaringan, dan 16,777,214 host tiap jaringannya. Alamat dengan oktet awal 127 tidak diizinkan, karena digunakan untuk mekanisme Interprocess Communication (IPC) di dalam mesin yang bersangkutan.

Kelas B

Alamat-alamat unicast kelas B dikhususkan untuk jaringan skala menengah hingga skala besar. Dua bit pertama di dalam oktet pertama alamat IP kelas B selalu diset ke bilangan biner **10**. 14 bit berikutnya (untuk melengkapi dua oktet pertama), akan membuat sebuah *network identifier*. 16 bit sisanya (dua oktet terakhir) merepresentasikan *host identifier*. Kelas B dapat memiliki 16,384 network, dan 65,534 host untuk setiap network-nya.

Kelas C

Alamat IP unicast kelas C digunakan untuk jaringan berskala kecil. Tiga bit pertama di dalam oktet pertama alamat kelas C selalu diset ke nilai biner **110**. 21 bit selanjutnya (untuk melengkapi tiga oktet pertama) akan membentuk sebuah *network identifier*. 8 bit sisanya (sebagai oktet terakhir) akan merepresentasikan *host identifier*. Ini memungkinkan pembuatan total 2,097,152 buah network, dan 254 host untuk setiap network-nya.

Kelas D

Alamat IP kelas D disediakan hanya untuk alamat-alamat *IP multicast*, sehingga berbeda dengan tiga kelas di atas. Empat *bit* pertama di dalam IP kelas D selalu diset ke bilangan biner **1110**. 28 *bit* sisanya digunakan sebagai alamat yang dapat digunakan untuk mengenali host. Untuk lebih jelas mengenal alamat ini, lihat pada bagian Alamat *Multicast* IPv4.

Kelas E

Alamat IP kelas E disediakan sebagai alamat yang bersifat "eksperimental" atau percobaan dan dicadangkan untuk digunakan pada masa depan. Empat bit pertama selalu diset kepada bilangan biner **1111**. 28 bit sisanya digunakan sebagai alamat yang dapat digunakan untuk mengenali host.

b. Alamat Multicast IP versi 4

Alamat IP Multicast (*multicast IP address*) adalah alamat yang digunakan untuk menyampaikan satu paket kepada banyak penerima. Dalam sebuah intranet yang memiliki alamat *multicast* IPv4, sebuah paket yang ditujukan ke sebuah alamat *multicast* akan diteruskan oleh router ke subjaringan di mana terdapat host-host yang sedang berada dalam kondisi "*listening*" terhadap lalu lintas jaringan yang dikirimkan ke alamat multicast tersebut. Dengan cara ini, alamat multicast pun menjadi cara yang efisien untuk mengirimkan paket data dari satu sumber ke beberapa tujuan untuk beberapa jenis komunikasi. Alamat multicast didefinisikan dalam RFC 1112.

Alamat-alamat multicast IPv4 didefinisikan dalam ruang alamat **kelas D**, yakni **224.0.0.0/4**, yang berkisar dari 224.0.0.0 hingga 239.255.255.255. Prefiks alamat 224.0.0.0/24 (dari alamat 224.0.0.0 hingga 224.0.0.255) tidak dapat digunakan karena dicadangkan untuk digunakan oleh lalu lintas multicast dalam subnet lokal.

c. Alamat Broadcast IP versi 4

Alamat *broadcast* IP versi 4 digunakan untuk menyampaikan paket-paket data "satu-untuk-semua". Jika sebuah *host* pengirim yang hendak mengirimkan paket data dengan tujuan alamat *broadcast*, maka semua *node* yang terdapat di dalam segmen jaringan tersebut akan menerima paket tersebut dan memprosesnya. Berbeda dengan alamat *IP unicast* atau alamat *IP multicast*, alamat *IP broadcast* hanya dapat digunakan sebagai alamat tujuan saja, sehingga tidak dapat digunakan sebagai alamat sumber.

Ada empat buah jenis alamat IP broadcast, yakni *network broadcast*, *subnet broadcast*, *all-subnets-directed broadcast*, dan *Limited Broadcast*. Untuk setiap jenis alamat *broadcast* tersebut, paket IP *broadcast* akan dialamatkan kepada lapisan antarmuka jaringan dengan menggunakan alamat *broadcast* yang dimiliki oleh teknologi antarmuka jaringan yang digunakan. Sebagai contoh, untuk jaringan *Ethernet* dan *Token Ring*, semua paket *broadcast* IP akan dikirimkan ke alamat *broadcast Ethernet* dan *Token Ring*, yakni 0xFF-FF-FF-FF-FF-FF.

2. Alamat IP Versi 6

Berbeda dengan IPv4 yang hanya memiliki panjang 32 bit (jumlah total alamat yang dapat dicapainya mencapai 4,294,967,296 alamat), IPv6 memiliki panjang 128 bit. IPv4, meskipun total alamatnya mencapai 4 miliar, pada kenyataannya tidak sampai 4 miliar alamat, karena ada beberapa limitasi, sehingga implementasinya saat ini hanya mencapai beberapa ratus juta saja. IPv6, yang memiliki panjang 128 bit, memiliki total alamat yang mungkin hingga 2^{128}=3,4 x 10^{38} alamat. Total alamat yang sangat besar ini bertujuan untuk

menyediakan ruang alamat yang tidak akan habis (hingga beberapa masa ke depan), dan membentuk infrastruktur routing yang disusun secara hierarkis, sehingga mengurangi kompleksitas proses routing dan tabel routing.

Sama seperti halnya IPv4, IPv6 juga mengizinkan adanya DHCP Server sebagai pengatur alamat otomatis. Jika dalam IPv4 terdapat *dynamic address* dan *static address*, maka dalam IPv6, konfigurasi alamat dengan menggunakan DHCP Server dinamakan dengan *stateful address configuration*, sementara jika konfigurasi alamat IPv6 tanpa DHCP Server dinamakan dengan *stateless address configuration*.

Seperti halnya IPv4 yang menggunakan bit bit pada tingkat tinggi (high-order bit) sebagai alamat jaringan sementara bit bit pada tingkat rendah (low-order bit) sebagai alamat *host*, dalam IPv6 juga terjadi hal serupa. Dalam IPv6, bit bit pada tingkat tinggi akan digunakan sebagai tanda pengenal jenis alamat IPv6, yang disebut dengan ***Format Prefix (FP)***. Dalam IPv6, tidak ada subnet mask, yang ada hanyalah *Format Prefix*.

Jenis-jenis Alamat IP versi 6

IPv6 mendukung beberapa jenis format prefix, yakni sebagai berikut:

- Alamat Unicast, yang menyediakan komunikasi secara *point-to-point*, secara langsung antara dua *host* dalam sebuah jaringan.
- Alamat Multicast, yang menyediakan metode untuk mengirimkan sebuah paket data ke banyak *host* yang berada dalam *group* yang sama. Alamat ini digunakan dalam komunikasi *one-to-many*.
- Alamat Anycast, yang menyediakan metode penyampaian paket data kepada anggota terdekat dari sebuah group. Alamat ini digunakan dalam komunikasi *one-to-one-of-many*. Alamat ini juga digunakan hanya sebagai alamat tujuan (*destination address*) dan diberikan hanya kepada router, bukan kepada *host-host* biasa.

Jika dilihat dari cakupan alamatnya, alamat unicast dan anycast terbagi menjadi alamat-alamat berikut:

- Link-Local, merupakan sebuah jenis alamat yang mengizinkan sebuah komputer agar dapat berkomunikasi dengan komputer lainnya dalam satu subnet.
- Site-Local, merupakan sebuah jenis alamat yang mengizinkan sebuah komputer agar dapat berkomunikasi dengan komputer lainnya dalam sebuah intranet.
- Global Address, merupakan sebuah jenis alamat yang mengizinkan sebuah komputer agar dapat berkomunikasi dengan komputer lainnya dalam Internet IPv6.

Sementara itu, cakupan alamat multicast dimasukkan ke dalam struktur alamat.

Unicast Address

Alamat unicast IPv6 dapat diimplementasikan dalam berbagai jenis alamat, yakni:

- Alamat unicast global
- Alamat unicast site-local
- Alamat unicast link-local
- Alamat unicast yang belum ditentukan (unicast unspecified address)
- Alamat unicast loopback
- Alamat Unicast 6to4
- Alamat Unicast ISATAP

2.2.3.3 Perbandingan Alamat IP versi 4 dan Alamat IP versi 6

Kriteria	Alamat IP versi 4	Alamat IP versi 6
Panjang alamat	32 bit	128 bit
Jumlah total host (teoretis)	2^{32}=±4 miliar host	2^{128}
Menggunakan kelas alamat	Ya, kelas A, B, C, D, dan E. Belakangan tidak digunakan lagi, mengingat telah tidak relevan dengan perkembangan jaringan Internet yang pesat.	Tidak
Alamat multicast	Kelas D, yaitu **224.0.0.0/4**	Alamat *multicast IPv6*, yaitu **FF00:/8**
Alamat *broadcast*	Ada	Tidak ada
Alamat yang belum ditentukan	**0.0.0.0**	::
Alamat *loopback*	**127.0.0.1**	**::1**
Alamat IP publik	Alamat IP publik IPv4, yang ditetapkan oleh otoritas Internet (IANA)	Alamat IPv6 *unicast global*
Alamat IP pribadi	Alamat IP pribadi IPv4, yang ditetapkan oleh otoritas Internet	Alamat IPv6 *unicast site-local* (FEC0::/48)
Konfigurasi alamat otomatis	Ya (APIPA)	Alamat IPv6 *unicast link-local* (FE80::/64)
Representasi tekstual	*Dotted decimal format notation*	*Colon hexadecimal format notation*
Fungsi Prefiks	*Subnet mask* atau panjang prefiks	Panjang prefiks
Resolusi alamat DNS	*A Resource Record (Single A)*	*AAAA Resource Record (Quad A)*

(source: Wikipedia)

2.2 Teori Khusus

2.3.1 *Data Flow Diagram* (DFD)

Menurut Sutabri (2005:163), *Data Flow Diagram* adalah suatu *network* yang menggambarkan suatu sistem automat/komputerisasi, manualisasi atau gabungan dari keduanya, yang penggambarannya disusun dalam bentuk kumpulan komponen sistem yang saling berhubungan sesuai dengan aturan mainnya.

Menurut Kristanto (2008:61), *Data Flow Diagram* adalah 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 menghasilakan data tersebut dan interaksi data yang tersimpan dan proses yang dikenakan pada data tersebut.

Data Flow Diagram merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur (*structured analysis and design*). Adapun ciri-ciri *Data Flow Diagram* :

1. Menggambarkan arus data di dalam sistem dengan terstruktur dan jelas.

2. *Data Flow Diagram* sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem yang baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir, atau lingkungan fisik dimana data tersebut akan disimpan.

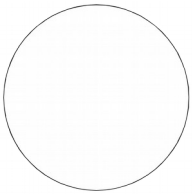
Langkah–langkah di dalam membuat *data flow diagram* dibagi menjadi 3 (tiga) tahap atau tingkatan konstruksi, yaitu sebagai berikut:

1. Diagram Konteks
   Diagram ini dibuat untuk menggambarkan sumber serta tujuan data yang akan diproses atau dengan kata lain diagram tersebut digunakan untuk menggambarkan sistem secara umum/global dari keseluruhan sistem yang ada.
2. Diagram Nol
   Diagram ini dibuat untuk menggambarkan tahapan proses yang ada di dalam diagram konteks, yang penjabarannya lebih terperinci.
3. Diagram Detail / Rinci
   Diagram ini dibuat untuk menggambarkan arus data secara lebih mendetail lagi dari tahapan proses yang ada di dalam diagram nol.

Simbol yang digunakan dalam *Data Flow Diagram (DFD)* menurut teknik Yourdon/De Marco dapat dilihat pada tabel 2.1.

Tabel 2.1 Simbol-simbol di dalam *Data Flow Diagram (DFD)*

No	Simbol	Keterangan

1		***External Entity* (Kesatuan Luar) atau *Boundry* (Batas Sistem)** Yaitu kesatuan (*entity*) di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan *input* atau menerima *output* dari sistem.
2		***Process* (Proses)** Yaitu kegiatan atau kerja yang dilakukan oleh orang, mesin, atau komputer dari hasil suatu arus data yang masuk kedalam proses untuk dihasilakn arus data yang akan keluar dari proses.
3		***Data Flow* (Arus Data)** Arus data di simbolkan dengan panah. Arus data ini mengalir diantara proses (*process*), simpanan data (*data store*), dan kesatuan luar (*external entity*). Arus data ini menunjukan arus data yang dapat berupa masukkan untuk sistem atau hasil dari proses sistem.
4		***Data Store* (Simpanan Data)** Merupakan simpanan dari data yang dapat berupa : o Suatu file atau database di sistem komputer. o Suatu arsip atau catatan manual. o Suatu kotak tempat data di meja seseorang. o Suatu tabel acuan manual. o Suatu agenda atau buku.

Sumber : Kristanto (2004 : 58)

2.3.2 *Entity Relationship Diagram* (ERD)

Menurut Fathansyah (2007 : 79) *Model Entity Relationship* (*ERD*) adalah suatu diagram yang berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang

mempresentasikan seluruh fakta dari 'dunia nyata' yang kita tinjau, dan dapat digambarkan dengan lebih sistematis dengan menggunakan *Diagram Entity-Relationship* (Diagram E-R).

Notasi-notasi simbolik didalam Diagram E-R yang dapat digunakan adalah, Fathansyah (2007 : 80) :

1. Persegi panjang, menyatakan Himpunana Entitas.
2. Lingkaran/*Elip*, menyatakan Atribut (Atribut yang berfungsi sebagai *key* digarisbawahi).
3. Belah Ketupat, menyatakan Himpunan Relasi.
4. Garis, 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 a untuk relasi satu ke satu, dan N untuk relasi satu-ke-banyak atau N dan N untuk relasi banyak-ke-banyak).

Tabel 2.2 Notasi-notasi simbolik didalam Diagram E-R

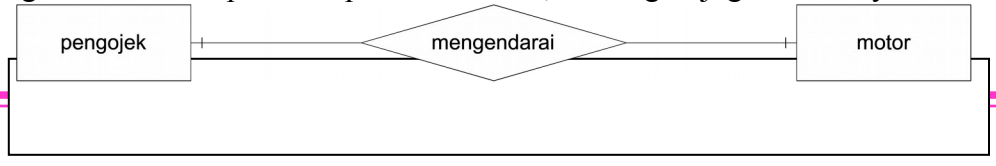
No.	Simbol	Keterangan
1.	E	Himpunan Entitas E
2.	R	Himpunan Relasi R
3.		Link
4.	a	Atribut a sebagai *key*

Sumber : Fathansyah (2007 : 80)

Menurut Fathansyah (2007 : 77) Kardinalitas relasi menunjukan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Kardinalitas Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dapat berupa :

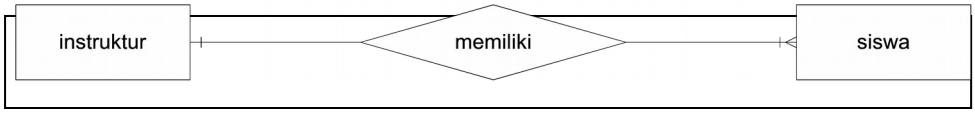
1. Satu ke Satu (*One to One* / 1-1)

   Yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B, dan begitu juga sebaliknya.



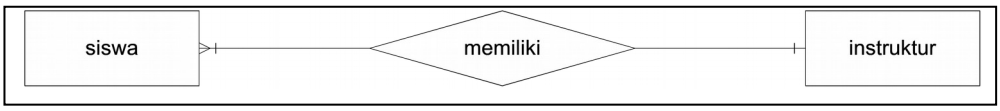
Gambar 2.1 *Relationship One to One*
Sumber : Fathansyah (2007 : 77)

2. Satu ke Banyak (*One to Many*)
Yaitu setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya.



Gambar 2.2 *Relationship One to Many*
Sumber : Fathansyah (2007 : 77)

3. Banyak ke Satu (*Many to Many*)
Yaitu setiap entitas pada himpunan entitas A dapat berhubungan banyak dengan banyak pada himpunan entitas B, dan sebaliknya.



Gambar 2.3 *Relationship Many to Many*
Sumber : Fathansyah (2007 : 77)

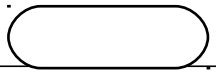
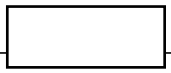
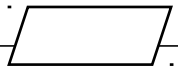
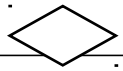
Menurut Fathansyah (2007 : 84) Tahapan Pembuatan Diagram E-R adalah:

1. Mengidentifikasi dan menetapkan seluruh himpunan entitas yang akan terlibat.
2. Menentukan atribut-atribut *key* (kunci) dari masing-masing himpunan entitas.
3. Mengidentifikasi dan menetapkan seluruh himpunan relasi diantara himpunan entitas-himpunan entitas yang berserta *foreign-key*nya (kunci asing/ kunci tamu).
4. Menentukan derajat/kardinalitas relasi untuk setiap himpunan relasi. Melengkapi himpunan entitas dan himpunan relasi dengan atribut deskritif (atribut yang bukan kunci).

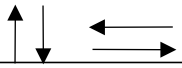
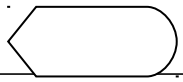
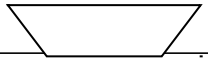
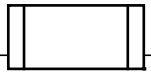
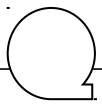
2.3.3 *Flowchart*

Flowchart adalah symbol-simbol pekerjaan yang menunjukkan bagian aliran proses yang saling terhubung. Jadi, setiap simbol *flowchart* melambangkan pekerjaan dan intruksinya. Adapun simbol-simbol *Flowchart* adalah sebagai berikut:

Tabel 2.3. *Simbol-Simbol* Flowchart

No.	Simbol	Arti
1.		Simbol **Start** atau **End** yang mendefinisikan awal atau akhir dari sebuah *flowchart*
2.		Simbol pemrosesan yang terjadi pada sebuah alur kerja
3.		Simbol ***Input/Output*** yang mendefinisikan masukan dan keluaran proses
4.		Simbol untuk memutuskan proses lanjutan dari kondisi tertentu
5.		Simbol konektor untuk menyambung proses pada lembar kerja yang sama
6.		Simbol konektor untuk menyambung proses pada lembar kerja yang berbeda

Lanjutan ***Tabel 2.3.*** *Simbol-Simbol* Flowchart

7.		Simbol untuk menghubungkan antar proses atau antar symbol
8.		Simbol yang menyatakan piranti keluaran, seperti layar monitor, *printer*, dll
9.		Simbol yang mendefinisikan proses yang dilakukan secara manual
10.		Simbol masukan atau keluaran dari atau ke sebuah dokumen
11.		Simbol yang menyatakan bagian dari program (subprogram)
12.		Simbol masukan atau keluara dari atau ke sebuah pita magnetic

13.		Simbol database atau basis data

Sumber: Ewolf Community (2012:17)

2.4 Teori Program

2.4.1. *Hypertext Preprocessor* (PHP)

2.4.1.1 Pengertian *Hypertext Preprocessor* (PHP)

Kadir, Abdul (2008:2), *Hypertext Preprocessor* (PHP) merupakan Bahasa berbentuk skrip yang ditempatkan dalam server dan diproses diserver. Hasilnya yang akan dikirim ke klien, tempat pemakai menggunakan *browser*.

Madcoms (2010:341), *PHP* adalah bahasa pemrograman yang bekerja dalam sebuah webserver, dimana script *PHP* dibuat harus tersimpan dalam sebuah server dan dieksekusi atau diproses dalam server tersebut.

2.4.1.2 *Script Hypertext Preprocessor* (PHP)

Kadir, Abdul (2008:3), skrip *Hypertext Preprocessor* berkedudukan sebagai tag dalam Bahasa HTML (*Hypertext Markup Language*) adalah bahasa standar untuk membuat halaman-halaman web. Adapun kode berikut contoh kode PHP (*Hypertext Preprocessor*) yang berada dalam kode HTML.

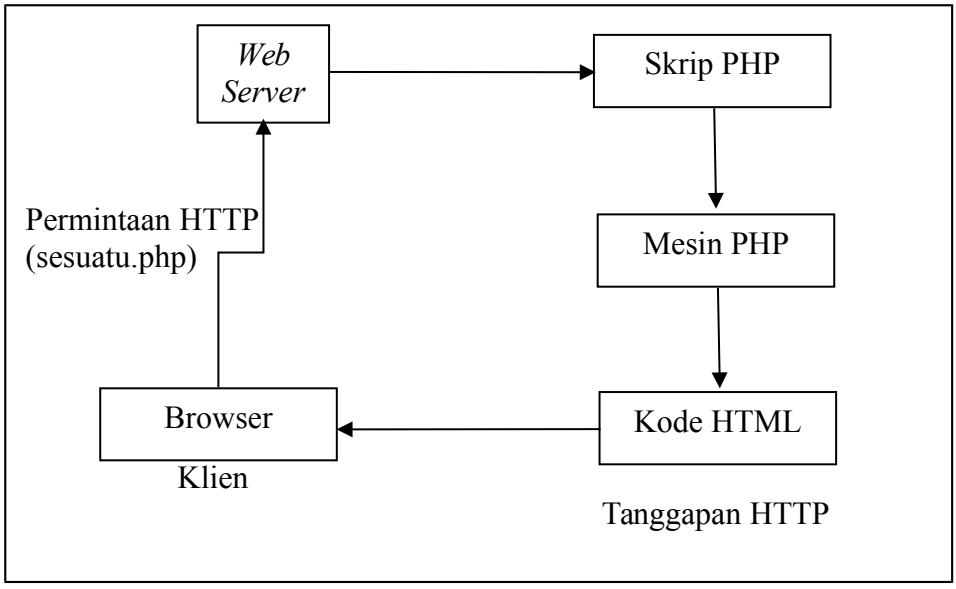
```
<HTML>
<HEAD>
<TITLE>Latihan Pertama</TITLE>
</HEAD>
<BODY>
Selamat belajar PHP</BR>
<?php
printf("Tgl. Sekarang: %s", Date ("d F Y");
?>
</BODY>
</HTML>
```

Kode *Hypertext Preprocessor* (PHP) diawali dengan <? PHP dan diakhiri dengan ?>. Pasangan kedua kode inilah yang berfungsi sebagai tag kode PHP.

Berdasarkan tag inilah, pihak server dapat memahami kode PHP dan kemudian memprosesnya.

2.4.1.3 Konsep Kerja PHP

Kadir, Abdul (2008:4), model kerja HTML (*Hypertext Markup Language*) diawali dengan permintaan suatu halaman web oleh *browser*. Berdasarkan URL (*Uniform Resources Localator*) atau dikenal dengan sebutan alamat internet, *browser* merupakan alamat dari web server, mengindentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *web server*. Selanjutnya, *web server* akan mencarikan file yang diminta dan memberikan isinya ke *web browser*. *Browser* yang mendapatkan isinya segera melakukan proses penerjemahan kode HTML (*Hypertext Markup Language*) dan menampilkan ke layar pemakai. Sedangkan model kerja PHP (*Hypertext Preprocessor*) pada prinsipnya serupa dengan kode HTML (*Hypertext Markup Language*). Hanya saja, ketika berkas PHP (*Hypertext Preprocessor*) yang diminta didapatkan oleh web server, isinya segera dikirimkan ke mesin PHP (*Hypertext Preprocessor*) dan mesin inilah yang memproses dan memberikan hasilnya ke *web server*. Selanjutnya, *web server* menyampaikan ke klien.



Gambar 2.7 Skema PHP (*Hypertext Preprocessor*)

Sumber: Kadir, Abdul. (2008:6)

2.4.2. MySQL

2.4.2.1 Pengertian MySQL

Kadir, Abdul (2008:2), MySQL merupakan *software* yang tergolong sebagai DBMS (*Database Mnagement System*) yang bersifat *Open Source*.

Yakub (2012:51), basis data (*database*) merupakan kumpulan data yang saling berhubungan (punya relasi).

2.4.2.2 Fitur MySQL

Sebagai *software DBMS*, *MySQL* memiliki sejumlah fitur seperti yang dijelaskan di bawah ini:

a. *Multiplatform*
   MySQL tersedia pada beberapa *platform* (*Windows, Linux, Unix*, dan lain-lain).
b. Andal, cepat, dan mudah digunakan
   MySQL tergolong sebagai *database server* (*server* yang melayani permintaan terhadap *database*) yang andal, dpaat menangani *database* yang besar dengan kecepatan tinggi, mendukung banyak sekali fungsi untuk mengaskses *database*, dan sekaligus mudah untuk digunakan.
c. Jaminan keamanan akses
   MySQL mendukung pengamanan *database* dengan berbagai kriteria penaksesan. Sebagai gambaran, dimungkinkan untuk mengatur *user* tertentu agar bisa mengakses data yang bersifat rahasia (misalnya gaji pegawai), sedangkan *user* lain tidak boleh.
d. Dukungan SQL
   Seperti tersirat dalam namanya, MySQL mendukung perintah *SQL* (*Structured Query Language*). Sebagaimana diketahui, SQL merupakan standar pengaksesan *database* relasional. Pengetahuan akan SQL akan memudahkan siapa pun untuk menggunakan MySQL.

2.4.3. Basis Data (*Database*)

Madcoms (2010:367), basis data (*database*) berfungsi sebagai penampung data yang di*input*kan melalui *form website*.

Yakub (2012:51), basis data (*database*) merupakan kumpulan data yang saling berhubungan (punya relasi).

2.5 Referensi Penelitian Sebelumnya

Agus Irawan dan Tajudin Noor (2014) "Dalam jurnal yang berjudul sistem informasi absensi mahasiswa (studi kasus jurusan Administrasi Bisnis POLIBAN), Jurusan Administrasi Bisnis Politeknik Negeri Banjarmasin adalah lembaga pendidikan tinggi. Banyak aktivitas pengolahan data masih dilakukan dengan manual, salah satunya pengelolaan administrasi absensi. Kegiatan yang sangat menyita waktu adalah membuat rekapitulasi kehadiran berdasarkan harian, mingguan atau bulanan dimana harus menghitung data dari berkas yang satu ke berkas yang lain. Atas dasar itu lah maka penelitian ini dilakukan dengan membuat sebuah sistem absensi dengan menggunakan model sistem pengembangan. Dimana dapat membantu dalam membuat data dan laporan tentang kehadiran/absensi.". (Jurnal INTEKNA, Tahun XIV, No. 2, Nopember 2014 : 102 – 209).

Juwanda Natali, Fajrillah, T.M.Diansyah **(**2016**)**, "Dalam jurnal yang berjudul implementasi *static* NAT terhadap jaringan *VLAN* menggunakan *IP Dynamic Host Configuration Protocol (DHCP)*, Untuk membangun jaringan interkoneksi *Local Area Network (LAN)* yang akan dibutuhkan dalam bentuk grup *Virtual Local Area Network (LAN)*. Alamat *IP DHCP* diberikan oleh router ke PC yang terletak di jaringan. NAT (Network Address Translation) adalah salah satu metode yang digunakan sebagai terjemahan *IP* untuk mendapatkan masuk ke yang berbeda jaringan. *NAT (Network Address Translation*) dapat memungkinkan host untuk masuk ke jaringan yang berbeda tanpa mengijinkan host dimaksudkan untuk memasuki jaringan mereka menggunakan *VLAN*. Dengan dua jaringan yang berbeda menjadi satu saklar dapat terhubung. Memberikan *IP DHCP* akan memungkinkan administrator jaringan untuk memberikan alamat *IP* ke *PC* untuk *IP* ditugaskan secara otomatis oleh router. *Host IP* diteruskan dalam jaringan dengan *NAT*". (Jurnal Ilmiah Informatika Volume 1 No. 1 / Desember 2016).

Siswo Wardoyo, Taufik Ryadi, Rian Fahrizal (2014) ”Dalam jurnal yang berjudul analisis performa file transport protocol pada perbandingan metode IPv4 murni, IPv6 murni dan tunneling 6to4 berbasis router mikrotik, penelitian ini membandingkan tingkat performa jaringan IPv4 murni, IPv6 murni dan tunneling 6to4 dengan performa FTP mencari nilai throughput. Hasil throughput yang diperoleh IPv6 murni lebih kecil 42,9% dari nilai throughput IPv4 murni, tunneling 6to4 lebih kecil 39,4% dari IPv4 murni dan IPv6 murni memiliki nilai throughput lebih besar 53,626% dari konfigurasi tunneling 6to4”. (Vol: 3 No. 2 September 2014 ISSN: 2302 – 2949).

Muhammad Yusuf, R. V, Hari Ginardi, dan Adhatus Solichah A (2016) “Dalam jurnal yang berjudul Rancang Bangun Aplikasi Absensi Perkuliahan Mahasiswa dengan Pengenalan Wajah, Proses absensi yang dilakukan secara manual dinilai kurang efektif karena terbukanya kesempatan melakukan kecurangan. Selain itu, proses rekapitulasi manual membutuhkan waktu yang lama. Sistem absensi dengan teknologi dapat diterapkan untuk membantu proses absensi dan rekapitulasi yang efektif. Pada tugas akhir ini, teknologi yang digunakan adalah sistem pengenalan wajah. Pembuatan aplikasi absensi dengan pengenalan wajah ini menggunakan metode Eigenface untuk melakukan proses pengenalan wajah. Sedangkan data-data yang dibutuhkan sistem adalah data mata kuliah, dosen, jadwal, kelas, mahasiswa, dan dataset foto wajah yang disimpan dalam sistem manajemen relasional basis data. Hasil dari aplikasi yang dibangun yaitu dapat mengelola data-data pada sistem, serta melakukan pencatatan dan perekapan data absensi. Proses absensi mahasiswa berhasil dilakukan pada kondisi pencahayaan yang bagus dan resolusi yang sama dengan kondisi foto wajah yang disimpan dalam basis data”. (JURNAL TEKNIK ITS Vol. 5, No. 2, (2016) ISSN: 2337-3539 (2301-9271 Print) A766).

Eko Budi Setiawan, Bobi Kurniawan (2016) “Dalam jurnal yang berjudul Perancangan Sistem Absensi Kehadiran Perkuliahan dengan Menggunakan *Radio Frequency Identification (RFId*), Salah satu permasalahan yang ada pada lingkungan akademik adalah absensi kehadiran kuliah. Seringnya terjadi kesalahan serta banyaknya data absensi setiap matakuliah untuk setiap mahasiswa, menjadikan prosesnya menjadi tidak efektif dan tidak efisien. Penelitian ini

membahas mengenai perancangan sistem absensi kehadiran perkuliahan di Universitas Komputer Indonesia (UNIKOM) dengan menggunakan teknologi *Radio Frequency Identification (RFID)* sehingga dapat menjadi pendukung dalam kelancaran proses perkuliahan akademik di UNIKOM". (Jurnal CoreIT, Vol.1, No.2, Desember 2015 ISSN: 2460-738X (Cetak)).