

BAB II KAJIAN PUSTAKA

2.1. Teori Dasar

2.1.1 Jaringan Komputer

Jaringan komputer adalah sebuah kumpulan dari komputer, printer, dan peralatan lainnya yang terhubung dalam satu kesatuan dan membentuk suatu sistem tertentu. Informasi bergerak melalui kabel atau tanpa kabel sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar informasi (data), mencetak data pada printer yang sama dan dapat secara simultan menggunakan program aplikasi yang sama (Maslan, 2012: 6).

Sedang menurut (Madcoms, 2009:1) dengan semakin berkembangnya kebutuhan data dan informasi, dalam sebuah organisasi dibutuhkan lebih dari satu komputer yang digunakan oleh banyak orang yang bekerja dalam sebuah tim. Untuk saling bertukar data dan informasi, maka komputer-komputer yang digunakan akan terhubung satu dengan yang lainnya. Kumpulan komputer yang saling terhubung ini disebut sebagai jaringan komputer.

Sedang Menurut (Husda, 2013: 78) jaringan komputer secara sederhana dapat dikatakan sebagai komunikasi antara dua atau lebih komputer yang saling terhubung. Dan ini sangat membantu para pengguna komputer dalam bekerja dan berkomunikasi, seperti saling bertukar data, program serta sumber daya komputer seperti penyimpanan, printer dan lain lain.

2.1.1.1 Manfaat Jaringan Komputer

Ada beberapa manfaat dari penggunaan jaringan komputer, berikut ini adalah manfaatnya (Maslan, 2012:5):

1. Dapat saling *sharing* file

Pengguna dapat saling *sharing* file kesesama komputer teman atau rekan kerja, baik itu menggunakan media kabel atau nirkabel. Dan sewaktu melakukan *sharing* pengguna bisa mengatur hak akses *user* pada saat file akan digunakan.

2. Tukar menukar data, baik data suara, gambar dan video

Tukar menukar data ini maksudnya adalah kita bisa melakukan kirim file sesama teman dan rekan kerja dalam waktu yang sangat cepat, baik menggunakan media kabel ataupun media nirkabel. Contoh pemanfaatan *Bluetooth* dan *WIFI*.

3. Memungkin dapat memakai printer secara bersamaan

Untuk penghematan biaya maka dalam manajemen perusahaan, bahwa tiap tiap departemen tidak diharuskan untuk menggunakan printer masing-masing, karena bisa saling berbagi printer.

4. Dapat menghemat biaya

Segala suatu pekerjaan dapat dikerjakan oleh satu alat saja, sehingga biaya pengeluaran dapat di minimalkan. Karena jaringan komputer segala perangkat keras bisa dihubungkan asalkan teknologi yang digunakan mendukung.

5. Efisiensi kerja meningkat

Segala pekerjaan dapat di tangani dengan memanfaatkan teknologi jaringan. Seperti tidak harus bolak-balik untuk menggunakan printer di kantor

ditempat bekerja, kirim file dengan media POS dan segala proses surat menyurat sudah bersifat *Office Automation*.

6. File-file lebih mudah di pelihara

Pengelolaan file-file sangat mudah di pelihara karena tempat penyimpanan bisa saja berpusat keserver dan keamanan terhadap data bisa terjamin, karena server dikelola oleh seorang admin server jaringan.

7. Dapat meningkatkan kinerja sistem

Kinerja sistem lebih baik karena pemeliharaan rutin dilakukan dengan mengecek komputer berdasarkan waktu yang telah ditentukan.

2.1.1.2 Jenis Media Jaringan Komputer

Menurut (Maslan, 2012:8) secara umum ada dua jenis komunikasi data:

1. Melalui Infrastruktur Terrestrial

Menggunakan media kabel dan nirkabel sebagai aksesnya. Membutuhkan biaya yang tinggi untuk membangun infrastruktur jenis terrestrial ini. Beberapa layanan yang termasuk terresterial antara lain: Sambungan data langsung (SDL), *Frame Relay*, VPN multi service, dan sambungan komunikasi data paket (SKDP).

2. Melalui Satelit

Menggunakan satelit sebagai aksesnya. Biasanya wilayah yang dicakup akses satelit lebih luas dan mampu menjangkau lokasi yang tidak memungkinkan dibangunnya infrastruktur terrestrial namun membutuhkan waktu lama untuk berlangsungnya komunikasi. Kelemahan lain komunikasi via satelit adalah adanya gangguan yang disebabkan oleh radiasi gelombang matahari (*Sun Outage*) dan

yang paling parah terjadi setiap 11 tahun sekali dan baru-baru ini sering di siarkan keberbagai televisi bahwa akan ada gangguan satelit, sehingga mengganggu komunikasi data termasuk jaringan *internet*.

Untuk menghubungkan jaringan komputer tersebut dapat menggunakan berbagai macam media akses komunikasi seperti: kabel Lan , gelombang radio, saluran telepon, satelit, maupun serat fiber optik.

2.1.2 Jenis - Jenis Jaringan Komputer

Sebenarnya, konsep dari jenis jaringan komputer sama, yaitu untuk menghubungkan berbagai perangkat jaringan untuk dapat berkomunikasi dan berbagi sumber daya. Hanya yang membedakan adalah dari letak *geologis*-nya (mencakup wilayah dan area jaringan).

Jenis Jaringan Komputer berdasarkan jangkauan area atau lokasi secara umum kita kenal ada 3 macam, yaitu LAN (*Local Area Network*), MAN (*Metropolitan Area Network*), WAN (*Wide Area Network*) (Madcoms, 2009:2)

Sedang menurut (Husda, 2013:93) jenis jaringan komputer terdiri dari 5 macam, yaitu LAN (*Local Area Network*), MAN (*Metropolitan Area Network*), WAN (*Wide Area Network*), *INTERNET* dan *Wireless*.

Berikut ini jenis-jenis jaringan komputer menurut (Maslan, 2012:20)

1. LAN (*Local Area Network*)

LAN (*Local Area Network*) merupakan jaringan milik pribadi didalam gedung atau kampus yang berukuran sampai dengan beberapa kilometer. LAN

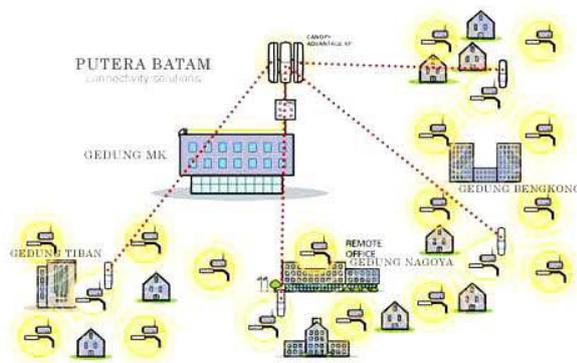
sering digunakan untuk menghubungkan komputer-komputer pribadi dan workstation dalam kantor atau perusahaan untuk pemakaian bersama dan saling bertukar informasi.



Gambar 2.1. Model LAN (*Local Area Network*)

2. MAN (*Metropolitan Area Network*)

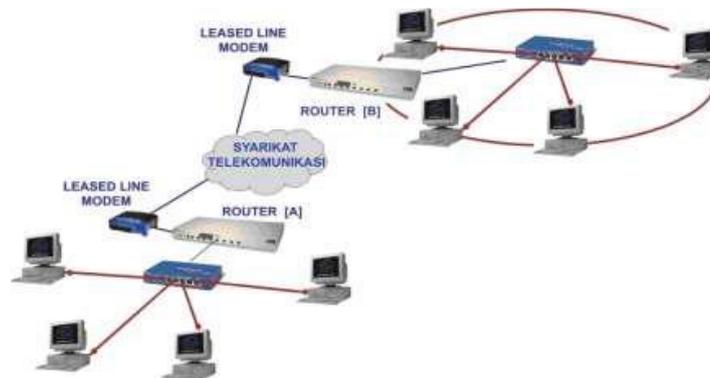
MAN (*Metropolitan Area Network*) merupakan *versi* LAN yang berukuran lebih besar, biasanya menggunakan teknologi yang sama dengan LAN. MAN dapat mencakup kantor-kantor perusahaan yang letaknya berdekatan atau juga sebuah kota dan dapat dimanfaatkan untuk keperluan pribadi atau umum. MAN mampu menunjang data dan suara bahkan dapat digunakan untuk aplikasi TV kabel.



Gambar 2.2. Model MAN (*Metropolitan Area Network*)

3. WAN (Wide Area Network)

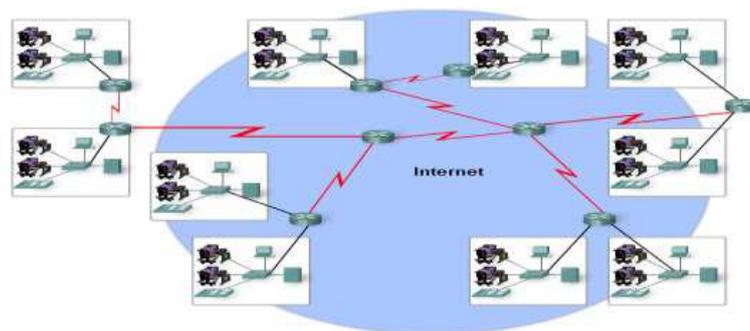
WAN (*Wide Area Network*) jangkauannya mencakup daerah geografis yang luas seringkali mencakup negara bahkan benua. Teknologi yang digunakan hampir sama dengan LAN



Gambar 2.3. Contoh Sebuah Jaringan WAN

4. INTERNET

Internet merupakan jaringan komputer yang global atau mendunia. Karena *Internet* merupakan jaringan-jaringan komputer yang terhubung secara mendunia, sehingga komunikasi dan transfer data atau file menjadi lebih mudah. *Internet* bisa dikatakan perpaduan antara berbagai jenis jaringan komputer beserta Topologi dan Tipe Jaringan yang saling berhubungan satu sama lain.

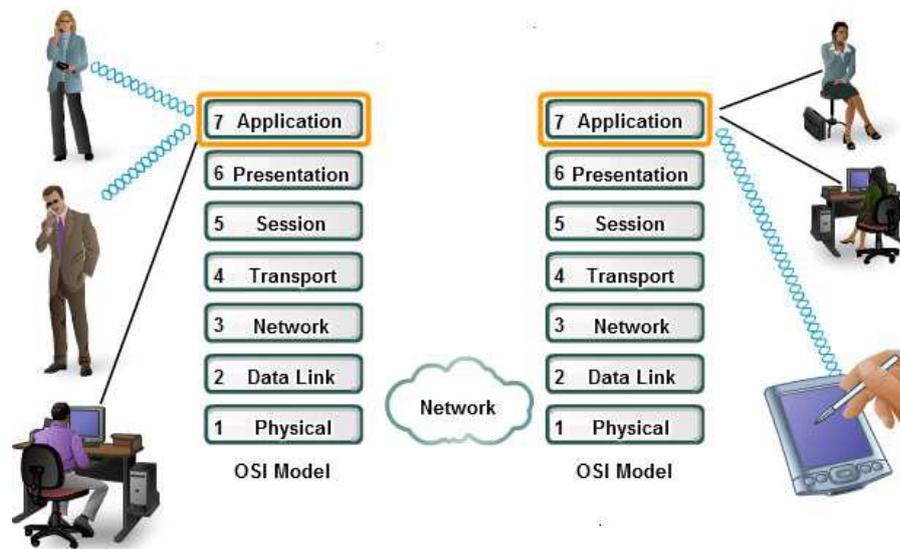


Gambar 2.4. Model Jaringan *Internet*

2.1.3 Model OSI Layer

Suatu Jaringan komputer LAN dibangun dengan memperhatikan arsitektur standar yang dibuat lembaga standar industry dunia. Standar jaringan yang saat ini di akui adalah *The Open System Connection* atau OSI yang dibuat oleh lembaga ISO (*The International Standard Organization*), Amerika Serikat. Seluruh fungsi kerja jaringan komputer dan komunikasi antar terminal diatur dalam standar ini. OSI adalah suatu standar komunikasi antar mesin yang terdiri atas 7 lapisan. Ketujuh lapisan tersebut mempunyai peran dan fungsi yang berbeda satu terhadap yang lain. Setiap layer bertanggung jawab secara khusus pada proses komunikasi data. Misal, satu layer bertanggung jawab untuk membentuk koneksi antarperangkat, sementara layer lainnya bertanggung jawab untuk mengoreksi terjadinya eror selama proses pengiriman paket data berlangsung. Model Osi dibagi dalam dua tingkatan grup yaitu: *upper layer* dan *lower layer*. Yang mana pada masing – masing grup mempunyai focus yang berbeda. Untuk *Upper layer* focus pada aplikasi pengguna dan file direpresentasikan di komputer. Sedangkan untuk *lower layer* befokus pada para *network engineering* yang membuat hardware (Maslan, 2012:34)

Tujuan utama penggunaan model OSI adalah untuk membantu desainer jaringan memahami fungsi dari tiap-tiap layer yang berhubungan dengan aliran komunikasi data, termasuk jenis-jenis *Protocol* jaringan dan metode transmisi. Model dibagi menjadi 7 layer, dengan karakteristik dan fungsinya masing-masing. Tipe layer harus dapat berkomunikasi dengan layer diatasnya maupun dibawahnya secara langsung melalui serentetan *Protocol* dan standard.(Maslan, 2012:35)



Gambar 2.5. OSI Model

Menurut (Maslan, 2012:36) penjelasannya adalah sebagai berikut:

7. Application Layer

Layer ini bertanggung jawab atas pertukaran informasi antara program komputer, dan service lain yang jalan di jaringan, seperti *server* printer atau aplikasi komputer lainnya. Lapisan ini juga menyediakan jasa untuk aplikasi pengguna. Contoh Email, *Web browser* dan lain lainnya.

6. Presentation Layer

Layer ini bertanggung jawab bagaimana data dikonversi dan diformat untuk transfer data. Contoh konversi format text ASCII untuk dokumen, gif dan JPEG untuk gambar. Layer ini juga membentuk kode konversi, translasi data, enkripsi, dan kompresi.

5. Session Layer

Layer ini menentukan bagaimana dua terminal menjaga, memelihara dan mengatur koneksi mereka saling berhubungan satu sama lain. Membentuk Virtual Circuit, dan membentuk komunikasi antar sistem. Contohnya Client Software.

4. Transport Layer

Layer ini bertanggung jawab membagi data menjadi segmen, menjaga koneksi logika “end to end” antar terminal, dan menyediakan penanganan eror. Pengiriman data yang handal, pengurutan, keabsahan data, transmisi ulang, dan flow control.

3. Network Layer

Layer ini bertanggung jawab menentukan alamat jaringan, menentukan rute yang harus diambil selama perjalanan, dan menjaga antrian *traffic* di jaringan. Data pada layer ini berbentuk paket. Terdapat juga Pengkapsulan, Packet Forwarding, Packet Filtering, dan fragmentation.

2. Data Link Layer

Layer ini menyediakan link untuk data, memaketkannya menjadi frame yang berhubungan dengan hardware kemudian diangkut melalui media. Komunikasinya dengan kartu jaringan, mengatur komunikasi layer physical antara sistem koneksi dan penanganan eror. Pada layer ini juga dapat mengubah bit ke

byte dan byte ke dalam frame, akses ke media menggunakan MAC *address*, error detection dan not connection.

1. Physical Layer

Bertanggung jawab atas proses data menjadi bit dan mengirimkan melalui media, seperti kabel, radio, fiber optik dan menjaga koneksi fisik antar system.

2.2 Teori Khusus

2.2.1 Monitoring Dan Manajemen Jaringan

Monitoring jaringan adalah pemantauan yang dapat memberikan informasi tentang status dan kecenderungan bahwa pengukuran dan evaluasi yang diselesaikan berulang dari waktu ke waktu, pemantauan umumnya dilakukan untuk tujuan tertentu, untuk memeriksa terhadap proses berikut objek atau untuk mengevaluasi kondisi atau kemajuan menuju tujuan hasil manajemen atas efek tindakan dari beberapa jenis antara lain tindakan untuk mempertahankan manajemen yang sedang berjalan (Arie Setya Putra, 2013).

Sedangkan manajemen jaringan adalah kemampuan untuk memonitor, mengontrol, dan merencanakan suatu jaringan komputer dan komponen system di dalamnya. *Monitoring* jaringan merupakan bagian dari manajemen jaringan. Hal yang paling mendasar dalam konsep manajemen jaringan adalah tentang adanya manajer atau perangkat yang memajemen dan agen atau perangkat yang dimanajemen (Anong, Hutama, Affandi, & Setijadi, 2013).

2.2.2 Komponen Monitoring Jaringan

2.2.2.1 *Simple Network Management Protocol (SNMP)*

Simple Network Management Protocol (SNMP) adalah sebuah protokol aplikasi pada jaringan TCP/IP yang dapat digunakan untuk pengelolaan dan pemantauan sistem jaringan komputer (Pradikta, Affandi, & Setijadi, 2013).

Protokol ini didesain sehingga pengguna dapat dengan mudah memantau kondisi jaringan komputer. Pemantauan kondisi jaringan dapat dilakukan dengan cara pengumpulan nilai-nilai informasi dari kondisi jaringan secara jarak jauh atau menggunakan satu pusat pengamatan.

SNMP menjadi protokol yang terus dikembangkan karena banyak perangkat jaringan yang mendukung dan tersedia layanan SNMP seperti *router*, *switch*, *server*, *workstation*, dan *printer*. Protokol SNMP pada jaringan TCP/IP menggunakan *transport* UDP oleh karena itu dalam penggunaannya tidak akan membebani *traffic* jaringan.

Pada sistem pemantauan jaringan dengan menggunakan layanan SNMP, terdapat tiga komponen dasar antara lain (Pradikta et al., 2013):

1. Manajer SNMP

Manajer adalah perangkat yang menjalankan dan dapat menangani tugas-tugas manajemen jaringan.

2. Agen SNMP

Agen SNMP adalah perangkat pada jaringan yang akan diamati dan dikelola.

Setiap agen akan merespon dan menjawab permintaan manajer SNMP.

3. *Management Information Base (MIB)*

MIB pada SNMP dapat dikatakan sebagai tempat penyimpanan informasi yang dimiliki agen. MIB yang terdapat pada SNMP didefinisikan secara hirarki dan setiap bagian mempunyai identifikasi objek (OID).



Gambar 2.6. Manajer, Agen, dan MIB SNMP

2.2.2.2 *Availability*

Availability system atau ketersediaan sistem adalah keadaan dimana suatu sistem, subsistem, atau peralatan dalam keadaan beroperasi atau berfungsi. Ketersediaan sistem biasanya diukur sebagai faktor kehandalan atau *reability*. Ketersediaan dapat mengacu pada kemampuan dari sistem atau perangkat untuk memberikan layanan pada *user*. Periode ketika sistem atau perangkat dalam kondisi hidup disebut dengan *uptime* dan untuk kondisi sebaliknya disebut *downtime* (Pradikta et al., 2013).

Downtime dapat diartikan sebagai *mean time to restore (MTTR)* dan *uptime* disebut juga *mean time between failures (MTBF)*, periode saat kondisi normal sampai terjadi gangguan berikutnya.

Availability dapat digunakan sebagai salah satu parameter dalam *Service Level Agreement* (SLA). SLA sering merujuk pada *downtime* atau *availability* untuk menentukan ketersediaan layanan.

2.2.2.3 Transport TCP

TCP/IP (*Transmission Control Protocol/Internet Protocol*) adalah standar komunikasi data yang digunakan dalam proses tukar data dari satu komputer ke komputer lain di dalam jaringan. *Transport Layer* melakukan segmentasi dan menyatukan kembali data yang tersegmentasi. Dua protokol lapisan yang paling umum *Transport TCP / IP* adalah *Transmission Control Protocol* (TCP) dan *User Datagram Protocol* (UDP) (Pradikta et al., 2013).

2.2.2.4 SNMP Traffic Graph (STG)

SNMP Traffic Grapher (STG) adalah alat untuk memantau lalu lintas di jaringan. STG menghasilkan interface yang berisi gambar yang memberikan representasi visual secara nyata dan terus menerus. Lalu lintas jaringan yang di pantau STG terdiri dari capture data di *router* yang menggunakan SNMP untuk membaca penghitung lalu lintas dari *router* dan memiliki program C yang mencatat lalu lintas data dan grafik yang mewakili menciptakan lalu lintas di koneksi jaringan yang termonitor. Grafik ini di masukan ke halaman *Web* yang dapat dilihat dari berbagai *Web-browser*.

2.2.2.5 IP Address

Menurut (Madcoms, 2009:89) IP (*Internet Protocol*) *address* merupakan alamat yang diberikan kepada komputer-komputer yang terhubung dalam suatu jaringan. IP *address* terdiri dari dua bagian, yaitu: *Network ID* dan *Host ID*. *Network ID* menentukan alamat dalam jaringan (*network address*) sedangkan *Host ID* menentukan alamat dari peralatan jaringan yang sifatnya unik untuk membedakan antara satu mesin dengan mesin yang lain. Ibarat sebuah alamat rumah, *Network ID* seperti alamat rumah dan *Host ID* seperti nomor rumah.

IP *address* berdasarkan perkembangan dibagi menjadi dua jenis:

1. IPv4 (*internet Protocol versi 4*) merupakan IP *Address* yang terdiri dari 32 bits yang dibagi menjadi 4 segmen berukuran 8 bit
2. IPv6 (*internet Protocol versi 6*) merupakan IP *Address* yang terdiri dari 128 bit yang digunakan untuk mengatasi permintaan ip *address* yang semakin meningkat.

Untuk IP *Address versi 4* terdiri dari 32 bit angka biner yang dituliskan dalam bentuk empat kelompok dan masing-masing kelompok terdiri dari delapan bit yang dipisahkan oleh tanda titik.

Contoh: 11000000.10101000.00000000.01100100

IP *Address* dapat juga ditulis dalam bentuk angka desimal dalam empat kelompok, dari angka 0 – 255.

Contoh: 172.186.0.12

2.2.2.5.1 Kelas IP Address

IP Address dibedakan menjadi 3 kelas, yaitu: kelas A, kelas B, dan Kelas C. Tujuan membedakan kelas IP adalah untuk menentukan jumlah komputer yang bisa terhubung dalam sebuah jaringan (Madcoms, 2009:89).

1. Kelas A

IP kelas A terdiri dari: 8 bit pertama digunakan untuk *Network ID* dan 24 bit berikutnya merupakan *Host ID*. IP kelas A terdapat 126 *Network*, yakni: dari nomor 1.xxx.xxx.xxx sampai dengan 126.xxx.xxx.xxx (xxx merupakan variabel yang nilainya dari 0 sampai dengan 255).

Format IP Kelas A: NNNNNNNN.HHHHHHHH.HHHHHHHH.HHHHHHHH

Contoh IP Address Kelas A: 10.11.22.33 maka *Network ID* adalah 10 dan *Host ID* adalah 11.22.33

2. Kelas B

IP kelas B terdiri dari: 16 bit pertama digunakan untuk *Network ID* dan 16 bit berikutnya merupakan *Host ID*. IP kelas B terdapat 16.384 *Network*, yakni: dari nomor 128.0.xxx.xxx sampai dengan 191.255.xxx.xxx (xxx merupakan variabel yang nilainya dari 0 sampai dengan 255).

Format IP Kelas B: NNNNNNNN.NNNNNNNN.HHHHHHHH.HHHHHHHH

Contoh IP Address Kelas B: 130.1.2.3 maka *Network ID* adalah 130.1 dan *Host ID* adalah 2.3

3. Kelas C

IP kelas C terdiri dari: 24 bit pertama digunakan untuk *Network ID* dan 8 bit berikutnya merupakan *Host ID*. IP kelas C terdapat 2.097.152 *Network*, yakni:

dari nomor 192.0.0.xxx sampai dengan 223.255.255.xxx (xxx merupakan variabel yang nilainya dari 0 sampai dengan 255).

Format IP Kelas C: NNNNNNNN.NNNNNNNN.NNNNNNNN.HHHHHHHH

Contoh IP Address Kelas C: 192.168.0.100 maka *Network ID* adalah 192.168.0 dan *Host ID* adalah 100

2.2.2.5.2 IP Address Private

IP Address private merupakan alamat-alamat IP yang disediakan untuk digunakan pada jaringan local (LAN). *IP Address private* digunakan untuk komunikasi pada jaringan yang tidak terhubung langsung dengan *internet*. *IP Address private* hanya dapat dipakai untuk komunikasi pada jaringan local dan tidak dapat digunakan pada jaringan *internet* (Madcoms, 2009:92).

Berikut daftar blok *IP Address private*:

- Range *IP Address private* kelas A: 10.0.0.1 – 10.255.255.254
- Range *IP Address private* kelas B: 172.16.0.1 – 172.31.255.254
- Range *IP Address private* kelas C: 192.168.0.1 – 192.168.255.254

2.2.2.5.3 Subnet Mask

Subnet mask merupakan angka biner 32 bit yang digunakan untuk membedakan *Network ID* dan *Host ID*. *Subnet mask* menunjukkan letak suatu *host*, apakah berada di jaringan lokal atau berada di jaringan luar (Madcoms, 2009:92).

Pada *subnet mask* bit yang berhubungan dengan *Network ID* diset 1 sedangkan yang berhubungan dengan *Host ID* diset 0.

Pembagian *subnet mask*

- IP Address kelas A menggunakan *subnet mask*

11111111.00000000.00000000.00000000 = 255.0.0.0

- IP Address kelas B menggunakan *subnet mask*

11111111.11111111.00000000.00000000 = 255.255.0.0

- IP Address kelas C menggunakan *subnet mask*

11111111. 11111111. 11111111.00000000 = 255.255.255.0

2.2.3 Komponen WEB

2.2.3.1 Apa Itu Web

Web merupakan media informasi berbasis jaringan komputer yang dapat diakses dimana saja dengan biaya relatif murah. *Web* merupakan bentuk implementasi dari bahan pemrograman *web* (*web programming*). Sejarah perkembangan bahasa pemrograman *web* diawali dengan munculnya HTML (*hypertext markup language*) yang kemudian dikembangkan dengan munculnya CSS (*cascading style sheet*) yang bertujuan untuk memperindah tampilan *website* (Komputer, 2009:2).

Sedang menurut (Raharjo, Heryanto, & RK, 2010:40) aplikasi *web* adalah aplikasi yang disimpan dan dieksekusi di lingkungan *web server*. Setiap

permintaan yang dilakukan oleh *user* melalui aplikasi klien (*web browser*) akan direspon oleh aplikasi *web* dan hasilnya akan dikembalikan lagi ke hadapan *user*.

Dengan aplikasi *web*, halaman yang tampil di layar web browser dapat bersifat dinamis, tergantung dari nilai data atau parameter yang dimasukkan oleh *user*. Data yang ditampilkan ini dapat berupa: dokument html, file gambar, file teks dan lain-lainnya.

Untuk dapat membuat suatu situs *web* yang bersifat dinamis dan di dukung dengan adanya database, dibutuhkan software sebagai berikut:

- Sebuah aplikasi *web server* (Apache, Abyss dan ISS dll).
- Bahasa pemrograman *web* (PHP, HTML dan PERL dll).
- *Server data base* (MySQL dll).
- *Web browser* klien (Microsoft IE, Mozilla, Firefox dll).
- Sebuah text editor (Note++ dll)

2.2.3.2 Hypertext Preprocessor (PHP)

PHP adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi *web*. Ketika dipanggil dari *web browser*, program yang ditulis dengan PHP akan di parsing di dalam *web server* oleh *interpreter* PHP dan di terjemahkan ke dalam dokumen HTML, yang selanjutnya akan ditampilkan ke *web browser*. Karena pemrosesan program PHP dilakukan di lingkungan *web server*, PHP dikatakan sebagai bahasa sisi *server* (*server-side*) (Raharjo et al., 2010:41)

Secara khusus, PHP dirancang untuk membentuk aplikasi *web* dinamis, dapat membentuk suatu tampilan berdasarkan permintaan. Didalam tampilan *web* yang dibentuk dengan PHP, dapat ditampilkan isi *database* ke dalam *web*. PHP tidak terbatas pada hasil *HTML (HyperText Markup Languages)*. PHP juga memiliki kemampuan untuk mengolah keluaran gambar, file PDF, dan *Movies Flash*. PHP juga dapat menghasilkan *teks* seperti XHTML dan file XML lainnya.

2.2.3.3 Sintaks PHP

Sintaks program/script PHP ditulis dalam apitan tanda khusus PHP. Ada empat macam pasangan *tag* PHP yang dapat dipergunakan untuk menandai *blok csript PHP*.

1. `<?php ...?>`
2. `<script language = "PHP"> ... </script>`
3. `<? ... ?>`
4. `<% ...%>`

Cara 1 dan 2 merupakan cara yang paling umum digunakan sekalipun cara 3 tampak lebih praktis karena cara 3 tidak selalu diaktifkan pada konfigurasi file PHP ini yang terdapat pada direktori `c:\apache\php`.

Cara 4 juga dimungkinkan sebagai kemudahan bagi pengguna yang sudah biasa dengan ASP (*Active Server Pages*).

2.2.3.4 Tipe Data PHP

PHP mendukung delapan tipe nilai atau tipe data. Empat nilai skalar (*singel value*): *integer*, *floating point*, *string* dan *boolean* dan dua nilai koleksi: *array* dan *objek*. Sisanya adalah tipe khusus: *resource* dan *null* (Raharjo et al., 2010:52).

Penjelasannya:

- a. *boolean*: adalah tipe data paling sederhana, yaitu untuk menyatakan suatu nilai kebenaran *TRUE* (benar) atau *FALSE* (salah) yang bersifat *case sensitive*.

Contoh:

```
$ lulus =TRUE;
```

- b. *integer*: adalah tipe data yang menyatakan bilangan bulat.
- c. *floating-point*: adalah tipe data bilangan *float*, *double*, atau *real*.

Contoh:

```
$a =1.234;
```

```
$a =1.2e3;
```

- d. *string*: adalah sekumpulan karakter. Dalam PHP suatu karakter dianggap sebagai suatu *byte* sehingga 256 karakter yang berbeda.
- e. *array*: adalah satu rangkaian elemen (*variable*) bertipe sama yang menempati memori secara teratur yang dapat diakses secara individu dengan menambahkan suatu *index* (*subscript*) kepada suatu nama unik. *Array* sering juga disebut larik.
- f. *object*: adalah tipe data yang memiliki kombinasi struktur data/attribut dan beberapa fungsi/*method*.

Tipe data khusus:

- g. *resource*: adalah suatu variabel khusus sebagai suatu acuan terhadap suatu *external resource*. *Resource* diciptakan dan digunakan oleh fungsi khusus.
- h. *NULL*: adalah nilai yang menyatakan bahwa suatu variabel tidak memiliki nilai.

2.2.3.5 MySQL Database Server

Hampir semua aplikasi web yang dikembangkan saat ini membutuhkan teknologi database untuk menyimpan dan mengelola data-data yang digunakan didalamnya. PHP memberikan dukungan terhadap banyak jenis database baik yang bersifat komersial maupun yang tidak.

MySQL merupakan sistem database yang banyak digunakan untuk pengembangan aplikasi web. Alasannya mungkin karena gratis, pengelolaan datanya sederhana, memiliki tingkat keamanan yang bagus, mudah diperoleh dan lain-lain.

Beberapa perintah dalam MySQL untuk menjalankan mysql client:

- `> mysql -uUSER -pPASSWORD -hHOST` untuk mengganti argumen *user*, *password* dan *host* dengan argumen yang kita inginkan.
- `mysql> CREATE DATABASE CONTOHDB;` untuk membuat database dengan nama data base CONTOHDB.
- `mysql> SHOW DATABASES;` untuk menampilkan daftar database yang terdapat didalam *server* mysql.

- `mysql> USE contohdb;` untuk memilih salah satu database yang diinginkan dari dalam daftar database.
- `mysql> show tables;` untuk menampilkan semua tabel yang terdapat didalam database.
- `mysql> DESC user;` untuk menampilkan daftar kolom dari suatu tabel, diikuti nama tabelnya.
- `mysql> select user from user where host 'localhost';` untuk menampilkan data dari suatu tabel.

2.2.4 Rekayasa Perangkat Lunak

2.2.4.1 Perangkat Lunak

Menurut (A.S & Shalahuddin, 2013:2) perangkat Lunak (*software*) adalah merupakan program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*). Sebuah program komputer tanpa terasosiasi dengan dokumentasinya maka belum dapat disebut perangkat lunak. Sebuah perangkat lunak juga sering disebut dengan sistem perangkat lunak. Sistem bearti kumpulan komponen yang saling terkait dan mempunyai satu tujuan yang ingin dicapai.

Sangat banyak sekali aplikasi dari perangkat lunak, berikut beberapa aplikasi perangkat lunak (A.S & Shalahuddin, 2013:4):

1. Perangkat lunak sistem (*system software*)
2. Perangkat lunak waktu nyata (*real-time software*)

3. Perangkat lunak bisnis (*business software*)
4. Perangkat lunak untuk keperluan rekayasa dan keilmuan (*engineering and scientific software*)
5. Perangkat lunak tambahan untuk membantu mengerjakan suatu fungsi dari perangkat lunak lainnya (*embedded software*)
6. Perangkat lunak komputer personal (*personal computer software*)
7. Perangkat lunak berbasis *web* (*web based software*)
8. Perangkat lunak berinteligensi buatan (*artificial intelligence software*)

2.2.4.2 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin. Perangkat lunak banyak dibuat dan pada akhirnya sering tidak digunakan karena tidak memenuhi kebutuhan pelanggan atau karena masalah non-teknis seperti keengganan *user* atau pemakai perangkat lunak untuk mengubah cara kerja dari manual ke otomatis (komputerisasi) (A.S & Shalahuddin, 2013:5).

Beberapa kriteria rekayasa perangkat lunak yang harus terpenuhi agar perangkat lunak yang dibuat bisa terpakai dengan baik (A.S & Shalahuddin, 2013:5):

1. Dapat terus diperlihara setelah perangkat lunak selesai dibuat seiring berkembangnya teknologi dan lingkungan (*maintability*)

2. Dapat diandalkan dengan proses bisnis yang dijalankan dan perubahan yang terjadi
3. Efisiensi dari segi sumber daya dan penggunaan
4. Kemampuan dipakai sesuai dengan kebutuhan (*usability*)

2.2.4.3 Fase Rekayasa Perangkat Lunak

Dalam rekayasa perangkat lunak ada beberapa fase yang harus dilalui, menurut (A.S & Shalahuddin, 2013:6) terdapat 3 fase Yaitu :

1. Fase pendefinisian “what”

Mencari tahu atau mengidentifikasi informasi apa yang harus diproses, seperti apa fungsi dan performansi yang diinginkan.

2. Fase pengembangan “how”

Mendefinisikan bagaimana data distrukturkan dan bagaimana fungsi-fungsi yang dibutuhkan diimplementasikan di dalam arsitektur perangkat lunak, bagaimana *detail* prosedural diimplementasikan, bagaimana karakter antarmuka tampilan, bagaimana desain ditranslasikan ke bahasa pemrograman, dan bagaimana pengujian akan dijalankan.

3. Fase pendukung (*support phase*)

Ini fase perbaikan pada kesalahan, adaptasi yang dibutuhkan, perbaikan akibat perubahan kebutuhan pelanggan.

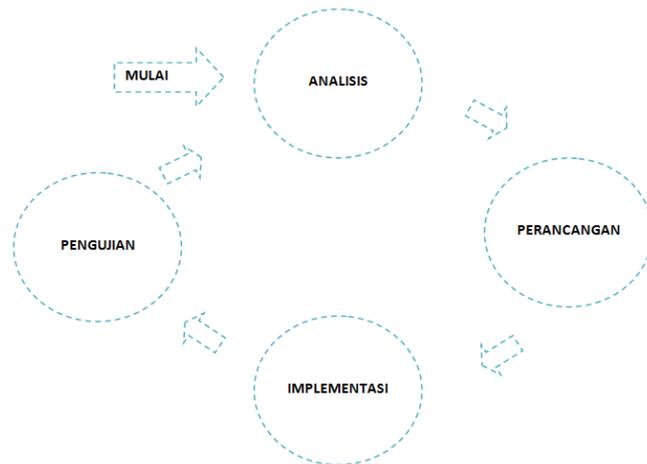
Pada fase ini ada 4 tipe perubahan:

- a. Koreksi (*correction*)
- b. Adaptasi (*adaptation*)

- c. Perbaikan (*enhacement*)
- d. Pencegahan (*prevention*)

2.2.4.4 Proses Rekayasa Perangkat Lunak

Proses Rekayasa Perangkat Lunak dilakukan selama pembangunan perangkat lunak, menurut (A.S & Shalahuddin, 2013:9) proses-proses yang dilakukan dalam rekayasa perangkat Lunak secara garis besar adalah sebagai berikut:



Gambar 2.7. Tahapan Umum Rekayasa Perangkat Lunak

Proses rekayasa perangkat lunak umumnya adalah sebagai berikut:

1. Pengumpulan spesifikasi (*specification*)

Mengetahui apa saja yang harus dapat dikerjakan sistem perangkat lunak dan batasan pengembangan perangkat lunak.

2. Pengembangan (*development*)

Pengembangan perangkat lunak untuk menghasilkan sistem perangkat lunak.

3. Validasi (*Validation*)

Memeriksa apakah perangkat lunak sudah memenuhi keinginan pelanggan (*customer*)

4. Evolusi (*Evolution*)

Mengubah perangkat lunak untuk memenuhi perubahan kebutuhan pelanggan (*customer*).

2.2.4.5 Cara Konversi Rekayasa Perangkat Lunak

Melakukan konversi dari cara kerja yang lama ke cara kerja yang baru menggunakan perangkat lunak dikembangkan akan menghabiskan lebih banyak sumber daya, dana, dan waktu.

Ada beberapa cara dalam konversi rekayasa perangkat lunak, seperti berikut ini (A.S & Shalahuddin, 2013:13):

1. Konversi Paralel



Gambar 2.8. Ilustrasi Konversi Paralel

Konversi paralel dilakukan dengan melakukan beberapa waktu transisi dimana ada waktu dimana kedua sistem (sistem lama dan sistem baru) berjalan bersama untuk keperluan transisi sampai sistem baru dapat berjalan mandiri. Sumber daya yang dibutuhkan pada konversi paralel akan banyak terkuras pada waktu transisi.

2. Konversi Langsung



Gambar 2.9. Ilustrasi Konversi Langsung

Konversi langsung dilakukan di mana sistem lama secara ekstrim langsung diganti dengan sistem yang baru. Konversi ini akan mengalami waktu yang sangat sulit di awal berjalannya sistem baru

3. Konversi Per Fase



Gambar 2.10. Ilustrasi Konversi Per Fase

Konversi per fase dilakukan dengan berpindah per fase dari sistem lama ke sistem baru misalkan pada awal konversi hanya pada pekerjaan memasukkan data-data saja, pada tahap berikutnya mulai menggunakan proses perhitungan, lalu fase berikutnya mulai menggunakan proses pelaporan sistem baru, dan seterusnya (lebih fokus pada per fungsi sistem).

4. Konversi Pilot atau *Single Location*



Gambar 2.11. Ilustrasi Konversi Pilot

Konversi pilot dilakukan dengan melakukan konversi per unit kerja atau per lokasi di dalam sebuah lingkungan kerja. Misalnya pada tahap awal unit kerja yang sistemnya berubah adalah bagian keuangan, berikutnya pada bagian sumber daya manusia, dan seterusnya.

2.2.5 Unified Modeling Language (UML)

2.2.5.1 Sejarah UML

UML (*Unified Modeling Language*) muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak.

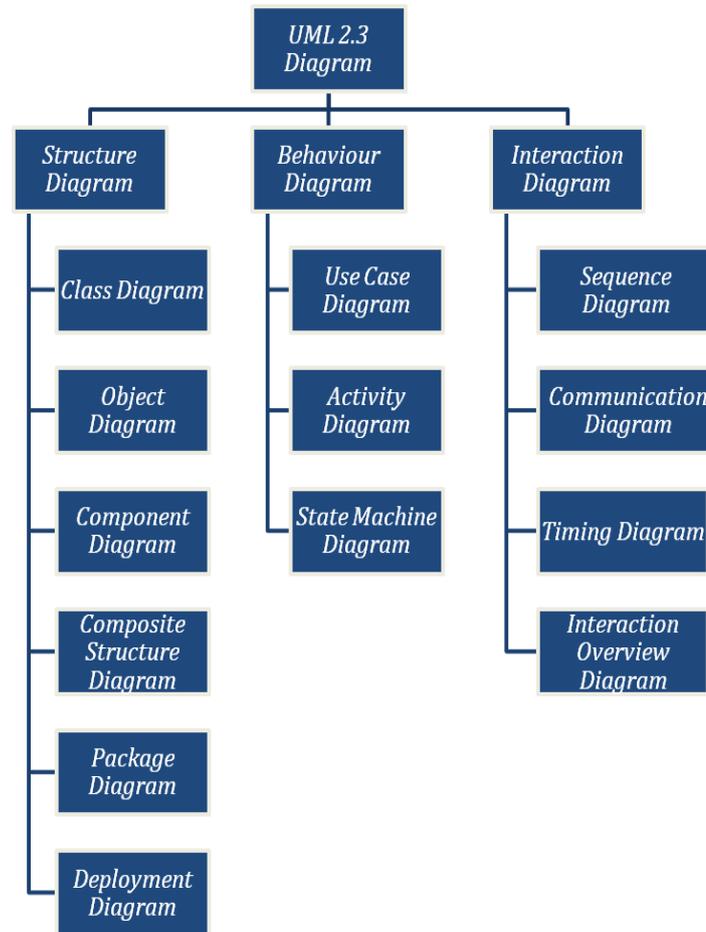
Berawal pada bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama *Simula-67* yang dikembangkan pada tahun 1967. Bahasa pemrograman ini kurang berkembang dan dikembangkan lebih lanjut, namun dengan kemunculannya telah memberikan sumbangan yang besar pada developer pengembang bahasa pemrograman berorientasi objek selanjutnya.

Pada akhirnya, tahun 1996 Object Management Group (OMG) mengajukan proposal adanya standarisasi pemodelan berorientasi objek dan pada bulan september 1997 UML diakomodasi OMG sehingga sampai saat ini UML telah memberikan kontribusi cukup besar dalam metodologi berorientasi objek.

UML (*Unified Modeling Language*) adalah salah satu standar bahasa pemodelan yang banyak digunakan di dunia industri untuk mendefinisikan permintaan untuk membuat analisa dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (A.S & Shalahuddin, 2013:133).

2.2.5.2 Diagram UML

Pada UML *versi* 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori (A.S & Shalahuddin, 2013:140), berikut gambar diagram UML:



Gambar 2.12. Diagram UML

Berikut ini penjelasan dari pembagian kategori diatas:

1. *Structure Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

- *Class Diagram*: menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.
- *Object Diagram*: menggambarkan struktur sistem dari segi penanaman objek dan jalannya objek pada sistem. Pada diagram objek harus dipastikan semua

kelas yang sudah didefinisikan pada diagram kelas harus dipakai objeknya, karena jika tidak, pendefinisian kelas itu tidak dapat dipertanggung jawabkan.

- *Object Diagram*: dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada di dalam sistem. Diagram komponen juga dapat digunakan untuk memodelkan hal-hal berikut:

1. *Source code* program perangkat lunak
2. Komponen *executable* yang dilepas ke *user*
3. Basis data secara fisik
4. Sistem yang harus beradaptasi dengan sistem lain
5. *Framework* sistem

Komponen yang biasanya ada dalam suatu sistem adalah sebagai berikut:

1. Komponen *user interface* yang menangani tampilan
 2. Komponen *bussines processing* yang menangani tampilan
 3. Komponen data yang menangani fungsi-fungsi proses bisnis
 4. Komponen *security* yang menangani keamanan sistem
- *Composite Structure Diagram*: diagram ini digunakan untuk menggambarkan struktur dari bagian-bagian yang saling terhubung maupun mendeskripsikan struktur pada saat berjalan (*runtime*) dari *instance* yang saling terhubung.

Contoh: Menggambarkan deskripsi dari setiap bagian mesin yang saling terkait untuk menjalankan fungsi mesin tersebut, menggambarkan aliran *router* pada jaringan komputer.

- *Package Diagram*: menyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam diagram UML. Hampir semua diagram dalam UML dapat dikelompokkan menggunakan *package* diagram.

- *Deployment Diagram*: menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga digunakan untuk memodelkan hal-hal berikut:

1. Sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, node dan hardware
2. Sistem *client/server* misalnya seperti gambar berikut
3. Sistem terdistribusi murni
4. Rekayasa ulang aplikasi

2. *Behaviour Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

- *Use Case Diagram*: merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. Use Case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang dibuat. Use Case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Ada 2 (dua) hal utama pada use case yaitu pendefinisian apa yang disebut aktor dan use case:

1. Aktor, merupakan orang atau proses atau sistem lain yang berinteraksi dengan sistem informasi.

2. Use Case, merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

- *Activity Diagram*: menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokkan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

- *State Machine Diagram*: diagram digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem. Perubahan tersebut digambarkan dalam suatu graf berarah. State Machine Diagram merupakan pengembangan dari diagram Finite State Automata dengan penambahan beberapa fitur dan konsep baru.

3. *Interaction Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

- *Sequence Diagram*: menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima objek. Untuk menggambar diagram sekuen perlu diketahui objek-objek yang terlibat didalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Banyaknya diagram sekuen yang harus digambar sesuai dengan pendefinisian use case yang memiliki proses sendiri.
- *Communication Diagram*: merupakan penyederhanaan dari diagram kolaborasi. Diagram komunikasi menggambarkan interaksi antar objek/bagian dalam bentuk urutan pengiriman pesan. Diagram komunikasi merepresentasikan informasi yang diperoleh dari diagram kelas, diagram sekuen, dan diagram use case untuk mendeskripsikan gabungan antara struktur statis dan tingkah laku dinamis dari suatu sistem. Diagram komunikasi mengelompokkan message pada kumpulan diagram sekuen menjadi sebuah diagram.
- *Timing Diagram*: merupakan penggambaran yang fokus pada penggambaran terkait waktu. Timing diagram digunakan untuk menggambarkan sistem dalam periode waktu tertentu. Timing diagram biasa digunakan untuk mendeskripsikan operasi dari alat digital karena penggambaran secara visual akan lebih mudah dipahami daripada dengan kata-kata
- *Interactive Overview Diagram*: Interaction overview diagram mirip dengan diagram aktivitas yang berfungsi untuk menggambarkan sekumpulan urutan aktivitas. Interaction overview diagram bentuk aktivitas diagram setiap titik merepresentasikan diagram interaksi. Interaksi diagram dapat meliputi diagram

sekuen, diagram komunikasi, interaction overview diagram dan timing diagram.

2.3 Tools/Software/Aplikasi/System

2.3.1 Perangkat Keras

Adapun perangkat keras yang dibutuhkan (minimum) sebagai berikut:

1. Personal Computer, Processor Intel Pentium IV 1.8 GHz,
2. Harddisk 500 Gbytes
3. Memory 2048 Mbytes
4. Layar Monitor
5. Keyboard
6. Mouse

2.3.2 Perangkat Lunak

Adapun perangkat lunak yang dibutuhkan (minimum) sebagai berikut:

1. Sistem Operasi Linux, yaitu Linux U-Buntu seri 14.0, berfungsi sebagai OS web *server*.
2. *SSH Server versi OpenSSH_4.7*, berfungsi untuk bisa diremote dari jarak jauh.
3. *Apache versi Apache/2.2.8*, *PHP versi PHP5.2.4-2ubuntu5.10* dan *MySQL versi MySQL 5.0.51a-3ubuntu5.4*.
4. *Net-snmp versi 5.4.1*, merupakan peralatan yang di gunakan sebagai pengontrol *Protocol SNMP*.

5. STG (*SNMP Traffic Grapher*) versi 1.4.5, merupakan alat untuk memantau lalu lintas di jaringan.

2.4 Penelitian Terdahulu

Berdasarkan teori yang telah diuraikan diatas maka di dapatkan hasil penelitian terdahulu sebagai berikut:

1. Tangaguling, Limpraptono, and Sotyohadi, (2012), Perancangan dan Pembuatan Aplikasi Monitoring *Traffic* Jaringan Intranet Berbasis *Web* Dengan Menggunakan Protokol SNMP. Dengan Uraian: Dengan perkembangan teknologi informasi pada saat sekarang ini para administrator jaringan dituntut untuk bekerja dengan cepat,handal,dan professional ketika terjadi masalah pada lalu lintas (*traffic*) infrastruktur jaringan yang ditanganinya. Dari permasalahan tersebut maka diperlukan suatu aplikasi monitoring *traffic* jaringan yang mampu memantau jaringan intranet suatu instansi yang dapat diakses melalui *web*, sehingga secara otomatis para administrator memiliki mobilitas yang cukup tinggi. Dengan dibuatnya “Aplikasi monitoring *traffic* jaringan intranet berbasis *Web* dengan menggunakan protokol SNMP” diharapkan dapat membantu administrator jaringan komputer untuk mengetahui,memantau dan menjaga stabilitas lalu lintas (*traffic*) dari suatu jaringan intranet melalui *web*.
2. Pradikta, Affandi, and Setijadi, (2013), Rancang Bangun Aplikasi Monitoring Jaringan dengan Menggunakan *Simple Network Management Protocol*.

Dengan uraian: Perkembangan teknologi telekomunikasi yang sangat cepat akan menghadirkan beragam bentuk layanan bagi konsumen. Hal tersebut berbanding lurus dengan bisnis untuk membangun jaringan yang lebih besar dan lebih baik dengan harga yang lebih terjangkau. Manajemen jaringan terutama sistem monitoring menjadi sesuatu yang penting dilakukan. *Simple Network Management Protocol* (SNMP) adalah sebuah protokol aplikasi pada jaringan TCP/IP yang dapat digunakan untuk pengelolaan dan pemantauan sistem jaringan komputer. Tetapi layanan dan informasi SNMP hanya dapat diakses melalui tampilan pada command prompt atau terminal sehingga dalam penggunaannya tidak efektif dan sulit dilakukan karena masih membutuhkan pengolahan dan tampilannya sulit dimengerti. Solusi yang pernah dilakukan adalah membuat *Graphical User Interface* (GUI) sebagai perantara untuk mengambil dan menampilkan nilai SNMP. Tetapi solusi yang ditawarkan masih mempunyai kekurangan, karena hasil yang ditampilkan hanya sebatas informasi kondisi jaringan pada saat itu dan masih belum ada sistem untuk menyimpan dan mengolah nilai SNMP lebih lanjut.

3. Kijazi, Email, and Michael, (2014), A Step on Developing *Network Monitoring Tools*. Dengan uraian: *Network Monitoring involves Using Software or hardware based Systems or a combination of both to constantly observe the status of Network devices and hosts, and notifies the Network administrator via email, SMS or other alarms in case of error or fail. Observing the status of Network device and hosts is done when the Monitoring System speaks with the Networking devices or hosts using*

different *Protocols* within the *Protocols* stack (OSI Layer) ,see Figure1.The aim of this paper is to provide a footstep on developing a *Network* monitoring tool for monitoring *Network devices* and *hosts*. This is a software based *Network* Monitoring tool using a combination of Simple *Network* Monitoring *Protocol* (SNMP), *Internet* Control Message *Protocol* (ICMP) and Port scanning concept.

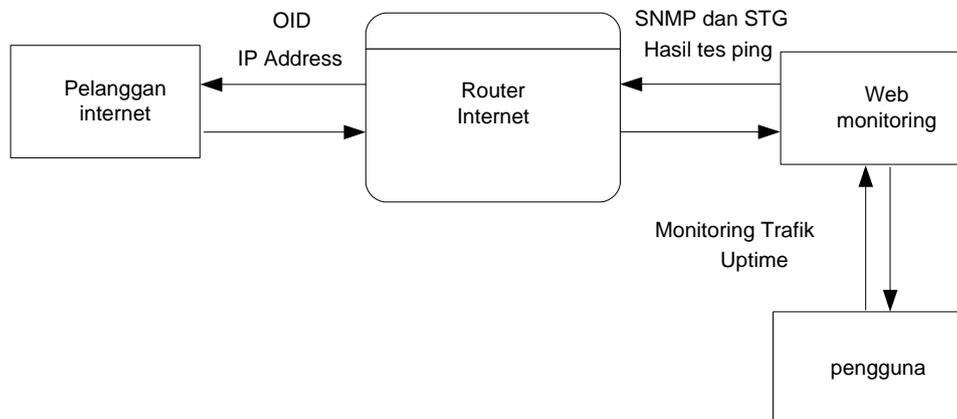
4. Setiawan and Adelia, (2011), Monitoring *Traffic* Jaringan dan Pengaturan PC *Router* Berbasis web (studi kasus: Labkom STIKOM Surabaya). Dengan uraian: Laboratorium merupakan sarana untuk melakukan berbagai macam riset atau percobaan salah satunya jaringan komputer termasuk laboratorium komputer STIKOM Surabaya. Seorang administrator jaringan pada labkom bertugas untuk memanajemen dan memonitoring jaringan dari manajemen bandwidth, vlan, gateway dan security ,a dmin mengalami kendala untuk memanajemen *traffic* jaringan dan membatasi penggunaan bandwith serta pemblokiran *website* yang tidak di perkenankan. Berdasarkan permasalahan yang terjadi maka diperlukan sebuah aplikasi yang dapat memberikan informasi tentang kegiatan yang terjadi pada jaringan lokal untuk menjaga performa jaringan pada labkom dan juga memberikan informasi visualiasi kondisi *traffic* jaringan pada labkom serta dapat memfasilitasi administrator jaringan untuk membatasi penggunaan bandwith serta pemblokiran *website* yang tidak diperkenankan pada saat praktikum berlangsung.
5. Nugroho, Affandi, and Rahardjo,(2014),Ranca Bangun Aplikasi Monitoring Jaringan Menggunakan SNMP (*Simple Network Management Protocol*)

dengan Sistem Peringatan Dini dan Mapping Jaringan. Dengan uraian: Pada periode sebelumnya, di Laboratorium Jaringan Telekomunikasi Jurusan Teknik Elektro ITS telah dibuat tiga sistem monitoring jaringan, yaitu *network monitor* yang dilengkapi dengan *database*, sistem peringatan dini, dan pemetaan jaringan (*Network Mapping*). Ketiga sistem ini masih berdiri sendiri, sehingga perlu dilakukan integrasi agar didapat suatu sistem yang memiliki fungsi lengkap. Dalam penelitian ini dilakukan perancangan dan pembuatan suatu Network Monitoring System yang merupakan integrasi antara *Network Monitoring*, *Network Mapping*, dan Sistem Peringatan Dini. Hasilnya adalah aplikasi *network monitoring system* yang dibuat menjalankan semua fungsi sesuai perancangan. Waktu yang diperlukan untuk menampilkan peta jaringan berbanding lurus dengan banyaknya *device* yang terhubung dengan *server* aplikasi. Pada pengujian hasil *availability*, aplikasi memiliki tingkat keakuratan 100% dibandingkan dengan hasil perhitungan menggunakan persamaan.

2.5 Kerangka Pemikiran

Menurut (Sudaryono, 2015:21) kerangka berfikir merupakan model konseptual tentang bagaimana teori berhubungan dengan berbagai faktor yang telah di indentifikasi sebagai masalah yang penting. Berdasarkan permasalahan yang ada maka pembuatan web untuk monitoring jaringan ini akan membantu dalam kegiatan memonitoring dan melakukan pengecekan jaringan.

Berdasarkan tinjauan kepustakaan yang telah diuraikan, dalam kerangka pemikiran melalui paradigma penelitian digambarkan sebagai berikut:



Gambar 2.13. Kerangka Berfikir

Pada gambar diatas di jelaskan bahwa semua pelanggan *internet* tersambung ke sebuah *router internet*. Di *router* ini setiap pelanggan memiliki *ip address* dan *oid* masing masing. Dengan menggunakan *oid* dan *ip address* inilah dibuat aplikasi monitoringnya. Aplikasi monitoringnya berbasis web. Informasi yang bisa di monitoring adalah *traffic* data dan status jaringan untuk masing-masing pelanggan. Pengguna dapat melihat informasi tersebut melalui *web*.