

BAB II

DASAR TEORI

2.1 KONSEP DASAR WI-FI

Komunikasi *nirekabael* atau lebih dikenal dengan istilah *wireless* sudah menjadi media komunikasi yang banyak digunakan didalam kehidupan sehari - hari, salah satu teknologi yang dikembangkan adalah *Wi-Fi*, dimana teknologi ini mempunyai banyak keuntungan karena medianya tanpa kabel maka akan lebih fleksibel berbeda dengan teknologi LAN (*Local Area Network*) yang menggunakan kabel sebagai media pengiriman data dimana kabel yang digunakan adalah kabel jenis UTP (*Unshieled Twisted Pair*) dan kabel jenis koaksial. Setiap perangkat yang mempunyai *Wi-Fi device* dapat saling terkoneksi dengan hanya memerlukan ruang atau *space* dengan catatan jarak dibatasi oleh kuat sinyal dari masing - masing perangkat.[4]

2.1.1 Teknologi jaringan *Wi-Fi*

Wi-Fi (Wireless Fidelity) merupakan suatu standar jaringan tanpa kabel, dimana perangkat yang dapat terhubung ke jaringan tersebut harus memiliki komponen yang sesuai. Teknologi *Wi-Fi* mempunyai standar yang mengacu ke IEEE (*Institute of Electrical and Electronic Engineers*), dengan standarisasi sebagai berikut:

- 802.11a merupakan *Wi-Fi* yang menggunakan frekuensi 5 GHz dan mempunyai kecepatan data 54 Mbps dalam jangkauan jaringan 300m
- 802.11b merupakan *Wi-Fi* yang menggunakan frekuensi 2,4 GHz dan mempunyai kecepatan data 11 Mbps dalam jangkauan jaringan 100m
- 802.11g merupakan *Wi-Fi* yang menggunakan frekuensi 2,4 GHz dan mempunyai kecepatan data 54 Mbps dalam jangkauan jaringan 300m

- 802.11n merupakan *Wi-Fi* yang menggunakan frekuensi 2,4 GHz dan mempunyai kecepatan data 100 Mbps dalam jangkauan jaringan 300m.[4]

Atau dapat dilihat pada tabel 2.1,

Tabel 2.1 Spesifikasi *Wi-Fi*

Spesifikasi <i>Wi-Fi</i>			
Spesifikasi	Kecepatan	Frekuensi Band	Cocok dengan
802.11b	11 Mb/s	2,4 GHz	b
802.11a	54 Mb/s	5 GHz	a
802.11g	54 Mb/s	2,4 GHz	b,g
802.11n	100 Mb/s	2,4 GHz	b,g,n

2.1.2 Tipe Jaringan *Wi-Fi*

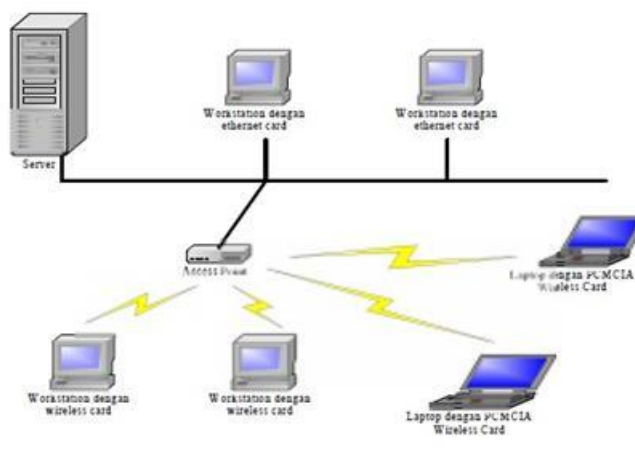
Jaringan *Wi-Fi* mempunyai dua buah tipe yaitu *Wireless LAN* dan *Infrastruktur Wireless*.

- Jaringan *Ad Hoc Wireless LAN* atau *peer to peer*, dimana masing – masing komputer mampu melakukan komunikasi dengan menggunakan SSID (*Service Set Identifier*). SSID merupakan sebuah identitas dari komputer yang sudah support *Wi-Fi*, atau dari sebuah *Access Point*.



Gambar 2.1 . Ad-Hoc *Wireless LAN*[4]

- *Wireless Infrastructure* atau Jaringan *Server Based*, dimana dalam tipe ini membutuhkan suatu komponen khusus yang mampu untuk menyediakan layanan *Wi-Fi* atau bertindak sebagai *Access Point*.[4]

Gambar 2.2 Infrastruktur *Wireless LAN*[4]

2.1.3 Komponen Utama jaringan *Wi-Fi*

Dalam jaringan *Wi-Fi* terdapat 4 buah komponen utama untuk yaitu sebagai berikut:

- 1) *Access Point* : merupakan komponen yang berfungsi untuk mengirimkan data, menerima data yang masuk, serta untuk mengubah sinyal radio menjadi sinyal digital atau sebaliknya. *Access point* bertindak seperti hub/switch di jaringan LAN *Ethernet*, dimana satu perangkat *Access Point* mampu untuk menampung maksimal 40 *client*.
- 2) *Wireless LAN Device* : merupakan jenis komponen yang terdapat disisi PC/mobile.
- 3) *Mobile/Desktop PC* : merupakan komponen yang memiliki fungsi untuk mengakses data yang terjadi disisi *client*, diaman mobil PC biasanya sudah terpasang PCMCIA (*Personal Computer Memory Card International Association*) sedangkan perangkat *desktop PC* diperlukan tambahan komponen khusus yaitu PCI (*Peripheral Componen Interconnect*) Card, dan USB (*Universal Serial Bus*) Adapter.
- 4) *Ethernet LAN* : merupakan jaringan kabel yang sudah terpasang apabila diperlukan.[4]

2.2 Merancang jaringan *Wi-Fi*

- a. Untuk merancang jaringan *Wi-Fi* diperlukan komponen berikut :

- *Server* (Penyedia layanan/*service* dalam jaringan komputer) harus memiliki spesifikasi minimum komputer untuk instalasi jaringan *Wi-Fi* :
 - 1) CPU Pentium IV MHz
 - 2) *Memory* RAM 256 MB
 - 3) *Harddisk Space* 2 GB
 - 4) *Graphic Card Super* VGA 1204x768
 - 5) *Network Interface Card* (NIC) kecepatan 10/100 Mbps
 - 6) Sebuah *Wireless Access Point*
 - *Client/Wokstation* (Menerima Layanan dari *Server*) memiliki Spesifikasi perangkat yang bisa dijadikan *Client* adalah memiliki fasilitas *Wi-Fi*, dapat berupa PC atau sebuah *Mobile Phone*.
- b. Keunggulan dan kekurangan *Wi-Fi*
- 1) Keunggulan jaringan *Wi-Fi* :
 - Biaya dalam melakukan pemeliharaan murah
 - Infrastruktur jaringan yang berdimensi kecil
 - Pembangunan jaringan yang cepat
 - Perelokasian yang mudah
 - Membangun *portabilitas*
 - 2) Kelemahan jaringan *Wi-Fi* :
 - Biaya untuk peralatan yang mahal
 - Mempunyai *delay* yang besar
 - Mempunyai kesulitan dalam hal propagasi radio
 - Interferensi
 - Keamanan data kurang[4]

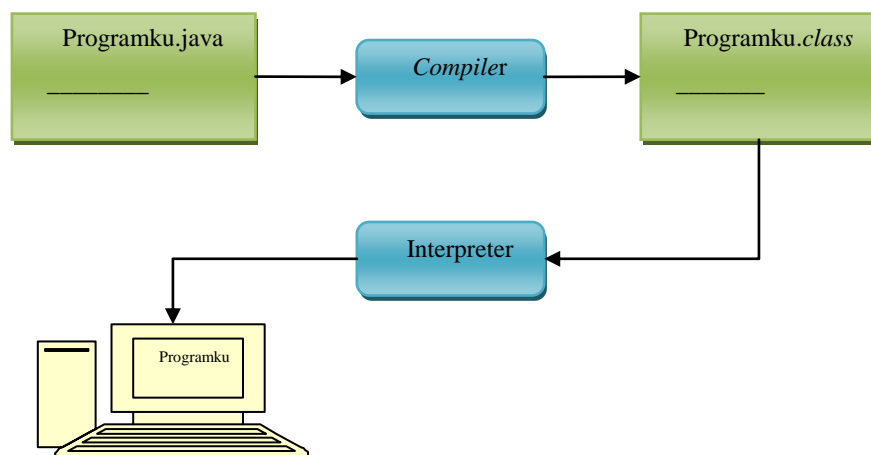
2.3 JAVA

Sebuah program yang berbasis java dapat berjalan di komputer apabila kode-kode dari program tersebut telah di *compile* dan di *interpeter*/diterjemahkan. Untuk melakukan kompilasi diperlukan *software* yang bernama *compiler* sedangkan untuk melakukan penerjemahan diperlukan sebuah *software* yang bernama *interpeter*, keduanya mempunyai perbedaan dalam mengeksekusi kode, dimana untuk *compiler* eksekusi kode dilakukan secara menyeluruh sedangkan untuk *interpeter*

dilakukan dengan cara mengeksekusi kode baris demi baris sehingga membuat waktu eksekusi lebih lama dari *compiler*.

Java sendiri adalah program yang mempunyai dua proses tersebut yaitu *compiler* dan *interpeter*. Adapun cara kerjanya yaitu kode program yang masuk akan di *compile* terlebih dahulu dengan merubahnya kedalam bentuk *bytecode* atau kode objek, yang selanjutnya kode objek tersebut akan dieksekusi per baris. Sehingga dalam Java proses *compile* kode terjadi sekali, sedangkan untuk proses penerjemahan dilakukan setiap kali program dieksekusi.

Didalam JVM (*Java Virtual Machine*) terdapat sebuah bahasa mesin yang merupakan kumpulan perintah yang disebut dengan *Bytecode*. Sebuah program yang berbasis java tidak dapat dikerjakan di perangkat yang tidak mempunyai JVM.[5]



Gambar 2.3 Diagram block konsep kerja Java[6]

Dalam pembuatan program yang berbasis java diperlukan minimal dua *software*, yaitu:

1. *Text*(JDK)

Text adalah sebuah program yang berfungsi untuk menterjemahkan kode-kode java ke *bytecode* sehingga kode-kode tersebut dapat berjalan di *Java Runtime Environment*. *Java Runtime Environment* sendiri adalah sebuah *software* yang berfungsi untuk menjalankan program yang berbasis java.

Sebuah Program Language yang mempunyai kemampuan untuk dapat berjalan di *platform Operation System* yang berbeda serta berorientasi ke objek adalah Java. Java bersifat *open source* sehingga dapat dikembangkan sesuai dengan sistem operasi yang diperlukan yang artinya tidak mengacu kepada satu sistem saja. Java sendiri mempunyai kemampuan untuk membuat *web*, *desktop*, aplikasi serta yang lainnya, seperti dikerjakan dengan menggunakan bahasa pemrograman lainnya[6]. Java memiliki karakteristik yaitu:

a. Sederhana (*Simple*)

Java menggunakan bahasa pemrograman yang hampir sama dengan bahasa pemrograman yang digunakan oleh C++ tetapi Java mempunyai kelebihan dimana *pointer* dan *multiple inheritance* dihilangkan. Selain itu Java memakai *memory garbage collection* dan juga *automatic memory allocation*..

b. Berorientasi objek (*Object Oriented*)

Java adalah pemrograman dengan berorientasi ke objek sehingga dapat dikerjakan dengan cara modular serta dapat digunakan kembali. Pemrograman yang berorientasi objek artinya adalah memetakan dunia nyata ke bentuk objek serta berinteraksi antara objek-objek tersebut.

c. Terdistribusi (*Distributed*)

Java terintegrasi dengan *Libraries networking* sehingga akan memudahkan dalam pendistribusian pembuatan aplikasi.

d. *Interpreted*

Java berperan sebagai *interpeter* dengan menggunakan *Java Virtual Machine (JVM)* sehingga data yang sudah dikompilasi ke bentuk *bytecode* dapat berjalan di *platform* yang berbeda.

e. Robust

Java memiliki reliabilitas yang cukup tinggi. *Compiler* di Java memiliki kemampuan untuk mendeteksi kesalahan secara lebih teliti

dibandingkan dengan bahasa pemrograman lainnya. Java sendiri memiliki *runtime-Exception handling* yang berfungsi untuk membetulkan program yang salah.

f. *Secure*

Sebagai salah satu bahasa pemrograman yang digunakan untuk pembuatan aplikasi yang berbasis internet, java mempunyai berbagai mekanisme keamanan yang berfungsi untuk menjaga aplikasi yang dibuat tidak merusak sistem komputer yang menjalankan aplikasi tersebut.

g. *Architecture Neutral*

Java merupakan program yang berbasis pada *platform independent*, dimana program ini cukup memiliki satu jenis versi yang bisa berjalan di *platform* yang berbeda-beda dengan menggunakan JVM..

h. *Portable*

Java memiliki *source code* yang ringan serta dapat berjalan di berbagai *platform* yang berbeda tanpa harus mengkompilasi ulang program tersebut.

i. *Performance*

Sebagai salah satu bahasa pemrograman, Java tergolong memiliki *performance* yang kurang baik, namun hal itu dapat diatasi dengan menambah *compile* pada Java dengan *compile* java lainnya seperti buatan Inprise.

j. *Multithreaded*

Java adalah salah satu bahasa pemrograman yang mampu untuk melakukan berbagai pekerjaan sekaligus dengan suatu program.

k. *Dynamic*

Java adalah bahasa pemrograman yang dirancang agar mampu berjalan di lingkungan yang berbeda atau dinamis, dimana perubahan

pada suatu *class* bisa dikerjakan walaupun *class* tersebut sedang dijalankan oleh suatu program.[7]

2. *Text Editor*

Software text editor adalah salah satu *software* yang memiliki fungsi sebagai media untuk menuliskan *source code* dari program yang akan dibangun dan memberikan kemudahan kepada penggunanya untuk menulis *source code* yang berbasis android *text* dengan menggunakan Java IDE (*Integrated Development Environment*) yaitu *software* eclipse.

➤ Instalasi Eclipse

Dalam pembuatan widget android diperlukan 3 *file* yaitu:

- 1) Eclipse
- 2) ADT Plugin
- 3) Android SDK

Langkah awal yang harus dilakukan dalam menginstal ketiga *file* tersebut adalah menginstal terlebih dahulu JDK (*Java Development Kit*) yang merupakan lingkungan dari sebuah pemrograman dalam menuliskan sebuah program aplikasi java. *Java Development Kit* mempunyai lingkungan yang berbeda dalam hal mengeksekusi suatu program pada sistem operasi seperti dalam mengkompilasi sebuah program, membetulkan *books*, serta menjalankan berbagai tambahan yang ditulis dalam bahasa java.

Eclipse adalah salah satu *text editor* yang bisa digunakan untuk mendevlop android namun dengan tambahan plugin yang bernama *Android Development Tools (ADT)* sehingga mampu membuat aplikasi yang berbasis android. Dalam plugin itu sendiri sudah terdapat *emulator* yang digunakan untuk menjalankan aplikasi yang sudah dibuat, namun terlebih dahulu harus menginstal *Software Text (SDK)*.

SDK adalah alat tambahan dan merupakan *tools Application Programming Interface (API)* yang digunakan untuk membuat sebuah aplikasi berbahasa java pada *platform* Android. Android adalah sebuah

sistem operasi pada perangkat seluler yang dikembangkan oleh Google dengan beberapa fitur penting seperti berikut:

- a. *Framework* merupakan aplikasi yang berfungsi untuk mendukung dalam mengganti komponen.
- b. *Mesin Virtual Dalvik* dapat ditingkatkan.
- c. *Integrated browser*.
- d. *Global Position System, Camera, serta accelerometer*.
- e. Lingkungan dari *development* yang memadai.



Gambar 2.4 Bagan Instalasi Eclipse[8]

Sehingga eclipse dapat berjalan apabila sudah menginstal dan mengkonfigurasi ADT dan SDK terlebih dahulu. Aplikasi yang sudah dibuat di eclipse dapat dijalankan terlebih dahulu di *emulator* yang ada di eclipse. *Emulator* merupakan sistem yang dapat menjalankan Sistem Operasi pada sistem komputer yang berbeda, dimana *emulator* di eclipse dapat dijalankan sesuai dengan sistem operasi android yang diinginkan sehingga bisa diketahui aplikasi tersebut dapat berjalan dengan baik atau tidak, dan apabila aplikasi itu berhasil berjalan maka tinggal di masukan ke *Handphone* yang berbasis android dengan mengambil *file .apk* dari eclipse tersebut.[8]

2.4 XAMPP

XAMPP adalah paket program *web* lengkap yang digunakan untuk pemograman *web*, khususnya PHP dan MySQL, didalam XAMPP terdapat PHP dan MySQL[9].

2.4.1 MySQL

MySQL adalah *software* atau program *Database Server*, sedangkan SQL (*Structured Query Language*) adalah bahasa pemogramannya yang merupakan bahasa permintaan (*query*) dalam *database server*, termasuk dalam MySQL itu sendiri. MySQL mendukung perintah *Data Definition Statements*, yaitu perintah dasar MySQL untuk mengelola *database* dan tabel di dalamnya. Perintah ini sering disebut dengan istilah DDL (*Data Definition Language*), atau DDS (*Data Definition Statements*). Untuk membuat *database* menggunakan *xmapp control panel* dimana MySQL dan *Apache* dijalankan, kemudian membuka *localhost* dengan menggunakan *web browser*, alamat yang dituju adalah *localhost*[9].

2.4.2 PHP

PHP (*PHP Hypertext Preprocessor*) merupakan bahasa pemograman berbasis *web*, yaitu bahasa program yang digunakan untuk membuat aplikasi berbasis *web* (*website*, *blog*, atau aplikasi *web*). PHP termasuk bahasa program yang hanya bisa dijalankan disis *server*, atau sering disebut *Side Server Language*, tanpa adanya *server web* yang terus berjalan PHP tidak akan bisa di jalankan. Dimana *server* yang dibutuhkan bisa berupa komputer yang dipakai, salah satu *web server* yang biasa dipakai adalah *Apache*. Hasil dari PHP adalah berbentuk *web*, baik itu *website*, *blog* maupun aplikasi berbasis *web*. Karena berbasis *web* maka bisa dionlinekan di *hosting*. PHP tidak ada aplikasi/*software* nyata yang digunakan untuk menuliskan kodenya, PHP yang telah terinstal berbentuk folder *compiler*[9].

2.5 Android

- Android Versi 4.4 (*KitKat*)

Android dengan sistem operasi kit kat mulai diluncurkan pada 4 september 2013, sebelum bernama kitkat android 4.4 ini diberi nama *key lime pie* tetapi terdapat masalah dalam pengejaannya secara umum di masyarakat, oleh karena itulah diganti menjadi kitkat yang sudah familiar.



Gambar 2.5 Logo Android 4.4 (*KitKat*)[10]

2.5.1 Fitur Android

Fitur - fitur yang terdapat dalam android yaitu:

- Kerangka aplikasi yang merupakan fitur untuk mempermudah pengguna untuk melakukan penghapusan komponen.
- *Dalvik mesin virtual* pengoptimalan dalam mesin ini ditunjukkan untuk perangkat *mobile*.
- Grafik: dalam android terdapat dua grafik yaitu grafik 2D dan 3D.
- SQLite: fitur ini berfungsi untuk menyimpan data di Perangkat.
- Mendukung media seperti diantaranya yaitu audio, video, dan berbagai *format* gambar.
- Mendukung teknologi GPS (*Global Position System*).[3]

2.5.2 Kelebihan dan Kekurangan Android

1. Kelebihan Android

- Android merupakan sistem operasi yang bersifat *open source* sehingga setiap orang bebas untuk ikut serta dalam pengembangan aplikasi android.
- Android sendiri merupakan *realisasi* dari *cloud computing* yang sudah terintegrasi dengan Google sehingga setiap yang disimpan

di telepon seluler akan otomatis tersimpan juga di *account* Google.

- Android memiliki market tersendiri sehingga dalam pengupdatean OS akan lebih mudah.
- Bersifat *Fleksible*, karena android dapat berjalan di berbagai *platform* yang berbeda.
- Android membebaskan penggunanya untuk memilih jenis *hardware* yang ingin digunakan.
- Android memiliki kemampuan untuk dapat menjalankan banyak aplikasi dalam sekali waktu, serta memberikan kemudahan dalam hal pemberitahuan untuk SMS, Email serta pemberitahuan lainnya.[11]

2. Kekurangan Android

- Android bergantung pada koneksi internet
- Dalam pemakaiannya, android menguras banyak daya *battery*. [11]

2.5.3 Arsitektur Android

Arsitektur sistem operasi android ditunjukkan pada gambar 2.6

1. *Applications*

Aplikasi yang terdapat dalam android merupakan aplikasi yang dibuat dengan menggunakan bahasa pemrograman Java seperti dalam pembuat aplikasi SMS, Browser dan yang lainnya.

2. *Application Framework*

Aplikasi ini dibuat untuk lebih menyederhanakan pemakaian ulang dari komponen, dimana aplikasi lain akan dapat menggunakannya apabila aplikasi tersebut mempublishnya, dimana semua itu didasari pada:

- a. Sepaket tampilan yang digunakan untuk pengembangan aplikasi termasuk *button* dan *browser*.
- b. *Content Provider* yang berfungsi untuk menjalankan aplikasi lain dan mengaksesnya serta mampu melakukan *sharing* data.

- c. *Resource Manage*, memberikan akses untuk dapat melakukan *resource code* pada grafik.
- d. *Notification Manager*, yang berfungsi untuk menampilkan notifikasi pada status bar.
- e. *Activity Manager*, berfungsi agar pengguna dapat melakukan *navigasi backstack*. [11]

3. *Libraries*

Didalam android terdapat seperangkat C/C++ *libraries* yang berfungsi untuk pengembangan android, diantaranya yaitu:

- a. *System C library*, merupakan *implementasi* dari turunan BSD yang tidak lain adalah *system C library standar* (*libc*) yang sudah sesuai dengan perangkat Linux.
- b. *Media Libraries*, *libraries* ini merupakan *libraries* yang berbasis *Packet Video's OpenCORE*, dimana mendukung *library playback* dan juga berbagai jenis gambar.
- c. *Surface Manager*
- d. *Lib Web Core*.
- e. SGL, yang merupakan engine grafis 2D.
- f. *3D libraries*.
- g. SQLite.

4. *Android Runtime*

Masing - masing aplikasi android bekerja berdasarkan prosesnya sendiri seperti contohnya yaitu *Dalvik Virtual Mesin* yang bertugas untuk mengeksekusi *file* yang memiliki *format file* berupa *.dex*, dimana tugas itu berfungsi untuk membuat memori yang dipakai lebih efektif. DVM akan di kompilasi oleh *compiler* Java dan dirubah kedalam bentuk *.dex* yang dilakukan oleh *dx tool*.

5. Linux Kernel

Android merupakan sistem operasi yang berbasis Linux dimana versi Linux yang digunakan adalah versi 2.6 yang digunakan sebagai layanan sistem *core* diantaranya adalah layanan keamanan, model

driver, manajemen memori dan proses, serta *network stack*, selain itu Kernel juga berfungsi sebagai sebuah *layer* abstrak yang menghubungkan antara *hardware* dengan *software*. [3]



Gamabr 2.6 Arsitektur Android[11]

2.6 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan salah satu bahasa visual yang berfungsi untuk menjembatani pengguna yang ingin melakukan analisis serta desain sistem yang mempunyai orientasi ke arah objek guna mevisualisasikannya, serta membangun dan mendokumentasikan artefak dari setiap sistem perangkat lunak.

1. Use Case Diagram

Use Case Diagram merupakan sebuah metode yang dilakukan dengan cara menggambarkan mekanisme sistem dimana didalamnya terdapat aktor serta tugas dari masing - masing aktor. Diagram ini memberikan solusi untuk menggambarkan keseluruhan dari sistem yang sedang berjalan serta memberikan kemudahan dalam perencanaan sistem yang akan dibuat selanjutnya untuk mengembangkan sistem yang sudah ada.

2. *Activity Diagram*

Activity diagram merupakan sebuah diagram yang menggambarkan alur dari kerja sistem yang sedang berjalan serta bisa dipakai dalam berbagai cara, selain itu digram ini juga mempunyai kemampuan untuk memberikan detail dari masing - masing alur kerja yang terjadi di antara ataupun yang terjadi didalam *use case*.

3. *Sequence Diagram*

Sequence Diagram merupakan digram yang mampu menyesuaikan dengan lingkungan atau bersifat dinamis dan digunakan sebagai diagram untuk komunikasi UML. Diagram ini digunakan untuk memperbaiki logika serta skenario dari sistem yang sedang berjalan dan juga berfungsi untuk memvisualisasikan tahapan dari sistem yang sedang berjalan.

4. *Class Diagram*

Class Diagram merupakan diagram yang berfungsi untuk menjelaskan tipe dari masing - masing objek didalam sebuah sistem serta menjelaskan jenis hubungan yang terjadi diantara objek-objek tersebut.[11]

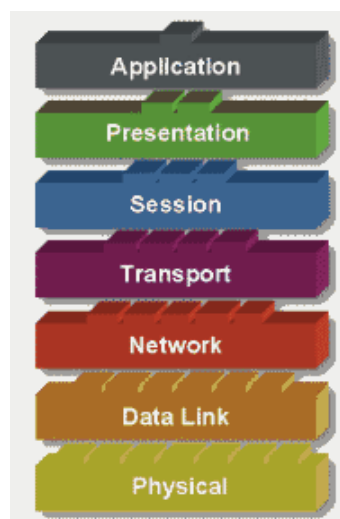
2.7 *Client/Server*

Komputer *server* merupakan komputer yang mempunyai kemampuan untuk menawarkan sebuah layanan khusus ke komputer *client* ataupun ke jaringan lainnya. *Server* mempunyai berbagai jenis, salah satunya yaitu *file server* yang merupakan layanan untuk menyimpan data serta dapat menemukannya kembali apabila data tersebut diperlukan.

Sedangkan untuk komputer *client* merupakan komputer yang bertugas sebagai pemakai layanan dari *server*, dimana komputer *client* dilengkapi dengan sebuah program yang khusus untuk menjalankannya. Layanan yang digunakan oleh komputer *client* adalah mengambil *file - file* dari *server*, dimana komputer *client* akan mengirimkan permintaan ke *server* dan *server* akan membalasnya dengan mengirimkan *file - file* yang diminta tersebut.

Komputer *Server* dan komputer *client* akan dapat berkomunikasi apabila keduanya menggunakan suatu cara yang sama, dimana cara tersebut dikenal dengan istilah protokol.[1]

Protokol merupakan suatu aturan yang mengatur serta mengijinkan adanya komunikasi atau pemindahan *file* dari satu titik ke titik lain, baik itu antara dua buah komputer atau lebih dari dua buah komputer. Protokol bekerja pada semua sistem, dimana dapat bekerja di *hardware*, *software*, atau diantara keduanya, dimana pada tingkat yang rendah protokol bertugas untuk menterjemahkan koneksi pada *hardware*. Didalam protokol terdapat tiga prinsi penting yang harus diperhatikan yaitu kehandalan, efektivitas serta kemampuan untuk meminimalisir koneksi gagal pada sebuah jaringan. Protokol terdapat pada masing - masing *layer*, terdapat dua jenis *layer* yaitu TCP/IP dan OSI *layer*, perbedaanya adalah banyaknya *layer* yang bekerja, dimana untuk TCP/IP *layer* yang bekerja adalah, *application*, *Transport*, *Network* dan *data link*, sedangkan untuk OSI *Layer* terdapat tujuh buah *layer* yaitu *Application*, *Presentation*, *Session*, *Transport*, *Network*, *Data Link*, dan *Physical*. *Open System Interconnection* (OSI) merupakan sebuah model yang konseptual dimana masing - masing *layer* mempunyai perannya masing - masing, OSI sendiri di kembangkan oleh sebuah badan internasional yang bernama ISO (*International Organization for Standardization*).[12]



Gambar 2.7 OSI Layers[12]

Application merupakan *layer* yang berfungsi sebagai *interface* antar *user* dengan aplikasi dimana didalam *layer* ini terdapat protokol HTTP yang berfungsi untuk menghubungkan aplikasi ke jaringan, selain itu di dalam *layer* ini juga terdapat protokol lain seperti SMTP, FTP dan NFS.

Presentation merupakan *layer* yang berfungsi untuk mengubah *format* data yang akan dikirimkan ke *format* yang dapat di transmisikan ke jaringan, didalam *layer* ini terdapat beberapa protokol yaitu perangkat lunak *redirektor* sebagai layanan semacam *Virtual Network Compute* (VNC) atau *Remote Desktop Protocol* (RDP).

Session merupakan *layer* yang berfungsi untuk membuat sebuah koneksi, memelihara koneksi yang telah dibuat serta menghancurkan koneksi tersebut apabila sudah tidak diperlukan lagi, selain itu didalam *layer* ini terjadi resolusi nama.

Transport merupakan *layer* yang mempunyai fungsi untuk memotong data menjadi paket - paket yang masing - masing paket diberi nomer urut sehingga mempermudah dalam penyusunannya kembali. Didalam *layer* ini setiap pengiriman data akan mendapat timbal balik atau sering dikenal dengan *acknowledgement* sehingga data yang dikirimkan akan benar - benar sampai ke tujuan.

Network merupakan *layer* yang berfungsi untuk menterjemahkan alamat IP serta memberikan *Header* untuk paket yang masuk ke *layer* ini dari arah *layer transport*.

Data Link merupakan *layer* yang berfungsi untuk mendefinisikan alamat MAC (*Media Access Control*) serta memberikan framer berupa header dan tailer ke data yang masuk dari *layer network*, dalam *layer* ini dibagi menjadi dua level yaitu lapisan MAC (*Media Access Control*) dan lapisan LLC (*Logical Link Control*).

Physical merupakan *layer* yang berfungsi untuk mendefinisikan bagaimana transmisi data dilakukan apakah menggunakan media kabel atau menggunakan media *wireless*, dan juga mengatur bagaimana media tersebut berinteraksi dengan *Network Interface Card* (NIC).

Pembentukan paket data dimulai dari *layer application* yang kemudian akan masuk ke *layer presentation*, *session* dan selanjutnya data akan dipotong menjadi beberapa paket dengan diberi nomer urut di *layer transport* yang kemudian akan masuk ke *layer network* dimana dalam *layer* ini paketan tersebut diberi *header* IP kemudian masuk ke *layer data link* dan diberi *header* dan *tailer* berupa *MAC address* yang selanjutnya akan di transmisikan melalui media yang sudah terpasang dengan melewati *layer physical*, jika data sampai tujuan maka akan dilakukan pembongkaran data seperti pembongkaran *header* dari masing - masing *layer* sehingga diperoleh data yang sebenarnya, dimulai dari *layer physical* sampai di tampilkan di *layer application*. [12]

2.8 WIRESHARK

Wireshark merupakan sebuah aplikasi komputer yang berfungsi untuk menganalisa pengiriman paket data yang disajikan secara *realtime*, dan memiliki kemampuan untuk menangkap dan menyimpan data yang dikirimkan tersebut. wireshark mempunyai fungsi diantaranya:

1. Melakukan perbaikan pada masalah yan timbul di jaringan
2. Melakukan testing terhadap masalah keamanan
3. Mampu untuk men-*debugging implementasi* protokol
4. Sebagai media untuk belajar protokol pada jaringan

Wireshark merupakan program yang sering digunakan unutm menganalisa paket data yang dikirimkan karena beberapa fitur unggulan sebagai berikut:

1. Memiliki berbagai versi yang dapat diinstal diberbagai *platform* seperti: UNIX, Linux, Windows, dan Mac.
2. Dapat melakukan *capture* paket data jaringan secara *real time*
3. Dapat menampilkan informasi protokol secara lengkap
4. Paket data yang sudah di *capture* dapat disimpan serta dapat dimunculkan kembali.
5. Mampu melakukan penyaringan paket data jaringan.
6. Pencarian paket data dapat dilakukan dengan lebih spesifik.
7. Dapat menampilkan data yang statistik. [13]

2.9 IP (*Internet Protocol*)

2.9.1 Alamat IP

IP merupakan alamat dari masing - masing *host* yang mendefinisikan bahwa *host* tersebut terhubung ke suatu jaringan. Dalam alamat IP terdapat *Host ID* dan *Network ID*:

1. *Network ID*, menunjukkan bahwa sistem berada pada jaringan fisik yang sama di router IP.
2. *Host ID*, menunjukkan stasiun kerja, alamat router, alamat *server* dan juga *host* TCP/IP dalam suatu jaringan.

Alamat IP merupakan sekumpulan dari 32 bit yang dipecah menjadi 4 bagian sehingga setiap bagian mempunyai 8 bit yang dikenal dengan istilah *oktet*, dimana *oktet* tersebut akan dirubah kedalam bentuk desimal dari 0-255 seperti contoh pada tabel 2.2.

Tabel 2.2 Konversi alamat IP

Format Biner	Format Desimal
11000000 10101000 00000001 00011000	192.168.1.24

2.9.2 Kelas Alamat

Alamat IP dibagi menjadi tiga kelas yaitu kelas A, B dan C seperti pada tabel 2.3 dimana untuk kelas D dan E digunakan untuk tujuan penelitian. Pengalamatan tersebut bertujuan untuk memudahkan *menajemen* alamat IP dari suatu *host*.

Tabel 2.3 Perbandingan *Network ID*

Kelas IP	<i>Network ID</i>
A	1.0.0.0 – 126.0.0.0
B	128.0.0.0 – 191.255.0.0
C	192.0.0.0 – 223.255.255.0

2.9.3 PROTOKOL

1. *Address Resolution Protocol* (ARP) merupakan protokol yang berfungsi untuk mengatur mekanisme komunikasi mesin sehingga dikenal dilingkungkannya dengan cara memetakan alamat MAC ke alamat IP sehingga diperoleh hubungan yang terjadi diantara keduanya.
2. *Dynamic Host Configuration Protocol* (DHCP) merupakan protokol yang berfungsi untuk memberikan alamat IP secara otomatis ke setiap perangkat yang terhubung ke jaringan, karena setiap perangkat/*host* yang terhubung ke jaringan harus memiliki sebuah alamat IP.
3. *Transmission Control Protocol* (TCP) merupakan protokol yang berfungsi untuk memberikan kehandalan dalam pengiriman paket data sehingga data tersebut dapat sampai ke tujuan sepenuhnya dimana protokol ini bekerja pada *layer transport* sehingga memiliki sistem *acknowledgement* yang berfungsi sebagai timbal balik dalam pengiriman data, dan jika ada data yang hilang di perjalanan maka data yang hilang tersebut akan dikirimkan kembali.
4. *Hypertext Transfer Protocol* (HTTP) merupakan protokol yang berfungsi sebagai penyedia alamat yang berbasis *World Wide Web* dimana protokol ini mempunyai *default port* yaitu 80.[13]