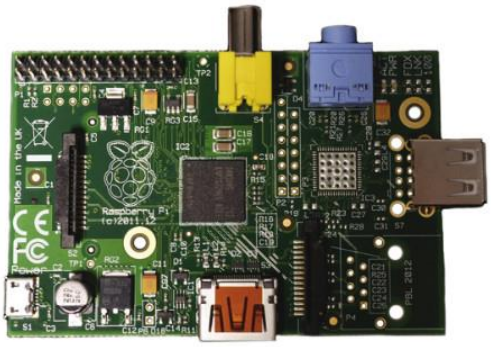
BAB II

TINJAUAN PUSTAKA

Pada bab ini akan dijelaskan mengenai studi pustaka dan teori-teori penunjang perancangan sistem *printer* nirkabel baik perangkat keras (*hardware*), maupun perangkat lunak (*software*), serta beberapa teori penunjang lainnya.

2.1. Raspberry Pi

Raspberry Pi (juga dikenal dengan RasPi) adalah sebuah SBC (*Single Board Computer*) yang menggunakan *system on a chip* (SoC) dari Broadcomm BCM2835, juga sudah termasuk prosesor ARM1176JZF-S 700 MHz, GPU Video Core IV dan RAM sebesar 256 MB (untuk Rev. A) dan 512 MB (untuk Rev. B).



Gambar 2. 1 Raspberry Pi Rev.B

2.1.1. Spesifikasi Raspberry Pi

Raspberry memiliki spesifikasi sebagai berikut.

1. USB *port*

   Jumlah USB *port* pada tipe-tipe *raspberry* berbeda, untuk Rev.A memiliki 1 buah USB *port* dan untuk Rev.B memiliki 2 buah USB port. USB *port*

yang digunakan adalah USB 2.0. tarikan awal arus maksimumnya adalah 100mA untuk jangka pendek.

2. LAN Port

   Pada sebuah *raspberry* terdapat 1 LAN *port* yang digunakan untuk mengakses jaringan.

3. CSI Header

   Pada bagian belakang LAN port terdapat sebuah CSI *header* yang terdiri dari 15 pin *flat flex* dan terhubung pada GPU. CSI *header* sebagai standar *interface serial* yang bisa dihubungkan pada suatu kamera CSI-COMPLIANT.

4. DSI Header

   DSI (*Display Serial Interface*) secara luas dapat digunakan untuk modul LCD. Seperti halnya CSI, DSI juga terhubung dengan GPU. Apabila DSI digunakan dengan tambahan *inter-Integrated Circuit Bus* (I2C *bus*) maka akan memberikan kemampuan *touch-interface*.

5. SD *Card Slot*

   Media penyimpanan pada sebuah *raspberry* adalah *card*. Dimana *card* tersebut sebagai media penyimpan OS yang akan digunakan *raspberry*. Jenis-jenis card yang bisa digunakan yaitu SD/MMC/SDIO.

6. GPIO *Headers*

   GPIO (*General-Purpose Input/Output*) merupakan pin generik berjumlah 26 pin. Pada konektor GPIO digunakan inisial “P1-XX” agar tidak membingungkan penggunanya, “XX” tersebut menandakan letak posisi pin tersebut. Jika dilihat bagian bawah PCB pada GPIO header terdapat

sebuah label bertuliskan “Pi”, itu menandakan pin 1 GPIO atau inisialnya Pi-01. Berhadapan dengan itu adalah P1-02. Pin di akhir kebalikan P1-01 adalah P1-25 dan pin di akhir kebalikan P1-02 adalah P1-26. Beberapa pin diberi label seperti “NC” atau “DNC”. Itu berarti “*No Connect*” atau “ *Do Not Connect*” dan untuk pin yang berlabel seperti itu tidak dihubungkan dengan apapun.

Tabel 2.1 Konfigurasi Pin GPIO

Pin Number	**Primary Function**	**Alternate Function**	**Alternate 1 Function**
P1-01	3.3 V	I2C SDA	
P1-03	GPIO 0	I2C SCL	
P1-05	GPIO 1		GPCLK0
P1-07	GPIO 4		
P1-09	NC		
P1-11	GPIO 17		ALT3=UART0_RTS ALT5=UART1_RTS
P1-13	GPIO 21	PCM_DIN	ALT5=GPCLK1
P1-15	GPIO 22		ALT3=SD1_CLK ALT4 = ARM_TRST
P1-17	NC		
P1-19	GPIO 10	SPI0_MOSI	
P1-21	GPIO 9	SPI0_MISO	
P1-23	GPIO 11	SPI0_SCLK	
P1-02	5 V		
P1-04	NC		
P1-06	GND		
P1-08	GPIO 14	UART0_TXD	ALT5=UART1_TDX
P1-10	GPIO 15	UART0_RDX	ALRT5=UART1_RXD
P1-12	GPIO 18		ALT4=SPI1_CE0_N ALT5 =PWM0

P1-14	NC		
P1-16	GPIO 23		ALT3=SD1_CMD ALT4=ARM_RTCK
P1-18	GPIO 24		ALT3=SD1_DATA0 ALT4=ARM_TD0
P1-20	NC		
P1-22	GPIO 25		ALT4=ARm_TCK
P1-24	GPIO 8	SPI0_CE0_N	

7. *Analog Video Output*

   Pada *raspberry* terdapat *analog video output*. *Analog video output* tersebut dalam wujud gabungan RCA *output*. Selain analog video output terdapat juga HDMI *output*, tetapi kedua *output* tersebut tidak bisa digunakan sekaligus sebagai *output*. Jika mempunyai keluaran kedua-duanya, HDMI akan menjadi sebagai keluaran aktif.

8. *Audio Output*

   Audio output dalam bentuk 3,5mm *stereo audio socket*. Seperti halnya *analog video output*, *audio output* juga tidak digunakan bersamaan dengan HDMI, ketika HDMI digunakan maka *output* suara pada audio *output* tidak akan keluar.

9. LED

   Terdapat LED yang berfungsi sebagai indikator dan menyatakan status atau kondisi sebuah raspberry yang dijelaskan pada table dibawah ini.

Tabel 2.2 Status LED

LED and LED Color	Description
D5 *Green*	*System okay*/SD *card access*
D6 *Red*	*Power okay,* 3.3 V
D7 *Green*	*Full duplex*; *half duplex if the* LED *is off*
D8 *Green*	*Link activity for the* LAN

10. Pi Brains

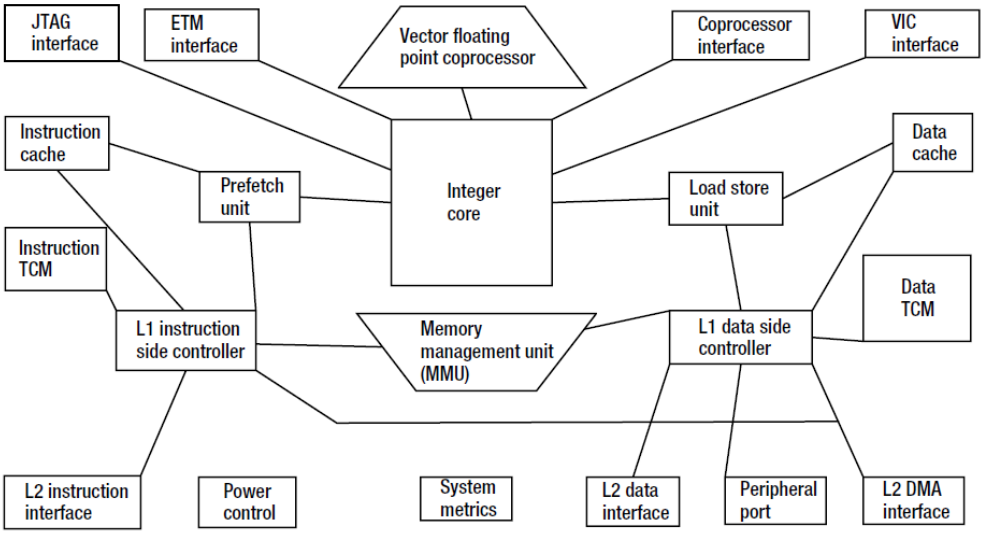
    Pi *Brains raspberry* menggunakan ARM1176JZF-S 700 MHz, versi ARM yang dipakai adalah arsitektur ARMv6.

11. *Memory Chips*

    Untuk Raspberry Rev.A sebesar 256 MB dan untuk Rev.B memiliki 512 MB. *Memory chip* ini tidak utuh digunakan sebagai RAM, tetapi penggunaan memorinya dibagi dengan penggunaan memori GPU.

2.1.2. Arsitektur Raspberry Pi

Raspberry Pi menggunakan sistem operasi berbasis kernel Linux.Raspbian merupakan Sistem operasi berbasis Debian yang dapat bebas dioptimalkan untuk perangkat keras Raspberry Pi.



Gambar 2. 2 Blok Diagram Arsitektur Raspberry Pi

GPU *hardware* diakses melalui gambar *firmware* yang di-*load* ke GPU saat *boot* dari SD-*card*. Gambar *firmware* dikenal sebagai kumpulan *biner*, sementara *driver* Linux yang terkait adalah sumber tertutup (*closed source*). Aplikasi perangkat lunak menggunakan panggilan ke sumber tertutup *run-time library* yang pada gilirannya menjadi panggilan *open source driver* dalam Linux kernel. API driver kernel spesifik untuk perpustakaan tersebut bersifat tertutup. Aplikasi video menggunakan OpenMAX, aplikasi 3D menggunakan OpenGL ES dan 2D aplikasi menggunakan OpenVG yang pada nantinya menggunakan EGL. OpenMAX dan EGL menggunakan *open source kernel driver*.

2.1.3. Sistem Operasi Raspberry Pi

Berikut ini adalah daftar sistem operasi yang berjalan pada SBC (*Single Board Computer*) Raspberry Pi.

1. Full OS :
    a. AROS

b. Haiku

c. Linux :

- Android : Android 4.0 (Ice Cream Sandwich)
- Arch Linux ARM
- R_Pi Bodhi Linux
- Debian Squeeze
- Firefox OS
- Gentoo Linux
- Google Chrome OS : Chromium OS
- PiBang Linux
- Raspberry Pi Fedora Remix
- Raspbian (Debian Wheezy port with faster floating point support)
- Slackware ARM (formerly ARMslack)
- QtonPi a cross-platform application framework based Linux
- distribution based on the Qt framework
- WebOS : Open webOS

d. Plan 9 from Bell Labs

e. RISC OS

f. Unix :

- FreeBSD
- NETBSD

2. *Multi-purpose light distributions*:

- Moebius, ARMHF distribusi berdasarkan Debian. Menggunakan repositori Raspbian, cocok di kartu 1 GB microSD. Ini memiliki

layanan hanya minimal dan penggunaan memori yang dioptimalkan untuk menjaga *footprint* kecil.

- Squeezed Arm Puppy, versi Puppy Linux (Puppi) untuk ARMv6 (sap6) khusus untuk Raspberry Pi.

3. *Single-purpose light distributions*:

- IPfire
- OpenELEC
- Raspbmc
- XBMaC
- XBian

Aplikasi berikut dapat dengan mudah diinstal pada sistem operasi Raspbian melalui apt-get.

- Asterisk (PBX), Open source PBX dapat digunakan melalui IP *phones* atau *Wi-Fi softphones*.
- BOINC *clien*t, namun sangat sedikit proyek BOINC memberikan ARM *compatible client* paket *software*.
- Minidlna, DLNA kompatibel *home* LAN *multimedia server*.
- Firefly *Media Server* (*new* RPiForked-Daapd), *server* iTunes kompatibel *open source audio*.

Membangun dari sumber-sumber:

- Firefly *Media Server* (*original* mt-daapd), an iTunes kompatibel *open source audio server*.

2.1.4. Pemograman Raspberry Pi

Raspberry Pi dengan OS (*Operating System*) raspbian atau OS (*Operating System*) yang berbasis Linux lainnya mendukung beberapa pemograman yaitu PERL dan phyton. Phyton merupakan bahasa pemograman utama yang mendukung Raspberry Pi selain bahasa pemograman lainnya seperti C/C++.

2.1.4.1. Bahasa Pemograman PERL

PERL merupakan singkatan dari "*Practical Extraction and Report Language*" adalah sebuah bahasa interpreter sekaligus kompiler. Dalam hal ini PERL akan mendeteksi setiap baris untuk mencari *syntax error* sebelum program dijalankan (*run*). PERL diciptakan oleh Larry Wall pada tahun 1986, awalnya hanya beroperasi pada sistem operasi UNIX, namun saat ini PERL telah diterapkan pada berbagai platform seperti Windows, OS/2, Macintosh dan Linux.

PERL adalah sebuah bahasa pemrograman yang menggabungkan kemampuan dari bahasa C, utilitas sed dan awk bahkan fitur shell. PERL juga mirip dengan bahasa C sehingga programer C pun mampu mempelajari PERL dengan cepat dan mudah. Salah satu contoh kemiripannya adalah adanya tanda titik koma atau *semicolon* ";" pada akhir suatu perintah. Berikut salah satu contoh sederhana program PERL:

```
#!usr/bin/perl

Print "Selamat datang didunia PERL !\n"
```

Simpan file tersebut dengan nama tes.pl, kemudian dalam shell UNIX ubah hak akses file dengan pernyataan:

Chmod 755 tes.pl

Dari shell UNIX, script dapat dijalankan dengan perintah “perl tes.pl”. sehingga akan menampilkan : Selamat datang didunia PERL ! pada layar.

2.1.4.2. Bahasa Pemograman Phyton

Python merupakan bahasa pemrograman yang *freeware* atau perangkat bebas dalam arti sebenarnya, tidak ada batasan dalam penyalinannya atau mendistribusikannya. Lengkap dengan *source code*-nya, *debugger* dan *profiler*, antarmuka yang terkandung di dalamnya untuk pelayanan antarmuka, fungsi sistem, GUI (antarmuka pengguna grafis), dan basis datanya. Python dapat digunakan dalam beberapa sistem operasi, seperti kebanyakan sistem UNIX, PCs (DOS, Windows, OS/2), Macintosh, dan lainnya. Pada kebanyakan sistem operasi linux, bahasa pemrograman ini menjadi standarisasi untuk disertakandalam paket distribusinya.

Tipe data di bahasa pemrograman python dibagi menjadi dua kelompok ialah sebagai berikut.

- Immutable = tipe data yang tidak bisa diubah (*string* dan bilangan)
- Mutable = tipe data yang bisa diubah (*list* dan *dictionary*)

Tidak seperti pemrograman lainnya, variabel pada Python tidak harus dideklarasikan secara eksplisit. Pendeklarasian variabel terjadi secara otomatis ketika kita memberikan sebuah nilai pada suatu variabel. Untuk pemberian nilai, bisa langsung dengan tanda "=". Contohnya sebagai berikut.

```
>>> a = ‘hello”
>>> b = “world”
>>> a+b
‘hello world’
```

```
>>> a = 7
>>> b = 5
>>> a +b
12
>>>
```

Tipe data number ada 4 macam, sebagai berikut.

1. *Plain integer*. *Integer* ini mempunyai range nilai antara -2^32 sampai 2^31 – 1.
2. Long integer. Perhitungan di luar *range* nilai *integer*.
3. *Floating Point Real Number*. Bilangan *real*.
4. *Complex number*. Untuk bilangan *real* dan imajiner.

Berikut ini adalah contoh penulisan tipe data number sebagai berikut.

```
>>> v_i  = 10
>>> v_f  = 24.5
>>> v_c = 3+4j
>>> print v_i
10
>>>print v_f
24.5
>>>print v_c
(3+4j)
>>>
```

Penulisan *string* bisa ditulis dengan cara, sebagai berikut.

1. Diapit dengan tanda petik tunggal '.
2. Diapit dengan tanda petik ganda ".

3. Diapit dengan tiga tanda petik tunggal dan tiga tanda petik ganda ''' atau """.

Berikut ini contoh penulisan *string* pada pemograman phyton.

>>>print ’hello world’

hello world

>>>print “hello world”

hello world

>>>print ‘\’phyton\’ selamat belajar’

‘phyton’ selamat belajar

>>> print “hello world\n belajar phyton”

hello world

belajar phyton

>>>

Berikut ini beberapa *whitespace character* pada pemograman phyton.

- \n = sebagai garis baru
- \r = untuk menghapus string sebelumnya
- \t = untuk tab (tabulation)
- \v = untuk vertical tab
- \e = escape
- \f = karakter sesudahnya pada garis baru
- \b = backspace
- \a = bell

Berikut ini contoh pemanfaatan variable pada pemograman phyton.

#!/usr/bin/phyton

```
nama = raw_input(“Siapakah namamu?\n”)
print “_____”
print “hello “+ nama + “ , selamat belajar phyton”
```

Maka hasil keluarannya adalah sebagai berikut.

```
Siapa namamu?
Toda
_____
Hello Toda, selamat belajar phyton
```

2.2. *Bluetooth*

Bluetooth adalah sebuah teknologi nirkabel dengan gelombang radio yang bekerja pada frekuensi 2.4 GHz. *Bluetooth* menggunakan sistem *Frequency Hopping Spread Spectrum* (FHSS) yang mempunyai kecepatan maksimum 1 Mbps.

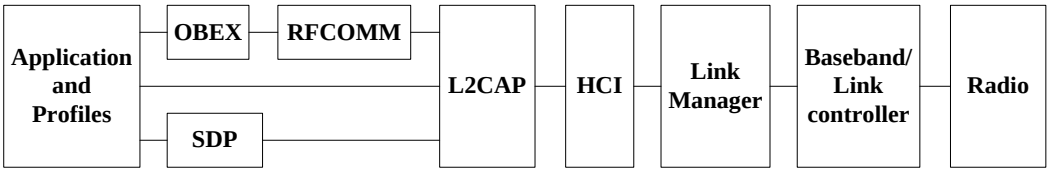


Gambar 2. 3 Logo *Bluetooth*

Bluetooth terbagi menjadi 2 kelas yaitu kelas 1 yang mempunyai jangkauan maksimum kurang lebih 100 m dan kelas 2 yang mempunyai jangkauan maksimum 15 meter. pada awalnya teknologi *Bluetooth* dipromosikan untuk penggunaan LAN. Namun, mengingat jangkauan maksimum yang tidak terlalu luas, *Bluetooth* kemudian dipromosikan untuk penggunaan dalam *personal area network* (PAN).

2.2.1. Arsitektur *Bluetooth*

Bluetooth memiliki lapisan-lapisan yang mempunyai fungsi berbeda sehingga secara umum digambarkan seperti Gambar 2. 1 di bawah ini.



Gambar 2. 4 Arsitektur *Bluetooth*

1. *Profiles* adalah aplikasi yang dibuat pada arsitektur *Bluetooth*.
2. *Object Exchange* (OBEX) adalah protokol yang berfungsi untuk melakukan transfer objek antar aplikasi yang berjalan dengan menggunakan *Bluetooth*.
3. *Service Discovery Protocol* (SDP) merupakan sebuah protokol yang berfungsi untuk memberikan layanan pencarian *service* terhadap *Bluetooth device* yang berada disekitarnya.
4. RFCOMM adalah protokol yang mempunyai fungsi untuk melakukan emulasi koneksi serial antar *device Bluetooth*.
5. *Logical Link Control and Adaption Protocol* (L2CAP) merupakan suatu lapisan dalam arsitektur *Bluetooth* yang dapat membuat paket-paket yang siap untuk digunakan oleh lapisan-lapisan yang berada di atasnya.
6. *Human Computer Interface* (HCI) adalah *layer* antar muka antara perangkat keras *Bluetooth* dengan perangkat lunak atau *platform*.
7. *Link Manager* (LM) adalah protokol yang berfungsi untuk melakukan membangun kanal hubungan antara *device Bluetooth* setelah melakukan

mekanisme sistem keamanan serta melakukan negosiasi berapa besar paket *baseband* yang akan dikirimkan.

8. *Layer baseband* mempunyai fungsi untuk melakukan sinkronisasi pengiriman data antar *device Bluetooth*.
9. Radio mempunyai fungsi untuk membuat gelombang radio yang mempunyai frekuensi 2.4 GHz sesuai dengan data yang akan diterima maupun dikirimkan.

2.2.2. *Security Bluetooth*

Sistem keamanan pada *Generic Access Bluetooth* dibagi menjadi 3 *mode*, yaitu tanpa keamanan (*mode* 1), keamanan pada tingkat *service* (*mode* 2) serta keamanan pada tingkat *link* komunikasi (*mode* 3). Faktor-faktor yang membangun sistem keamanan pada *level link* komunikasi adalah sebagai berikut.

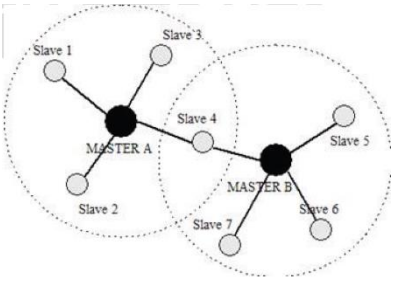
1. *Bluetooth Device Address* (BD_ADDR) yang mempunyai 48 bit yang unik untuk semua peralatan *Bluetooth*.
2. *Private Authentication Key* yang mempunyai panjang 128 bit dan dibuat secara acak.
3. *Privat Encryption Key* yang mempunyai panjang 8-128 bit yang digunakan untuk enskripsi.
4. *Random Number* (RAND) yang mempunyai panjang 128 bit yang dibuat oleh *device Bluetooth*.

Perbedaan sistem keamanan pada mode 2 dan mode 3 adalah pada keamanan mode 3 *device Bluetooth* melakukan inisiasi prosedur keamanan sebelum kanal komunikasi terjalin.

2.2.3. Topologi jaringan *Bluetooth*

Pada dasarnya fungsi *Bluetooth* adalah untuk menggantikan jaringan menjadi nirkabel, namun *Bluetooth* mempunyai keterbatasan dalam segi perangkat keras apabila ingin menyamai kinerja dari jaringan kabel.

Berikut ini adalah gambar topologi jaringan komunikasi *Bluetooth*.



Gambar 2. 5. Topologi Jaringan Komunikasi *Bluetooth*

2.3. Protokol

Protokol merupakan persetujuan antara dua entitas atau lebih tentang bagaimana entitas-etintas tersebut akan berkomunikasi satu dengan yang lainnya. Persetujuan –persetujuan yang dimaksud antara lain:

- Bagaimana entitas-entitas akan memulai komunikasi atau mengakhiri komunikasi.
- Proses sinkronisasi antara pengirim dan penerima.
- Deteksi dan perbaikan kesalahan.
- Format dan *encoding* data.

2.3.1. Elemen Protokol

Sebuah protokol memiliki lima bagian terpisah. Untuk menjadi lengkap, seluruh bagian ini harus dimasukkan ke dalam sebuah protokol:

1. Layanan yang disediakan oleh sebuah protokol.
2. Asumsi mengenai ruang lingkup protokol.
3. *Vocabulary* atau kosakata dari suatu pesan yang digunakan dalam protokol.
4. Format atau susunan dari pesan.
5. Prosedur dan aturan-aturan yang menjaga agar pertukaran pesan tetap konsisten.

2.3.2. Layanan dan Ruang Lingkup Protokol

Untuk memudahkan perancangan, protokol biasanya dibagi menjadi beberapa tahapan atau layer. *Layer-layer* yang ada ini juga bertujuan mengurangi kompleksitas dalam memecahkan masalah-masalah yang akan timbul ketika dua atau lebih komputer saling berkomunikasi melalui jaringan.

Pada masing masing layer, terdapat beberapa protokol yang bertujuan untuk memecahkan masalah yang ditangani oleh layer tersebut. Gabungan dari layer dan protokol dari masing-masing *layer* biasanya disebut hirarki protokol atau model. Jumlah layer maupun protokol yang digunakan pada masing – masing layer tergantung dari model protokol yang digunakan.

2.3.3. *Vocabulary* dan Format Protokol

Tiga metode *formating* yang biasa digunakan saat ini adalah :

1. *Bit-oriented*.

Protokol *bit-oriented* mengirimkan data sebagai suatu aliran dari *bit-bit*. Supaya penerima dapat mengenali dimana awal atau akhir dari suatu paket, digunakan *flag* atau penanda yang terdiri dari sekumpulan *bit* yang unik diawal dan diakhir data *user*. Penanda ini biasa juga disebut *control codes*. Jadi biasanya dalam suatu paket data *bit-oriented*, terdapat data user dan diapit *control codes*.

2. *Character-oriented*

Pada protokol jenis ini, ditentukan panjang dari sebuah karakter, biasanya 7 atau 8 *bit*. Penerima akan membaca *bit-bit* sampai 7 atau 8 *bit*, baru kemudian menerjemahkannya ke dalam karakter-karakter baik berupa data *user* atau *control codes*. Contoh *control codes* yang biasa digunakan sebagai penanda awal dari paket *character-oriented* adalah STX (*Start of Text*) dan akhir paket adalah ETX (*End of Text*).

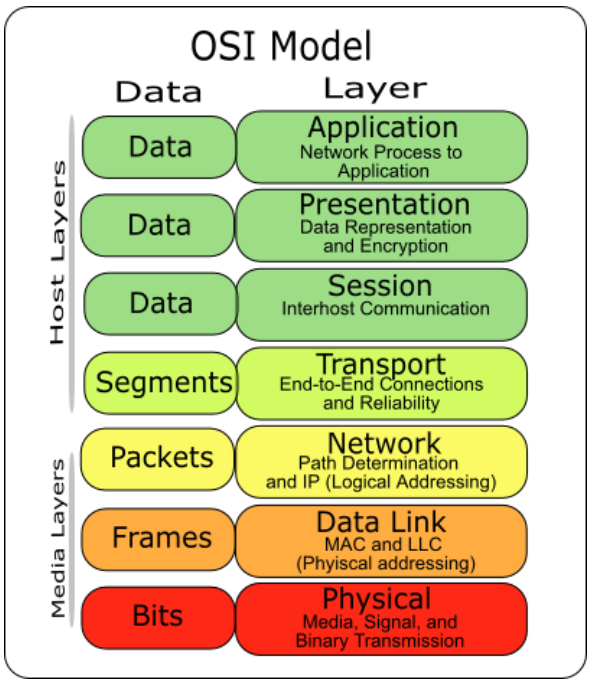
3. *Byte-count-oriented*

Sedikit berbeda dengan *character-oriented* dan *bit-oriented*, *byte-count-oriented* menambahkan sejumlah data setelah STX (*Start of Text*). Data tersebut berupa panjang dari *data user* yang ada dalam paket tersebut. Dengan demikian, ETX (*End of Text*) sudah tidak diperlukan lagi. Saat ini protokol ini adalah protokol yang paling umum digunakan.

2.4. OSI (*Operating System Interconnection*) *Layer*

Model referensi OSI terdiri dari 7 lapisan, mulai dari lapisan fisik hingga aplikasi. Model referensi ini tidak hanya berguna untuk produk-produk LAN saja, tetapi juga sangat diperlukan dalam membangun jaringan Internet. OSI menjelaskan bagaimana data dan informasi jaringan berkomunikasi dari sebuah

aplikasi pada sebuah komputer berjalan melalui jaringan, menuju ke aplikasi di komputer lain. OSI menjelaskan melalui pendekatan pemecahan menjadi lapisan-lapisan (*layer*). Analogi konsep *layer* adalah seperti dalam departemen/bidang dalam sebuah perusahaan, setiap departemen memiliki tugas yang berbeda, dan hanya terfokus padahal tertentu sesuai pembagian tugas.



Gambar 2. 6 OSI Layer

Tabel 2. 3 OSI Layer

Layer	Nama Layer	Keterangan
7	Application	Berfungsi sebagai antarmuka antara aplikasi dengan fungsionalitas jaringan, mengatur bagaimana aplikasi dapat mengakses jaringan, dan kemudian membuat pesan-pesan kesalahan.
6	Presentation	Berfungsi untuk mentranslasikan data yang hendak ditransmisikan oleh aplikasi ke dalam format yang dapat ditransmisikan oleh aplikasi ke dalam format yang dapat ditransmisikan melalui jaringan.

5	Session	Berfungsi untk mendefinisikan bagaimana koneksi dapat dibuat, dipelihara, atau dihancurkan. Selain itu, di level ini juga dilakukan resolusi nama.
4	Transportation	Berfungsi untuk memecah data ke dalam paket-paket data serta memberikan nomor urut ke paket-paket tersebut sehingga dapat disusun kembali pada sisi tujuan setelah diterima. Selain itu, pada level ini juga membuat sebuah tanda bahwa paket diterima dengan sukses (*acknowledgement*), dan mentransmisikan ulang terhadap paket-paket yang hilang ditengah jalan.
3	Network	Berfungsi untuk mendefinisikan alamat-alamat IP, membuat *header* untuk paket-paket, dan kemudian melakukan melalui *internetworking* dengan menggunakan *router* dan *switch layer*-3.
2	Data Link	Berfungsi untuk menentukan bagaimana bit-bit data dikelompokkan menjadi format yang disebut sebagai *frame*. Selain itu, pada level ini terjadi koreksi kesalahan, *flow control*, pengalamatan perangkat keras (seperti halnya *Media Acces Control Address* (MAC Address)), dan menentukan bagaimana perangkat-perangkat jaringan seperti hub, bridge, repeater, dan switch layer-2 beroperasi. Spesifikasi IEEE 802, membagi level ini menjadi dua level anak, yaitu lapisam *Logical Link Control* (LLC) dan lapisan *Media Access Control* (MAC).
1	Physical	Berfungsi untuk mendefinisikan media transmisi jaringan, metode pensinyalan, sinkronasi bit, arsitektur jaringan, topologi jaringan dan pengkabelan. Selain itu, level ini juga

		mendefinisikan bagaimana *Network Interface Card* data berinteraksi dengan media kabel atau radio

2.4.1. Enkapsulasi dan Dekapsulasi Data

Enkapsulasi adalah suatu proses untuk menyembunyikan atau memproteksi sustu proses dari kemungkinan interferensi atau penyalahgunaan dari luar sistem sekaligus menyederhanakan penggunaan sistem itu sendiri, juga membuat satu jenis paket data jaringan menjadi jenis data lainnya. Enkapsulasi terjadi ketika sebuah protokol yang berada pada layer yang lebih rendah menerima data dari protokol yang berada pada layer yang lebih tinggi dan meletakkan data ke format data yang dipahami oleh protokol tersebut. Akses ke internal sistem diatur sedemikian rupa melalui seperangkat *interface*.

Proses enkapsulasi data:

1. Awalnya data dibuat, ketika memulai proses pengiriman, data turun melalui *Application layer* (*layer* 7) yang bertanggung jawab dalam pertukaran informasi dari komputer ke jaringan, pada dasarnya *layer* ini merupakan *interface* antara jaringan dengan aplikasi yang digunakan pengguna. *Layer* ini juga dapat berfungsi untuk mendefinisikan permintaan dari pengguna. Kemudian data diteruskan ke *layer Presentation* (*layer* 6), yang mana layer ini bertanggung jawab dalam menentukan perlu tidaknya melakukan enkripsi terhadap permintaan ini ataupun ke bentuk lain dari translasi data. Contohnya kata ”Teknik Elektro”, kata tersebut secara umum akan diubah ke dalam bahasa ASCII.

Dalam tabel ASCII karakter ”T”= 0x54, karakter ”e”= 0x65, karakter ”k”= 0x6B, karakter ”n”= 0x6E, karakter ”i”= 0x69, karakter SP (spasi)= 0x20, karakter ”E”= 0x45, dan seterusnya. Jika proses sudah lengkap, selanjutnya hasil enkripsi tersebut dilanjutkan ke *Session layer* (*layer* 5) yang mana layer ini akan memeriksa apakah aplikasi meminta suatu informasi dan memverifikasi layanan yang diminta pada *server*. Setiap informasi yang akan dilewatkan akan ditambahkan *header* setiap turun 1 *layer*.

2. Setelah melewati *layer* 5, data masuk *Transport layer* (*layer* 4), pada *layer* ini data dipastikan sudah mempunyai suatu koneksi dengan *server* dan memulai proses mengubah informasi ke dalam bentuk segmen. Pengecekan kesalahan (*error*) dan penggabungan data serta keutuhan data yang berasal dari aplikasi yang sama dilakukan di *layer transport* ini. Header yang dimasukkan pada data tersebut berfungsi untuk menentukan kecocokan data yang dikirimkan, karena data yang akan dikirimkan dibagi ke dalam bentuk segmen dan paket. Pada saat data dikirimkan setiap bagian data tersebut memiliki header, apabila header sesuai maka data tersebut diambil dan bagian paket yang tidak sesuai akan diabaikan.
3. Data yang telah melewati *layer* 4 diteruskan ke *Network layer* (*layer* 3), di *layer* ini segmen-segmen data diterima dan ditambahkan alamat *network* untuk station yang me-*request* dan alamat *network* untuk *server* yang di-*request*. Segmen-segmen tersebut akan diubah menjadi paket-paket, kemudian *layer Network* membuat *header Network*, dimana didalamnya

terdapat juga alamat *layer network*, dan ditempatkan L4PDU dibaliknya, dan terebentuklah L3PDU.

4. Paket-paket kemudian dilewatkan ke *layer Data Link* (*layer* 2) dan paket-paket diatur dan kemudian akan dibungkus ke dalam individual frame, salah satu contoh dalam proses ini adalah memberikan alamat MAC tujuan dan MAC *address* sumber yang kemudian informasi tersebut digunakan untuk membuat *trailer*. Peran MAC *address* penting dalam mengirimkan paket antara satu router ke router lainnya. Kemudian akan ditransmisikan ke media. *Data link layer* bertanggung jawab untuk mengirimkan frame menurut topologi yang digunakan. Terbentuklah L2PDU pada proses ini.
5. Pada tahap terakhir pada *layer Physical* (*layer* 1), informasi akan dibawa dari *source* menuju *destination*. Karena *Physical layer* tidak mengenal *frame*, maka melewatkan informasi itu ke bentuk *bit*. *Bit-bit* tersebut berasal dari hasil enkripsi yang dilakukan pada layer *Presentation* yang diubah ke dalam bentuk bit. Karakter "T' dienkripsi menjadi 0x54 pada layer *Presentation* dan diubah kedalam bentuk biner menjadi 01010100 dan seterusnya. Bit-bit tersebut akan disinkronisasi dan kemudian diubah menjadi sinyal listrik yang berupa tinggi rendahnya tegangan dan selnajutnya ditransmisikan melalui media.

Pada tiap layer terdapat LxPDU (*Layer* N *Protocol Data Unit*), dimana merupakan bentuk dari *byte* pada *header-trailer* pada data. PDU merupakan proses-proses pada setiap *layer* dari model OSI. Pada tiap-tiap *layer* juga terbentuk bentukan baru, pada *layer* 2 PDU termasuk *header* dan *trailer* disebut bentukan *frame*. Pada *layer* 3 disebut paket (*packet*). Sedangkan pada *layer* 4 disebut segmen

(*segment*). Setelah dilakukan proses enkapsulasi, lalu dikirimkan ke *server* dan *server* akan melakukan proses tadi secara terbalik, yaitu dari *Physical layer* ke *Application layer*, proses ini disebut dekapsulasi. Jika pada enkapsulasi dilakukan pembungkusan, maka pada de-enkapsulasi akan melakukan pembukaan dari bungkus-bungkus tadi melalui *layer-layer* nya. Proses Dekapsulasi merupakan kebalikan dari proses Enkapsulasi. Proses Enkapsulasi merupakan proses pembungkusan data sedangkan Dekapsulasi merupakan proses pembukaan bungkus. Prosesnya pun terbalik dari proses Enkapsulasi. Proses Enkapsulasi bermula dari *layer* yang paling atas (*Application Layer*) hingga *layer* yang paling bawah (*Physical Layer*) sedangkan proses Dekapsulasi bermula dari layer yang paling bawah (*Physical Layer*) menuju *layer* yang paling atas (*Application Layer*).

2.5. TCP/IP (*Transport Control Protocol/Internet Protocol*)

TCP/IP (*Transport Control Protocol/Internet Protocol*) adalah sekumpulan protokol yang didesain untuk melakukan fungsi-fungsi komunikasi data pada *Wide Area Network* (WAN). TCP/IP terdiri dari sekumpulan protokol yang masing-masing bertanggung jawab atas bagian-bagian tertentu dari komunikasi data. Berkat prinsip ini, tugas masing-masing protokol menjadi jelas dan sederhana. Protokol yang satu tidak perlu mengetahui cara kerja protokol yang lain, sepanjang masih bisa saling mengirim dan menerima data. Berkat penggunaan prinsip ini, TCP/IP menjadi protokol komunikasi data yang fleksibel. TCP/IP dapat diterapkan dengan mudah di setiap jenis komputer dan *interface* jaringan, karena sebagian besar isi kumpulan protokol ini tidak spesifik terhadap satu komputer atau peralatan jaringan tertentu. Agar TCP/IP dapat berjalan diatas *interface* jaringan

tertentu, hanya perlu dilakukan perubahan pada protokol yang berhubungan dengan *interface* jaringan saja.

2.6. IP *Address*

Alamat IP adalah alamat yang diberikan ke jaringan dan peralatan jaringan yang menggunakan protokol TCP/IP.

2.6.1. Format IP *Address*

Jika dilihat dari bentuknya, alamat IP terdiri atas 4 buah bilangan biner 8 bit yang dipisahkan dengan tanda titik. Nilai terbesar dari bilangan biner 8 bit adalah 255 ($=2^7 +2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 +1$). Karena alamat IP terdiri atas 4 buah bilangan 8 bit, maka jumlah IP yang tersedia adalah 255 x 255 x 255 x 255. Contoh alamat IP adalah 192.168.0.1.

Untuk mempermudah proses pembagiannya, alamat IP dikelompokkan dalam kelas-kelas. Dasar pertimbangan pembagian alamat IP ke dalam kelas kelas adalah untuk memudahkan pendistribusian pendaftaran alamat IP. Alamat IP ini dikelompokkan dalam tiga kelas, yaitu Kelas A, Kelas B, dan Kelas C Perbedaan pada tiap kelas tersebut adalah pada ukuran dan jumlahnya. IP Kelas A dipakai oleh sedikit jaringan namun jaringan ini memiliki anggota yang besar. Kelas C dipakai oleh banyak jaringan, namun anggota masing masing jaringan sedikit.

Pembagian kelas-kelas alamat IP didasarkan pada dua hal yaitu *network* ID dan *host* ID. Setiap alamat IP merupakan sebuah pasangan dari *network* ID (identitas jaringan) dan *host* ID (identitas *host* dalam jaringan tersebut). *Network* ID adalah bagian dari alamat IP yang digunakan untuk menunjukkan jaringan tempat komputer ini berada, sedangkan *host* ID adalah bagian dari alamat IP yang digunakan untuk menunjukkan *workstation*, *server*, *router*, dan semua *host* TCP/IP

lainnya dalam jaringan tersebut. Dalam satu jaringan, *host* ID haruslah unik (tidak boleh ada yang sama). Jumlah kelompok angka yang termasuk *network ID* dan berapa yang termasuk *host ID* tergantung kepada kelas dari alamat IP yang dipakai.

Tabel 2. 4 Jenis IP *Address*

Kelas	Network ID	Host ID	Default Subnet Mask
A	W	x.y.z	255.0.0.0
B	w.x	y.z	255.255.0.0
C	Wx.y	Z	255.255.255.0

Untuk dapat menandai kelas satu dengan kelas yang lain, maka dibuat beberapa peraturan sebagai berikut.

- Oktet pertama dari kelas A harus dimulai dengan angka binari 0.
- Oktet pertama dari kelas B harus dimulai dengan angka binari 10.
- Oktet pertama dari kelas C harus dimulai dengan angka binari 110.

Oleh sebab itu, alamat IP dari masing-masing kelas harus dimulai dengan angka desimal tertentu pada oktet pertama.

Tabel 2. 5 Class dan Jumlah Max Network

Kelas	Range	Jumlah Maksimum Network
A	1.xxx.xxx.xxx – 126.xxx.xxx.xxx	127
B	128.0.xxx.xxx – 191.255.xxx.xxx	16384
C	192.0.0.xxx – 223.255.255,xxx	2097152

Di samping itu terdapat beberapa aturan dasar dalam menentukan *network* ID dan *host* ID yang hendak digunakan. Aturan tersebut sebagai berikut.

- Angka 127 di oktet pertama digunakan untuk loopback (menunjuk komputer itu sendiri).

- *Network ID* dan *host ID* tidak boleh semuanya terdiri atas angka 0 atau 1.
- *Host* ID harus unik dalam satu *network*, tidak boleh ada dua jaringan yang memiliki *host* ID yang sama.

Bit-bit dari *network ID* maupun *host ID* tidak boleh semuanya berupa angka binari 0 atau 1. Apabila semua *network ID* dan *host ID* semuanya berupa angka binari 1 yang dapat ditulis sebagai 255.255.255.255, maka alamat ini disebut *flooded broadcast*. Jika *host ID* semua berupa angka binari 0, alamat IP ini menyatakan alamat *network* dari jaringan yang bersangkutan. Jika *host ID* semuanya berupa angka binari 1, maka alamat IP ini ditujukan untuk semua *host* di dalam jaringan yang bersangkutan, yang dipergunakan untuk mengirim pesan (*broadcast*) kepada semua *host* yang berada didalam jaringan lokal.

Di dalam Internet, alamat IP digunakan untuk memberikan alamat pada suatu situs. Agar pemakaian alamat IP ini seragam di seluruh dunia, maka ada sebuah badan internasional yang mengatur pemberian alamat IP yang bernama *Internet Assign Number Authority* (IANA) untuk digunakan di Internet. Dalam pemberian alamat IP, IANA hanya memberikan alamat IP dengan *network ID* saja, sedangkan *host ID*-nya diatur oleh pemilik alamat IP tersebut.

Disamping itu, IANA juga menyediakan beberapa alamat IP khusus yang disebut *private address*. *Private address* biasanya digunakan server *Network Address Translation* (NAT), atau *proxy server* untuk memberikan konektivitas *host-host* ke *public network*. Dalam hal ini, *traffic-traffic* masuk pada *address-address* dalam *range private address* tidak akan di*route* dalam Internet. *Private address* sering juga digunakan oleh *host-host* yang tidak terhubung dengan Internet, atau mereka yang tidak memiliki ketersediaan *public addres* yang mencukupi.

Yang sekarang masih umum digunakan merupakan IPv4 yang terdiri dari 32 bit. Hal ini menyebabkan IP ini akan habis bila jumlah komputer atau *device* yang memerlukan IP mencapai 2_{32} atau 4.294.967.296. IANA menyimpan beberapa alamat IP sebagai alamat IP untuk jaringan *Private* (RFC 1918) yaitu:

- 10.0.0.0 - 10.255.255.255
- 172.16.0.0 - 172.31.255.255
- 192.168.0.0 - 192.168.255.255

2.7. Wi-Fi

Wi-Fi merupakan singkatan dari *Wireless Fidelity*, memiliki pengertian yaitu sekumpulan standar yang digunakan untuk Jaringan Lokal Nirkabel (*Wireless Local Area Network*, WLAN) yang didasari pada spesifikasi *The Institute of Electrical and Electronics Engineer* (IEEE) 802.11. *Wi-Fi* (atau WLAN) diciptakan terperinci untuk mengoperasikan sebagai *ethernet* tanpa kawat. Ini adalah satu teknologi *open-standard* yang memperbolehkan koneksi nirkabel di antara area alat-alat perlengkapan dan lokal jaringan. Akses publik jasa WLAN didesain untuk menghubungkan dan melayani jaringan LAN dengan jarak radius 50 sampai 150 meter dari titik akses.



Gambar 2. 7 Logo Wi-Fi (Wireless Fidelity)

Wi-Fi dirancang berdasarkan spesifikasi IEEE 802.11. Sekarang ini ada empat variasi dari 802.11, yaitu: 802.11a, 802.11b, 802.11g dan 802.11n. Spesifikasi *b* merupakan produk pertama *Wi-Fi*.

Tabel 2. 6 Spesifikasi Wi-Fi

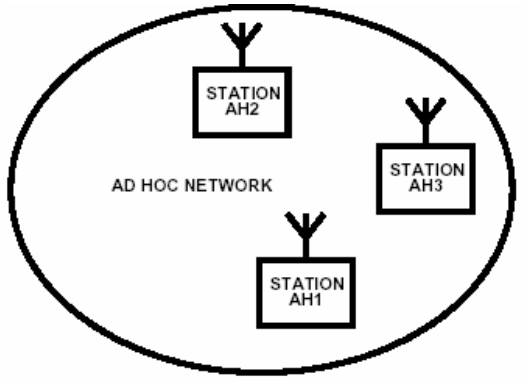
Spesifikasi	Kecepatan	Frekuensi Band	*Support*
802.11b	11 Mbps	~2.4 Ghz	b
802.11a	54 Mbps	~5 Ghz	a
802.11g	54 Mbps	~2.4 Ghz	b, g
802.11n	100 Mbps	~2.4 Ghz	b, g, n

2.7.1. Topologi Jaringan Wi-Fi

Topologi dasar dari jaringan *Wi-Fi*, bentuk paling sederhana terdiri dari dua atau lebih *wireless node*, atau stasiun (STA), yang terhubung satu sama lain dan mendirikan sebuah komunikasi.

1. *Independent Basic Service Set* (IBSS)

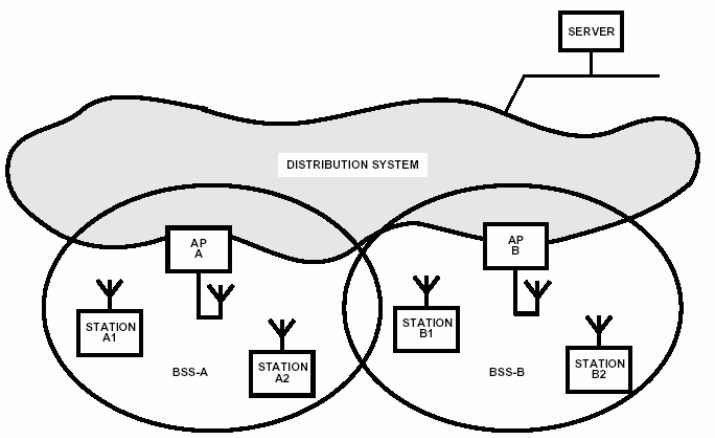
Pada satu IBSS, STA merupakan komunikasi pada taraf *peer-to-peer*. Jenis jaringan ini dikenal sebagai jaringan *ad hoc*. Jaringan *ad hoc* memiliki pengaturan yang fleksibel dan hemat biaya, meliputi *hard-to-wire* lokasi komunikasi dari beberapa perangkat, seperti laptop pada satu ruang konferensi.



Gambar 2. 8. Jaringan *Ad hoc*

2. *The Extended Service Set* (ESS)

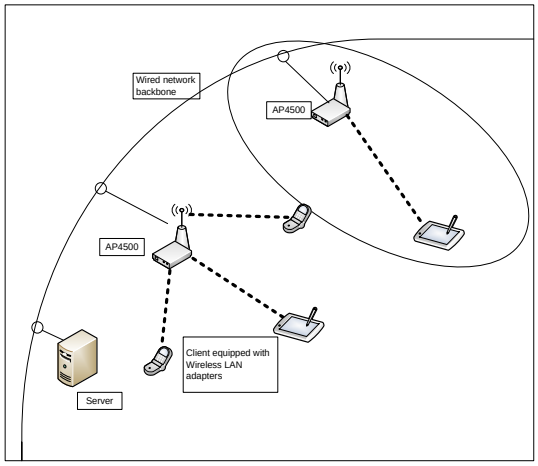
Terdiri dari satu rangkaian BSS (masing-masing mengandung satu AP) dihubungkan dengan suatu Sistem Distribusi (DS). Walau DS dapat dari jenis jaringan apapun (meliputi jaringan nirkabel), ini tanpa alternatif *Ethernet* LAN. Pada ESS, STA (Statsiun) dapat menjelajahi dari suatu BSS ke BSS yang lain dan berkomunikasi dengan perangkat *mobile* atau *fixed client*. ESS memperbolehkan memperluas cakupan, baik berada di luar jangkauan dari satu WLAN radio. Dengan mempergunakan ESS, *campus-wide coverage* dapat dilakukan. Jasa ini biasanya dikenal sebagai mode infrastruktur.



Gambar 2. 9. Mode Infrastruktur

2.7.2. Komponen Jaringan Wi-Fi

802.11 LAN berlandaskan arsitektur selular dimana sebuah sistem membagi-bagi ke dalam sel, dimana masing-masing sel (*Basic Service Set*-BS) dikontrol oleh base tersebut. Stasiun (*Access Point* atau AP).



Gambar 2. 10. Komponen Jaringan

Ada tiga hubungan terkait utama di dalam rangkai WLAN yang itu bentuk basis dari jaringan, yaitu:

1. *Access Point* (Titik Akses/AP)

Suatu titik akses mengoperasikan di antara spektrum frekuensi yang spesifik dan penggunaan 802.11 standar menetapkan teknik modulasi. Itu juga memberikan informasi kepada *client* membuktikan keaslian dan menghubungkan *client* ke jaringan.

Sebuah AP (*access point*) juga mengkoordinasi *client* dari sumber daya yang terhubung. Titik akses umumnya mempunyai dua tugas utama:

- Sebagai sebuah stasiun dasar ke pengguna.
- Sebagai suatu jembatan di antara jaringan nirkabel dan jaringan yang terhubung melalui kabel.

Titik akses merupakan sebuah lapisan penghubungan data (*Data Link Layer*), dan mendukung kecepatan koneksi 1, 2, 5. 5, atau 11 Mbps bergantung kepada standar penerapan.

Area cakupan dari AP sampai 375 kaki atau berjarak 114 meter. Angka pengguna sebuah dukungan titik akses umumnya sekitar 60 - 200 pengguna.

2. *Network interface card* (NIC)/*Client Adapter*

Sebuah PC (*Personal Computer*) atau *workstation* mempergunakan sebuah *wireless* NIC untuk menghubungkan ke jaringan. NIC meneliti spektrum frekuensi tersedia dijaringan dan pengguna dihubungan ke sebuah titik akses atau pengguna yang lain.

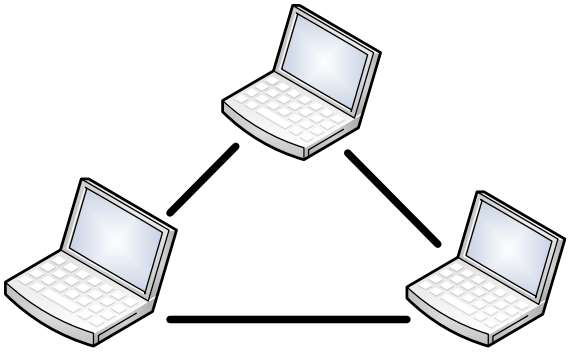
Sebuah NIC dapat digunakan pada sebuah PC (*Personal Computer*) atau sebuah *operating system* dengan menggunakan sebuah *driver*.

2.7.3. Tata Letak Jaringan *Wi-Fi*

Wi-Fi dapat diterapkan dalam beberapa cara, tergantung kepada kompleksitas diinginkan. Umumnya, *Wi-Fi* memiliki 3 cara penerapan, yaitu:

1. *Peer-To-Peer*

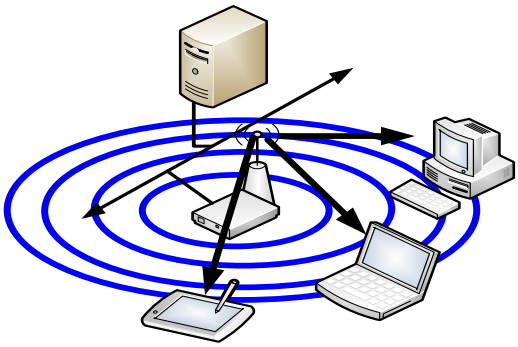
Sebuah adalah jaringan *peer-to-peer* merupakan bentuk komunikasi jaringan *Wi-Fi* yang paling sederhana. Terdiri dari 2 buah PC atau Laptop yang dilengkapi sebuah *wireless adaptor card* atau modul *Wi-Fi* dapat berkomunikasi dengan bentuk jaringan *peer-to-peer*. Dengan jaringan *peer-to-peer* pengguna dapat membagi data yang ada di dalam perangkat masing-masing kepada pengguna yang lain.



Gambar 2. 11. Jaringan Peer-To-Peer

2. *Client & Access Point*

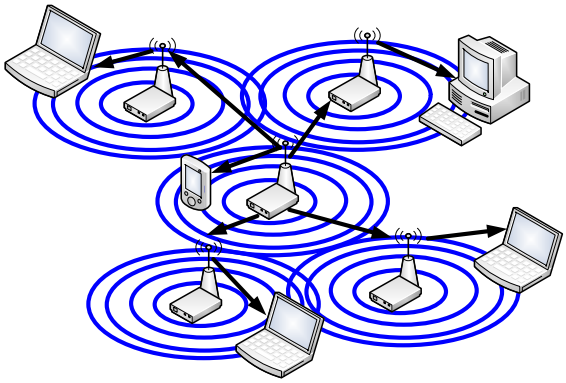
Pada jaringan ini, pengguna dapat terkoneksi ke jaringan melalui titik akses yang terhubung ke LAN *backbone*. Pada umumnya dalam bentuk jaringan ini dapat menampung 15-50 pengguna.



Gambar 2. 12. Client & Access Point

3. *Multiple Access Points*

Bentuk jaringan ini umumnya digunakan dalam radius atau zona akses yang cukup luas, salah satunya *hostspot* pada sebuah kampus. Jaringan ini memerlukan menempatakan beberapa titik akses pada titik-titik tertentu agar jaringan tersebut memiliki cakupan yang luas.



Gambar 2. 13. Multiple Access Point