

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Wireless LAN**

Seiring perkembangan teknologi dan kebutuhan untuk akses jaringan *mobile* (bergerak) yang tidak membutuhkan kabel sebagai media transmisi, maka muncul *Wireless Local Area Network (Wireless LAN/WLAN)*. Jaringan WLAN adalah suatu jaringan area lokal nirkabel yang menggunakan gelombang radio sebagai media transmisi, link terakhir yang digunakan adalah nirkabel, untuk memberi sebuah koneksi jaringan ke seluruh pengguna dalam area sekitar. *Wireless LAN* dapat mengirim dan menerima data melalui media udara dengan menggunakan teknologi frekuensi radio. WLAN juga memiliki berbagai kelebihan yang ditawarkan, antara lain:

1. Mobilitas : pengguna dapat mengakses jaringan WLAN dimanapun selama masih berada dalam *coverage* jaringan WLAN.
2. Kemudahan Instalasi : instalasi sistem WLAN bisa cepat dan sangat mudah dibandingkan dengan menggunakan kabel karena perangkat yang digunakan tidak terlalu banyak dan mudah dikonfigurasi.
3. Fleksibilitas Tempat : teknologi *wireless* memungkinkan suatu jaringan untuk bisa mencapai tempat-tempat yang tidak dapat dicapai dengan jaringan kabel.
4. Skalabilitas : Jaringan WLAN dapat dikonfigurasi dengan beberapa bentuk topologi tergantung kebutuhan pengguna.

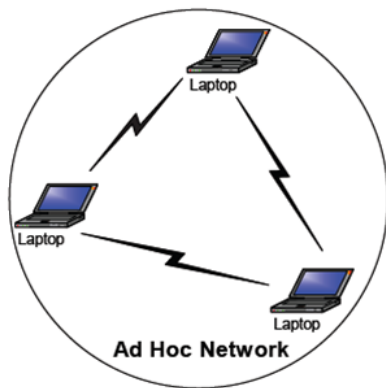
Berdasarkan kemudahan yang didapat dengan menggunakan teknologi WLAN, pengguna dapat pula mempertimbangkan kelemahan yang ada pada teknologi tersebut dalam implementasinya dimana terdapat pengaruh interferensi radio dan halangan akibat bangunan maupun pohon dan lain-lain. (Gede Sukadarmika, 2010).

#### **2.2 Topologi WLAN**

*Wireless LAN* memungkinkan dua bentuk koneksi, yang dikenal sebagai *AdHoc* dan mode *Infrastructure*.

### 2.2.1 Mode *AdHoc*

Mode *AdHoc* yang konfigurasinya ditunjukkan pada Gambar 2.1 merupakan konfigurasi WLAN *peer-to-peer*, dimana dua atau lebih perangkat dapat terkoneksi melalui jaringan *wireless* tanpa memerlukan *access point*. Pada arsitektur ini, perangkat akan mengkonfigurasi sendiri pada kanal radio yang sama untuk mengaktifkan komunikasi *peer-to-peer* (Singh, 2009)

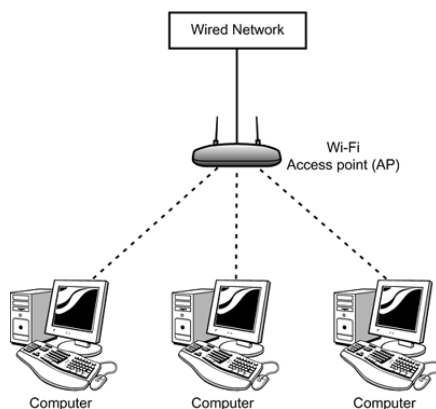


**Gambar 2.1** Topologi Mode *AdHoc* (Sumber: Tri Arianto, 2009)

### 2.2.2 Mode *Infrastructure*

Model *infrastructure* adalah kondisi suatu jaringan dengan menggunakan suatu titik pusat yaitu *access point*, yang berfungsi untuk melayani komunikasi pada jaringan *wireless*, dimana *access point* tersebut terhubung dengan suatu *backbone (distribution system)*.

Untuk topologi infrastruktur seperti yang ditunjukkan pada Gambar 2.2, tiap PC mengirim dan menerima data dari sebuah titik akses, yang dipasang di dinding atau langit-langit berupa sebuah kotak kecil berantena. Saat titik akses menerima data, ia akan mengirimkan kembali sinyal radio tersebut (dengan jangkauan yang lebih jauh) ke PC yang berada di area cakupannya, atau dapat mentransfer data melalui jaringan *Ethernet* kabel. Titik akses pada sebuah jaringan infrastruktur memiliki area cakupan yang lebih besar. (Tri Arianto, 2009)



**Gambar 2.2** Topologi Mode *Infrastructure* (Sumber : eTutorials.org)

### 2.3 Model TCP/IP

Arsitektur protocol *Transmission Control Protocol / Internet Protocol* (TCP/IP) merupakan hasil dari penelitian dan pengembangan protokol yang dilakukan pada jaringan percobaan *packet switched* dan secara umum ditujukan sebagai satu *set* protokol TCP/IP. *Set* protokol ini terdiri atas sekumpulan besar protokol yang telah diajukan sebagai standar internet oleh *Internet Architecture Board*. (Aldya Dwiki, 2015). Model TCP/IP terdiri atas empat *layer* yaitu:

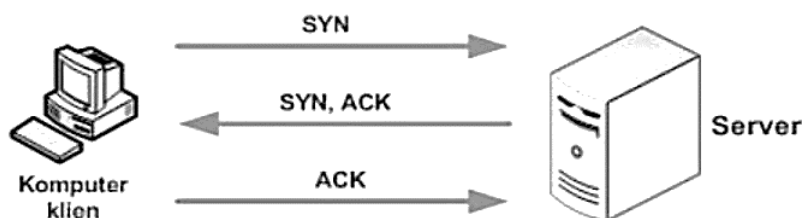
1. ***Application Layer***, merupakan *layer* program aplikasi yang menggunakan protokol TCP/IP. Beberapa diantaranya adalah: *Telnet*, *FTP (File Transfer Protocol)*, *SMTP (Simple Mail transport Protocol)*, *SNMP (Simple Network Management Protocol)*, *HTTP (Hypertext Transfer Protocol)*, *DHCP (Dynamic Host Configuration Protocol)* dan *DNS (Domian Name System)* .
2. ***Transport Layer***, berisi protokol yang bertanggung jawab untuk mengadakan komunikasi antar dua komputer. Pada layer ini terdiri atas dua protokol, yaitu: *TCP (Transport Control Protocol)* dan *UDP (User Datagram Protocol)*.
3. ***Internet Layer***, berfungsi untuk menangani pergerakan paket data dalam jaringan dari komputer pengirim ke komputer tujuan. Protokol yang berada dalam fungsi ini antara lain: *IP (Internet Protocol)*, *ICMP (Internet Control Message Protocol)*, dan *IGMP (Internet Group Management Protocol)*.
4. ***Network Layer***, merupakan *layer* paling bawah yang bertanggung jawab mengirim dan menerima data dari dan ke media fisik.

### 2.3.1 TCP (*Transmission Control Protocol*)

(*Transmission Control Protocol*) TCP merupakan protokol yang berada pada *layer transport* dari *layer* TCP/IP. TCP adalah protokol yang bersifat *byte stream*, *connection oriented* dan *reliable* dalam pengiriman data. TCP menggunakan komunikasi *byte stream*, yang berarti bahwa data dinyatakan sebagai suatu urutan-urutan *byte*. *Connection oriented* berarti sebelum terjadi proses pertukaran data antar komputer terlebih dahulu harus dibentuk suatu hubungan. Hal ini dapat dianalogikan dengan proses *dial-up* nomor telepon dan akhirnya terbentuk hubungan.

Kehandalan TCP dalam mengirimkan data didukung oleh mekanisme yang disebut *Positive Acknowledgement with Re-transmission* (PAR). Data yang dikirim dari *layer* aplikasi akan dipecah-pecah dalam bagian-bagian yang lebih kecil dan diberi nomor urut sebelum dikirim ke *layer* berikutnya. Unit data yang sudah dipecah-pecah tadi disebut *segment*. TCP selalu meminta konfirmasi setiap kali selesai mengirimkan data, apakah data tersebut sampai pada komputer tujuan dan tidak rusak. Jika data berhasil sampai tujuan, TCP akan mengirimkan data urutan berikutnya. Jika tidak berhasil, maka TCP akan melakukan pengiriman ulang urutan data yang hilang atau rusak tersebut. (Aldya Dwiki, 2015).

Dalam kenyataannya TCP menggunakan sebuah *acknowledgement* (ACK) sebagai suatu pemberitahuan antara komputer pengirim dan penerima. Proses pembuatan koneksi TCP disebut juga dengan *Three-way Handshake*. Tujuan metode ini adalah agar dapat melakukan sinkronisasi terhadap nomor urut dan nomor *acknowledgement* yang dikirimkan oleh kedua pihak dan saling bertukar ukuran *TCP Window*. Prosesnya dapat digambarkan sebagai berikut :



**Gambar 2.3** Proses Pembuatan Koneksi TCP

(Sumber : Aldya Dwiki, 2015)

Keterangan dari Gambar 2.3 adalah sebagai berikut:

- *Host* pertama (yang ingin membuat koneksi) akan mengirimkan sebuah segmen TCP dengan *flag SYN* diaktifkan kepada *host* kedua (yang hendak diajak untuk berkomunikasi).
- *Host* kedua akan meresponsnya dengan mengirimkan segmen dengan *acknowledgment* dan juga *SYN* kepada *host* pertama.
- *Host* pertama selanjutnya akan mulai saling bertukar data dengan *host* kedua. TCP menggunakan proses *handshake* yang sama untuk mengakhiri koneksi yang dibuat. Hal ini menjamin dua *host* yang sedang terkoneksi tersebut telah menyelesaikan proses transmisi data dan semua data yang ditransmisikan telah diterima dengan baik. Itulah sebabnya, mengapa TCP disebut dengan koneksi yang *reliable*.

### 2.3.2 UDP (*User Datagram Protocol*)

UDP merupakan protokol yang juga berada pada *layer transport* selain TCP. Protokol ini bersifat *connectionless* dan *unreliable* dalam pengiriman data. *Connectionless* berarti tidak diperlukannya suatu bentuk hubungan terlebih dahulu untuk mengirimkan data. *Unreliable* berarti pada protokol ini tidak dijamin akan sampai pada tujuan yang benar dan dalam kondisi yang benar pula. Keandalan pengiriman data pada protokol ini menjadi tanggung jawab dari program aplikasi pada *layer* atasnya. Jika dibandingkan dengan TCP, UDP adalah protokol yang lebih sederhana dikarenakan proses yang ada didalamnya lebih sedikit. Dengan demikian aplikasi yang memanfaatkan UDP sebagai protokol *transport* dapat mengirimkan data tanpa melalui proses pembentukan koneksi terlebih dahulu. Hal ini pun terjadi pada saat mengakhiri suatu koneksi, sehingga dalam banyak hal proses yang terjadi sangatlah sederhana dibanding jika mengirimkan data melalui protokol TCP. (Aldya Dwiki, 2015).

Beberapa hal yang harus diperhatikan jika suatu program aplikasi akan menggunakan protokol UDP sebagai protokol *transport* :

- Tidak ada pembentukan koneksi. Protokol UDP hanya mengirim informasi begitu saja tanpa melakukan proses awal sebelumnya.
- Tidak ada pengkondisian koneksi. Protokol UDP tidak melakukan penentuan kondisi koneksi yang berupa parameter-parameter seperti *buffer* kirim dan terima, nomor urutan *segment*, dan *acknowledgement*.

- Memiliki *header* kecil. Protokol UDP memiliki 8 *byte header* dibanding 20 *header byte* pada TCP.
- Tidak ada pengaturan laju pengiriman. Protokol UDP hanya menekankan kecepatan kirim pada laju program aplikasi dalam menghasilkan data dan *bandwidth* akses menuju Internet.

## 2.4 Standar WLAN IEEE 802.11

Bermula pada tahun 1997, IEEE sebagai lembaga standarisasi internasional untuk perangkat elektronik menginisiasi dan menetapkan sebuah standar IEEE 802.11 sebagai standar regulasi pertama untuk teknologi jaringan nirkabel. Dalam rangka meningkatkan kehandalan WLAN, IEEE merilis IEEE 802.11a pada tahun 1999 sebagai amandemen pertama terhadap standar IEEE 802.11. Ada dua hal utama yang direvisi yaitu: metode modulasi dan alokasi pita frekuensi. Metode modulasi yang digunakan untuk IEEE 802.11a adalah *Orthogonal Frequency Division Multiplexing* (OFDM), sedangkan alokasi pita frekuensi yang ditetapkan untuk pengoperasiannya adalah 5 GHz.

Pada bulan Juli 1999, IEEE kembali mengeluarkan spesifikasi baru bernama 802.11b. Peralatan yang menggunakan standar 802.11b juga bekerja pada frekuensi 2,4 GHz. Salah satu kekurangan peralatan *wireless* yang bekerja pada frekuensi ini adalah kemungkinan terjadinya interferensi dengan *microwave oven* atau peralatan lain yang menggunakan gelombang radio pada frekuensi sama. (Aldya Dwiki, 2015).

Pada tahun 2002, IEEE membuat spesifikasi baru yang dapat menggabungkan kelebihan 802.11b dan 802.11a. Spesifikasi yang diberi kode 802.11g ini bekerja pada frekuensi 2,4 GHz dengan kecepatan transfer data maksimal 54 Mbps. Peralatan 802.11g kompatibel dengan 802.11b. Misalkan saja sebuah komputer yang menggunakan jaringan 802.11g dapat memanfaatkan *access point* 802.11b, dan sebaliknya. (Aldya Dwiki, 2015). Pada Tabel 2.1 dijelaskan standar IEEE 802.11 yang bekerja pada 2,4 GHz.

**Tabel 2.1** Spesifikasi Standar IEEE 802.11 pada 2,4 GHz

Standar	Data Rate		Jarak Jangkauan		Frekuensi	Kompatibel
	<i>Typical</i>	<i>Maximum</i>	<i>Indoor</i>	<i>Outdoor</i>		
802.11b	4,5 Mbps	11 Mbps	± 38 meter	± 125 meter	2,4 GHz	802.11b
802.11g	19 Mbps	54 Mbps	± 38 meter	± 125 meter	2,4 GHz	802.11b/g
802.11n	74 Mbps	150 Mbps	± 70 meter	± 230 meter	2,4 GHz	802.11b/g/n

### 2.4.1 Standar IEEE 802.11n

IEEE 802.11n adalah sebuah perubahan standar jaringan nirkabel 802.11 untuk meningkatkan *throughput* lebih dari standar sebelumnya. IEEE 802.11n meningkatkan data *rate* maksimum dalam lapisan fisik OSI (PHY) sampai 300 Mbit/s dengan lebar saluran 40 MHz. MIMO adalah teknologi yang menggunakan beberapa antena untuk menyelesaikan informasi lebih lanjut secara koheren dari pada menggunakan satu antena. Dua manfaat penting MIMO adalah menyediakan keragaman antena dan spasial multiplexing untuk 802.11n.

Kemampuan lain teknologi MIMO adalah menyediakan *Spatial Division Multiplexing* (SDM). SDM secara spasial multiplexes beberapa stream data independen, ditransfer secara serentak dalam satu saluran spektral bandwidth. MIMO SDM dapat meningkatkan *throughput* data seperti jumlah dari pemecahan stream data spasial yang ditingkatkan. Setiap aliran spasial membutuhkan antena yang terpisah baik pada pemancar dan penerima.

*Channel* 40 MHz adalah teknologi yang dipasang pada standar IEEE 802.11n yang secara simultan menggunakan dua *channel* terpisah yang *non-overlapping* untuk mentransfer data. *Channel* 40 MHz menaikkan jumlah data yang dapat ditransmisikan. *Channel* 40 MHz menggunakan dua band 20 MHz yang berdekatan. Hal ini dapat diaktifkan di 5 GHz mode, atau dalam 2,4 GHz. Arsitektur coupling MIMO dengan saluran bandwidth yang lebih luas menawarkan peningkatan fisik transfer rate melebihi 802.11a (5 GHz) dan 802.11g (2,4 GHz). (Mariza Azhar, 2010)

## 2.5 Interferensi

Jika suatu daerah mempunyai beberapa unit komunikasi pemancar-penerima (*transceiver*) dan beberapa pengguna menggunakan kanal yang sama atau kanal yang berdekatan, maka kinerja dipengaruhi oleh interferensi. Masing-masing pemancar dan penerima tidak hanya dipengaruhi oleh karakteristik daerah sekitar. Pengaruh interferensi ini biasanya lebih besar dari pengaruh *noise*. (Endy Hendrawan, 2016).

### 1. Direct Interference

*Direct interference* interferensi yang disebabkan oleh perangkat 802.11 lain yang beroperasi pada frekuensi atau kanal yang sama dalam satu area.

## 2 Indirect Interference

*Indirect interference* merupakan interferensi yang disebabkan oleh perangkat selain 802.11 tetapi bekerja pada spektrum frekuensi yang sama.

## 3. Path Interference

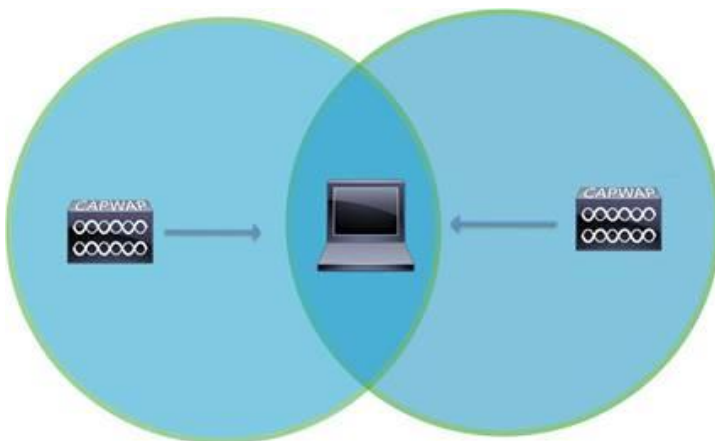
*Path interference* dibagi dalam 4 kategori : *Reflection*, *Refraction*, *Diffraction*, dan *Scattering*. Frekuensi radio (terutama pada *range* 5 GHz) memiliki kecenderungan yang kuat untuk dipantulkan oleh benda-benda logam, cermin, dan benda keras lainnya.

## 4. Line of Sight Interference

*Line of Sight Interference* merupakan interferensi yang disebabkan oleh penyerapan sinyal oleh benda-benda yang dilaluinya.

### 2.5.1 Pengaruh Interferensi Antar Access Point WLAN 802.11n

Interferensi yang berasal dari jaringan *wi-fi* lain menjadi masalah salah satu masalah umum pada penggunaan *wi-fi* pada frekuensi 2,4 GHz. Penggunaan *access point* pada frekuensi yang sama dan dalam daerah yang berdekatan, dapat mengakibatkan *Co-Channel Interference*.



**Gambar 2.4** *Co-Channel Interference*

(Sumber: Cisco)

IEEE 802.11 bekerja berdasarkan mekanisme *Clear Channel Assessment (CCA)* untuk menilai kerja medium. Mekanisme tersebut yaitu tidak mentransmisikan sinyal jika sibuk, dan mentransmisikan sinyal jika tidak sibuk. Pada Gambar 2.4, performansi pada *client*

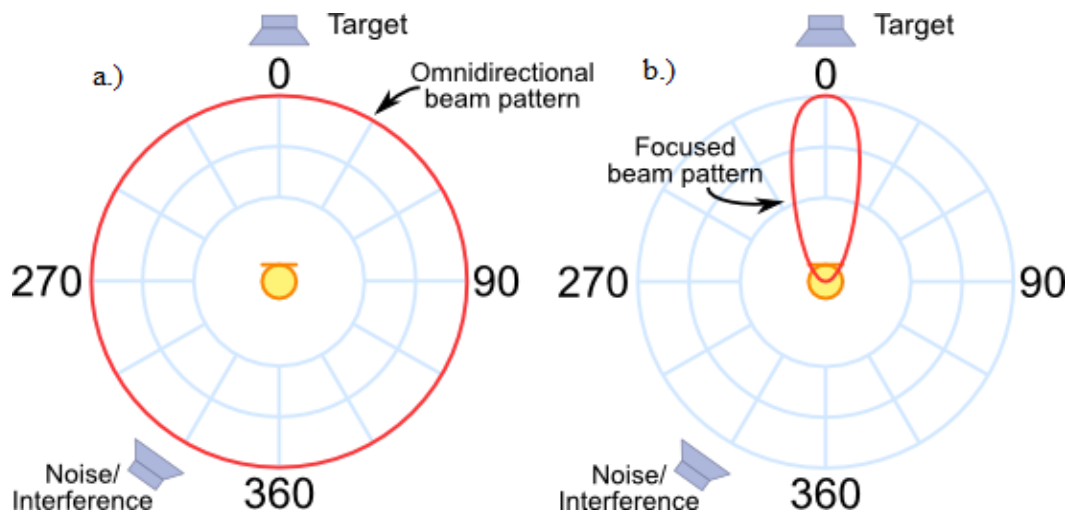


terkena dampak karena *client* dapat mendengar kedua *access point*. Pada *client*, kedua *access point* bersatu dan bertindak sebagai satu *cell*. Pada *uplink*, transmisi pada kedua *access point* akan terlihat sebagai *channel* yang sibuk oleh *client* dan *client* akan menunggu kesempatan untuk transmisi. Pada *downlink*, transmisi pada kedua *access point* akan berpotensi bertabrakan dan akan menimbulkan *delay* dan *packet loss*.

## 2.6 Beamforming

*Beamforming* adalah sebuah metode pemrosesan sinyal menggunakan sensor *array* untuk mengarahkan pentransmisiannya. Dengan menggunakan *beamforming* dapat mengarahkan daya sinyal pada daerah atau sudut yang ditentukan. Metode ini dilakukan dengan membuat pola radiasi dari antena *array* dengan cara menambahkan fasa dari sebuah sinyal pada arah target yang diinginkan, dan *nulling* pola pada target yang tidak diinginkan. *Beamforming* melakukan penggabungan sinyal dari masing-masing elemen *array* dengan mengoptimalkan respon *array* terhadap arah tertentu, sehingga dapat menaikkan SINR (*Signal to Interference plus Noise Ratio*). Arah tertentu dimana *array* memiliki respon maksimum ini disebut sebagai arah *beam-pointing*.

Pada Gambar 2.5, ditunjukkan perbedaan antara penggunaan teknologi *beamforming* dan tidak menggunakan teknologi *beamforming*. Pada Gambar 2.5 a, merupakan pola radiasi dari *access point* tanpa menggunakan teknologi *beamforming*, dimana memiliki pola radiasi *omnidirectional*. *Access point* memancarkan gelombang radio ke segala arah. Hal itu dapat mengakibatkan terjadinya interferensi. Pada Gambar 2.5 b, merupakan pola radiasi dari *access point* menggunakan teknologi *beamforming*. Saat menggunakan teknologi *beamforming*, *access point* memancarkan gelombang dengan fokus pada satu arah yang diinginkan dan dapat menghindari interferensi dari arah lain.

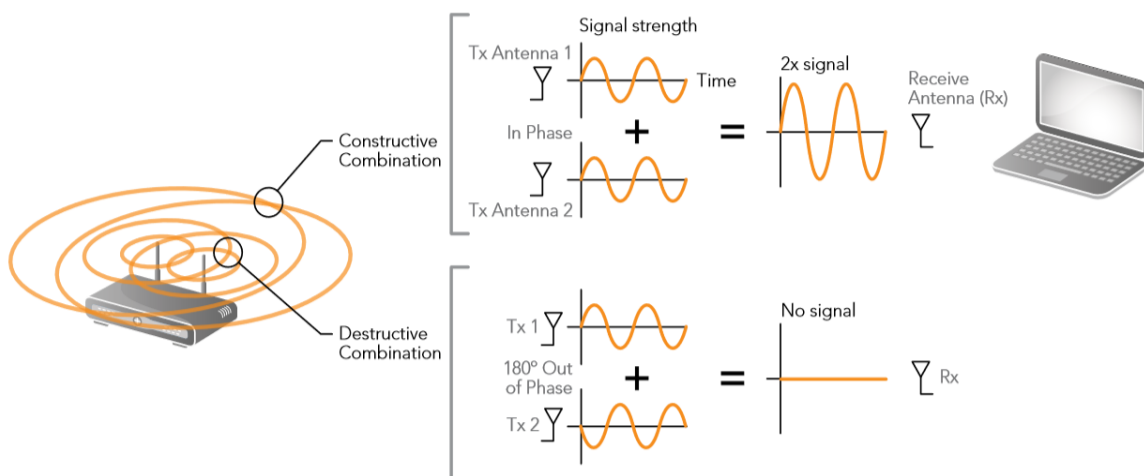


**Gambar 2.5** Teknologi *Beamforming*

(Sumber: labbookpages.co.uk)

Metode *beamforming* dapat dibagi menjadi dua kategori: *fixed beamforming* dan *adaptive beamforming*. Pada *fixed beamforming*, sistem antena membentuk banyak *beam* tetap dengan sensitivitas yang tinggi pada arah tertentu. Maka dari itu, sinyal dapat ditransmisikan dalam satu arah yang telah ditentukan. *Adaptive beamforming* adalah sistem yang menggunakan *adaptive spatial signal processing* dengan array pada *transmitter* dan *receiver*-nya yang memiliki kemampuan untuk mendeteksi dan melacak berbagai tipe sinyal secara efektif. Maka dari itu, *adaptive beamforming* dapat meminimalisir interferensi dan memaksimalkan *gain* dari sinyal yang diinginkan.

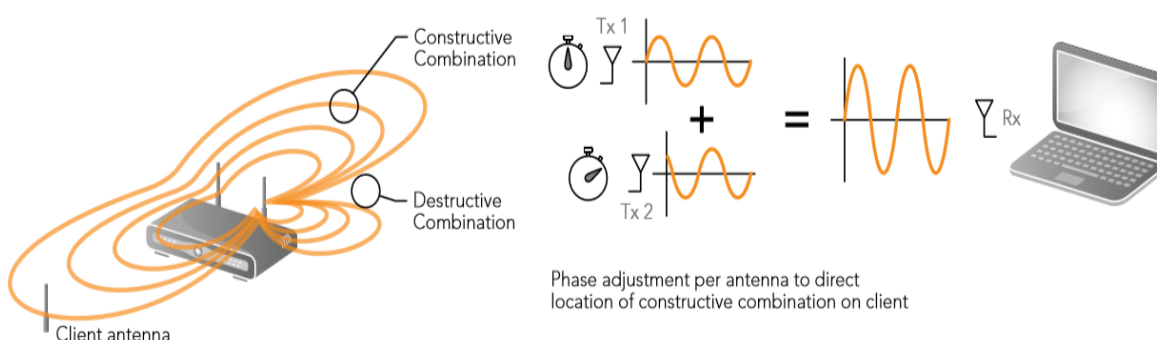
### 2.6.1 Prinsip Kerja *Beamforming*



**Gambar 2.6** *Constructive Combination* dan *Destructive Combination*

(Sumber : Ruckus)

Pada Gambar 2.6, dua buah sinyal yang sama dari dua antenna dapat membentuk kombinasi *constructive* dan *destructive* pada pentransmisian sinyal. Jika dua sinyal berada pada fasa yang sama, maka kedua sinyal tersebut akan saling menguatkan. Dan jika kedua sinyal tersebut berada pada fasa berbeda  $180^\circ$ , maka kedua sinyal itu akan saling menghilangkan. Pada teknologi *beamforming*, kedua sinyal akan membentuk kombinasi konstruktif pada arah atau *client* yang dituju, dan akan membentuk kombinasi destruktif pada arah atau *client* yang tidak diinginkan (interferensi). Pada Gambar 2.7, *access point* akan melakukan kombinasi konstruktif pada sinyal pada arah yang diinginkan (*client antenna*) dan melakukan kombinasi destruktif pada arah yang tidak diinginkan.

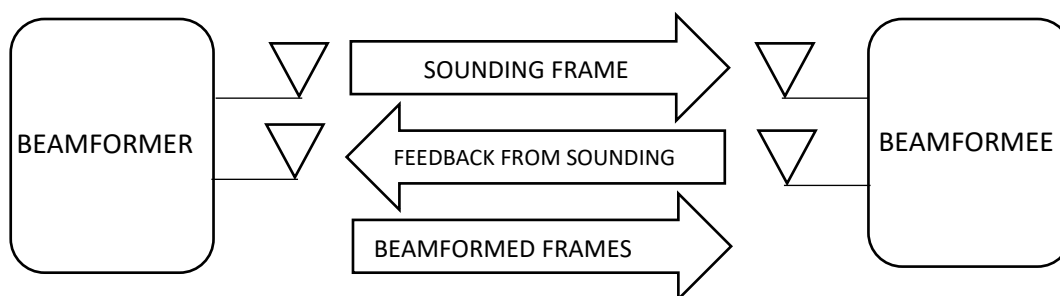


**Gambar 2.7** Pengaturan Fasa Sinyal

(Sumber : Ruckus)

### 2.6.2 Transmit *Beamforming*

Dalam melakukan transmisi *beamforming*, dibutuhkan respon atau *feedback* dari *client*. Pada Gambar 2.8, *beamformer* merupakan *access point* dengan teknologi *beamforming*. *Beamformer* mengirimkan “*sounding frame*” untuk diterima oleh *beamformee* (*client*). Lalu, *beamformee* memberikan *feedback* ke *beamformer*. Dan *beamformer* dapat mengirimkan paket-paket data.

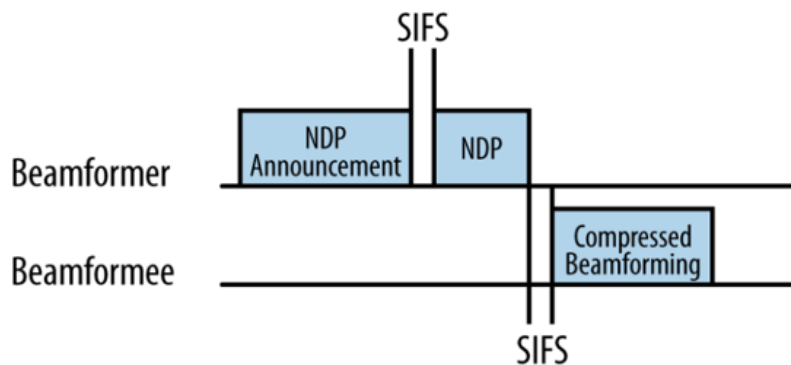


**Gambar 2.8** Channel Sounding

(Sumber: Aruba Networks)

Pada Gambar 2.9 menggambarkan bagaimana transmit *beamforming* dilakukan.

1. *Beamformer* mengirimkan paket NDP (*Null Data Packet*) Announcement untuk mengontrol *channel* dan mengidentifikasi *beamformee*. Lalu, *beamformee* akan merespon NDP Announcement tersebut.
2. *Beamformer* kemudian mengirimkan *null data packet* (NDP). NDP dapat membaca respon dari *channel* berisi matriks untuk melakukan transmisi *beamforming*.
3. Respon dari *beamformee* yang berupa matriks untuk melakukan *beamforming* diterima oleh *beamformer*. Menggunakan respon matriks dari *beamformee*, *beamformer* dapat melakukan transmisi *beamforming* ke *client/beamformee*.



**Gambar 2.9** Transmit *Beamforming*

(Sumber: O'Reilly)

## 2.7 Quality of Service (QoS)

*Quality of Service* (QoS) merupakan mekanisme jaringan yang memungkinkan aplikasi-aplikasi atau layanan dapat beroperasi sesuai dengan yang diharapkan.

Kinerja jaringan komputer dapat bervariasi akibat beberapa masalah, seperti halnya masalah *throughput*, *delay/latency* dan *packet loss*, yang dapat membuat efek yang cukup besar bagi kecepatan akses internet. Sebagai contoh, saat melakukan *download* suatu file, pengguna akan merasa kesal ketika file yang sedang *download* begitu lama selesai dikarenakan *bandwidth* yang tidak cukup, dengan *jitter* yang tinggi, atau *packet loss* yang berlebih

Menurut ITU-T E. 800, QoS adalah : “Sekumpulan efek performansi yang menentukan derajat kepuasan pengguna terhadap *service* yang diperlukan oleh jaringan”. Sedangkan dari sudut pandang jaringan telekomunikasi QoS adalah: “Kemampuan suatu

jaringan untuk menyediakan layanan yang lebih baik pada trafik data tertentu pada berbagai jenis *platform* teknologi”. (Onno W. Purbo, 2001).

### 2.7.1 Throughput

*Throughput* merupakan kecepatan (*rate*) transfer data efektif, yang diukur dalam bit per sekon (bps). Terkadang istilah *bandwidth* digunakan sebagai sinonim dari *throughput*. Tetapi hal tersebut adalah salah. *Bandwidth* adalah suatu ukuran dari banyaknya informasi yang dapat mengalir dari suatu tempat ke tempat lainnya dalam suatu waktu tertentu. Sedangkan *throughput* adalah *bandwidth* yang sebenarnya (aktual) pada saat pengguna menggunakan internet. *Throughput* dapat dikategorikan berdasarkan persentase seperti pada Tabel 2.2.

**Tabel 2.2** Kategori *Throughput*

Kategori <i>Throughput</i>	<i>Throughput</i>
Sangat Bagus	76-100 %
Bagus	51-75 %
Sedang	26-50 %
Buruk	< 25 %

(Sumber : European Telecommunication Standards Institute)

### 2.7.2 Jitter

*Jitter* merupakan variasi *delay* antar paket yang terjadi pada jaringan IP. Besarnya nilai *jitter* akan sangat dipengaruhi oleh variasi beban trafik dan besarnya tumbukan antar paket (*congestion*) yang ada dalam jaringan IP. Semakin besar beban trafik di dalam jaringan akan menyebabkan semakin besar pula peluang terjadinya *congestion* dengan demikian nilai *jitter*-nya akan semakin besar. Semakin besar nilai *jitter* akan mengakibatkan nilai QoS akan semakin turun. Untuk mendapatkan nilai QoS jaringan yang baik, nilai *jitter* harus dijaga seminimum mungkin. *Jitter* sebuah jaringan dapat dikategorikan berdasarkan Tabel 2.3 sebagai berikut :

**Tabel 2.3** Kategori *Jitter*

Kategori <i>Jitter</i>	Besar <i>Jitter</i>
Sangat Bagus	0 ms
Bagus	0 - 75 ms
Sedang	76 - 125 ms
Buruk	> 125 ms

(Sumber : European Telecommunication Standards Institute)

### 2.7.3 Packet Loss

*Packet loss* merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang, dan dapat terjadi karena interferensi pada jaringan. Hal ini berpengaruh pada semua aplikasi karena transmisi ulang akan mengurangi efisiensi jaringan secara keseluruhan meskipun jumlah *bandwidth* cukup tersedia untuk aplikasi tersebut. Di dalam implementasi jaringan IP, nilai *packet loss* ini diharapkan mempunyai nilai yang minimum. Persentase *packet loss* dapat dikategorikan berdasarkan Tabel 2.4 sebagai berikut :

**Tabel 2.4** Kategori *Packet Loss*

Kategori <i>Packet Loss</i>	Besar <i>Packet Loss</i>
Sangat Bagus	0 %
Bagus	1- 3 %
Sedang	4-15 %
Buruk	16-25 %

(Sumber : European Telecommunication Standards Institute)

## 2.8 iPerf

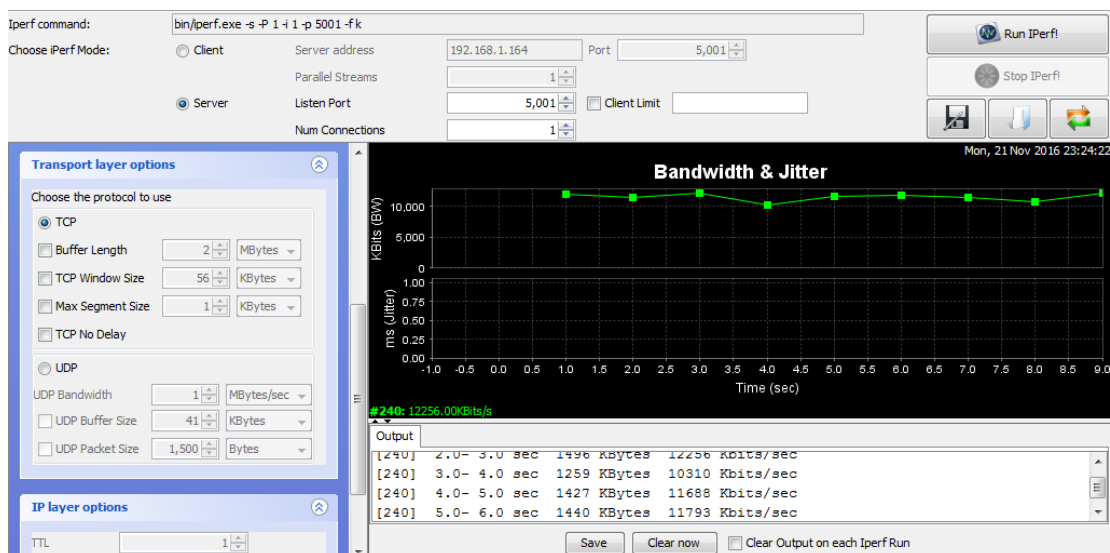
Iperf adalah salah satu tool untuk mengukur *throughput* dalam sebuah *link network*, agar bisa dilakukan pengukuran diperlukan Iperf yang terinstall *point to point*, baik disisi *server* maupun *client*. Iperf sendiri bisa digunakan untuk mengukur performansi *link* dari sisi TCP maupun UDP. (Gigih Fordanama, 2011)

### 1. TCP

- Pengukuran *bandwidth*.
- Mendukung TCP *windows size via socket buffers*.

- *Client* dan *server* dapat membuat beberapa koneksi secara bersamaan.

Setelah menjalankan *iPerf* dengan mengirimkan paket TCP maka didapatkan *output* seperti pada Gambar 2.6. *Throughput* jaringan dapat dilihat pada kolom *bandwidth*.



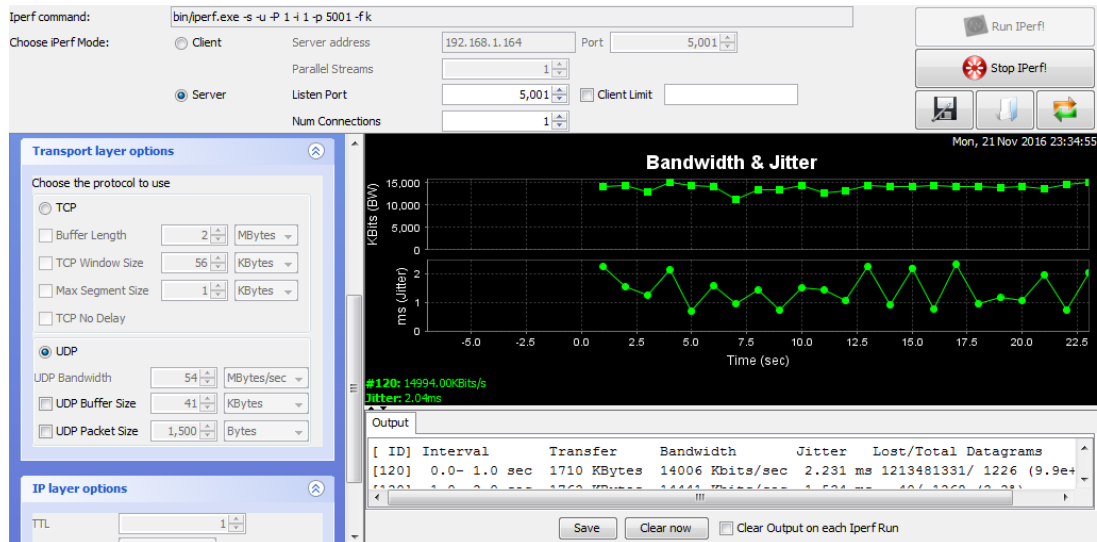
**Gambar 2.10** Hasil *iPerf* pada TCP

(Sumber: Perancangan)

## 2. UDP

- *Client* dapat membuat paket UDP sesuai dengan *bandwith* yang diinginkan.
- Pengukuran *packet loss*.
- Pengukuran *jitter*.
- *Client* dan *server* dapat membuat beberapa koneksi secara bersamaan.

Setelah menjalankan *iPerf* dengan mengirimkan paket UDP maka didapatkan *output* seperti pada Gambar 2.7. Pada pengukuran dengan paket UDP didapatkan data *throughput*, *jitter*, dan *packet loss*.



**Gambar 2.11** Hasil *iPerf* pada UDP

(Sumber: Perancangan)