

## ANALISIS DAN IMPLEMENTASI PERFORMANSI DATAGRAM CONGESTION CONTROL PROTOCOL (DCCP) UNTUK LAYANAN VIDEO STREAMING PADA WLAN

### ANALYSIS AND IMPLEMENTATION PERFORMANCE OF DATAGRAM CONGESTION CONTROL PROTOCOL (DCCP) FOR VIDEO STREAMING OVER WLAN

<sup>1</sup>Andri PiterParulian      <sup>2</sup>Asep Mulyanan, ST.,MT<sup>3</sup>R.Rumani M, BcTT,Msc,Ir.,Drs  
<sup>1,2,3</sup>Fakultas Elektrodan Komunikasi - Institut Teknologi Telkom

Jln. Telekomunikasi Dayeuhkolot Bandung 40257 Indonesia

<sup>1</sup>piter.giallorossi@gmail.com

<sup>2</sup>ASM@it.telkom.ac.id

<sup>3</sup>RRM@ it.telkom.ac.id

#### ABSTRAK

Berkembangnya teknologi saat ini semakin memudahkan banyak orang untuk mendapatkan informasi secara cepat dan tepat. Informasi tersebut tidak hanya berupa *text* ataupun gambar tetapi juga melibatkan *multimedia*. Salah satu layanan yang sering diakses untuk mendapatkan informasi adalah melalui *video-streaming*. Layanan *video-streaming* mengajak kita untuk mendapatkan informasi tanpa harus men-download-nya terlebih dahulu. Namun layanan *streaming* ini belum sepenuhnya sempurna karena masih adanya *delay* pada saat kita mengakses layanan tersebut. Oleh sebab itu, peningkatan kualitas *video-streaming* diperlukan agar setiap orang bisa menikmati layanan ini dengan baik.

Salah satu cara agar pelayanan dalam bentuk *video-streaming* menjadi lebih baik adalah dengan menggunakan protokol *streaming* yang baik. Salah satu protokol yang digunakan untuk layanan *video-streaming* adalah *Datagram Congestion Control Protocol* (DCCP). DCCP yang merupakan pengembangan dari *User Datagram Protocol* (UDP) dalam menangani *congestion* menerapkan pola pengiriman *bidirectional-unicast* pada paket datagram yang disesuaikan dengan tingkat *congestion* pada *network*.

Dalam tugas akhir ini, dilakukan simulasi dengan menggunakan *Network Simulator 2* untuk mengetahui performansi dari DCCP. Dengan pengaruh jumlah user yang berbeda, yaitu 5, 10 dan 20, didapatkan hasil sebagai berikut (secara berurutan): delay (0.656, 0.398, 0.384), packet loss (0.04%,0.13%,0.06%), throughput (422.14 kbps, 397.71 kbps, 395.68 kbps). Selain itu, simulasi juga dilakukan dengan pengaruh perubahan bitrate, yang menghasilkan delay 4.98, throughput 4507.07 dan packet loss 0.03%.

**Kata kunci:**DCCP, Congestion Control, video streaming,wireless,QoS,network simulator 2

#### ABSTRACT

The development of technology nowadays makes it easy for many people to obtain information quickly and accurately. Not only it is a text or picture information, but it can also be multimedia information. One of the services that is frequently accessed to obtain information is video-streaming. Video-streaming service enables us to obtain information without having to download it first. However, this service is not really perfect yet because some delay may happen in the process of accessing the information. Therefore, it is really important to improve the quality of this service so people can enjoy this service better.

One way to make this video-streaming service better is to use a better streaming protocol. One of the protocols which is used for video-streaming is *Datagram Congestion Control Protocol* (DCCP). DCCP which is the developing of *User Datagram Protocol* (UDP) in handling *congestion* works by using *bidirectional-unicast* sending pattern on datagram packet which is adjusted with *congestion* level on *network*.

In this thesis, the simulation is carried out by using *Network Simulator 2* to know the performance of DCCP. With the influence of different number of users, that is, 5 users, 10 users, and 20 users, this is the result of the simulation (in order of number of users): delay (0.656, 0.398, 0.384), packet loss (0.04%,0.13%,0.06%), throughput (422.14 kbps, 397.71 kbps, 395.68 kbps). Besides, simulation is also carried out with the influence of bitrate change, which results in delay 4.98, throughput 4507.07 and packet loss 0.03%.

**Keywords:**DCCP, Congestion Control, video streaming,network simulator 2

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Berkembangnya teknologi saat ini semakin memudahkan banyak orang untuk mendapatkan informasi secara cepat dan tepat. Informasi tersebut tidak hanya berupa text ataupun gambar tetapi juga melibatkan multimedia. Salah satu layanan yang sering diakses untuk mendapatkan informasi adalah melalui video streaming. Layanan video streaming mengajak kita untuk mendapatkan informasi tanpa harus men-download-nya terlebih dahulu. Namun layanan streaming ini belum sepenuhnya sempurna karena masih adanya a delay pada saat kita mengakses layanan tersebut. Oleh sebab itu peningkatan kualitas video streaming diperlukan agar setiap orang bisa menikmati layanan ini dengan baik.

Salah satu cara agar pelayanan dalam bentuk video streaming menjadi lebih baik adalah dengan menggunakan protokol streaming yang baik. Salah satu protokol yang digunakan untuk layanan video streaming adalah Datagram Congestion Control Protocol (DCCP). DCCP yang merupakan pengembangan dari User Datagram Protocol (UDP) dalam menangani congestion menerapkan pola pengiriman bidirectional-unicast pada paket datagram yang disesuaikan dengan tingkat congestion pada network.

Berdasarkan masalah di atas, maka pada tugas akhir yang berjudul "Analisis dan Implementasi Performansi Datagram Congestion Control Protocol (DCCP) untuk Layanan Video Streaming pada WLAN" ini akan dilakukan analisis performansi DCCP pada video streaming di WLAN. Dari hasil analisis tersebut diharapkan mendapatkan hasil yang baik sehingga kualitas video streaming dapat diakses dengan baik.

### 1.2 Tujuan Penulisan Proyek Akhir

Tujuan dari tugas akhir ini adalah untuk mengetahui tingkat performansi dari protokol *Datagram Congestion Control Protocol* (DCCP) untuk layanan *video streaming*. Pada tugas akhir ini juga akan diukur *Quality of Services* berupa *delay*, *jitter*, *packet loss*, dan *throughput* untuk mengetahui tingkat performansi dari protokol tersebut. Sehingga dari hasil penelitian ini dapat diambil kesimpulan apakah protokol DCCP merupakan protokol yang baik untuk menangani layanan *video streaming*.

### 1.3 Rumusan Masalah

Permasalahan yang akan dibahas dalam tugas akhir ini adalah :

- a. Melakukan implementasi untuk mengetahui kualitas performansi dari *congestion control* DCCP pada WLAN.
- b. Melakukan analisis tingkat performansi DCCP pada layanan *video streaming*-GSstreamer dengan algoritma CCID 3.
- c. Melakukan analisis parameter-parameter kualitas seperti: *delay*, *jitter*, *throughput* dan *packet loss*.
- d. Mengambil kesimpulan dari hasil yang didapat

### 1.4 Metode Penelitian

Metode penyelesaian masalah dalam tugas akhir ini adalah sebagai berikut:

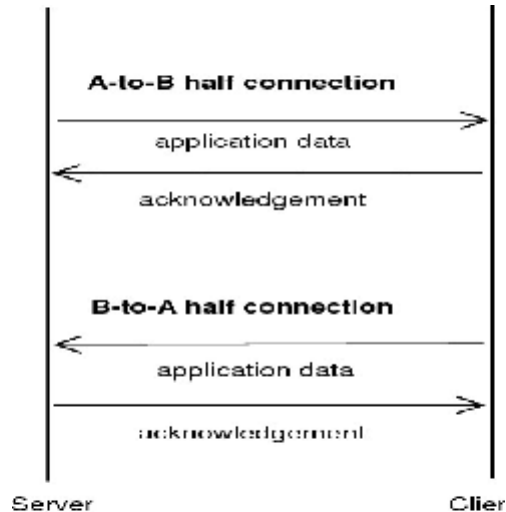
1. Studi literatur  
Mempelajari teori dan konsep melalui literatur yang didapatkan melalui berbagai media seperti buku referensi, artikel, jurnal dan sumber-sumber lainnya, yang ada di perpustakaan maupun *internet* yang berkaitan dengan permasalahan Tugas Akhir ini.
2. Perancangan sistem  
Merancang sistem manajemen pada DCCP dan aplikasi *video streaming* sesuai dengan topologi jaringan yang direncanakan.
3. Implementasi  
Melakukan implementasi *Datagram Congestion Control Protocol* (DCCP) pada WLAN dengan menggunakan layanan *video streaming* pada sistem yang telah dirancang.
4. Analisis dan pengambilan data  
Melakukan pengambilan data hasil implementasi dan melakukan analisis terhadap data tersebut berupa *delay*, *jitter*, *throughput* dan *pack*

## 2. DASAR TEORI

### 2.1 Datagram Congestion Control Protocol (DCCP)

Transport protokol yang merupakan pengembangan dari *User Datagram Protocol* (UDP) dalam menangani *congestion* adalah *Datagram Congestion Control Protocol* (DCCP). DCCP menerapkan pola pengiriman *bidirectional-unicast* pada paket datagram yang disesuaikan dengan tingkat *congestion* pada *network*. DCCP sangat sesuai untuk

diterapkan pada aplikasi yang melakukan transfer data dengan skala besar dan membutuhkan optimalisasi antara efisiensi waktu maupun *reability*.



Gambar 2.1 : *Two-half Connection* pada DCCP

DCCP menerapkan mekanisme *handshake*

yang *reliable* dimana mekanisme ini tidak diimplementasikan pada UDP. Melalui *handshake*, DCCP dapat mengidentifikasi adanya *packet-loss* melalui *acknowledgement* yang ditransmisikan oleh *destination node* untuk menjamin *reability* dari data yang terlibat. Dalam menangani *congestion*, DCCP menerapkan *Explicit Congestion*

*Notification* (ECN) pada IP yang terlibat seperti yang diterapkan pada *Transmission Control Protocol* (TCP). ECN mendeteksi adanya *congestion* melalui penyisipan pesan pada IP *header* yang diinisialisasi oleh *source node* untuk dapat memastikan ketersediaan *route* yang bebas dari *congestion* menuju *destination node*.

DCCP menggunakan *Congestion Control Identifiers* (CCID) untuk mendeteksi terjadinya *congestion*. Selama *proses connection setup*, *source* dan *destination* akan menegosiasikan mekanisme *congestion-control* yang digunakan melalui fitur CCID. Salah satu mekanisme *congestion-control* pada DCCP adalah *TCP-like Congestion Control* (CCID 2). Mekanisme CCID 2 memodelkan *Additive Increase, Multiplicative Decrease* (AIMD) *congestion-control* seperti pada TCP, dengan implementasi *congestion window*, *slow start* dan *timeouts*. CCID 2 memanfaatkan *bandwidth* secara optimum dan menangani *end-to-end congestion* secara konsisten. Penanganan *congestion* pada CCID 2 direspon melalui pengurangan *congestion window* menjadi  $\frac{1}{2}$  dari *congestion window* awal sehingga terjadi perubahan nilai *rate* dari paket yang dikirimkan.

*Congestion window* merupakan variabel yang digunakan dalam membatasi jumlah data yang dapat dikirim oleh *Transport Protocol*. Nilai *congestion window* ditentukan oleh ACK yang diterima oleh *source node*. *Source node* yang mengalami penurunan nilai *congestion window* mengidentifikasi terhentinya penerimaan ACK yang disebabkan oleh *congestion* melalui penurunan nilai menjadi  $\frac{1}{2}$  dari nilai semula. Mekanisme *congestion control* dilakukan dengan penyesuaian *rate* dari pengiriman *DCCP-Data Packet* pada *source node* (*sender*).

Setiap paket DCCP dimulai dengan *header* yang memiliki ukuran yang *fixed*. Tipe paket tertentu menambahkan *header* khusus seperti *DCCP-Ack* yang menambahkan *Acknowledgement Number*. Berikut ini merupakan tipe paket yang digunakan oleh DCCP :

1. DCCP-Request  
Paket ini dikirimkan oleh *client-node* untuk menginisialisasi sebuah *connection*.
2. DCCP-Response  
Paket ini merupakan respon *server-node* terhadap *DCCP-Request*.
3. DCCP-Data  
Paket yang digunakan untuk mentransmisikan data.
4. DCCP-Ack  
Paket yang digunakan untuk mentransmisikan *acknowledgement*.
5. DCCP-Data Ack  
Paket yang digunakan dalam merespon informasi *acknowledgement*.
6. DCCP-CloseReq  
Digunakan oleh *server-node* untuk menyampaikan *request* kepada *client-node* agar menutup *connection*.
7. DCCP-Close  
Digunakan oleh *client* ataupun *server* untuk menutup *connection*.
8. DCCP-Reset  
Digunakan dalam membatalkan *connection* yang sedang aktif.
9. DCCP-Sync,DCCP-SyncAck  
Digunakan dalam sinkronisasi ulang terhadap *sequence numbers* apabila terjadi *packet-loss* dalam skala besar.

DCCP secara khusus ditujukan bagi aplikasi seperti *streaming media*, dimana optimalisasi antara *delay* dan *reability* pada *traffic* menjadi pertimbangan utama. TCP tidak sesuai untuk diimplementasikan pada aplikasi seperti ini karena proses transmisi yang *reliable* dan mekanisme *congestion control* yang digunakan menyebabkan terjadinya *delay* yang panjang. Sementara itu, UDP dapat meminimalisir *delay* namun tidak tersedianya mekanisme *congestion*

control berisiko mengakibatkan terjadinya *packet loss* yang mempengaruhi *reliability* dan data *integrity*.

## 2.2 Video Streaming

*Video streaming* merupakan sebuah teknologi yang memungkinkan kita untuk memutar *file video* dari komputer lain tanpa harus *download* keseluruhan *file*, dengan kata lain *file video* diputar ketika sebagian dari *file* diterima dan dikodekan. Dalam *video streaming* terdapat komputer yang menjalankan *encoder* (seperti *windows media encoder*), *server streaming* yang berfungsi sebagai penyedia layanan *streaming*, dan *player* pada *client* yang memutar *video streaming*.

### 2.2.1 Aplikasi pada Video Streaming

Ada berbagai macam jenis aplikasi streaming yang memiliki kondisi operasi yang berbeda, seperti: aplikasi dapat berupa *on demand* maupun *live streaming* dan aplikasi juga dapat berupa *broadcast*, *multicast* maupun *unicast*.

#### 2.2.1.1 Live Streaming dan On Demand

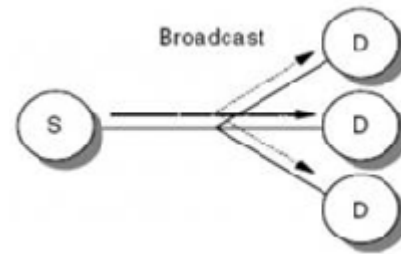
##### Streaming

Pada aplikasi *live streaming* sumber video diambil dan dikodekan secara *real time*. Aplikasi interaktif adalah salah satu contoh dari aplikasi yang membutuhkan *real time encoding*, seperti *video conference*. Sedangkan pada aplikasi *on demand streaming*, *file video* di-*encode* dan disimpan dalam *storage* terlebih dahulu (*pre-encoded*) sebelum ditransmisikan. Kelebihan yang dimiliki adalah bahwa aplikasi *on demand streaming* tidak memerlukan proses *encoding* secara *real time*. Salah satu aplikasinya adalah *Video on Demand*. Pada tugas akhir ini, aplikasi streaming yang akan dilakukan adalah *Video on Demand*.

#### 2.2.1.2 Broadcast, Multicast, Unicast

##### a. Broadcast

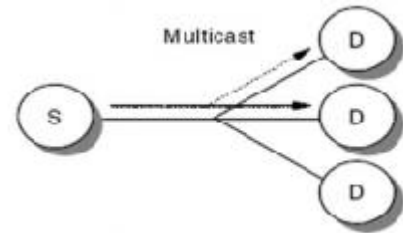
*Broadcast* adalah bentuk komunikasi *one to many* (yang pada dasarnya adalah *one to all*) yang paling dikenal, salah satu contohnya adalah siaran TV. Dengan menggunakan *broadcast* maka informasi yang akan dikirimkan berasal dari satu sumber atau titik kepada semua penerima yang tergabung dalam jaringan. Pada *broadcast* semua penerima mau tidak mau akan menerima informasi ini.



Gambar 2.2 Broadcast

##### b. Multicast

*Multicast* juga merupakan bentuk komunikasi *one to many*, tetapi tidak seperti *broadcast*. Pada *multicast* informasi yang dikirimkan berasal dari satu sumber atau titik kepada semua penerima yang menginginkan informasi tersebut. *Server* akan membuat *stream* satu kali kemudian *stream* ini diduplikasi dan dikirimkan ke setiap *client*. Setiap *client* akan menerima *stream* yang sama dengan *client* lainnya. Salah satu contoh penggunaan *multicast* adalah *live video*.



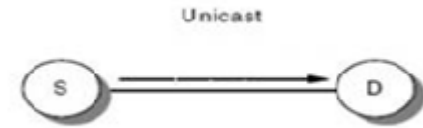
Gambar 2.3 Multicast

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

##### c. Unicast

*Unicast* merupakan bentuk komunikasi *one to one* atau *point to point*. Pada *unicast* informasi yang dikirimkan berasal dari satu sumber atau titik ke satu titik lainnya. *Server* akan mengirimkan *file streaming* ke komputer *client* berkali-kali bergantung pada banyaknya jumlah permintaan. Setiap *client* akan menerima

file *streaming* yang terpisah dari *client* lainnya. Contoh penggunaan *unicast* adalah *video on demand*, sehingga pada tugas akhir ini akan dilakukan secara *unicast*.



Gambar 2.4 Unicast

### 2.3 Permasalahan Dalam Jaringan

Beberapa masalah utama yang timbul dalam implementasi video adalah antara lain :

#### 2.4.1 Bandwidth

*Bandwidth* merupakan pemegang kunci terpenting dalam teknologi video. Karena agar sebuah *file media* yang akan di-*broadcast* dapat dinikmati sebagaimana mestinya, akses internet ataupun intranet haruslah memiliki *bandwidth* yang memadai. Keterbatasan antara dua titik pada jaringan internet maupun intranet secara umum tidak dapat diketahui. Jika pengirim mengirimkan data lebih cepat dibandingkan dengan ketersediaan *bandwidth* yang ada, maka akan terjadi *congestion* (kemacetan) pada jaringan, *packet loss*, dan kualitas video yang kurang baik. Tapi apabila pengirim (*sender*) mengirimkan paket data lebih kecil dari *bandwidth* yang ada, maka kualitas video yang akan sampai ke penerima (*reciever*) akan kurang optimal.

#### 2.4.2 Delay

*Delay* adalah waktu untuk suatu paket yang diakibatkan oleh proses transmisi dari satu titik ke titik lain yang menjadikannya. *Delay* di dalam jaringan dapat digolongkan sebagai berikut:

##### Packetisasi delay

*Delay* yang disebabkan oleh waktu yang diperlukan untuk proses pembentukan paket *IP* dari informasi *user*. *Delay* ini hanya terjadi sekali saja, yaitu di *source* informasi.

##### Queuing delay

*Delay* ini disebabkan oleh waktu proses yang diperlukan oleh *router* di dalam menangani transmisi paket di sepanjang jaringan. Umumnya *delay* ini sangat kecil, kurang lebih sekitar 100 *micro second*.

##### Delay propagasi

*Delay* ini merupakan waktu proses perjalanan informasi selama di dalam media transmisi.

#### 2.4.3 Jitter

*Jitter* merupakan variasi *delay* antar paket yang terjadi pada jaringan *IP*. Besarnya nilai *jitter* sangat dipengaruhi oleh besarnya tumbukan antar paket *congestion* yang ada pada jaringan *IP*. Semakin besar

beban trafik pada jaringan akan menyebabkan semakin besar peluang terjadinya *congestion* dengan demikian *jitter* akan semakin besar.

#### 2.4.4 Packet Loss

*Packet Loss* ini dapat menimbulkan kerusakan pada kualitas video hasil rekonstruksi. Untuk mengatasi *Packet Loss* ini, pada sistem *video streaming* dapat didesain dengan fasilitas *error rate control*. Untuk perhitungannya sebagai berikut:

$$\text{Packet Loss} = \frac{\text{Loss}}{\text{Rate}} \times 100\%$$

#### 2.4.5 Throughput

Dalam komunikasi jaringan, seperti *ethernet* atau *packet radio*, atau jaringan *throughput* adalah rata-rata tingkat keberhasilan pengiriman pesan melalui saluran komunikasi. Data ini dapat disampaikan melalui fisik atau logis link, atau melalui suatu jaringan *node*. *Throughput* yang biasanya diukur dalam bit per detik (bit/s atau bps), kadang-kadang dalam paket data per detik atau data paket per waktu slot. Sistem *throughput* adalah *throughput aggregate* atau jumlah data *rate* yang dikirim ke semua terminal dalam jaringan. Dengan perhitungannya sebagai berikut:

$$\text{Throughput} = \frac{\text{Data}}{\text{Time}} \times 100\%$$

### 2.5 GStreamer

*GStreamer* adalah kerangka *plugin* yang berbasis, dan *plugin* masing-masing berisi elemen. Masing-masing unsur menyediakan fungsi tertentu seperti pengkodean, menampilkan, atau *data render* serta kemampuan untuk membaca dari atau menulis ke *file*. Dengan menggabungkan dan menghubungkan elemen-elemen, *programmer* dapat membangun jaringan pipa untuk melakukan fungsi yang lebih kompleks. Sebagai contoh, adalah mungkin untuk membuat pipa untuk membaca dari file *MP3*, decoding isinya, dan bermain *MP3*.

#### 2.6 Wireless LAN

*Wireless* merupakan teknologi yang bertujuan untuk menggantikan kabel yang menghubungkan terminal komputer dengan jaringan,

dengan begitu komputer dapat berpindah dengan bebas dan tetap dapat berkomunikasi dalam jaringan dengan kecepatan transmisi yang memadai. *Wireless LAN* distandarisasi oleh IEEE dengan kode 802.II b yang bertujuan untuk menyamakan semua teknologi

nirkabel yang digunakan di bidang komputer dan untuk menjamin interoperabilitas antara semua produk-produk yang menggunakan standar ini.

**2.5.1 Standarisasi Wireless LAN**

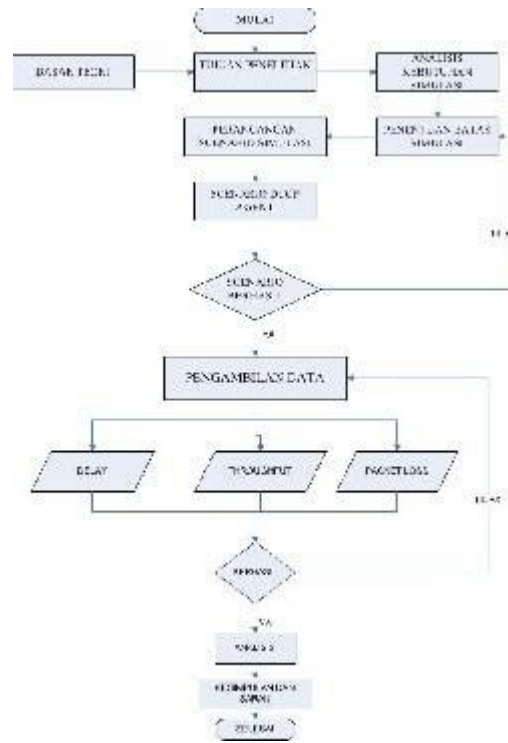
Karena wireless LAN mengirim data dengan menggunakan frekuensi radio, wireless LAN diaturoleh jenishukum yang samadandigunakan untuk mengatur hal-hal seperti AM/FM radio. Federal Communications Commission (FCC) mengaturlatdari wireless LAN. Dalam pemasaran wireless LAN sekarang, menerima beberapa standar operasional dan syarat dari Amerika Serikat yang diciptakan dan dirawat oleh Institute of Electrical Electronic Engineers (IEEE).

Beberapa standar wireless LAN:

- IEEE 802.11: standar asli wireless LAN menetapkan tingkat perpindahan data yang paling lambat dalam teknologi transmisi light-based dan RF.
- IEEE 802.11b: menggambarkan tentang beberapa transfer data yang lebih cepat dan lebih bersih di dalam lingkup teknologi transmisi.
- IEEE 802.11a: gambar tentang pengiriman data lebih cepat dibandingkan (tetapi kurang sesuai dengan) IEEE 802.11b, dan menggunakan 5 GHz frekuensi band UNIL.
- IEEE 802.11g : syarat yang paling terbaru berdasarkan pada 802.11 standar yang menguraikan transfer data samadengan cepatnya seperti IEEE 802.11a, dan sesuai dengan 802.11b yang memungkinkan untuk lebih murah.

**2. KONFIGURASI DAN IMPLEMENTASI**

**3.1 Flow Perencanaan**



**DAFTAR PUSTAKA**

1. Baniadam, Mahardi, 2009. "Layanan Streaming TV Over IP Multichannel Berbasis Web di Jaringan Intranet IT TELKOM". Proyek akhir jurusan teknik telekomunikasi ITTELKOM, Bandung.
2. Kohler, Edie, Mark Handley, Sally Floyd. "Designing DCCP: Congestion Control Without Reliability". ICSI Center for Internet Research.
3. Khalid, Muhammad Naveed. "Simulation Based Comparison of SCTP, DCCP and UDP Using MPEG-4 Traffic Over Mobile WiMAX/IEEE 802.16e". School of Computing Blekinge Institute of Technology SE - 371 79 Karlskrona, Sweden.
4. Statev, Stanimir, Seferin Mirtchev, 2008. "Experimental Study of DCCP Transport Protocol in Varied Network States". Technical University of Sofia, Bulgaria