

BAB II

TINJAUAN PUSTAKA

2.1 Layanan Jaringan Internet

Layanan internet merupakan jaringan yang saling berhubungan dari jaringan - jaringan komputer yang menghubungkan orang - orang dan komputer - komputer diseluruh dunia melalui telepon, satelit dan sistem-sistem komunikasi yang lainnya. Layanan jaringan internet dibentuk oleh jutaan komputer yang terhubung bersama dari seluruh dunia, memberi jalan bagi informasi (mulai dari teks, audio, gambar, video, dan lain-lain) untuk dapat dikirim dan dinikmati secara bersama-sama [3].

Banyaknya kebutuhan akan penggunaan layanan jaringan internet maka memerlukan jaringan yang berada pada kondisi yang baik. Hal ini tentunya selain menimbulkan tingginya angka kebutuhan perangkat dari segi *user* (pengguna) juga mengakibatkan perlunya peningkatan infrastruktur penunjangnya seperti pengembangan jaringan, alokasi *bandwidth* yang memadai dan optimal.

2.2 Bandwidth Management System

Bandwidth Management System (BMS) adalah sebuah metode yang diterapkan untuk mengatur besarnya *bandwidth* yang akan digunakan oleh masing-masing *user* di sebuah jaringan sehingga penggunaan *bandwidth* akan terdistribusi secara merata. Ada beberapa metode yang dapat diterapkan untuk mengimplementasikan manajemen *bandwidth* ini diantaranya melalui *proxy server*, QoS atau *traffic shapping*, atau pembatasan *bandwidth* atau limiter [4].

Dalam penelitian ini digunakan dua metode untuk manajemen *bandwidth* yaitu metode HTB (*Hierarchical Token Bucket*) dan metode PCQ (*Peer Connection Queue*).

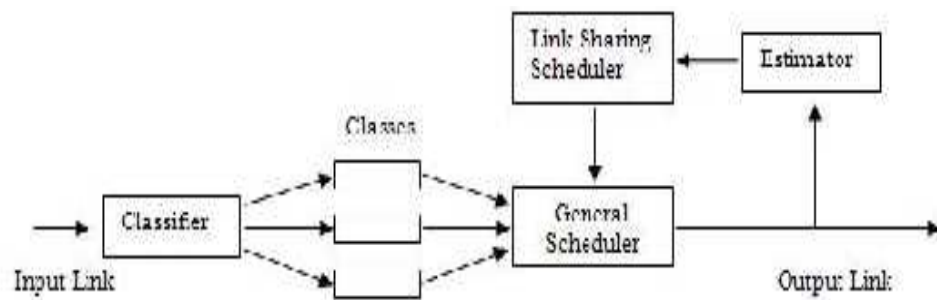
2.2.1 Metode Hierarchical Token Bucket (HTB)

Hierarchical Token Bucket (HTB) merupakan teknik penjadwalan paket yang digunakan kebanyakan *router* berbasis Linux. HTB diklaim

menawarkan kemudahan pemakaian dengan teknik peminjaman dan implementasi pembagian trafik yang lebih akurat. Dasar kerja HTB hampir sama dengan disiplin antrian CBQ bahkan diagram blok sistem CBQ dengan HTB tidak ada bedanya, hanya saja pada *General Scheduler* HTB menggunakan mekanisme *Deficit Round Robin* (DRR) dan pada blok umpan balik, Estimator, HTB tidak menggunakan *Exponential Weighted Moving Average* (EWMA) melainkan *Token Bucket Filter* (TBF) [6].

Penjadwalan pengiriman paket antrian, maka HTB menggunakan suatu proses penjadwalan yang dapat dijelaskan sebagai berikut [7]:

1. *Class*, merupakan parameter yang diasosiasikan dengan rate yang dijamin AR (*assured rate*), CR (*ceil rate*), prioritas P, level dan quantum. *Class* dapat memiliki *parent*. Selain AR dan CR, didefinisikan juga *actual rate* atau R, yaitu *rate* dari aliran paket yang meninggalkan *class* dan diukur pada suatu perioda waktu tertentu.
2. *Leaf*, merupakan *class* yang tidak memiliki anak. Hanya *leaf* yang dapat memegang antrian paket.
3. *Level*, dari kelas menentukan posisi dalam suatu hirarki. *Leaf-leaf* memiliki level 0, *root class* memiliki level=jumlah level-1 dan setiap *inner class* memiliki level kurang dari satu dari parentnya.
4. *Mode*, dari *class* merupakan nilai-nilai buatan yang diperhitungkan dari R, AR dan CR. Mode-mode yang mungkin adalah: Merah: $R > CR$; Kuning: $R \leq CR$ and $R > AR$; Hijau selain di atas.



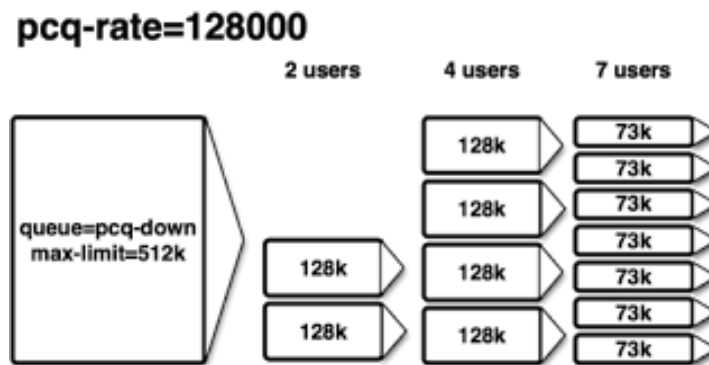
Gambar 2.1 *Mekanisme Deficit Round Robin (DRR)* [7]

2.2.2 *Metode Peer Connection Queue (PCQ)*

Prinsip kerja PCQ dengan menerapkan simple queue atau queue trees dimana hanya ada satu klien aktif yang menggunakan bandwidth, sementara klien lain berada dalam posisi idle maka klien aktif tersebut dapat menggunakan bandwidth maksimum yang tersedia, tetapi jika klien lain aktif, maka bandwidth yang maksimal dapat digunakan oleh kedua klien (bandwidth atau jumlah klien yang aktif) sehingga bandwidth dapat terdistribusi secara adil untuk semua klien.

Pada prinsipnya, penggunaan metode antrian untuk menyeimbangkan bandwidth yang digunakan pada beberapa klien. Dalam OS mikrotik, PCQ adalah program untuk mengelola jaringan Lalu Lintas Kualitas Layanan (QoS). Tujuan utama dari metode ini adalah untuk melakukan bandwidth sharing otomatis dan merata ke multiclient.

Kerja prinsip PCQ dengan menerapkan simple queue atau queue trees dimana hanya ada satu klien aktif yang menggunakan bandwidth, sementara klien lain berada dalam posisi idle maka klien aktif tersebut dapat menggunakan bandwidth maksimum yang tersedia, tetapi jika klien lain aktif, maka bandwidth yang maksimal dapat digunakan oleh kedua klien (bandwidth atau jumlah klien yang aktif) sehingga bandwidth dapat terdistribusi secara adil untuk semua klien [3].



Gambar 2.2 PCQ Rate [4]

PCQ bekerja dengan membuat *sub-stream* berdasarkan parameter *pcq classifier* yang dapat berupa *IP Address* pengirim berdasarkan pengirim (*src-address*), *IP Address* tujuan (*dst-address*), *Port* pengirim (*src-port*) maupun *Port* tujuan (*dst-port*) [5].

2.3 Quality of Service (QoS)

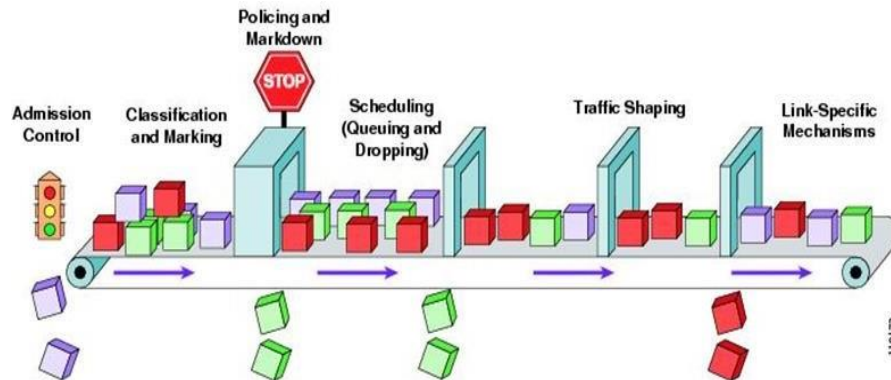
Quality Of Service (QoS) merupakan kemampuan suatu *network* untuk menyediakan *service* yang lebih baik untuk pengguna dalam membagi *bandwidth* sesuai kebutuhan data dan *voice* yang digunakan [6].

Tabel 2.1 Kategori Kualitas QoS

Nilai	Persentase (%)	Indeks
3,8 – 4	95 – 100	Sangat memuaskan
3 – 3,79	75 – 94,75	Memuaskan
2 – 2,99	50 – 74,75	Kurang memuaskan
1 – 1,99	25 – 49,75	Jelek

QoS merupakan peralatan-peralatan yang tersedia untuk menerapkan berbagai jaminan, dimana tingkat minimum layanan dapat disediakan. Banyak protokol dan aplikasi yang tidak begitu sensitif terhadap *network congestion*. *File Transfer Protocol (FTP)* contohnya, mempunyai toleransi yang besar untuk *network delay* dan terbatasnya *bandwidth*.

Di sisi pengguna, kejadian tersebut akan menyebabkan proses transfer file seperti *download* atau *upload* yang lambat, walaupun mengganggu pengguna, namun kelambatan ini tidak akan menggagalkan operasi dari aplikasi tersebut. Berbeda dengan aplikasi-aplikasi baru seperti *voice* dan *video*, yang pada umumnya sensitif terhadap *delay*. Jika paket dari *voice* mengalami proses yang lama untuk sampai ke tujuan, maka akan dapat merusak *voice* yang didengarkan.



Gambar 2.3 Qos dalam Mengatur Sebuah Jaringan [8]

Dalam hal ini QoS dapat digunakan untuk menyediakan jaminan layanan untuk aplikasi-aplikasi tersebut. Ada beberapa alasan mengapa kita memerlukan QoS, yaitu [8] :

1. Untuk memberikan prioritas untuk aplikasi-aplikasi yang kritis pada jaringan.
2. Untuk memaksimalkan penggunaan investasi jaringan yang sudah ada.
3. Untuk meningkatkan performansi untuk aplikasi-aplikasi yang sensitif terhadap *delay* seperti *voice* dan *video*.
4. Untuk merespon terhadap adanya perubahan-perubahan pada aliran *traffic* di jaringan.

2.3.1 Parameter *Quality of Service (QoS)*

Performansi mengacu ke tingkat kecepatan dan keandalan penyampaian berbagai jenis beban data di dalam suatu komunikasi. Performansi merupakan kumpulan dari beberapa parameter teknis, yaitu [6] :

2.3.1.1 Delay

Delay Adalah waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. *Delay* dapat dipengaruhi oleh jarak, media fisik, kongesti atau juga waktu proses yang lama [9].

Delay dapat dipengaruhi oleh jarak, media fisik, kongesti atau juga waktu proses yang lama. Perhitungan *delay* dapat dicari menggunakan rumus sebagai berikut [6]:

$$\text{Delay} = \frac{\text{Total Delay}}{\text{Packet packet yang diterima}} \quad (2.1)$$

Delay versi *Telecommunications and Internet Protocol Harmonization Over Networks* (TIPHON) dikelompokkan menjadi empat kategori seperti terlihat pada Tabel 2.2 [6].

Tabel 2.2 Kategori *Delay*

Kategori	Besar <i>Delay</i>	Indeks
Sangat Bagus	<150 ms	4
Bagus	150 s/d 300 ms	3
Sedang	300 s/d 450 ms	2
Jelek	> 450 ms	1

2.3.1.2 Jitter

Jitter merupakan penyimpangan yang tidak diinginkan dari periodisitas benar sebuah periodik diasumsikan sinyal dalam elektronik dan telekomunikasi, sering dalam kaitannya dengan referensi sumber jam. *Jitter* dapat diukur dalam kondisi yang sama seperti semua sinyal waktu yang bervariasi, misalnya, RMS, atau puncak-ke puncak perpindahan. Sama halnya seperti sinyal waktu bervariasi lainnya, *jitter* dapat dinyatakan dalam hal kepadatan spektral (frekuensi konten) [6].

Perhitungan *jitter* dapat dicari menggunakan rumus sebagai berikut [6]:

$$\text{Jitter} = \frac{\text{Total variasi delay}}{(\text{Total packet}-1)} \quad (2.2)$$

Kategori kinerja jaringan berbasis IP dalam *jitter* versi *Telecommunications and Internet Protocol Harmonization Over Networks* (TIPHON) mengelompokkan menjadi empat kategori penurunan kinerja jaringan berdasarkan nilai *Jitter* seperti terlihat pada Tabel 2.3 [6].

Tabel 2.3 Kategori Jitter

Kategori	Besar <i>Jitter</i>	Indeks
Sangat Bagus	0	4
Bagus	0 s/d 75 ms	3
Sedang	75 s/d 125 ms	2
Jelek	125 s/d 225 ms	1

2.3.1.3 Packet Loss

Packet loss merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang, dapat terjadi karena *collision* dan *congestion* pada jaringan dan hal ini berpengaruh pada semua aplikasi karena *retransmisi* akan mengurangi efisiensi jaringan secara keseluruhan meskipun jumlah *bandwidth* cukup tersedia untuk aplikasi-aplikasi tersebut. Umumnya perangkat jaringan memiliki *buffer* untuk menampung data yang diterima. Jika terjadi kongesti yang cukup lama, *buffer* akan penuh, dan data baru tidak akan diterima [6].

Perhitungan packet loss dapat dicari menggunakan rumus sebagai berikut [6]:

$$\text{Packet Loss} = \frac{\text{Packet data yang dikirim} - \text{Packet data yang diterima}}{\text{Packet data yang dikirim}} \times 100\%$$

$$* \text{ Packet data yang diterima} = \text{Packet data yang dikirim} - \text{Packet data yang hilang} \quad (2.3)$$

Packet loss versi *Telecommunications and Internet*

Protocol Harmonization Over Networks (TIPHON) dikelompokkan menjadi empat kategori seperti terlihat pada Tabel 2.4.

Tabel 2.4 Kategori Packet Loss

Kategori	Packet Loss	Indeks
Sangat Bagus	0 %	4
Bagus	3 %	3
Sedang	15 %	2
Jelek	25 %	1

2.3.1.4 Throughput

Throughput merupakan kecepatan (*rate*) transfer data efektif, yang diukur dalam bps. *Troughput* merupakan jumlah total kedatangan paket yang sukses yang diamati pada *destination* selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut [6].

Perhitungan *throughput* dapat dicari menggunakan rumus sebagai berikut [6]:

$$\text{Throughput} = \frac{\text{Packet data yang diterima}}{\text{Lama pengamatan}} \quad (2.4)$$

Berikut merupakan tabel kategori *throughput* yang didapatkan berdasarkan standar TIPHON, antara lain sebagai berikut [6] :

Tabel 2.5 Kategori *Throughput*

Kategori	Throughput	Indeks
Sangat Bagus	> 2.1 Mbps	5
Lebih Bagus	1.2 Mbps s/d 2.1 Mbps	4
Bagus	700 Kbps s/d 1.2 Mbps	3
Cukup Bagus	338 s/d 700 Kbps	2
Jelek	0 s/d 388 Kbps	1

2.3.2 Perangkat Lunak Pendukung QoS

Perangkat lunak (*software*) sebagai pendukung dalam mengukur besar *Quality of Service* (QoS) adalah :

2.3.2.1 Wireshark

Wireshark adalah tool open source terkemuka yang banyak di gunakan untuk melakukan analisis dan pemecah masalah jaringan, Memungkin kan kita untuk mengetahui masalah di jaringan.

2.3.2.2 Axecence NetTools

NetTools adalah salah satu Network analyzer yang sangat handal. Tool ini dipakai unuk mengukur/menganalisa performance network dan men-diagnosa problem yang terjadi pada network tersebut. NetTools sangat populer karena dilengkapi dengan trace, lookup, port scanner, network scanner, dan SNMP browser.

2.3.2.3 Colasoft Capsa 11

Colasoft Capsa 11 adalah freeware penganalisa jaringan untuk pemantauan, analisis ethernet, *problem solving*,. Hal ini memudahkan pengguna untuk mengetahui cara memantau aktivitas jaringan, mengetahui masalah pada jaringan, dan meningkatkan keamanannya.

2.4 Algoritma Naïve Bayes

2.4.1 Pengertian Algoritma Naïve Bayes

Algoritma Naive Bayes adalah algoritma yang mampu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya dengan menggunakan metode probabilitas dan statistik [12]. Rumus yang digunakan pada algoritma Naive Bayes adalah sebagai berikut [12] :

$$P(H|X) = \frac{P(X|H)}{P(X)} \quad (2.5)$$

Di mana :

X : Data dengan class yang belum diketahui.

H : Hipotesis data merupakan suatu class spesifik.

P(H|X): Probabilitas hipotesis H berdasar kondisi X (posteriori probabilitas)

$P(H)$: Probabilitas hipotesis H (prior probabilitas)

$P(X|H)$: Probabilitas X berdasarkan kondisi pada hipotesis H

$P(X)$: Probabilitas X

Agar mampu mengerti mengenai Algoritma Naive Bayes, perlu diketahui bahwa diperlukan beberapa petunjuk guna mengetahui kelas apa yang cocok bagi parameter yang akan dianalisis untuk melakukan proses klasifikasi. Oleh karena itu, penyesuaian dilakukan terhadap Algoritma Naive Bayes seperti pada persamaan 2.6 berikut [12] :

$$P(C|F_1 \dots F_n) = \frac{P(C|F_1 \dots F_n)}{P(C|F_1 \dots F_n)} \quad (2.6)$$

Varibel C pada Persamaan 2.6 dapat digunakan untuk mewakili kelas, sedangkan variabel $F_1 \dots F_n$ digunakan untuk mewakili karakteristik petunjuk yang merupakan bagian penting guna melakukan klasifikasi. Dari persamaan dapat diartikan bahwa kemungkinan masuknya sampel karakteristik tertentu pada kelas C (Posterior) adalah kemungkinan munculnya Kelas C (seringkali disebut prior, karena belum masuk sampel yang baru), dikalikan dengan kemungkinan munculnya karakteristik-karakteristik sampel dalam kelas C (dapat dikatakan likelihood), dibagi dengan kemungkinan munculnya karakteristik-karakteristik contoh secara global (dapat dikatakan evidence). Dari penjelasan diatas, Persamaan 2.6

dapat dijelaskan secara sederhana seperti

pada Persamaan 2.7 berikut [12] :

$$\text{Posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}} \quad (2.7)$$

Karakteristik-karakteristik sampel secara global atau disebut juga evidence pada satu sampel selalu tetap untuk setiap kelas. Nilai dari posterior pada akhirnya dibandingkan dengan nilai posterior pada kelas lain guna membantu menentukan ke kelas apa suatu sampel akan digolongkan. Penjelasan mengenai rumus Bayes dapat dilakukan dengan cara menjabarkan $(C|F_1, \dots)$ lebih rinci mengikuti aturan perkalian sebagai berikut [12] :

$$P(C|F_1, \dots, F_n) = P(C) P(F_1, \dots, F_n|C) \quad (2.8)$$

$$\begin{aligned}
&= P(C) P(F_1|C) P(F_2, \dots, F_n|C, F_1) \\
&= P(C) P(F_1|C) P(F_2|C, F_1) P(F_3, \dots, F_n|C, F_1, F_2) \\
&= P(C) P(F_1|C) P(F_2|C, F_1) P(F_3|C, F_1, F_2) P(F_4, \dots, F_n|C, F_1, F_2, F_3) \\
&= P(C) P(F_1|C) P(F_2|C, F_1) P(F_3|C, F_1, F_2) \dots P(F_n|C, F_1, F_2, F_3, \dots, F_{n-1})
\end{aligned}$$

Dari penjabaran di atas maka diketahuilah bahwa hasil dari penjabaran tersebut menyebabkan semakin panjang dan semakin rumitnya faktor-faktor yang mempengaruhi nilai probabilitas, yang menyebabkan hamper tidak mungkin faktor tersebut untuk dianalisis satu per satu. Dampak dari penjabaran tersebut maka perhitungan Bayes menjadi sulit untuk dilakukan.

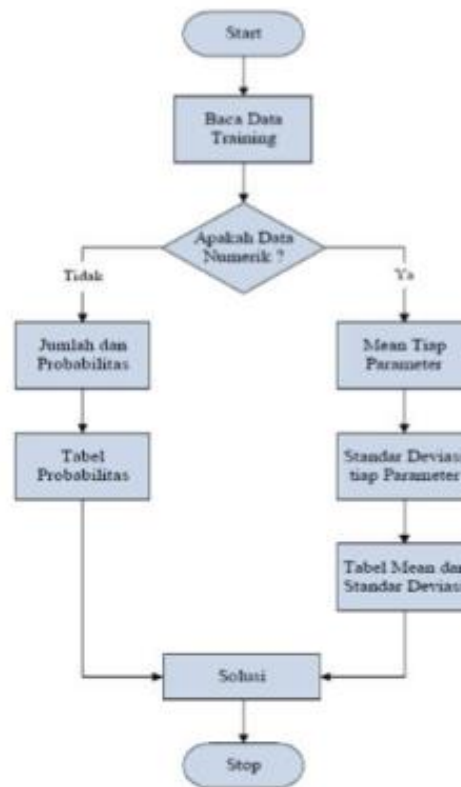
Pada bagian ini digunakanlah naif (asumsi independensi yang sangat tinggi), bahwa setiap petunjuk (F_1, F_2, \dots, F_n) independen atau tidak ada keterkaitan antara satu sama lain. Dari penjelasan diatas, maka digunakanlah kesamaan seperti pada Persamaan 2.8 [12] :

$$P(F_i | F_j) = P(F_i \cap F_j) = P(F_i)P(F_j) \quad P(F_j) = P(F_i) \quad (2.9)$$

Untuk $i \neq j$, sehingga

$$P(F_i | C, F_j) = P(F_i | C) \quad (2.10)$$

Berdasarkan Persamaan 2.9 dan Persamaan 2.10 dapat dilihat sebuah persamaan yang merupakan model dari teorema Naive Bayes yang dapat digunakan untuk melakukan proses klasifikasi.



Gambar 2.4 Alur Metode Naïve Bayes [12]

2.4.2 Pengukuran Algoritma Naïve Bayes

Algoritma Naive Bayes akan melakukan proses pengklasifikasian menggunakan dataset yang telah disediakan. Setelah didapatkan nilai dari setiap perhitungan parameter, maka algoritma Naive Bayes akan melakukan proses pengklasifikasian. Setelah didapatkan hasil dari parameter yaitu setelah didapat rata-rata Delay, Jitter dan Packet Loss. Naive Bayes akan melakukan pengklasifikasian bahwa jika kondisi seperti hasil pengujian diatas masuk kategori manakah status jaringan internetnya.

2.5 Penelitian Terdahulu

Tabel 2.6 Penelitian Terdahulu

No .	Judul	Keywords	Penulis	Tahun	Kelebihan	Kekurangan
1.	Analisis Kualitas Layanan Internet Kampus Menggunakan Metode HTB (Hierarchical Token Bucket)	<i>Quality of Service, HTB, Bandwidth Managemen t, Throughput , Delay, Packet Loss</i>	A.R. Walad Mahfuz hi, Bambang Soedijono dan Eko Pramono	2017	Hasil yang didapatkan dari metode yang diterapkan mendapatkan hasil yang optimal.	Pada pengujian sampel perlu ditambahkan agar dapat mewakili dari keadaan objek penelitian.
2.	Optimalisasi Jaringan Wireless Dan Analisis Quality Of Service (Qos) Menggunakan Metode Hierarchical Token Bucket (HTB)	<i>Hierarchical Token Bucket, Quality of Service, Mikrotik, Jaringan Wireless</i>	Risna, Isnawaty dan Sutardi	2017	Kualitas jaringan dengan metode ini lebih optimal, karena semua client mendapatkan kuota bandwidth sesuai dengan rule yang diterapkan pada bandwidth management.	Untuk pengujian ini dibutuhkan koneksi internet yang stabil
3.	Analisa Bandwidth Menggunakan Metode Antrian Per Connection Queue	<i>Bandwidth, PCQ, Simple Queue, Queue Tree, Quality Of Service</i>	Sukri, Jumiati	2017	Metode yang dipakai dapat memberikan hasil yang optimal dalam memenuhi kebutuhan.	Pada saat pengujian sistem harus menggunakan Koneksi jaringan internet yang stabil
4.	Implementasi Quality Of Service Dengan Metode Queue Tree Pada Kampus	<i>Queue Tree, Bandwidth Managemen t, Network System</i>	Daniel Setiawan , Sigit Setyowibowo	2017	Dalam penyampaian analisa dilakukan sangat detail sehingga mudah dalam	Untuk memaksimalkan queue tree tidak berlaku jika bandwidth dari ISP langganan

	Stmik Pradnya Paramita Malang				dipahami.	tidak stabil atau sedang dalam keadaan gangguan.
5.	Manajemen Bandwidth Simple Queue Dan Queue Tree Pada Pt. Endorsindo Makmur Selaras	<i>Bandwidth Managemen, Mikrotik, Simple Queue, Queue Tree</i>	Arif Budiman	2015	Penulisan data sangat detail sehingga dapat mudah dipahami.	Tidak didapatkan hasil yang optimal karena menerapkan sistem yang sama untuk bandwidth management pada setiap bagian.
6.	Implementasi Dan Analisa Per Connection Queue (Pcq) Sebagai Kontrol Penggunaan Internet Pada Laboratorium Komputer	<i>Bandwidth Managemen, Per Connection Queue (PCQ), Network</i>	Mirsantoso, Toibah Umi Kalsum, dan Reno Supardi	2015	Manajemen bandwith dengan tipe ini, dapat membatasi penyedotan bandwith oleh download manager seperti internet download manager (IDM) dan sejenisnya.	Jumlah pengguna yang sangat banyak dan tidak sesuai dengan jumlah bandwith yang ada.
7.	Implementasi Metode Simple Queue Dan Queue Tree Untuk Optimasi Manajemen Bandwith Jaringan Komputer Di Politeknik Aceh Selatan	<i>Simple Queue, Queue tree, Optimasi Manajemen Bandwith</i>	Dirja Nur Ilham	2018	Metode Ini dinilai lebih Sederhana dalam proses konfigurasinya dan dapat menggunakan semua bandwith yang tersedia	Metode ini dapat ditembus oleh download manager dan harus melakukan setting manggle terlebih dahulu

8.	Analisis Penerapan Metode Antrian Hirarchical Token Bucket Untuk Management Bandwidth Jaringan Internet	<i>Bandwidth, HTB, Management, Internet</i>	Ira Puspita Sari, Sukri	2018	Dalam penulisan hasil data, perhitungan dan analisa dilakukan sangat detail sehingga mudah dalam memahaminya.	Pada pengujian sistem manajemen bandwidth, hendaknya menggunakan koneksi internet yang stabil.
9.	Analisa Kualitas Layanan Internet Menggunakan Metode Peer Connection Queue (PCQ) dan Optimalisasi Menggunakan Metode Hierarchical Token Bucket (HTB).	<i>PCQ, HTB, QoS</i>	Setyowati, Triani Ajeng	2019	Dalam penulisan dan pengukuran didapat hasil data yang cukup detail sehingga mudah untuk dimengerti	Hasil analisa sebaiknya diklasifikasikan agar lebih mudah untuk diidentifikasi.
10.	Analisis Pemantauan LAN menggunakan Metode QoS dan Pengklasifikasi an Status Jaringan Internet Menggunakan Algoritma Naïve Bayes.	<i>Naïve Bayes, Network Administrator, QoS, status Jaringan Internet.</i>	Sabloak, Sachin	2018	Pada penulisan laporan sudah cukup jelas mengenai pengklasifikasi n status layanan internet.	Metode pengukuran QoS sebaiknya digunakan 2 metode agar lebih baik dalam pengukuran parameter QoS nya.