

Optimasi QOS (*Quality of Services*) Jaringan Digitalisasi SPBU Menggunakan Metode HTB (*Hirarki Token Bucket*) dan Metode Auto Failover Berbasis Web (Studi Kasus SPBU 4453313 Purbalingga)

Optimization of QOS (Quality of Services) of Gas Station Digitalization Network Using HTB (Hierarchy Token Bucket) Method and Web-Based Auto Failover Method (Case Study of Gas Station 4453313 Purbalingga)

Thoriq Firza Yudhatama*, Akhmad Fathurahman, Safuan

Universitas Muhammadiyah Semarang, Semarang

Corresponding author : thoriqfirzayudhatama@gmail.com

Abstrak

Pengelolaan Stasiun Pengisian Bahan Bakar Umum (SPBU) yang efisien membutuhkan strategi yang tepat dalam menghadapi kompleksitas kebutuhan teknologi informasi dan jaringan. Penelitian ini bertujuan untuk mengoptimalkan jaringan digitalisasi SPBU 4453313 Purbalingga dengan menerapkan Metode Hierarchical Token Bucket (HTB) untuk manajemen trafik data dan Metode *Auto Failover* untuk meningkatkan ketahanan jaringan berbasis web. Optimalisasi ini bertujuan untuk mengeksplorasi konsep dasar manajemen lalu lintas data, Metode HTB, dan *Auto Failover* melalui tinjauan literatur yang komprehensif. Selain itu, penulis menganalisis dampak digitalisasi pada industri SPBU dan mengeksplorasi teknologi jaringan yang biasa digunakan di SPBU. Metode penelitian ini mencakup implementasi Metode HTB untuk meningkatkan alokasi bandwidth berbasis prioritas, memastikan layanan-layanan penting mendapatkan prioritas yang optimal. Selain itu, Metode Auto Failover diintegrasikan untuk meningkatkan ketahanan jaringan, dengan fokus pada deteksi gangguan dan transfer otomatis ke jalur cadangan. Hasil Pengujian dengan metode HTB terbukti mampu meningkatkan QoS manajemen Bandwidth di SPBU 4453313, dengan peningkatan disisi *throughput & Delay*, dan pengujian *auto failover* juga terbukti efisien dengan meminimalisir waktu downtime sebesar 5,20 *second*.

Kata Kunci : Dasar Jaringan, Optimalisasi Jaringan, *Hierarchical Token Bucket*, *Auto Failover*.

PENDAHULUAN

Jaringan Digitalisasi SPBU mempunyai peran krusial pada penyaluran BBM (Bahan Bakar Minyak) terutama pada label BBM Subsidi. Dengan pentingnya peran digitalisasi SPBU (Stasiun Pengisian bahan bakar umum) ini maka wajib dimbangi dengan kualitas jaringan yang mumpuni. Keberhasilan SPBU untuk membagikan pelayanan yang optimal dan efisien kepada konsumen sangat tergantung pada kehandalan infrastruktur jaringan yang dipergunakan, penelitian Optimasi jaringan ini dijalankan di lokasi SPBU 4453313 Purbalingga.

Optimasi jaringan digitalisasi menjadi suatu aspek penting, dalam menghadapi dinamika kebutuhan konsumen yang semakin tinggi wajib diselaraskan juga dengan kualitas jaringan yang ada. Penyediaan akses internet yang cepat dan stabil, manajemen lalu lintas

data yang efisien, dan ketahanan jaringan pada kegagalan menjadi aspek-aspek kunci yang perlu diperhatikan.

Masalah yang ada pada jaringan SPBU / Kendala yang dikeluhkan oleh pengelola SPBU yaitu kecepatan sinyal yang kurang stabil sehingga menyebabkan beberapa transaksi error serta jika terjadi putus koneksi SPBU tidak dapat menjual BBM Subsidi, dikarenakan pemerintah mewajibkan pencatatan data plat nomor *realtime* pada BBM Subsidi, maka saat *offline* tidak diperbolehkan berjualan.

Metode HTB dipilih oleh penulis untuk memecahkan masalah diatas, Metode HTB adalah Satu dari banyak pendekatan yang sanggup dipergunakan guna meningkatkan efisiensi manajemen lalu lintas data. Pendekatan ini memungkinkan pengelolaan lalu lintas data dengan membagikan prioritas berdasarkan hirarki yang ditentukan, sehingga aplikasi atau layanan yang kritis mendapatkan alokasi *bandwidth* yang optimal.

Jaringan Digitalisasi SPBU menuntut jaringan internet yang stabil dan selalu terkoneksi, tetapi kenyataannya terkadang jaringan internet sanggup saja mendapati putus koneksi / bermasalah, Pendekatan *Auto Failover* menjadi solusi yang potensial guna mengatasi kendala ini, guna meningkatkan ketahanan jaringan, diperlukan suatu mekanisme otomatis yang sanggup mengatasi kegagalan secara cepat dan efisien, dengan secara otomatis beralih ke jalur alternatif saat terjadi gangguan pada jalur utama.

Mekanisme *Quality of Service* dalam penelitian ini juga dibutuhkan untuk melakukan kontrol & menguji hasil metode Hirarki *token bucket & Auto Failover*. Apakah akan berjalan dengan baik menggunakan standar perhitungan *packet loss, jitter, delay & throghput*. Penelitian ini sangat penting dan sangat dibutuhkan guna mengoptimalkan jaringan digitalisasi SPBU 4453313 Purbalingga lewat penerapan Pendekatan HTB pada manajemen lalu lintas data dan Pendekatan *Auto Failover* guna meningkatkan ketahanan jaringan. Dengan demikian, diharapkan sanggup meningkatkan kualitas layanan SPBU, mengoptimalkan pemanfaatan *bandwidth*, dan membagikan pengalaman yang lebih baik bagi konsumen.

METODE PENELITIAN

Pada penelitian ini penulis memakai pendekatan pengumpulan data statistik guna menunjang kualitas hasil Optimasi jaringan digitalisasi SPBU 4453313 Purbalingga. Penulis menggunakan pendekatan pengambilan data statistik, untuk bagaimana data diambil Penulis melakukan pengambilan data pada PC Server Digitalisasi SPBU menggunakan *tools* Wireshark untuk mendapatkan datashet kualitas jaringan pada Digitalisasi SPBU 4453313 Purbalingga dimana dari data tersebut akan di olah untuk diukur *Quality of Service* dengan

menggunakan beberapa acuan seperti jitter, paket *loss*, *throughput*, dan *Delay* sebelum dan sesudah diterapkan metode HTB & *Auto failover* untuk mengukur efektifitas kedua metode tersebut.

Studi ini akan membahas tentang bagaimana mengoptimasi jaringan digitalisasi SPBU dengan menerapkan dua pendekatan, yakni pendekatan HTB dan pendekatan *Auto Failover*. Langkah awal penelitian dijalankan dengan identifikasi kebutuhan jaringan, dengan cara melakukan analisa pada data jaringan yang ada pada SPBU 4453313 Purbalingga. Lalu melakukan implementasi pendekatan HTB dan *Auto Failover* dengan memprioritaskan sektor yang perlu di lakukan manajemen bandwidth. Pada penelitian ini instrumen yang dipergunakan pada penelitian yakni memakai PC Server Dell optiplex core i5 ram 8 GB dan beberapa *tools* pembantu misal mikrotik.

HASIL DAN PEMBAHASAN

Untuk melakukan implementasi HTB penulis disini melakukan *coding* berbasis *javascript*, dengan menggunakan node js sebagai evnirontmentnya, untuk langkah pertama yaitu install Node.js. Digunakan untuk dasar pembuatan website, bahasa yang digunakan adalah javascript. Cara instalasinya dengan melakukan download melalui situs resmi node.Js, node.Js digunakan untuk melakukan eksekusi *code javascript* di sisi server, lalu perlu juga install next.js sebagai framework dari node Js.

Yang pertama penulis lakukan adalah pembuatan API untuk melakukan interaksi dengan router menggunakan website, jadi API disini digunakan untuk jembatan dari website dan *router*, dalam *script* diatas tercantum IP, User & password dari router yang akan dihubungkan.

```
import { RouterOSClient, RosApiClient } from "routeros-client";

const rosClient = new RouterOSClient({
  host: "192.168.1.2",
  user: "admin",
  password: "password",
  keepalive: true,
});

/** @type {Promise<RosApiClient>} */
export const getClient = async ({ keepalive = true }) => {
  rosClient.setOptions({ keepalive });

  if (!rosClient.isConnected()) {
    return rosClient.connect();
  }

  return rosClient.api();
};
```

Gambar 1. API

Langkah kedua pembuatan `route.js`, dalam *script* ini digunakan oleh penulis untuk menangkap sumber koneksi yang terhubung ke mikrotik, gunana adalah pada website nanti akan tampil ada berapa koneksi yang akan terkoneksi ke *router* tersebut, dan ditanamkan *script* juga untuk *auto failover*, jadi saat koneksi utama mengalami *loss* akan otomatis terhubung ke koneksi backup.

Queue digunakan untuk mendapatkan *client* yang terkoneksi ke *router* mikrotik, jadi setelah melakukan koneksi ke mikrotik, dan melakukan pembacaan koneksi yang terdapat pada *router*, langkah berikutnya adalah pembuatan pembacaan *queue* yang terkoneksi, jadi nanti pada website dapat tampil seluruh user yang terkoneksi ke *router* tersebut.

```

import { NextResponse } from "next/server";
import { getClient } from "utils";
import HttpException from "~/libs/httpException";

export const dynamic = "force-dynamic";
export async function GET(request) {
  try {
    const client = await getClient({ keepalive: false });
    const menu = client.menu("/queue simple");

    let result = null;
    await menu
      .where({ disabled: false, parent: "Total" })
      .json()
      .then((response) => {
        if (!response) {
          throw new HttpException({ status: 404, message: "Queue not found" });
        }

        result = response;
      })
      .catch((error) => {
        throw error;
      });

    return NextResponse.json({ success: true, data: result });
  } catch (error) {
    console.log(error);

    let message = "Something went wrong";
    let status = 500;

    if (error instanceof HttpException) {
      message = error.message;
      status = error.status;
    }

    return NextResponse.json({ success: false, error: { message } }, { status });
  }
}

```

Gambar 2. Queue

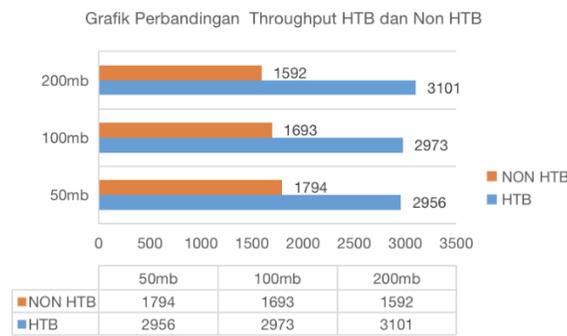
Pada langkah selanjutnya adalah pembuatan *list* menu yang akan di konfigurasi pada website dan akan ditampilkan, karena pada penelitian kali ini menggunakan metode HTB, maka menu yang akan ditampilkan adalah *max limit*, *limit at*, dan *priority*. Selanjutnya adalah melakukan pembuatan *login page*, ini digunakan untuk menampilkan halaman *login*, dan pembuatan *username* admin, jadi website lebih aman karena ada *login page*, tidak seluruh orang dapat mengakses.

Selanjutnya adalah pembuatan *page home*, pada halaman ini penulis ingin menampilkan seluruh *client* yang terkoneksi ke *router*, untuk melakukan monitoring *client* yang terkoneksi. Langkah selanjutnya adalah Download `express js` digunakan sebagai *server* website yang kana dibuat, untuk perintah menjalankan server. Soket server digunakan untuk mengirim *real time data qos & route*, sehingga data akan terupdate setiap waktu yang penulis inginkan, misal di set 10 detik, nanti data akan melakukan perubahan setiap 10 detik. Service digunakan untuk mengambil data realtime dari mikrotik, sehingga server akan mengambil data secara terus menerus melalui perintah tersebut. Pada coding ini digunakan untuk menampilkan route yang aktif sehingga dapat terlihat jaringan apa saja yang terkoneksi ke

router, dan pada *coding* ini juga di berikan switch jaringan sehingga dapat juga melakukan pergantian jaringan. Pada menu kalkulator digunakan untuk penulis untuk melakukan perhitungan *loss*, *delay*, *jitter* dan *throughput*, sehingga dapat dibandingkan dengan jelas sebelum dan sesudah penelitian.

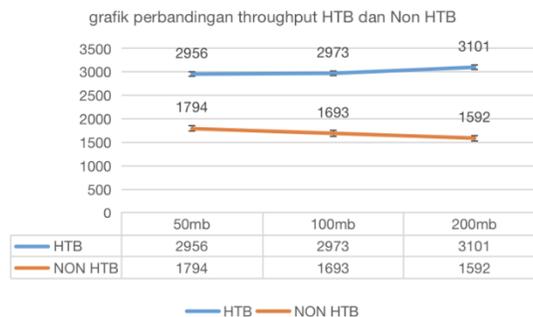
3.1. Parameter *Throughput*

Pengujian ini dilakukan menggunakan 3 skenario dengan jumlah paket yang berbeda beda ukurannya, diantaranya paket pertama yaitu 50mb,100mb dan terakhir 200mb. Pengujian dilakukan dari sebelum dan sesudah di implementasikanya HTB, hasil perbandingan dapat kita lihat pada gambar 3 seperti dibawah ini.



Gambar 3. Grafik *bar* pengujian *throughput*

Pada gambar 3 terlihat perbandingan grafik diagram batang dari ketiga indikator pengujian, dimana batang berwarna biru menunjukan indikator sebelum di implementasikanya HTB, dan untuk indikator berwarna orange menunjukan setelah implementasi HTB, dapat dilihat *throughput* masing masing indikator dimana batang berwarna orange dengan angka 1592,1693,1794 lebih pendek yang artinya setelah di implementasikan HTB dapat melakukan optimasi ke jaringan Digitalisasi SPBU 4453313.



Gambar 4. Grafik *line* pengujian *throughput*

Pada gambar 4 terlihat perbandingan grafik diagram garis dari ketiga indikator pengujian, dimana garis berwarna biru menunjukkan indikator sebelum di implementasikannya HTB, dan untuk garis berwarna *orange* menunjukkan setelah implementasi HTB, dapat dilihat *throughput* masing masing indikator dimana garis berwarna *orange* dengan angka 1794,1693,1592 lebih rendah yang artinya setelah di implementasikan HTB dapat melakukan optimasi ke jaringan Digitalisasi SPBU 4453313.

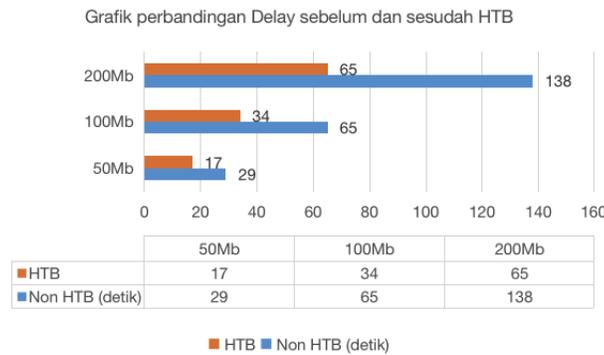
Tabel 1. Pengujian *Throughput*

Jumlah paket	HTB	Non HTB	Optimasi
50Mb	2956kbps	1794kbps	165%
100Mb	2973kbps	1693kbps	176%
200Mb	3101kbps	1592kbps	195%

Dari tabel diatas terlihat bahwa hasil setelah implementasi HTB dapat meningkatkan nilai *throughput* sebesar 165-195% sesuai dengan paket data yang sudah di tes.

3.2. Parameter Delay

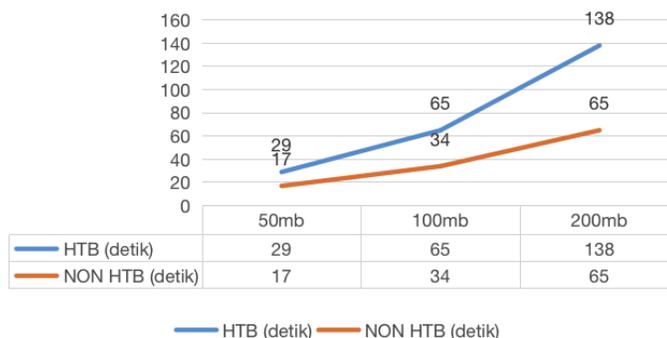
Pengujian selanjutnya menggunakan *parameter Delay* dilakukan dengan melakukan pengujian menggunakan 3 skenario berbeda berdasarkan jumlah paket yang telah ditentukan diantaranya: 50mb,100mb dan 200mb, pengujian dilakukan sebelum dan sesudah diterapkannya metode HTB, Delay yang dihasilkan dapat dilihat pada gambar 5 dan 6.



Gambar 5. Grafik *bar* pengujian *Delay*

Pada gambar diatas terlihat perbandingan grafik diagram batang dari ketiga indikator pengujian, dimana batang berwarna biru menunjukkan indikator sebelum di implementasikannya HTB, dan untuk indikator berwarna *orange* menunjukkan setelah implementasi HTB, dapat dilihat *delay* masing masing indikator dimana batang berwarna *orange* lebih pendek yang artinya setelah di implementasikan HTB dapat melakukan optimasi ke jaringan Digitalisasi SPBU 4453313.

Grafik Perbandingan Delay HTB dan non HTB



Gambar 6. Grafik line pengujian Delay

Pada gambar di atas terlihat perbandingan grafik diagram garis dari ketiga indikator pengujian, dimana garis berwarna biru menunjukkan indikator sebelum di implementasikanya HTB, dan untuk garis berwarna *orange* menunjukkan setelah implementasi HTB, dapat dilihat delay masing masing indikator dimana garis berwarna *orange* lebih rendah yang artinya setelah di implementasikan HTB dapat melakukan optimasi ke jaringan Digitalisasi SPBU 4453313.

Tabel 2. Penujian Delay

Jumlah paket	Non HTB (detik)	HTB	Optimasi
50Mb	29	17	171%
100Mb	65	34	191%
200Mb	138	65	212%

Dari grafik dan tabel diatas terlihat bahwa hasil setelah implementasi HTB dapat meningkatkan nilai Delay sebesar 171-212% sesuai dengan paket data yang sudah di tes.

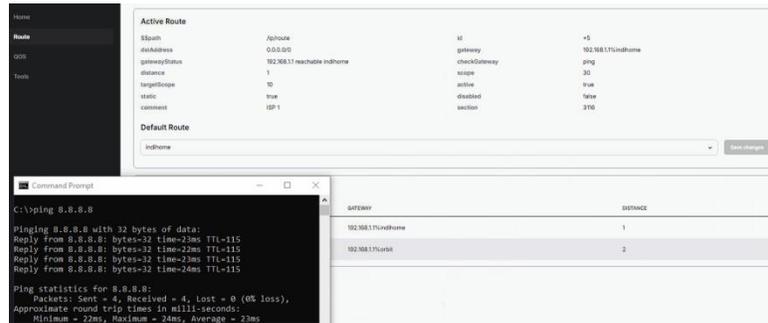
3.3. Parameter Loss

Tabel 3. Pengujian Loss

Jumlah paket	HTB (detik)	Non HTB	Optimasi
50Mb	0	0	0%
100Mb	0	0	0%
200Mb	0	0	0%

Pengujian selanjutnya menggunakan parameter *Loss* dilakukan dengan melakukan pengujian menggunakan 3 skenario berbeda berdasarkan jumlah paket yang telah ditentukan diantaranya: 50mb,100mb dan 200mb, pengujian dilakukan sebelum dan sesudah diterapkanya metode HTB, Delay yang dihasilkan dapat dilihat pada tabel 3.

3.4. Pengujian *Auto Failover*



Gambar 7. Pengujian *failover* ISP 1

Metode *failover* bertujuan apabila salah satu link ISP bermasalah/*down* maka pemakaian koneksi internet dapat di-*backup* dengan link ISP lainnya. Kondisi yang dibahas pertama ketika link ISP 1 dengan link ISP 2 berada dalam kondisi hidup. *User* mencoba melakukan *ping* pada dns google yaitu 8.8.8.8 maka terlihat *user* terhubung ke internet dengan link ISP1 dan ISP2 berada dalam keadaan aktif secara bersamaan. Kondisi selanjutnya akan dilakukan percobaan saat salah satu link ISP bermasalah / *down* maka pemakaian koneksi internet dapat di-*backup* dengan link ISP lainnya.

Pada pengujian ping 8.8.8.8 diatas dilakukan percobaan saat menggunakan ISP utama yaitu indihome, berhasil tanpa kendala melakukan ping dns google 8.8.8.8, pada percobaan kedua, akan dilalkukan pencabutan link ISP utama.



Gambar 8. cabut *ISP* Utama

Pada pengujian berikutnya dilakukan cabut *port* ISP utama, untuk menguji *failover* apakah berjalan dengan baik atau tidak, dan menghitung *downtime* peralihan dari ISP 1 ke ISP 2.



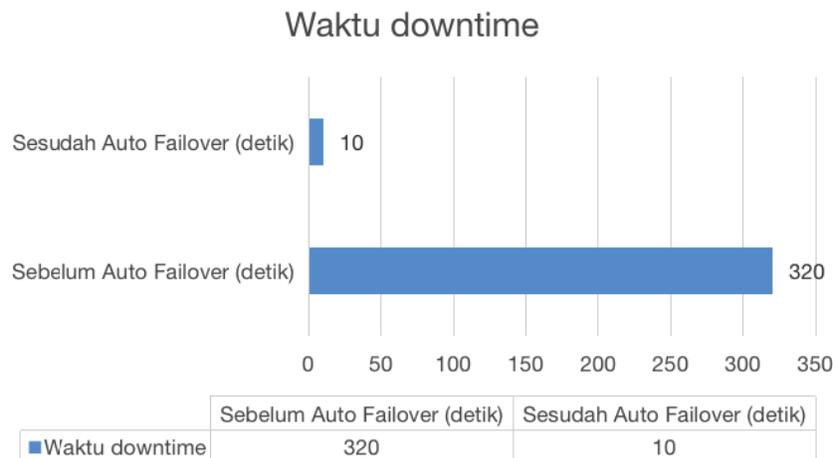
Gambar 9. percobaan *ISP 2*

Setelah dilakukan pengujian pencabutan, dapat kita lihat pada gambar 9 dimana saat melakukan ping dns google ke 8.8.8.8 sempat mengalami *down* selama 10s, dan kembali konek dan langsung beralih ke ISP 2 yaitu *orbit*, sehingga dapat disimpulkan *auto failover* berhasil, jika tanpa adanya auto failover maka user SPBU 4453313 harus melakukan *setting* manual dan memerlukan waktu yang lama sehingga dapat mengganggu operasional SPBU, tetapi dengan adanya *auto failover* waktu peralihan dari ISP 1 ke ISP 2 menjadi sangat singkat yaitu 10s.

Tabel 4. Tabel Hasil Pengujian *Failover*

Indikator	Sebelum Implementasi	Sesudah Auto Failover	Optimasi
Waktu Downtime	320	10	Berhasil menyingkat waktu sebanyak 310 detik

Dari tabel 4.4 dapat di dapati bahwa dengan adanya *auto failover* dapat menyingkat 310 detik, sehingga uji coba *auto failover* pada penelitian kali ini terbukti dapat mengoptimasi jaringan Digitalisasi SPBU 4453313 Purbalingga.



Gambar 10. Waktu *Downtime Failover*

Grafik diatas menunjukkan perbandingan grafik dari sebelum dan sesudah implementasi *auto failover*, dimana indikator berwarna biru setelah *auto failover* di terapkan jauh lebih efektif dibandingkan indikator berwarna biru disisi sebelum *auto failover*.

KESIMPULAN

Berdasarkan hasil rangkaian uji coba dan analisis penelitian yang sudah dilakukan, maka dapat disimpulkan bahwa (1) Penerapan manajemen *bandwidth* dengan metode HTB telah berhasil memaksimalkan *bandwidth*, sehingga kualitas pelayanan *bandwidth* menjadi meningkat, dapat dilihat dari perbandingan sebelum dan sesudah implementasi HTB melalui 3 indikator *throughput delay & loss* (2) dengan penerapan metode *auto failover* maka dapat mencegah terjadinya gangguan internet saat ISP utama mengalami loss jaringan, serta mempercepat waktu pergeseran dari ISP 1 ke ISP 2 yang sebelumnya dilakukan secara manual sekarang menjadi tersistem, saat downtime terjadi maka akan langsung melakukan failover ke ISP 2.

SARAN

Metode HTB dan *Auto failover* memang terbukti sangat efektif dalam meningkatkan QoS suatu jaringan tetapi masih dapat di kembangkan lebih baik lagi dengan mengkombinasikan dengan metode QoS lainnya.

DAFTAR PUSTAKA

- Edy, Y. Syarifudin, S. I. A. 2022. "Implementasi Redundant Link Untuk Meminimalisir Downtime Dengan Metode Failover."
- Eko Ari Puspita H. 2019. "Optimasi Manajemen Bandwidth Memakai Pendekatan HTB (Hierarchical Token Bucket) Pada Dinas Pariwisata Provinsi Lampung."
- I. Gede, A. Y. U., I. Gusti, N. A. J. S., Lie, J. M. J. 2020. "Optimasi Internet Di Dinas Kesehatan Provinsi Bali Dengan Menggunakan Hierarchical Token Bucket."
- Irfa, I. M., Dwiki, J. n.d. 2021 "Perancangan Jaringan Ethernet Link Dengan Menggunakan Teknologi Link Aggregation Dan Auto Failover."
- Mohamad Badrul, Akmaludin. 2019. "Implementasi Automatic Failover Memakai Router Mikrotik Guna Optimasi Jaringan."
- Valia, Y. P. A., M. Dermawan, M. n.d. 2023 "Analisis Quality Of Service (QoS) Jaringan Internet Universitas Menggunakan Metode Hierarchical Token Bucket (HTB)."
- Karlinda, A E., Azizi P., & Sopali, M. F. 2021. "Pengaruh pengalaman kerja, prestasi kerja,

pendidikan dan pelatihan terhadap pengembangan karir pada PT.PLN (Persero) Kota Padang Rayon Kuranji ”

Sofana Iwan 2012. “Cisco CCNA & Jaringan komputer Bandung Informatika Bandung”

Madcoms. 2010. “Sistem Jaringan Komputer untuk pemula, Andi Offset, Yogyakarta.”

James A. O'Brien, G. M. 2011. “Management information System 10th ed. New York.”

Utami 2020. “Analisis perbandingan QoS Jaringan Internet Berbasis Wireless pada layanan internet service provider (ISP) Indihome dan firstmedia.”

Wisnu Saputra 2024. “Rancang Bangun Portal Magang berbasis Website menggunakan Node.Js (studi kasus : fasilkom UNSIKA).”

Renaldi & Pramudhita Ferdiansyah 2019. “Analisis perbandingan parameter QoS standar TIPHON pada jaringan Nirkabel dalam penerapan Metode PCQ.”

Akhmad Fathurohman, Asdani Kindarto, Reiza Maulana 2022. “implementasi QoS HTB pada manajemen bandwidth Internet pada jaringan komputer di laboratorium Komputer terintegrasi.”

Akhmad Fathurohman 2022. “Buku Jaringan Komputer & Komunikasi Data”