**PETIR: Jurnal Pengkajian dan Penerapan Teknik Informatika**
Vol. 17, No. 1, Maret 2024, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v17i1.2320

# Optimisasi Beban *VPN* menggunakan *WireGuard* pada koneksi *Multi WAN*

*Oky Tria Saputra[1*)]; Widyastuti Andriyani[1]; Bambang Purnomosidi Dwi Putranto[1]*

1. Jurusan Magister Teknologi Informasi, Universitas Teknologi Digital Indonesia, Jl. Raya Janti Jl. Majapahit No.143, Jaranan, Banguntapan, Kec. Banguntapan, Kabupaten Bantul, Daerah Istimewa Yogyakarta 55198, Indonesia

*[*)]Email: okytria@gmail.com*

*Received: 1 Januari 2024 | Accepted: 17 Januari 2024 | Published: 7 Juni 2024*

***ABSTRACT***

*Currently, the internet has become a need for many parties, both individuals and companies. If there is a company that has several Branch Offices, it needs data communication, be it voice, video, or other data to the Head Office. One way to connect several networks in different locations is to create Virtual Private Networks (VPN) over the Internet. What needs to be considered is the need for high security and good speed when sending data over the Internet. There are several methods that can be used to create a VPN such as OpenVPN, IPSec, and WireGuard, but the best and fastest right now is WireGuard. To avoid Single Point of Failure (SPoF), it is necessary to have a main VPN line and a backup path, and to maximize data delivery, it is necessary to do Load Balance using Equal Cost Multiple Path (ECMP) to divide the load on both paths so that both links are used simultaneously and load sharing on both links becomes more effective. This research uses the PnetLab simulator application to test it with several methods such as FTPS to determine data transmission speed, delay, packet loss. The result is that Wireguard Loadbalance is better than previous VPN technologies such as OpenVPN.*

***Keywords:*** *ECMP, loadbalance, SPoF, Wireguard, VPN*

***ABSTRAK***

*Saat ini internet sudah menjadi kebutuhan banyak pihak baik perorangan maupun perusahaan. Jika ada sebuah perusahaan yang memiliki beberapa Kantor Cabang perlu nya komunikasi data baik itu voice, video, atau pun data lain nya ke Kantor Pusat. Salah satu cara menghubungkan beberapa jaringan yang berbeda lokasi dengan membuatkan Virtual Private Networks (VPN) melalui Internet. Yang perlu diperhatikan adalah perlu nya keamanan yang tinggi serta kecepatan yang bagus ketika mengirimkan data melewati Internet. Ada beberapa metode yang bisa digunakan untuk membuat VPN seperti OpenVPN, IPSec, WireGuard. Tapi yang terbaik dan tercepat saat ini adalah WireGuard. Untuk menghindari Single Point of Failure (SPoF) maka perlu nya jalur VPN utama dan jalur backup. Dan untuk memaksimalkan pengiriman data, maka perlu nya lakukan Load Balance menggunakan Equal Cost Multiple Path ( ECMP ) untuk membagi beban dikedua jalur tersebut. Sehingga kedua link tersebut digunakan secara bersamaan sehingga pembagian beban dikedua link menjadi lebih efektif. Penelitian ini menggunakan aplikasi simulator PnetLab untuk menguji nya dengan beberapa metode seperti File Transfer Protocol Secure (FTPS) untuk mengetahui kecepatan pengiriman data, delay, packetloss. Hasilnya Loadbalance Wireguard lebih baik dibandikan teknologi VPN sebelumnya seperti OpenVPN.*

***Kata kunci:*** *ECMP, loadbalance, SPoF, Wireguard, VPN*

Vol. 17, No. 1, Maret 2024, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v17i1.2320

## 1. PENDAHULUAN

Pertumbuhan internet di indonesia terus bertambah, menurut survey yang dilakukan Asosiasi Penyelenggara Internet di Indonesia (APJII) mencapai 210.026.769 jiwa dari total populasi 272.682.600 pada 2021. [2] Internet menjadi kebutuhan manusia untuk bisa berkomunikasi ke siapapun dan dimanapun, baik itu komunikasi menggunakan voice, videocall, game, email, dan lain nya. Salah satu manfaat internet dirasakan oleh perusahaan yang memiliki beberapa kantor yang berbeda lokasi secara geografis dan ingin berkomunikasi melalui internet.

Ada beberapa cara untuk menghubungkan kantor pusat dengan kantor cabang nya seperti *MetroEthernet*, *IPVPN, MPLS, SD-WAN*, dan Internet *VPN*. Tapi untuk solusi yang *cost-effective* adalah Internet *VPN*. Selama kedua lokasi memiliki internet, maka bisa menggunakan Internet *VPN* tanpa penambahan service seperti lain nya.

Untuk Internet *VPN* yang *opensource* ada beberapa yang sering digunakan seperti *OpenVPN*, *IPSec*, *WireGuard*, tapi dari semua itu, *WireGuard* merupakan salah satu teknologi *OpenSource* yang *free* dan hasil lebih baik. Untuk menghindari *Single Point of Failure (SPOF)* dalam *design* jaringan, maka perlu adanya backup ketika jalur pertama mati.

Pada penelitian ini akan dibahas bagaimana performa *WireGuard* ketika melakukan pengiriman data dari kantor pusat ke kantor cabang, baik satu jalur atau dua jalur, dan akan dihitung hasil *throughput*, *delay*, *jitter*, *packetloss* pada simulator PnetLab.

Penelitian terkait *loadbalance vpn* telah dilakukan oleh Nur Fatih Nadhirah Norazlan [5] membandingkan *loadbalance* dan tidak *loadbalance* pada *VPN GRE Tunnel*, tapi *GRE* kurang aman dibandingkan *WireGuard*. Lalu penelitian lain dilakukan oleh Oky Tria Saputra [2] untuk membandingkan *loadbalance vpn* dan tidak *loadbalance* pada *OpenVPN*. Tapi dari sisi kecepatan dan keamanan *WireGuard* lebih baik. Penelitian lain dilakukan oleh Lukas Osswald [6] membandingkan antara *OpenVPN, IPSec, Wireguard*, ternyata hasil nya *WireGuard* lebih baik dibanding ketiganya. Penelitian VPN lain diimplementasikan oleh Budi Santoso [1] untuk melihat keamanan pada *VPN L2TP* di salah satu perusahaan. Penelitian lain dilakukan oleh Ilmalik Muhammad Alviendra [3] untuk menghubungkan alat *IoT* menggunakan *VPN PPTP* dan *L2TP*, tapi *WireGuard* lebih aman dan cepat dibanding keduanya. Penelitian untuk keamanan data melalui internet dilakukan juga oleh Kingsley A. Ogudo [4] menggunakan *GRE* dan *IPSec* untuk mengirimkan data dan di analisa dengan *Wireshark*, *GRE* dan *IPSec less secure* dibanding *Wireguard*. Penelitian terkait *Wireguard* dilakukan oleh Samu Saukkonen [7] ketika ada nya Covid-19 banyak *staff* bekerja di rumah, dan menggunakan *wireguard* salah satu opsi yang digunakan kala itu. Untuk mengetahui performa terbaik, peneliti Erik Dekker [8] melakukan penelitian terkait *VPN* mana yang terbaik, dan ternyata *Wireguard* merupakan salah satu yang terbaik dalam hal initiating connection. Peneliti Steven Mackey [9] melakukan perbandingan antara *OpenVPN* dan *WireGuard*, hasilnya *WireGuard* hasil nya lebih baik dari *OpenVPN*. Peneliti Peter Wu [10] meneliti di bagian keamanan pada *Wireguard*, tetep ada celah untuk *denial of services* nya. Peneliti Roberta Avanzato[11] meneliti menggunakan *Smart Bonding VPN* untuk menghubungkan drone dengan di *remote* jarak jauh dengan loadbalance *VPN*. Peneliti TIAZ R. FAKHRURRASI[12] melakukan penelitian loadbalance menggunakan *ECMP* membuat kedua jalur digunakaan bersamaan sehingga lebih efektif. Peneliti Aji Triwerdaya [13] melakukan penelitian *ECMP* pada *OSPF* dan *BGP* mana yang terbaik, ternyata *BGP* lebih baik dari *OSPF*. Peneliti Muhammad Fikri [14] melakukan *loadbalance VPN* antara *MPLS* dan *IPSec* untuk menghindari *VPN*. Peneliti Antonio Francesco Gentile [15] melakukan pengujian pada *OpenWRT*, *WireGuard* lebih baik hasilnya daripada *OpenVPN*. Dari seluruh penelitian yang relevan, belum ada penelitian terkait yang membahas tentang Optimisasi Beban VPN menggunakan WireGuard dengan Loadbalance *ECMP*. State of the art pada

Vol. 17, No. 1, Maret 2024, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v17i1.2320

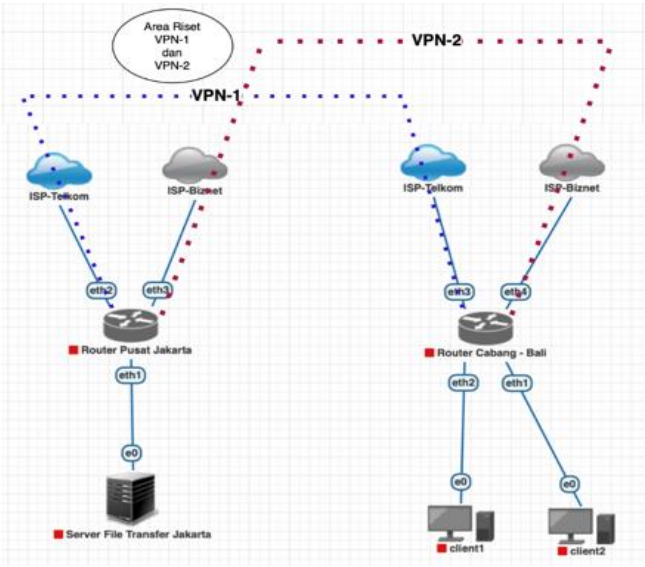
penelitian ini yaitu menguji *loadbalance vpn* dengan *WireGuard* yang lebih baik dan bagus dibandingkan *OpenVPN* yang dilakukan oleh penelitian sebelumnya.

## 2. METODE/PERANCANGAN PENELITIAN

Metode Penelitian adalah tahapan dalam suatu penelitian. Penelitian ini dibagi menjadi 5 bagian mulai dari design topologi jaringan, lalu *hardware* dan *software* yang digunakan, menggunakan 30 data set dan diuji dengan model TIPHON (1998) untuk analisa *Quality of Service (QoS)* untuk menghitung nilai *Throughput, Delay, Packet Loss, Jitter* untuk membandingkan kedua hal.

### 2.1 Design Topologi Jaringan

Untuk melakukan penelitian ini menggunakan aplikasi simulator jaringan yang bernama *PnetLab*, peneliti akan membuatkan 2 *VPN* dengan *WireGuard* dan melakukan pengujian. Masing2 jalur memiliki bandwith 5Mbps.




**Gambar 1.** Topologi Pengujian WireGuard

Pada gambar 1, peneliti membuat design topologi jaringan menggunakan 1 router yang terhubung ke 2 ISP di lokasi Pusat Jakarta dan Cabang di Bali. Peneliti akan membuatkan *VPN* sebanyak 2 untuk melakukan pengujian.

### 2.2. Spesifikasi Hardware

Untuk melakukan penelitian ini, berikut spesifikasi *hardware* yang dibutuhkan, peneliti menggunakan server untuk pengujian, bisa juga pake workstation.

**Tabel 1.** Spesifikasi *hardware*

| No | Nama Perangkat | Jumlah | Spesifikasi |
|---|---|---|---|
| 1 | Fujitsu Server | 1 | CPU : 8vCPU core<br>RAM : 64GB<br>HDD : 300GB |

Vol. 17, No. 1, Maret 2024, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v17i1.2320

### 2.3. Spesifikasi Software

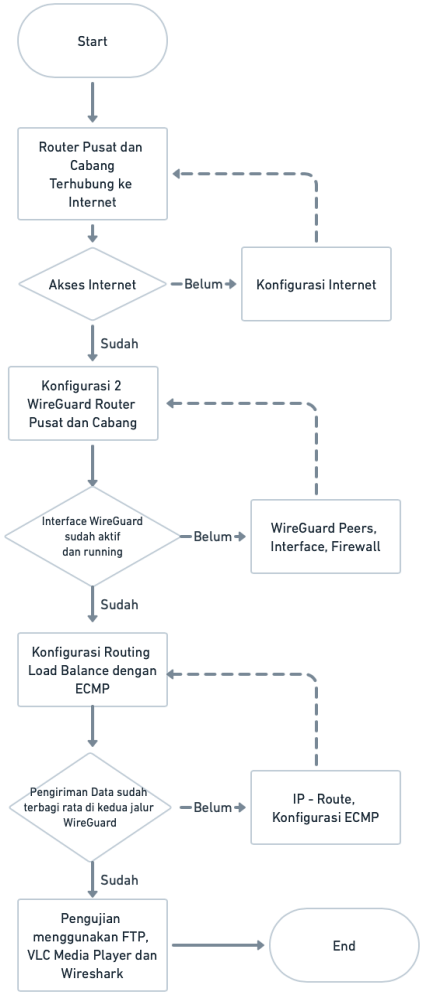
Untuk melakukan penelitian ini, berikut spesifikasi *hardware* yang dibutuhkan, peneliti menggunakan *server* untuk pengujian, bisa juga pake *workstation.*

**Tabel 2.** Spesifikasi *SOFTWARE*

| No | Nama Perangkat | Jumlah | Spesifikasi |
|---|---|---|---|
| 1 | OS Ubuntu Linux 64bit | 1 | Untuk Pnetlab |
| 2 | RouterOS pada MikroTik | 2 | v7.8 |
| 3 | OS Windows pada Virtual PC | 3 | Windows 10 |
| 4 | *File Zilla FTP Server (Secure)* | 1 | 3.66.1 |
| 5 | *Wireshark* | 1 | 3.4.4 |
| 6 | PnetLab Cloud ISP | 4 | 5Mbps |

### 2.4. Analisis dan Perancangan Sistem

Berikut proses untuk melakukan perancangan sistem dan pengujian yang dilakukan peneliti.




**Gambar 2.** Analisa perancangan sistem

Vol. 17, No. 1, Maret 2024, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v17i1.2320

Untuk melakukan pengujian, peneliti menggunakan dataset 30 yang berbeda untuk mendapatkan hasil yang maksimal. Berikut *dataset* yang digunakan dengan total 30 dataset berupa video yang *variatif* :

**Tabel 3.** Data Yang Akan Diuji

| No | Nama File | File Type | Extension | Size (KB) |
|---|---|---|---|---|
| 1 | 100Mb.dat | System | .dat | 102.400 |
| 2 | pdf 10.5MB.pdf | Dokumen | .pdf | 10.822 |
| 3 | UTDI Profile.mp4 | Video | .mp4 | 85.758 |
| 4 | vlc-3.0.18-win32.exe | Aplikasi | .exe | 41.730 |
| 5 | Beach.avi | Video | .avi | 9.677 |
| 6 | Surfing.mkv | Video | .mkv | 69.548 |
| 7 | Trim1.mp4 | Video | .mp4 | 10.369 |
| 8 | Trim2.mp4 | Video | .mp4 | 7.657 |
| 9 | Trim3.mp4 | Video | .mp4 | 5.657 |
| 10 | Trim4.mp4 | Video | .mp4 | 5.292 |
| 11 | Trim5.mp4 | Video | .mp4 | 7.137 |
| 12 | Trim6.mp4 | Video | .mp4 | 13.573 |
| 13 | Trim7.mp4 | Video | .mp4 | 12.456 |
| 14 | Trim8.mp4 | Video | .mp4 | 6.662 |
| 15 | Trim9.mp4 | Video | .mp4 | 7.320 |
| 16 | Trim10.mp4 | Video | .mp4 | 10.146 |
| 17 | Trim11.mp4 | Video | .mp4 | 11.310 |
| 18 | Trim12.mp4 | Video | .mp4 | 6.900 |
| 19 | Trim13.mp4 | Video | .mp4 | 17.927 |
| 20 | Trim14.mp4 | Video | .mp4 | 10.550 |
| 21 | Trim15.mp4 | Video | .mp4 | 8.036 |
| 22 | Trim16.mp4 | Video | .mp4 | 15.154 |
| 23 | Trim17.mp4 | Video | .mp4 | 9.231 |
| 24 | Trim18.mp4 | Video | .mp4 | 12.113 |
| 25 | Trim19.mp4 | Video | .mp4 | 10.569 |
| 26 | Trim20.mp4 | Video | .mp4 | 8.385 |
| 27 | Trim21.mp4 | Video | .mp4 | 8.278 |
| 28 | Trim22.mp4 | Video | .mp4 | 7.722 |
| 29 | Trim23.mp4 | Video | .mp4 | 8.365 |
| 30 | video last.mp4 | Video | .mp4 | 196.045 |

### *2.5. Quality of Services*

*TIPHON* (*Telecommunications and Internet Protokol Harmonization over Network*) adalah inisiatif yang diusulkan oleh ETSI untuk mendukung pasar komunikasi suara dan multimedia terkait antara pengguna jaringan berbasis *IP* dan pengguna jaringan circuit switched. *ETSI (European*

Vol. 17, No. 1, Maret 2024, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v17i1.2320

*Telecommunications Standards Institute)* adalah organisasi Eropa yang didirikan pada tahun 1988 yang bertanggung jawab untuk menetapkan standar teknis telekomunikasi

1) *Throughput*

*Throughput* merupakan jumlah total kedatangan paket yang sukses yang diverifikasi oleh destination selama waktu tertentu dibagi oleh durasi waktu tersebut.

**Tabel 4.** Indeks throughput

| Kategori | *Throughput* | Indeks |
|---|---|---|
| Sangat Bagus | 76% - 100% | 4 |
| Bagus | 51% - 75% | 3 |
| Sedang | 26% - 50% | 2 |
| Buruk | < 25% | 1 |

$$\boldsymbol{Throughput} = \frac{\boldsymbol{Total\ Data\ yang\ diterima}}{\boldsymbol{Total\ Waktu}} \ldots\ldots\ldots (\mathbf{1})$$

*2) Packet Loss*

*Packet loss* adalah banyaknya paket yang hilang selama proses transmisi ke tujuan. Ada beberapa factor yang menyebabkan paket yang hilang ketika dikirim seperti koneksi yang buruk, protocol yang digunakan, dan sebagai nya.

**Tabel 5.** Indeks Packetloss

| Kategori | Packet Loss (%) | Indeks |
|---|---|---|
| Sangat Bagus | 0 | 4 |
| Bagus | 3 | 3 |
| Sedang | 15 | 2 |
| Buruk | 25 | 1 |

$$\boldsymbol{Packet\ Loss} = \frac{\boldsymbol{Paket\ yang\ dikirim - Paket\ data\ diterima}}{\boldsymbol{Paket\ data\ dikirim}} \boldsymbol{x}\ \mathbf{100}\% \ldots\ldots\ldots (\mathbf{2})$$

*3) Delay*

*Delay* merupakan nilai waktu yang digunakan data untuk melakukan pengiriman dari *source* menuju ke *destination*. Dengan parameter *delay* bisa kita melihat *performance* pengiriman data dari titik A ke B.

**Tabel 6.** Indeks Delay

| Kategori | Besar *Delay* (ms) | Indeks |
|---|---|---|
| Sangat Bagus | < 150ms | 4 |
| Bagus | 150ms - 300ms | 3 |
| Sedang | 300ms - 450ms | 2 |
| Buruk | > 450ms | 1 |

$$\boldsymbol{Delay\ rata - rata} = \frac{\boldsymbol{Total\ Delay}}{\boldsymbol{Jumlah\ paket\ yang\ diterima}} \ldots\ldots\ldots (\mathbf{3})$$

**PETIR: Jurnal Pengkajian dan Penerapan Teknik Informatika**
Vol. 17, No. 1, Maret 2024, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v17i1.2320

*4) Jitter*

*Jitter* adalah variasi *delay* perbedaan selang waktu kedatangan antar paket di tujuan. Untuk mengatasi jitter maka paket data yang datang dikumpulkan dulu dalam jitter buffer selama waktu yang telah ditentukan sampai paket dapat diterima pada sisi penerima dengan urutan yang benar.

**Tabel 7**. Indeks JITTER

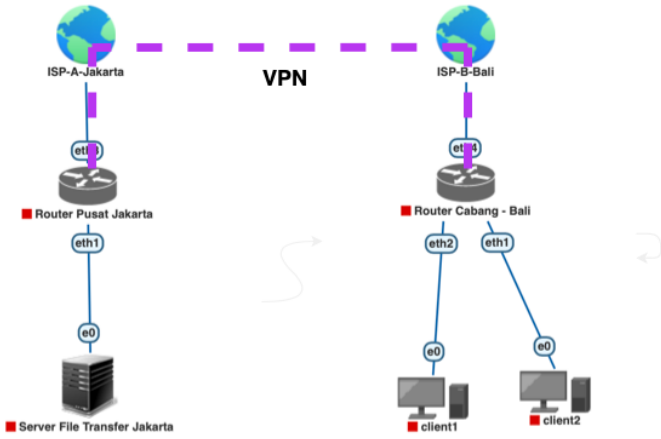
| Kategori | Besar *Jitter* (ms) | Indeks |
|---|---|---|
| Sangat Bagus | 0ms | 4 |
| Bagus | 0ms - 75ms | 3 |
| Sedang | 75ms - 125ms | 2 |
| Buruk | 125ms - 225ms | 1 |

$$\textit{Jitter rata} - \textit{rata} = \frac{\textit{Total variasi delay}}{\textit{Total paket yang diterima}} \ldots\ldots (4)$$

## 3. HASIL DAN PEMBAHASAN

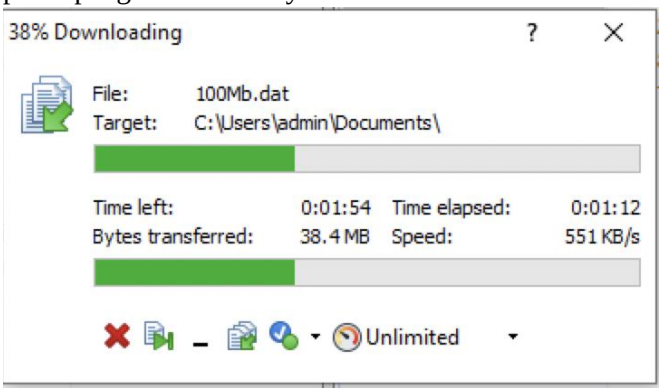
Setelah *VPN* terbentuk dan dilakukan pengujian pengiriman data, jika hanya satu ISP saja yang digunakan maka kecepatan pengiriman data nya hanya sekitar 551KB/s atau sekitar 5Mbps. Karena hanya menggunakan 1 *VPN* saja.

### *3.1. Studi Kasus 1 VPN WireGuard*




**Gambar 3**. Topologi Studi Kasus 1 *VPN*

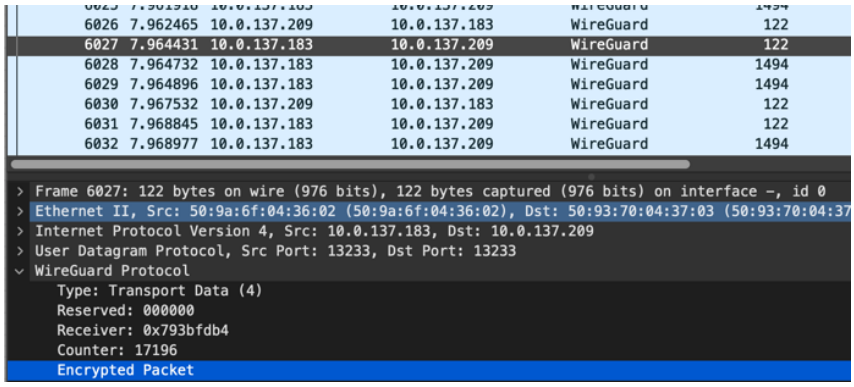
Pada gambar 3, peneliti menggunakan satu *ISP* dan mengkonfigurasi *VPN* hanya satu jalur saja untuk melihat kecepatan pengiriman data nya.




**Gambar 4**. Pengujian pengiriman data melalui FTPS

Vol. 17, No. 1, Maret 2024, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v17i1.2320

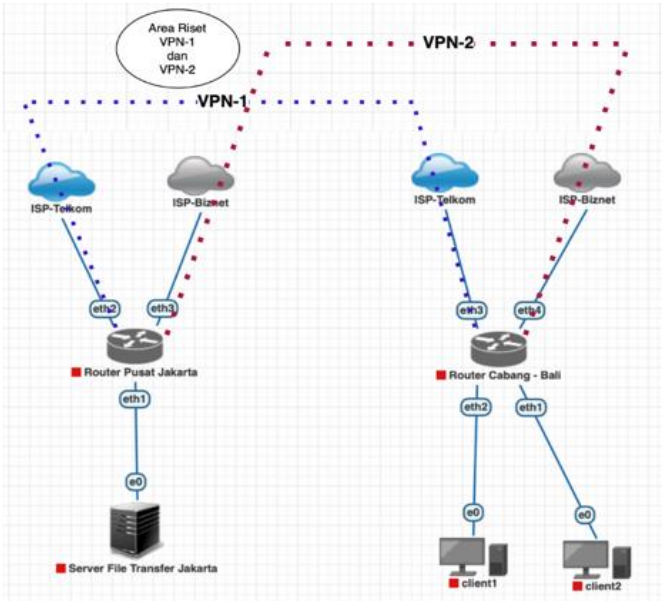
Pada Gambar 4 ketika dilakukukan pengiriman data melalui *VPN* dari server ke client mendapatkan kecepatan sebesar 551KB/s. Selain kecepatan pengiriman data, peneliti juga melakukan pengujian dan analisa dalam hal keamanan enkripsi pengiriman data.




**Gambar 5**. Wireshark Pengecekan Encrypted Packet.

Pada Gambar 5, ketika server melakukan pengiriman data ke client, ketika menggunakan *Packet Analyzer* bisa disimpulkan bahwa pengiriman data dilakukan secara aman dan terenkripsi. Tapi ada kekurangan nya ketika kantor di Jakarta dan di Bali hanya berlangganan satu ISP saja, jika ISP nya bermasalah atau kabel fiber optik nya putus, maka terputus juga komunikasi antara Jakarta dan Bali. Maka dari itu peneliti mengoptimisasi menggunakan teknik *loadbalance* sehingga distribusi pengiriman secara merata dan jika salah satu ada masalah, maka tetap dikirimkan ke jalur satu nya lagi untuk menghindari *Single Point of Failure (SPoF).*

### 3.2. Single Point of Failure (SPoF)



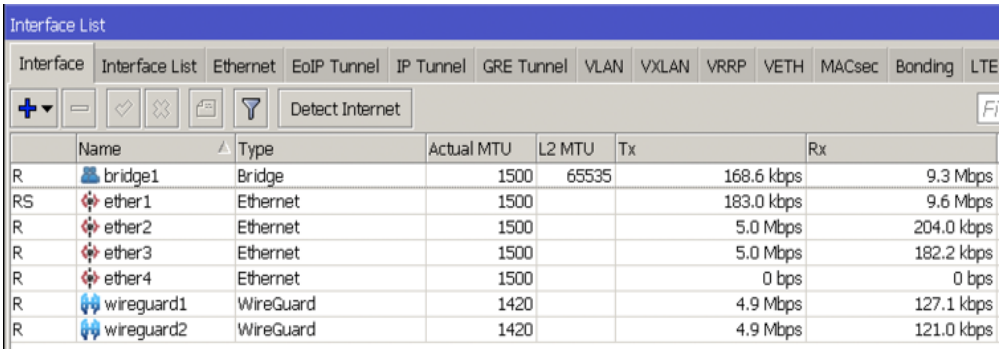

**Gambar 6.** Topologi Pengujian *Single Point of Failure*

Untuk pengujian *Single Point of Failure*, berikut skenario yang dilakukan peneliti. Ada sebuah perusahaan yang berpusat di Jakarta, memiliki kantor cabang di Bali. Untuk menghubungkan kantor cabang di Bali ke Jakarta, diperlukan *Virtual Private Networks* ecara aman dan *reliable*. Awal nya

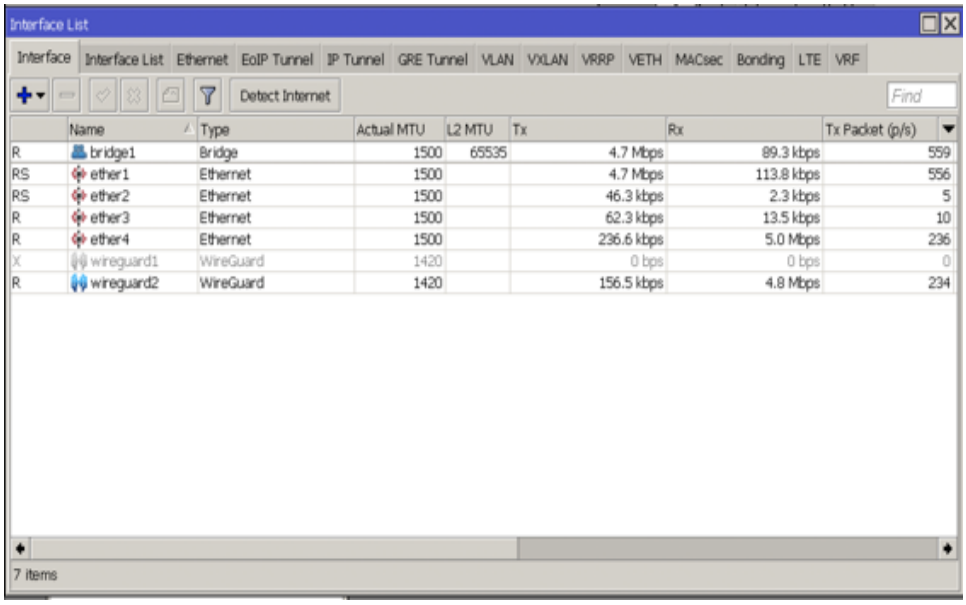
**PETIR: Jurnal Pengkajian dan Penerapan Teknik Informatika**
Vol. 17, No. 1, Maret 2024, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v17i1.2320

Bali hanya memiliki satu ISP saja 5Mbps untuk menghubungkan Kantor Cabang Bali ke Kantor Pusat Jakarta, tetapi ketika satu ISP itu bermasalah maka tidak bisa terjadikomunikasi data antara Kantor Cabang Bali ke Kantor Pusat Jakarta.

Untuk menghindari kejadian tersebut, maka Kantor Cabang Bali memiliki 2 ISP untuk adanya jalur *backup* membuat *VPN* nya dengan 5Mbps. Dengan ada nya *WireGuard* dikonfigurasi secara Loadbalance menghasilkan kecepatanyang tinggi, keamanan yang baik, serta memiliki nilai *delay, jitter, packet loss* yang sedikit dibandingkan tidak mengkonfigurasi secara *Loadbalance*.




**Gambar 7**. Pengujian ketika *Loadbalance*



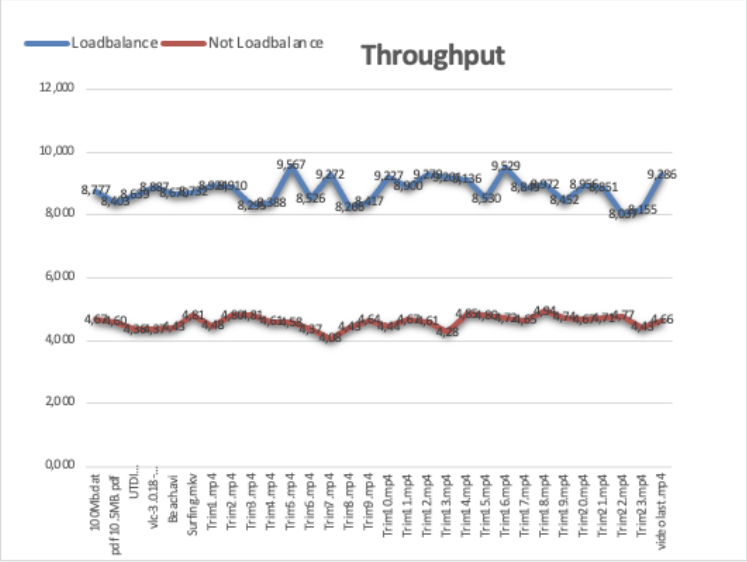

**Gambar 8.** Pengujian ketika Tanpa *Loadbalance*

Pada Gambar 7, bisa dilihat bahwa ketika dilakukan loadbalance pengiriman data terdistribusi secara merata 5Mbps melalui *WireGuard* pertama, dan 5Mbps ke *WireGuard* kedua, sehingga per detik nya bisa didapatkan 10Mbps. Sedangkan pada Gambar 8, ketika *WireGuard1* mati, maka otomatis akan dilewati satu jalur saja, ketika aktif kembali *WireGuard1*, maka akan *loadbalance* lagi secara merata.

### *3.3.* Pengujian *Quality of Service (QoS)*

Peneliti melakukan pengujian terhadap 30 dataset yang disiapkan untuk melihat hasil berdasarkan rumus yang sudah dituliskan pada Metode Penelitian. Setelah dilakukan pengujian, bisa disimpulkan bahwa Optimisasi Loadbalance lebih baik dibandingkan tanpa *loadbalance*.

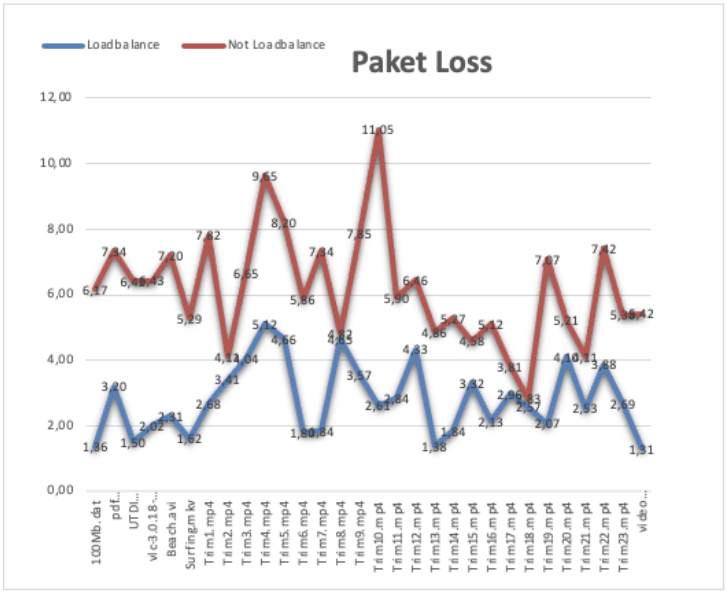
1. Pengujian *Throughput*




**Gambar 9**. Grafik perbedaan Pengujian Throughput

Pada Gambar 9, bisa dilihat bahwa dengan optimisasi loadbalance pada WireGuard menghasilkan efektifitas dalam pengiriman data 2x lipat dibandingkan Tanpa loadbalance.

2. Pengujian *Packet Loss*

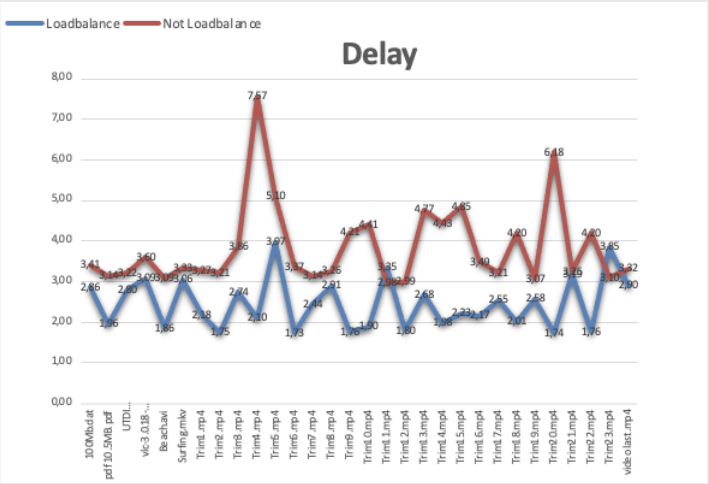



**Gambar 10**. Grafik perbedaan Pengujian *Paket Loss*

Pada Gambar 10, bisa disimpulkan bahwa ketika tanpa loadbalance, banyak paket yang hilang, sehingga dilakukan pengiriman data kembali, tapi jika menggunakan loadbalance. Paket yang loss lebih sedikit.

**PETIR: Jurnal Pengkajian dan Penerapan Teknik Informatika**
Vol. 17, No. 1, Maret 2024, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v17i1.2320

3. Pengujian Delay

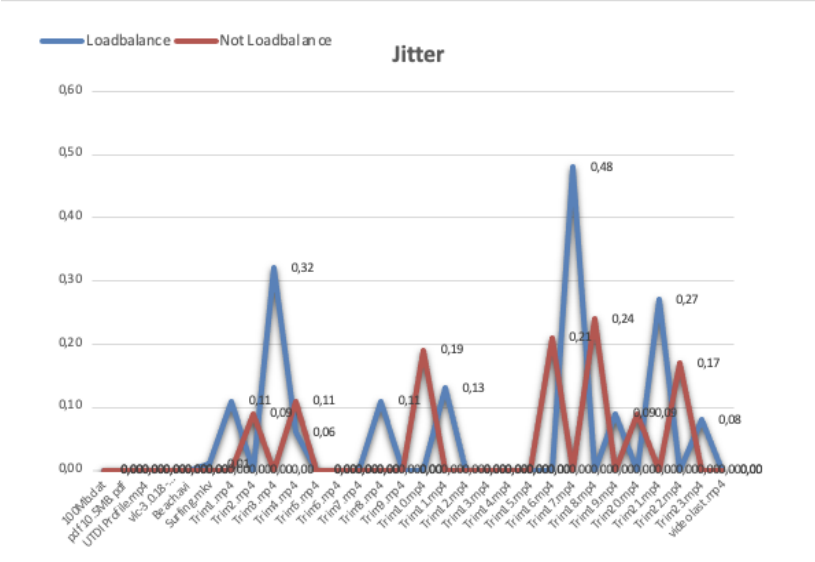



**Gambar 11.** Grafik perbedaan Pengujian Delay

Pada Gambar 11, delay *loadbalance* lebih rendah dibandingkan tanpa *loadbalance* karena pengiriman data nya terdistribusi kedua jalur, sehingga lebih efektif dan *delay* nya menjadi lebih rendah.

4. Pengujian *Jitter*

Pada Gambar 12, perbandingan *jitter* antara *loadbalance* dan tanpa *loadbalance* tidak terlihat signifikan, mungkin bisa dicoba dengan besaran paket yang lebih besar untuk melihat hasilnya.




**Gambar 12**. Grafik perbedaan Pengujian *Jitter*

Pada Gambar 12, perbandingan *jitter* antara *loadbalance* dan tanpa *loadbalance* tidak terlihat signifikan, mungkin bisa dicoba dengan besaran paket yang lebih besar untuk melihat hasilnya.

Jika dikumpulkan dan ditotal dari 30 dataset yang dilakukan pengujian, maka hasilnya ditunjukkan pada tabel 8 berikut.

Vol. 17, No. 1, Maret 2024, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v17i1.2320

**Tabel 8.** Indeks QoS berdasarkan TIPHON (1998)

| Pengujian | Loadbalance/No | Parameter | Hasil | Indeks |
|---|---|---|---|---|
| Data 1-30 | Loadbalance | Throughput | 88% | 4 |
| Data 1-30 | Loadbalance | Paket Loss | 2,81% | 4 |
| Data 1-30 | Loadbalance | Delay | 2,46ms | 4 |
| Data 1-30 | Loadbalance | Jitter | 0ms | 4 |

Setelah dirata-rata dari data yang diuji, nilai pada data yang *diloadbalance* memiliki hasil yang lebih baik dengan nilai indeks 4 untuk semua parameter *Throughput*, *Packet Loss*, *Delay*, dan *Jitter* dikarenakan kedua jalur *VPN* digunakan secara bersamaan sehingga pengiriman data dibagi rata jadi mengurangi nilai *delay*, *jitter, dan packet loss* serta meningkatkan *Throughput*, sehingga Total Rata - Rata pada Indeks diatas adalah **4**, yaitu **MEMUASKAN**

Adapun jika yang tanpa loadbalance berikut hasil nya :

**Tabel 9.** Indeks QoS berdasarkan TIPHON (1998)

| Pengujian | No Loadbalance | Parameter | Hasil | Indeks |
|---|---|---|---|---|
| Data 1-30 | No Loadbalance | Throughput | 46% | 2 |
| Data 1-30 | No Loadbalance | Paket Loss | 6,19% | 3 |
| Data 1-30 | No Loadbalance | Delay | 3,84ms | 4 |
| Data 1-30 | No Loadbalance | Jitter | 0ms | 4 |

Adapun tanpa *loadbalance* memiliki nilai cukup memuaskan dikarenakan pada *throughput* hanya 46% saja karena yang aktif hanya satu saja jalur yang aktif, lalu pada nilai *packet loss* lebih banyak paket yang hilang dikarenakan antrian paket yang penuh sehingga di *drop*, sehingga Total Rata - Rata adalah **3,25**, yaitu **CUKUP MEMUASKAN**.

## 4. KESIMPULAN DAN SARAN

Dari pengujian yang telah dilakukan ada beberapa hal yang bisa disimpulkan seperti :

1. Optimasi jaringan *Virtual Private Network (VPN)* menggunakan *WireGuard* dengan merupakan salah satu *VPN* yang terbaik saat ini untuk kebutuhan komunikasi data dari kantor pusat ke kantor cabang secara baik.
2. *WireGuard* mengalahkan tipe *VPN* sebelum nya seperti *OpenVPN* dan *IPSec* dalam hal kecepatan pengiriman data, dan *latency* dari kantor cabang ke kantor pusat.
3. *WireGuard* merupakan salah satu *VPN* yang aman ketika pengiriman data dilakukan dan di *sniffing packet* nya terlihat *encrypted traffic* dengan aplikasi *Wireshark*
4. Untuk menghindari *Single Point of Failure* dan ditambahkan jalur *VPN* baru, lebih baik dioptimalisasi dengan *Loadbalancing* karena pengiriman data di distribusikan secara merata di kedua jalur, dibandingkan hanya aktif 1 jalur saja.
5. Ketika hanya satu jalur saja yang aktif, beberapa parameter *Quality of Service* seperti *Throughput*, *Delay*, *Packetloss* mendapatkan hasil yang lebih buruk dibandingkan *Loadbalancing*, yaitu *Throughput* 46%, *Paket Loss* 6,19%, *Delay* 3,84ms.
6. Ketika dua jalur yang aktif, beberapa parameter *Quality* of *Service* seperti *Throughput*, *Delay*, *Packetloss*, *Delay*, mendapatkan hasil yang lebih baik dibandingkan Tanpa *Loadbalancing*, yaitu *Throughput* 88%, *Paket Loss* 2,81%, *Delay* 2,46ms

Vol. 17, No. 1, Maret 2024, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v17i1.2320

7. Dengan ada nya *Round Robin* pengiriman data dikirim secara merata melalui kedua jalur *VPN WireGuard* yang dibuatkan.

Selain itu untuk penelitian lebih lanjut ada beberapa saran yang bisa diimplementasikan kedepan nya, yaitu :

1. Pada penelitian ini menggunakan aplikasi network simulator yaitu Pnetlab, untuk mendapatkan hasil yang maksimal bisa menggunakan perangkat riil dan *hardware* fisik sehingga nilai pengujian mendapatkan hasil yang lebih riil.
2. Penelitian ini dilakukan di satu server yang sama, untuk mendapatkan nilai yang maksimal bisa dilakukan dengan riil berbeda lokasi tidak hanya secara local saja
3. Wireguard memang cepat dan aman, tapi masih ada celah ketika ada nya serangan seperti *Distributed Denial of Service (DDoS)* atau *PostQuantum* pada *kernel* [7]
4. Routing yang digunakan untuk pada lab ini yaitu *Static Routing*, jika ada beberapa jaringan lain nya, bisa menggunakan *Dynamic Routing Protocol* seperti *OSPF, BGP*
5. Pengujian ini menggunakan Internet dengan kapasitas 5Mbps tiap lokasi, kedepan nya bisa coba menggunakan kapasitas yang lebih besar seperti 1Gbps atau 10Gbps.

## DAFTAR PUSTAKA

[1] S. Sadi and T. Budiawan, "Kontrol Pendingin Ruangan (Fan) dengan Logika Fuzzy Menggunakan Atmega 8535, LM35 Dan PIR," TELKA, vol. 2, no. 2, pp. 94–105, 2016.

[2] A. Alhafiz, "Implementasi Metode Fuzzy Logic Pada Intensitas Lampu di Laboratorium Berbasis Arduino," Jurnal Sains Manajemen Informatika dan Komputer, vol. 19, no. 2, pp. 36–45, 2020, [Online]. Available: https://ojs.trigunadharma.ac.id/

[3] A. Budiman and Y. Ramdhani, "PENGONTROLAN ALAT ELEKTRONIK MENGGUNAKAN MODUL NODEMCU ESP8266 DENGAN APLIKASI BLYNK BERBASIS IOT," 2021.

[4] E. Sri Rahayu and dan Romi Achmad Mukthi Nurdin, "Perancangan Smart Home Untuk Pengendalian Peralatan Elektronik Dan Pemantauan Keamanan Rumah Berbasis Internet Of Things," 2019.

[5] J. Ambarita, R. P. Ardianto, A. Surya Wibowo, K. Kunci, and N. Esp, "RANCANG BANGUN PROTOTIPE SMARTHOME BERBASIS INTERNET OF THINGS (IoT) MENGGUNAKAN APLIKASI BLYNK DENGAN MODUL ESP 8266 DESIGN SMARTHOME PROTOTYPE BASED ON IOT USING BLYNK APPLICATION WITH THE ESP MODULE 8266," 2019.

[6] J. Minsyah Putra, "PROTOTYPE SMART CLASSROOM BERBASIS," 2020.

[7] A. Imran Lubis and M. Yetri, "Sistem Kendali Lampu Ruangan Menggunakan Metode Fuzzy Logic Dan Android Berbasis Mikrokontroler," 2022. [Online]. Available: https://ojs.trigunadharma.ac.id/index.php/jskom

[8] S. Anam et al., "RANCANG BANGUN SISTEM DETEKSI DAN PEMADAM KEBAKARAN PADA SMART HOME MENGGUNAKAN METODE FUZZY," JIP (Jurnal Informatika Polinema), vol. 6, pp. 9–16, 2020.

[9] R. Rizqi Wijayanti, R. Sabti Septarini, and S. Maulana Husain, "MODEL RUMAH PINTAR DENGAN MENGGUNAKAN LOGIKA FUZZY SEBAGAI PENGENDALI KEAMANAN DAN KESELAMATAN PENGHUNI RUMAH," 2020.

Vol. 17, No. 1, Maret 2024, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v17i1.2320

[10] A. Pranata, A. Stmik, and T. Dharma, "J-SISKO TECH Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD Implementasi Fuzzy Logic pada Sistem Pendingin Ruangan Otomatis berbasis Proggrammable Logic Controller (PLC)," vol. 1, no. 2, pp. 51–59, 2018.

[11] R. Aulia, R. Aulia Fauzan, and I. Lubis, "PENGENDALIAN SUHU RUANGAN MENGGUNAKAN MENGGUNAKAN FAN DAN DHT11 BERBASIS ARDUINO," CESS (Journal of Computer Engineering System and Science), vol. 6, pp. 30–38, 2021.

[12] D. Alit, A. Widiasa, I. P. Pangaribuan, and E. Kurniawan, "Purwarupa Smart Home dengan Multi Sensor dan Kontrol Buka Tutup Jendela Serta Tirai Otomatis Menggunakan Logika Fuzzy Prototype Smart Home with Multi Sensor and Control Open Close Window and Curtain Auto Using Fuzzy Logic."

[13] R. Madani, "SISTEM MONITORING DAN KONTROL IRIGASI TETES PADA CABAI BERBASIS INTERNET OF THINGS," 2020.

[14] M. Artiyasa et al., "APLIKASI SMART HOME NODE MCU IOT UNTUK BLYNK," 2020.

[15] L. Ade Putra and A. Rahman Hakim, "Sistem Kendali Lampu Cerdas Pada Smarthome Berbasis Android mengunakan Metode Fuzzy Logic Control Smart Lights Control System On Smarthome Based Android using Fuzzy Logic Control Method," CSRID Journal, vol. 10, pp. 33–43, 2018, doi: 10.22303/csrid.10.1.2018.33-43.

[16] R. Rizal and I. Karyana, "Sistem Kendali dan Monitoring pada Smart Home Berbasis Internet of Things (IoT)," vol. 1, no. 2, pp. 43–50, 2019.

[17] A. Hanani and M. Amin Hariyadi, "Smart Home Berbasis IoT Menggunakan Suara Pada Google Assistant," Jurnal Ilmiah Teknologi Informasi Asia, vol. 14, pp. 49–56, 2020.