Akses Perangkat Switch OLT ZTE menggunakan Bot Chat Telegram dengan Telnet menggunakan Algoritma Kontrol Akses Berbasis Peran (RBAC) untuk membantu konfigurasi

DISUSUN OLEH

DAVA AZAKI

2009020116

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

MEDAN

2024

Akses Perangkat Switch OLT ZTE menggunakan Bot Chat Telegram dengan Telnet menggunakan Algoritma Kontrol Akses Berbasis Peran (RBAC) untuk membantu konfigurasi

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom) dalam Program Studi Teknologi Informasi pada Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara

DAVA AZAKI
NPM. 2009020116

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMDIYAH SUMATERA UTARA
MEDAN
2024

LEMBAR PENGESAHAN

Judul Skripsi	:Akses Perangkat Switch OLT ZTE menggunakan Bot Chat Telegram dengan Telnet menggunakanAlgoritmaKontrolAkses Berbasis Peran (RBAC) untuk membantu konfigurasi
Nama Mahasiswa	: DAVA AZAKI
NPM	: 2009020116
Program Studi	: TEKNOLOGI INFORMASI

Menyetujui
Komisi Pembimbing

(Farid Akbar Siregar, S.Kom., M.Kom)
NIDN. 0104049401

Ketua Program Studi

(Fatma Sari Hitagalung, M.Kom)
NIDN. 0117019301

Dekan

(Dr. Al-Khowarizmi, S.Kom., M.Kom.)
NIDN. 0127099201

PERNYATAAN ORISINALITAS

Akses Perangkat Switch OLT ZTE menggunakan Bot Chat Telegram dengan Telnet menggunakan Algoritma Kontrol Akses Berbasis Peran (RBAC) untuk membantu konfigurasi

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, Juli 2024

...uat pernyataan

...KI

NPM. 2009020116

PERNYATAAN PERSETUJUAN PUBLIKASI

KARYA ILMIAH UNTUK KEPENTINGAN

AKADEMIS

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama	: Dava Azaki
NPM	: 2009020116
Program Studi	: Teknologi Informasi
Karya Ilmiah	: Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bedas Royalti Non-Eksekutif *(Non-Exclusive Royalty free Right)* atas penelitian skripsi saya yang berjudul:

Akses Perangkat Switch OLT ZTE menggunakan Bot Chat Telegram dengan Telnet menggunakan Algoritma Kontrol Akses Berbasis Peran (RBAC) untuk membantu konfigurasi

Beserta perangkat yang ada (jika diperlukan). Dengan hak bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasi Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dana tau sebagai pemilik hak cipta.
Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, Juli 2024
Yang membuat pernyataan

Dava Azaki
NPM. 2009020116

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap	: Dava Azaki
Tempat dan Tanggal Lahir	: Tanjungbalai, 19 Juli 2002
Alamat Rumah	: Jl.Pasir Raya, LK V
Telepon/Faks/HP	: 085157127721
E-mail	: direktur@metan.id
Instansi Tempat Kerja	: PT.MEDIA BALAI NUSA
Alamat Kantor	: Jl.Letjend Suprapto, Beting Kuala Kapias

DATA PENDIDIKAN

SD : SDN 132406 TAMAT: 2013/2014

SMP : SMPN 1 Tanjungbalai TAMAT: 2016/2017

SMA : SMA SWASTA SISINGAMANGARAJA TAMAT: 2019/2020

KATA PENGANTAR

Pendahuluan

Penulis tentunya berterima kasih kepada berbagai pihak dalam dukungan serta doa dalam penyelesaian skripsi. Penulis juga mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Agussani, M.AP., Rektor Universitas Muhammadiyah Sumatera Utara (UMSU)
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom. Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
3. Ibu Fatma Sari Hutagalung, M.Kom. Ketua Program Studi Teknologi Informasi
4. Bapak Mhd. Bari, S.Si, M.Kom. Sekretaris Program Studi Teknologi Informasi
5. Pembimbing Farid Akbar Siregar, S.Kom., M.Kom.
6. Keluarga yang telah memberikan dukungan dan doa selama proses penelitian
7. Direktur PT.Media Balai Nusa
8. Teman-teman dan sahabat, yang telah membantu selama penelitian yang banyak membantu kepada penulis
9. Semua pihak yang terlibat langsung ataupun tidak langsung yang tidak dapat penulis ucapkan satu-persatu yang telah membantu penyelesaian skripsi ini.

Akses Perangkat Switch OLT ZTE menggunakan Bot Chat Telegram dengan Telnet menggunakanAlgoritmaKontrolAkses Berbasis Peran (RBAC) untuk membantu konfigurasi

ABSTRAK

Dalam era digital ini, akses internet telah menjadi integral dalam kehidupan manusia. Hampir semua aktivitas manusia, mulai dari komunikasi, edukasi, hingga pekerjaan, dilakukan dengan menggunakan internet. Penelitian ini berfokus pada meningkatkan efisiensi dan kemudahan dalam mengakses dan mengelola perangkat Switch OLT (Optical Line Terminal) ZTE melalui penggunaan Bot Chat Telegram. Dalam konteks ini, perangkat Switch OLT ZTE merupakan bagian vital dari infrastruktur jaringan, khususnya dalam jaringan akses optik, yang membutuhkan pengaturan dan pemantauan yang terus-menerus.Otomatisasi jaringan menjadi semakin penting saat ini karena jaringan menjadi lebih kompleks dan penting bagi keberhasilan bisnis dan perusahaan. Mengotomatiskan tugas jaringan sehari-hari dapat membantu administrator dan insinyur jaringan mengurangi kesalahan, meningkatkan produktivitas, dan meningkatkan skala infrastruktur mereka dengan mudah. penelitian ini menggunakan metode Role Based Access Control (RBAC) merupakan mekanisme pengelolaan sejumlah besar hak akses pada basis data berukuran besar yang fleksibel. Dibandingkan model kontrol akses tradisional yaitu Mandatory Access Control (MAC) dan Discretionary Access Control (DAC) penelitian ini, dilakukan implementasi integrasi antara Telegram Bot sebagai antarmuka pengguna dan Telnet sebagai protokol komunikasi untuk mengakses perangkat Switch OLT ZTE. Telegram Bot dipilih karena kemudahan dalam penggunaan dan fleksibilitas dalam menyediakan antarmuka yang interaktif untuk pengguna. BotFather adalah bot resmi dari Telegram yang memungkinkan pengguna untuk membuat, mengelola, dan mengkonfigurasi bot Telegram. Berikut adalah beberapa hal yang bisa dilakukan dengan BotFather.Otomatisasi pada bot telegram yang saya buat ini dapat mengerjakan pekerjaan berulang yang biasanya terjadi pada teknisi di PT.Media Balai Nusa. Ini mengurangi beban kerja manual yang terlalu banyak dan terus menerus, dan dapat membuat pekerjaan Noc untuk lebih fokus kepada pekerjaan yang lebih kompleks. Bot ini juga tidak akan mengeksekusi script yang salah ketika inputan yang diberi user ambigu dan akan tidak merespon jika terjadi dual input yang diberikan pengguna, jadi tidak akan merusak sistem atau konfigurasi pada OLT yang sedang berjalan. Perlu dilakukan pengembangan untuk menggunakan beberapa platform lain seperti Whatsapp mengingat saat ini aplikasi tersebut sudah menjadi kewajiban di Smartphone semua orang.

Kata Kunci: Internet; *Switch OLT ZTE*; *Telnet*; *Otomatisasi Jaringan*; Role Based Access Control (RBAC); Bot Chat Telegram

Access ZTE OLT Switch Devices using Telegram Chat Bot with Telnet using Role-Based Access Control (RBAC) algorithm to help configure

ABSTRACT

In this digital age, internet access has become an integral part of human life. Almost all human activities, from communication, education, to work, are done using the Internet. The research focuses on improving the efficiency and ease of accessing and managing ZTE's OLT (Optical Line Terminal) Switch devices through the use of Telegram Chat Bot. In this context, the ZTE OLT Switch device is a vital part of the network infrastructure, especially in the optical access network, which requires continuous adjustment and monitoring.Network automation is becoming increasingly important nowadays as networks are becoming more complex and essential for business and corporate success. Automating day-to-day network tasks can help network administrators and engineers reduce errors, increase productivity, and scale their infrastructure easily. The study uses the Role Based Access Control (RBAC) method, which is a mechanism for managing large amounts of access rights on flexible large-scale databases. Comparing the traditional access control models Mandatory Access Control (MAC) and Discretionary Access control (DAC) of the study, implementation of integration between Telegram Bot as a user interface and Telnet as a communication protocol for accessing the ZTE OLT Switch device was carried out. Telegram Bot was chosen because of its ease of use and flexibility in providing an interactive interface for users. BotFather is the official bot of Telegram that allows users to create, manage, and configure Telegram bots. Here are a few things that can be done with botFather.The automation on the telegram boat that I've made can do the repetitive work that normally happens to the technicians at the Nusa Media Hall. It reduces too much and continuous manual workload, and can make Noc work more focused on more complex work. This bot will also not execute the wrong script when the input given to the user is ambiguous and will not respond if there is a dual input given by the user, so it will not damage the system or configuration on the running OLT. It needs to be developed to use some other platforms like Whatsapp given that these apps are now a must on everyone's smartphones.

Keywords: Internet; Switch OLT ZTE; Telnet; Network Automation*;* Role Based Access Control (RBAC); Telegram Chat Bot

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1. Latar Belakang Masalah

Dalam era digital ini, akses internet telah menjadi integral dalam kehidupan manusia. Hampir semua aktivitas manusia, mulai dari komunikasi, edukasi, hingga pekerjaan, dilakukan dengan menggunakan internet. Penelitian ini berfokus pada meningkatkan efisiensi dan kemudahan dalam mengakses dan mengelola perangkat Switch OLT (Optical Line Terminal) ZTE melalui penggunaan Bot Chat Telegram. Dalam konteks ini, perangkat Switch OLT ZTE merupakan bagian vital dari infrastruktur jaringan, khususnya dalam jaringan akses optik, yang membutuhkan pengaturan dan pemantauan yang terus-menerus.

Perkembangan teknologi yang semakin canggih mengikuti kebutuhan masyarakat, salah satunya dalam dunia telekomu-nikasi yaitu FTTH (Fiber to the Home) yang merupakan format penghantaran sinyal optik dari pusat penyedia (provider) ke rumah-rumah pelanggan dengan menggunakan fiber optik sebagai media penghantar. Penghantaran dengan menggunakan teknologi FTTH ini dapat menghemat biaya dan mampu mengurangkan biaya operasi serta memberikan pelayanan yang lebih baik kepada pelanggan. Penelitian dilakukan dengan metode wawancara dan pengamatan dengan objek jaringan dan kompo-nen FTTH milik Telkom Semarang serta melalui metode kepustakaan dengan pencarian beberapa literatur. Konfigurasi FTTH dimulai dengan ONT ke Roset dihubungkan menggunakan kabel Patchcord kemudian dari Rosetmenuju ke OTP menggunakan kabel indoor lalu dari OTP menuju ke ODP yang berada di luar rumah dihubungkan menggunakan kabel dropcore. ODP itu sendiri

merupakan hasil pembagian kabel distribusi/kabel udara yang menuju ke ODC. Lalu dari ODC menuju STO/FTM/ODF menggunakan kabel feeder. Jaringan FTTH sangat efektifdipakai, karena pelanggan dapat menggunakan layanan triple play (telepon, internet dan IPTV) hanya dengan 1ONT/ modem saja disisi pelanggan.(Muliandhi et al., 2020)

Prisip kerja dari GPON yaitu ketika data atau sinyal dikirimkan dari OLT, maka ada bagian yang bernama splitter yang berfungsi untuk memungkinkan serat optik tunggal dapat mengirim ke berbagai ONT. Untuk ONT sendiri akan memberikan data-data dan sinyal yang diinginkan oleh user.

Konfigurasi GPON terdiri dari 3 bagian utama yaitu:

1) OLT (*Optical Line Terminal)* adalah perangkat utama terpasang di sisi sentral

2) ODN (*Optical Data Network*) adalah perangkat *fiber optic* meliputi ODF, ODC, ODP, Splitter.

3) ONT (*Optical Network Terminal*) adalah perangkat aktif disisi pelanggan.

GPON mampu memberikan layanan *Triple Play* yaitu *voice*, data dan video [8].

1) *Voice:* Bila kita memberikan layanan *voice* via jaringan FTTH GPON maka layanan *voice* dapat diberikan melalui

a. POTS port pada ONT dengan antar muka FXS (RJ11)

b. Dengan POTS ini maka telepon yang digunakan berupa telepon analog (telepon biasa)

c. SIP /H.248 *Phone* (RJ45)

d. Dengan *protocol* SIP/H.248 maka terminal telepon yang digunakan berupa IP *phone*. *Interface* yang digunakan antara IP *phone* dan ONT berupa RJ45. (*2430-8541-2-PB*, n.d.)

Otomatisasi jaringan menjadi semakin penting saat ini karena jaringan menjadi lebih kompleks dan penting bagi keberhasilan bisnis dan perusahaan. Mengotomatiskan tugas jaringan sehari-hari dapat membantu administrator dan insinyur jaringan mengurangi kesalahan, meningkatkan produktivitas, dan meningkatkan skala infrastruktur mereka dengan mudah. Python adalah bahasa pemrograman penting yang telah mendapatkan popularitas signifikan di industri jaringan karena kesederhanaan, keserbagunaan, dan perpustakaan modulnya yang luas. Pada kali ini peneliti mengeksplorasi Python untuk otomatisasi jaringan dan dampaknya terhadap manajemen, konfigurasi, dan pemantauaan link dan jaringan. ini ditujukan bagi administrator jaringan, insinyur, dan pengembang yang ingin mengeksplorasi manfaat penggunaan Python untuk otomatisasi jaringan.
(Dang, n.d.).

Otomatisasi jaringan menjadi semakin penting saat ini karena jaringan menjadi lebih kompleks dan penting bagi keberhasilan bisnis dan perusahaan. Mengotomatiskan tugas jaringan sehari-hari dapat membantu administrator dan insinyur jaringan mengurangi kesalahan, meningkatkan produktivitas, dan meningkatkan skala infrastruktur mereka dengan mudah. Python adalah bahasa pemrograman penting yang telah mendapatkan popularitas signifikan di industri jaringan karena kesederhanaan, keserbagunaan, dan perpustakaan modulnya yang luas.

Peneliti memilih menggunakan Python dengan beberapa alasan. Python adalah bahasa pemrograman tingkat tinggi yang mudah dipelajari, menjadi bagian integral dari jaringan skala besar, dan memecahkan masalah untuk mempersingkat

operasi pada jaringan (Chou, 2018). Python menjadi andalan bagi network administrator untuk memonitor, *konfigurasi* dan administrasi jaringan. Network admnistrator dapat melakukan monitor secara *real-time* apa yang sedang terjadi di jaringan menggunakan Python. Selain itu, netadmin tidak perlu mengetikkan perintah dengan CLI untuk mengkonfigurasi jaringan yang berulang karena dengan pyhon bisa membuat antarmuka secara *graphical* (Chou, 2020). Python adalah bahasa pemrograman yang memiliki fitur lengkap dengan library yang terdokumentasi dengan baik, sehingga akan memudahkan network programming untuk bereksperimen membuat program. Seorang network programming akan dengan mudah mengambil dan meminta data dari sebua web, serta dengan mudah melakukan ekstraksi data menjadi format yang umum melalui web hanya menggunakan Python.

Jaringan komputer saat ini semakin kompleks dan dinamis. Jaringan komputer yang kompleks akan membutuhkan banyak perangkat jaringan termasuk *router* dengan berbagai jenis, tipe dan merk. Jaringan kompleks saat ini biasanya melibatkan integrasi dan interkoneksi banyak perangkat *router*. Operator jaringan bertanggung jawab untuk mengonfigurasi jaringan dan merespons peristiwa yang terjadi di jaringan. Operator jaringan harus menerapkan tugas kompleks dengan serangkaian perintah konfigurasi dalam lingkungan *command line interface* (CLI). Konfigurasi pada jaringan akan semakin sulit implementasinya karena sebuah kebijakan pada level pimpinan akan berakibat penerapan konfigurasi yang berbeda pada beberapa *router* dengan jenis, tipe dan merk yang berbeda. Kondisi jaringan dinamis berarti kondisi jaringan yang terus berubah. Penambahan perangkat, peningkatan jumlah pemakai adalah beberapa penyebab jaringan

bersifat dinamis. Operator jaringan harus menyesuaikan konfigurasi jaringan sebagai tanggapan terhadap perubahan kondisi jaringan (Nugroho & Pujiarto, 2022).

Saat ini untuk dapat mengoperasikan perangkat switch OLT *(Optical Line Terminal)* harus menggunakan CLI *(Command Line Interface)*, akan menjadi sedikit menyusahkan jika yang harus dikonfigurasi adalah hal-hal yang berulang. Oleh sebab itu peneliti tertarik untuk melakukan penelitian tentang **Akses Perangkat Switch OLT ZTE menggunakan Bot Chat Telegram dengan Telnet menggunakan Algoritma Kontrol Akses Berbasis Peran (RBAC) untuk membantu konfigurasi**

1.2. Rumusan Masalah

Berdasarkan penjelasan pada latar belakang masalah diatas, maka rumusan masalah yang dapat dikaji pada penelitian ini adalah peneliti menginginkan untuk konfigurasi perangkat yang pada kali ini adalah switch multi layer dimana untuk pengoperasian perangkat ini harus melalui *Command Line Interface* menjadi menggunakan *shortcut* atau kata singkat yang mudah untuk diingat semua pengguna untuk pengoperasian nya.

1.3. Batasan Masalah

Berdasarkan Batasan Masalah dalam Penelitian ini adalah sebagai berikut :

1. Menggunakan bahasa pemrograman Python
2. Menggunakan *platform* Telegram
3. Menggunakan sistem operasi ubuntu desktop
4. Menggunakan *telnet* untuk berkomunikasi

5. Fokus pada perusahaan PT.MEDIA BALAI NUSA

1.4. Tujuan Penelitian

Berdasarkan Rumusan Masalah diatas, adapun tujuan dari penelitian ini adalah sebagai berikut :

1. Mengetahui Bagaimana Implementasi Bot Telegram dengan Python ?
2. Mengetahui Bagaimana Implementasi Bot Telegram dapat memudahkan pekerjaan operator dan *technical support* ?
3. Mengetahui Bagaimana Penerapan Bot Telegram dapat membuat *shortcut* dan *push config* pada perangkat *switch* ?

1.5. Manfaat Penelitian

Hasil studi ini dinantikan dapat memperkaya ilmu pengetahuan dan bisa digunakan untuk rekanan *Internet Servise Provider* untuk mengelola perangkat switch mereka jika menggunakan perangkat yang sama

1. Manfaat Praktis

a. Bagi Peneliti

Hasil penelitian dapat membantu memperluas pengetahuan peneliti mengenal tentang Python, Penggunaan sistem operasi Ubuntu Server yang digunakan sebagai penjalan script python tersebut untuk terhubung ke perangkat switch maupun router, dapat dikembangkan lagi kedepan untuk skala industri.

b. Bagi Universitas

Hasil penelitian ini dapat menjadi referensi tambahan diperpustakaan dan perbandingan dalam penelitian-penelitian berikutnya.

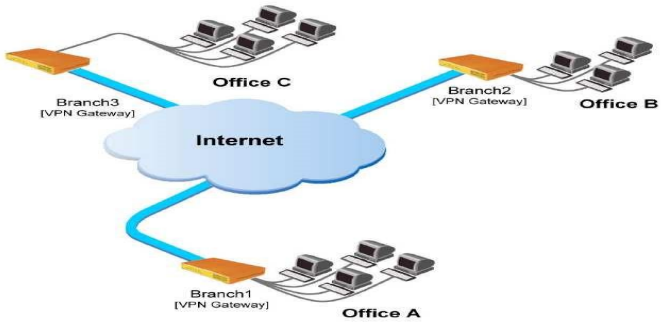
c. Bagi Pengguna Internet

Hasil penelitian ini dapat meningkatkan pemahaman tentang *Autonomous Configuration,* dan dapat dikembangkan menjadi dapat mendukung semua perangkat *Router,Switch,Server.*

BAB II
LANDASAN TEORI

2.1 *Internet Servise Provider*

Jaringan internet adalah sesuatu yang dibutuhkan untuk mengakses situs, website, atau media sosial. Biasanya, jaringan internet ini berupa sambungan antar jaringan yang berada di seluruh bagian dunia, yang terdiri dari Personal Area Network (PAN), Local Area Network (LAN), Metropolitan Area Network (MAN), Campus Area Network (CAN), hingga Wide Area Network (WAN). Kata internet juga dapat berupa singkatan dari kata "*Interconnected Network*" yang membuat kamu sangat mungkin untuk berkomunikasi dan berbagi informasi dengan siapa saja, dari siapa saja, dan di mana saja. Selain itu juga, ada yang mendefinisikan jika internet adalah *International Network*.



Gambar 2.1 Jaringan WAN

Interconnection-networking (Internet) adalah sebuah sistem global jaringan komputer yang saling menghubungkan antara satu dengan yang lain di seluruh penjuru dunia dengan menggunakan standart *Internet Protocol Suite*. Sejarah internet di Indonesia pertama kali dikenal pada tahun 1990an. Adanya teknologi informasi seperti internet telah membuka mata dunia akan sebuah dunia, interaksi

dan market place baru serta sebuah jaringan bisnis dunia yang tanpa batas. Dunia didalam internet disebut juga dengan dunia maya (*cyberspace*). Hadirnya internet sebagai sebuah infrastruktur dan jaringan telah menunjang efektifitas dan efisiensi operasional sebuah perusahaan, terutama peranannya sebagai sarana publikasi, komunikasi, serta sarana untuk mendapatkan berbagai informasi yang dibutuhkan. Informasi dalam internet umumnya disebarkan melalui suatu halaman yang disebut dengan istilah situs jaringan (*website)* yang dibuat dengan format bahasa pemrograman HTML *(Hypertext Markup Languange*). Internet sendiri merupakan ruang komunikasi baru yang salah satu fungsinya adalah dapat menjadi media massa. Perlu diketahui bahwa Internet tidak memiliki sentralisasi pemerintahan (Gani, n.d.).

2.2 Python

Bahasa pemrograman Python, yang sangat populer saat ini, pertama kali ditemukan oleh Guido van Rossum di Stichting Mathematisch Centrum (CWI), Amsterdam pada tahun 1991(Awangga et al., 2019). Pengembangan Python terinspirasi oleh bahasa pemrograman ABC yang berkembang pada saat itu. Salah satu perbedaan utama Python dengan bahasa pemrograman lainnya adalah pengembangannya melibatkan jutaan programmer, peneliti, dan pengguna dari berbagai latar belakang, tidak hanya dari kalangan IT, karena Python bersifat open source.Python adalah bahasa pemrograman yang menggunakan interpreter untuk menjalankan kode programnya. Interpreter tersebut dapat menerjemahkan kode secara langsung, dan Python dapat dijalankan di berbagai platform seperti Windows, Linux, dan lain-lain. Pythonmengadopsi paradigma pemrograman dari beberapa bahasa lain,

termasuk paradigma pemrograman prosedural seperti bahasa C, pemrograman berorientasi objek seperti Java, dan bahasa fungsional seperti Lisp. Kombinasi paradigma ini memudahkan para programmer dalam mengembangkan berbagai proyek menggunakan Python. Ada beberapa alasan mengapa Python menjadi pilihan utama, yaitu:1.Python dapat berjalan di berbagai platform seperti Windows, Linux, macOS, Android, Raspberry Pi, dan lain-lain.Python memiliki sintaks yang sederhana dan mirip dengan bahasa Inggris.Sintaks Python memungkinkan penulisan kode yang lebih ringkas dibandingkan dengan bahasa pemrograman lain.Python menggunakan interpreter, sehingga program dapat dieksekusi dengan cepat setelah selesai dibuat.Python mendukung paradigma pemrograman prosedural, berorientasi objek, dan fungsional.Berdasarkan penjelasan di atas, mari kita mulai menggunakan Python (Rahman et al., n.d.).

2.3 Telnet

Jaringan Telekomunikasi (Telnet) adalah protokol jaringan yang digunakan untuk mengakses dan mengontrol gadget yang jauh melalui protokol jaringan dengan Telnet, perangkat dapat berasosiasi dengan gadget, seperti server, switch, atau PC yang jauh dan berkomunikasi dengannya seolah-olah mereka sebelum gadget. Telnet dapat diserang dengan memanfaatkan teknologi telekomunikasi dan sistem informasi jaringan komputer sebagai media transmisi dengan tujuan menipu orang lain untuk mendapatkan milik publik dan swasta (Dharma & Rino, 2023).

2.4 Telegram

Telegram adalah aplikasi pesan instan yang berbasis cloud dan alat enkripsi, memungkinkan pengguna untuk berbagi informasi dan berkomunikasi dengan orang lain secara efisien dan cepat. Telegram menyediakan enkripsi end-to-end, penghancuran pesan secara otomatis, dan infrastruktur multi-pusat data, membuatnya sangat populer di kalangan pengguna. Chatbot Telegram adalah sebuah robot chat virtual yang dapat menjawab pertanyaan pelanggan dan menjalankan perintah yang diimplementasikan di Telegram. Chatbot Telegram menggunakan teknologi Natural Language Processing (NLP) untuk memahami bahasa alami dan berinteraksi dengan pengguna. Dengan demikian, chatbot Telegram dapat membantu meningkatkan pelayanan publik dengan memberikan informasi dan menjawab pertanyaan secara real-time (Furqan et al., 2023).

2.5 Algoritma Kontrol Akses Berbasis Peran (RBAC)

Pengontrolan terhadap hak akses berguna untuk membatasi pengguna yang akan mengakses informasi dan menjaga keamanan atas informasi yang dianggap rahasia, sehingga informasi tersebut tidak dapat diakses oleh pengguna yang tidak diinginkan. Informasi hanya dapat diberikan kepada pengguna yang telah diberikan otoritas akses terhadap sistem tersebut. Oleh sebab itu sangat diperlukan kontrol akses guna membatasi hak-hak apa saja dapat diakses oleh pengguna. Sehingga keamanan atas informasi dapat terjaga dari pihak-pihak yang tidak diinginkan. Kontrol akses akan mendukung kerahasian dan integritas keamanan dalam sebuah sistem sehingga pengguna akan dibatasi hak apa yang boleh

dilakukan dalam mengakses sistem tersebut. Kontrol akses merupakan cara mencegah ancaman keamanan internal. *Role Based Access Control* (RBAC) merupakan mekanisme pengelolaan sejumlah besar hak akses pada basis data berukuran besar yang fleksibel. Dibandingkan model kontrol akses tradisional yaitu *Mandatory Access Control* (MAC) dan *Discretionary Access Control* (DAC) (Habib, 2011) (Avila Putri et al., n.d.).

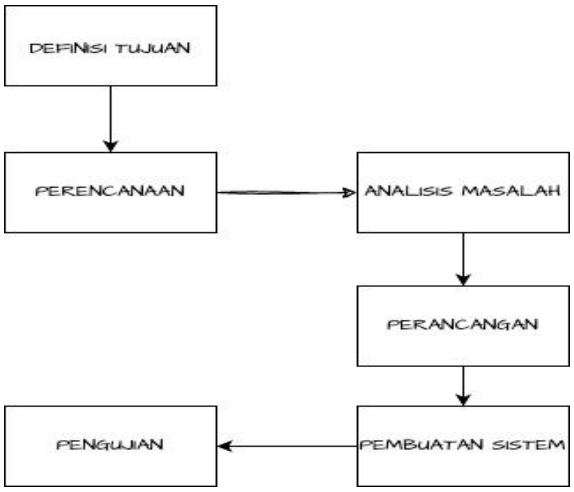
BAB III
METODOLOGI PENELITIAN

3.1. Metode Penelitian

Metode pengumpulan data yang penulis gunakan didalam penelitian ini adalah observasi dan studi pustaka. Penulis melakukan observasi metode observasi dengan cara terjun langsung kelapangan dan menemukan permasalahan atau kesulitan dalam hal terkait judul penulis. Metode observasi adalah suatu teknik pengumpulan data yang dilakukan melalui sesuai dengan pegamatan, dengan di sertai pecatatan-pencatatan tentang keadaan objek terkait dengan sasarannya.

3.2. Prosedur Penelitian

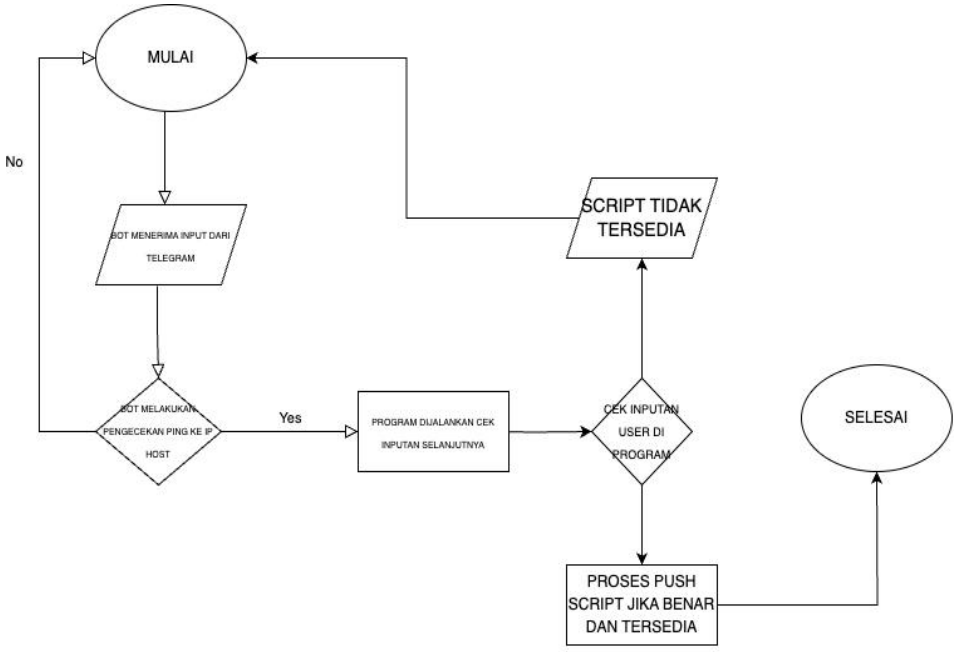
Penelitian ini memiliki beberapa tahap, yaitu defenisi tujuan, perencanaan, analisis masalah, perancangan, pembuatan sistem, pengujian



Gambar 3.1 Prosedur Penelitian

3.3. *Flowchart* implementasi algoritma RBAC pada python

Flowchart implementasi algoritma RBAC pada Chat Bot Telegram alur ini menggambarkan langkah-langkah yang diperlukan untuk mengetahui cara kerja mengimplementasikan algoritma RBAC.



Gambar 3.2 *Flowchar*t Implementasi algoritma RBAC

Adapun sistematika *flowchart* diatas yaitu, diawali dengan menjalankan program python pada ubuntu server lalu menginisialisasikan user yang dikhususkan untuk login kedalam *switch* setelah bot dijalankan dan berhasil masuk kedalam perangkat bot akan menerima imputan selanjutnya dari user dan akan memproses dari script yang sudah kita tetapkan untuk dikirimkan kedalam perangkat, setelah berhasil memasukkan script kedalam perangkat bot akan memberikan keluaran dari perangkat kembali ke telegram menyatakan bot sudah selesai bekerja dan sukses memasukkan configurasi tanpa harus menggunakan *cli*

3.4 Jadwal Penelitian

Setiap rancangan pada penelitian pastinya perlu dilengkapi dengan jadwal yang sudah dilakukan. Berikut ini adalah rincian penilaian

Tabel 3.1 Jadwal Penelitian

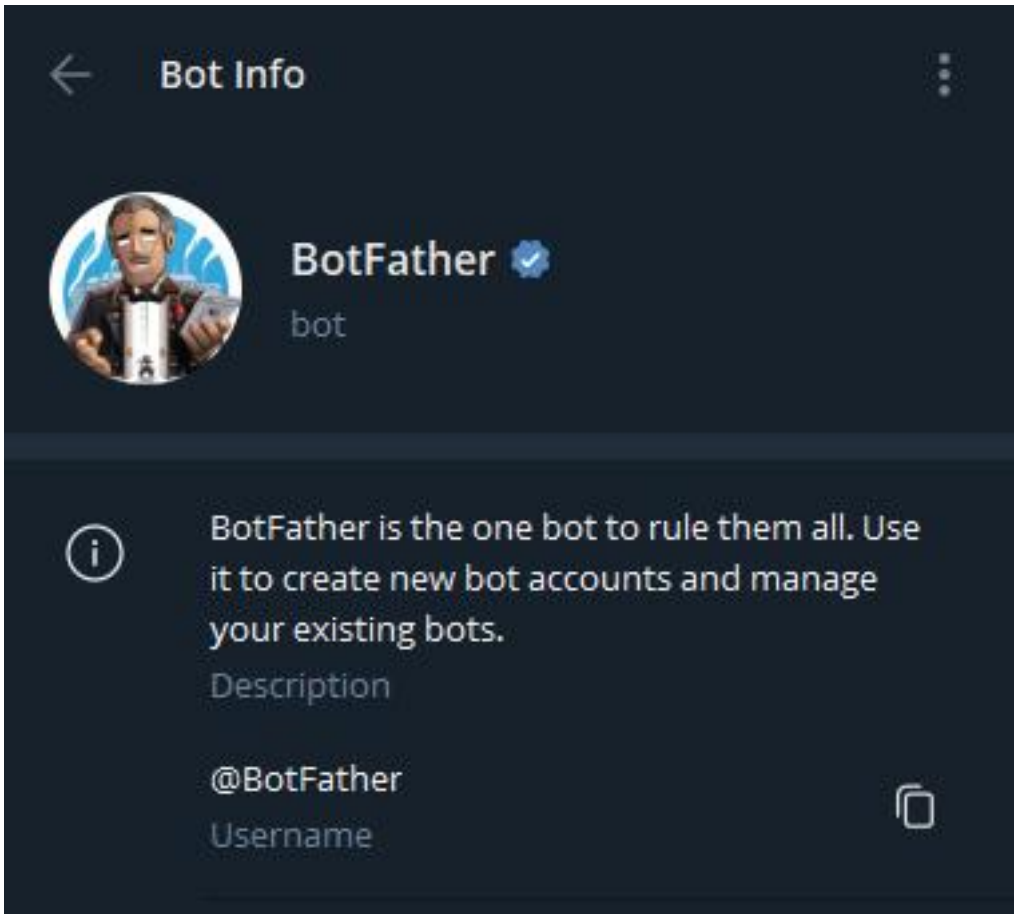
Kegiatan Penelitian	Bulan					
	Dec	Apr	Mei	Juni	Juli	Ags
1. Persiapan Penelitian						
a. Pengajuan Judul	■					
b. Pengajuan SK Pembimbing	■					
c. Observasi		■				
d. Penyesunan Proposal			■			
e. Seminar Proposal			■			
2. Implementasi & Pengumpulan Data						
a. Pembuatan Sistem			■			
b. Pengumpulan Data				■		
3. Pemrosesan Data Dan Pelaporan						
a. Validasi dan Hasil					■	
b. Penyusunan Laporan Skirpsi						■

BAB IV

HASIL DAN PEMBAHASAN

4.1. Implementasi Telnet dan Telegram Bot

Pada penelitian ini, dilakukan implementasi integrasi antara Telegram Bot sebagai antarmuka pengguna dan Telnet sebagai protokol komunikasi untuk mengakses perangkat Switch OLT ZTE. Telegram Bot dipilih karena kemudahan dalam penggunaan dan fleksibilitas dalam menyediakan antarmuka yang interaktif untuk pengguna. BotFather adalah bot resmi dari Telegram yang memungkinkan pengguna untuk membuat, mengelola, dan mengkonfigurasi bot Telegram. Berikut adalah beberapa hal yang bisa dilakukan dengan BotFather.



Gambar 4.1 Bot Father

```
[19072002@ro.corporate] > system/telnet 10.1.2.2
Connecting to 10.1.2.2
Connected to 10.1.2.2
WELCOME NOC ASTRONET.ID
--------by--------
PT.MEDIA BALAI NUSA

Last login time is 07.18.2024-18:30:59-utc, 0 authenticati
ince that time.
Username:mtnnet
Password:
%Error 20209: No username or bad password
Username:astro
Password:
OLT-1-METAN.ID#
```

Gambar 4.2 Telnet

4.2. Algoritma Kontrol Akses Berbasis Peran (RBAC)

Algoritma Kontrol Akses Berbasis Peran (RBAC) diimplementasikan untuk mengatur akses pengguna terhadap perintah-perintah konfigurasi pada perangkat Switch OLT ZTE. RBAC memungkinkan untuk menetapkan peran-peran yang sesuai dengan tugas atau tanggung jawab pengguna, sehingga membatasi akses hanya pada perintah-perintah yang relevan dan dibutuhkan.

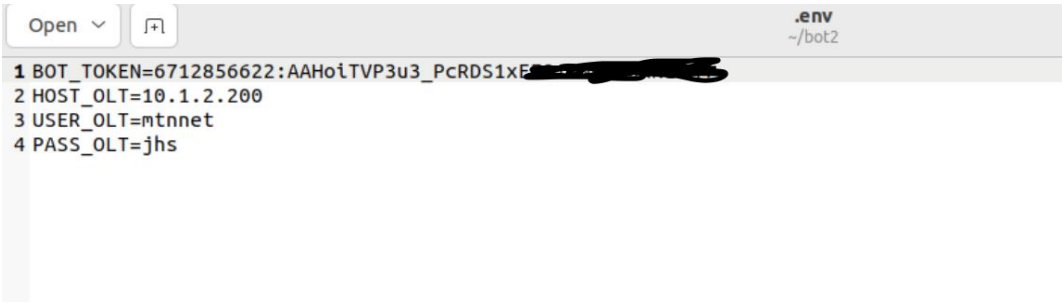
RBAC pada Switch OLT ZTE menjadikan manajemen akses lebih terstruktur dan efisien. Pengguna diberi peran berdasarkan fungsi atau tanggung jawab mereka dalam jaringan, seperti:

1. Administrator Jaringan: Memiliki akses penuh untuk melakukan konfigurasi, monitoring, dan manajemen umum perangkat serta jaringan.

2. Teknisi Jaringan: Memiliki akses untuk melakukan konfigurasi perangkat dengan batasan pada perintah-perintah teknis yang relevan, seperti pengaturan port dan konfigurasi VLAN.

3. Operator Jaringan: Memiliki akses terbatas untuk melakukan monitoring dan operasi rutin, seperti melihat status koneksi atau memulai ulang perangkat.

Dengan RBAC, perintah-perintah yang bisa diakses oleh setiap peran telah ditentukan dan disesuaikan dengan kebutuhan spesifik perangkat Switch OLT ZTE dan tugas yang diemban oleh pengguna. Hal ini tidak hanya meningkatkan keamanan dengan mengurangi risiko akses tidak sah, tetapi juga mempermudah administrasi jaringan dengan membatasi akses hanya pada fungsionalitas yang sesuai dengan peran pengguna.



Gambar 4.2 RBAC user

4.3. Pengujian Sistem

Sistem yang dikembangkan telah diuji secara menyeluruh untuk menguji fungsionalitas dan kinerjanya. Berbagai skenario pengujian telah dilakukan untuk mengevaluasi respons sistem terhadap perintah-perintah yang diberikan oleh pengguna melalui Telegram Bot. Pengujian ini juga bertujuan untuk memastikan keamanan akses dan kestabilan koneksi dengan menggunakan Telnet ke perangkat Switch OLT ZTE.

Pengujian dilakukan untuk memvalidasi beberapa aspek kritis, termasuk:

1. Respons Sistem: Evaluasi terhadap respons sistem terhadap perintah-perintah yang diterima dari Telegram Bot, dengan

mempertimbangkan waktu respons dan keakuratan eksekusi perintah.

2. Keamanan Akses: Pengujian keamanan dilakukan untuk memastikan bahwa hanya pengguna yang memiliki otorisasi yang tepat yang dapat mengakses dan menggunakan perintah-perintah konfigurasi yang sensitif pada perangkat Switch OLT ZTE.

3. Kestabilan Koneksi: Uji coba dilakukan untuk memeriksa kestabilan koneksi menggunakan Telnet ke perangkat Switch OLT ZTE, termasuk penanganan situasi-situasi yang mungkin terjadi seperti koneksi yang putus atau respons yang lambat.

Pengujian ini merupakan bagian penting dari siklus pengembangan perangkat lunak dan infrastruktur jaringan untuk memastikan bahwa sistem dapat beroperasi dengan baik dalam kondisi penggunaan sehari-hari dan dalam skenario yang mungkin kompleks. Hasil dari pengujian ini membantu memastikan bahwa sistem dapat diandalkan, aman, dan responsif sesuai dengan kebutuhan pengguna.

```
import os,sys
import logging
from datetime import datetime
from datetime import timedelta, date
```

```
import time
import logging
from time import sleep

fllog=os.path.join(os.getcwd(), 'bot.log')
logging.basicConfig(filemode='w',level=logging.ERROR,filename=fllog,
format='%(asctime)s - %(message)s',datefmt='%m/%d/%Y %I:%M:%S %p')
tn=None
flag = 1
pCMD='OLT-1-METAN.ID#' #'ZXAN#'

try:
    from dotenv import load_dotenv
    from subprocess import call
    import telebot
    import logging,platform,traceback
    from telnetlib import Telnet
except Exception as e:
    logging.error(e,exc_info=True)
    sys.exit()

load_dotenv()
BOT_TOKEN =  os.getenv('BOT_TOKEN')
AKUN_HOST =  os.getenv('HOST_OLT')
AKUN_USER =  os.getenv('USER_OLT')
AKUN_PASS =  os.getenv('PASS_OLT')

if BOT_TOKEN=='' or BOT_TOKEN==None:
    BOT_TOKEN = '6473337317:AAFNKBDArpHyU3SVeIVnS3WCm3CgeEl0_-k'
bot = telebot.TeleBot(BOT_TOKEN)
```

Pada source code diatas kita dapat menganalisis potongan kode tersebut untuk mengidentifikasi potensi fungsi atau tujuan dari kode tersebut:

1. **Logging Setup**:
    - Kode melakukan konfigurasi logging untuk mencatat pesan error ke file bot.log di direktori saat ini (os.getcwd()).

- Level logging diatur ke ERROR, yang berarti hanya pesan-pesan error yang akan dicatat.
- Format untuk log disetel agar mencakup timestamp (%(asctime)s) dan pesan error (%(message)s).

2. **Penggunaan Library**:
   - Kode mengimpor beberapa library standar seperti os, sys, logging, datetime, time, dan beberapa modul dari library telebot, dotenv, subprocess, dan telnetlib.
   - Penggunaan from dotenv import load_dotenv menunjukkan kode ini menggunakan file .env untuk mengambil variabel lingkungan seperti BOT_TOKEN, HOST_OLT, USER_OLT, dan PASS_OLT.
3. **Inisialisasi Bot Telegram**:
   - Kode menginisialisasi bot Telegram menggunakan telebot.TeleBot dengan token yang diambil dari variabel lingkungan BOT_TOKEN.
   - Jika BOT_TOKEN tidak ada dalam file .env, kode memberikan token bot default sebagai gantinya.
4. **Exception Handling**:
   - Terdapat blok try-except yang menangani ekspor modul yang diperlukan seperti telebot dan telnetlib. Jika ekspor gagal, pesan error dicatat ke log dan program keluar (sys.exit()).

Dari analisis tersebut, tidak ada fungsi eksplisit bernama sc dalam kode yang diberikan. Mungkin sc adalah singkatan atau variabel yang tidak terlihat dalam potongan kode tersebut, atau mungkin merupakan bagian dari implementasi yang lebih luas di luar kode yang diberikan.

```
def olt_connect(host,user,password):
    global tn
    tn=Telnet(host)
    tn.read_until(b"Username:")
    tn.write(user.encode('ascii') + b"\n")
    if password:
        tn.read_until(b"Password:")
        tn.write(password.encode('ascii') + b"\n")
    output=tn.read_until(b"DB>", 5)
    return output.decode('utf-8')
def olt_disconnect():
    tn.close
def cmd_uncfgONT():
    try:
        msg='Start uncfgONT'
        olt_connect(AKUN_HOST,AKUN_USER,AKUN_PASS)
        tn.write(b"show gpon onu uncfg\n")
        output=tn.read_until(b"DB>", 5)
        olt_disconnect()
        msg=output.decode('utf-8')
    except Exception as e:
        logging.error(e,exc_info=True)
        msg='uncfgONT Failed!'

    return msg
```

Pada source code diatas kita dapat terdapat tiga fungsi yang saling terkait untuk melakukan koneksi ke perangkat OLT melalui Telnet, menjalankan perintah, dan mengelola koneksi tersebut. Mari kita jelaskan fungsi-fungsi tersebut satu per satu:

Fungsi `olt_connect(host, user, password)`

Fungsi ini bertujuan untuk melakukan koneksi ke perangkat OLT menggunakan protokol Telnet.

Parameter:

1. host: Alamat host dari perangkat OLT yang akan dihubungi.
2. user: Username untuk otentikasi Telnet.
3. password: Password untuk otentikasi Telnet.

Langkah-langkah:

1. tn = Telnet(host): Membuat koneksi Telnet ke host yang ditentukan.
2. tn.read_until(b"Username:"): Membaca output dari koneksi Telnet sampai menemukan string "Username:".
3. tn.write(user.encode('ascii') + b"\n"): Mengirim username yang telah dienkripsi ke dalam format byte ASCII ke koneksi Telnet.
4. if password: ...: Jika ada password yang diberikan, maka:
    - tn.read_until(b"Password:"): Membaca output dari koneksi Telnet sampai menemukan string "Password:".
    - tn.write(password.encode('ascii') + b"\n"): Mengirim password yang telah dienkripsi ke dalam format byte ASCII ke koneksi Telnet.
5. output = tn.read_until(b"DB>", 5): Membaca output dari koneksi Telnet sampai menemukan string "DB>" dengan batasan waktu 5 detik.
6. return output.decode('utf-8'): Mengembalikan output dari perintah Telnet dalam bentuk string yang dienkripsi menggunakan UTF-8.

Fungsi

```
msg='Start uncfgONT'
        olt_connect(AKUN_HOST,AKUN_USER,AKUN_PASS)
        tn.write(b"show gpon onu uncfg\n")
```

Fungsi ini bertujuan untuk menjalankan perintah show gpon onu uncfg pada perangkat OLT setelah berhasil terhubung menggunakan Telnet. Jika berhasil, ia akan mengembalikan output dari perintah tersebut dalam bentuk string; jika tidak, akan mencatat pesan kesalahan dan mengembalikan pesan "uncfgONT Failed!".

```
def cmd_shPON(card,pon):
    try:
        msg='Start shPON'
        olt_connect(AKUN_HOST,AKUN_USER,AKUN_PASS)
        cmd='show running-config interface gpon-olt_1/{}/{}\n'.format(card,pon)
        tn.write(cmd.encode('ascii'))
        tn.write(b'exit\n')
        outputx=tn.read_until(b"DB>", 3)

        while True:
            tn.write(b'\n')
            output1=tn.read_until(b"\n\r",1)
            output1=output1.replace(b'\x08 ',''.encode('ascii'))
            output1=output1.replace(b'\x08',''.encode('ascii'))
            outputx +=output1
            if (pCMD in output1.decode('utf-8')):
                break
        outputx=outputx.decode('utf-8').replace('--More--','')
        olt_disconnect()
        msg=outputx
    except Exception as e:
        logging.error(e,exc_info=True)
        msg='shPON Failed!'

    return msg
```

Fungsi cmd_shPON(card, pon)

Fungsi cmd_shPON() ini dirancang untuk mengambil konfigurasi running dari port PON tertentu pada perangkat OLT melalui Telnet. Hal ini memungkinkan untuk memonitor atau mengelola konfigurasi secara otomatis dan mengatasi kondisi-kondisi khusus seperti kebutuhan untuk membaca lebih banyak output (--More--) menggunakan loop yang sesuai.

```
def cmd_addONT(card,pon,ont,type,sn):
    try:
        msg='Start addONT'
        olt_connect(AKUN_HOST,AKUN_USER,AKUN_PASS)
        tn.write(b'configure terminal\n')
        time.sleep(1)
        '''tn.read_until(b"%Info 2027:Enter Configuration commands, one per line. End
with CTRL/Z.")'''
        cmd='interface gpon-olt_1/{}/{}'.format(card,pon)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='onu {} type {} sn {}'.format(ont,type,sn)
        tn.write(cmd.encode('ascii')+b"\n")
        tn.write(b"exit\n")
        output=tn.read_until(b"DB>", 5)
        msg=output.decode('utf-8')
    except Exception as e:
        logging.error(e,exc_info=True)
        msg='addONT Failed!'
    return msg
```

Fungsi cmd_addONT

Fungsi ini bertujuan untuk memfasilitasi penambahan ONT ke dalam jaringan GPON dengan mengirimkan perintah melalui Telnet ke perangkat OLT, dan memberikan umpan balik tentang keberhasilan atau kegagalan operasi tersebut melalui variabel msg.

```
def cmd_delONT(card,pon,ont):
    try:
```

```
        msg='Start delONT'
        olt_connect(AKUN_HOST,AKUN_USER,AKUN_PASS)
        tn.write(b'configure terminal\n')
        time.sleep(1)
        cmd='interface gpon-olt_1/{}/{}'.format(card,pon)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='no onu {}'.format(ont)
        tn.write(cmd.encode('ascii')+b"\n")
        tn.write(b"exit\n")
        output=tn.read_until(b"DB>", 5)
        msg=output.decode('utf-8')
    except Exception as e:
        logging.error(e,exc_info=True)
        msg='delONT Failed!'
    return msg
```

Fungsi ini mirip dengan cmd_addONT, tetapi bertujuan untuk menghapus ONT dari jaringan GPON. Ini melakukan pengaturan dan mengirim perintah melalui Telnet ke OLT untuk menjalankan operasi penghapusan ONT yang dimaksud.

```
def cmd_rebootONT(card,pon,ont):
    try:
        msg='Start rebootONT'
        olt_connect(AKUN_HOST,AKUN_USER,AKUN_PASS)
        tn.write(b'configure terminal\n')
        time.sleep(1)
        cmd='pon-onu-mng gpon-onu_1/{}/{}:{}'.format(card,pon,ont)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='reboot'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='yes'
        tn.write(cmd.encode('ascii')+b"\n")
        time.sleep(1)
        tn.write(b"exit\n")
        output=tn.read_until(b"DB>", 5)
        msg=output.decode('utf-8')
    except Exception as e:
        logging.error(e,exc_info=True)
        msg='rebootONT Failed!'
    return msg
```

Fungsi dari source code cmd_rebootONT adalah untuk melakukan reboot pada ONT (Optical Network Terminal) pada perangkat GPON OLT menggunakan Telnet.Ini memungkinkan untuk mengelola ONT pada perangkat GPON OLT dengan melakukan operasi reboot pada ONT yang ditentukan.

```
def cmd_resetONT(card,pon,ont):
    try:
        msg='Start resetONT'
        olt_connect(AKUN_HOST,AKUN_USER,AKUN_PASS)
        tn.write(b'configure terminal\n')
        time.sleep(1)
        cmd='pon-onu-mng gpon-onu_1/{}/{}:{}'.format(card,pon,ont)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='restore factory'
        tn.write(cmd.encode('ascii')+b"\n")
        time.sleep(1)
        output=tn.read_until(b"DB>", 5)
        tn.write(b"exit\n")
        msg=output.decode('utf-8')
        msg='resetONT sukses!'
    except Exception as e:
        logging.error(e,exc_info=True)
        msg='resetONT Failed!'
    return msg
```

Fungsi dari source code reset_ONT adalah melakukan reset pada perangkat modem yang ada pada rumah atau kantor pelanggan, sehingga teknisi atau NOC tidak perlu kerumah pelanggan hanya dengan mengirimkan script ini dapat melakukan reset pada modem perangkat

```
def cmd_write():
    try:
        msg='Start shINT'
        olt_connect(AKUN_HOST,AKUN_USER,AKUN_PASS)
        #tn.write(b'configure terminal\n')
        #time.sleep(1)
        cmd='write'
```

```
        tn.write(cmd.encode('ascii')+b"\n")
        tn.write(b"exit\n")
        output=tn.read_until(b"DB>", 5)
        msg=output.decode('utf-8')
    except Exception as e:
        logging.error(e,exc_info=True)
        msg='bootONT Failed!'
    return msg
```

fungsi cmd_write berfungsi untuk melakukan operasi 'write' pada perangkat GPON OLT menggunakan Telnet, dan mengembalikan pesan yang sesuai berdasarkan hasil operasi yang berhasil atau gagal.

```
def cmd_shINT(card,pon,ont):
    try:
        msg='Start write'
        olt_connect(AKUN_HOST,AKUN_USER,AKUN_PASS)
        cmd='show run int gpon-onu_1/{}/{}:{}'.format(card,pon,ont)
        tn.write(cmd.encode('ascii')+b"\n")
        tn.write(b"exit\n")
        output=tn.read_until(b"DB>", 5)
        msg=output.decode('utf-8')
    except Exception as e:
        logging.error(e,exc_info=True)
        msg='bootONT Failed!'
    return msg
```

Fungsi cmd_shINT adalah untuk melakukan koneksi ke perangkat GPON OLT, menampilkan konfigurasi dari sebuah interface GPON ONT berdasarkan parameter yang diberikan, dan mengembalikan hasilnya dalam bentuk teks yang bisa dibaca. Fungsi ini juga menangani penanganan kesalahan jika terjadi masalah dalam proses koneksi atau eksekusi perintah.

```
def cmd_shMNG(card,pon,ont):
    try:
        msg='Start write'
        olt_connect(AKUN_HOST,AKUN_USER,AKUN_PASS)
        cmd='show onu run conf gpon-onu_1/{}/{}:{}'.format(card,pon,ont)
```

```
        tn.write(cmd.encode('ascii')+b"\n")
        tn.write(b"exit\n")
        output=tn.read_until(b"DB>", 5)
        msg=output.decode('utf-8')
    except Exception as e:
        logging.error(e,exc_info=True)
        msg='bootONT Failed!'
    return msg
```

fungsi cmd_shMNG adalah untuk melakukan koneksi ke perangkat GPON OLT, menampilkan konfigurasi dari sebuah ONT berdasarkan parameter yang diberikan, dan mengembalikan hasilnya dalam bentuk teks yang bisa dibaca. Fungsi ini juga menangani penanganan kesalahan jika terjadi masalah dalam proses koneksi atau eksekusi perintah.

```
def cmd_sttPON(card,pon):
    try:
        msg='Start sttPON'
        olt_connect(AKUN_HOST,AKUN_USER,AKUN_PASS)
        cmd='show gpon onu state gpon-olt_1/{}/{}'.format(card,pon)
        tn.write(cmd.encode('ascii')+b"\n")
        tn.write(b"exit\n")
        output=tn.read_until(b"DB>", 5)
        msg=output.decode('utf-8')
    except Exception as e:
        logging.error(e,exc_info=True)
        msg='bootONT Failed!'
    return msg
```

fungsi cmd_sttPON adalah untuk Membuat koneksi ke perangkat OLT.

Mengirim perintah untuk memeriksa status GPON untuk slot card dan nomor PON tertentu.Membaca dan mengembalikan hasil status dari perangkat.Menangani dan melaporkan kesalahan jika terjadi masalah selama eksekusi.

```
def cmd_attONT(card,pon,ont):
```

```
    try:
        msg='Start attONT'
        olt_connect(AKUN_HOST,AKUN_USER,AKUN_PASS)
        cmd='show pon power attenuation gpon-onu_1/{}/{}:{}'.format(card,pon,ont)
        tn.write(cmd.encode('ascii')+b"\n")
        tn.write(b"exit\n")
        output=tn.read_until(b"DB>", 3)
        msg=output.decode('utf-8')
        if msg=='':
            msg='Start attONT sukses!'
    except Exception as e:
        logging.error(e,exc_info=True)
        msg='attONT Failed!'
    return msg
```

fungsi cmd_attONT adalah untuk Membuat koneksi ke perangkat OLT.

Mengirim perintah untuk mengukur daya attenuasi ONT untuk card, PON, dan ONT tertentu. Membaca dan mengembalikan hasil dari output perintah.

Menangani dan melaporkan kesalahan jika terjadi masalah selama eksekusi.

```
def cmd_attPON(card,pon):
    try:
        msg='Start attPON'
        olt_connect(AKUN_HOST,AKUN_USER,AKUN_PASS)
        cmd='show pon power onu-rx gpon-olt_1/{}/{}'.format(card,pon)
        tn.write(cmd.encode('ascii')+b"\n")
        outputx=tn.read_until(b"DB>", 3)
        while True:
            tn.write(b'\n')
            output1=tn.read_until(b"\n\r",1)
            output1=output1.replace(b'\x08 ',''.encode('ascii'))
            output1=output1.replace(b'\x08',''.encode('ascii'))
            outputx +=output1
            if (pCMD in output1.decode('utf-8')):
                break
        tn.write(b"exit\n")
        outputx=outputx.decode('utf-8').replace('--More--','')
        #olt_disconnect()
        msg=outputx
```

```
    except Exception as e:
        logging.error(e,exc_info=True)
        msg='attPON Failed!'
    return msg
```

fungsi cmd_attPON adalah Membuat koneksi ke perangkat OLT. Mengirim perintah untuk mendapatkan daya penerimaan PON untuk card dan PON tertentu.Membaca output dari perangkat, menangani output yang panjang dengan loop, dan membersihkan hasilnya. Menangani dan melaporkan kesalahan jika terjadi masalah selama eksekusi.

```
def cmd_shONU(card,pon,ont):
    try:
        msg='Start shONU'
        olt_connect(AKUN_HOST,AKUN_USER,AKUN_PASS)
        cmd='show gpon onu detail-info gpon-onu_1/{}/{}:{}'.format(card,pon,ont)
        tn.write(cmd.encode('ascii')+b"\n")
        outputx=tn.read_until(b"DB>", 3)
        while True:
            tn.write(b'\n')
            output1=tn.read_until(b"\n\r",1)
            output1=output1.replace(b'\x08 ',''.encode('ascii'))
            output1=output1.replace(b'\x08',''.encode('ascii'))
            outputx +=output1
            if (pCMD in output1.decode('utf-8')):
                break
        tn.write(b"exit\n")
        outputx=outputx.decode('utf-8').replace('--More--','')
        msg=outputx
    except Exception as e:
        logging.error(e,exc_info=True)
        msg='shONU Failed!'
    return msg
```

fungsi cmd_shONU Membuat koneksi ke perangkat OLT.Mengirim perintah untuk mendapatkan informasi detail ONT untuk card, PON, dan ONT tertentu.

Membaca output dari perangkat, menangani output yang panjang dengan loop, dan membersihkan hasilnya. Menangani dan melaporkan kesalahan jika terjadi masalah selama eksekusi.

```
def cmd_setPppoe(card,pon,ont,name,vlan,user,psswd):
    try:
        msg='Start setPppoe'
        olt_connect(AKUN_HOST,AKUN_USER,AKUN_PASS)
        tn.write(b'configure terminal\n')
        time.sleep(1)
        cmd='interface gpon-onu_1/{}/{}:{}'.format(card,pon,ont)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='name {}'.format(name)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tcont 1 profile UP-PPPOE'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tcont 2 profile UP-MNG'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='gemport 1 name PPPOE tcont 1'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='gemport 2 name TR-69 tcont 2'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='service-port 1 vport 1 user-vlan {} vlan {}'.format(vlan,vlan)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='service-port 2 vport 2 user-vlan 100 vlan 100'
        tn.write(cmd.encode('ascii')+b"\n\r")
        cmd='!'
        tn.write(cmd.encode('ascii')+b"\n\r")
        cmd='pon-onu-mng gpon-onu_1/{}/{}:{}'.format(card,pon,ont)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tr069-mgmt 1 state unlock'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tr069-mgmt 1 acs http://acs@metan.id:7547 validate basic username
acs@metan.id password acs@metan.id'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tr069-mgmt 1 tag pri 7 vlan 100'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='service PPPOE gemport 1 iphost 1 vlan {}'.format(vlan)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='service TR-69 gemport 2 iphost 2 vlan 100'
```

```
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='pppoe 1 nat enable user {} password {}'.format(user,psswd)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='security-mgmt 1 state enable mode forward protocol web telnet'
        tn.write(cmd.encode('ascii')+b"\n\r")
        time.sleep(1)
        cmd='!'
        tn.write(cmd.encode('ascii')+b"\n\r")
        tn.write(b"exit\n")
        output=tn.read_until(b"DB>", 5)
        msg=output.decode('utf-8')
    except Exception as e:
        logging.error(e,exc_info=True)
        msg='setPppoe Failed!'
    return msg
```

Fungsi `cmd_setPppoe` membuat koneksi ke perangkat OLT, Masuk ke mode konfigurasi terminal, mengonfigurasi interface GPON, PPPoE, TR-069, dan VLAN, mengatur pengaturan manajemen TR-069 dan PPPoE, mengaktifkan mode keamanan, menutup sesi Telnet dan membaca hasil output, dan menangani dan melaporkan kesalahan jika terjadi masalah selama konfigurasi.

```
def cmd_setPppoe_HS(card,pon,ont,name,vlan,user,psswd,vlanhs):
    try:
        msg='Start setPppoe'
        olt_connect(AKUN_HOST,AKUN_USER,AKUN_PASS)
        tn.write(b'configure terminal\n')
        time.sleep(1)
        cmd='interface gpon-onu_1/{}/{}:{}'.format(card,pon,ont)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='name {}'.format(name)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tcont 1 profile UP-PPPOE'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tcont 2 profile UP-HOTSPOT'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='gemport 1 name PPPOE tcont 1'
        tn.write(cmd.encode('ascii')+b"\n")
```

```
        cmd='gemport 2 name HOTSPOT tcont 2'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='service-port 1 vport 1 user-vlan {} vlan {}'.format(vlan,vlan)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='service-port 2 vport 2 user-vlan {} vlan {}'.format(vlanhs,vlanhs)
        tn.write(cmd.encode('ascii')+b"\n\r")
        cmd='!'
        tn.write(cmd.encode('ascii')+b"\n\r")
        cmd='pon-onu-mng gpon-onu_1/{}/{}:{}'.format(card,pon,ont)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='service PPPOE gemport 1 iphost 1 vlan {}'.format(vlan)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tr069-mgmt 1 state unlock'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tr069-mgmt 1 acs http://acs@metan.id:7547 validate basic username
acs@metan.id password acs@metan.id'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tr069-mgmt 1 tag pri 7 vlan 100'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='service HS gemport 2 vlan {}'.format(vlanhs)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='pppoe 1 nat enable user {} password {}'.format(user,psswd)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='vlan port wifi_0/2 mode tag vlan {}'.format(vlanhs)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='ssid auth wep wifi_0/2 open-system'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='vlan port wifi_0/6 mode tag vlan {}'.format(vlanhs)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='ssid auth wep wifi_0/6 open-system'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='security-mgmt 1 state enable mode forward protocol web telnet'
        tn.write(cmd.encode('ascii')+b"\n\r")
        time.sleep(1)
        cmd='!'
        tn.write(cmd.encode('ascii')+b"\n\r")
        tn.write(b"exit\n")
        output=tn.read_until(b"DB>", 5)
        msg=output.decode('utf-8')
    except Exception as e:
        logging.error(e,exc_info=True)
        msg='setPppoe Failed!'
```

```
    return msg
```

Fungsi `cmd_setPppoe_HS` membuat koneksi ke perangkat OLT, masuk ke mode konfigurasi terminal, mengonfigurasi interface GPON, PPPoE, dan Hotspot, mengatur pengaturan VLAN dan Wi-Fi, mengaktifkan mode keamanan, menutup sesi Telnet dan membaca hasil output, dan menangani dan melaporkan kesalahan jika terjadi masalah.

```
def cmd_setFH(card,pon,ont,name,vlan):
    try:
        msg='Start setFH'
        olt_connect(AKUN_HOST,AKUN_USER,AKUN_PASS)
        tn.write(b'configure terminal\n')
        time.sleep(1)
        cmd='interface gpon-onu_1/{}/{}:{}'.format(card,pon,ont)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='name {}'.format(name)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tcont 1 profile UP-PPPOE'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='gemport 1 name PPPOE tcont 1'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='gemport 1 traffic-limit downstream DOWN-PPPOE'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='service-port 1 vport 1 user-vlan {} vlan {}'.format(vlan,vlan)
        tn.write(cmd.encode('ascii')+b"\n\r")
        cmd='!'
        tn.write(cmd.encode('ascii')+b"\n\r")
        cmd='pon-onu-mng gpon-onu_1/{}/{}:{}'.format(card,pon,ont)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tr069-mgmt 1 state unlock'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tr069-mgmt 1 acs http://acs@metan.id:7547 validate basic username
acs@metan.id password acs@metan.id'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tr069-mgmt 1 tag pri 7 vlan 100'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='service PPPOE gemport 1 iphost 1 vlan {}'.format(vlan)
```

```
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='vlan port veip_1 mode hybrid def-vlan {}'.format(vlan)
        tn.write(cmd.encode('ascii')+b"\n\r")
        cmd='!'
        tn.write(cmd.encode('ascii')+b"\n\r")
        tn.write(b"exit\n")
        output=tn.read_until(b"DB>", 5)
        msg=output.decode('utf-8')
        if output==":
            output=='setFH Sukses!'
    except Exception as e:
        logging.error(e,exc_info=True)
        msg='setFH Failed!'
    return msg
```

Fungsi `cmd_setFH` membuat koneksi ke perangkat OLT, masuk ke mode konfigurasi terminal, mengonfigurasi interface GPON untuk PPPoE, termasuk pengaturan Tcont, gemport, dan service port, mengonfigurasi manajemen TR-069 dan layanan PPPoE, mengatur VLAN untuk port VEIP, menutup sesi Telnet dan membaca hasil output, dan menangani dan melaporkan kesalahan jika terjadi masalah.

```
def cmd_setFH2(card,pon,ont,name,vlan,vlanh):
    try:
        msg='Start setFH'
        olt_connect(AKUN_HOST,AKUN_USER,AKUN_PASS)
        tn.write(b'configure terminal\n')
        time.sleep(1)
        cmd='interface gpon-onu_1/{}/{}:{}'.format(card,pon,ont)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='name {}'.format(name)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tcont 1 profile UP-PPPOE'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tcont 2 profile UP-HOTSPOT'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tcont 3 profile MONITORING'
        tn.write(cmd.encode('ascii')+b"\n")
```

```
        cmd='gemport 1 name PPPOE tcont 1'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='gemport 2 name HS tcont 2'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='gemport 3 name TR069 tcont 3'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='service-port 1 vport 1 user-vlan {} vlan {}'.format(vlan,vlan)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='service-port 2 vport 2 user-vlan 69 vlan 69'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='service-port 3 vport 3 user-vlan {} vlan {}'.format(vlanh,vlanh)
        tn.write(cmd.encode('ascii')+b"\n\r")
        cmd='!'
        tn.write(cmd.encode('ascii')+b"\n\r")
        cmd='pon-onu-mng gpon-onu_1/{}/{}:{}'.format(card,pon,ont)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='service PPPOE gemport 1 iphost 1 vlan {}'.format(vlan)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='service TR069/HS gemport 3 vlan {}'.format(vlanh)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='service TR069-2 gemport 2 vlan 69'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='vlan port veip_1 mode hybrid def-vlan {}'.format(vlan)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='vlan port veip_1 vlan 69,{}'.format(vlanh)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tr069-mgmt 1 state unlock'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tr069-mgmt 1 acs http://acs@metan.id:7547 validate basic username
acs@metan.id password acs@metan.id'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tr069-mgmt 1 tag pri 7 vlan 69'
        tn.write(cmd.encode('ascii')+b"\n\r")
        cmd='!'
        tn.write(cmd.encode('ascii')+b"\n\r")
        tn.write(b"exit\n")
        output=tn.read_until(b"DB>", 5)
        msg=output.decode('utf-8')
    except Exception as e:
        logging.error(e,exc_info=True)
        msg='bootONT Failed!'
    return msg
```

Fungsi `cmd_setFH2` membuat koneksi Telnet ke perangkat OLT, masuk ke mode konfigurasi terminal, mengonfigurasi interface GPON dengan parameter Tcont, gemport, dan service port untuk PPPoE, Hotspot, dan Monitoring, mengatur layanan dan VLAN untuk interface VEIP, mengonfigurasi manajemen TR-069, menutup sesi Telnet dan membaca output dari perangkat, dan menangani dan melaporkan kesalahan jika terjadi masalah.

```
def cmd_setHSMONITOR(card,pon,ont,name,vlan,user,pwd,vlanh):
    try:
        msg='Start setHSMONITOR'
        olt_connect(AKUN_HOST,AKUN_USER,AKUN_PASS)
        tn.write(b'configure terminal\n')
        time.sleep(1)
        cmd='interface gpon-onu_1/{}/{}:{}'.format(card,pon,ont)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='name {}'.format(name)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tcont 1 profile UP-MNG'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tcont 2 profile UP-HOTSPOT'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='gemport 1 name MONITORHS tcont 1'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='gemport 2 name HS tcont 2'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='service-port 1 vport 1 user-vlan {} vlan {}'.format(vlan,vlan)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='service-port 2 vport 2 user-vlan {} vlan {}'.format(vlanh,vlanh)
        tn.write(cmd.encode('ascii')+b"\n\r")
        cmd='!'
        tn.write(cmd.encode('ascii')+b"\n\r")
        cmd='pon-onu-mng gpon-onu_1/{}/{}:{}'.format(card,pon,ont)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='service PPPOE gemport 1 iphost 1 vlan {}'.format(vlan)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='service HS gemport 2 vlan {}'.format(vlanh)
        tn.write(cmd.encode('ascii')+b"\n")
```

```
        cmd='pppoe 1 nat enable user {} password {}'.format(user,pwd)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='vlan port wifi_0/1 mode tag vlan {}'.format(vlanh)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='ssid auth wep wifi_0/1 open-system'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='vlan port wifi_0/5 mode tag vlan {}'.format(vlanh)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='ssid auth wep wifi_0/5 open-system'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='vlan port eth_0/1 mode tag vlan {}'.format(vlanh)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='vlan port eth_0/2 mode tag vlan {}'.format(vlanh)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='vlan port eth_0/3 mode tag vlan {}'.format(vlanh)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='vlan port eth_0/4 mode tag vlan {}'.format(vlanh)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tr069-mgmt 1 state unlock'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tr069-mgmt 1 acs http://acs@metan.id:7547 validate basic username
acs@metan.id password acs@metan.id'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tr069-mgmt 1 tag pri 7 vlan 100'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='security-mgmt 1 state enable mode forward protocol web telnet'
        tn.write(cmd.encode('ascii')+b"\n\r")
        cmd='!'
        tn.write(cmd.encode('ascii')+b"\n\r")
        tn.write(b"exit\n")
        output=tn.read_until(b"DB>", 5)
        msg=output.decode('utf-8')
    except Exception as e:
        logging.error(e,exc_info=True)
        msg='setHSMONITOR Failed!'
    return msg
```

Fungsi `cmd_setHSMONITOR` membuat koneksi Telnet ke perangkat OLT, masuk ke mode konfigurasi terminal, mengonfigurasi interface GPON dengan parameter Tcont, gemport, dan service port untuk Monitoring dan Hotspot, mengatur

layanan dan VLAN untuk interface Wi-Fi dan Ethernet, mengonfigurasi manajemen TR-069, menutup sesi Telnet dan membaca output dari perangkat, dan menangani dan melaporkan kesalahan jika terjadi masalah.

```
def cmd_setBridge(card,pon,ont,name,vlan):
    try:
        msg='Start setBridge'
        olt_connect(AKUN_HOST,AKUN_USER,AKUN_PASS)
        tn.write(b'configure terminal\n')
        time.sleep(1)
        cmd='interface gpon-onu_1/{}/{}:{}'.format(card,pon,ont)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='name {}'.format(name)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='tcont 1 profile UP-PPPOE'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='gemport 1 name BRIDGE tcont 1'
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='service-port 1 vport 1 user-vlan {} vlan {}'.format(vlan,vlan)
        tn.write(cmd.encode('ascii')+b"\n")
        tn.write(cmd.encode('ascii')+b"\n\r")
        cmd='!'
        tn.write(cmd.encode('ascii')+b"\n\r")
        cmd='pon-onu-mng gpon-onu_1/{}/{}:{}'.format(card,pon,ont)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='service BRIDGE gemport 1 vlan {}'.format(vlan)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='vlan port eth_0/1 mode tag {}'.format(vlan)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='vlan port eth_0/2 mode tag {}'.format(vlan)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='vlan port eth_0/3 mode tag {}'.format(vlan)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='vlan port eth_0/4 mode tag {}'.format(vlan)
        tn.write(cmd.encode('ascii')+b"\n")
        cmd='security-mgmt 1 state enable mode forward protocol web telnet'
        tn.write(cmd.encode('ascii')+b"\n\r")
        cmd='!'
        tn.write(cmd.encode('ascii')+b"\n\r")
        tn.write(b"exit\n")
```

```
        output=tn.read_until(b"DB>", 5)
        msg=output.decode('utf-8')
    except:
        logging.error(e,exc_info=True)
        msg='setBridge Failed!'
    return msg
```

Fungsi `cmd_setBridge` membuka koneksi Telnet ke perangkat OLT, masuk ke mode konfigurasi terminal,mengonfigurasi interface GPON dengan parameter Tcont, gemport, dan service port untuk bridge, mengatur VLAN untuk port Ethernet, mengonfigurasi manajemen keamanan, menutup sesi Telnet dan membaca output dari perangkat, dan menangani dan melaporkan kesalahan jika terjadi masalah.

```
def cmd_shoPON(card,pon):
    try:
        msg='Start write'
        olt_connect(AKUN_HOST,AKUN_USER,AKUN_PASS)
        cmd='sho pon power onu-rx gpon-olt_1/{}/{}'.format(card,pon)
        tn.write(cmd.encode('ascii')+b"\n")
        #tn.write(b"exit\n")
        outputx=tn.read_until(b"--More--", 3)
        tn.write(b'\n\r')
        time.sleep(1)
        while True:
            try:
                output1=tn.read_until(b"--More--", 2)
                if (pCMD in output1.decode('utf-8')):
                    tn.write(b"exit\n")
                    time.sleep(1)
                    break
                else:
                    output1=output1.replace(b'\x08 ',''.encode('ascii'))
                    output1=output1.replace(b'\x08','.encode('ascii'))
                    outputx +=output1
                    tn.write(b'\n\r')
                    time.sleep(1)
            except:
```

```
            print(output1)
            break

        outputx=outputx.decode('utf-8').replace('--More--','')
        print('outputx3',outputx)
        msg=outputx
    except Exception as e:
        logging.error(e,exc_info=True)
        msg='shoPON Failed!'
```

Fungsi `cmd_shoPON` membuka koneksi Telnet ke perangkat OLT, mengirim perintah untuk menampilkan informasi tentang PON (Passive Optical Network), mengambil output dari perangkat OLT, termasuk menangani output bertingkat (jika ada) dengan membaca data tambahan hingga seluruh hasil diperoleh, menghapus karakter khusus dari output dan membersihkan hasilnya, dan menangani kesalahan dan mengembalikan hasil atau pesan kesalahan.

```
def cmd_sendMsg(message,hasil):
    if len(hasil) > 4096:
        for x in range(0, len(hasil), 4096):
            bot.reply_to(message, hasil[x:x+4096])
        else:
            bot.reply_to(message, hasil)
    else:
        bot.reply_to(message, hasil)
```

Fungsi `cmd_sendMsg` memeriksa apakah panjang pesan (`hasil`) melebihi 4096 karakter, jika ya, membagi pesan menjadi potongan-potongan kecil dan mengirimkan setiap potongan secara terpisah, jika tidak, mengirimkan pesan secara langsung, dan menggunakan metode `bot.reply_to` untuk mengirim pesan kepada pengguna yang sesuai.

```
def send_setBridge(message):
```

```
    cmd=message.text.split(" ")
    cmd1=cmd[1].split('/')
    card=cmd1[0]
    pon=cmd1[1]
    ont=cmd[2]
    name=cmd[3]
    vlan=cmd[4]
    hasil=cmd_setBridge(card,pon,ont,name,vlan)
    cmd_sendMsg(message, hasil)
```

Fungsi `send_setBridge` mengambil dan memproses perintah dari pesan dengan memecah teks pesan menjadi parameter yang diperlukan untuk konfigurasi, menggunakan parameter tersebut untuk memanggil fungsi `cmd_setBridge`, yang melakukan konfigurasi berdasarkan parameter yang diberikan, dan mengirimkan hasil konfigurasi menggunakan fungsi `cmd_sendMsg` untuk memberikan umpan balik kepada pengirim pesan.

```
def send_setHSMONITOR(message):
    cmd=message.text.split(" ")
    cmd1=cmd[1].split('/')
    card=cmd1[0]
    pon=cmd1[1]
    ont=cmd[2]
    name=cmd[3]
    vlan=cmd[4]
    user=cmd[5]
    pwd=cmd[6]
    vlanh=cmd[7]
    hasil=cmd_setHSMONITOR(card,pon,ont,name,vlan,user,pwd,vlanh)
    cmd_sendMsg(message, hasil)
```

Fungsi `send_setHSMONITOR` mengambil dan memproses perintah dari pesan dengan memecah teks pesan menjadi parameter yang diperlukan untuk konfigurasi, menggunakan parameter tersebut untuk memanggil fungsi `cmd_setHSMONITOR`, yang melakukan konfigurasi berdasarkan parameter

yang diberikan, dan mengirimkan hasil konfigurasi menggunakan fungsi `cmd_sendMsg` untuk memberikan umpan balik kepada pengirim pesan.

```
def send_setFH2(message):
    cmd=message.text.split(" ")
    cmd1=cmd[1].split('/')
    card=cmd1[0]
    pon=cmd1[1]
    ont=cmd[2]
    name=cmd[3]
    vlan=cmd[4]
    vlanh=cmd[5]
    hasil=cmd_setFH2(card,pon,ont,name,vlan,vlanh)
    cmd_sendMsg(message, hasil)
```

Fungsi `send_setFH2` mengambil dan memproses perintah dari pesan dengan memecah teks pesan menjadi parameter yang diperlukan untuk konfigurasi, menggunakan parameter tersebut untuk memanggil fungsi `cmd_setFH2`, yang melakukan konfigurasi berdasarkan parameter yang diberikan, dan mengirimkan hasil konfigurasi menggunakan fungsi `cmd_sendMsg` untuk memberikan umpan balik kepada pengirim pesan.

```
def send_setFH(message):
    cmd=message.text.split(" ")
    cmd1=cmd[1].split('/')
    card=cmd1[0]
    pon=cmd1[1]
    ont=cmd[2]
    name=cmd[3]
    vlan=cmd[4]
    hasil=cmd_setFH(card,pon,ont,name,vlan)
    cmd_sendMsg(message, hasil)
```

Fungsi `send_setFH` mengambil dan memproses perintah dari pesan dengan memecah teks pesan menjadi parameter yang diperlukan untuk konfigurasi,

menggunakan parameter tersebut untuk memanggil fungsi `cmd_setFH`, yang melakukan konfigurasi berdasarkan parameter yang diberikan, dan mengirimkan hasil konfigurasi menggunakan fungsi `cmd_sendMsg` untuk memberikan umpan balik kepada pengirim pesan.

```
def send_shONU(message):
    #/shONU info 1/6 1
    cmd=message.text.split(" ")
    cmd1=cmd[2].split('/')
    card=cmd1[0]
    pon=cmd1[1]
    ont=cmd[3]
    hasil=cmd_shONU(card,pon,ont)
    cmd_sendMsg(message, hasil)
```

Fungsi `send_shONU` mengambil dan memproses perintah dari pesan dengan memecah teks pesan menjadi parameter yang diperlukan, menggunakan parameter tersebut untuk memanggil fungsi `cmd_shONU`, yang melakukan operasi atau query sesuai dengan parameter yang diberikan, dan mengirimkan hasil dari fungsi `cmd_shONU` sebagai balasan kepada pengirim pesan menggunakan `cmd_sendMsg`.

```
def send_attPON(message):
    #attPON 1/1/6
    cmd=message.text.split(" ")
    cmd1=cmd[1].split('/')
    card=cmd1[0]
    pon=cmd1[1]
    hasil=cmd_attPON(card,pon)
    cmd_sendMsg(message, hasil)
```

Fungsi `send_attPON` mengambil dan memproses perintah dari pesan dengan memecah teks pesan menjadi parameter yang diperlukan, menggunakan parameter

tersebut untuk memanggil fungsi `cmd_attPON`, yang melakukan operasi atau query sesuai dengan parameter yang diberikan, dan mengirimkan hasil dari fungsi `cmd_attPON` sebagai balasan kepada pengirim pesan menggunakan `cmd_sendMsg`.

```
def send_attONT(message):
    #/attONT 1/1 6
    cmd=message.text.split(" ")
    cmd1=cmd[1].split('/')
    card=cmd1[0]
    pon=cmd1[1]
    ont=cmd[2]
    hasil=cmd_attONT(card,pon,ont)
    cmd_sendMsg(message, hasil)
```

Fungsi `send_attONT` memproses pesan untuk mendapatkan parameter `card`, `pon`, dan `ont` dari format pesan yang diterima, menggunakan parameter tersebut untuk memanggil fungsi `cmd_attONT`, yang akan melakukan tindakan konfigurasi atau query berdasarkan parameter yang diberikan, dan mengirimkan hasil dari fungsi `cmd_attONT` kembali ke pengirim pesan dengan menggunakan `cmd_sendMsg`.

```
def send_sttPON(message):
    cmd=message.text.split(" ")
    cmd1=cmd[1].split('/')
    card=cmd1[0]
    pon=cmd1[1]
    hasil=cmd_sttPON(card,pon)
    cmd_sendMsg(message, hasil)
```

Fungsi `send_sttPON` memproses pesan untuk mendapatkan parameter `card` dan `pon` dari format pesan yang diterima, menggunakan parameter tersebut untuk memanggil fungsi `cmd_sttPON`, yang akan melakukan

query untuk mendapatkan status PON berdasarkan parameter yang diberikan, dan mengirimkan hasil dari fungsi `cmd_sttPON` kembali ke pengirim pesan menggunakan `cmd_sendMsg`.

```
def send_shMNG(message):
    cmd=message.text.split(" ")
    cmd1=cmd[1].split('/')
    card=cmd1[0]
    pon=cmd1[1]
    ont=cmd[2]
    hasil=cmd_shMNG(card,pon,ont)
    cmd_sendMsg(message, hasil)
```

Fungsi `send_shMNG` memproses pesan untuk mendapatkan parameter `card`, `pon`, dan `ont` dari format pesan yang diterima, menggunakan parameter tersebut untuk memanggil fungsi `cmd_shMNG`, yang akan melakukan query untuk mendapatkan informasi manajemen berdasarkan parameter yang diberikan, dan mengirimkan hasil dari fungsi `cmd_shMNG` kembali ke pengirim pesan menggunakan `cmd_sendMsg`.

```
def send_shINT(message):
    cmd=message.text.split(" ")
    cmd1=cmd[1].split('/')
    card=cmd1[0]
    pon=cmd1[1]
    ont=cmd[2]
    hasil=cmd_shINT(card,pon,ont)
    cmd_sendMsg(message, hasil)
```

Fungsi `send_shINT` memproses pesan untuk mengekstrak parameter `card`, `pon`, dan `ont` dari format pesan yang diterima, menggunakan parameter yang diekstrak untuk memanggil fungsi `cmd_shINT`, yang mengquery informasi antarmuka (INT)

berdasarkan parameter tersebut, dan mengirimkan hasil dari `cmd_shINT` kembali ke pengirim pesan menggunakan `cmd_sendMsg`.

```
def send_write(message):
    hasil=cmd_write()
    cmd_sendMsg(message, hasil)
```

Fungsi `send_write` memanggil `cmd_write` untuk menjalankan perintah penulisan, menyimpan hasil dari `cmd_write` dalam variabel `hasil`, dan mengirimkan hasil tersebut ke pengirim pesan menggunakan `cmd_sendMsg`.

```
@bot.message_handler(commands=['start'])
def send_welcome(message):
    bot.reply_to(message, "Selamat Datang di bot OLT-ZTE\n")
```

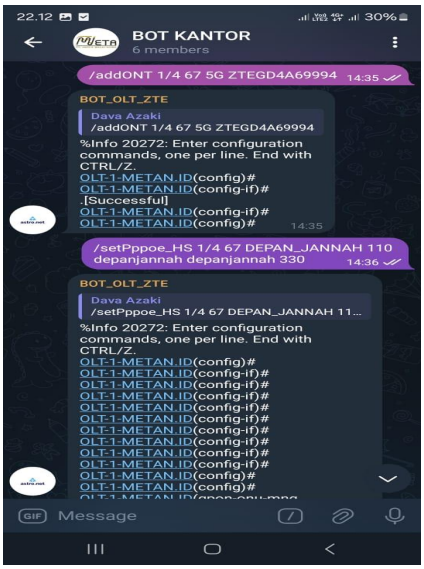
Fungsi send_welcome bertugas menyambut pengguna baru dengan pesan ketika perintah /start diterima, dan pesan sambutan yang dikirim adalah: "Selamat Datang di bot OLT-ZTE\n".

```
@bot.message_handler(commands=['help'])
def send_welcome1(message):
    bot.reply_to(message, "\
    /help\n\
/uncfgONT\n\
/shPON $card/$pon (nengok bangku)\n\
/addONT $card/$pon $ont $type $sn (regis ont)\n\
/setPppoe $card/$pon $ont $name $vlan $user $psswd (wihome only)\n\
/setPppoe_HS $card/$pon $ont $name $vlan $user $psswd $vlanhs (Wihome &
hs(ssid2&6))\n\
/setFH $card/$pon $ont $name $vlan (wihome fiberhome)\n\
/setFH2 $card/$pon $ont $name $vlan $vlanhs (wihome & hs)\n\
/setHSMONITOR $card/$pon $ont $name $vlanMTR(110) $user $psswd $vlanHS(zte
& monitor)\n\
/setBridge $card/$pon $ont $name $vlan (Broadband)\n\
```

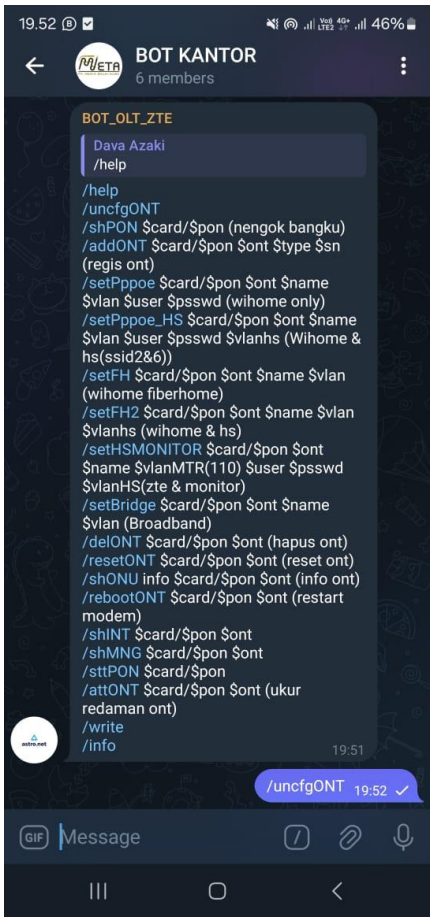
```
/delONT $card/$pon $ont (hapus ont)\n\
/resetONT $card/$pon $ont (reset ont)\n\
/shONU info $card/$pon $ont (info ont)\n\
/rebootONT $card/$pon $ont (restart modem)\n\
/shINT $card/$pon $ont\n\
/shMNG $card/$pon $ont\n\
/sttPON $card/$pon\n\
/attONT $card/$pon $ont (ukur redaman ont)\n\
/write\n\
/info\n\
\n"
   )
```

Fungsi send_welcome1menyediakan daftar lengkap perintah yang tersedia di bot kepada pengguna, setiap perintah disertai dengan deskripsi singkat tentang fungsinya dan format penggunaannya, dan mempermudah pengguna baru atau yang belum familiar dengan bot untuk mengetahui apa yang bisa dilakukan dengan bot tersebut.

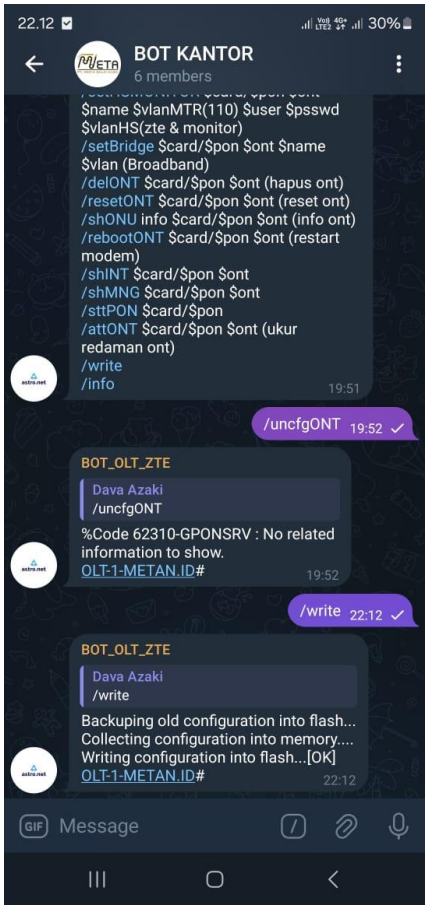
Gambar 4.1 Test script /info



Gambar 4.2 Test script /addONT



Gambar 4.3 Test script /help



Gambar 4.3 Test script /write

BAB V

PENUTUP

5.1. Kesimpulan

Dalam peneltian ini dapat memberikan beberapa poin-poin kesimpulan dalam implementasi enkripsi dan deskripsi berdasarkan alamat IP yaitu sebagai berikut:

1. Proses dimulai dari *BotFather* untuk mendapatkan token akses yang kemudian saya hubungan dengan python untuk menangani beberapa perintah dan menjalankan beberapa perintah. Kode python ini terhubung dengan API telegram untuk mengirim dan membalas pesan.
2. Otomatisasi pada bot telegram yang saya buat ini dapat mengerjakan pekerjaan berulang yang biasanya terjadi pada teknisi di PT.Media Balai Nusa. Ini mengurangi beban kerja manual yang terlalu banyak dan terus menerus, dan dapat membuat pekerjaan Noc untuk lebih fokus kepada pekerjaan yang lebih kompleks. Bot ini juga tidak akan mengeksekusi script yang salah ketika inputan yang diberi user ambigu dan akan tidak merespon jika terjadi dual input yang diberikan pengguna, jadi tidak akan merusak sistem atau konfigurasi pada OLT yang sedang berjalan.
3. Penerapan bot Telegram untuk pengelolaan perangkat jaringan sudah sangat umum digunakan dikalangan penyelenggara besar

seperti PT.Telekomunikasi Indonesia, bot sudah sangat lumrah digunakan oleh tim teknis yang berkerja dilapangan, jadi tidak terlalu merepotkan Noc akan permintaan yang banyak dan berulang, script hanya dijalankan di sebuah server dan dapat diakses oleh semua orang yang memiliki akses khusus.

5.2 Saran

Berdasarkan hasil dari Implementasi yang saya terapkan kali ini penulis dapat memberikan beberapa saran yaitu sebagai berikut:

1. Perlu dilakukan pengembangan untuk menggunakan beberapa platform lain seperti Whatsapp mengingat saat ini aplikasi tersebut sudah menjadi kewajiban di *Smartphone* semua orang.
2. Untuk penelitian selanjutnya dapat meningkatkan keamanan pada komunikasi antara user dan server agar tidak dapat terjadi pencurian akses pada server atau *Man In The Middle*
3. Untuk penelitian selanjutnya untuk dapat melakukan push data secara *real time* jadi tidak ada waktu tunggu untuk server membalas informasi yang diinputkan user, dapat menggunakan fitur SNMP pada router atau switch lalu di tampilan di sebuah website

DAFTAR PUSTAKA

2430-8541-2-PB. (n.d.).

Avila Putri, F., Afrianto, I., Kunci-Komputasi Awan, K., & Komputasi Awan, K. (n.d.). *Tinjauan Literatur : Review Perkembangan Keamanan dari Teknologi Cloud Computing*.

Dang, K. (n.d.). *NETWORK AUTOMATION WITH PYTHON*.

Dharma, J. A., & Rino. (2023). Network Attack Detection Using Intrusion Detection System Utilizing Snort Based on Telegram. *Bit-Tech*, *6*(2), 118–126. https://doi.org/10.32877/bt.v6i2.943

Furqan, M., Sriani, S., & Shidqi, M. N. (2023). Chatbot Telegram Menggunakan Natural Language Processing. *Walisongo Journal of Information Technology*, *5*(1), 15–26. https://doi.org/10.21580/wjit.2023.5.1.14793

Gani, A. G. (n.d.). *SEJARAH dan PERKEMBANGAN INTERNET DI INDONESIA*.

Muliandhi, P., Faradiba, H., Nugroho, B. A., Teknik, J., Semarang, E. U., Pt,), Akses, T., Semarang, W., Hatta, J. S., Kulon, T., Pedurungan, K., Semarang, K., Tengah, J., Singotoro, J., 20, N., & Candisari, K. (2020). Analisa Konfigurasi Jaringan FTTH dengan Perangkat OLT Mini untuk Layanan Indihome di PT. Telkom Akses Witel Semarang. In *Tahun* (Vol. 12, Issue 1).

Nugroho, S., & Pujiarto, B. (2022). *NETWORK AUTOMATION PADA BEBERAPA PERANGKAT ROUTER MENGGUNAKAN PEMROGRAMAN PYTHON*. *9*(1), 79–86. https://doi.org/10.25126/jtiik.202293947

Rahman, S., Sembiring, A., Siregar, D., Khair, H., Prahmana, G., Puspadini, R., & Zen, M. (n.d.). *PYTHON : DASAR DAN PEMROGRAMAN BERORIENTASI OBJEK TAHTA MEDIA GROUP*.

Tsidylo, Ivan, et al. "Designing a Chat Bot for Learning a Subject in a Telegram Messenger." *ICTERI Workshops*. 2020.

Prastowo, Bambang Nurcahyo, Nur Achmad Sulistyo Putro, and Oktaf Agni Dhewa. "PLO User Interface based on Telegram Bot." *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)* 13.1 (2019): 21-30.

Efendi, Mohamad Yusuf. *Implementasi Internet of Things Pada Sistem Kendali Lampu Rumah Menggunakan Telegram Messenger Bot Dan Nodemcu Esp 8266*. Diss. Prodi Teknik Informatika, 2019.

Avisyah, Gisnaya Faridatul, Ivandi Julatha Putra, and Sidiq Syamsul Hidayat. "Open Artificial Intelligence Analysis using ChatGPT Integrated with Telegram Bot." *Jurnal ELTIKOM* 7.1 (2023): 60-66.

Zulherry, Andi, et al. "Optimalisasi Website untuk Monitoring Jaringan OPD di Dinas Kominfo Kota Medan dengan Metode Triangulasi." *Bulletin of Computer Science Research* 3.5 (2023): 357-363.

Ramadhan, Syahrul, Agung Setia Budi, and Mochammad Hannats Hanafi Ichsan. "Rancang Bangun Sistem Auto-Config Sensor Baru pada Perangkat IoT secara Over-The-Air menggunakan Protokol HTTP berbasis Raspberry-Pi." *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* 6.1 (2022): 216-224.

Richardson, Matt, and Shawn Wallace. *Getting Started with Raspberry Pi: Electronic Projects with Python, Scratch, and Linux*. Maker Media, Inc., 2014.