



SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

**TINGKATKAN EFISIENSI VALIDASI DATA *BILL OF
QUANTITY(BOQ)* DENGAN PENERAPAN OTOMASI
*CAPTURE BANNER***

TUGAS AKHIR

MUHAMMAD ALIF LUTHAN

0110220136

PROGRAM STUDI TEKNIK INFORMATIKA

DEPOK

AGUSTUS 2024



**STT TERPADU
NURUL FIKRI**

SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

**TINGKATKAN EFISIENSI VALIDASI DATA *BILL OF
QUANTITY(BOQ)* DENGAN PENERAPAN OTOMASI
*CAPTURE BANNER***

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer

STT - NF
MUHAMMAD ALIF LUTHAN

0110220136

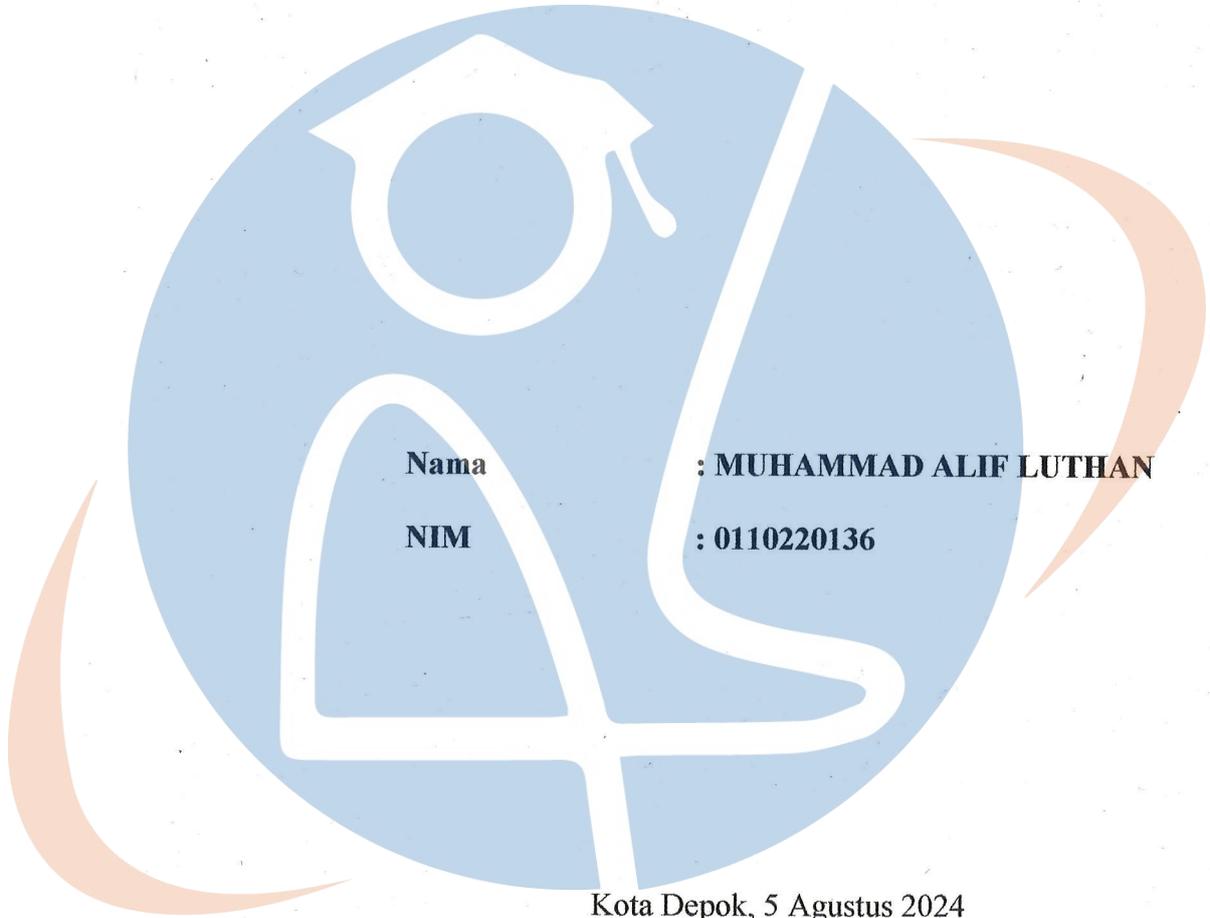
PROGRAM STUDI TEKNIK INFORMATIKA

DEPOK

AGUSTUS 2024

HALAMAN PERNYATAAN ORISINALITAS

Tugas Akhir ini adalah hasil karya penulis, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.



Nama : MUHAMMAD ALIF LUTHAN

NIM : 0110220136

Kota Depok, 5 Agustus 2024

STT - NF Tanda Tangan

A handwritten signature in black ink, appearing to read 'M. Alif Luthan', written over the 'NF' part of the 'STT - NF' text.

Muhammad Alif Luthan

HALAMAN PENGESAHAN

Skripsi/Tugas Akhir ini diajukan oleh :

Nama : MUHAMMAD ALIF LUTHAN

NIM : 0110220136

Program Studi : TEKNIK INFORMATIKA

Judul Skripsi : **TINGKATKAN EFISIENSI VALIDASI DATA *BILL OF QUANTITY*(BOQ)
DENGAN PENERAPAN OTOMASI *CAPTURE BANNER***

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri

DEWAN PENGUJI

Pembimbing



(Ahmad Rio Adriansyah, S.Si., M.Si)

Penguji



(Dr Sirojul Munir, S.Si.,M.Kom)

STT - NF

Ditetapkan di : Kota Depok

Tanggal : 23 Juli 2024

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan skripsi/Tugas Akhir ini. Penulisan skripsi/Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana komputer Program Studi Teknik Informatika pada Sekolah Tinggi Teknologi Terpadu Nurul Fikri Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi penulis untuk menyelesaikan skripsi/tugas akhir ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Allah SWT.
2. Orang tua dan semua anggota keluarga yang telah memberikan dorongan baik secara moral maupun materil dalam penyelesaian tugas ini.
3. Bapak Dr. Lukman Rosyidi selaku Ketua Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
4. Bapak/Ibu Tiffany Nabarian, S. Kom., M.T.I selaku Ketua Program Studi Teknik Informatika Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
5. Bapak/Ibu Nasrul, S.Pd.I, S.Kom, M.Kom, selaku Dosen Pembimbing Akademik yang telah membimbing penulis selama berkuliah di Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
6. Bapak/Ibu Ahmad Rio Adriansyah, S.Si, M.Si selaku Dosen Pembimbing Tugas Akhir penulis dalam menyelesaikan penulisan ilmiah ini.
7. Para Dosen di lingkungan Sekolah Tinggi Teknologi Terpadu Nurul Fikri yang telah membimbing penulis dalam menuntut ilmu yang telah diberikan.
8. Bapak Bayu Hastomo selaku Proyek Manajer serta karyawan yang telah meluangkan waktunya untuk memberikan data yang diperlukan bagi penulisan ilmiah ini.
9. Sahabat dan kerabat pada Angkatan 2020, khususnya Adam Maulana, Rangga Saputra Abbas, dan Ichsan Rustiansyah Yusuf, yang telah memberikan dukungan dan doanya.

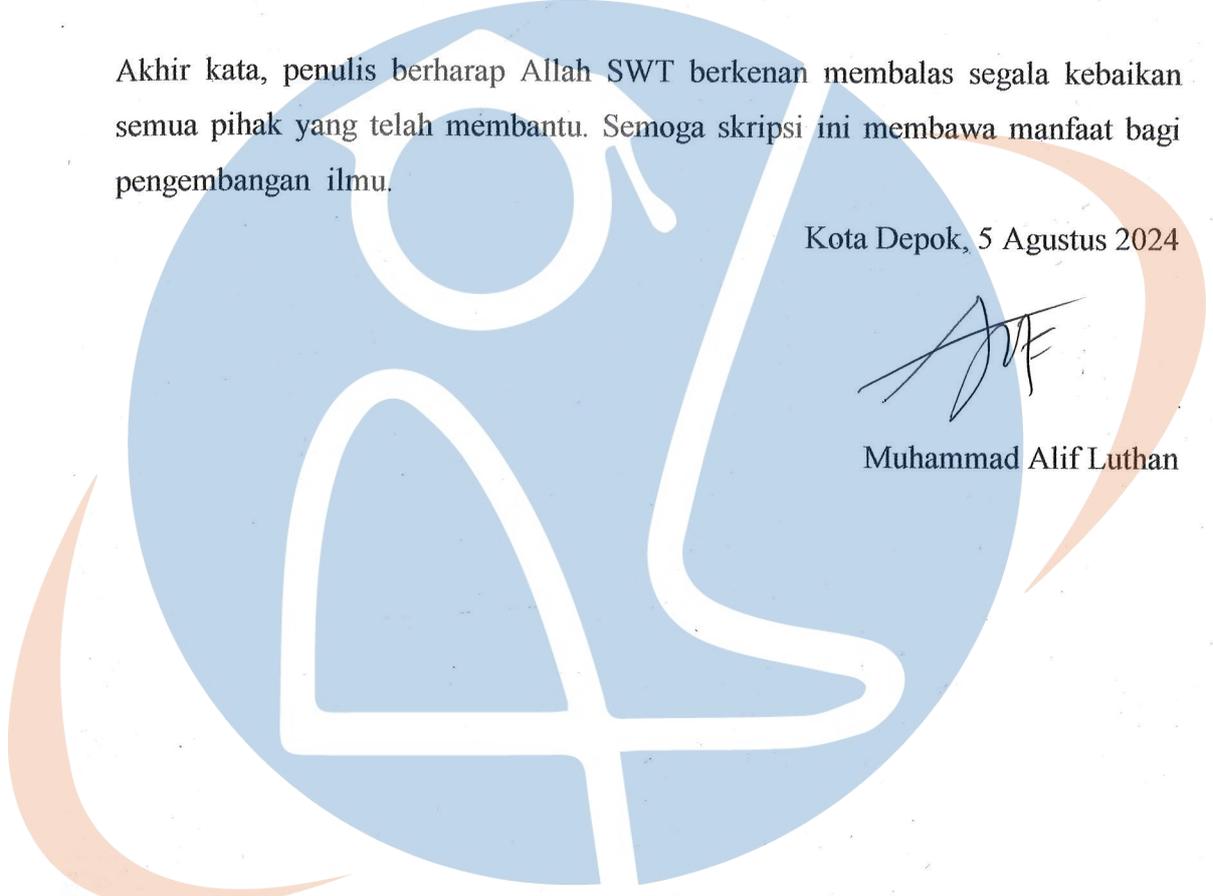
Dalam penulisan ilmiah ini tentu saja masih banyak terdapat kekurangan-kekurangan yang mungkin disebabkan oleh keterbatasan kemampuan dan pengetahuan yang penulis miliki. Walaupun demikian, penulis telah berusaha menyelesaikan penulisan ilmiah ini sebaik mungkin. Oleh karena itu apabila terdapat kekurangan di dalam penulisan ilmiah ini, dengan rendah hati penulis menerima kritik dan saran dari pembaca.

Akhir kata, penulis berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Kota Depok, 5 Agustus 2024



Muhammad Alif Luthan



STT - NF

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Sekolah Tinggi Teknologi Terpadu Nurul Fikri, saya yang bertanda tangan di bawah ini:

Nama : MUHAMMAD ALIF LUTHAN

NIM : 0110220136

Program Studi : Teknik Informatika

Jenis karya : Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada STT-NF **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty - Free Right*)** atas karya ilmiah saya yang berjudul :

Penerapan Otomasi Untuk Tingkatkan Efisiensi Validasi Pada Data Bill Of Quantity (BOQ)

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini STT-NF berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Kota Depok

Pada tanggal : 05 Agustus 2024

STT - NF

Yang Menyatakan



Muhammad Alif Luthan

ABSTRAK

Nama : MUHAMMAD ALIF LUTHAN
NIM : 0110220136
Program Studi : TEKNIK INFORMATIKA
Judul : Tingkatkan Efisiensi Validasi Data *Bill Of Quantity*(BOQ)
Dengan Penerapan Otomasi *Capture Banner*

Pada pengembangan layanan 5G yang dilakukan oleh Indosat, Indosat menggandeng Sisindokom Lintasbuana untuk melakukan tahapan peningkatan layanan 5G dengan penerapan SRv6. Untuk meningkatkan pengembangan layanan 5G dilakukan validasi pada functional router secara repetitif, cukup memakan waktu dan juga pengguna harus teliti, karena apabila ada data yang tidak valid, maka pada saat evaluasi akan memakan waktu ulang validasi. Untuk mengatasi masalah tersebut, Penulis melakukan perancangan skrip otomasi untuk validasi data menggunakan pemrograman Python. Rancangan otomasi *capture banner* secara otomatis yang berfungsi untuk mempermudah pekerjaan yang repetitif oleh Tim Sisindokom, serta mempermudah validasi data dengan dokumen. Penelitian ini merupakan jenis penelitian kualitatif deskriptif, dan tujuan dari penelitian ini adalah untuk menjelaskan implementasi program berdasarkan data yang sudah ada sebelumnya. Pada tahap analisis kebutuhan, dilakukan observasi dan wawancara untuk menggali kebutuhan dari Tim CyberArk selaku pengguna yang akan menggunakan program otomasi *capture banner* demi kelulusan program yang dibuat sesuai dengan kebutuhan. Tahap pengujian yang dilakukan bertujuan untuk menguji program yang telah dikembangkan melalui penerapan skenario uji coba, yang mencakup pengujian efektivitas sistem dan evaluasi hasil. Keberhasilan dari tahap pengujian adalah efektivitas program yang diuji tidak dihitung oleh waktu pekerjaan namun diambil dari keberhasilan pengambilan data *banner* saja. Hasil dari pengujian tersebut, diuji dengan lima kali iterasi dengan 20 Perangkat Cisco dengan rata-rata waktu sekitar 19 detik dalam melakukan validasi secara otomatis. Ini membuktikan bahwa sistem otomasi efektif dalam mengurangi pekerjaan repetitif saat melakukan validasi.

Kata kunci : Cisco, Otomasi, Python, 5G, SRv6

ABSTRACT

Name : MUHAMMAD ALIF LUTHAN
NIM : 0110220136
Study Program : INFORMATICS ENGINEERING
Title : *Improving The Efficiency of Bill Of Quantity(BOQ) Data Validation By Implementing Banner Capture*

In the development of 5G services carried out by Indosat, Indosat cooperates with Sisindokom Lintasbuana to carry out the stages of improving 5G services by implementing SRv6. To improve the development of 5G services, validation is carried out on the functional router repetitively, it is quite time consuming and also the user must be careful, because if there is invalid data, then during the evaluation it will take time to revalidate. To overcome these problems, the author designs automation scripts for data validation using Python programming. Automated banner capture design that functions to simplify repetitive work by the Sisindokom Team, and facilitate data validation with documents. This research is a type of descriptive qualitative research, and the purpose of this research is to explain the implementation of the program based on pre-existing data. In the needs analysis stage, observations and interviews are conducted to explore the needs of the CyberArk Team as users who will use the banner capture automation program for the sake of the smoothness of the program made according to the needs. The testing phase aims to test the program that has been developed through the application of a test scenario, which includes testing the effectiveness of the system and evaluating the results. The success of the testing phase is that the effectiveness of the tested program is not calculated by job time but is taken from the success of retrieving banner data only. The results of the test, tested with five iterations with 20 Cisco devices with an average time of about 19 seconds in validating automatically. This proves that the automation system is effective in reducing repetitive work when performing validation.

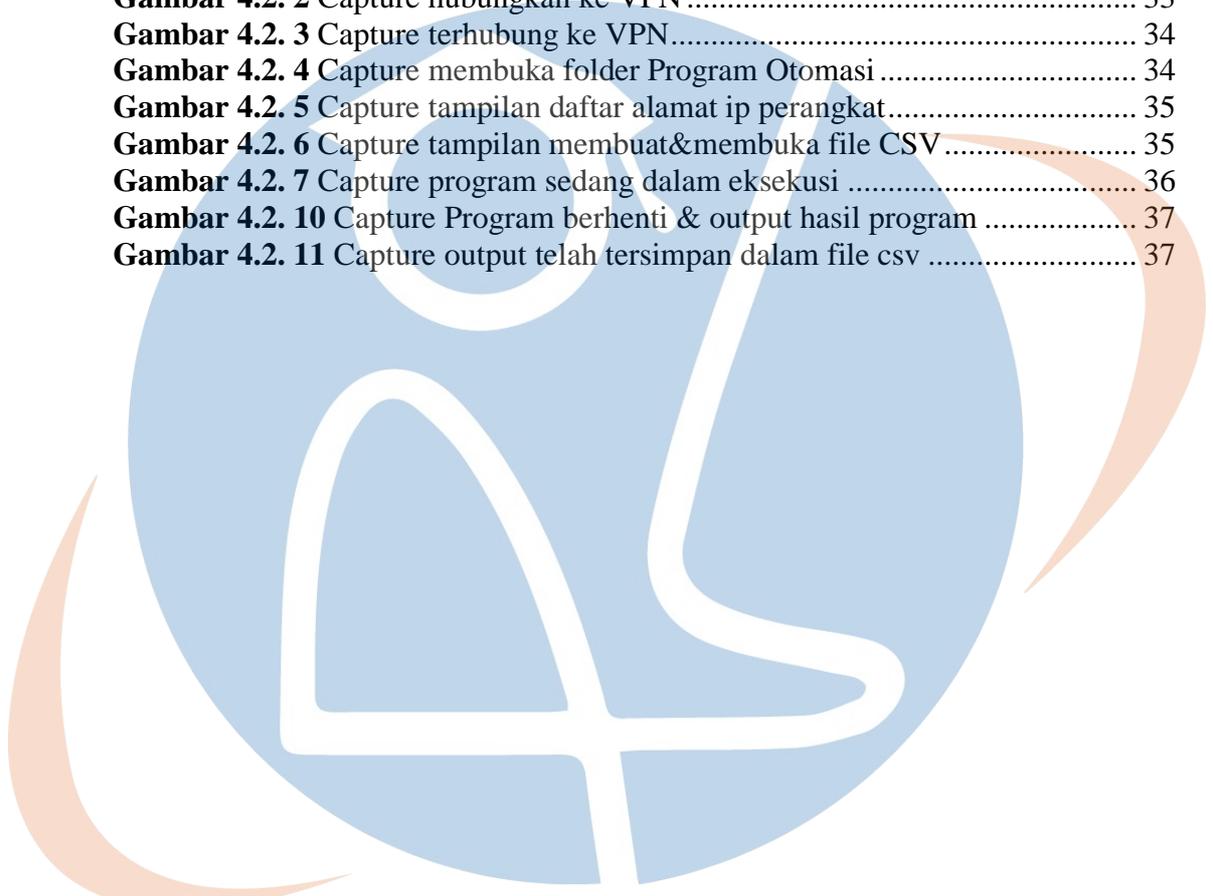
Key words : Cisco , Otomasi , Python , 5G , SRv6

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS	iii
HALAMAN PENGESAHAN	iv
KATA PENGANTAR	v
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI	vii
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS	vii
ABSTRAK	viii
ABSTRACT	ix
DAFTAR ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan dan Manfaat Penelitian.....	4
1.4 Batasan Masalah.....	5
1.5 Sistematika Penulisan.....	5
BAB II KAJIAN LITERATUR	6
2.1 Tinjauan Penelitian.....	6
BAB III METODOLOGI PENELITIAN	19
3.1 Tahapan Penelitian	19
3.2 Jenis Penelitian	21
3.2.1 Metode Analisis Data.....	22
3.2.2 Metode Pengumpulan Data.....	23
3.2.3 Lingkungan Pengembangan	23
3.2.4 Alur Pekerjaan Pengembangan	24
3.2.5 Metode Pengujian.....	25
3.2.6 Metode Implementasi dan Evaluasi	26
BAB IV IMPLEMENTASI DAN EVALUASI	28
4.1 Rancangan Penelitian	28
4.1.1 Flowchart.....	28
4.1.2 Requirement User.....	30
4.1.3 Topology Logical	30
4.2 Implementasi Rancangan Penelitian.....	31
4.2.1 Rencana Pengujian.....	32
4.2.2 Pengujian Sistem Otomasi	32
4.2.3 Hasil Pengujian	38
4.2.4 Analisis Evaluasi Pengujian.....	39
BAB V KESIMPULAN DAN SARAN	41
5.1 Kesimpulan.....	41
5.2 Saran	41
DAFTAR PUSTAKA	43
LAMPIRAN	45

DAFTAR GAMBAR

Gambar 2. 1	Alur Kerja Otomasi Netmiko.....	12
Gambar 3. 1	Tahapan Penelitian.....	19
Gambar 3. 2	<i>Workflow</i> FATP Sisindokom.....	25
Gambar 4. 1	Flowchart	29
Gambar 4. 2	Use Case Diagram	30
Gambar 4. 3	Topologi logika.....	31
Gambar 4.2. 1	Capture tampilan awal vscode	33
Gambar 4.2. 2	Capture hubungkan ke VPN	33
Gambar 4.2. 3	Capture terhubung ke VPN.....	34
Gambar 4.2. 4	Capture membuka folder Program Otomasi	34
Gambar 4.2. 5	Capture tampilan daftar alamat ip perangkat.....	35
Gambar 4.2. 6	Capture tampilan membuat&membuka file CSV	35
Gambar 4.2. 7	Capture program sedang dalam eksekusi	36
Gambar 4.2. 10	Capture Program berhenti & output hasil program	37
Gambar 4.2. 11	Capture output telah tersimpan dalam file csv	37



STT - NF

DAFTAR TABEL

Tabel 2. 1 Penelitian Terkait	16
Tabel 3. 1 Jadwal Penelitian.....	24
Tabel 4. 1 Tabel Iterasi hasil pengujian.....	38
Tabel 4. 2 Tabel Pengujian Efektivitas.....	39



STT - NF

BAB I

PENDAHULUAN

Pada bab ini, akan membahas terkait latar belakang, rumusan masalah, tujuan penelitian, Batasan masalah, serta sistematika penulisan.

1.1 Latar belakang

Pada era yang sangat maju ini, teknologi telekomunikasi sangat penting karena merupakan komponen penting dalam kehidupan masyarakat yang membutuhkan komunikasi yang lancar dan efisien. Tidak hanya masyarakat umum yang membutuhkan layanan telekomunikasi, tetapi juga bisnis yang bekerja dalam berbagai industri seperti infrastruktur, pemerintahan, dan kesehatan, antara lain. Mereka menggunakan telekomunikasi untuk meningkatkan efisiensi dan produktifitas dalam menjalankan berbagai tugas mereka.

Indosat adalah perusahaan penyedia telekomunikasi terdepan di Indonesia yang menjadikan teknologi komunikasi sebagai fondasi utama bisnisnya, menyediakan beragam layanan seperti seluler, telepon kabel, internet, dan lainnya. Sebagai penyedia layanan yang berprestasi, Indosat bertekad untuk mengembangkan serta memajukan jaringan infrastruktur terkini, yaitu teknologi 5G, untuk memberikan koneksi yang lebih cepat, handal, dan penuh inovasi kepada konsumennya di Indonesia.

Teknologi 5G, singkatan dari teknologi seluler generasi kelima dari sistem nirkabel yang dialihkan paket dengan cakupan area yang luas dan throughput yang tinggi. Teknologi ini telah berubah untuk mendukung penggunaan ponsel dengan bandwidth yang sangat tinggi. [1].

Perkembangan infrastruktur jaringan telekomunikasi 5G di Indonesia terus mengalami kemajuan. Sebagai penyelenggara layanan telekomunikasi, Indosat senantiasa berinovasi untuk memperlebar cakupannya guna memenuhi keperluan yang kian rumit. Dalam usaha peningkatan tersebut, fokus Indosat tertuju pada pengembangan jaringan telekomunikasi 5G yang mengadopsi *Segment Routing* berbasis IPv6 (SRv6). SRv6 digunakan sebagai prinsip utama dalam upaya memperkaya efisiensi dan memperlebar cakupan jaringan, untuk pengembangan SRv6 ini, Indosat bekerja sama dengan perusahaan Cisco sebagai vendor pemasok.

Salah satu cara untuk mengembangkan proyek SRv6 yang dilakukan oleh Indosat, Indosat menjalin kerja sama dengan PT Sisindokom Lintasbuana sebagai mitra[2]. Sisindokom bertanggung jawab dalam melaksanakan tugas *Staging Router*, Integrasi Router dan *Acceptance Test Plan (ATP) (Physical Acceptance Test Procedure (PATP) & Functional Acceptance Test Procedure (FATP))* dalam proses pengembangan SRv6 menggunakan perangkat dari Cisco.

Sebelum masuk ke tahapan ATP, proses *Staging* melibatkan Router yang akan dilakukan. *Staging* adalah proses atau lingkungan di mana pengembangan, pengujian, dan evaluasi perangkat jaringan dilakukan sebelum diterapkan secara luas atau diproduksi. *Staging* yang dilakukan oleh pihak Sisindokom, dilakukan pada Gudang yang tersedia dengan perangkat yang ada.

Selanjutnya Setelah tahapan *Staging* Selesai, beralih ke tahapan Integrasi. Integrasi adalah proses pemasangan perangkat ataupun penggantian perangkat untuk mendukung infrastruktur jaringan 5G. Integrasi disini dibagi menjadi 2 tahap yaitu Integrasi Baru dan *Swap* Integrasi. Integrasi baru melakukan pemasangan perdana perangkat Router Cisco sedangkan *Swap* Integrasi melakukan penggantian Router dari Nokia ke Cisco dengan alasan perangkat yang menggunakan vendor Nokia tidak mendukung *Segment Routing* berbasis IPv6 untuk pengembangan infrastruktur jaringan 5G.

Setelah proses *Staging* dan Integrasi sukses dilakukan, masuk ke tahapan ATP(*Acceptance Test Plan*). ATP adalah serangkaian tes yang dirancang untuk memverifikasi apakah sebuah sistem atau produk memenuhi persyaratan spesifikasi dan kinerja yang telah ditentukan. Dalam tahap ATP dibagi menjadi 2 bagian yaitu PATP dan FATP, dalam tahapan PATP melakukan tugas seperti melakukan *Functional Test* terhadap *Physical Device* dan FATP melakukan tugas seperti validasi *functional* Router Metro.

Dalam pengembangan SRv6 pada tahap FATP, didalamnya dibagi menjadi 2 tim yaitu Tim *System Performance Evaluation (SPE)* dan *Functional Acceptance Test Procedure (FATP)*, Penulis termasuk ke dalam tim FATP Sisindokom yaitu *Hotnews(HN)* untuk pengembangan tahap FATP. Dalam ruang lingkup pekerjaannya tim SPE berfokus pembuatan dokumen UAT dan pemeliharaan perangkat metro sedangkan untuk Tim *Hotnews(HN)* validasi *Functional* metro.

Pada tahapan FATP untuk pengembangan SRv6 yang dilakukan oleh Tim *Hotnews*, agar dapat lolos tahapan validasi *functional* router metro untuk tujuan *Submission*. Untuk mencapai hal itu dibutuhkan kelengkapan dokumen validasi seperti *topology*, *log router*, *Capture Device*, dan *CyberArk Change Initial Password (CIP)*. Salah satu syarat untuk *Submission* FATP adalah kelengkapan dokumen validasi yaitu *CyberArk Change Initial Password(CIP)*.

Dalam *Submission* FATP, *Change Initial Password* Dalam *CyberArk* berperan penting karena hasil *capture CyberArk Change Initial Password (CIP)*. digunakan untuk melakukan *Submission*, seluruh vendor yang ingin melakukan migrasi dan integrasi pada *site-site* tertentu, dan CIP digunakan untuk konfigurasi awal, *troubleshoot* dan lain sebagainya.

Untuk melengkapi kelengkapan *Submission* beberapa Kriteria dalam *Request Change Initial Password* dalam platform *CyberArk* itu antara lain, router dapat di akses dan menggunakan *chasis* yang sesuai dengan dokumen *Bill of Quantity(BoQ)* yang ada. Apabila terdapat kegagalan terhadap permintaan *Request CyberArk Change Initial Password (CIP)*, biasanya karena router tersebut tidak dapat diakses, belum adanya *default password* atau *Chasis* pada router tersebut tidak sesuai terhadap dokumen *Bill of Quantity(BoQ)* atau dengan data pada banner router langsung.

Masih dalam tahap pekerjaan *Request Change Initial Password* oleh Tim *CyberArk*, identifikasi dan pengelolaan *chasis* pada router yang dilakukan oleh tim *CyberArk* menjadi sangatlah penting untuk *request* perdana/baru maupun *request* kembali *CyberArk Change Initial Password (CIP)* pada platform *CyberArk*. Selain itu saat mengakses satu per satu router serta membandingkan *chasis* pada setiap banner router *metro* dengan dokumen *BoQ* secara manual. Pekerjaan yang berulang tersebut dapat menjadi tugas yang rumit dan memakan waktu yang lama.

Untuk menyelesaikan masalah ini, pemanfaatan *Network Automation* atau otomatisasi dengan menggunakan modul Python yaitu *Netmiko* dapat memberikan solusi yang efisien. modul *Netmiko* merupakan modul khusus untuk mengotomasi tugas-tugas yang terkait dengan perangkat jaringan melalui *Secure shell(SSH)*. dengan menggunakan modul *netmiko*, kita dapat dengan mudah mengirim perintah

melalui koneksi SSH, mengambil output perintah ,dan melakukan tugas tugas lainnya secara otomatis.

Penelitian ini akan membahas tentang implementasi program otomasi menggunakan netmiko untuk meningkatkan efektivitas waktu validasi serta memudahkan pekerjaan dari Tim CyberArk Sisindokom untuk melakukan pengujian keaktifan router serta mendapatkan data banner dari banyaknya router untuk meningkatkan validasi BoQ dan *Request CyberArk Initial Password*.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, berikut adalah rumusan masalah penelitian:

1. Bagaimana rancangan sistem program otomasi *capture banner* pada setiap metro router untuk mempermudah validasi data BoQ.
2. Bagaimana efektivitas sistem yang dibangun untuk meningkatkan validasi BoQ dan *Request CyberArk Initial Password*.

1.3 Tujuan dan Manfaat Penelitian

Adapun tujuan serta manfaat dari penelitian ini. Adalah :

Tujuan Penelitian

1. Merancang skrip otomasi *capture banner* serta fitur simpan hasil ke dalam *file Comma Separated Values (CSV)* untuk mempermudah tim *CyberArk* melakukan validasi *site* serta perbandingan data BoQ.
2. Mempermudah tim *CyberArk* dalam melakukan *Request CyberArk Initial Password* dengan Otomasi *capture banner* untuk kebutuhan *Submission FATP*.

Manfaat Penelitian

1. Mengefisiensikan waktu & pekerjaan Tim *CyberArk* menggunakan Program otomasi python.
2. Membantu memberikan manfaat bagi perusahaan telekomunikasi khususnya tim *CyberArk* dalam proses validasi data BoQ dengan Otomasi *Capture Banner* pada perangkat jaringan mereka.
3. Meminimalisir kesalahan dalam validasi *site* serta perbandingan data dengan dokumen BoQ terkait.

1.4 Batasan Masalah

Berdasarkan uraian latar belakang di atas, berikut ini merupakan Batasan masalah pada penelitian ini :

1. Penelitian ini hanya berfokus pada data banner yang dikelola.
2. Penelitian ini memakai bahasa pemrograman python dengan modul Netmiko.
3. Ukuran efektivitas tidak dihitung oleh waktu pekerjaan namun diambil dari keberhasilan pengambilan data banner saja.

1.5 Sistematika Penulisan

Laporan tugas akhir ini terdiri dari 5 bab, Antara lain:

1. Bab I Pendahuluan
Menjelaskan mengenai latar belakang masalah, tujuan, manfaat, batasan masalah dan sistematika penulisan.
2. Bab II Kajian Literatur
Menjelaskan mengenai Kajian Literatur penulisan Tugas akhir yang berisikan Definisi definisi, Teori dengan analisis penelitian, Penelitian Terkait yang merujuk untuk membangun Penulisan maupun pengembangan hasil penulisan.
3. Bab III Metodologi Penelitian
Menjelaskan mengenai metode penelitian, parameter penelitian, rincian kerja prosedur penelitian, serta alat dan bahan yang digunakan.
4. Bab IV Implementasi dan Evaluasi
Menjelaskan mengenai Implementasi dan Evaluasi terhadap objek penulisan yaitu program Otomasi. Pada Bab ini berisikan pembahasan Prototipe, Hasil penelitian dan Evaluasi Akhir.
5. Bab V Penutup
Menjelaskan mengenai kesimpulan akhir penelitian dan saran-saran yang direkomendasikan berdasarkan pengalaman di lapangan untuk perbaikan proses pengujian selanjutnya.

BAB II KAJIAN LITERATUR

Bab ini mencakup kajian literatur dan definisi yang digunakan sebagai acuan dan landasan teori untuk tugas akhir ini.

2.1 Tinjauan Penelitian

Terlebih dahulu, beberapa istilah yang digunakan dalam tugas akhir ini akan dijelaskan. Dalam proses pembuatan dan program, istilah-istilah yang muncul diantaranya :

2.1.1 Bill Of Quantity

Merupakan dokumen yang merinci semua item pekerjaan yang harus diselesaikan dalam sebuah proyek, bersama dengan kuantitas masing-masing item. Dokumen *Bill Of Quantity* digunakan untuk menghitung biaya komponen dan proyek untuk memastikan bahwa semua pekerjaan yang diperlukan telah diidentifikasi dan diperhitungkan. Dalam pengembangan layanan 5G, dokumen ini menjadi sangat penting untuk memastikan bahwa setiap aspek pekerjaan telah direncanakan dengan detail dan anggaran yang tepat.

Dokumen *Bill of Quantity* menyimpan informasi yang sangat rinci dan mendetail tentang router yang digunakan dalam proyek. Informasi ini sangat penting untuk proses *submission* FATP, memastikan bahwa semua komponen dan konfigurasi telah tercatat dengan akurat. Data yang disertakan dalam dokumen ini mencakup tipe *chasis, device type, hostname, siteid, dan sitemame*. Dengan menyediakan data-data tersebut, dokumen ini membantu memastikan bahwa setiap elemen proyek telah teridentifikasi dan diperhitungkan dengan baik, sehingga mendukung keberhasilan implementasi dan pengelolaan layanan yang direncanakan.

2.1.2 Capture Banner

Proses *submission* FATP dalam pengembangan layanan 5G memerlukan langkah penting yaitu *request CyberArk Initial Password(CIP)* pada *platform CyberArk*. Salah satu elemen kunci dalam validasi dan permintaan CIP adalah pengumpulan data dari banner router. Proses pengumpulan data ini dikenal sebagai *Capture Banner*.

Capture Banner adalah proses yang bertujuan untuk mengambil dan mencatat berbagai informasi penting yang terdapat pada banner router. Informasi yang dikumpulkan meliputi *hostname*, *device type*, *sitename*, dan *siteid*. Setiap data ini memiliki peranan penting dalam memastikan bahwa Router yang digunakan memenuhi spesifikasi yang telah ditetapkan dalam dokumen *Bill of Quantity* (BoQ).

Proses *Capture Banner* dilakukan secara teliti untuk memastikan bahwa semua data yang diperlukan tersedia dan akurat. *Hostname* memberikan identifikasi unik untuk setiap perangkat, memastikan bahwa tidak ada kebingungan antara satu router dengan yang lain. *Device type* mencakup informasi tentang jenis perangkat yang digunakan, yang penting untuk memastikan kompatibilitas dengan jaringan 5G yang sedang dikembangkan. *Sitename* dan *site ID* memberikan lokasi spesifik dari setiap router, memastikan bahwa setiap perangkat berada di tempat yang tepat dalam jaringan.

Setelah data banner berhasil dikumpulkan, informasi ini digunakan untuk melakukan request CIP dalam *platform CyberArk*. Permintaan ini harus melalui berbagai tahap validasi untuk memastikan bahwa setiap router dapat diakses dengan benar dan memiliki chassis yang sesuai dengan yang tercantum dalam BoQ. Kesalahan dalam proses ini, seperti Router yang tidak dapat diakses atau ketidaksesuaian *chassis*, dapat menyebabkan kegagalan dalam permintaan CIP dan berdampak pada keseluruhan proses *submission FATP*.

Dengan demikian, proses *Capture Banner* dan *request CIP* menjadi komponen krusial dalam *submission FATP* untuk pengembangan layanan 5G. Proses ini memastikan bahwa semua data yang diperlukan telah tersedia dan akurat, mendukung kelancaran implementasi dan pengelolaan layanan 5G yang handal dan efisien. Keberhasilan dalam proses ini akan membantu memastikan bahwa jaringan 5G yang dibangun sesuai dengan spesifikasi dan siap untuk digunakan sesuai dengan tujuan yang telah direncanakan.

2.1.3 Tim Hotnews

Dalam proses tahapan *submission FATP* dalam pengembangan layanan 5G yang dilakukan oleh *Tim Hotnews*, *Tim Hotnews* memegang kewajiban dalam pelaksanaan tahapan *submission FATP* seperti validasi *Functional Router*. Tahapan

validasi *Functional Router* yang dilakukan oleh Tim *Hotnews* meliputi beberapa aktivitas, antara lain:

1. *Monitoring Alarm Router Secara Berkala*: Tim *Hotnews* melakukan pemantauan berkelanjutan terhadap alarm pada router untuk mendeteksi potensi masalah dan mengambil tindakan korektif secepat mungkin. Ini termasuk memantau kondisi jaringan untuk memastikan operasional yang optimal dan meminimalkan gangguan.
2. *Monitoring Registrasi Backup Router Secara Berkala*: Tim juga bertanggung jawab untuk memastikan bahwa registrasi backup pada router dilakukan secara rutin dan akurat. Pemantauan ini memastikan bahwa data cadangan selalu *up-to-date* dan tersedia untuk pemulihan cepat jika terjadi kegagalan sistem.
3. *Monitoring Koneksi Router dengan Node Lain*: Memastikan bahwa setiap router terhubung dengan baik ke node-node router lainnya adalah langkah krusial untuk menjaga integritas dan stabilitas jaringan. Tim *Hotnews* melakukan pemantauan ini secara terus-menerus untuk memastikan bahwa semua node berkomunikasi dengan lancar dan efisien.
4. *Pembuatan Low-Level Design Topology*: Dalam upaya untuk mendukung pelaksanaan tahapan submission FATP, Tim *Hotnews* merancang topologi jaringan secara rinci. *Low-level design topology* ini mencakup detail teknis dari konfigurasi jaringan, memastikan bahwa semua komponen dan koneksi bekerja sesuai dengan rencana.
5. *Capture dan Pengumpulan Log Router*: Tim *Hotnews* juga melakukan *capture* dan pengumpulan log router secara berkala. Proses ini melibatkan pengambilan dan analisis log aktivitas router untuk memastikan bahwa tidak ada anomali atau kesalahan yang dapat mempengaruhi performa jaringan. Pengumpulan log ini penting untuk mendeteksi masalah secara proaktif dan menyediakan data historis yang dapat digunakan untuk troubleshooting dan optimasi jaringan di masa mendatang.

Dengan melakukan langkah-langkah ini, Tim *Hotnews* berusaha untuk memastikan bahwa semua aspek fungsional dari router telah divalidasi dengan benar,

mendukung pengembangan layanan 5G yang andal dan efisien. Validasi yang teliti dan terstruktur ini tidak hanya memastikan bahwa semua perangkat berfungsi sebagaimana mestinya, tetapi juga membantu dalam mengidentifikasi dan mengatasi potensi masalah sebelum mereka berdampak pada operasional jaringan.

2.1.4 Tim CyberArk

Dalam proses tahapan *submission* FATP dalam pengembangan layanan 5G yang dilakukan oleh Tim *Hotnews*, terdapat kolaborasi penting dengan Tim *CyberArk* yang berperan mendukung kelancaran pelaksanaan tahap ini. Tim *CyberArk* bertanggung jawab terhadap validasi data-data yang terkait dengan *submission* FATP, baik data yang terdapat dalam database maupun yang diperlukan untuk *request CyberArk Initial Password(CIP)* pada *platform CyberArk*.

Peran mereka meliputi pengecekan dan verifikasi detail-detail teknis, memastikan bahwa setiap data telah memenuhi standar yang ditetapkan. Dengan keahlian Tim *CyberArk* dalam menangani aspek-aspek kritis ini, mereka memastikan bahwa proses *submission* FATP berjalan dengan akurat dan efisien, mendukung keseluruhan proyek dalam mencapai target yang diinginkan

2.1.5 Python

Menurut Rosmala *et al*, Python adalah bahasa pemrograman tingkat tinggi yang memiliki beberapa karakteristik utama, seperti Python bersifat *interpreter*, yang berarti kode program dieksekusi baris per baris, tanpa memerlukan proses kompilasi sebelumnya. Selain itu, Python mendukung mode interaktif, di mana kode dapat dieksekusi secara langsung dalam sesi interaktif atau dalam lingkungan *shell*. Python adalah bahasa pemrograman yang merujuk pada objek, yang berarti ia mendukung paradigma pemrograman berbasis objek. Ini memungkinkan pengembang untuk membuat dan menggunakan objek dalam pemrograman, yang memungkinkan pengelolaan kode yang lebih terstruktur dan modular. Meskipun Python adalah bahasa yang diinterpretasikan, *source code* Python akan dikompilasi menjadi format *bytecode* sebelum dieksekusi. Ini memungkinkan untuk eksekusi kode yang lebih cepat dibandingkan dengan interpretasi langsung dari *source code*, meskipun masih lebih lambat daripada bahasa pemrograman *low-level*. [3].

Salah satu keunggulan utama Python adalah portabilitasnya, yang berarti bahwa kode Python dapat dijalankan di berbagai platform sistem operasi, termasuk PC, *Macintosh*, dan *UNIX*. Hal ini membuat Python menjadi pilihan yang populer bagi pengembang perangkat lunak yang mencari fleksibilitas dan kemudahan dalam mengembangkan aplikasi yang dapat berjalan di berbagai lingkungan. [4]. Keunggulan Python tidak hanya terletak pada portabilitasnya, tetapi juga pada sintaksisnya yang jelas dan mudah dipahami. Bahasa ini menggunakan pendekatan yang tidak jauh dengan bahasa manusia daripada bahasa pemrograman lainnya, sehingga memudahkan pengembang dalam menulis, membaca, dan memahami kode program. Selain itu, Python memiliki berbagai fitur dan modul bawaan yang efisien dan siap digunakan, memungkinkan pengembang untuk mempercepat proses pengembangan dan mengimplementasikan fungsionalitas yang kompleks dengan mudah. Victor Saputra Ginting *et al* menyatakan beberapa keuntungan bahasa pemrograman python dalam jurnalnya [4], antara lain:

Pengembangan program dilakukan cepat dan coding yang lebih sedikit.

1. Pengembangan program dilakukan lebih efisien dari segi waktu dan jumlah coding yang lebih sedikit, Python terkenal karena memungkinkan pengembangan program dengan cepat. Hal ini disebabkan oleh sintaksis yang mudah dipahami dan minimnya kode yang diperlukan untuk mencapai tujuan tertentu. Python menyediakan banyak fitur bawaan dan pustaka yang dapat digunakan langsung, mengurangi kebutuhan untuk menulis kode dari awal. Ini memungkinkan pengembang untuk fokus pada logika program daripada menangani detail teknis.
2. Mendukung berbagai platform: Python berjalan diberbagai platform seperti *Windows*, *macOS*, *Linux*, dan sistem operasi lainnya. Ini berarti kode yang ditulis dalam Python dapat berjalan di berbagai lingkungan operasional tanpa mengalami perubahan yang signifikan, yang memudahkan pengembang untuk membuat aplikasi yang berfungsi di berbagai platform.
3. Python mendukung paradigma *Object Oriented Programming* (OOP): Python adalah bahasa pemrograman yang berbasis objek, yang berarti ia mendukung paradigma ini. Dalam OOP, program dibangun dengan membuat objek yang memiliki atribut dan metode yang terkait dengan objek

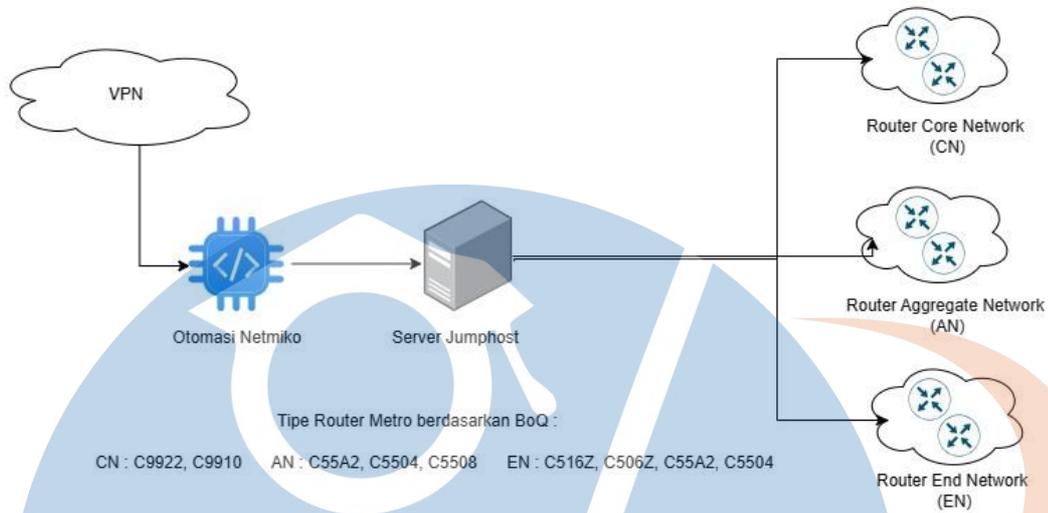
tersebut. Metode ini membantu memecah masalah yang sukar menjadi bagian yang lebih spesifik dan dikelola dengan baik.

4. Bahasa pemrograman Python sangat efisien, Meskipun Python biasanya dianggap sebagai suatu bahasa program yang mudah dipahami dan digunakan, ia juga sangat efisien. Python memiliki interpreter yang efisien dan pustaka bawaan yang dioptimalkan, sehingga mampu menangani tugas tugas komputasi dengan baik, terutama dalam konteks pengolahan data dan kecerdasan buatan.
5. Program yang berorientasi terhadap objek yang lebih sederhana namun cukup efektif, Python memungkinkan pengembang untuk menulis program yang berorientasi terhadap objek dengan cara yang lebih sederhana dan elegan. Dengan sintaks yang mudah dipahami dan dukungan penuh untuk OOP, Python memungkinkan pembuatan kode yang bersih, terstruktur, dan mudah dipelihara.
6. Serta dapat digabungkan bersama bahasa pemrograman yang lain: Python dapat digunakan bersama dengan bahasa pemrograman lain dalam satu aplikasi. Ini memungkinkan pengembang untuk memanfaatkan kekuatan dan kelebihan berbagai bahasa pemrograman untuk menyelesaikan tugas tugas yang kompleks. Python memiliki kemampuan untuk berinteraksi dengan banyak bahasa pemrograman lain melalui pustaka dan modul yang tersedia.

2.1.6 Netmiko

Netmiko adalah *library* Python yang didasarkan pada *library* Paramiko, yang mendukung koneksi SSH di berbagai vendor perangkat jaringan (multivendor). Berbeda dengan Paramiko, Netmiko memiliki fitur yang lebih mudah untuk terhubung ke dalam perangkat jaringan, yaitu dengan menggunakan fitur *Connect Handler*. Fitur ini membuat Netmiko dalam membangun koneksi dengan perangkat jaringan menjadi lebih mudah. Netmiko memerlukan beberapa informasi agar dapat masuk ke dalam perangkat jaringan, seperti alamat IP (*Internet Protocol*), tipe perangkat, Username, dan Password. Tipe perangkat jaringan diperlukan karena Netmiko tidak dapat mendeteksi secara otomatis suatu tipe perangkat jaringan[5]. Adapun gambaran terkait cara kerja netmiko pada system

jumphost server diperlihatkan pada Gambar 2.1 Alur kerja otomatisasi netmiko sebagai berikut ini :



Gambar 2. 1 Alur Kerja Otomasi Netmiko

Netmiko adalah sebuah *library* yang memungkinkan para pengembang untuk mengotomatisasi dan mengelola perangkat jaringan melalui protokol *Secure Shell* (SSH) dengan menggunakan bahasa pemrograman Python. Meskipun Netmiko dibangun di atas *library* Paramiko untuk menyederhanakan manajemen SSH, namun tetap ada beberapa *interface* dan metode terbaru yang diperlukan untuk bisa dipakai pada berbagai perangkat jaringan dari vendor yang berbeda. Dengan menggunakan Netmiko, para pengembang dapat mengakses dan mengelola perangkat jaringan dari berbagai vendor, seperti Arista, Cisco, HP, Juniper, dan lainnya, tanpa harus memiliki pengetahuan mendalam tentang konfigurasi khusus untuk masing-masing perangkat tersebut. Beberapa perangkat yang didukung oleh Netmiko termasuk Arista vEOS, Cisco ASA, Cisco IOS, Cisco IOS-XE, Cisco IOS-XR, Cisco NR-OS, Cisco SG300, HP Comware 7, HP ProCurve, Juniper Junos, Linux, dan banyak lagi. [6].

Dengan adanya dukungan untuk berbagai vendor perangkat jaringan, Netmiko menjadi salah satu pilihan utama bagi para pengembang yang ingin mengotomatisasi tugas tugas administratif dan konfigurasi pada jaringan mereka. Melalui Netmiko, mereka dapat dengan mudah mengembangkan skrip Python yang dapat berinteraksi dengan perangkat jaringan dari berbagai vendor tanpa harus

mempelajari protokol dan konfigurasi spesifik dari masing-masing perangkat. Ini memungkinkan pengembang untuk lebih fokus pada pengembangan solusi yang inovatif dan efektif dalam mengelola jaringan mereka.

Boumezrag Abdelhak juga meringkas tujuan dari *library* ini dalam jurnalnya dalam 4 pernyataan[6]. yaitu sebagai berikut:

- Berhasil melakukan pembuatan koneksi *Secure Shell* (SSH) ke perangkat jaringan.
- Menyederhanakan penerapan perintah *show* dan pengambilan data *output* pada perangkat jaringan.
- Menyederhanakan eksekusi perintah konfigurasi, termasuk kemungkinan melakukan tindakan eksekusi.
- Lakukan hal di atas pada berbagai vendor dan platform jaringan.

Luis Geraldo Mauboy dan Theophilus menjelaskan dalam jurnalnya menuturkan metode atau fungsi dalam *library* Netmiko[7], Berikut adalah penjelasan mengenai beberapa metode atau fungsi yang disediakan oleh *library* Netmiko:

- Metode *send*: Metode ini berguna untuk mengirim perintah ke perangkat jaringan. Ketika Anda terhubung ke perangkat menggunakan objek *net_connect* (hasil dari fungsi *ConnectHandler*), Anda dapat menggunakan metode *send* untuk mengirimkan perintah-perintah tertentu ke perangkat tersebut. Contoh penggunaan: `net_connect.send_command('show interfaces')`.
- Metode *enable*: Metode ini berguna untuk masuk ke mode *enable* pada perangkat jaringan yang memerlukan autentikasi tambahan untuk mengakses beberapa perintah atau konfigurasi yang lebih sensitif. Ketika Anda terhubung ke perangkat menggunakan objek *net_connect*, Anda dapat menggunakan metode *enable* untuk beralih ke mode *enable*. Contoh penggunaan: `net_connect.enable()`.
- Metode *disconnect*: Metode ini berguna untuk menutup koneksi ke perangkat jaringan. Setelah selesai melakukan interaksi dengan perangkat, penting untuk menutup koneksi dengan benar untuk membebaskan sumber

daya dan memastikan keamanan. Metode *disconnect* digunakan untuk tujuan ini. Contoh penggunaan: *net_connect.disconnect()*.

Dengan menggunakan metode-metode tersebut, pengguna dapat dengan mudah mengirim perintah-perintah, masuk ke mode *enable*, dan menutup koneksi ke perangkat jaringan dengan mudah dan efisien menggunakan *library* Netmiko. Ini memungkinkan pengguna untuk mengotomatiskan tugas-tugas administrasi jaringan dan mempermudah manajemen perangkat jaringan dalam infrastruktur mereka.

2.1.7 SecureCRT

SecureCRT adalah perangkat lunak emulasi terminal yang kokoh dari VanDyke Software untuk meningkatkan produktivitas dengan manajemen sesi tingkat lanjut dan menghemat waktu dengan merampingkan tugas berulang. *SecureCRT* menyediakan akses jarak jauh yang aman, *transfer file*, dan *tunneling data*. Fungsi *SecureCRT* untuk klien Telnet berbasis GUI dan emulator terminal menggunakan keamanan *Secure Shell* (SSH) [8].

SecureCRT ini juga memberikan fasilitas kepada penulis untuk melakukan Observasi secara langsung untuk memeriksa dan menganalisa *Site/Router* yang sedang divalidasi untuk kebutuhan *Request Change Initial Password* dalam kebutuhan *Submission FATP*. Selain itu juga keuntungan dari kemudahan penggunaan *SecureCRT* itu sendiri adalah adanya *bookmark button* yang dapat di custom untuk melaksanakan perintah eksekusi.

2.1.8 Visual Studio Code

Visual Studio Code (VScode) adalah sebuah editor teks yang dikembangkan oleh *Microsoft*, dirancang untuk memberikan pengalaman pengkodean yang kuat dan efisien bagi para pengembang perangkat lunak. Dengan fokus pada kinerja yang ringan dan dukungan untuk berbagai bahasa pemrograman, VScode telah menjadi salah satu editor teks yang paling populer di kalangan pengembang. Salah satu keunggulan utama dari VScode adalah kemampuannya untuk berjalan dengan lancar dan efisien pada berbagai sistem operasi, termasuk *Linux*, *macOS*, dan *Windows*. Meskipun memiliki fitur yang kaya, VScode tetap ringan dalam penggunaan sumber daya, membuatnya cocok untuk digunakan pada berbagai jenis

perangkat dan lingkungan pengembangan. VScode mendukung berbagai bahasa pemrograman utama, seperti *JavaScript*, *TypeScript*, *Python*, *Java*, *C#*, dan banyak lagi. Ini memungkinkan para pengembang untuk bekerja dengan berbagai proyek dan teknologi tanpa perlu beralih antara editor teks yang berbeda. Selain itu, melalui penggunaan plugin atau ekstensi yang tersedia di marketplace, dukungan untuk bahasa pemrograman lainnya juga dapat ditambahkan sesuai kebutuhan. [8].

Seiring dengan keluarnya versi-versi baru Visual Studio Code, fitur-fitur canggih VScode terus diperbarui dan ditambahkan, seperti *Intellisense* (memberi saran kode), integrasi *Git*, *debugging*, dan ekstensi yang meningkatkan kemampuan editor teks. [8]. Visual Studio Code menghadirkan pembaruan versi secara berkala setiap bulan. Hal ini memastikan bahwa pengguna selalu mendapatkan fitur terbaru dan perbaikan bug yang diperlukan. Editor kode Visual Studio Code adalah proyek sumber terbuka, yang berarti kode sumbernya dapat diakses dan dilihat oleh siapa pun. Pengguna juga dapat berkontribusi pada pengembangan kode sumber melalui *GitHub*. Ini memungkinkan komunitas untuk ikut serta dalam pengembangan dan perbaikan Visual Studio Code, menjadikannya pilihan yang disukai oleh banyak pengembang aplikasi. [9].

Secara keseluruhan, *Visual Studio Code* adalah editor teks yang sangat populer dan kuat, yang menawarkan kombinasi yang luar biasa antara kinerja ringan, dukungan bahasa pemrograman yang luas, fitur canggih, pembaruan berkala, dan sifat sumber terbuka. Ini membuatnya menjadi pilihan utama bagi pengembang perangkat lunak di seluruh dunia untuk meningkatkan produktivitas dan efisiensi dalam proses pengembangan perangkat lunak.

2.1.9 Black box

Metode pengujian *Black box* menguji perangkat lunak dengan fokus pada spesifikasi fungsional tanpa memperhatikan logika atau struktur kode program. Dalam metode ini, pengujian menguji program dengan memasukkan berbagai kondisi input yang telah ditentukan sebelumnya, untuk mengevaluasi apakah program tersebut berperilaku sesuai dengan spesifikasi fungsionalnya dan memenuhi kebutuhan pengguna. [10]. Proses *Black box-testing* dilakukan dengan cara menguji program menggunakan data *sitelist* atau list perangkat yang telah disiapkan sebelumnya. Pengujian ini penting untuk memastikan bahwa program sesuai

dengan desainnya dan mampu memenuhi permintaan pengguna. Penguji tidak perlu mengetahui atau memahami bagaimana program tersebut diimplementasikan secara internal yang mereka perhatikan adalah input dan output dari program.

Metode pengujian *Black box* dipilih karena memiliki beberapa kelebihan dari perspektif pengguna. Pertama, metode ini memungkinkan pengembang untuk dengan mudah mengidentifikasi fungsi yang tidak lengkap atau tidak berjalan dengan benar, karena fokus pada evaluasi spesifikasi fungsional yang dibutuhkan pengguna. Kedua, pengujian *Black box* memudahkan partisipasi pengguna yang tidak memiliki latar belakang dalam *programming* untuk ikut berkontribusi dalam peningkatan program. Berlawanan dengan *white box*, dimana penguji harus memahami kode program secara internal untuk dapat melakukan pengujian. [11]. Dengan demikian, metode pengujian *Black box* adalah pendekatan yang efektif dalam memastikan bahwa program berperilaku sesuai dengan harapan pengguna dan memenuhi spesifikasi fungsionalnya. Dengan fokus pada kebutuhan pengguna dan memberikan kemungkinan partisipasi dari berbagai pihak, pengujian *Black box* menjadi salah satu pendekatan yang populer dalam industri pengujian perangkat lunak[11].

1. Penelitian Terkait

Tabel 2. 1 Penelitian Terkait

No	Nama dan Tahun	Judul	Topik	Subjek	Hasil
1	Muchamad Mashudi,2022	Network Automation Menggunakan Bahasa Pemograman Python	<i>Otomasi Jaringan</i>	Mahasiswa	Telah berhasil mengembangkan program skrip otomasi dengan <i>library</i> Netmiko
2	Wiji Prabowo,2021	Rancang Bangun Otomasi Jaringan Komputer Dengan Python	<i>Otomasi Jaringan</i>	Mahasiswa	Menambah informasi untuk pengembangan skrip program berbasis Python.

3	Luis Geraldo Mauboy, Theophilus Wellem, 2022	Studi Perbandingan Library Untuk Implementasi Network Automation Menggunakan Paramiko Dan Netmiko Pada Router MikroTik	Otomasi Jaringan	Mahasiswa	Menambah Informasi untuk pengembangan skrip program dengan 2 modul yang berbeda.
---	--	--	------------------	-----------	--

Adapun penjelasan dari Tabel 2.1 Penelitian terkait pada tahapan yang dilakukan dalam penelitian sebagai berikut :

1. Pada penelitian pertama dalam penelitian terkait adalah Network Automation menggunakan Bahasa Pemrograman Python dapat memudahkan penyedia internet dalam hal efisiensi waktu, biaya, dan tenaga. Penelitian terkait menggunakan GNS3 sebagai contoh tempat uji coba sesuai dengan keadaan peneliti yang langsung menggunakan system vendor terkait. Untuk memudahkan otomasi penulis menggunakan 2 *library* yang berbeda karena tergantung pada penggunaan perangkat vendor terkait, penulis menuturkan apabila menggunakan satu vendor saja maka direkomendasikan menggunakan *library* Paramiko apabila lebih dari satu vendor disarankan untuk menggunakan *library* Netmiko, sebab dalam *library* Netmiko mampu menuliskan vendor apakah yang dipakai dengan penulisan skrip tertentu.
2. Pada penelitian kedua dalam penelitian terkait adalah Program otomasi yang dibangun penulis dapat mendapatkan hasil positif yaitu membantu mempermudah pekerjaan *Administrator* dalam melakukan konfigurasi selain itu otomasi yang dilakukan dengan waktu yang pesat dan dapat meminimalisir kesalahan dalam melakukan konfigurasi pada router Mikrotik. Selain manfaat efisiensi waktu yang didapat, program otomasi juga dapat melakukan *backup* pada konfigurasi yang telah diimplementasi pada Mikrotik dalam wujud *file*.
3. Pada penelitian ketiga dalam penelitian terkait adalah implementasi otomasi yang dilakukan terhadap 2 *library* Python yaitu Netmiko dan Paramiko memiliki kelebihan dan kekurangannya masing masing. Pada saat

melakukan otomasi menggunakan Paramiko, Didapatkan bahwa Paramiko menunjukkan hasil yang lebih baik dari sisi waktu eksekusi skrip. Namun dalam penggunaannya, pengembangan dan kemudahan penulisan skrip, Netmiko lebih mudah dibandingkan Paramiko karena penamaan fungsi yang lebih mudah dipahami , *Library* Netmiko sendiri juga mendukung berbagai macam vendor perangkat jaringan dan platform.



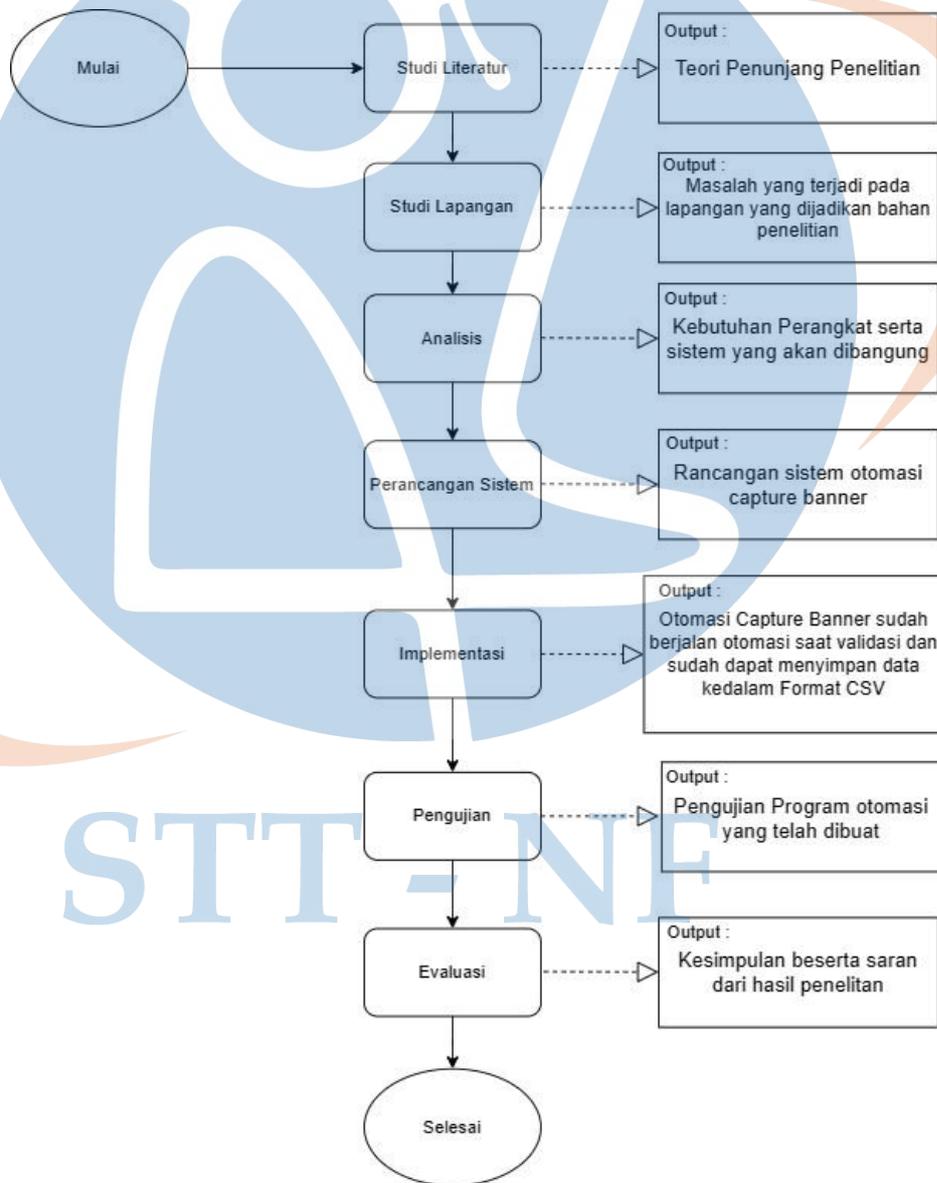
STT - NF

BAB III METODOLOGI PENELITIAN

Pada tahap ini, terdapat beberapa alur yang menjelaskan secara rinci setiap proses yang telah disusun untuk memungkinkan pelaksanaan penelitian yang jelas dan terorganisir.

3.1 Tahapan Penelitian

Tahapan penelitian secara umum dibagi sebagai berikut:



Gambar 3. 1 Tahapan Penelitian

Adapun penjelasan Gambar 3.1 Tahapan penelitian yang dilakukan dalam penelitian ini, serta penjelasan lebih lanjut mengenai tahapan-tahapan tersebut berdasarkan Gambar 3.1 adalah sebagai berikut :

1. Studi Literatur:

Tahap pertama dalam penelitian ini adalah studi literatur, dimana peneliti menggunakan referensi dan teori-teori yang relevan dari berbagai sumber untuk mendukung penelitian ini. Studi literatur membantu peneliti dalam memahami konteks masalah yang akan diteliti dan kerangka teoritis yang relevan.

2. Studi Lapangan:

Peneliti melakukan observasi dan wawancara langsung dengan orang-orang di lapangan. Tujuannya adalah guna mengetahui masalah yang dihadapi dan kebutuhan pengguna dengan lebih baik. Hasil dari proses ini adalah perancangan program yang didasarkan pada informasi yang dikumpulkan di lapangan. Interaksi langsung dengan pemangku kepentingan memungkinkan peneliti untuk memperoleh wawasan yang lebih mendalam tentang konteks praktis dari masalah yang dihadapi.

3. Analisis Kebutuhan:

Menganalisis sistem dan pekerjaan yang saat ini digunakan. Mereka menemukan bahwa sistem dan pekerjaan saat ini tidak efektif karena menggunakan proses manual yang tidak efisien. Analisis ini digunakan sebagai dasar untuk merumuskan persyaratan yang harus terpenuhi oleh sistem yang akan dibangun. Pemahaman wajib mendalam tentang kebutuhan pengguna memungkinkan penulis untuk merancang solusi yang tepat dan relevan.

4. Perancangan Sistem:

Di sini, melanjutkan hasil analisis kebutuhan untuk merancang sistem yang akan dibuat. Hasil dari analisis ini adalah *topology* logika sistem dan fitur yang akan dibuat; perancangan ini memiliki tujuan agar sistem yang dibangun sudah dapat dipastikan dapat mengatasi masalah saat ini dan memenuhi kebutuhan pengguna.

5. Implementasi:

Tahap di mana program otomasi yang telah dirancang diimplementasikan. Penulis menerapkan program otomasi yang telah dirancang, yang menghasilkan pembuatan *file* otomatis dalam format CSV. Tujuan implementasi ini adalah untuk menyimpan data banner untuk setiap perangkat secara otomatis.

6. Pengujian:

Tahap ini bertujuan untuk menguji efektivitas program otomasi yang telah dirancang. Pengujian dilakukan dengan melakukan beberapa iterasi pada sejumlah perangkat yang telah ditentukan oleh pengguna. Tujuannya adalah untuk memastikan bahwa program beroperasi dengan baik dan memenuhi ekspektasi.

7. Evaluasi:

Pada tahap evaluasi, program otomasi yang telah digunakan oleh peneliti dievaluasi kembali. Evaluasi ini melihat standar dan fitur program otomasi yang sudah berjalan, serta apakah program tersebut memenuhi kebutuhan dan harapan pengguna.

3.2 Jenis Penelitian

Penelitian ini bersifat deskriptif, yang bertujuan untuk menjelaskan perancangan dan implementasi program berdasarkan masalah dan data yang telah ada sebelumnya. Dalam penelitian deskriptif, fokus utama adalah pada pemahaman mendalam terhadap situasi atau fenomena yang diamati, serta menjelaskan fakta-fakta yang terkait dengan masalah yang diteliti. Dalam konteks ini, informasi aktual terkait masalah dan data diperoleh melalui wawancara dengan Tim *CyberArk* serta observasi langsung terhadap sistem yang akan diimplementasikan programnya. Metode ini memungkinkan peneliti untuk memahami masalah yang akan diselesaikan oleh program yang akan dirancang dan dilaksanakan.

Menurut Kutha dalam jurnal[12], metode penelitian deskriptif bukan hanya sekedar menggambarkan fakta-fakta yang diamati, tetapi juga memberikan pemahaman dan penjelasan yang memadai terhadap fenomena yang diteliti. Dengan demikian, penelitian deskriptif tidak hanya mengidentifikasi masalah atau keadaan yang ada, tetapi juga mencoba untuk menguraikan alasan di balik masalah

tersebut dan memberikan interpretasi yang mendalam. Pendekatan deskriptif akan membantu peneliti untuk dalam memahami permasalahan yang dihadapi oleh Tim *CyberArk* dan sistem yang mereka tangani secara mendalam. Selain itu, dengan menjelaskan fakta-fakta yang ada dan memberikan analisis yang memadai, penelitian ini akan memberikan kontribusi yang berharga dalam memecahkan masalah yang dihadapi dan merancang solusi yang efektif.

3.2.1 Metode Analisis Data

Studi ini menggunakan metode analisis data kualitatif. Seperti yang dinyatakan oleh Sukmadinata dalam jurnal[13], penelitian kualitatif adalah pendekatan penelitian yang bertujuan untuk mendefinisikan dan menganalisis berbagai elemen subjektif, seperti fakta, peristiwa, kegiatan sosial, sikap, kepercayaan, tanggapan, dan pemikiran individu dan kelompok.

Metode analisis data kualitatif memberikan penekanan pada pemahaman dalam konteks dan makna dari data yang diperoleh. Berbeda dengan metode analisis data kuantitatif yang menggunakan angka dan statistik untuk mengukur fenomena, metode kualitatif lebih berfokus pada interpretasi, deskripsi, dan pemahaman mendalam terhadap fenomena yang diamati.

Penggunaan teknik analisis kualitatif dibarengi dengan metode wawancara mendalam pada penelitian ini, dapat membantu penulis dalam mengumpulkan informasi dan permasalahan yang ada dalam pekerjaan Tim *CyberArk* yang repetitif. Selain mengumpulkan informasi dan permasalahan yang terjadi, metode wawancara yang mendalam ini dapat membantu penulis dalam memahami kebutuhan pengguna dan mengidentifikasi masalah yang mungkin terjadi dalam pengembangan program. Data yang diperoleh dari kegiatan wawancara dapat digunakan untuk memperbaiki atau mengoptimalkan perancangan dari program otomasi.

Adapun kebutuhan dari sistem otomasi *capture banner* adalah sebagai berikut:

1. Tim *CyberArk* berperan sebagai pengguna yang hanya bertugas untuk memastikan data *banner* dapat tersimpan ke dalam *file csv*.
2. Tim *CyberArk* menentukan daftar *device/Site* yang ingin dilakukan validasi dengan penerapan otomasi *capture banner*.

3.2.2 Metode Pengumpulan Data

Berikut ini adalah tahapan penulis dalam melakukan proses pengumpulan data:

1. Wawancara

Tahapan pertama adalah melakukan wawancara dengan Tim CyberArk. Wawancara dilakukan untuk mendapatkan masukan langsung dari mereka mengenai masalah yang sering mereka alami dalam pekerjaan mereka. Dengan berkomunikasi langsung, penulis dapat memahami dengan lebih baik tantangan dan kebutuhan yang dihadapi oleh Tim CyberArk.

2. Observasi

Tahap observasi dilakukan dengan cara langsung mengamati akses pada router yang akan diamati. Melalui observasi ini, peneliti mendapatkan informasi mengenai kondisi dan masalah yang terjadi secara langsung. Observasi langsung memberikan kesempatan untuk mengidentifikasi masalah yang mungkin tidak terungkap melalui wawancara saja, serta untuk mengumpulkan data yang lebih akurat dan mendalam.

3. Studi Pustaka

Pengumpulan data juga dilakukan melalui studi pustaka. Penulis mengacu pada jurnal penelitian dan buku-buku literatur yang relevan atau sesuai dengan kebutuhan penelitian ini. Melalui studi pustaka, penulis dapat memperoleh informasi dan pengetahuan yang telah dipublikasikan sebelumnya mengenai topik penelitian ini. Hal ini membantu penulis dalam memperluas pemahaman mereka tentang konteks, teori, dan konsep yang terkait dengan masalah yang diteliti.

3.2.3 Lingkungan Pengembangan

Adapun Modul/Laptop/Lokasi yang digunakan oleh penulis dalam melakukan penelitian yang berlangsung :

1. Lokasi Penelitian

PT. Sisindokom Lintasbuana adalah tempat penelitian ini dilakukan. *Address*: Jl. Pintu Air Raya No.2-F, RT.7/RW.1, Ps. Baru, Kecamatan Sawah Besar, Kota Jakarta Pusat, Daerah Khusus Ibukota Jakarta 10710..

2. Bahan & Alat

Adapun *Hardware* yang penulis gunakan untuk merancang program tersebut, adalah laptop dengan spesifikasi *Intel(R) Core(TM) i5-12500H RAM 16GB*.

Selain itu alat serta modul pengembangan yang digunakan penulis, adalah sebagai berikut:

1. OS Windows 11
2. Visual Studio Code
3. Modul Netmiko
4. VPN Cisco *Anyconnect secure mobility client*

3. Jadwal Penelitian

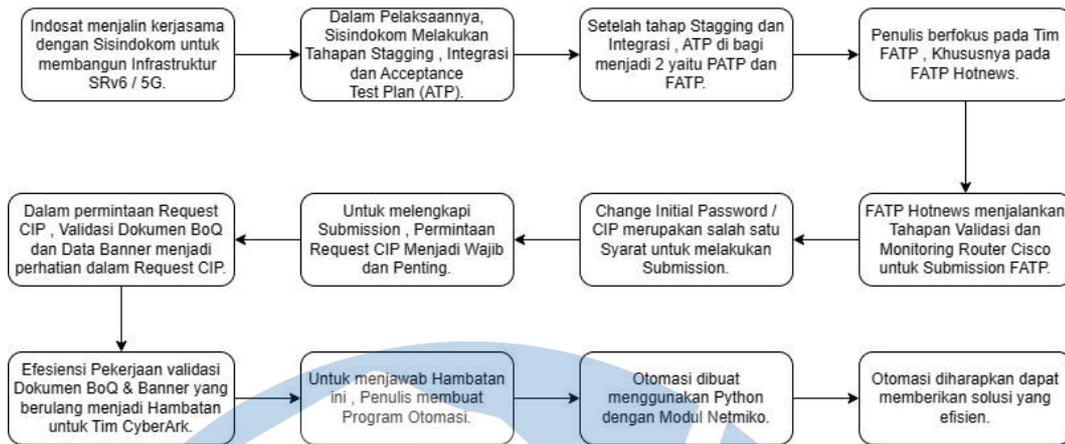
Tahap penelitian ini digunakan untuk memberitahu kondisi waktu penelitian yang akan dilaksanakan sampai penelitian selesai. Adapun gambaran terkait jadwal penelitiannya

Tabel 3. 1 Jadwal Penelitian

No	Nama Kegiatan	2023				2024				
		Apr	Mei	Juni	Jul	Mar	Apr	Mei	Jun	Jul
1	Studi Literatur									
2	Studi Lapangan									
3	Analisis									
4	Perancangan Sistem									
5	Implementasi									
6	Evaluasi									
7	Penulisan Penelitian									
8	Penarikan Kesimpulan dan Saran									
9	Sidang									

3.2.4 Alur Pekerjaan Pengembangan

Adapun Alur Pekerjaan yang dilakukan oleh penulis pada proyek fatp , adalah sebagai berikut :



Gambar 3. 2 Workflow FATP Sisindokom

Berdasarkan Gambar 3.3 yang dilakukan pada PT. Sisindokom Lintasbuana, diketahui bahwa Indosat menggandeng Sisindokom menjadi mitra mereka dengan pekerjaan yang sudah disebutkan di awal. Dibawah arahan Sisindokom , penulis melakukan pekerjaan validasi dan *submission FATP* dengan Tim *FATP Hotnews* .

Untuk melengkapi *Task Submission FATP* dibutuhkan *CyberARK Change Initial Password (CIP)* , permintaan *request CyberArk Change Initial Password(CIP)* hanya dapat dilakukan oleh Tim *CyberARK*. Dalam proses *request CIP* yang dilakukan oleh Tim *CyberARK*, Tim *CyberARK* melakukan validasi dokumen *Bill Of Quantity* , *capture data banner* dan perbandingan *Chasis* pada setiap *hostname* router menjadi pekerjaan yang repetitif untuk melakukan permintaan *request CIP*. Untuk mengatasi hambatan ini , penulis membuat program otomasi yang diharapkan dapat memberikan solusi yang efisien untuk melakukan tugas yang repetitif yang dilakukan oleh Tim *CyberARK*.

3.2.5 Metode Pengujian

Penerapan metode pengujian menjadi salah satu tahapan krusial dalam pengembangan program yang dikembangkan. Dua metode pengujian yang diterapkan adalah *black-box testing* dan pengujian efektivitas program.

Pertama-tama, penulis menggunakan metode *black-box testing*. Metode ini berfokus pada pengujian fungsionalitas program tanpa memperhatikan struktur internal atau kode sumbernya. Dalam konteks pengembangan

perangkat lunak, *black-box testing* memungkinkan pengetesan dari perspektif pengguna akhir, di mana fokus utamanya adalah pada fitur-fitur yang diharapkan dari program tersebut. Dengan demikian, penulis melakukan serangkaian tes tanpa mengetahui detail implementasi dari program yang sedang dikembangkan. Hal ini membantu mengevaluasi apakah program berfungsi sesuai dengan spesifikasi yang telah ditetapkan, serta untuk mengidentifikasi kekurangan atau kelemahan yang mungkin ada. Selain *black-box testing*, penulis juga melakukan pengujian terhadap tingkat efektivitas program yang sedang dirancang. Pengujian ini bertujuan untuk mengukur seberapa efektif program tersebut dalam menyelesaikan tugas-tugasnya sesuai dengan tujuan yang telah ditetapkan. Pengujian efektivitas ini sering kali melibatkan pengumpulan data empiris, seperti pengukuran waktu eksekusi, keberhasilan dalam menyelesaikan tugas-tugas tertentu, dan evaluasi kinerja secara keseluruhan. Dengan membandingkan hasil pengujian dengan standar yang ditetapkan atau harapan yang diinginkan, penulis dapat mengevaluasi kualitas dan kinerja program secara lebih objektif.

Kedua metode pengujian ini menjadi penting dalam pengembangan program karena mereka memberikan wawasan yang berbeda tentang kualitas dan kinerja program. *Black-box testing* membantu memastikan bahwa rancangan program berfungsi sesuai dengan harapan pengguna, sementara pengujian efektivitas memberikan gambaran yang lebih luas tentang kinerja program secara keseluruhan.

3.2.6 Metode Implementasi dan Evaluasi

Pada tahap ini, implementasi dilakukan dengan cara menggunakan program otomasi ke sistem yang telah ada sebelumnya. Implementasi ini dibarengi oleh pengujian efektivitas dengan lima kali iterasi dengan 20 perangkat cisco yang telah ada. Apabila terdapat malfungsi atau ketidaksesuaian dengan program akan dilakukan Evaluasi dengan Tim *CyberArk* sebagai pengguna.

Pada tahap implementasi dalam pengembangan program otomasi, langkah pertama adalah menerapkan program tersebut ke dalam sistem yang sudah ada sebelumnya. Proses implementasi ini melibatkan memasukkan

program otomasi ke dalam lingkungan yang sudah ada, seperti sistem operasi atau jaringan yang telah diatur sebelumnya. Tujuannya adalah untuk mengintegrasikan program baru ke dalam infrastruktur yang sudah ada tanpa mengganggu kinerja keseluruhan sistem.

Selanjutnya, implementasi ini dibarengi dengan pengujian efektivitas. Pengujian dilakukan dengan melakukan serangkaian iterasi pada program otomasi tersebut. Dalam konteks ini, disebutkan bahwa program diuji dengan melakukan lima kali iterasi pada 20 perangkat Cisco yang telah ada. Hal ini bertujuan untuk memastikan bahwa program beroperasi dengan baik dan konsisten dalam berbagai situasi yang mungkin terjadi. Dengan melakukan pengujian secara berulang, penulis dapat mengevaluasi kinerja program secara menyeluruh dan mengidentifikasi masalah potensial yang perlu diperbaiki. Apabila terjadi malfungsi atau ketidaksesuaian dengan program selama tahap implementasi dan pengujian, langkah selanjutnya adalah evaluasi. Evaluasi dilakukan dengan melibatkan Tim *CyberArk* sebagai pengguna program. Mereka akan memberikan umpan balik tentang keberhasilan dan kegagalan program serta memberikan saran untuk perbaikan yang diperlukan. Evaluasi ini penting untuk memastikan bahwa program memenuhi kebutuhan dan harapan pengguna, serta untuk melakukan penyesuaian yang diperlukan agar program dapat berjalan sesuai dengan yang diinginkan.

STT - NF

BAB IV

IMPLEMENTASI DAN EVALUASI

Bab ini tersusun atas penjelasan hasil uji coba terhadap sistem otomasi *capture banner* pada perangkat jaringan yang dikembangkan beserta hasil uji coba. Pengujian dan hasil dilakukan untuk melihat performa dan efektivitas yang dimiliki oleh sistem yang dikembangkan.

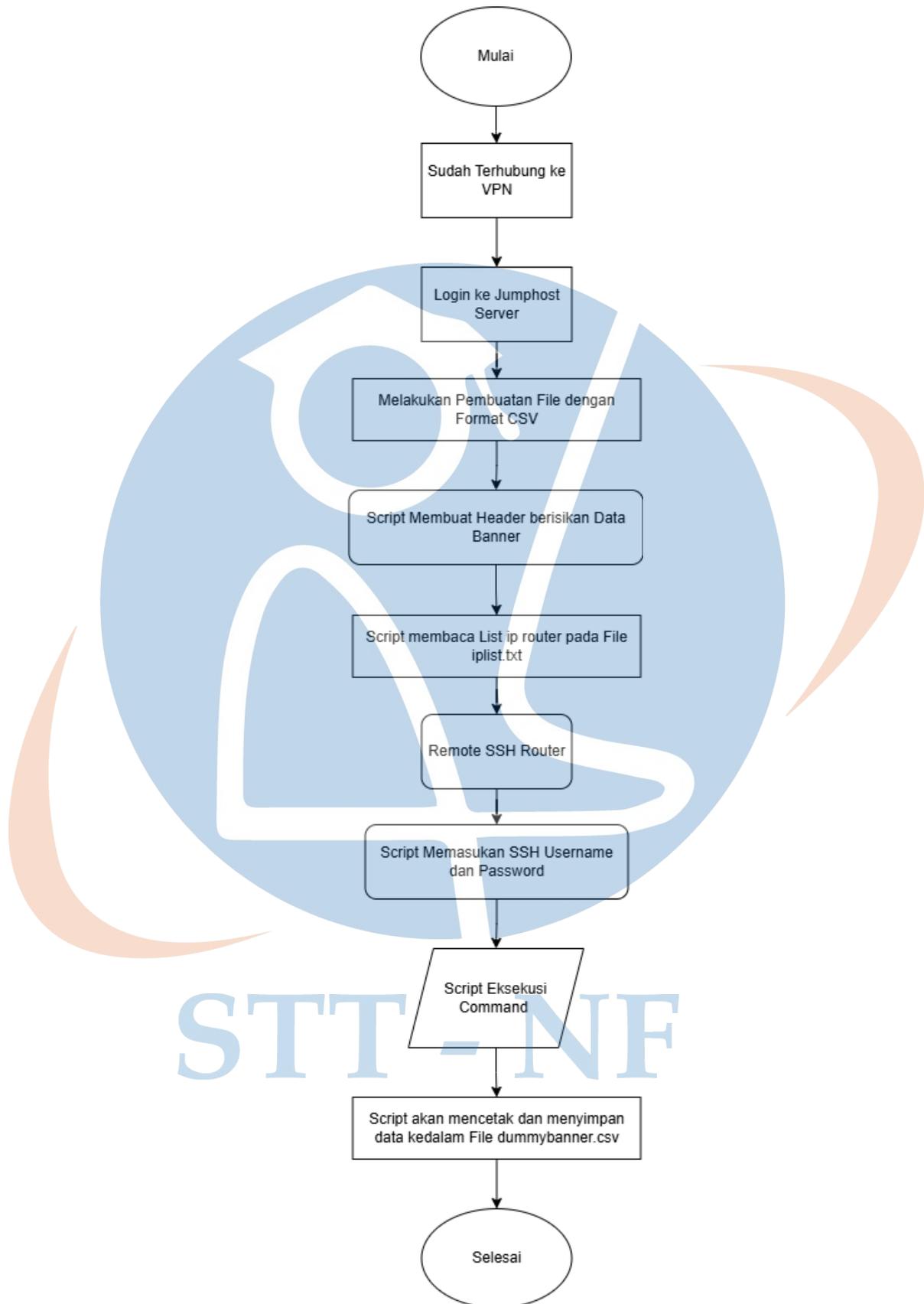
4.1 Rancangan Penelitian

Perancangan sistem yang dibuat mengikuti alur program yang sudah di implementasi langsung ke dalam sistem *Jumphost server* terkait. Perancangan desain ini, digunakan untuk menggambarkan proses pengguna dan pengembang dalam menjalankan otomasi *capture banner*. Adapun perancangan sistem yang digunakan berupa *Flowchart* , *Topology Logical* serta didukung dengan *Requirement User*.

4.1.1 Flowchart

Pada tahapan ini, terdapat hasil dari perancangan program yaitu *Flowchart* dari alur kerja program otomasi yang dilakukan oleh penulis. Wibawanto menuturkan[14] bahwa, *Flowchart* adalah representasi visual yang rinci dan terintegrasi dari proses-proses yang berhubungan, yang menggunakan simbol-simbol khusus untuk menggambarkan langkah-langkah dalam suatu prosedur analisis masalah.

Pada tahap ini, *Flowchart* merupakan representasi visual dari alur kerja program otomasi yang dirancang untuk memudahkan pemahaman dan implementasi pada Pengujian sistem. berikut ini Gambar 4.1 *Flowchart* merupakan alur kerja program otomasi :

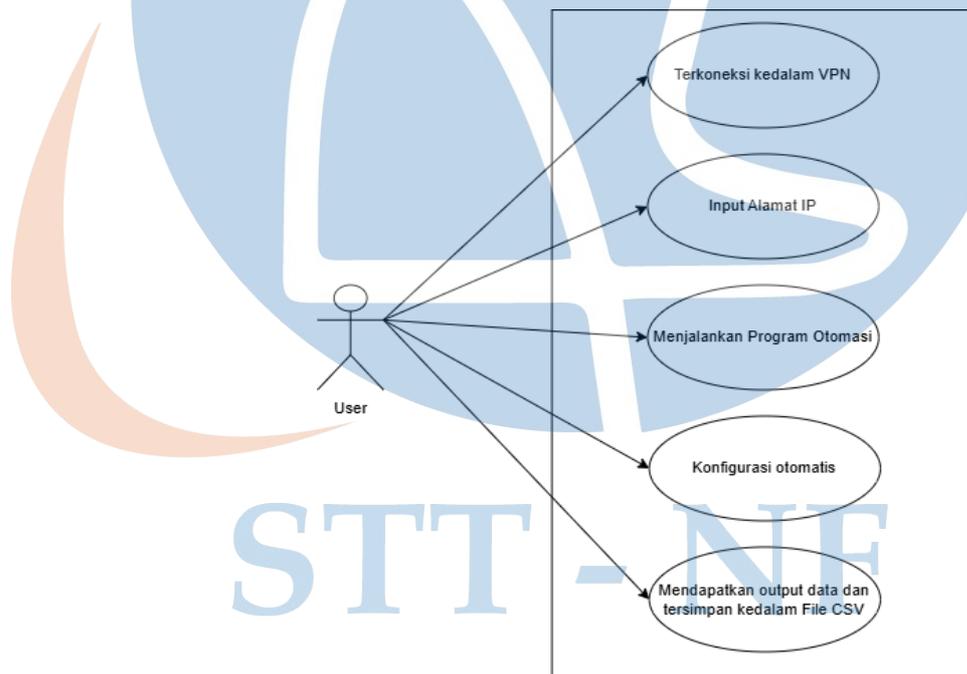


Gambar 4. 1 Flowchart

4.1.2 Requirement User

Pada tahapan ini, merupakan tahapan penjelasan *Requirement User* dalam pengembangan skrip otomatisasi jaringan untuk mempermudah validasi. Tahapan pada *Requirement User* ini menggunakan *use case diagram* sebagai *tools*, penggunaan *use case diagram* dalam tahapan *Requirement User* untuk menggambarkan interaksi antara pengguna dan skrip.

Use case diagram terdiri dari *actor*, *use case* dan relasi hubungannya. Penggunaan *use case diagram* penting untuk memvisualisasi, merinci, dan mendokumentasikan kebutuhan perilaku skrip pada sistem yang berjalan. *Use case diagram* digunakan untuk menjelaskan kegiatan apa saja yang dapat dilakukan oleh *user*/pengguna pada sistem yang berjalan. Adapun penggambaran sistem dalam bentuk *use case diagram* terlihat pada Gambar 4.2 berikut ini :

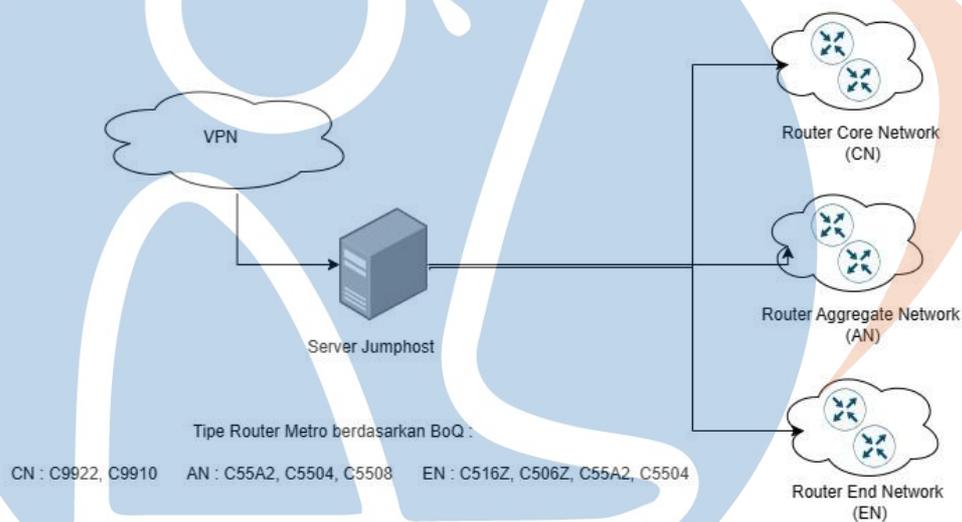


Gambar 4. 2 Use Case Diagram

4.1.3 Topology Logical

Pada tahapan ini merupakan tahapan penjelasan terkait penggunaan *topology* logical untuk menggambarkan tahap perancangan desain. Menurut Tambunan *et al* [15] bahwa Topologi logika (Logical

topology) adalah perancangan arsitektur jaringan komputer yang berkaitan erat dengan gambaran aliran data dan traffic pada suatu lintasan jaringan komputer. Sedangkan topologi fisik (*Physical topology*) adalah bentuk pengkabelan dan implementasinya pada suatu jaringan komputer. Topologi logika (*Logical topology*) akan digunakan oleh pengembang untuk melakukan perancangan sistem yang akan di bangun serta melakukan tahap pengujian sistem pada tahap selanjutnya. Berikut merupakan *topology* logical pada saat *testing* saat implementasi program otomatis terlihat dibawah ini :



Gambar 4. 3 Topologi logika

Seperti diperlihatkan dalam Gambar 4.3 *Topology* Logical diatas menunjukan Topologi logika (*Logical topology*) yang merupakan gambaran kecil terkait beberapa *site/router* yang terhubung kedalam sistem bernama *jump host* server, pada sistem yang berjalan terdapat *site/router* dengan tipe dan penamaan chassis yang berbeda. Pengguna yang sudah terhubung dengan VPN dapat langsung mengakses *jump host* server untuk melakukan validasi dan *capture* data-data yang diperlukan pada *banner* router.

4.2 Implementasi Rancangan Penelitian

Implementasi rancangan pengujian dari penelitian yang dilakukan bertujuan untuk mengetahui program pengujian yang dikembangkan, sekaligus dapat

mengetahui sejauh mana tingkat efektivitas pengujian sistem otomasi *capture banner*. Dalam Implementasi rancangan ini, data hasil pengujian menggunakan sebanyak 20 perangkat Cisco router dengan eksekusi *capture banner* sebanyak 20 kali dengan lima Iterasi.

4.2.1 Rencana Pengujian

Skenario uji coba dari penelitian yang dilakukan bertujuan untuk mengetahui sampai mana efektivitas sistem otomasi *capture banner* yang dikembangkan, skenario ini dibagi menjadi 2 yaitu pengujian efektivitas sistem dengan metode black box dan hasil pengujian efektivitas sistem.

4.2.2 Pengujian Sistem Otomasi

Tahapan Pengujian Sistem Otomasi ini merupakan tahap implementasi sistem otomasi *capture banner* dari hasil analisis serta kebutuhan pada keadaan pada Tim *CyberArk* yang tertera di bab sebelumnya. Sebelum sistem ini berjalan dibutuhkan beberapa sumber daya diantaranya list *ip address* pada perangkat/site yang akan diambil data *banner* untuk selanjutnya dilakukan validasi serta perbandingan dengan dokumen *Bill of Quantity*(BoQ) oleh Tim *CyberArk*. Pada hasil wawancara yang dilakukan dengan pengguna, didapatkan informasi sumber daya untuk melakukan implementasi otomasi kedalam *jumphost* server , adapun rincian informasi tersebut.

Berikut dibawah ini rincian yang dibutuhkan :

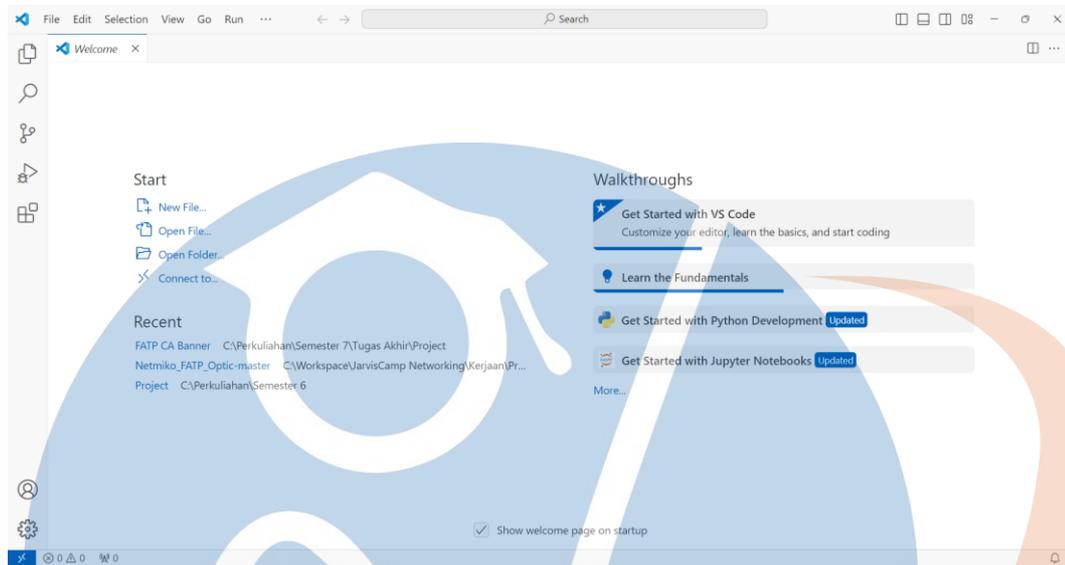
1. *List IP Address* perangkat/site
2. *Username Login ssh*
3. *Password Login ssh*

Setelah rincian informasi sumber daya yang dibutuhkan sudah lengkap, kita bisa langsung ke tahapan pengujian sistem otomasi dengan tahapan rancangan penelitian yaitu flowchart yang diterapkan pada perancangan sistem yang dibangun. Adapun tahapan-tahapan pengujian yaitu :

- Membuka Aplikasi *Text editor* (*Visual Studio Code*)

Pada penelitian kali ini penulis menggunakan aplikasi *Integrated Development Environment* (IDE) yaitu *Visual studio code*. Tahapan pertama yang

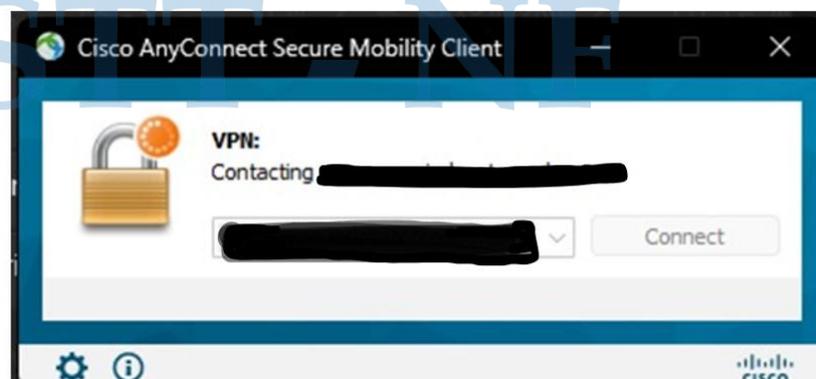
dilakukan tentunya membuka terlebih dahulu aplikasi *Text editor* yang dimiliki, di tahap ini penulis menggunakan *Visual studio code* seperti yang terlihat pada Gambar 4.2.1.



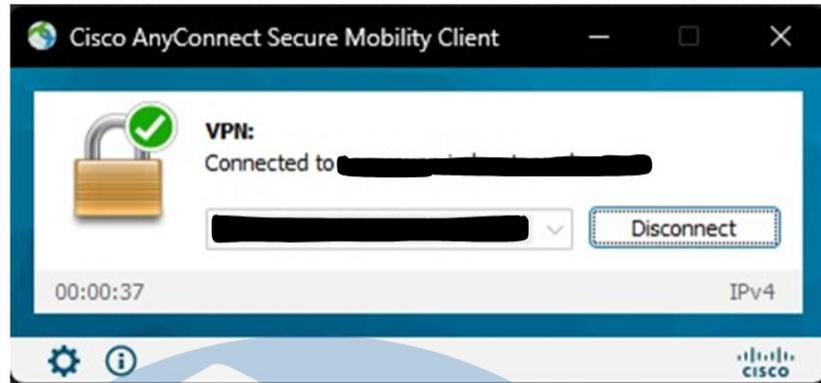
Gambar 4.2. 1 Capture tampilan awal vscode

- Membuka VPN *Anyconnect*

Setelah itu, penulis menghubungkan *Virtual Private Network*(VPN) menggunakan aplikasi *Cisco Anyconnect Secure Mobility Client* seperti yang terlihat pada Gambar 4.2.2 dan Gambar 4.2.3. VPN berfungsi untuk memungkinkan penulis dapat terhubung dan masuk *Jump host Server*. Dengan demikian, penulis dapat mengakses perangkat atau *site* yang akan dilakukan validasi menggunakan skrip otomatis.



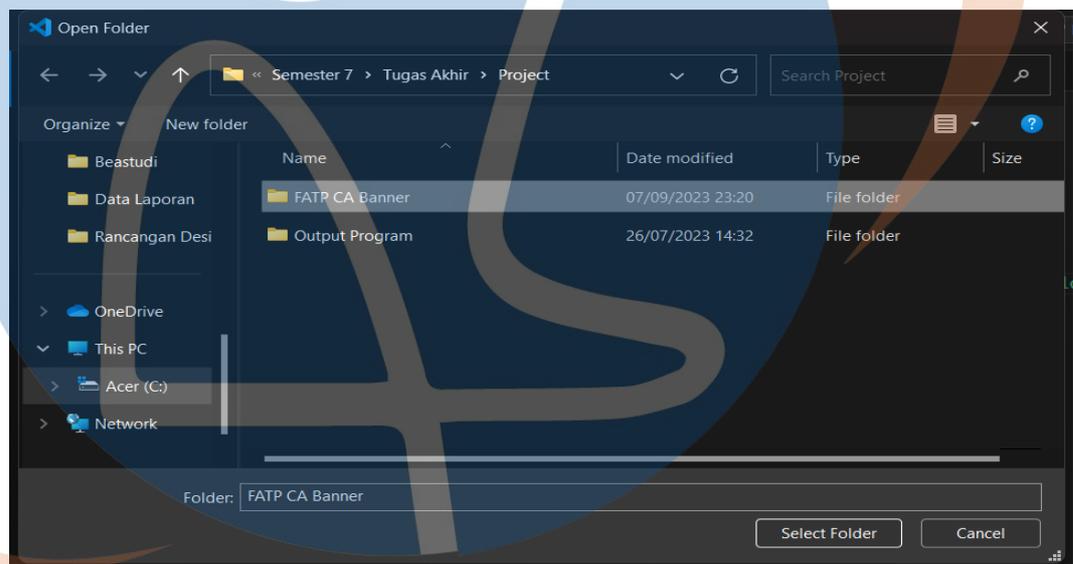
Gambar 4.2. 2 Capture hubungan ke VPN



Gambar 4.2. 3 *Capture* terhubung ke VPN

- Membuka *Folder* Otomasi

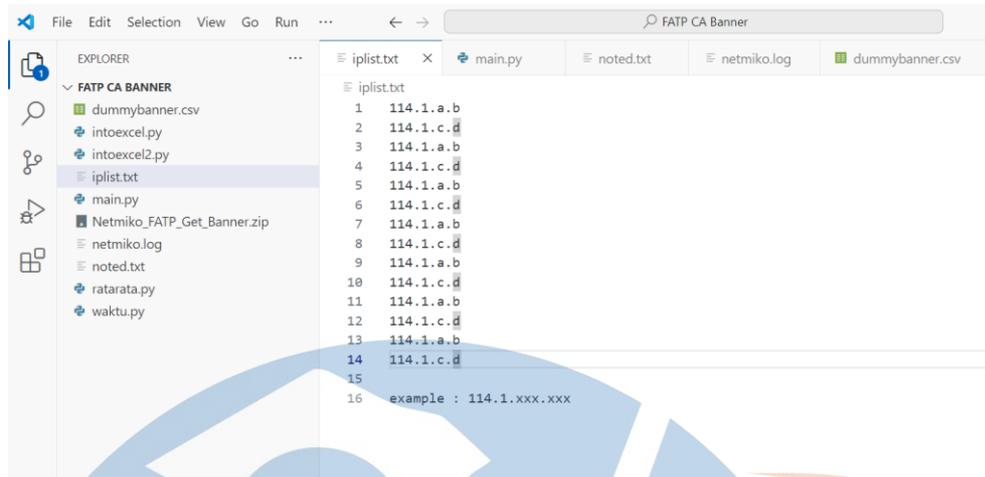
Tahap selanjutnya adalah membuka *folder* pada *Windows* 11 yang sudah terbuat dan pilih *folder file* yang berisikan skrip otomasi ke dalam *code* editor.



Gambar 4.2. 4 *Capture* membuka folder Program Otomasi

- Membuka List IP Address

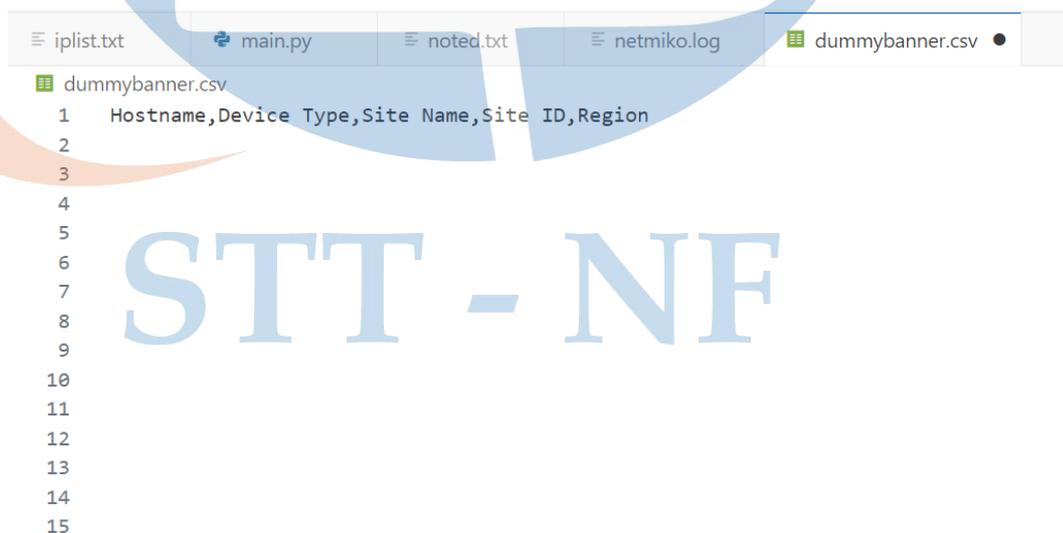
Langkah selanjutnya, penulis membuka *file* yang berisikan daftar *ip address*. *File* yang berisikan daftar *ip address* yang diberikan oleh Tim *CyberArk* melalui hasil wawancara, akan digunakan untuk dilakukan pengujian pada skrip otomasi *capture banner* secara otomatis. Adapun daftar alamat ip perangkat yang terlihat pada Gambar 4.2.5 dibawah ini.



Gambar 4.2. 5 Capture tampilan daftar alamat ip perangkat

- Membuat *File* format csv

Setelah itu, program otomatis membuat *file* yang berformat *Comma-separated Values(CSV)* yang akan menampung semua data *banner* yang telah diambil oleh skrip otomatis sesuai dengan hasil diskusi oleh Tim *CyberArk* sebagai pengguna. Jika *file* belum dibuat oleh pengguna sebelumnya, program otomatis akan membuat *file CSV* serta *layout header* secara otomatis ketika menjalankan skrip otomatis. Adapun *layout header* seperti berikut (*Hostname , Device Type , Site Name, Site ID, Region*) yang terlihat pada Gambar 4.2.6 dibawah ini.



Gambar 4.2. 6 Capture tampilan membuat&membuka *file* CSV

- Eksekusi Program dimulai

Tahap awal eksekusi dalam pengujian efektivitas dari program otomasi *capture banner* adalah menjalankan otomasi sesuai dengan *input command* yang telah diberikan ke dalam *file* utama yang telah dirancang. Nantinya, hasil *output* dari program otomasi tersebut akan disimpan ke dalam *file* format *CSV* sesuai dengan arahan dan diskusi yang diadakan pengembang dengan Tim *CyberArk*.

```

main.py > ...
17
18 koneksi = ConnectHandler(**jumphost)
19 prompt = koneksi.find_prompt()
20
21 # Create & Open CSV file
22 file = open('dummybanner.csv', 'w')
23
24 # Header Default CSV
25 header = ['Hostname', 'Device Type', 'Site Name', 'Site ID', 'Region']
26
27 # Cetak Data Output Python Pada File
28 writer = csv.writer(file)
29 # Cetak Data Header Output Python Pada File
30 writer.writerow(header)
31
32 # Membuka List Txt

```

```

File "C:\Software Support\Python\Data\Lib\site-packages\netmiko\channel.py", line 74, in write_channel
PS C:\Perkuliahan\Semester 7\Tugas Akhir\Project\FATP CA Banner> & "C:/Software Support/Python/Data/python.exe"
"c:/Perkuliahan/Semester 7/Tugas Akhir/Project/FATP CA Banner/main.py"
['BDG-UJING-AN1-C5504']
['BDG-UJING-AN1-C5504', 'NCS-5504']
['BDG-UJING-AN1-C5504', 'NCS-5504', 'UJUNG_BERUNG']
['BDG-UJING-AN1-C5504', 'NCS-5504', 'UJUNG_BERUNG', '03BDG013']
['BDG-UJING-AN1-C5504', 'NCS-5504', 'UJUNG_BERUNG', '03BDG013', 'WEST']
['PWK-PRWK-AN1-C5508']
['PWK-PRWK-AN1-C5508', 'NCS-5508']
['PWK-PRWK-AN1-C5508', 'NCS-5508', 'PURWAKARTA']
['PWK-PRWK-AN1-C5508', 'NCS-5508', 'PURWAKARTA', '03PWK004']
['PWK-PRWK-AN1-C5508', 'NCS-5508', 'PURWAKARTA', '03PWK004', 'WEST']
['TSM-TWNG-AN1-C5504']

```

Gambar 4.2. 7 Capture program sedang dalam eksekusi

- Eksekusi Program Selesai

Tahap terakhir Eksekusi yang dilakukan secara otomatis oleh program otomasi, sesuai dengan perancangan sistem yang dibangun dan diimplementasikan pada sistem *Jumphost server*. Eksekusi dinyatakan telah selesai dengan keberhasilan fungsi skrip yaitu mengambil dan menyimpan output data *banner* pada setiap *site* yang dieksekusi kedalam *file* csv, hasil output tersebut akan digunakan untuk mempermudah validasi data BoQ oleh Tim *CyberArk*. Adapun bukti keberhasilan pengujian efektivitas dari program otomasi *capture banner* adalah pengambilan output data banner yang dapat dilihat pada Gambar 4.2. 8 dan Gambar 4.2. 9 berikut:

```

dummybanner.csv
1  Hostname,Device Type,Site Name,Site ID,Region
2
3  BDG-UJNG-AN1-C5504,NCS-5504,UJUNG_BERUNG,03BDG013,WEST
4
5  PWK-PRWK-AN1-C5508,NCS-5508,PURWAKARTA,03PWK004,WEST
6
7  TSM-TWNG-AN1-C5504,NCS-5504,TAWANG,03TSK006,WEST
8
9  TSM-CAWI-AN1-C5504-XYX,NCS-5504,CIAWI,03TSM006,WEST
10
11 TGL-LMPG-AN1-C5504,NCS-5508,LIMPUNG,14BAG003,CENTRAL
12
13 YOG-MGLT-AN1-C5504,NCS-5504,MAGELANGTM,14MGL002,CENTRAL
14
15 BDG-NICL-AN1-C5504,NCS-5504,NICOLE,03BDG045,WEST
16

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
File "C:\Software Support\Python\Data\Lib\site-packages\netmiko\channel.py", line 74, in write_channel
PS C:\Perkuliahan\Semester 7\Tugas Akhir\Project\FATP CA Banner> & "C:/Software Support/Python/Data/python.exe"
khir/Project/FATP CA Banner/main.py"
['BDG-UJNG-AN1-C5504']
['BDG-UJNG-AN1-C5504', 'NCS-5504']
['BDG-UJNG-AN1-C5504', 'NCS-5504', 'UJUNG_BERUNG']
['BDG-UJNG-AN1-C5504', 'NCS-5504', 'UJUNG_BERUNG', '03BDG013']
['BDG-UJNG-AN1-C5504', 'NCS-5504', 'UJUNG_BERUNG', '03BDG013', 'WEST']
['PWK-PRWK-AN1-C5508']
['PWK-PRWK-AN1-C5508', 'NCS-5508']
['PWK-PRWK-AN1-C5508', 'NCS-5508', 'PURWAKARTA']
['PWK-PRWK-AN1-C5508', 'NCS-5508', 'PURWAKARTA', '03PWK004']
['PWK-PRWK-AN1-C5508', 'NCS-5508', 'PURWAKARTA', '03PWK004', 'WEST']
['TSM-TWNG-AN1-C5504']

```

Gambar 4.2. 10 Capture Program berhenti & output hasil program

```

dummybanner.csv
1  Hostname,Device Type,Site Name,Site ID,Region
2
3  BDG-UJNG-AN1-C5504,NCS-5504,UJUNG_BERUNG,03BDG013,WEST JAVA
4
5  PWK-PRWK-AN1-C5508,NCS-5508,PURWAKARTA,03PWK004,WEST JAVA
6
7  TSM-TWNG-AN1-C5504,NCS-5504,TAWANG,03TSK006,WEST JAVA
8
9  TSM-CAWI-AN1-C5504-XYX,NCS-5504,CIAWI,03TSM006,WEST JAVA
10
11 TGL-LMPG-AN1-C5504,NCS-5508,LIMPUNG,14BAG003,CENTRAL JAVA
12
13 YOG-MGLT-AN1-C5504,NCS-5504,MAGELANGTM,14MGL002,CENTRAL JAVA
14
15 BDG-NICL-AN1-C5504,NCS-5504,NICOLE,03BDG045,WEST JAVA
16
17 TGL-KMTN-AN1-C5504,NCS-5504,KEMANTRAN,14SLW004,CENTRAL JAVA
18
19 CRB-AMPL-AN1-C5508,NCS-5504,KARANGAMPEL,03CRN010,WEST JAVA
20
21 BDG-KLJT-AN1-C5504,NCS-5504,KALIJATI,03SUB008,WEST JAVA
22
23 TGL-PTRK-AN1-C5504,NCS-5504,PETARUKAN,14PHL002,CENTRAL JAVA
24
25

```

Gambar 4.2. 11 Capture output telah tersimpan dalam file csv

Pada Gambar 4.2. 12 dan Gambar 4.2. 13 terlihat penerapan rancangan penelitian yang dilakukan sesuai dengan harapan pengguna , yang dimana output data yang berhasil diambil oleh program langsung tersimpan ke dalam file berformat csv. Hasil output yang diperoleh dari skrip otomatisasi dapat digunakan

untuk melakukan validasi data BoQ maupun *Request CyberArk Initial Password* oleh Tim CyberArk

4.2.3 Hasil Pengujian

Hasil rancangan pengujian pada sistem otomasi *capture banner* dengan menggunakan metode pengujian *black box* melibatkan 20 perangkat Cisco dengan lima kali iterasi, rancangan pengujian dibuat untuk menyelesaikan masalah terkait dengan rumusan masalah serta masalah yang terjadi pada pekerjaan Tim *CyberArk*. Adapun hasil pengujian tersebut dipaparkan dalam bentuk Tabel 4.1 , sebagai berikut :

Tabel 4. 1 Tabel Iterasi hasil pengujian

Tahap	Keberhasilan Iterasi	
Iterasi	<i>Capture</i> Konfigurasi	Tersimpan ke <i>file</i> csv
Iterasi 1	100% Berhasil	100% Berhasil
Iterasi 2	80% Berhasil Sebagian	80% Berhasil Sebagian
Iterasi 3	60% Berhasil Sebagian	60% Berhasil Sebagian
Iterasi 4	Tidak berhasil	Tidak berhasil
Iterasi 5	100% Berhasil	100% Berhasil

Berdasarkan Tabel 4.1 iterasi hasil pengujian diatas menunjukkan 2 dari 5 iterasi pada skrip otomasi telah berhasil dalam persentase 100% keberhasilan, namun, beberapa hasil pengujian juga menghasilkan output sebagian persentase keberhasilan yang menunjukkan persentase 80% dan 60% . Selain itu, terdapat satu dari 5 iterasi yang tidak berhasil.

Selain pengujian iterasi guna menunjukkan rancangan pengujian dalam mempermudah validasi , adapun pengujian efektivitas sistem yang dibangun untuk menguji seberapa efisien otomasi yang berjalan pada sistem untuk membantu pengguna melakukan validasi. Berikut pengujian efektivitas sistem yang dipaparkan dalam bentuk tabel pada Tabel 4.2 Tabel Pengujian Efektivitas sebagai berikut ini :

Tabel 4. 2 Tabel Pengujian Efektivitas

Tahap	Pengujian Efektivitas	
Validasi	Waktu Validasi Manual	Waktu Validasi Otomatis
Validasi 1	00:47:31	00:29:15
Validasi 2	00:41:64	00:18:55
Validasi 3	00:29:60	00:19:57
Validasi 4	00:39:42	00:19:71
Validasi 5	00:32:53	00:08:30

Berdasarkan Tabel 4.2 Pengujian Efektivitas yang dilakukan, Dalam melakukan observasi pada pengujian efektivitas dibutuhkan waktu sekitar 47 detik dalam melakukan validasi manual untuk satu *site/router* dan hasil waktu yang dilakukan dengan otomatisasi sekitar 19 detik . Dimana data tersebut harus kita ambil dari banner router terlebih dahulu lalu kita simpan data tersebut kedalam file csv. Tentunya hal tersebut kurang efektif jika dalam satu perusahaan memiliki proyek ratusan *site/router* dan harus dilakukan validasi secara repetitif.

4.2.4 Analisis Evaluasi Pengujian

Keutamaan dari implementasi yang telah dilakukan adalah untuk menilai atau mengukur operasional yang telah dijalankan, apakah sudah sesuai dengan harapan pengguna. Evaluasi hasil implementasi yang dilaksanakan pada hasil pengujian menyatakan bahwa 3 dari 5 pengujian yang dilakukan terdapat malfungsi pada program maupun pengambilan hasil yang dilakukan oleh program. Adapun beberapa malfungsi yang terjadi dalam pengembangan program saat implementasi, sebagai berikut :

1. *Site/Router* yang divalidasi belum lolos dalam tahapan PATP
2. *Site/Router* yang sedang divalidasi berada dalam keadaan *un-manage* atau tidak dapat diakses dari luar jumphost.
3. Data pada banner belum memiliki data yang terbaru dan masih menggunakan data dari sebelum *swap* ke perangkat cisco.

Analisis evaluasi dilakukan terhadap malfungsi yang terjadi pada saat implementasi tersebut oleh pengguna dan pengembang untuk diambil titik penyelesaiannya, penyelesaian yang dilakukan dengan metode wawancara dan diskusi menghasilkan penyelesaian berupa sebagai berikut :

1. *Site/router* yang divalidasi lebih dahulu telah lolos dalam tahapan *PATP*.
2. *Site/router* yang divalidasi berada dalam keadaan aktif atau dapat diakses dari luar jumphost.
3. Data-data validasi pada daftar alamat *IP/site* yang diambil merupakan data terbaru.

Penyelesaian pada analisis evaluasi pengujian diharapkan dapat mempercepat implementasi pada skenario uji coba yang telah dilakukan sebelumnya. Selain itu, uji coba dilakukan tanpa adanya malfungsi yang disebabkan oleh masalah pada perangkat internal maupun layanan yang sedang berjalan di sistem. Hasil dari analisis evaluasi pengujian yang dilakukan, dapat membantu pengembang untuk merancang sistem dan pengembangan program otomatisasi untuk mempermudah validasi yang dilakukan.

Adapun saran pada pengujian program yang dapat disampaikan dari hasil penelitian ini untuk meningkatkan manfaat dan mendapatkan hasil yang lebih rinci pada waktu efektivitas sistem yang diuji, membangun fitur baru pada program yaitu menambahkan fitur pengukur waktu yang diperlukan untuk eksekusi kode.

STT - NF

BAB V

KESIMPULAN DAN SARAN

Pada bab V ini penulis menjelaskan tentang kesimpulan serta saran untuk penelitian ini.

5.1 Kesimpulan

Dengan merujuk pada hasil implementasi dan evaluasi skrip otomasi, peneliti menyimpulkan temuan yang dapat diperoleh dari penelitian terkait implementasi otomasi *capture banner*. Diambil kesimpulan sebagai berikut:

1. Dengan menggunakan otomasi capture banner, rancangan sistem program berjalan dengan baik pada setiap metro router untuk mempermudah validasi data BoQ sesuai dengan Iterasi yang dilakukan. Hasil pengujian pada bab sebelumnya, menunjukkan 2 dari 5 iterasi pada skrip otomasi telah berhasil dalam persentase 100% keberhasilan. Adapun malfungsi yang terdapat pada pengujian iterasi pada rancangan sistem program sudah dilakukan evaluasi mendalam kepada pengguna dan pengembang untuk menyelesaikan masalah tersebut.
2. Penerapan otomasi capture banner pada sisi jumhost server, menghasilkan efektivitas pekerjaan yang semakin efektif. Adapun pengujian terhadap efektivitas sistem otomasi yang dibangun menghasilkan rata-rata waktu dalam penggunaan otomasi sekitar 19 detik dan melakukan validasi secara manual sekitar 47 detik, serta jarak antara kedua rata-rata tersebut sekitar 28 detik membuktikan bahwa efektivitas yang dibangun dapat menekan pekerjaan yang repetitif dalam melakukan validasi.

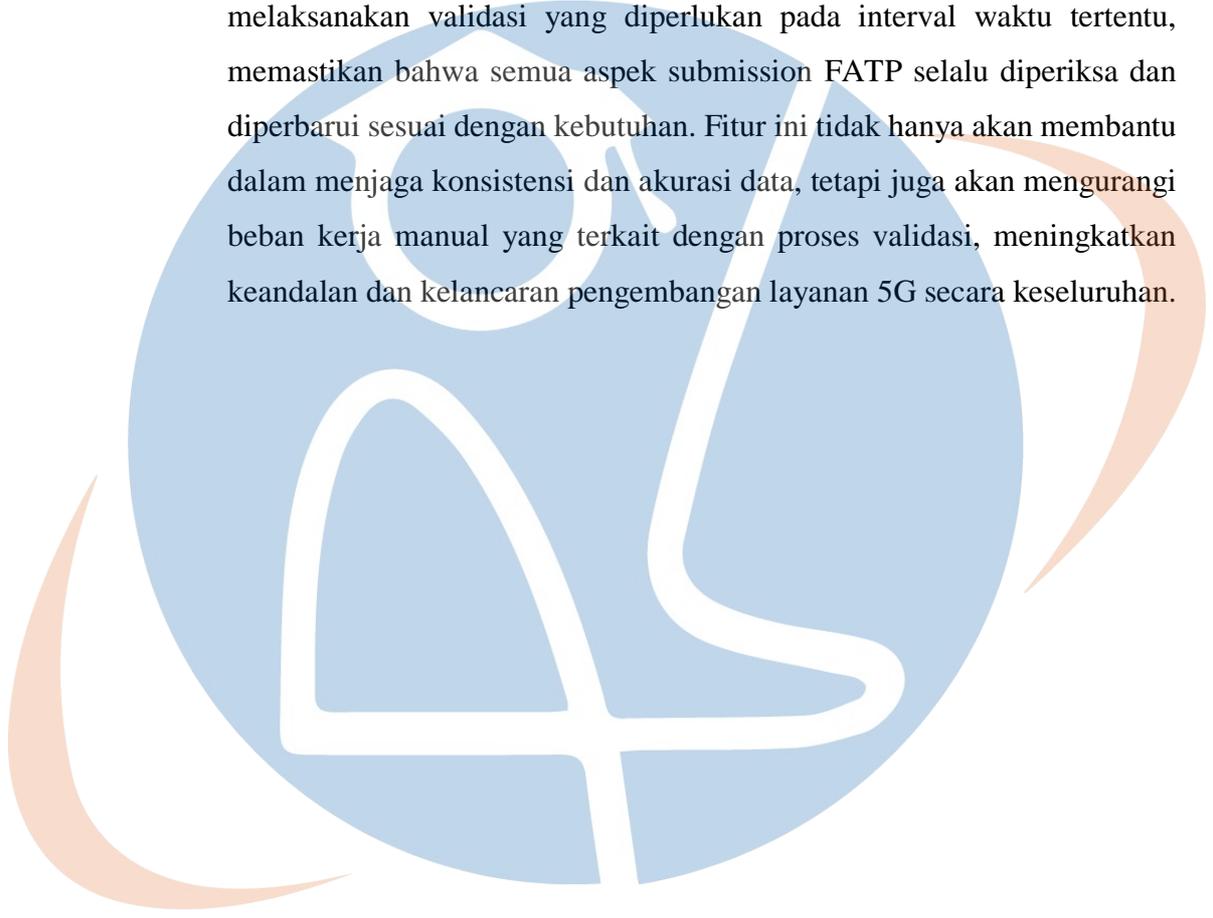
5.2 Saran

Berikut beberapa saran yang dapat disampaikan dari hasil penelitian ini untuk meningkatkan manfaat dan mendapatkan hasil yang lebih rinci:

1. Pada pengembangan program selanjutnya diharapkan peneliti untuk mengembangkan fitur baru pada program yaitu menambahkan fitur pengukur waktu yang diperlukan untuk eksekusi kode. Dengan tujuan

melihat seberapa lama efisiensi dari skrip otomatis yang telah berjalan dalam mengeksekusi perintah dalam membantu pekerjaan Tim CyberArk yang berulang.

2. Selain mengembangkan fitur baru pada program yaitu menambahkan fitur pengukur, peneliti diharapkan dapat menambahkan fitur baru yaitu fitur Schedule. Fitur ini diharapkan mampu secara otomatis menjadwalkan dan melaksanakan validasi yang diperlukan pada interval waktu tertentu, memastikan bahwa semua aspek submission FATP selalu diperiksa dan diperbarui sesuai dengan kebutuhan. Fitur ini tidak hanya akan membantu dalam menjaga konsistensi dan akurasi data, tetapi juga akan mengurangi beban kerja manual yang terkait dengan proses validasi, meningkatkan keandalan dan kelancaran pengembangan layanan 5G secara keseluruhan.



STT - NF

DAFTAR PUSTAKA

- [1] S. Hossain, "5G Wireless Communication Systems", University of Bangladesh, American Journal of Engineering Research (AJER),2013.
- [2] Sisindokom Lintasbuana, "Indosat Ooredoo Gandeng Sisindokom Luncurkan 5G" , www.sisindokom.com, 15 Maret, 2024, [Indosat Ooredoo Gandeng Sisiindokom Luncurkan 5G - Sisindokom](#).
- [3] A. M. Syahbandi, "Otomasi Backup Konfigurasi Perangkat Cisco Dengan Fitur Notifikasi Menggunakan Telegram Bot", Sekolah Tinggi Terpadu Nurul Fikri,2023.
- [4] Victor Saputra Ginting, Kusri, Emha Taufiq Luthfi," Penerapan Algoritma C4.5 Dalam Memprediksi Keterlambatan Pembayaran Uang Sekolah Menggunakan Python",Universitas Amikom Yogyakarta,2020.
- [5] M. Mashud,"Network Automation Menggunakan Bahasa Pemrograman Python", Fakultas Teknik, Universitas Bakti Indonesia,2022.
- [6] A. Boumezrag, "A Proposed of Novel Network Management Platform for Network Automation and Programmability with Implementation on GNS3," Master's Thesis, University of Biskra, 2020.
- [7] Luis Geraldo Mauboy & Theophilus Wellem , "Studi Perbandingan Library Untuk Implementasi Network Automation Menggunakan Paramiko Dan Netmiko Pada Router Mikrotik" , Universitas Kristen Satya Wacana,2022.
- [8] M. Maharani, "Monitoring Backup GSM Network to Handle Disorders on M2M SIM Cards Using Zabbix and SecureCRT", Politeknik Negeri Jakarta,2019.
- [9] Gerald Costa Pinaria, Yaulie Deo Rindengan, Xaverius B. N. Najoan , "Web Based E-Commerce Application Buying and Selling Food Ingredients for Manado City" , Universitas Sam Ratulangi,2021.
- [10] Jafar Shadiq, Ahmad Safei, Rayhan Wahyudin Ratu Loly, "Pengujian Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan Black Box Testing", Universitas Bina Insani,2021.

- 
- [11] Dona Yuliawati, Anggi Andriyadi, Nursiyanto, "Pengujian Sistem Informasi E-Monitoring Pengelolaan Pembangunan Desa Dengan Menggunakan Metode Blackbox Testing", Institut Informatika dan Bisnis Darmajaya,2022.
- [12] Yoyo Zakaria Ansori, "Pembinaan Karakter Siswa Melalui Pembelajaran Terpadu Di Sekolah Dasar", Universitas Majalengka,2020.
- [13] Dewi Kurniawati, Arta Ekayanti, "*PENTINGNYA BERPIKIR KRITIS DALAM PEMBELAJARAN MATEMATIKA*", Fakultas Keguruan dan Ilmu Pendidikan Universitas Muhammadiyah Ponorogo, 2020.
- [14] Febri Gilang Pratama, "Front End Sistem Informasi Dinas Koperasi Dan Usaha Kecil Menengah Kabupaten Mesuji Berbasis Web Menggunakan ReactJS", Politeknik Negeri Lampung,2021.
- [15] Muchamat Nur Rokim, Esron Rikardo Nainggolan, "Pemanfaatan Manajemen Jaringan Menggunakan Virtual Local Area Network (Vlan) Pada PT. Jantra Reka Saksanamas Cengkareng Timur Jakarta Barat",Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri Jakarta,2021.

STT - NF

LAMPIRAN

1. Adapun Script dari Otomasi menggunakan Netmiko bisa terlihat dibawah ini:

```
import time
import logging
import re
import csv
from netmiko import ConnectHandler

# Membuat Logging NetMiko Untuk Debug Error
logging.basicConfig(filename='netmiko.log', level=logging.DEBUG)

# Jumphost
jumphost = {
    "device_type" : "xxx",
    "ip" : "xxx",
    "username" : "xxxxxxxxxxx",
    "password" : "xxxxxxxxxxx",
}

# Membuat Koneksi ke Jumphost
koneksi = ConnectHandler(**jumphost)
prompt = koneksi.find_prompt()

# Create & Open CSV file
file = open('dummybanner.csv', 'w')

# Create Header Default CSV
header = ['Hostname', 'Device Type', 'Site Name', 'Site ID' ,
'Region']

# Simpan Data Output Python Pada File
writer = csv.writer(file)
# Cetak Data Header Output Python Pada File
writer.writerow(header)

# Membuka List Txt
with open('iplist.txt', 'r') as file:
    ipaddr = [line.strip() for line in file]

    for ip in ipaddr:
        # Akses SSH Remote
        koneksi.write_channel(f"xxxxxxxxxxx{ip}\n")
        # SSH Password
        time.sleep(3)
        koneksi.write_channel("xxxxxxxxxxx\n")
        time.sleep(2)
        koneksi.write_channel("terminal leng 0\n")
        time.sleep(2)

    # Perintah Otomasi
```

```

        commands = ['sh run banner | i Hostname', 'sh run banner |
i Device', 'sh run banner | i Site_Name' , 'sh run banner | i
Site_Id' , 'sh run banner | i Region']

        # Buat List
        filtered_output = []

        for command in commands:
            # Eksekusi Perintah
            output = koneksi.send_command(command)
            time.sleep(2)
            # Menghapus Tanda "=" & : dengan Spasi
            removed_unused_char = output.replace('=',
''.replace(':', ''))
            # Filter Menghapus Tanggal
            date_compiler = re.compile('\n.*?\n')
            # Filter Menghapus Simbol yang tidak digunakan
            hostname_compiler = re.compile('\n.*?#')
            # Eksekusi Filter Menghapus Tanggal
            removed_date = date_compiler.sub('',
removed_unused_char)
            # Eksekusi Filter Menghapus Simbol yang tidak digunakan
            removed_hostname = hostname_compiler.sub('',
removed_date)
            output_list = removed_hostname.split()
            # Eksekusi Menambahkan data yang sudah difilter dengan
append
            filtered_output.append(output_list[1])
            print(filtered_output)

        # Mencetak Data filtered_ouput ke dalam file CSV
        writer.writerow(filtered_output)
        # Program selesai
        koneksi.write_channel('exit\n')
        time.sleep(3)
        # Tutup File CSV
        file.close

```

STT - NF