

SISTEM AUTOMASI METHOD OF PROCEDURE UNTUK UPGRADE SOFTWARE ROUTER NOKIA BERBASIS CLI

Puri Ayu Purwaningsih^{1*}, Indyah Hartami Santi², Wahyu Dwi Puspitasari³

¹Fakultas Teknik dan Informatika, Universitas Islam Balitar

E-mail: puriap2611@gmail.com¹, indyhartamisanti@gmail.com², pushpitasari23@gmail.com³

INFO ARTIKEL

Sejarah Artikel

Diterima : 15/08/2025

Direvisi : 02/08/2025

Diterbitkan : 01/12/2025

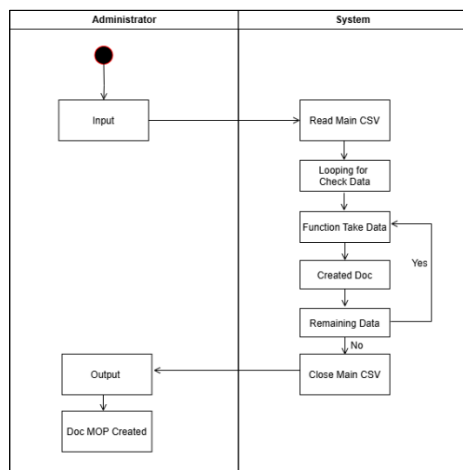
*Corresponding author

puriap2611@gmail.com

DOI: 10.xxxxxx/xxxx

<https://ojs.amiklps.ac.id>

GRAPHICAL ABSTRACT



ABSTRACT

PT XYZ, a telecommunications consulting company in Indonesia, particularly in West Java, currently creates Method of Procedure (MOP) documents for manual software updates, resulting in low productivity and a higher risk of errors. This research develops an automated system for creating MOPs using a Python-based command line interface to improve efficiency and accuracy. Using the Rapid Application Development (RAD) method, data was collected through observation, interviews, and literature review. The system utilizes Python modules such as csv and docxtpl to automate data extraction and document generation from templates. Testing involved black-box testing, validation by IT experts, and User Acceptance Testing (UAT). Results show the system successfully reads CSV data and generates well-organized MOP documents, achieving a 95% success rate in black-box testing, 97% in expert validation, and approximately 85% in user experience, integration, and functionality during UAT. This automated system has proven viable for improving efficiency and minimizing errors in MOP document creation at PT XYZ.

Keywords: Automation, MOP, Python, RAD, UAT, Telecommunications

ABSTRAK

PT XYZ, perusahaan konsultan telekomunikasi di Indonesia khususnya Jawa Barat, mengelola ribuan site jaringan. Pembuatan Method of Procedure (MOP) Upgrade Software yang masih manual menyebabkan penurunan produktivitas dan risiko kesalahan tinggi. Penelitian ini mengembangkan sistem automasi dokumen MOP berbasis Command Line Interface (CLI) dengan Python untuk meningkatkan efisiensi dan akurasi. Metode Rapid Application Development (RAD) digunakan, meliputi perencanaan, workshop desain, implementasi, dan pengujian. Data dikumpulkan melalui observasi, wawancara, dan studi pustaka. Sistem memanfaatkan modul Python seperti csv dan docxtpl untuk otomatisasi pengambilan data dan pembuatan dokumen berbasis template. Pengujian dilakukan dengan black box, validasi ahli IT, dan User Acceptance Testing (UAT). Hasil menunjukkan aplikasi mampu membaca data CSV dan menghasilkan dokumen MOP otomatis dengan output terstruktur. Black box testing berhasil 95%, validasi ahli 97%, dan UAT menunjukkan kemudahan sistem 85%, integrasi automasi 85%, serta fungsi automasi MOP 85%. Sistem dinilai layak meningkatkan efisiensi dan mengurangi kesalahan di PT XYZ.

Kata kunci: Automasi, MOP, Python, RAD, UAT, Telekomunikasi

PENDAHULUAN

PT XYZ merupakan perusahaan konsultan yang berfokus pada layanan di bidang telekomunikasi, khususnya dalam integrasi jaringan. Perusahaan ini berkomitmen menyediakan solusi inovatif serta layanan konsultasi untuk mendukung transformasi teknologi di sektor telekomunikasi, IP, dan IT. Dengan lokasi proyek yang tersebar di pulau Sumatra, Jawa, Kalimantan, Sulawesi, dan Bali-Nusra dan memiliki sebaran proyek terbesar di Pulau Jawa dengan pembagian berdasarkan *Micro Cluster* antara lain *cluster West Java Region*, *Central Java Region*, dan *East Java Region*. Dengan jangkauan geografis yang luas, PT XYZ telah berhasil menyelesaikan berbagai proyek skala besar yang melibatkan penyediaan layanan teknis dan konsultasi strategis. Fokus utama PT XYZ adalah memastikan infrastruktur jaringan yang andal dan efisien, seiring dengan perkembangan kebutuhan teknologi yang semakin kompleks.

Dengan kehadirannya yang mencakup ribuan site dalam skala target harian, penyusunan Method of Procedure (MOP) dianggap sangat penting sebagai langkah awal dalam melaksanakan proses integrasi, dan migrasi perangkat jaringan. Method of Procedure (MOP) adalah dokumen yang berisi serangkaian instruksi langkah demi langkah yang terperinci, menjelaskan tindakan dan urutan tugas yang diperlukan untuk menyelesaikan proses atau operasi tertentu, seperti instalasi, migrasi jaringan, dan *upgrade software*[1]. Dengan kemajuan teknologi yang semakin cepat dan terus berkembang saat ini, pekerjaan yang sebelumnya dilakukan secara konvensional kini dapat dikerjakan oleh mesin melalui metode otomatisasi. Dalam bahasa Indonesia, otomatisasi memungkinkan pekerjaan rutin yang dilakukan berulang kali dapat diselesaikan dengan lebih cepat dan efisien.

Berdasarkan data observasi yang telah dilakukan, untuk pekerjaan Upgrade Software PT XYZ di daerah Jawa Barat masih terdapat kendala dalam pembuatan MOP. Dalam tahap pembuatan MOP *Upgrade Software* semua data masih dimasukkan secara manual ke dalam template MOP satu per satu. Hal ini mengakibatkan penurunan produktivitas tim, karena waktu yang dihabiskan untuk tugas-tugas rutin mengurangi fokus pada peningkatan kualitas dan efisiensi kerja. Pembuatan dokumen MOP secara manual juga rentan terjadi masalah lain yaitu kesalahan manusia (*human error*)[2]. Bentuk kesalahan dapat berupa kesalahan leksikal atau bahasa maupun kesalahan substantif. Kesalahan leksikal atau bahasa dapat berupa kesalahan gramatikal, ejaan, dan diksi. Sedangkan Kesalahan substantif dapat terjadi pada tahap input, proses, maupun output.

Target harian untuk pembuatan MOP Upgrade Software di wilayah Jawa Barat adalah 30 hingga 50 MOP berdasarkan *Micro Cluster*. Angka target ini terus meningkat, tetapi metode manual yang digunakan tidak efektif untuk mencapai efisiensi yang diinginkan.

Tantangan utama dalam pembuatan MOP adalah tingginya volume pekerjaan yang sering kali menyebabkan keterlambatan serta meningkatkan risiko kesalahan dalam dokumentasi. Dokumen MOP Upgrade Software biasanya memuat berbagai data penting, seperti informasi perangkat (Hostname, Platform, IP Address, Site Name, Site ID, Versi Timos), topology jaringan, *site impact* (informasi site yang akan terpengaruh dalam proses *upgrade software*), *date/time execution*, *script* (*Upgrade version*, ME EX 3ID, NFMP, BGP Handling at L3VPN, Rollback). Dengan bentuk dan format yang sama, tim teknis IT seharusnya dapat membuat salinan dokumen MOP Upgrade Software secara otomatis dan mengurangi pekerjaan manual yang memakan waktu. Menindak lanjuti masalah yang terjadi maka diterapkan automasi pembuatan *Method of Procedure* (MOP) Upgrade Software dengan menggunakan pemrograman Python. Python dapat mempermudah otomatisasi tugas di berbagai bidang berkat sintaks yang sederhana, perpustakaan standar yang lengkap, dan banyaknya paket pihak ketiga. Python dapat digunakan untuk mengotomatiskan tugas rutin sistem file seperti menyalin, mengganti nama, dan mengatur direktori. Selain itu, Python juga mampu mengotomatiskan entri data, termasuk interaksi dengan formulir web atau aplikasi desktop. Dengan skrip otomatis, Python dapat mengisi formulir berdasarkan data dari database atau spreadsheet, sehingga mempercepat proses dan mengurangi risiko kesalahan.

Penelitian yang sama menjelaskan penerapan Python-pptx untuk menghasilkan laporan secara otomatis dan dom-to-image untuk mengintegrasikan data visual, sehingga membantu menyelesaikan permasalahan yang dihadapi. Proses manual sebelumnya, yang memakan waktu, rentan terhadap kesalahan, dan menghambat operasi, kini dapat diminimalkan secara signifikan. Penelitian ini juga menggambarkan bahwa Python-pptx adalah pustaka Python yang memungkinkan pembuatan dan pengelolaan dokumen presentasi PowerPoint secara otomatis dari berbagai sumber data [3].

Pembuatan sistem automasi MOP untuk proses upgrade software dilakukan melalui penelitian berbasis pengembangan menggunakan metode *Rapid Application Development* (RAD). Metode ini menekankan pada percepatan pengembangan aplikasi, dengan tujuan utama untuk memenuhi kebutuhan pengguna atau pemilik sistem secara lebih efisien. Metode ini mirip dengan prototyping, tetapi memiliki cakupan yang lebih luas[4]. Dalam proses pengembangan sistem automasi *Method of Procedure* (MOP), data dikumpulkan melalui metode observasi, wawancara, dan studi pustaka. Observasi dilakukan untuk memahami secara langsung proses pembuatan MOP yang sedang berlangsung, sehingga langkah-langkah yang memerlukan automasi serta potensi kendala dapat diidentifikasi. Wawancara dengan pihak terkait memberikan pemahaman lebih mendalam mengenai kebutuhan dan ekspektasi pengguna terhadap sistem yang

dirancang, serta masukan terkait fitur-fitur yang diharapkan. Selain itu, studi pustaka dilakukan untuk mengkaji literatur yang relevan, termasuk teori dan praktik dalam automasi dan pengembangan perangkat lunak, sehingga peneliti dapat merumuskan kerangka kerja yang solid dan berbasis bukti untuk implementasi automasi MOP. Tahap akhir dari metode ini yaitu tahap implementasi, sebelum pada tahap tersebut terdapat tahap pengujian. Pada tahap pengujian ini Rapid Application Development (RAD) menggunakan metode *User Acceptance Testing* (UAT). Implementasi metode *User Acceptance Testing* bertujuan untuk verifikasi output dokumen berdasarkan test case sesuai skenario real.

Berdasarkan analisis masalah yang telah disampaikan, maka perlu menentukan scope automasi yang akan dikembangkan untuk menentukan batasan dan ruang lingkup automasi, mengumpulkan dokumentasi MOP *Upgrade Software existing* untuk memahami prosedur yang telah diterapkan sebelumnya. Hal ini juga membantu dalam menentukan elemen-elemen mana yang perlu dipertahankan atau diperbaiki dalam sistem baru.

TINJAUAN PUSTAKA

Sistem Automasi

Sistem automasi merupakan merupakan bidang luas dan beragam yang mencakup proses, mesin-mesin, elektronika, perangkat lunak, dan sistem informasi yang bekerja sama menuju tujuan bersama yaitu peningkatan produksi, kualitas yang ditingkatkan, biaya yang rendah, dan fleksibilitas maksimal [5]. Dalam otomatisasi, berbagai sistem dan perangkat beroperasi secara mandiri untuk meningkatkan efisiensi, akurasi, dan konsistensi dalam pelaksanaan tugas-tugas tertentu [6].

MOP (Method Of Procedure)

Method of Procedure (MOP) merupakan sebuah dokumen yang secara rinci menjelaskan langkah-langkah yang perlu diambil dalam proses migrasi, termasuk penggantian perangkat keras jaringan yang diperlukan untuk menyelesaikan suatu proses atau operasi tertentu. Ini berfungsi sebagai panduan bagi individu atau tim untuk diikuti guna memastikan konsistensi, efisiensi, dan keselamatan dalam melaksanakan tugas-tugas yang kompleks (Ivo Pinto, 2023). Tingkat detail yang diperlukan dalam *Method of Procedure* (MOP) harus disesuaikan dengan kompleksitas pekerjaan dan besarnya potensi risiko. Pekerjaan yang relative rumit atau beresiko tinggi harus dirinci secara cermat dalam MOP [7]. MOP (*Method of Procedure*) merupakan alat yang digunakan secara luas di berbagai sektor industri dan aplikasi untuk memastikan bahwa setiap tugas atau proses yang memiliki tingkat kritisitas tinggi dilaksanakan dengan cara yang terorganisir dan aman.

Upgrade Software

Upgrade Software merupakan versi baru dari perangkat lunak yang ada guna memberikan pembaruan, perbaikan, dan peningkatan yang dirancang untuk secara signifikan meningkatkan produktivitas pengguna individu, yang pada gilirannya mengarah pada peningkatan produktivitas di tingkat organisasi [8] *Upgrade Software* merupakan salah satu jenis perubahan teknologi, tetapi itu pasti salah satu yang paling umum di organisasi modern, mengingat kecepatan cepat di mana vendor perangkat lunak merilis versi baru.

Python

Python adalah bahasa pemrograman tingkat tinggi yang dirancang untuk memudahkan pemrogram dalam menulis kode yang mudah dipahami dan dikelola. Sebagai bahasa yang bersifat interpretatif, *Python* mengeksekusi perintah-perintahnya melalui sebuah proses penerjemahan langsung per baris kode, tanpa perlu melalui tahap kompilasi terlebih dahulu [9]. *Python* mendukung berbagai paradigma pemrograman, termasuk pemrograman berorientasi objek, imperatif, dan fungsional, yang membuatnya sangat fleksibel dalam menangani berbagai jenis kebutuhan pengembangan perangkat lunak. Selain itu, *Python* dapat digunakan untuk berbagai macam tujuan pengembangan, mulai dari aplikasi desktop hingga pengolahan data besar, serta memiliki kompatibilitas yang luas dengan berbagai sistem operasi, menjadikannya pilihan populer di berbagai bidang pengembangan perangkat lunak [10].

Command Line Interface (CLI)

Command Line Interface merupakan sebuah aplikasi khusus saat membuka jendela windows. *Command Line* untuk windows ada CMD.exe, aplikasi ini bertanggung jawab untuk membuat jendela perintah, menerima perintah konfigurasi, dan menyediakan akses ke perintah bawaan seperti perintah `dir` [11] *Command Line* merupakan bagian terpenting dari baris perintah karena memengaruhi semua yang dilakukan di baris perintah. Perubahan kecil pada *Command Line* dapat membuat perubahan signifikan dalam cara aplikasi berjalan. Dengan menggunakan CLI, pengguna mengetikkan perintah tertentu untuk mengakses berbagai fungsi atau menjalankan tugas spesifik. Metode ini memberikan kontrol langsung dan lebih mendalam atas sistem, memungkinkan pengguna untuk melakukan berbagai operasi secara lebih efisien dan sering kali lebih cepat dibandingkan dengan antarmuka grafis.

Rapid Application Development (RAD)

Rapid Application Development (RAD) adalah pendekatan yang memungkinkan pengembangan sistem dengan cepat, karena sistem yang dihasilkan dirancang untuk memenuhi kebutuhan pengguna secara langsung. Hal ini membantu mengurangi waktu yang dibutuhkan untuk pengembangan ulang setelah implementasi. Menurut [12], metode RAD terdiri dari tiga fase utama yang melibatkan kerja

sama erat antara analis dan pengguna dalam proses evaluasi, perancangan, dan implementasi.

Black Box Testing

Pengujian *Black box* berfokus pada aspek fungsional perangkat lunak. Metode ini *Software Engineer* untuk merancang berbagai kondisi input yang dapat menguji secara menyeluruh seluruh persyaratan fungsional dari suatu program [13]. Istilah lain untuk pengujian *Black box* (BBT) termasuk pengujian spesifikasi, pengujian berbasis data, pengujian fungsional, dan pengujian berbasis input/output. Menurut [14] Teknik pengujian *Black box* dinamis diterapkan pada SUT (*System Under Test*) yang sedang berjalan tanpa pengetahuan sebelumnya tentang kode sumbernya. Konsep *Black box* berlaku untuk seluruh aplikasi perangkat lunak atau untuk satu unit atau fungsi/metode dari perangkat lunak tersebut.

User Acceptance Testing

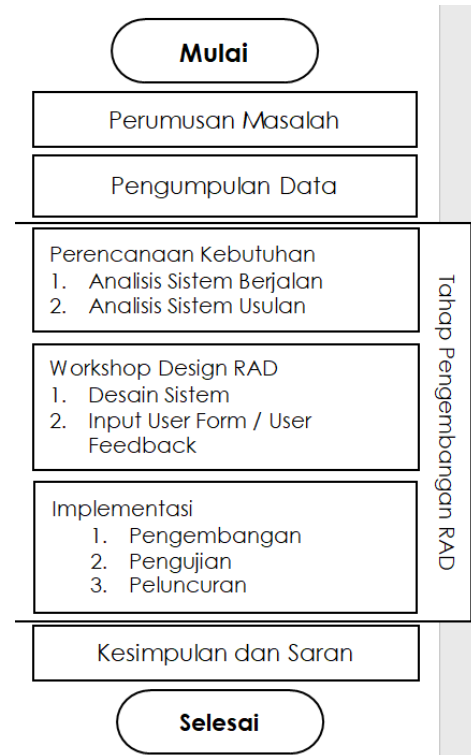
User Acceptance Testing (UAT) adalah proses validasi yang memastikan bahwa pengguna dapat melakukan tugas menggunakan sistem baru atau yang telah dimodifikasi, serta bahwa persyaratan yang ditetapkan telah didokumentasikan, ditafsirkan, dirancang, dan dikembangkan dengan baik (Cimperman, 2006). Tujuan *User Acceptance* adalah proses pengujian formal, yang sering kali didefinisikan, dilakukan, dan dimiliki oleh pelanggan atau pengguna sistem, dengan tujuan untuk menentukan apakah sistem memenuhi kriteria penerimaannya [16]. Penelitian ini memiliki keunggulan dibandingkan dengan penelitian sebelumnya karena menerapkan metode RAD (*Rapid Application Development*). Pendekatan ini menitikberatkan pada pengembangan sistem yang melibatkan pengguna secara langsung, sehingga memastikan bahwa setiap kebutuhan dapat terpenuhi dengan lebih efektif dan akurat.

Selain itu, penggunaan Python dalam penelitian ini memberikan fleksibilitas yang lebih luas. Sebagai bahasa pemrograman multipurpose, Python tidak hanya digunakan untuk analisis data, tetapi juga dapat diterapkan dalam berbagai bidang lain, seperti pengembangan web dan proses automasi. Hal ini menjadikan Python sebagai pilihan yang tepat dalam pengembangan sistem yang membutuhkan efisiensi dan kemudahan dalam implementasi.

METODOLOGI PENELITIAN

Tahapan Penelitian

Dalam penelitian ini ada beberapa tahapan yang akan dilakukan, mengacu pada jurnal karya M. Syaleh yang pada tahap selanjutnya dilakukan pengembangan di bagian RAD desain workshop dan tahap implementasi seperti pada gambar 1.1



Gambar 1. 1 Tahapan Penelitian

Perumusan masalah : Merumuskan masalah tentang pembuatan *Method of Procedure Upgrade Timos* dan mengetahui permasalahan yang terjadi dan kemudian dirumuskan oleh suatu sistem automasi.

Pengumpulan data : Pengumpulan data mengenai pembuatan MOP dilakukan melalui tiga metode utama wawancara, observasi, dan studi pustakas.

Perencanaan Kebutuhan : Tahap ini adalah tahap awal dari model pengembangan RAD. Di tahap ini, peneliti mengidentifikasi masalah yang ada lalu merancang rencana untuk menetapkan tujuan dan persyaratan yang diperlukan untuk mencapai tujuan.

Workshop Desain RAD : Tahap workshop design RAD yang meliputi proses perancangan desain sistem dan membangun sistem automasi *Method of Procedure* dilakukan dengan menggunakan metode *Rapid Application Development* (RAD) melalui proses *design workshop*.

Implementasi : Tahap implementasi, dilakukan pengembangan design yang telah dibuat pada tahap sebelumnya.

Kesimpulan dan Saran : Tahap ini, dapat ditarik kesimpulan mengenai hasil pengujian yang telah dilaksanakan, serta memberikan saran untuk menyempurnakan dan mengembangkan penelitian.

Pengumpulan Data

Pengumpulan data pada penelitian ini bertujuan untuk mendapatkan informasi dan data pada *MOP Upgrade Software*, selain itu untuk memahami kebutuhan pengguna serta keterbatasan terkait dengan perangkat lunak. Data yang

diperlukan dapat diperoleh dengan observasi, wawancara, dan studi pustaka.

Observasi : Observasi yang dilakukan bertujuan untuk mengamati objek yang akan diteliti. Metode observasi melibatkan pengamatan terhadap dokumen MOP yang tersedia di dalam *Sharepoint* proyek PT Lintas Teknologi Evolusi yang bertujuan untuk menentukan bagian-bagian dari MOP *existing*. Selain itu melakukan observasi terkait banyak nya data *site* yang berada pada area WJRO dengan jadwal yang sudah ditentukan.

Wawancara : Metode yang kedua adalah wawancara, metode ini dilakukan untuk menggali informasi maupun data melalui interaksi lisan. Wawancara dilakukan secara daring melalui *Google Meet*, yang memungkinkan interaksi langsung meskipun narasumber berada di lokasi yang berbeda. *Google Meet* dipilih karena kemudahan akses dan fitur yang mendukung komunikasi video secara efektif.

Studi Pustaka : Studi pustaka yang dilakukan terkait dengan pengembangan tersebut mencakup landasan teori yang relevan untuk mendukung pemahaman dan analisis dalam penelitian ini.

Tahapan Pengembangan

Perancangan dan pengembangan sistem automasi *Method of Procedure* (MOP) dilakukan dengan mengikuti tahapan *Rapid Application Development* (RAD), yang mencakup perencanaan kebutuhan, workshop desain, dan implementasi. Sistem ini dirancang untuk mempermudah pembuatan dokumen MOP dengan mengotomatisasi proses pengisian data yang sebelumnya dilakukan secara manual. Sistem bekerja melalui antarmuka *command prompt* yang sederhana, di mana data yang telah disiapkan dalam format *Microsoft Excel* secara otomatis dibaca dan diproses. Data tersebut kemudian diintegrasikan ke dalam template dokumen MOP berbentuk file *Microsoft Word*, sehingga menghilangkan kebutuhan pengetikan ulang dan mempercepat pembuatan dokumen sekaligus mengurangi risiko kesalahan input. Untuk memastikan kualitas dan keandalan sistem, pengujian dilakukan secara menyeluruh menggunakan metode *black box testing*, evaluasi oleh pakar IT terkait aspek teknis dan keamanan, serta *User Acceptance Testing* (UAT) untuk memastikan kemudahan penggunaan dan kesesuaian hasil dokumen dengan standar yang diharapkan pengguna.

Analisa Kebutuhan Sistem

Tahap perencanaan ini dianalisa terkait kebutuhan sistem dalam proses pengembangan, baik itu dari segi perangkat keras (*hardware*) maupun perangkat lunak (*software*). Hasil dari analisa kebutuhan sistem dapat dilihat pada table 1.1

Tabel 1. 1 Analisa Kebutuhan Sistem

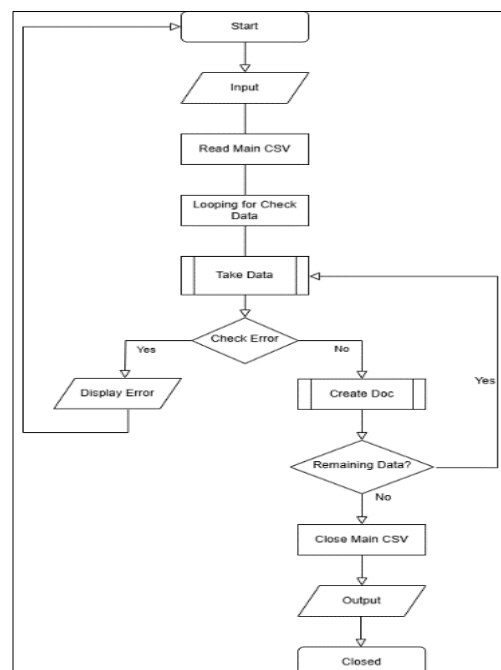
Kategori	Kebutuhan	Spesifikasi Minimal
Perangkat Keras	Komputer atau Laptop	-

(Hardware)	Prosesor	Intel Core i3 atau setara
	RAM	4 GB atau lebih
	Hardisk	256 GB SSD atau lebih
Perangkat Lunak (Software)	Sistem Operasi	Windows 10
	Aplikasi Pengembangan	Visual Studio Code
	Bahasa Pemrograman	Python 3 (module <i>pandas</i> , <i>os</i>)
	Terminal/Command Line	<i>Command prompt</i>
	Aplikasi Perkantoran	<i>Microsoft Office</i> (Word dan Excel)

Workshop Desain RAD

Tahap perancangan dilakukan dengan memanfaatkan alat pemodelan *Unified Modeling Language* (UML). Diagram yang diterapkan pada tahap ini meliputi desain sistem yaitu *flowchart*, *activity diagram*, *use case diagram*, *class diagram*, *Sequence diagram*, *struktur diagram*, dan desain *interface*.

Iterasi pertama dimulai dengan implementasi awal berdasarkan hasil wawancara dan diskusi awal dengan klien yang telah dirangkul dalam perancangan sistem. Pada tahap ini, dibuat prototipe aplikasi sesuai pemahaman awal untuk memberikan gambaran kepada klien. Prototipe ini belum tentu sepenuhnya sesuai dengan harapan klien, sehingga iterasi pertama berfungsi sebagai tahap eksplorasi untuk mengidentifikasi kebutuhan yang sebenarnya melalui umpan balik dari klien. Proses *Flowchart* alur program dimulai dengan membaca file CSV utama dan masuk ke dalam perulangan untuk memproses setiap baris data. Setiap kali data diambil, sistem memeriksa apakah pengambilan berhasil jika tidak, pesan kesalahan ditampilkan. *Flowchart* alur program dapat dilihat seperti pada gambar 1.2



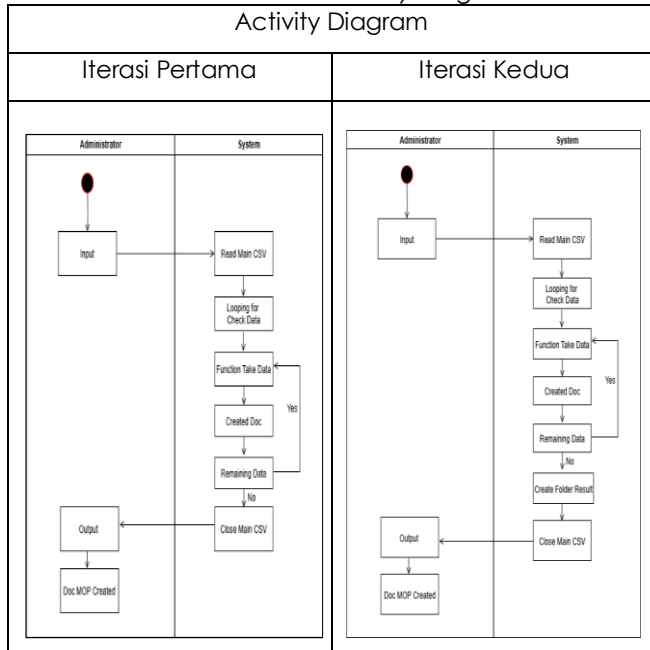
Gambar 1. 2 Flowchart

Proses *workshop* desain RAD pada pembuatan activity diagram dan use case diagram menghasilkan 2 proses iterasi, setelah menerima masukan dari klien berdasarkan prototype yang telah diserahkan pada iterasi pertama, dilakukan evaluasi untuk menentukan aspek-aspek yang perlu diperbaiki dan ditingkatkan agar sistem otomatis MoP upgrade software dapat berjalan lebih optimal dan sesuai dengan kebutuhan pengguna. Perubahan utama yang dilakukan yaitu:

1. Proses pembuatan *Method of Procedure* (MoP) diatur agar berjalan berdasarkan kategori jenis perangkat, yaitu membedakan antara device ring dan non-ring.
2. Penambahan fitur untuk secara otomatis membuat folder penyimpanan hasil MoP yang berhasil dibuat.
3. Penambahan peran engineer sebagai pengguna dengan akses penuh ke aplikasi.

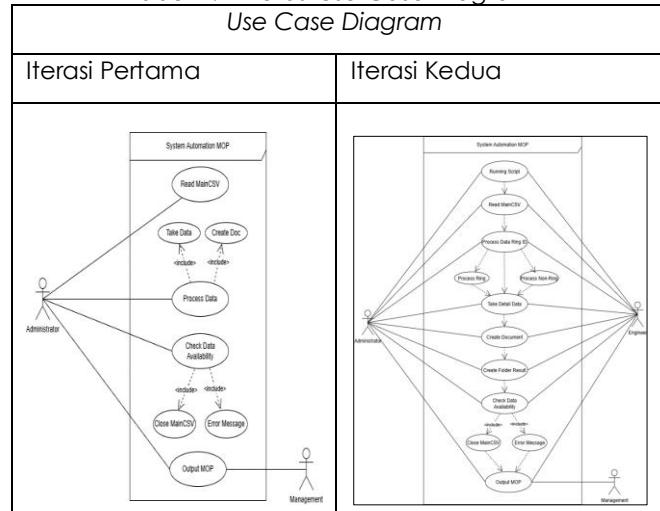
Gambaran rancangan dapat dilihat seperti pada tabel 1.2 dan tabel 1.3

Tabel 1. 1 Iterasi Activity Diagram



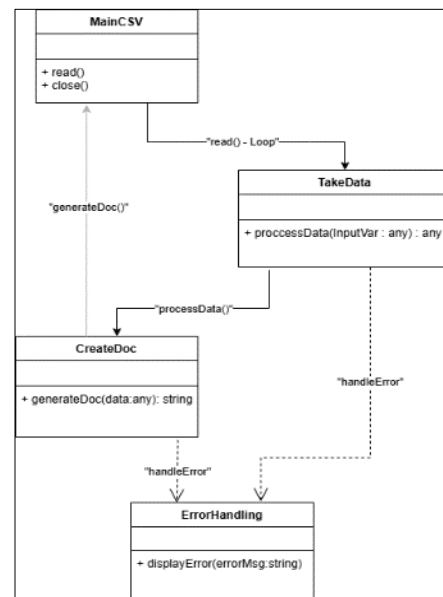
Iterasi pertama dimulai dengan implementasi awal berdasarkan hasil wawancara dan diskusi awal dengan klien yang telah dirangkum dalam perancangan sistem. Pada tahap ini, dibuat prototipe aplikasi sesuai pemahaman awal untuk memberikan gambaran kepada klien. Pada iterasi pertama, activity diagram hanya menggambarkan proses dasar mulai dari input, membaca file CSV, looping untuk mengambil data, membuat dokumen, hingga menutup file CSV dan menghasilkan output dokumen MOP. Sedangkan pada iterasi kedua terdapat penyempurnaan dengan penambahan langkah *Create Folder Result* sebelum penutupan file CSV, sehingga hasil dokumen yang dibuat lebih terorganisasi.

Tabel 1. 2 Iterasi Use Case Diagram
Use Case Diagram



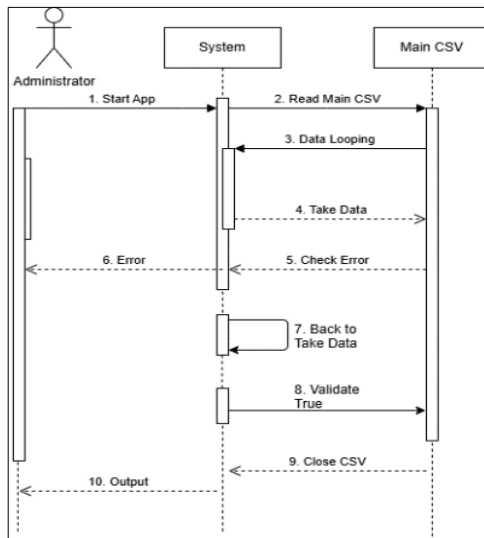
Tabel iterasi 2 memperlihatkan use case diagram final dari aplikasi setelah dua iterasi pengembangan menggunakan metode RAD, di mana fitur-fitur baru telah ditambahkan dan dapat diakses oleh semua aktor dalam sistem, sehingga meningkatkan fungsionalitas dan interaksi sesuai kebutuhan pengguna. Gambar memperlihatkan activity diagram final dari aplikasi setelah dua iterasi pengembangan menggunakan metode RAD, terdapat penambahan fitur pada proses pengembangan sehingga aktivitas dan alur proses yang merepresentasikan fungsi-fungsi tersebut juga harus diperbarui agar sesuai dengan kebutuhan baru.

Class diagram memiliki peran penting dalam menggambarkan struktur sistem otomatisasi pembuatan dokumen *Method of Procedure* (MOP). Diagram ini menunjukkan berbagai kelas yang terlibat dalam sistem, beserta atribut dan metode yang dimiliki oleh masing-masing kelas yang dapat dilihat seperti pada gambar 1.3



Gambar 1. 3 Class Diagram

Sequence diagram berfungsi untuk menggambarkan interaksi dinamis antara objek dalam sistem otomatisasi pembuatan dokumen *Method of Procedure* (MOP). Diagram ini menunjukkan urutan langkah-langkah yang diambil oleh berbagai objek saat mereka berkomunikasi satu sama lain untuk menyelesaikan suatu proses, dapat dilihat seperti pada gambar 1.4



Gambar 1. 4 Sequence Diagram

HASIL DAN PEMBAHASAN

1) Hasil Implementasi

1. Tampilan CLI

Gambar menunjukkan tampilan CLI dimana proses kerja sebuah script Python yang dijalankan untuk membuat dokumen secara otomatis. Pada awal eksekusi, script memberikan informasi bahwa proses pembuatan dokumen telah dimulai.

```

PS I:\My Drive\SKRIPSI\upgrade_timos> python .\script_upgrade_timos.py
C:\Users\PURI\AppData\Local\Programs\Python\Python313\Lib\site-packages\docxcompose\
API. See https://setuptools.pypa.io/en/latest/pkg_resources.html. The pkg_resources
rom using this package or pin to Setuptools<81.
import pkg_resources
>> Start to create a documents....

Document AG-ABCDE_123 has been created!!

Document AG-ABCDE_123 has been created!!

Document AG-ABCDE_123 has been created!!

Document AG-ABCDE_123 has been created!!
  
```

Gambar 1. 5 Tampilan CLI

2. Tampilan Halaman Cover

Gambar menunjukkan hasil dokumen yang dihasilkan oleh proses otomatisasi menggunakan script Python. Ini merupakan halaman pertama yang secara otomatis sistem buat setelah berhasil menjalankan script. Bagian tulisan yang ditebalkan dan diberi warna kuning, yaitu "AT-XYX-CMS-XYZ-AY123" dan tanggal "08 May 2025",

adalah data yang secara otomatis diisi oleh script pada template MOP berdasarkan input atau data yang telah ditentukan sebelumnya melalui integrasi dari Microsoft Excel.

IOH GENERIC MOP METHOD OF PROCEDURE UPGRADE TIMOS 23.9.R3 AT XYZ-CMS-XYZ-AY123			
Prepared by:	Puri Ayu Purwaningsih	Date:	04 Jul, 2025
Reviewer:	Rizky	Date:	04 Jul, 2025
Approver:	Bima	Date:	04 Jul, 2025

Gambar 1. 6 Halaman Cover

3. Tampilan Device Info Upgrade Timos

Gambar menunjukkan hasil dokumen yang dihasilkan secara otomatis oleh script yang telah dibuat. Ini merupakan halaman kedua yang secara otomatis sistem buat setelah berhasil menjalankan script. Tulisan yang ditebalkan dan diberi warna kuning, seperti AT-XYX-CMS-XYZ-AY123, SAS-Sx, dan AG-13CBRU0075-10N merupakan data penting yang secara otomatis dimasukkan ke dalam dokumen berdasarkan input atau data sumber yang sudah disiapkan sebelumnya melalui integrasi dari Microsoft Excel.

1 Description of Upgrade TIMOS at AG-ABCDE_123						
1.1 This MOP cover Plan for Upgrade TIMOS 23.10.R3 Nokia 7750 SR-2e						
1.2 Description of Site						
New Device info:						
No	Hostname	Platform	IP Address	Site Name	Site ID	Varisi.TIMOS
1	AG-13CBRU0075-10N	Nokia 7750 SR-2e	0.0.0.0	SITE_XYZ123	SITE_123ABC	TIMOS-B-10.0.R20

Gambar 1. 7 Tampilan Device Info

4. Tampilan Purpose and Requirement

Gambar menunjukkan bagian *Purpose and Requirements* pada halaman MOP ini menampilkan tabel yang berisi tujuan, ruang lingkup, jadwal, dan estimasi downtime dari aktivitas upgrade perangkat lunak TIMOS 23.9.R3 untuk perangkat Nokia 7210 SAS-Sx yang secara otomatis disikan setelah menjalankan *script automation*.

1.3 Purpose and Requirements	
Change purpose*	This MoP is prepared for Upgrade TIMOS 23.10.R3 for Nokia 7750 SR-2e. We will perform this activity at maintenance window .
Change scope	Upgrade TIMOS 23.10.R3 7750 SR-2e at AG-ABCDE_123
Change time*	2025-05-XX 00:00 - 2025-05-XX 05:00 will be execute at 22:00 WIB and already attached approval exception from VP region and VP Owner
Down Time	30 minutes (Expected Downtime during upgrading TIMOS)

Gambar 1. 8 Tampilan Purpose & Requirement

5. Tampilan Description of Change

Gambar 4.34 menunjukkan hasil automasi dokumen MOP pada tampilan "Description of Change" memberikan penjelasan mengenai tujuan dan pengaruh perubahan yang dilakukan.

NE Name	NE Type MPU Type	Software Version
XYZ-CBR-EN1-SASSx-OPT	7210 SAS- Sx	TIMOS-B-10.0.8.12
XYZ-CKL-EN1-SASSx-OPT	7210 SAS- Sx	TIMOS-B-10.0.8.12
XYZ-CKN-EN1-SASSx-OPT	7210 SAS- Sx	TIMOS-B-10.0.8.12
XYZ-SKR-EN1-SASSx-OPT	7210 SAS- Sx	TIMOS-B-10.0.8.12
XYZ-CKD-EN1-SASSx-OPT	7211 SAS- Sx	TIMOS-B-10.0.8.12

Gambar 1. 9 Tampilan Description of Change

2) Hasil Pengujian

1. Black Box Testing

a. Pengujian Black Box Input dan Pembacaan Data

Tabel 1. 3 Pengujian Input & Read Data

No	Fitur	Skenario Pengujian	Hasil yang diharapkan	Hasil yang diperoleh
1	Pembacaan File CSV Utama	File Main.csv tersedia dengan format dan data valid	File terbaca, perulangan berjalan sesuai jumlah baris	Berhasil
2	Pembacaan File CSV Utama	File Main.csv tidak ditemukan	Muncul pesan error FileNotFound atau pesan terkait	Berhasil
3	Pembacaan File CSV Device	File CSV device (ring_id.csv atau Router_no_n_sassx.csv) tersedia dan valid	Data device berhasil dibaca dan siap diproses	Berhasil
4	Pembacaan File CSV Device	File CSV device tidak ditemukan	Muncul pesan error "File [ring_id].csv not found"	Berhasil
5	Validasi Data Input	Data CSV mengandung kolom kosong atau format salah	Muncul pesan error atau handling error yang sesuai	Berhasil

Total skenario pengujian pada Pengujian Input dan Pembacaan Data= 5 Skenario Skenario berhasil = 5, Skenario gagal = 0, Presentase pengujiannya: $5/5 \times 100\% = 100\%$

b. Pengujian Proses dan Logika Bisnis

Tabel 1. 4 Pengujian Proses & Logika Bisnis

No	Fitur	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian
1	Perulangan Data	CSV berisi 10 baris data	Perulangan berjalan 10 kali, memanggil fungsi takeData sesuai device	Berhasil
2	Pengecekan Device	Device = 7210 SASx	Fungsi takeData dipanggil dengan parameter isRing = True	Berhasil
3	Pengecekan Device	Device selain 7210 SASx	Fungsi takeData dipanggil dengan parameter isRing = False	Berhasil
4	Fungsi hostname NFMP	Device 7210 SASx dan device lain	Menghasilkan hostname sesuai kondisi	Berhasil
5	Fungsi ip AddNFMP	Device 7210 SASx dan device lain	Menghasilkan IP address sesuai kondisi	Gagal
6	Fungsi version timos	Device berbeda	Menghasilkan versi TIMOS yang sesuai	Berhasil
7	Fungsi checkDevice	Device valid dan tidak valid	Menghasilkan mapping device atau error handling	Berhasil

Total skenario pengujian pada Pengujian Proses dan Logika Bisnis= 7 Skenario Skenario berhasil = 6

Skenario gagal = 1

Presentase pengujiannya: $6/7 \times 100\% = 85,71\%$

c. Pengujian Black box Output dan Penanganan Error

Tabel 1. 5 Pengujian Output & Error

No	Fitur	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian
1	Pemilihan Template	Template sesuai device dan event tersedia	Template dokumen berhasil dipilih dan digunakan	Berhasil
2	Pemilihan Template	Template tidak ditemukan	Muncul pesan error "Your input device are not match with any device in List"	Berhasil
3	Pembuatan Folder Output	Folder hasil belum ada	Folder hasil otomatis dibuat	Berhasil
4	Pembuatan Dokumen	Data context lengkap dan valid	Dokumen MOP berhasil dibuat dan disimpan	Berhasil
5	Penanganan Error	Error saat membaca file CSV atau menyimpan dokumen	Pesan error tampil jelas dan proses tidak crash	Berhasil
6	Output Akhir	Semua data di CSV utama	Pesan "Done create the documents!!"	Berhasil

berhasil diproses	tampil setelah semua dokumen dibuat
-------------------	-------------------------------------

Total skenario pengujian pada Output dan Penanganan Error = 6 Skenario Skenario berhasil = 6 Skenario gagal = 0

Presentase pengujiannya: $6/6 \times 100\% = 100\%$

$$x = \frac{285,71\%}{3} \times 100\% = 95,23\%$$

Jadi hasil presentase Black box integrasi pada iterasi pertama secara keseluruhan adalah 95,23%.

Pada pengujian proses dan logika bisnis terdapat iterasi kedua sebagai perbaikan pada pengujian yang terdapat di tabel sehingga pengujian yang diuraikan pada table.

d. Pengujian Black box Proses dan Logika Bisnis

Tabel 1. 6 Pengujian Proses & Logika Bisnis

No	Fitur	Skenario Pengujian	Hasil yang diharapkan	Hasil yang diperoleh
1	Perulangan Data	CSV berisi 10 baris data	Perulangan berjalan 10 kali, memanggil fungsi takeData sesuai device	Berhasil
2	Pengecekan Device	Device = 7210 SASSx	Fungsi takeData dipanggil dengan parameter isRing=True	Berhasil
3	Pengecekan Device	Device selain 7210 SASSx	Fungsi takeData dipanggil dengan parameter isRing=False	Berhasil
4	Fungsi hostnameNFM P	Device 7210 SASSx dan device lain	Menghasilkan hostname sesuai kondisi	Berhasil
5	Fungsi ipAddressNFM P	Device 7210 SASSx dan device lain	Menghasilkan IP address sesuai kondisi	Berhasil
6	Fungsi versi ontimos	Device berbeda	Menghasilkan versi TiMOS yang sesuai	Berhasil
7	Fungsi checkDevice	Device valid dan tidak valid	Menghasilkan mapping device atau error handling	Berhasil

Total skenario pengujian pada Pengujian Proses dan Logika Bisnis= 7 Skenario, Skenario berhasil = 7, Skenario gagal = 0

Presentase pengujiannya: $7/7 \times 100\% = 100\%$

Maka setelah proses pengujian yang telah dilakukan pada iterasi pertama dan iterasi kedua hasil presentase pengujian black box secara keseluruhan adalah 100%.

2. Validasi Ahli IT

Tabel 1. 7 Validasi IT

Skor Kecapaian	Hasil Skor	Total (Skala * Hasil Skor)
1	0	0
2	0	0
3	0	0
4	5	20
5	35	175
Total Skor		195
Skor Maksimum		200
(Jumlah Responden * Jumlah Butir Soal * Skala Tertinggi)		

Hasil dari perhitungan tabulasi ahli IT pada tabel diatas dimasukan kedalam rumus kelayakan. Berikut perhitungan rumus kelayakan sistem automasi MOP menggunakan bahasa pemrograman Python.

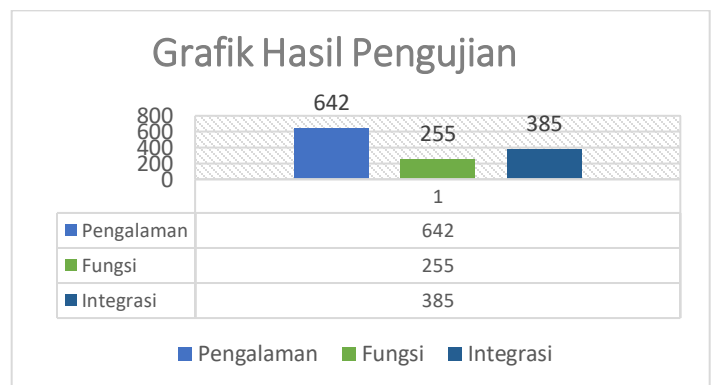
$$\text{Presentase kelayakan} = \frac{195}{200} \times 100\% = 97.5\%$$

3. User Acceptance Testing (UAT)

Tabel 1. 8 Pengujian UAT

No	Indikator	Presentase	Keterangan
1	Kemudahan Sistem	85,60%	Sangat Setuju
2	Fungsi Automasi MOP	85%	Sangat Setuju
3	Integrasi Automasi	85,50%	Sangat Setuju

Berdasarkan pengujian UAT, dapat disimpulkan bahwa penerimaan sistem automasi MOP Upgrade Timos dinilai sangat baik oleh responden. Indikator Kemudahan Sistem memperoleh persentase tertinggi sebesar 85,60%, diikuti oleh Integrasi Automasi sebesar 85,50%, dan Fungsi Automasi MOP sebesar 85%. Seluruh aspek tersebut mendapatkan keterangan "Sangat Setuju", yang menunjukkan bahwa sistem ini telah diterima dan berjalan dengan sangat baik menurut penilaian pengguna.



Gambar 1. 10 Grafik Pengujian

KESIMPULAN DAN SARAN

Sistem automasi Method of Procedure (MOP) Upgrade Timos yang dirancang menggunakan bahasa pemrograman Python berbasis CLI berhasil

mengotomatisasi proses pembuatan dokumen MOP dengan memanfaatkan modul csv dan docxtpl, serta pengelolaan data dan folder yang dinamis sesuai kebutuhan operasional jaringan. Hasil pengujian menunjukkan bahwa aplikasi automasi MOP pada pengujian black box menunjukkan tingkat keberhasilan aplikasi secara keseluruhan mencapai 95,23% pada pengujian pertama, dan terjadi pada iterasi kedua sudah mencapai 100%, validasi oleh ahli IT memberikan hasil yang sedikit lebih tinggi, yaitu 97,5%, dan User Acceptance Testing (UAT) memiliki tingkat penerimaan yang baik dari pengguna dengan indikator kemudahan sistem mencapai 85,60%, integrasi automasi 85,50%, dan fungsi automasi MOP sebesar 85%, menandakan aplikasi siap digunakan dan memberikan nilai tambah dalam proses upgrade Timos.

DAFTAR PUSTAKA

- [1] F. C. Ivo Pinto, *Automating and Orchestrating Networks with NetDevOps*. Pearson Education, 2023.
- [2] M. Y. Prabowo, A. Firdaus, D. A. Damanik, and T. Dedy, "Automasi Pembuatan Laporan Analisis dengan R Markdown," *Jurnal Studi Pemerintahan*, vol. 5, no. 1, pp. 49–61, 2022, [Online]. Available: <https://www.jurnal-umbuton.ac.id/index.php/Kybernan/article/view/2121>
- [3] M. K. A. Purbaningtyas and Y. A. Susetyo, "Implementasi Python-PPTX dan Dom-to-Image dalam Pembangunan Sistem Otomatisasi Laporan Berbasis Web di PT. XYZ," *Jurnal Indonesia: Manajemen Informatika dan Komunikasi*, vol. 5, no. 2, pp. 1373–1382, 2024, doi: 10.35870/jimik.v5i2.684.
- [4] S. McConnell, *Rapid Development*, 1st ed. Microsoft Press, 1996.
- [5] Y. J. M. B.R Mehta, *Industrial Process Automation System*. United States of America: Elsevier, 2015.
- [6] D. Mikelsten, *Otomasi dan Teknologi Berkembang*. Cambridge Stanford Books, 2019.
- [7] P. M. Curtis, *Maintaining Mission Critical Systems in a 24/7 Environment*, 1st ed. Canada: Wiley, 2011.
- [8] N. G. Shaw, *Strategies for Managing Computer Software Upgrades*, 1st ed. USA: IGI Global, 2000.
- [9] Jajang & Sumliyah, *Pemograman Python*, 1st ed. PT. Literasi Nusantara Abadi, 2024.
- [10] A. Kadir, *Dasar Pemograman Python 3*, 1st ed. Yogyakarta: Andi, 2018.
- [11] J. P. Mueller, *Windows Administration at the Command Line for Windows 2003, Windows XP, and Windows 2000*. John Wiley & Sons, 2006.
- [12] K. E. Kendall, *Systems Analysis and Design*. Pearson Prentice Hall, 2001.
- [13] A. Agrawal, *Enterprise Automation with Python*. BPB Publications, 2022.
- [14] K. A. Saleh, *Software Engineering*. J. Ross Pub., 2009.
- [15] R. Cimperman, *UAT Defined A Guide to Practical User Acceptance Testing (Digital Short Cut)*. Boston: Pearson Education, 2006.
- [16] A. Engel, *Verification, Validation, and Testing of Engineered Systems*. Wiley, 2010.