

Pengembangan Antarmuka Pengguna Website Sistem Otomasi Konfigurasi dan Monitoring Perangkat Jaringan Multivendor

1st Jesseline Regina Manalu

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

jesselineregina@student.telkomuniversity.ac.id

2nd Favian Dewanta

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

favian@telkomuniversity.ac.id

3rd Bagus Aditya

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

goesaditya@telkomuniversity.ac.id

Abstrak—Dalam konteks penggunaan perangkat jaringan dari berbagai vendor, perusahaan dan penyedia layanan internet (ISP) sering menghadapi tantangan terkait kompatibilitas dan biaya. Setiap vendor biasanya memiliki sistem dan protokol yang berbeda, memaksa perusahaan untuk mengadopsi seluruh rangkaian perangkat dari vendor yang sama. Untuk mengatasi masalah ini, penelitian ini mengembangkan sebuah situs web yang memungkinkan perusahaan untuk mengelola perangkat jaringan dari berbagai vendor melalui otomasi konfigurasi dan monitoring. Situs web ini dirancang untuk menyediakan alat otomatisasi yang mendukung berbagai vendor dengan menggunakan protokol standar seperti SSH dan SNMP, yang umumnya tersedia pada semua perangkat jaringan. Implementasi sistem dilakukan dengan menggunakan Python, yang memiliki library seperti Paramiko dan Netmiko untuk konfigurasi perangkat multi-vendor melalui protokol SSH. Website ini memberikan fleksibilitas kepada perusahaan dalam memilih dan mengelola perangkat jaringan sesuai kebutuhan dan anggaran mereka, tanpa terikat pada satu vendor. Alat ini juga membantu mengatasi masalah kompatibilitas dan protokol proprietary, mempermudah konfigurasi dan monitoring perangkat, serta meningkatkan efisiensi operasional.

Kata kunci— Konfigurasi, Monitoring, Otomasi, Python

I. PENDAHULUAN

Saat ini, banyak perusahaan dan penyedia layanan internet (ISP) menghadapi tantangan dalam menggunakan perangkat jaringan dari berbagai vendor yang berbeda. Hal ini memberatkan perusahaan secara finansial [1], karena setiap vendor memiliki keunggulannya masing-masing yang mengharuskan perusahaan untuk menggunakan seluruh rangkaian perangkat dari vendor tersebut.

Untuk mengatasi masalah ini, penulis telah mengembangkan sebuah situs *website* yang memungkinkan perusahaan menggunakan perangkat jaringan dari berbagai vendor [2] sesuai dengan kebutuhan mereka untuk melakukan otomasi konfigurasi dan *monitoring*. *Website* ini dibuat dengan tujuan untuk menyediakan alat otomatisasi yang mendukung berbagai vendor perangkat jaringan. Sistem ini dibangun menggunakan protokol SSH (RFC 4253) dan SNMP (RFC 1157), yang merupakan protokol standar yang dimiliki oleh semua vendor, sehingga batasan antar vendor dapat diatasi. Penelitian sebelumnya menunjukkan bahwa

Python telah diimplementasikan dalam otomasi jaringan karena memiliki berbagai *library* seperti Paramiko dan Netmiko [3] yang mendukung konfigurasi perangkat jaringan multi-vendor menggunakan protokol SSH.

Website ini dirancang untuk memberikan fleksibilitas lebih kepada perusahaan dalam memilih dan mengelola perangkat jaringan dari berbagai vendor, sehingga mereka dapat mengoptimalkan infrastruktur jaringan mereka sesuai dengan kebutuhan spesifik dan anggaran yang dimiliki, tanpa harus terikat kontrak dengan satu vendor untuk seluruh sistemnya. Juga terbantu dalam memecahkan masalah yang selama ini terhambat dikarenakan kompatibilitas atau protokol *proprietary* dari setiap vendor dan dengan alat ini tenaga kerja pada ISP atau perusahaan tersebut dapat terbantu dalam konfigurasi dan *monitoring* perangkat.

II. KAJIAN TEORI

Django telah menjadi platform yang paling banyak digunakan kalangan pengembang web karena memudahkan mereka untuk membangun aplikasi web yang dinamis, aman, dan efisien [4]. Meskipun Django pada dasarnya adalah kerangka kerja backend, ia juga dapat digunakan untuk membangun frontend dengan menggabungkan templat HTML, CSS, dan JavaScript. Django, sebagai kerangka kerja web berbasis Python, mendukung pengembang dalam menciptakan aplikasi web dengan cepat. Kerangka kerja ini menyediakan berbagai alat penting seperti sistem routing, manajemen basis data dan sistem templat [5]. Integrasi FrontEnd pada Django dioperasikan sebagai berikut.

A. Templat

Django menggunakan alat templating yang menjadi pemisah antara kode HTML dan logika Python. Pengembang dapat membuat templat yang mengandung placeholder untuk data dinamis, yang kemudian dapat diambil dari view dan dimasukkan ke dalam templat tersebut. Pemisahan ini membuat kode lebih mudah untuk dipelihara, sehingga pengembang frontend dapat fokus hanya pada frontend tanpa perlu memikirkan cara kerja kode backend.

B. Data Konteks

Pada Django, data konteks dari tampilan ditransmisikan ke templat. Data ini bisa berupa variabel, daftar atau objek

yang digunakan untuk menggabungkan data dalam templat HTML. Ini memungkinkan untuk menampilkan basis data [6], input formulir, dan berbagai data lain yang diperlukan untuk antarmuka pengguna.

C. File Statis

Untuk menyajikan tampilan web yang menarik dan responsif, Django menyediakan mekanisme pengelolaan file statis yang efisien. Dengan menyimpan file-file seperti gambar, CSS, dan JavaScript di direktori khusus, Django memastikan bahwa file-file ini dapat diakses dengan cepat oleh browser, sehingga meningkatkan kecepatan loading halaman web dan pengalaman pengguna secara keseluruhan.

D. FrontEnd Frameworks

Django sebagai kerangka kerja backend bekerja sama dengan kerangka kerja frontend seperti React, Vue.js atau Angular. Dengan kombinasi ini, pengembang bisa membuat aplikasi web modern yang dinamis dan interaktif. Bagian frontend akan berkomunikasi dengan bagian *backend* melalui API (antarmuka pemrograman aplikasi) untuk mengambil dan mengirim data.

Berikut merupakan Gambar 1 yang menampilkan *source code* yang mengelola data dari *Back-end* dan logika untuk navigasi data antara pengguna dan *Back-end*, *source code* pada *front-end* berisi *handler* untuk API dan HTML, Framework *front-end* yang digunakan masih sama dengan *back-end* yaitu framework *django*.

```

from django.shortcuts import render
from django.views import View
from django.contrib.auth.decorators import login_required
from django.utils.decorators import method_decorator
from datetime import datetime
import paramiko
import time
import requests

from django.shortcuts import render, redirect
from django.views import View
from django.contrib.auth import authenticate, login
from django.urls import reverse

class HomePageDataProviderImpl: ...
class LoginView(View): ...
    @method_decorator(login_required, name='dispatch')
class LogoutView(View): ...
    @method_decorator(login_required, name='dispatch')
class home(View): ...
    @method_decorator(login_required, name='dispatch')
class devices(View): ...
    @method_decorator(login_required, name='dispatch')
class ConfigureView(View): ...
    @method_decorator(login_required, name='dispatch')
class VerifyConfigView(View): ...
    @method_decorator(login_required, name='dispatch')
class log(View): ...
def error_404_view(request, exception): ...

```

Gambar 1
Framework Django

E. FrontEnd Library

Django sangat fleksibel sehingga kita bisa dengan mudah menambahkan perpustakaan-perpustakaan populer seperti jQuery, Bootstrap atau D3.js untuk membuat tampilan web menjadi lebih menarik dan interaktif. Perpustakaan-

perpustakaan ini menyediakan berbagai macam komponen siap pakai yang bisa kita gunakan.

F. Pengujian *FrontEnd*

Kerangka kerja pengujian yang dimiliki Django memungkinkan pengembang untuk melakukan pengujian unit pada komponen frontend aplikasi. Hal ini sangat penting untuk menjaga kualitas aplikasi dan memastikan bahwa perubahan kode tidak menimbulkan masalah baru.

III. METODE

A. Konsep Antarmuka Website Otomasi Konfigurasi dan Monitoring

Dalam rangka pengembangan antarmuka pengguna website otomasi konfigurasi dan monitoring ini, dapat digunakan beberapa metode diantaranya:

1. Studi Pustaka

Studi pustaka adalah tinjauan terhadap jurnal-jurnal yang relevan dengan masalah yang sedang diteliti. Jurnal-jurnal ini memberikan wawasan mendalam dan dapat membantu memahami konteks permasalahan yang diteliti.

2. Desain Sistem

Dalam penelitian ini, metode perancangan sistem digunakan untuk mengembangkan solusi yang efektif dan efisien sesuai dengan kebutuhan pengguna [7]. Proses perancangan sistem ini melibatkan beberapa tahapan utama sebagai berikut:

a. Analisis Kebutuhan

Dalam pengembangan sebuah *website* sistem otomasi konfigurasi dan monitoring perangkat jaringan multivendor dengan arsitektur *microservice*, perlu dipertimbangkan dalam menggunakan perangkat lunak yang diperlukan, seperti *Visual Studio Code* yang digunakan untuk pengembangan antar muka website tersebut. Dengan memilih perangkat lunak yang tepat akan mendukung proses pengembangan antarmuka *website* secara efektif.

B. Implementasi Detail Sistem

Proses perancangan *website* untuk otomasi jaringan menggunakan framework Django dengan pemrograman Python. Django dibangun menggunakan bahasa pemrograman Python, dengan kode sumber yang terstruktur dan mudah dipahami. Django dirancang khusus untuk mempermudah pengembang dalam menciptakan tampilan web yang interaktif dan menarik secara cepat dan efisien [8].

1. Halaman *Login*

Halaman *login* akan diimplementasikan dengan pengembangan sebagai berikut:

- Logo website: Agar tampilan website lebih menarik.
- Kotak Teks: Terdapat dua kolom utama yang sangat penting. Kolom-kolom ini memiliki label yang jelas dan field yang cukup besar untuk memudahkan pengguna dalam mengetikkan Username dan Password.
- Tombol Login: Didesain mencolok agar mudah terlihat dan diakses. Ukuran tombol yang tepat mempermudah pengguna dalam mengkliknya.

2. Halaman *Add Device* dan *Add Account*

Halaman ini dibuat dengan framework Django karena memiliki desain yang minimalis dengan focus utama pada pengelolaan data dalam database. Struktur navigasi pada admin Django secara otomatis dihasilkan berdasarkan model-model yang telah ditentukan dalam aplikasi. Fitur Tambah, Edit, dan Hapus: Simbol tambah (+) di samping setiap item menu menunjukkan adanya fitur untuk menambahkan data baru, mengedit data yang sudah ada dan menghapus data. Fitur ini adalah bagian standar dari admin Django.

Kotak pencarian dan opsi "Select device to change" menunjukkan adanya fitur untuk mencari dan memfilter data, yang merupakan fitur umum di admin Django. Checkbox untuk Seleksi Berganda: Checkbox di samping setiap item data memungkinkan pengguna untuk memilih beberapa item sekaligus, misalnya untuk melakukan tindakan massal seperti menghapus atau mengubah status secara bersamaan. Ini juga merupakan fitur standar di admin Django.

3. Halaman *Configure*

Halaman *Configure* akan diimplementasikan dengan pengembangan sebagai berikut:

- Checkbox:** Memungkinkan pengguna untuk memilih atau membatalkan pilihan satu, beberapa, atau semua perangkat dari daftar yang tersedia. Setiap target dalam daftar mencantumkan nama dan alamat IP perangkat jaringan yang akan dikonfigurasi.
- Daftar *Dropdown*:** Digunakan untuk memilih jenis vendor perangkat jaringan yang akan dikonfigurasi.
- Kotak Teks:** Berfungsi untuk memasukkan perintah konfigurasi yang akan dikirim ke perangkat yang telah dipilih.
- Tombol *Submit*:** Digunakan untuk mengirimkan perintah konfigurasi yang telah dimasukkan ke perangkat yang dipilih.

4. Halaman *Verify Config*

Halaman *Verify Config* akan diimplementasikan dengan pengembangan sebagai berikut:

- Checkbox:** Pengguna dapat memilih satu atau lebih perangkat jaringan yang ingin di verify config dengan mencentang checkbox di samping nama perangkat. Setiap perangkat yang terdaftar memiliki nama dan alamat IP yang berbeda.
- Kotak *Dropdown*:** Tersedia kotak dropdown untuk memilih jenis vendor perangkat jaringan yang ingin dikonfigurasi dari daftar yang ada.
- Kotak Teks:** Pengguna dapat menuliskan perintah konfigurasi spesifik untuk perangkat yang dipilih pada kotak teks yang disediakan.
- Tombol *Submit*:** Setelah perintah konfigurasi diketik, pengguna dapat menekan tombol Submit untuk mengirimkan perintah tersebut ke semua perangkat yang telah dipilih.

5. Halaman *Automation Template*

Halaman *Automation Template* akan diimplementasikan dengan pengembangan sebagai berikut:

- Tombol *Creat New Script*:** Digunakan untuk membuat skrip otomatis baru. Skrip ini berisi perintah-perintah yang akan dijalankan pada perangkat-perangkat yang dipilih.
- Checkbox:** Pengguna dapat memilih satu atau lebih perangkat jaringan yang ingin diotomasi dengan mencentang checkbox di samping nama perangkat. Bagian

ini berisi daftar perangkat atau kelompok perangkat yang akan menjadi target otomasi. Setiap target memiliki alamat IP dan deskripsi singkat.

- Kotak Teks:** Digunakan untuk memberikan nama pada skrip otomasi yang baru dibuat.
- Simpan Otomatisasi**
- Tombol *Save Automation*:** Digunakan untuk menyimpan skrip otomatis yang baru dibuat.
- Tombol *Submit*:** Digunakan untuk menjalankan skrip otomasi yang telah dipilih atau yang baru dibuat pada perangkat-perangkat yang telah dipilih.
- Save Automation*:** Bagian ini menampilkan daftar skrip otomatis yang telah disimpan sebelumnya. Untuk setiap skrip, ditampilkan informasi seperti nama skrip, perangkat target, dan tombol untuk mengedit atau menghapus skrip tersebut.
- Tombol *Trigger*:** Digunakan untuk melakukan trigger pada otomasi yang telah disimpan sebelumnya.
- Tombol *Delete*:** Digunakan untuk menghapus otomasi.

6. Halaman *Monitoring*

Halaman *Monitoring* akan diimplementasikan dengan pengembangan sebagai berikut:

- Kotak Teks:** Digunakan untuk memasukkan alamat IP, Hostname dan Vendor dari perangkat yang ingin monitoring.
- Tombol *Add*:** Tombol ini digunakan untuk menambahkan perangkat yang telah dikonfigurasi ke dalam daftar perangkat yang dimonitoring.
- Kotak *Dropdown*:** Digunakan untuk menampilkan perangkat vendor sesuai dengan yang ingin dilihat.

7. Halaman *Grafana-Log*

Halaman *Grafana-Log* akan diimplementasikan dengan pengembangan sebagai berikut:

Tabel: Digunakan untuk menampilkan tanggal dan waktu, Jenis peringatan yang terjadi, proses yang memicu peringatan, nama antarmuka jaringan yang sedang dimonitoring dan status akhir dari keadaan perangkat.

8. Halaman *Ping-Log*

Halaman *Ping-Log* akan diimplementasikan dengan pengembangan sebagai berikut:

Tabel: Digunakan untuk menampilkan alamat IP dari masing-masing perangkat yang sedang dimonitoring, serta menunjukkan status ketersediaan perangkat dan waktu terakhir kali status perangkat diperiksa oleh sistem.

9. Halaman *Konfigurasi-Log*

Halaman *Konfigurasi-Log* akan diimplementasikan dengan pengembangan sebagai berikut:

Tabel: Berfungsi untuk menampilkan target, tindakan yang dilakukan terhadap perangkat, serta Hasil, Waktu dan Pesan tambahan terkait dengan tindakan yang dilakukan.

IV. HASIL DAN PEMBAHASAN

Setelah melewati tahap perancangan yang mendalam, pengujian, yang dilakukan setelah pengembangan aplikasi, bertujuan untuk memverifikasi apakah aplikasi yang telah dibuat berfungsi dengan baik atau tidak sebelum dirilis kepada pengguna. Pengujian dilakukan dengan menguji

fungsionalitas aplikasi, jika pengujian berjalan lancar, maka aplikasi dinyatakan berfungsi dengan baik, namun jika tidak, aplikasi dianggap tidak berfungsi dengan baik [9]. berikut adalah hasil implementasi media pembelajaran dalam bentuk *website*. *Website* ini dirancang dengan antarmuka yang intuitif dan *user-friendly*, sehingga memudahkan pengguna dalam mengaksesnya. *Framework front-end* yang digunakan yaitu *framework Django* dan menggunakan *Visual Studio Code* untuk mengembangkan tampilan dari *website* yang dibuat.

1. Tampilan Halaman *Login*



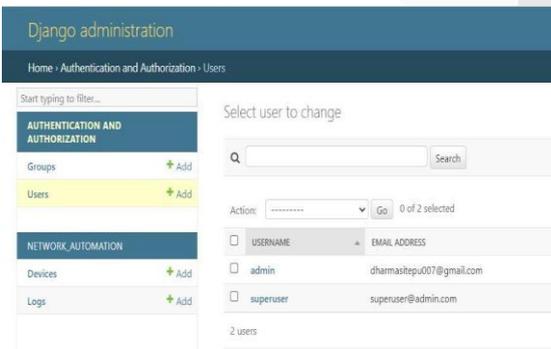
GAMBAR 2
Tampilan Login

Pada Gambar 2, sebelum dapat mengakses konten *website*, kita harus melakukan login terlebih dahulu dengan memasukkan *username* dan *password*. Untuk login, gunakan default user 'superuser' dan *password* '123super'. Setelah itu, pengguna akan diarahkan ke halaman berikutnya.

2. Tampilan *Add User* dan *Add account*



(a) Add Device

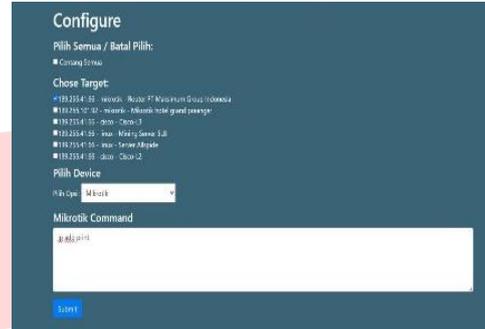


(b) Add Account

GAMBAR 3
Tampilan (a)Add Device (b)Add Account

Halaman *Add Device* dan *Add Account*, seperti yang ditunjukkan pada Gambar 3, berfungsi untuk menambahkan pengguna atau perangkat jaringan. Pada menu *Add Device*, pengguna memasukkan alamat IP perangkat yang akan dikonfigurasi dan dimonitor secara otomatis. Dalam menu ini, pengguna juga dapat menambahkan beberapa perangkat sekaligus. Sementara itu, pada menu *Add Account*, pengguna dapat menambahkan akun baru yang akan memiliki akses ke *website* ini.

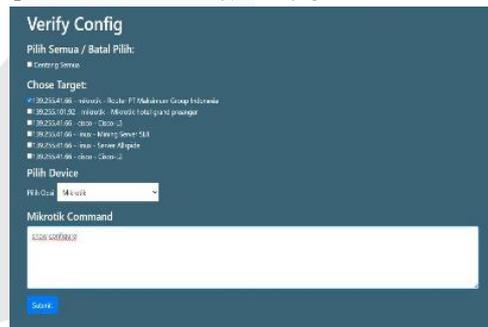
3. Tampilan Halaman *Configure*



GAMBAR 4
Tampilan Halaman Configure

Halaman *Configure* seperti pada Gambar 4 berfungsi untuk melakukan konfigurasi perangkat jaringan, baik secara individual maupun massal. Untuk menggunakan menu ini, pengguna memilih target yang diinginkan, lalu memilih perangkat dari target yang telah dipilih sebelumnya. Pengguna dapat memilih lebih dari satu target dengan perangkat dari berbagai vendor. Setelah itu, pengguna memasukkan perintah sesuai dengan vendor perangkat yang telah dipilih. Selanjutnya, pengguna menekan tombol *submit* untuk memulai proses konfigurasi.

4. Tampilan Halaman *Verify Config*



Gambar 5
Tampilan Halaman Verify Config

Halaman *Verify Config* digunakan untuk menerima respon balik dari perangkat jaringan atau untuk melakukan backup konfigurasi, baik secara individual maupun massal. Untuk menggunakan menu *Verify Config* seperti pada Gambar 5, pengguna harus memilih target yang diinginkan, kemudian memilih perangkat dari target yang telah dipilih. Pengguna dapat memilih lebih dari satu target dengan perangkat dari berbagai vendor. Setelah itu, pengguna memasukkan perintah yang sesuai dengan vendor perangkat yang telah dipilih. Kemudian, pengguna menekan tombol *submit* dan proses verifikasi konfigurasi pun akan dimulai.

5. Tampilan Halaman *Automation Template*



GAMBAR 6
Tampilan Halaman Automation Template

Halaman *Automation Template* digunakan untuk menambahkan skrip baru dan mengotomatisasikan perangkat yang dipilih. Untuk menggunakan menu *Automation Template*, pengguna terlebih dahulu membuat skrip baru melalui opsi 'create new script', seperti yang terlihat pada Gambar 6. Setelah itu, pengguna memilih target yang diinginkan, kemudian memilih perangkat dari target yang telah dipilih sebelumnya. Pengguna dapat memilih lebih dari satu target dengan perangkat dari berbagai vendor. Selanjutnya, pengguna membuat skrip otomatisasi yang diinginkan. Setelah skrip dibuat, pengguna menekan tombol submit, dan template otomatisasi akan mulai berjalan.

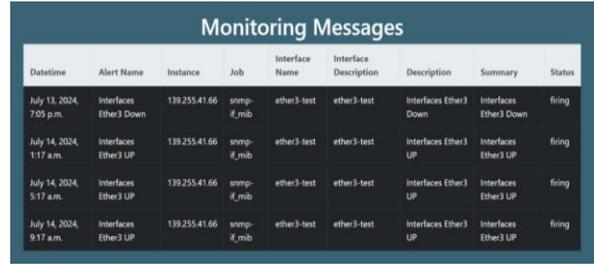
6. Tampilan Halaman *Monitoring*



GAMBAR 7
Tampilan Halaman Monitoring

Halaman *Monitoring* seperti yang ditunjukkan pada Gambar 7 digunakan untuk menambahkan dan memilih alamat IP, hostname, serta vendor perangkat yang akan dimonitoring. Untuk menggunakan menu Monitoring, pengguna terlebih dahulu memasukkan alamat IP, hostname, dan vendor perangkat. Setelah itu, pengguna mengklik tombol add, dan perangkat yang akan dimonitoring akan tersimpan. Perangkat tersebut juga akan muncul pada halaman monitoring.

7. Tampilan Halaman *Grafana-Log*



GAMBAR 8
Tampilan Halaman Grafana-Log

Gambar 8 menunjukkan tampilan Halaman *Grafana-Log*. Pada menu ini, kita dapat melihat data yang diperoleh dari Grafana, yang menggunakan koneksi Prometheus dan SNMP-Exporter. Data tersebut dikirimkan melalui webhook ke *backend*, yang kemudian terhubung dengan *frontend* sehingga data dapat ditampilkan dalam fitur ini.

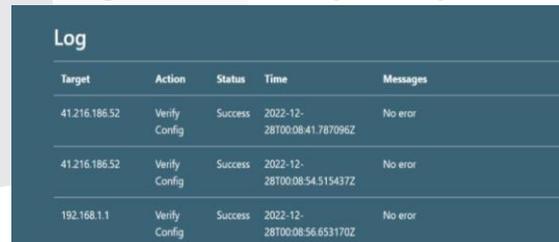
8. Tampilan Halaman *Ping-Log*



GAMBAR 9
Tampilan halaman Ping-Log

Gambar 9 menampilkan halaman *Ping-Log*. Pada halaman ini, kita dapat melihat status perangkat yang telah diotomatisasi, untuk mengetahui apakah perangkat tersebut mengalami downtime atau tidak. Jika Status Up artinya perangkat sedang aktif dan dapat diakses dan jika Status Down maka sebaliknya. Data yang ditampilkan pada menu Ping-Log ini diperoleh dengan menggunakan bash script.

9. Tampilan Halaman Konfigurasi-Log



GAMBAR 10
Tampilan halaman Konfigurasi-Log

Selanjutnya adalah halaman Konfigurasi-Log. Pada halaman ini, aktivitas yang terjadi saat kita menggunakan fitur *Configure*, *Verify Config* dan *Automation Template* [10] ditampilkan. Seperti yang terlihat pada Gambar 10, pengguna dapat memeriksa apakah ada kesalahan atau error pada fitur yang telah dijalankan.

V. KESIMPULAN

Penelitian ini, telah berhasil mengembangkan sebuah sistem otomatisasi konfigurasi dan monitoring perangkat jaringan multivendor menggunakan arsitektur *microservices* berbasis *website* dengan antarmuka yang intuitif dan ramah pengguna. Proses pengembangan mencakup pembuatan antarmuka yang intuitif dan fitur-fitur yang berfungsi sesuai dengan tujuannya, sehingga memudahkan pengguna dalam melakukan otomatisasi pada perangkat jaringan.

Pengujian yang dilakukan meliputi pemeriksaan fungsionalitas fitur serta evaluasi keseluruhan alur pengguna dari awal hingga akhir. Dengan melakukan pengujian yang menyeluruh, pengembang dapat mengidentifikasi dan memperbaiki masalah sebelum *website* digunakan oleh pengguna. Hal ini menunjukkan bahwa pengembangan antarmuka telah sesuai dengan rancangan yang direncanakan.

REFERENSI

- [1] M. Muchamad, "Network Automation Menggunakan Bahasa Pemrograman Python," *Jurnal Teknik Industri, Sistem Informasi dan Teknik Informatika*, vol. 1, no. 2, pp. 120-122, 2022.
- [2] S. Nugroho and B. Pujiarto, "Network Automation Pada Beberapa Perangkat Router Menggunakan Pemrograman Python.," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 9, no. 1, pp. 79-86, 2022.
- [3] B. Choi, *Python Network Automation Labs : SSH Paramiko and Netmiko*, Australia: Apress, Berkeley, CA, 2021, pp. 583-628.
- [4] S. Chen, S. Ahmmed, K. Lal and C. Deming, "Django Web Development Framework: Powering The Modern Web," *American Journal of Trade and Policy*, vol. 7, no. 3, pp. 99-106, 2020.
- [5] I.-k. donors, "The web framework for perfectionists with deadlines.," Django Software Foundation, 2005-2024. [Online]. Available: <https://www.djangoproject.com/>. [Accessed 30 August 2024].
- [6] J. Forcier, P. Bissex and W. Chun, *Python Web Development with Django*, America: Addison-Wesley Professional, 2008.
- [7] B. Fachri and R. W. Surbakti, "Perancangan Sistem Dan Desain Undangan Digital Menggunakan Metode Waterfall Berbasis Website," *Journal of Science and Social Research*, vol. 4, no. 3, pp. 263-267, 2021.
- [8] A. Chandiramani and P. Singh, "Management of Django Web Development in Python," *Journal of Management and Service Science*, vol. 1, no. 2, pp. 1-17, 2021.
- [9] Y. C. E. Paksi and I. R. Widiyari, "Website Network Automation Design And Implementation In Rt Rw Net Senden Dusun Magelang With Django Framework," *Jurnal Teknik Informatika*, vol. 3, no. 5, pp. 1313-1322, 2022.
- [10] P. Mihaila, T. Balan, R. Curpen and F. Sandu, "Network Automation and Abstraction using Python Programming Methods," *MACRo 2015*, vol. 2, no. 1, pp. 95-113, 2017.