

NETRO: NETWORK INVENTORY MANAGEMENT SYSTEM BERBASIS TELEGRAM BOT PADA INFRASTRUKTUR TELKOM INDONESIA REGIONAL 4

Andrean Atma Putra, Astrid Novita Putri

Teknik Informatika, Universitas Semarang

Jalan Soekarno Hatta, Tlogosari Kulon, Kec. Pedurungan, Kota Semarang, Jawa Tengah 50196

andreanatma@gmail.com

ABSTRAK

Pengelolaan inventaris perangkat jaringan merupakan aspek vital dalam menjaga keberlangsungan operasional telekomunikasi. Namun, PT Telkom Indonesia Regional 4 masih menghadapi kendala efisiensi akses data dan akurasi akibat pencatatan manual pada perangkat dinamis seperti *Small Form-factor Pluggable* (SFP), *Line Processing Unit Framework* (LPUF), dan *Card*. Penelitian ini bertujuan untuk merancang dan mengimplementasikan NETRO, sebuah sistem manajemen inventaris berbasis Telegram Bot menggunakan bahasa pemrograman Python dan basis data SQLite. Metode penelitian yang digunakan adalah model pengembangan *Waterfall* dengan pendekatan sistem modular yang mengintegrasikan fitur otomatisasi penomoran aset (*auto-sequencing*), *Role-Based Access Control* (RBAC), serta manajemen log aktivitas secara asinkron. Hasil penelitian menunjukkan bahwa NETRO berhasil mengotomatisasi standarisasi penamaan aset dengan akurasi tinggi dan mempercepat proses pelaporan inventaris secara signifikan dibandingkan metode konvensional. Implementasi fitur *audit logging* memastikan setiap perubahan data terdokumentasi secara transparan, sehingga meningkatkan akuntabilitas dan efisiensi manajemen infrastruktur jaringan di lingkungan Arnet Semarang.

Kata kunci : Inventaris Jaringan; Telegram Bot; Waterfall; RBAC; Audit Logging

1. PENDAHULUAN

Perkembangan infrastruktur telekomunikasi di Indonesia, khususnya pada PT. Telkom Indonesia Regional 4, telah mencapai titik kompleksitas yang sangat tinggi seiring dengan meningkatnya kebutuhan layanan pita lebar (*broadband*) berbasis serat optik. Sebagai tulang punggung layanan teknologi informasi, pengelolaan aset jaringan atau *Network Inventory Management* menjadi aspek kritical dalam menjaga keberlangsungan operasional bisnis. Infrastruktur pada Arnet Semarang mencakup ribuan unit perangkat optik aktif seperti *Small Form-factor Pluggable* (SFP), *Line Processing Unit Framework* (LPUF), dan *Card* yang tersebar di berbagai lokasi atau *site*. Dinamika perubahan posisi dan status perangkat yang sangat cepat akibat aktivitas migrasi, perbaikan, maupun ekspansi jaringan memerlukan sistem pencatatan yang adaptif.

Berdasarkan observasi di lapangan pada unit Arnet Semarang, ditemukan beberapa permasalahan utama dalam sistem pendataan inventaris perangkat: *Inkonsistensi Data*: Penggunaan metode semi-manual melalui *spreadsheet* statis menimbulkan risiko ketidaksinkronan data antara kondisi fisik di rak server dengan basis data digital. Rendahnya Akurasi: Tingginya risiko kesalahan input data (*human error*) yang mengorbankan validitas informasi aset serta isu ketidakurutan penomoran perangkat. Hambatan Aksesibilitas: Teknisi lapangan kesulitan mengakses data inventaris secara *real-time* dan *mobile-friendly* untuk mendukung pengambilan keputusan cepat saat melakukan instalasi atau perbaikan di lokasi server. Auditabilitas: Sulitnya melakukan identifikasi perangkat dan pelacakan mutasi aset saat proses audit *internal* perusahaan dilakukan.

Penelitian ini bertujuan untuk merancang dan mengimplementasikan NETRO, sebuah sistem manajemen inventaris berbasis Telegram Bot yang terintegrasi dengan basis data relasional SQLite. Sistem ini dikembangkan untuk mengelola siklus hidup perangkat telekomunikasi di Telkom Regional 4 secara efisien, akurat, dan akuntabel, sehingga setiap mutasi perangkat dapat diperbarui secara instan melalui perangkat seluler teknisi.

Untuk mengatasi permasalahan tersebut, diusulkan pengembangan sistem dengan pendekatan sebagai berikut: Implementasi Bot Pesan Instan: Menggunakan Telegram Bot API sebagai antarmuka pengguna karena memiliki responsivitas tinggi, hemat *bandwidth*, dan kompatibilitas perangkat *mobile* yang luas. Otomasi Asinkron: Menerapkan prinsip *Asynchronous I/O* pada bahasa pemrograman Python untuk menjamin performa sistem tetap stabil saat menangani trafik data tinggi dari banyak teknisi secara simultan. Keamanan Berlapis: Mengintegrasikan mekanisme *Role-Based Access Control* (RBAC) guna memastikan otorisasi akses data hanya diberikan kepada personel resmi sesuai perannya. Sentralisasi Data: Menggunakan basis data relasional SQLite untuk menjamin integritas data referensial antar entitas (SFP, LPUF, Card) melalui skema *Primary* dan *Foreign Key* yang ketat.

2. TINJAUAN PUSTAKA

Ulasan terhadap beberapa penelitian terdahulu menunjukkan bahwa penggunaan bot pesan instan dan sistem informasi berbasis digital telah banyak diadopsi untuk kebutuhan manajemen data serta pemantauan sistem secara *real-time*. Penelitian yang dilakukan oleh Pasaribu [1] dan Usnaini dkk. [2] menyoroti

pentingnya pengembangan sistem informasi inventaris berbasis web untuk meningkatkan akurasi pendataan aset kantor dan perusahaan, namun implementasi tersebut masih memerlukan perangkat komputer sebagai antarmuka utama.

Dalam konteks aksesibilitas, penelitian oleh Pangaribuan dkk. [3] menegaskan bahwa sistem informasi berbasis web sangat efektif untuk manajemen akademik, meskipun mobilitas aksesnya masih terbatas pada area dengan ketersediaan infrastruktur PC yang memadai. Selanjutnya, penelitian oleh Prasetyo dkk. [4] mengenai administrasi pemerintahan desa memberikan penekanan pada validitas data digital, namun penelitian tersebut belum menyentuh aspek otomatisasi penomoran data relasional yang bersifat dinamis.

Implementasi bot Telegram sebagai media monitoring jaringan telah diteliti oleh Wahyu dkk. [5] yang menunjukkan efektivitas penggunaan notifikasi *asinkron* untuk mempercepat penyampaian informasi gangguan. Namun, keterbatasan pada penelitian tersebut terletak pada fokus fungsionalitas yang hanya terbatas pada notifikasi, tanpa adanya fitur manajemen inventaris aset fisik yang mendalam. Di sisi teknis, studi oleh Guntupalli [6] dan Farnham [7] mengenai pemrograman *asinkron* pada Python melalui pustaka *asyncio* memberikan landasan kuat mengenai efisiensi performa aplikasi pesan instan dalam menangani traffic data yang tinggi.

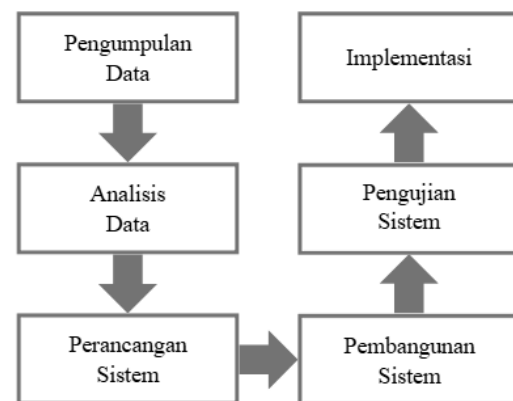
Penelitian mengenai kontrol stok barang berbasis web oleh Kurniawati dkk. [8] juga menjadi referensi penting dalam memahami siklus manajemen barang, namun sistem tersebut belum memanfaatkan fleksibilitas *platform mobile* seperti Telegram. Keamanan akses data merupakan aspek krusial lainnya, di mana penelitian oleh Prasetya dkk. [9] dan Phan [10] memberikan gambaran mengenai implementasi *Role-Based Access Control* (RBAC) dalam menjaga integritas sistem informasi berbasis *cloud*.

Perbedaan mendasar sekaligus kebaruan (*novelty*) dari penelitian NETRO dibandingkan dengan literatur-literatur di atas terletak pada integrasi *platform* Telegram sebagai antarmuka inventaris *on-the-go* yang dilengkapi dengan algoritma *gap-filling* dan *auto-sequencing* cerdas. Sistem ini tidak hanya berfungsi sebagai media pencatatan, tetapi juga mengotomatisasi standarisasi kode aset secara hierarkis berdasarkan lokasi (*site*) dan perangkat induk, yang belum ditemukan pada penelitian-penelitian sebelumnya.

3. METODE PENELITIAN

Metode yang digunakan dalam penelitian ini adalah metode pengembangan sistem dengan model *Waterfall* [11]. Pemilihan model *Waterfall* didasarkan pada karakteristik pengembangan sistem NETRO yang membutuhkan pendekatan sistematis dan berurutan guna memastikan setiap kebutuhan

fungsional inventaris di PT. Telkom Indonesia Regional 4 terpenuhi secara presisi sebelum melangkah ke tahap berikutnya. Alur penelitian ini terdiri dari enam tahapan utama sebagaimana diilustrasikan pada Gambar 1. Tahapan tersebut dimulai dari pengumpulan data hingga implementasi sistem untuk memastikan bahwa solusi yang dibangun memiliki skalabilitas yang baik dan meminimalkan risiko kegagalan integrasi antar modul perangkat lunak.



Gambar 1. Metode penelitian model *Waterfall*

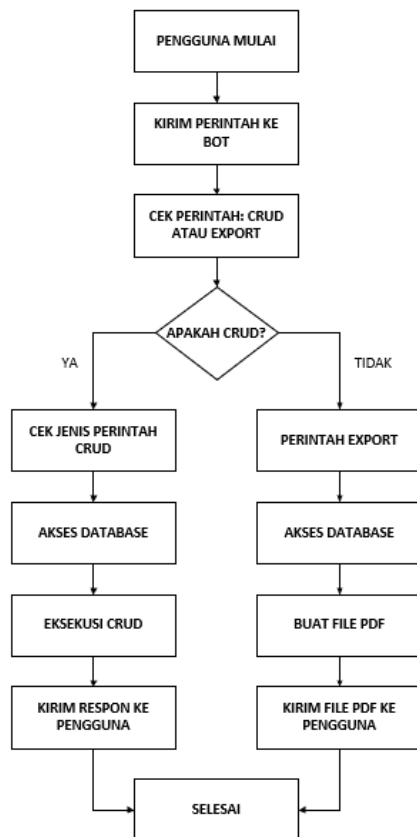
Tahap pertama adalah Pengumpulan Data. Pada tahap ini, dilakukan observasi langsung dan wawancara dengan teknisi bagian Area *Network* (Arnet) Semarang. Data yang dikumpulkan meliputi spesifikasi teknis perangkat optik, skema penamaan aset (*naming convention*) yang berlaku, serta daftar lokasi (*site*) di wilayah Jawa Tengah dan DIY. Metode ini bertujuan untuk memetakan kebutuhan informasi yang harus diakomodasi oleh sistem bot agar sesuai dengan standar dokumentasi internal perusahaan.

Tahap kedua adalah Analisis Data. Data yang telah terkumpul dianalisis menggunakan teknik pemodelan kebutuhan sistem. Pada tahap ini, dilakukan klasifikasi entitas data ke dalam tiga kategori utama perangkat, yaitu SFP, LPUF, dan *Card*. Penulis menganalisis logika relasional antara perangkat induk (LPUF) dan komponen kartu (*Card*) untuk menentukan skema basis data yang optimal, di mana setiap *Card* harus terikat secara hierarkis pada slot LPUF tertentu. Analisis juga mencakup kebutuhan fungsional seperti pelaporan otomatis dan sistem log aktivitas untuk menjaga akuntabilitas data operasional.

Tahap ketiga adalah Perancangan Sistem. Perancangan dilakukan dengan menggunakan beberapa alat bantu (*tools*) dan teknik. Perancangan basis data menggunakan pemodelan *Entity-Relationship Diagram* (ERD) yang diimplementasikan ke dalam skema basis data relasional pada SQLite. Implementasi skema ini menekankan pada penggunaan *Constraints* seperti *Primary Key* dan *Foreign Key* untuk menjamin integritas referensial data antar tabel perangkat. Untuk alur interaksi

pengguna, dirancang skema *Role-Based Access Control* (RBAC) berdasarkan prinsip *Least Privilege* yang membagi level pengguna menjadi *Admin*, *Staff*, dan *Viewer*.

Selain perancangan sistem, dirancang pula alur kerja sistem (*bot workflow*) untuk memetakan logika operasional bot dari sisi *backend*, sebagaimana diilustrasikan pada Gambar 2. Perancangan ini memanfaatkan fitur-fitur Telegram API seperti *Inline Keyboards* dan *Callback Queries* guna meminimalkan kesalahan *input* teks oleh teknisi.



Gambar 2. Flowchart alur kerja bot NETRO

Setiap perintah yang diterima akan divalidasi terlebih dahulu melalui pengecekan *role* pengguna (RBAC). Jika akses diberikan, Berdasarkan Gambar 2, sistem akan membagi eksekusi ke dalam dua kategori utama: fungsi manajemen inventaris untuk melakukan Operasi CRUD (SFP, LPUF, Card) yang berdampak pada pembaruan basis data, atau fungsi pelaporan untuk *Generate* Dokumen dalam format PDF dan Excel secara *in-memory*.

Tahap keempat adalah Pembangunan Sistem. Tahap ini merupakan proses pengkodean (*coding*) menggunakan bahasa pemrograman Python. Penulis menggunakan *framework* python-telegram-bot versi asinkron untuk menangani trafik pesan secara simultan tanpa menghambat performa server (*non-blocking*). Dalam proses ini, diimplementasikan algoritma *auto-sequencing* dan *gap-filling* untuk otomatisasi kode aset

secara cerdas. Pembangunan sistem juga melibatkan penggunaan pustaka *Pandas* dan *Openpyxl* untuk pengolahan laporan Excel, serta *ReportLab* untuk pembuatan dokumen PDF. Modul pelaporan ini mengadopsi teknik *In-Memory File Streaming* yang memungkinkan dokumen dikirimkan langsung dari memori server tanpa perlu menyimpannya secara fisik di *disk*, sehingga meningkatkan efisiensi penyimpanan dan kecepatan akses.

Tahap kelima adalah Pengujian Sistem. Teknik pengujian yang digunakan adalah *Black-Box Testing* untuk memastikan seluruh perintah (*command*) bot berfungsi sesuai spesifikasi kebutuhan pengguna. Pengujian mencakup validasi input menggunakan *Regular Expressions (Regex)* untuk menjamin setiap kode lokasi atau nomor seri yang masuk ke basis data telah mengikuti standar *naming convention* yang ketat. Penulis juga melakukan pengujian integritas data pada skenario penghapusan dan pembaruan perangkat untuk memastikan konsistensi relasi antar tabel tetap terjaga dengan baik meskipun terjadi mutasi data yang masif.

Tahap keenam adalah Implementasi. Pada tahap ini, sistem NETRO di-deploy ke dalam lingkungan *Virtual Private Server (VPS)* berbasis Linux. Proses implementasi diawali dengan menjalankan skrip *setup_db.py* untuk migrasi skema tabel dan inisialisasi token akses administratif. Setelah bot aktif, dilakukan sosialisasi penggunaan kepada tim operasional Arnet Semarang untuk mulai melakukan migrasi data inventaris dari format manual ke dalam ekosistem digital sistem NETRO.

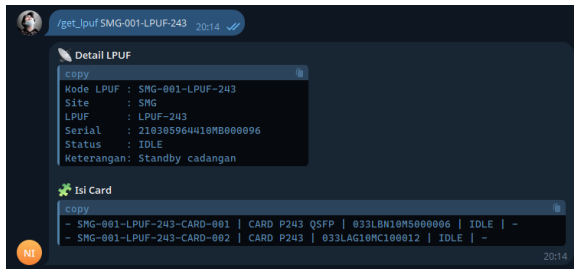
4. HASIL DAN PEMBAHASAN

Bagian ini memaparkan hasil perancangan dan implementasi sistem NETRO sebagai solusi manajemen inventaris pada infrastruktur Telkom Indonesia Regional 4. Pembahasan mencakup analisis mendalam terhadap arsitektur antarmuka, mekanisme keamanan pengguna, serta efisiensi algoritma yang diimplementasikan untuk menjamin integritas data inventaris secara *real-time*.

4.1. Implementasi Antarmuka dan Arsitektur Manajemen Pengguna

Implementasi NETRO menghasilkan antarmuka berbasis pesan teks yang interaktif dan responsif, memanfaatkan keunggulan *platform* Telegram dalam hal konsumsi *bandwidth* yang rendah dan kompatibilitas perangkat seluler yang tinggi. Antarmuka ini dirancang untuk meminimalkan beban kognitif teknisi melalui penggunaan *Inline Keyboards* dan *Callback Queries*, yang memungkinkan navigasi menu dilakukan melalui penekanan tombol tanpa harus mengetik perintah secara manual sebagaimana diilustrasikan pada Gambar 3. Melalui visualisasi tersebut, teknisi dapat memahami struktur data perangkat secara cepat saat berada di lapangan tanpa perlu membuka terminal komputer. Penggunaan tipografi *monospace* pada respon bot memastikan

bahwa karakter kritis seperti nomor seri (*Serial Number*) dan kode aset tetap simetris dan mudah dibaca, menghindari risiko kesalahan identifikasi akibat distorsi visual pada layar ponsel.



Gambar 3. Tampilan antarmuka hasil pencarian modul LPUF

Dari tampilan pada Gambar 3, dapat dilihat bahwa sistem mampu menyajikan informasi teknis secara terstruktur. Aspek keamanan pendaftaran pengguna dilakukan melalui sistem *Deep Linking* dan penggunaan *Access Token* divalidasi silang dengan tabel *access_tokens* untuk memastikan hanya personel resmi yang memiliki akses. Selain itu, integrasi teknik *In-Memory File Streaming* dalam modul pelaporan memungkinkan sistem untuk menghasilkan dokumen PDF dan Excel secara dinamis tanpa meninggalkan jejak residu file pada penyimpanan server. Hal ini tidak hanya mengoptimalkan kapasitas *disk VPS*, tetapi juga menjamin kerahasiaan data perusahaan karena dokumen hanya eksis dalam memori selama proses transmisi berlangsung.

4.2. Analisis Algoritma Auto-Sequencing dan Gap-Filling

Implementasi algoritma penomoran otomatis yang cerdas menjadi nilai kebaruan dalam penelitian ini. Permasalahan ketidakteraturan nomor urut aset akibat penghapusan data lama diatasi melalui fitur *Gap-Filling*. Algoritma ini bekerja dengan melakukan ekstraksi urutan numerik menggunakan fungsi SQL *SUBSTRING_INDEX* dan *CAST AS INTEGER*. Logika algoritma dirancang untuk melakukan iterasi pada kolom identitas guna mendeteksi nilai *integer* terkecil yang tidak tersedia dalam urutan. Apabila ditemukan selisih pada urutan ID (misalnya meloncat dari 0001 ke 0003), algoritma secara otomatis menetapkan angka yang hilang (0002) sebagai ID perangkat baru. Hal ini menjamin indeks identitas pada SQLite tetap rapat dan berurutan secara logis untuk mempermudah proses audit fisik, sekaligus mencegah

terjadinya "ID clash" yang sering ditemukan pada metode pendataan manual berbasis *spreadsheet*.

4.3. Analisis Efisiensi Pelaporan dan Transparansi Data

Implementasi sistem NETRO membawa dampak signifikan terhadap kecepatan rekapitulasi data operasional. Berdasarkan analisis di lapangan, proses pembuatan laporan inventaris yang sebelumnya dilakukan secara manual kini dapat diselesaikan secara instan melalui perintah */report*. Pengguna dapat memilih opsi "Laporan Aktivitas atau Semua Modul" untuk menghasilkan satu dokumen *spreadsheet* terintegrasi yang mencakup seluruh database perangkat. Hasil luaran laporan dalam format Excel tersebut divisualisasikan pada Gambar 4, yang menunjukkan pembagian data ke dalam beberapa sheet terstruktur seperti SFP, LPUF, Card, dan Log_Aktivitas untuk memudahkan navigasi informasi.

dickyk12	CREATE	LPUF	SMG	Menambah LPUF baru
dickyk12	CREATE	LPUF	SMG	Menambah LPUF baru
dickyk12	CREATE	Card	SMG	Menambah Card baru
dickyk12	UPDATE	Card	SMG	Perubahan pada S
dickyk12	CREATE	Card	SMG	Menambah Card baru
dickyk12	CREATE	LPUF	SMG	Menambah LPUF baru
dickyk12	CREATE	Card	SMG	Menambah Card baru

Gambar 4. Visualisasi struktur laporan excel terintegrasi

Kehadiran *sheet* riwayat aktivitas (*Log_Aktivitas*) yang disajikan secara kronologis memungkinkan pelacakan mutasi perangkat secara transparan. Setiap entri dalam laporan tersebut memuat informasi detail mengenai siapa yang melakukan perubahan, kapan waktu eksekusinya, serta rincian teknis aset yang dimodifikasi. Hal ini memastikan akuntabilitas manajemen infrastruktur jaringan di lingkungan Arnet Semarang tetap terjaga dan memudahkan proses audit *internal* tanpa perlu melakukan konsolidasi data secara terpisah.

4.4. Pengujian Black-Box

Pengujian sistem dilakukan menggunakan metode *Black-Box Testing* untuk memastikan fungsionalitas aplikasi sesuai dengan spesifikasi kebutuhan operasional teknisi. Hasil pengujian menunjukkan tingkat keberhasilan 100% pada seluruh fitur krusial, mulai dari aktivasi akses hingga pembuatan laporan otomatis sebagaimana disajikan pada Tabel 1.

Tabel 1. Hasil Pengujian *Black-Box* Sistem NETRO

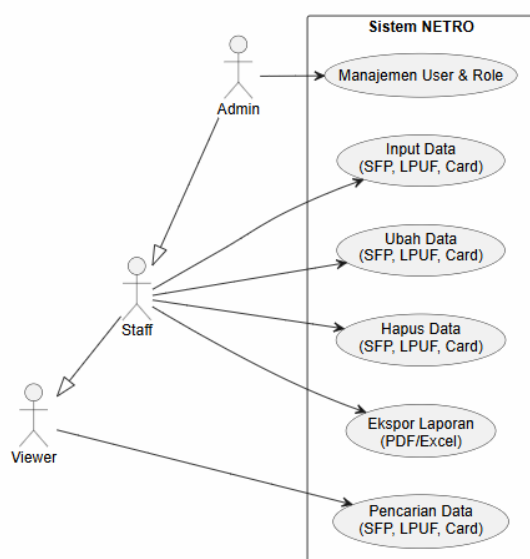
No	Fitur / Perintah	Skenario Pengujian	Output yang Diharapkan	Hasil
1	Aktivasi Akses	<i>Input /start</i> diikuti dengan <i>Access Token</i> yang valid.	Sistem memverifikasi token dan menetapkan role pengguna secara otomatis.	Berhasil
2	Manajemen Lokasi	Mengirim data lokasi baru menggunakan format <i>multi-line</i> pada <i>/tambah_tempat</i> .	Sistem menggenerasi kode tempat unik dan menyimpannya ke tabel tempat.	Berhasil

No	Fitur / Perintah	Skenario Pengujian	Output yang Diharapkan	Hasil
3	Inventarisasi SFP	Input data SFP dengan atribut merk, kapasitas, dan jarak melalui /tambah_sfp.	Validasi keunikan nomor seri dan pengecekan relasi terhadap kode lokasi.	Berhasil
4	Pembaruan Data	Memanggil data Card menggunakan perintah /update_card [KODE].	Bot menampilkan data lama untuk disunting kembali.	Berhasil
5	Sinkronisasi ID	Mengubah data Site pada modul Card yang sudah terdaftar.	Algoritma Auto-Rename memperbaiki prefix kode perangkat secara otomatis.	Berhasil
6	Pelaporan PDF	Eksekusi perintah /export dengan parameter pemilihan Site tertentu.	Bot menghasilkan dokumen .pdf secara in-memory dan mengirimkannya ke chat.	Berhasil
7	Pelaporan Excel	Eksekusi perintah /report untuk modul inventaris tertentu.	Bot mengirimkan file .xlsx dengan format tabel dan styling yang rapi.	Berhasil
8	Audit Trail	Melakukan perubahan data (Update/Delete) pada salah satu aset.	Detail aksi, waktu, dan ID pengguna tercatat secara otomatis pada tabel logs.	Berhasil
9	Keamanan Sistem	Pengguna dengan role 'Viewer' mencoba mengakses fungsi /delete_card.	Sistem menolak perintah dan menampilkan pesan peringatan "Akses Ditolak".	Berhasil

Berdasarkan Tabel 1, dapat dianalisis bahwa sistem memiliki tingkat reliabilitas yang sangat baik. Keberhasilan skenario nomor 9 mengonfirmasi bahwa *decorator* keamanan pada kode program berfungsi efektif dalam mencegah eskalasi hak akses. Hal ini membuktikan bahwa validasi berlapis yang diterapkan pada setiap *entry point* perintah mampu menepis upaya manipulasi data oleh pengguna tanpa otoritas yang sesuai.

4.5. Analisis Keamanan dan Manajemen Peran

Keamanan informasi dalam sistem NETRO dikelola melalui mekanisme *Role-Based Access Control* (RBAC) yang mengadopsi prinsip *Least Privilege*. Secara teknis, kontrol ini diintegrasikan pada *level handler* menggunakan *Python Decorators* asinkron, seperti *@admin_only* dan *@staff_only*, yang melakukan validasi identitas unik pengguna pada tabel *telegram_users* sebelum sebuah perintah dieksekusi. *Use Case Diagram* pembagian hak akses yang mendetail untuk setiap peran pengguna disajikan dalam Gambar 5.



Gambar 5. Use case diagram sistem NETRO

Berdasarkan Gambar 5, dapat dianalisis bahwa pembagian peran dirancang untuk mengakomodasi kebutuhan operasional Arnet Regional 4 secara spesifik. Peran *Staff* memiliki hak penuh terhadap siklus hidup data aset (*Create, Read, Update, Delete*) untuk mendukung mobilitas teknisi di lapangan, sementara peran *Viewer* dibatasi hanya pada fungsi pemantauan (*Read-Only*) guna menjaga integritas basis data. Peran *Admin* memegang otoritas tertinggi, terutama dalam mengelola *Access Token* dan manajemen akses pengguna, yang berfungsi sebagai kendali utama dalam siklus keamanan akun. Implementasi *use case diagram* ini memastikan adanya akuntabilitas digital pada setiap tindakan administratif yang dilakukan di dalam sistem.

4.6. Analisis Ketahanan Sistem (Error Handling)

NETRO menerapkan strategi *Robust Error Handling* menggunakan blok *try-except-finally* untuk menangani kesalahan input teknisi di lapangan tanpa menghentikan *event loop asinkron* aplikasi. Sistem secara spesifik menangani *sqlite3.IntegrityError* untuk mencegah duplikasi nomor seri dan memberikan pesan edukatif bagi pengguna alih-alih menyebabkan aplikasi berhenti mendadak (*crash*).

5. KESIMPULAN DAN SARAN

Berdasarkan seluruh rangkaian proses perancangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa penelitian ini telah berhasil membangun sistem NETRO sebagai platform manajemen inventaris perangkat jaringan berbasis Telegram Bot pada PT. Telkom Indonesia Regional 4 yang responsif dan dapat diakses secara *mobile* oleh teknisi di lapangan. Keberhasilan pengujian *Black-Box* dengan tingkat validitas 100% mengonfirmasi bahwa fitur otomatisasi penomoran aset (*auto-sequencing*), manajemen hak akses (*Role-Based Access Control*), serta pelaporan otomatis dalam format PDF dan Excel telah berfungsi sesuai dengan spesifikasi kebutuhan operasional Arnet Semarang. Kelebihan utama sistem NETRO terletak pada implementasi algoritma *gap-filling* yang secara signifikan mereduksi risiko duplikasi data, modul

audit *logging asinkron* untuk transparansi transaksi, serta teknik *in-memory file streaming* yang meningkatkan efisiensi operasional pengunduhan laporan. Meskipun efektif dalam mentransformasi manajemen inventaris konvensional menjadi ekosistem digital, sistem ini masih memiliki keterbatasan pada ketergantungan mutlak terhadap stabilitas server Telegram API serta potensi kesalahan tipografi (*typo*) pada proses input data berbasis teks. Oleh karena itu, saran untuk pengembangan selanjutnya adalah mengintegrasikan fitur pemindaian *barcode* atau *QR Code* secara native di dalam aplikasi guna meminimalkan kesalahan input manual dan meningkatkan akurasi pendataan aset secara *real-time* di lapangan.

DAFTAR PUSTAKA

- [1] J. S. Pasaribu, "PERANCANGAN SISTEM INFORMASI BERBASIS WEB PENGELOLAAN INVENTARIS ASET KANTOR DI PT. MPM FINANCE BANDUNG," *Jurnal Ilmiah Teknologi Infomasi Terapan*, vol. 7, no. 3, pp. 229–241, Aug. 2021, doi: 10.33197/JITTER.VOL7.ISS3.2021.655.
- [2] M. Usnaini, V. Yasin, and A. Z. Sianipar, "Perancangan sistem informasi inventarisasi aset berbasis web menggunakan metode waterfall," *Jurnal Manajemen Informatika Jayakarta*, vol. 1, no. 1, pp. 36–55, Feb. 2021, doi: 10.52362/JMIJAYAKARTA.V1I1.415.
- [3] I. Pangaribuan and F. Subakti, "Sistem Informasi Akademik Berbasis Web pada SMK (Sekolah Menengah Kejuruan) Teknologi Industri Pembangunan Cimahi," *Jurnal Teknologi dan Informasi*, vol. 9, no. 2, pp. 128–137, Sep. 2019, doi: 10.34010/JATI.V9I2.1836.
- [4] T. Prasetyo and R. P. Dhaniawaty, "Sistem Informasi Administrasi Pemerintahan Desa pada Desa Cilayung Kabupaten Kuningan," *Jurnal Teknologi dan Informasi*, vol. 10, no. 1, pp. 52–61, Mar. 2020, doi: 10.34010/JATI.V10I1.2852.
- [5] A. G. Wahyuriyanto and H. Yulianton, "Implementasi Sistem Monitoring Jaringan Berbasis Simple Network Management Protocol Dengan Notifikasi Telegram," *Jurnal Ilmiah Komputer*, 2024.
- [6] B. Guntupalli, "Asynchronous Programming in Java/Python: A Developer's Guide," *International Journal of Emerging Research in Engineering and Technology*, vol. 3, no. 2, pp. 70–78, Jun. 2022, doi: 10.63282/3050-922X.IJERET-V3I2P108.
- [7] Caleb Hattingh, B. Farnham, S. Tokyo, B. Boston, F. Sebastopol, and T. Beijing, *Using Asyncio in Python Understanding Python's Asynchronous Programming Features*. 2020.
- [8] E. Kurniawati and A. Ikhwan, "Perancangan Sistem Informasi Manajemen Inventaris Kontrol Stok Barang Berbasis Web," *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 6, no. 3, pp. 408–415, Jul. 2023, doi: 10.32493/jtsi.v6i3.30881.
- [9] Y. A. Prasetia and D. Manongga, "ROLE-BASED ACCESS CONTROL (RBAC) UNTUK SISTEM OTORISASI TERPUSAT BERBASIS FLASK STUDI KASUS PT. XYZ," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 9, no. 4, pp. 1768–1778, Nov. 2024, doi: 10.29100/JIPI.V9I4.5403.
- [10] I. Kurnia Phan, "Penerapan Role Based Access Control dalam Sistem Supply Chain Management Berbasis Cloud," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 3, no. 2, pp. 339–348, Nov. 2023, doi: 10.57152/MALCOM.V3I2.1259.
- [11] W. N. A. Wan Ali and W. A. J. Wan Yahaya, "Waterfall-Addie Model: An Integration Of Software Development Model And Instructional Systems Design In Developing A Digital Video Learning Application," *Asean Journal of Teaching and Learning in Higher Education*, vol. 15, no. 1, pp. 1–28, Jun. 2023, doi: 10.17576/ajtlhe.1501.2023.01