

Rancang Bangun Sistem Pendukung Keputusan Monitoring Jaringan GPON Berbasis *Rule-Based Classification*

DOI: <http://dx.doi.org/10.35889/progresif.v22i3.3802>

Creative Commons License 4.0 (CC BY –NC)



Ni Made Rai Masita Dewi^{1*}, Vincent Alfian Artha²

¹Sistem Informasi, Institut Teknologi dan Bisnis STIKOM Bali, Denpasar, Indonesia

²Sistem Komputer, Institut Teknologi dan Bisnis STIKOM Bali, Denpasar, Indonesia

*e-mail Corresponding Author: raimasita@stikom-bali.ac.id

Abstrak

Monitoring subscriber network devices in Internet Service Providers (ISPs) is generally conducted manually and reactively, causing the Network Operations Center (NOC) team to frequently experience delays in detecting disruptions within the GPON network. This study designs and develops a web-based Decision Support System (DSS) utilizing rule-based classification with a thresholding approach to diagnose GPON network anomalies proactively and in real-time at PT Lintas Data Prima (LDP) Bali. The system was developed using the Next.js 15, Python FastAPI, and Supabase (PostgreSQL) frameworks, integrated with the SmartOLT v2 API and Telegram Bot API. The diagnostic engine classifies the subscriber's ONT device conditions into three severity categories, namely Critical, Warning, and Normal, based on the SFP attenuation values and PPPoE status. The test results indicate that the system is capable of automatically detecting and classifying anomalies, as well as instantly sending emergency notifications via Telegram Bot to NOC technicians. This system is expected to serve as a reference for developing proactive network monitoring solutions in ISPs.

Keywords: Decision Support System; GPON; Rule-Based Classification; NOC; ISP

Abstrak

Pemantauan perangkat jaringan pelanggan pada *Internet Service Provider* (ISP) umumnya dilakukan secara manual dan reaktif, sehingga tim *Network Operations Center* (NOC) kerap terlambat mendeteksi gangguan pada jaringan GPON. Penelitian ini merancang dan membangun Sistem Pendukung Keputusan (SPK) berbasis *rule-based classification* dengan pendekatan *thresholding* untuk mendiagnosis gangguan jaringan GPON secara proaktif dan *real-time* pada PT Lintas Data Prima (LDP) Bali. Sistem dikembangkan menggunakan *framework* Next.js 15, Python FastAPI, dan Supabase (PostgreSQL) dengan integrasi SmartOLT v2 API dan Telegram Bot API. Mesin diagnosis mengklasifikasikan kondisi perangkat ONT pelanggan ke dalam tiga kategori keparahan, yaitu *Critical*, *Warning*, dan *Normal*, berdasarkan nilai redaman SFP dan status PPPoE. Hasil pengujian menunjukkan sistem mampu mendeteksi dan mengklasifikasikan gangguan secara otomatis serta mengirimkan notifikasi darurat melalui Telegram Bot kepada teknisi NOC secara instan. Sistem ini diharapkan dapat menjadi referensi pengembangan solusi pemantauan jaringan proaktif pada ISP.

Kata kunci: Sistem Pendukung Keputusan; GPON; Rule-Based Classification; NOC; ISP

1. Pendahuluan

Infrastruktur akses berbasis serat optik menjadi komponen penting dalam penyediaan layanan internet berkecepatan tinggi, terutama pada ISP yang melayani pelanggan residensial dan korporasi. Salah satu teknologi akses yang banyak digunakan adalah *Gigabit Passive Optical Network* (GPON), karena arsitektur *point-to-multipoint* memungkinkan satu *Optical Line Terminal* (OLT) melayani banyak *Optical Network Terminal* (ONT) melalui *Optical Distribution Network*

(ODN) [1],[2]. Pada lapisan fisik, kualitas layanan GPON sangat dipengaruhi oleh daya optik, redaman, kualitas sambungan, konektor, *splitter*, serta kondisi perangkat pelanggan. Oleh karena itu, monitoring parameter teknis GPON secara proaktif menjadi kebutuhan penting agar kualitas layanan dan ketersediaan jaringan dapat dipertahankan [1],[3].

Objek penelitian ini adalah PT Lintas Data Prima (LDP) Bali, yaitu unit operasional ISP yang mengelola lebih dari 1.600 perangkat ONT pelanggan aktif. Dalam proses operasional harian, tim NOC memanfaatkan SmartOLT untuk memantau status ONT, tetapi data yang ditampilkan masih bercampur antara indikasi gangguan fisik, seperti *Loss of Signal (LOS)*, *power failure*, *dying gasp*, dan redaman tinggi, dengan status administratif seperti isolir atau *admin disable*. Kondisi tersebut menyebabkan proses diagnosis awal masih memerlukan pemeriksaan manual dan verifikasi lintas sistem. Jika dibiarkan, proses manual ini dapat memperbesar risiko *slow response* dan meningkatkan *Mean Time to Repair (MTTR)*, terutama ketika jumlah pelanggan terus bertambah.

Sejumlah penelitian terdahulu telah membahas monitoring jaringan, pengelolaan FTTH, pemanfaatan REST API, monitoring PON, dan notifikasi gangguan. Mukiman dan Masruri [1] mengembangkan aplikasi laporan redaman FTTH berbasis GPON untuk membantu pencatatan parameter optik. Kusuma dkk. [3] membahas implementasi Mikhmon pada infrastruktur FTTH untuk kebutuhan ISP RT/RW Net. Widyaningtyas dan Wahyono [4] menunjukkan bahwa REST API dapat digunakan untuk mendukung integrasi aplikasi monitoring lintas platform. Fernandez dkk. [5] membahas teknik monitoring *Passive Optical Network*, sedangkan Amin dkk. [6] menggunakan Telegram Bot untuk mempercepat distribusi informasi monitoring jaringan. Namun, sebagian besar penelitian tersebut belum secara spesifik mengintegrasikan penarikan data SmartOLT v2 API, klasifikasi diagnosis berbasis aturan untuk membedakan gangguan fisik dan status administratif, *smart-logging* untuk efisiensi *database*, *dashboard real-time*, serta notifikasi *event-driven* dalam satu rancangan SPK monitoring GPON.

Berdasarkan gap tersebut, penelitian ini mengusulkan SPK monitoring jaringan GPON berbasis *rule-based classification* [7]. Konsep solusi memanfaatkan sifat gangguan GPON yang deterministik dan dapat dirumuskan melalui aturan *If-Then* [8], [9], misalnya status LOS sebagai indikasi gangguan kritis dan nilai RX Power di bawah ambang batas -27 dBm sebagai indikasi redaman tinggi [1],[10]. Mesin diagnosis dikembangkan pada *backend* FastAPI yang mendukung pengembangan *web service* dan pengujian performa API [11], [12], data disimpan pada Supabase PostgreSQL [13], tampilan dibangun menggunakan Next.js [14], dan notifikasi dikirim melalui Telegram Bot [6]. Kebaruan penelitian ini terletak pada integrasi diferensiasi otomatis antara kendala teknis jaringan dan status administratif pelanggan, serta penerapan *change-based logging* untuk mengurangi penumpukan data monitoring historis.

2. Metodologi

2.1 Metode Penelitian

Penelitian ini menggunakan metode *Research and Development (R&D)* dengan pendekatan pengembangan perangkat lunak berbasis *prototyping*. R&D dipilih karena tujuan penelitian tidak hanya menganalisis masalah, tetapi juga menghasilkan produk sistem yang dapat diuji pada skenario operasional nyata. Tahapan penelitian terdiri atas analisis kebutuhan, perancangan arsitektur, implementasi sistem, eksperimen fungsional, dan evaluasi hasil.

2.2 Objek dan Sumber Data

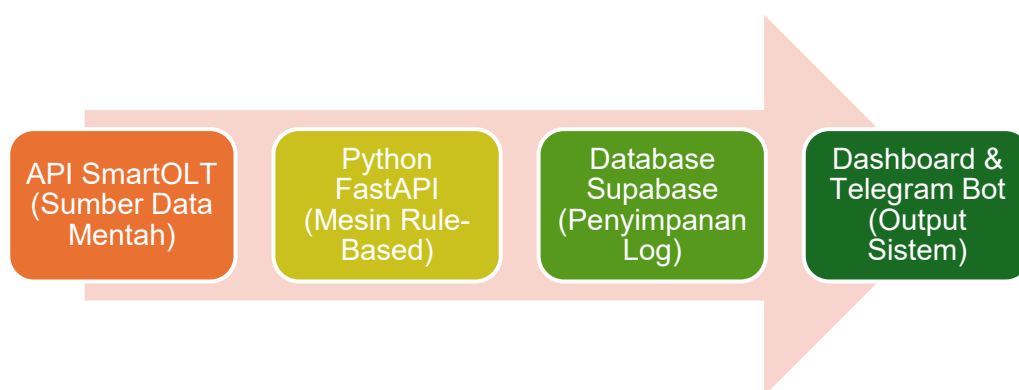
Objek penelitian adalah proses monitoring perangkat ONT pelanggan GPON pada PT Lintas Data Prima Bali. Sumber data utama berasal dari SmartOLT v2 API, khususnya *endpoint* `get_all_onus_details` yang menyediakan informasi teknis ONT, antara lain nama pelanggan, *serial number (SN)*, OLT, *board*, *port*, ONU ID, mode perangkat, status PPPoE, dan nilai daya optik. *Endpoint* tersebut digunakan sebagai sumber data operasional karena menyediakan informasi teknis ONT yang dibutuhkan untuk proses ETL dan klasifikasi diagnosis. Interval *polling* dirancang tidak terlalu agresif agar proses pengambilan data tetap stabil dan tidak mengganggu layanan API pihak ketiga.

2.3 Analisis Kebutuhan Sistem

Kebutuhan fungsional sistem meliputi: (a) penarikan data status ONT secara berkala dari SmartOLT v2 API, (b) klasifikasi otomatis kondisi perangkat berdasarkan aturan diagnosis, (c) penyimpanan log historis pada *database*, (d) visualisasi *dashboard real-time*, dan (e) pengiriman notifikasi darurat melalui Telegram Bot. Kebutuhan non-fungsional meliputi ketersediaan sistem untuk monitoring berkala, waktu respons notifikasi kurang dari 60 detik setelah transisi status terdeteksi, keamanan kredensial API melalui file environment, dan efisiensi penyimpanan data historis.

2.4 Perancangan Arsitektur Sistem

Arsitektur sistem dirancang secara *decoupled* agar setiap komponen memiliki tanggung jawab yang jelas. SmartOLT v2 API berperan sebagai sumber data mentah, *backend* FastAPI berperan sebagai layanan ETL dan mesin diagnosis, Supabase PostgreSQL menyimpan log historis dan status terbaru, sedangkan *dashboard* Next.js dan Telegram Bot menjadi lapisan keluaran informasi bagi tim NOC. Alur arsitektur makro ditunjukkan pada Gambar 1.



Gambar 1 Arsitektur Makro Sistem Diagnosis Jaringan

Tabel 1 menunjukkan tumpukan teknologi yang digunakan dalam pengembangan sistem.

Tabel 1 Tumpukan Teknologi Sistem

Komponen	Teknologi
<i>Backend API & Service</i>	Python 3.10+, FastAPI, APScheduler
<i>Frontend Framework</i>	React 19, Next.js 15 (App Router)
<i>Styling & UI</i>	Tailwind CSS v4, <i>Light/Dark Mode</i>
<i>Database</i>	Supabase (PostgreSQL)
Integrasi Eksternal	SmartOLT v2 API, Telegram Bot API

2.5 Perancangan Mesin Diagnosis (*Rule-Based Classification*)

Mesin diagnosis menggunakan *Nested Rule-Based Classification* dengan pendekatan *thresholding*. Aturan disusun berdasarkan prioritas keparahan sehingga kondisi kritis seperti LOS diperiksa lebih dahulu dibandingkan *warning*. Pendekatan ini dipilih karena parameter gangguan GPON bersifat terukur dan dapat diterjemahkan ke dalam aturan *If-Then*. Agar logika redaman konsisten, nilai RX Power yang lebih kecil dari -27 dBm dikategorikan sebagai redaman tinggi karena semakin negatif nilai dBm menunjukkan sinyal optik semakin lemah.

Tabel 2 Basis Aturan Diagnosis Jaringan GPON

Kode	Kondisi Logika (If-Then)	Hasil Diagnosis	Keparahan	Rekomendasi Tindakan
R1	<i>If status is "LOS"</i>	Kabel Putus (LOS)	<i>CRITICAL</i>	Cek sambungan FO / <i>Splicing</i> ulang. Gunakan OTDR.
R2	<i>If status is "Power Failure"</i>	Listrik Pelanggan Padam	<i>WARNING</i>	Konfirmasi ke pelanggan terkait adaptor/listrik rumah.
R3	<i>If mode is "Bridge"</i>	Salah Mode / <i>Bridge</i>	<i>WARNING</i>	Ubah konfigurasi perangkat ke mode <i>Routing</i> .
R4	<i>If rx_power < -27 dBm</i>	Sinyal Lemah / Redaman Tinggi	<i>WARNING</i>	Pembersihan konektor ODP/Roset atau cek lekukan kabel.
R5	<i>Otherwise</i>	Kondisi Baik	<i>NORMAL</i>	Tidak diperlukan tindakan teknis segera.

2.6 Mekanisme Smart-Logging dan Deduplikasi

Untuk mengefisienkan penggunaan kapasitas *database*, sistem menerapkan algoritma *Smart-Logging* berbasis *change-based logging*. Log baru hanya ditulis ke *database* apabila terjadi perubahan status perangkat (*state transition*). Jika kondisi tidak berubah, sistem membuang data dari memori tanpa operasi tulis. Selain itu, mekanisme heartbeat 24 jam memastikan setidaknya satu entri log per hari ditulis meskipun tidak ada perubahan status, guna menjaga validitas rekam jejak pemantauan.

Pada sisi penyajian data, sistem menerapkan algoritma deduplikasi dua lapis untuk menangani kasus penggantian perangkat ONT. Lapisan pertama melakukan filter unik berdasarkan *Serial Number* (SN) perangkat, sedangkan lapisan kedua melakukan filter berdasarkan nama pelanggan untuk menangani tabrakan data akibat penggantian perangkat.

2.7 Desain Eksperimen Fungsional

Mengacu pada kebutuhan fungsional, pengujian tidak hanya dilakukan sebagai *blackbox* sederhana, tetapi dirancang sebagai eksperimen fungsional. Setiap fitur diuji secara terpisah menggunakan beberapa perlakuan berbeda. Variabel yang diset merupakan input atau konfigurasi yang diberikan ke sistem, sedangkan variabel yang diamati merupakan keluaran aktual yang dicatat selama eksperimen. Pola ini sejalan dengan pengujian *blackbox* pada sistem pendukung keputusan yang menilai kesesuaian keluaran sistem terhadap kebutuhan fungsional [15].

Tabel 3 Desain Eksperimen Fungsional Sistem

Fitur yang Diuji	Variabel yang Diset	Variabel yang Diamati	Kriteria Keberhasilan
Akuisisi data SmartOLT	<i>Endpoint</i> API, token, <i>limit data</i> , interval <i>polling</i>	HTTP status, jumlah ONT, durasi siklus	Data berhasil ditarik dan tidak melewati interval 20 menit
Klasifikasi diagnosis <i>Smart-logging</i>	Status PPPoE, mode ONT, nilai RX Power perangkat tetap/berubah, <i>heartbeat</i>	Kategori, diagnosis, rekomendasi Operasi INSERT, jumlah baris log	<i>Output</i> sesuai <i>rule base</i> INSERT hanya terjadi saat transisi atau <i>heartbeat</i>
<i>Dashboard real-time</i>	Data agregat Normal, Warning, Critical	Kartu statistik, grafik redaman, tabel log	Visualisasi berubah mengikuti data terbaru
Telegram Bot	Transisi status <i>Critical</i> dan pemulihan ke Normal	Isi pesan, waktu pengiriman, tujuan grup	Notifikasi terkirim sesuai event <i>Critical</i> atau <i>recovery</i>

3. Hasil

3.1 Implementasi Akuisisi Data SmartOLT

Sistem berhasil menarik data ONT secara berkala dari SmartOLT v2 API menggunakan metode HTTP GET. Kredensial API disimpan pada file konfigurasi environment dengan variabel SMARTOLT_API_URL dan SMARTOLT_API_KEY agar tidak ditanam langsung pada kode aplikasi. Data yang diterima berupa array JSON yang kemudian dinormalisasi, terutama pada field nilai redaman yang semula bertipe string menjadi float agar dapat diproses oleh mesin diagnosis.

```

C> raw_smartolt_data.json > {} 0
1  [
2  {
3      "unique_external_id": "736620700157",
4      "pon_type": "gpon",
5      "sn": "736620700157",
6      "olt": "736620700157",
7      "olt_name": "LDP-OLT-Sanur",
8      "board": "2",
9      "port": "1",
10     "onu": "2",
11     "onu_type_id": "73",
12     "onu_type_name": "F670L",
13     "zone_id": "54",
14     "zone_name": "BTS SANUR",
15     "name": "Mohamad Arifin -14.27 dBm / -19.16 dBm",
16     "address": "Dusun xxxxx",
17     "odb_name": "",
18     "odb_port": "",
19     "mode": "Routing",
20     "wan_mode": "Static",
21     "ip_address": "10.x.x.x",
22     "subnet_mask": "255.x.x.x",
23     "default_gateway": "10.x.x.x",

```

Gambar 2 Struktur Objek JSON Raw Data SmartOLT v2 API

3.2 Performa Eksekusi Siklus Polling

Pengamatan empiris dilakukan pada tiga siklus *polling*. Setiap siklus menarik data 1.621 pelanggan aktif dengan parameter *bulk fetching*. Hasil pengamatan menunjukkan rata-rata durasi pemrosesan sekitar 485 detik atau 8,08 menit. Dengan interval polling 20 menit, sistem masih memiliki *headroom* sekitar 12 menit sebelum siklus berikutnya dimulai.

Tabel 4 Data Performa Eksekusi Siklus Monitoring

Log Waktu Eksekusi	Jumlah Pelanggan	Durasi Eksekusi (<i>Cycle Time</i>)	Status
11/04/26, 14:08	1.621	461.80 detik	Success
11/04/26, 14:28	1.621	524.99 detik	Success
11/04/26, 14:48	1.621	468.45 detik	Success
Rata-rata Durasi	1.621	≈ 485 detik (8,08 menit)	Optimal

3.3 Hasil Eksperimen Akuisisi Data

Tabel 5 Eksperimen Akuisisi Data SmartOLT v2 API

Perlakuan	Variabel yang Diset	Variabel yang Diamati	Hasil Pengamatan	Kesimpulan
P1	Token valid, endpoint get_all_onus_details, limit=5000	HTTP status, jumlah data, durasi	HTTP 200 OK, 1.621 data ONT, 461,80 detik	Berhasil
P2	Token valid, endpoint sama, polling siklus kedua	HTTP status, jumlah data, durasi	HTTP 200 OK, 1.621 data ONT, 524,99 detik	Berhasil

Perlakuan	Variabel yang Diset	Variabel yang Diamati	Hasil Pengamatan	Kesimpulan
P3	Token valid, endpoint sama, polling siklus ketiga	HTTP status, jumlah data, durasi	HTTP 200 OK, 1.621 data ONT, 468,45 detik	Berhasil
P4	Interval polling 20 menit	Potensi process overlapping	Headroom ± 12 menit; tidak overlap	Tidak overlap

3.4 Hasil Eksperimen Klasifikasi Diagnosis

Eksperimen klasifikasi dilakukan dengan memberikan variasi status PPPoE, mode ONT, dan nilai RX Power. Hasilnya menunjukkan bahwa mesin diagnosis mengikuti prioritas *rule base*. Status LOS selalu dikategorikan sebagai *Critical*, sedangkan *power failure*, *dying gasp*, *mode bridge*, dan redaman tinggi dikategorikan sebagai *Warning*.

Tabel 6 Eksperimen Rule-Based Classification

Perlakuan	Status PPPoE	Mode	RX Power	Output Diagnosis	Keparahan
P1	LOS	Routing	-31 dBm	Loss of Signal / kabel putus	Critical
P2	Power Failure	Routing	0 / kosong	Gangguan daya pelanggan	Warning
P3	Online	Bridge	-22 dBm	Kesalahan mode perangkat	Warning
P4	Online	Routing	-29 dBm	Sinyal lemah / redaman tinggi	Warning
P5	Online	Routing	-21 dBm	Kondisi koneksi baik	Normal
P6	Isolir/Admin Disable	Routing	-20 dBm	Status administratif, bukan gangguan fisik	Normal/Administratif

3.5 Hasil Eksperimen Smart-Logging

Smart-logging diuji untuk memastikan bahwa sistem tidak menyimpan data redundan ketika status perangkat tidak berubah. Berdasarkan perhitungan 1.621 unit dengan 72 siklus *polling* per hari, skenario tanpa filter dapat menghasilkan 116.712 baris log per hari. Setelah *change-based logging* diterapkan, data observasi riil hanya sekitar 1.750 baris per hari.

Tabel 7 Eksperimen *Smart-Logging* dan Efisiensi *Database*

Perlakuan	Variabel yang Diset	Variabel yang Diamati	Hasil	Kesimpulan
P1	Status ONT tetap Normal pada dua siklus berurutan	Operasi INSERT	Tidak ada INSERT baru	Redundansi berhasil dicegah
P2	Status berubah Normal ke Warning	Operasi INSERT dan isi log	INSERT dibuat dengan status baru	Transisi terekam
P3	Status berubah Critical ke Normal	Log pemulihan	INSERT dibuat dan Telegram recovery dikirim	Pemulihan terekam
P4	Tidak ada perubahan selama 24 jam	Heartbeat log	Minimal satu log harian tetap ditulis	Riwayat tetap valid
P5	Direct save dibandingkan smart-logging	Jumlah baris per hari	116.712 baris menjadi ± 1.750 baris	Efisiensi 98,50%

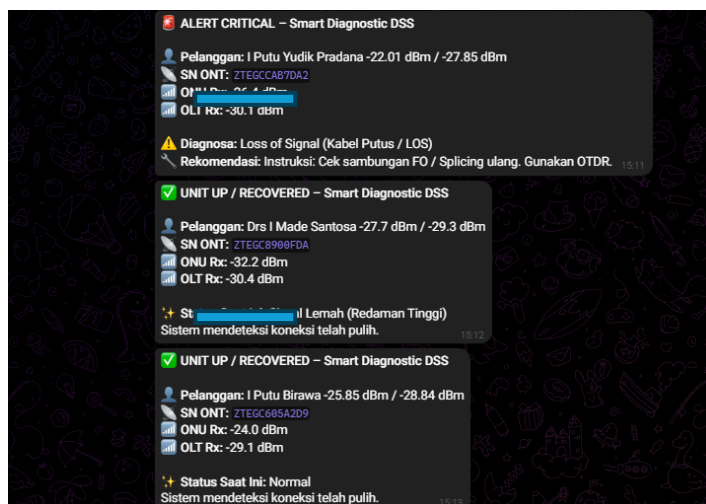
3.6 Visualisasi *Dashboard* dan Notifikasi

Dashboard utama menampilkan jumlah perangkat yang terpantau, distribusi status *Normal*, *Warning*, dan *Critical*, grafik RX Power, serta tabel log diagnosis. Pada data implementasi, *dashboard* menampilkan 1.621 pelanggan, terdiri dari 1.419 *Normal*, 198 *Warning*, dan 4 *Critical*. Tampilan ini membantu NOC membaca prioritas gangguan tanpa harus memeriksa seluruh data mentah secara manual.



Gambar 3 Tampilan Antarmuka *Dashboard* Utama Sistem

Notifikasi Telegram dikirim saat terjadi event penting, terutama gangguan kritis dan pemulihan layanan. Pesan notifikasi memuat nama pelanggan, *serial number* ONT, nilai ONU Rx dan OLT Rx, diagnosis, serta rekomendasi tindakan teknis.



Gambar 4 Push Notification *Alert Critical* dan Pemulihan pada Telegram

3.7 Hasil Eksperimen *Dashboard* dan Telegram Bot

Tabel 8 Eksperimen Visualisasi *Dashboard*

Perlakuan	Variabel yang Diset	Variabel yang Diamati	Hasil Pengamatan	Kesimpulan
P1	Dataset berisi Normal, Warning, Critical	Kartu statistik	<i>Dashboard</i> menampilkan total dan distribusi status	Berhasil
P2	Perubahan status pada log terbaru	Tabel diagnosis log	Baris log terbaru muncul sesuai status	Berhasil

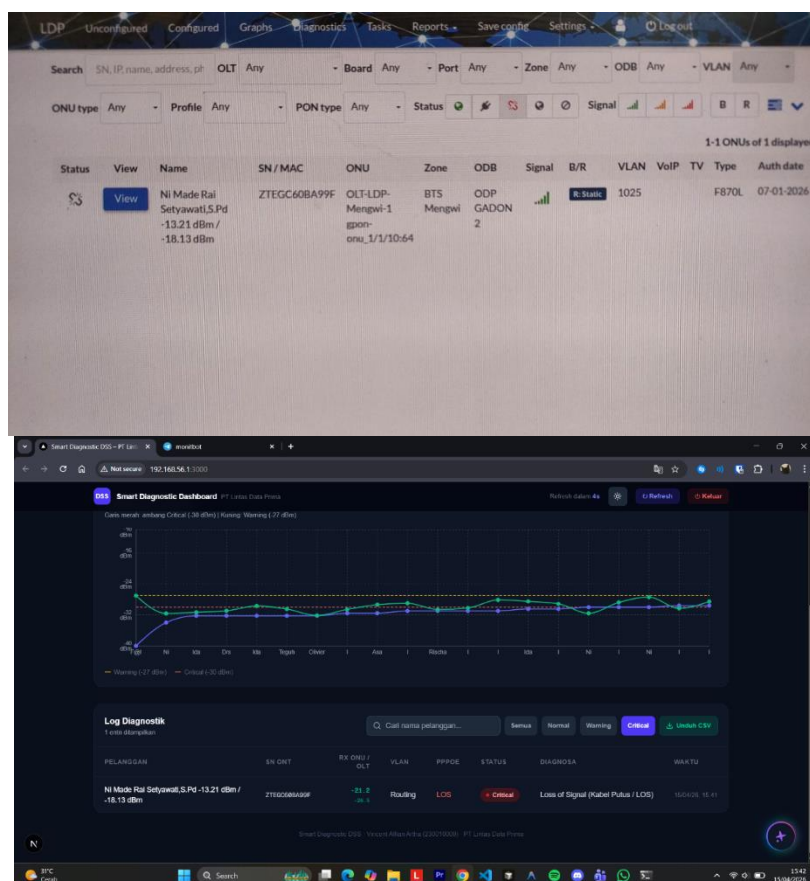
P3	Perubahan nilai RX Power		Grafik redaman	Grafik memetakan nilai dBm secara dinamis	Berhasil
P4	Auto-refresh detik	60	Sinkronisasi data <i>dashboard</i>	Tampilan memperbarui data tanpa reload manual	Berhasil

Tabel 9 Eksperimen Notifikasi Telegram Bot

Perlakuan	Event yang Diset	Variabel yang Diamati	Hasil Pengamatan	Kesimpulan
P1	Perangkat berubah Normal ke LOS	Isi pesan dan tujuan grup	Alert Critical terkirim dengan identitas pelanggan dan rekomendasi OTDR	Berhasil
P2	Perangkat berubah Critical ke Normal	Pesan recovery	Pesan UNIT UP / RECOVERED terkirim	Berhasil
P3	Status tidak berubah	Jumlah pesan baru	Tidak ada pesan duplikat	Berhasil

3.8 Validasi Visual Diagnosis

Validasi komparatif dilakukan dengan membandingkan status pada platform SmartOLT dengan hasil visualisasi *dashboard*. Sampel pelanggan yang terdeteksi LOS pada SmartOLT diterjemahkan oleh sistem menjadi kategori *Critical* dengan diagnosis *Loss of Signal*. Konsistensi ini menunjukkan bahwa *rule base* telah diterapkan sesuai fakta teknis pada sumber data.

**Gambar 5** Validasi Komparatif Akurasi Diagnosis Antara SmartOLT dan *Dashboard*

4. Pembahasan

Hasil eksperimen menunjukkan bahwa sistem yang dibangun telah menjawab tujuan utama penelitian, yaitu merancang SPK monitoring GPON yang mampu melakukan akuisisi data,

klasifikasi diagnosis, penyimpanan log, visualisasi *dashboard*, dan pengiriman notifikasi secara terintegrasi. Kemampuan sistem memproses 1.621 data ONT dalam satu siklus menunjukkan bahwa proses monitoring tidak lagi bergantung pada pemeriksaan manual satu per satu, tetapi sudah diarahkan menjadi proses berbasis data yang dapat dipantau secara berkala. Dengan demikian, luaran sistem tidak hanya berupa tampilan status perangkat, melainkan informasi diagnosis yang membantu tim NOC menentukan prioritas penanganan gangguan. Temuan ini sejalan dengan penelitian Cakir dkk. yang menunjukkan bahwa pemantauan data secara *real-time* dapat mendukung pengambilan keputusan operasional yang lebih cepat dan berbasis informasi [16].

Pada aspek akuisisi data, rata-rata durasi pemrosesan sebesar 485 detik masih berada di bawah interval *polling* 20 menit. Kondisi ini menunjukkan bahwa rancangan *polling* memiliki ruang waktu yang cukup sebelum siklus berikutnya berjalan, sehingga risiko *process overlapping* relatif kecil. Hasil tersebut penting karena sistem bergantung pada SmartOLT v2 API sebagai sumber data pihak ketiga. Oleh sebab itu, pemilihan interval *polling* tidak hanya mempertimbangkan kebutuhan *real-time*, tetapi juga stabilitas proses ETL dan batasan akses API. Temuan ini sejalan dengan konsep integrasi REST API pada aplikasi monitoring yang menekankan pentingnya komunikasi data lintas platform secara stabil dan terstruktur [4]. Selain itu, penggunaan FastAPI sebagai *backend* mendukung kebutuhan pemrosesan layanan API karena *framework* ini memang relevan untuk pengembangan *web service* dan pengujian performa API [11], [12].

Pada aspek diagnosis, *rule-based classification* sesuai digunakan pada domain GPON karena parameter gangguan jaringan bersifat teknis, terukur, dan dapat dirumuskan dalam aturan *If-Then*. Nilai RX Power, status PPPoE, mode perangkat, serta kondisi LOS merupakan fakta input yang dapat langsung dipetakan ke dalam kategori *Critical*, *Warning*, dan Normal/Administratif. Penggunaan ambang batas RX Power di bawah -27 dBm sebagai indikasi redaman tinggi juga konsisten dengan pembahasan kualitas redaman FTTH/GPON pada penelitian sebelumnya dan standar lapisan fisik GPON yang menempatkan kualitas sinyal optik sebagai faktor penting dalam performa jaringan [1], [10]. Dengan demikian, hasil klasifikasi tidak bersifat spekulatif, tetapi diturunkan dari parameter teknis yang dapat diamati pada sumber data.

Hasil klasifikasi juga menunjukkan kontribusi utama sistem dibandingkan monitoring konvensional. Platform monitoring dasar umumnya menampilkan status perangkat sebagai data mentah, sedangkan sistem ini menambahkan lapisan interpretasi melalui basis aturan. Status LOS langsung diprioritaskan sebagai *Critical* karena mengindikasikan hilangnya sinyal optik, sedangkan status isolir atau admin *disable* dipisahkan sebagai status administratif agar tidak diperlakukan sebagai gangguan fisik. Pemisahan ini menjawab permasalahan operasional PT Lintas Data Prima, yaitu kebutuhan untuk membedakan gangguan teknis jaringan dari kondisi administratif pelanggan. Pada titik ini, sistem berperan sebagai SPK karena membantu operator mengambil keputusan awal secara lebih objektif berdasarkan data dan aturan yang jelas [7], [8], [9].

Jika dibandingkan dengan penelitian terdahulu, penelitian Mukiman dan Masruri berfokus pada aplikasi laporan redaman FTTH berbasis GPON untuk membantu pencatatan dan analisis nilai redaman [1], sedangkan penelitian Kusuma dkk. membahas implementasi infrastruktur FTTH dengan dukungan Mikhmon pada lingkungan ISP RT/RW Net [3]. Penelitian Fernández dkk. membahas teknik monitoring PON dari sisi pendekatan optik, dan penelitian Amin dkk. memanfaatkan Telegram Bot untuk distribusi informasi monitoring jaringan [5], [6]. Penelitian ini mendukung arah penelitian-penelitian tersebut, tetapi memberikan pengembangan baru berupa integrasi SmartOLT v2 API, *rule-based* diagnosis, *smart-logging*, *dashboard*, dan Telegram Bot dalam satu alur SPK. Dengan demikian, kebaruan sistem bukan hanya pada penggunaan teknologi, melainkan pada penyatuan proses akuisisi, klasifikasi, pencatatan, visualisasi, dan notifikasi untuk kebutuhan operasional NOC.

Pada aspek penyimpanan, *smart-logging* memberikan dampak signifikan terhadap efisiensi *database*. Tanpa filter, penyimpanan langsung setiap siklus dapat menghasilkan 116.712 baris per hari dari 1.621 ONT dengan 72 siklus *polling*. Setelah *change-based logging* diterapkan, data observasi riil menjadi sekitar 1.750 baris per hari atau berkurang 98,50%. Hasil ini menunjukkan bahwa sistem hanya menyimpan peristiwa yang bernilai diagnostik, seperti perubahan status dan heartbeat harian. Pendekatan ini relevan dengan penggunaan *database cloud* seperti Supabase PostgreSQL, karena penyimpanan historis harus menjaga keseimbangan antara kelengkapan data dan efisiensi kapasitas [13]. Kebutuhan akan strategi

pengumpulan dan penyimpanan yang efisien pada data monitoring berskala besar juga ditekankan oleh Suthakar dkk. [17].

Pada aspek antarmuka, *dashboard* berfungsi sebagai jembatan antara data teknis dan kebutuhan keputusan operator. Kartu statistik membantu NOC membaca distribusi Normal, *Warning*, dan *Critical* secara cepat, grafik RX Power membantu melihat kecenderungan kualitas sinyal, sedangkan tabel log diagnosis menyediakan jejak historis untuk evaluasi. Hal ini memperkuat fungsi SPK karena informasi yang ditampilkan tidak hanya berupa data mentah, tetapi sudah disusun menjadi ringkasan kondisi jaringan dan rekomendasi tindakan. Penggunaan Next.js juga mendukung kebutuhan *dashboard* modern karena *framework* tersebut relevan untuk peningkatan performa aplikasi web dan penyajian antarmuka yang responsif [14]. Secara konseptual, *dashboard* yang disesuaikan dengan tujuan dan kebutuhan pengguna dapat meningkatkan penyampaian insight serta mendukung proses pengambilan keputusan [18].

Pada aspek notifikasi, Telegram Bot terbukti berperan sebagai kanal distribusi informasi untuk event penting, terutama status *Critical* dan pemulihan ke Normal. Sistem tidak mengirim notifikasi untuk setiap perubahan status atau setiap siklus *polling*, sehingga risiko *alert fatigue* dapat dikurangi. Pola *event-driven* ini sesuai dengan kebutuhan teknisi lapangan karena pesan yang dikirim hanya memuat kejadian yang membutuhkan perhatian segera, seperti LOS atau pemulihan layanan. Temuan ini sejalan dengan penelitian Amin dkk. yang menunjukkan bahwa integrasi Telegram Bot dapat mempercepat penyampaian informasi monitoring jaringan kepada pengelola jaringan [6].

Validasi visual antara SmartOLT dan *dashboard* menunjukkan konsistensi hasil diagnosis, terutama pada sampel pelanggan yang terdeteksi LOS dan diklasifikasikan sebagai *Critical*. Konsistensi ini memperkuat validitas *rule base* karena sistem mampu menerjemahkan fakta pada sumber data menjadi *output* diagnosis yang sesuai. Selain itu, eksperimen fungsional yang dilakukan pada akuisisi data, klasifikasi, *smart-logging*, *dashboard*, dan Telegram Bot menunjukkan bahwa setiap fitur utama berjalan sesuai kriteria keberhasilan. Pola pengujian ini sesuai dengan prinsip *blackbox testing* yang menilai kesesuaian keluaran sistem terhadap fungsi yang dirancang tanpa harus melihat struktur internal program secara langsung [15].

Meskipun hasil penelitian menunjukkan sistem berjalan sesuai kebutuhan, terdapat beberapa keterbatasan yang perlu diperhatikan. Pertama, tingkat *real-time* sistem masih bergantung pada ketersediaan dan *rate-limit* SmartOLT v2 API. Kedua, akurasi diagnosis masih dipengaruhi oleh kualitas data input dan ketepatan *threshold* operasional yang digunakan. Ketiga, penelitian ini belum melakukan pengukuran longitudinal terhadap penurunan MTTR berdasarkan data tiket gangguan sebelum dan sesudah sistem diterapkan. Oleh karena itu, penelitian lanjutan dapat diarahkan pada integrasi SNMP *direct polling* ke perangkat OLT atau router, pengembangan *work order* otomatis, penerapan *retention policy database*, serta evaluasi kuantitatif terhadap dampak sistem terhadap MTTR dalam periode operasional yang lebih panjang.

5. Simpulan

Penelitian ini berhasil merancang dan membangun SPK monitoring jaringan GPON berbasis *rule-based classification* pada PT Lintas Data Prima Bali. Sistem mampu mengotomatisasi penarikan data ONT dari SmartOLT v2 API setiap 20 menit, memproses data melalui *backend* FastAPI, menyimpan log pada Supabase PostgreSQL, menampilkan informasi diagnosis pada *dashboard* Next.js, dan mengirim notifikasi melalui Telegram Bot.

Hasil eksperimen fungsional menunjukkan bahwa lima fitur utama sistem berjalan sesuai kebutuhan. Sistem mampu memproses 1.621 data ONT dengan rata-rata durasi 485 detik per siklus, mengklasifikasikan status perangkat menjadi *Critical*, *Warning*, dan Normal/Administratif berdasarkan *rule base*, serta mengoptimalkan penyimpanan log hingga 98,50% melalui *smart-logging*. Validasi visual juga menunjukkan konsistensi antara status pada SmartOLT dan hasil diagnosis pada *dashboard*.

Kebaruan sistem terletak pada kemampuan membedakan gangguan fisik GPON dari status administratif pelanggan secara otomatis, sekaligus mengurangi penumpukan data historis menggunakan *change-based logging*. Pengembangan berikutnya disarankan menambahkan *direct polling* ke perangkat OLT atau router, integrasi *work order*, *retention policy database*, serta evaluasi kuantitatif terhadap penurunan MTTR.

Daftar Referensi

- [1] H. Haryono, K. Mukiman, and F. H. Masruri, "Rancang Bangun Aplikasi Report Redaman pada Jaringan Fiber to The Home Berteknologi Gigabit Passive Optical Network di PT Telkom Akses Cibitung," *JASIEK (Jurnal Aplikasi Sains, Informasi, Elektronika dan Komputer)*, vol. 5, no. 2, pp. 157-164, Dec. 2023, doi: 10.26905/jasiek.v5i2.11580.
- [2] S1 Teknik Telekomunikasi - Telkom University Jakarta, "Gigabit Capable Passive Optical Network (GPON), Apr. 27, 2023." <https://bte-jkt.telkomuniversity.ac.id/gigabit-capable-passive-optical-network-gpon/>.
- [3] D. Kusuma, S. Suryayusra, I. Z. Yadi, and D. Irawan, "Implementasi Mikhmon dengan Infrastruktur Fiber to the Home Sebagai Internet Service Provider RT RW Net Untuk Sekolah dan Rumah Penduduk," *Journal of Information System Research (JOSH)*, vol. 5, no. 4, pp. 1361–1370, Jul. 2024, doi: 10.47065/josh.v5i4.5357.
- [4] S. Widyaningtyas and T. Wahyono, "Implementasi Rest Api Menggunakan Retrofit Pada Aplikasi Monitoring Grooming Berbasis Android," *IT-Explore: Jurnal Penerapan Teknologi Informasi dan Komunikasi*, vol. 3, no. 2, pp. 147–161, 2024.
- [5] M. P. Fernández, L. A. Bulus-Rossini, and P. A. Costanzo-Caso, "PON Monitoring Technique Using Single-FBG Encoders and Wavelength-to-Time Mapping," *IEEE Photonics Technology Letters*, vol. 31, no. 21, pp. 1745–1748, Nov. 2019, doi: 10.1109/LPT.2019.2944528.
- [6] M. Amin, M. Hasbi, B. Widada, and B. W. Yudanto, "Implementasi Monitoring Jaringan Internet Pada Mikrotik Menggunakan Notifikasi Bot Telegram Di SMK," *Jurnal Teknologi Informasi dan Komunikasi (TIKomsin)*, vol. 13, no. 1, p. 67, Aug. 2025, doi: 10.30646/tikomsin.v13i1.997.
- [7] A. Zumarniansyah, R. Ardianto, Y. Alkhalifi, and Q. N. Azizah, "Penerapan Sistem Pendukung Keputusan Penilaian Karyawan Terbaik dengan Metode Simple Additive Weighting," *Jurnal Sistem Informasi*, vol. 10, no. 2, pp. 75-81, Aug. 2021, doi: 10.51998/jsi.v10i2.419.
- [8] XengineXperts, "Rule-based systems | XengineXperts," <https://www.xenginexperts.com/rule-based-systems>.
- [9] A. Halim and F. A. Putra, "Sistem Pakar Diagnosa Penyakit Gigi dan Mulut Menggunakan Metode Forward Chaining Berbasis Rule Based System," *Jurnal Teknologi Informasi dan Komputer*, vol. 1, no. 3, pp. 31-38, Nov. 2025.
- [10] ITU-T, "Recommendation ITU-T G.984.2: Gigabit-capable Passive Optical Networks (GPON): Physical Media Dependent (PMD) Layer Specification," International Telecommunication Union, pp. 1-50, Aug. 2019. [Online]. Available: <http://handle.itu.int/11.1002/1000/11>.
- [11] A. N. Safitri and I. Harkespan, "Pengembangan Web Service Menggunakan Framework Fastapi Untuk Meningkatkan Kemudahan Integrasi Sistem Informasi Akademik Multiplatform," *Jurnal Teknoif Teknik Informatika Institut Teknologi Padang*, vol. 12, no. 2, pp. 149–157, Oct. 2024, doi: 10.21063/jtif.2024.v12.2.149-157.
- [12] S. Santiago and B. Benisius, "Testing performa framework Express.js, Gin, dan FastAPI menggunakan API dan JMeter," *Journal of Information System, Applied, Management, Accounting and Research*, vol. 9, no. 3, p. 1219, Jul. 2025, doi: 10.52362/jisamar.v9i3.1986.
- [13] M. R. Nugraha, D. Galih, R. Rezkika, and W. Haryono, "Rancang Bangun Aplikasi Manajemen Proyek Berbasis Web Menggunakan React dan Supabase pada Yayasan Sa'adatul Atholibin," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 14, no. 1, pp. 12-20, Jan. 2026, doi: 10.23960/jitet.v14i1.8741.
- [14] D. W. Widarti and M. K. Akbar, "Peningkatan Performa Aplikasi Web SIAKAD Menggunakan Next.js di Puskordat Stimata," *Journal of Information System Management (JOISM)*, vol. 7, no. 2, pp. 218-226, Jan. 2026, doi: 10.24076/joism.2026v7i2.2453.
- [15] Y. F. Achmad and A. Yulfitri, "Pengujian Sistem Pendukung Keputusan Menggunakan Black Box Testing Studi Kasus E-Wisudawan di Institut Sains dan Teknologi Al-Kamal," *JIK: Jurnal Ilmu Komputer*, vol. 5, no. 1, pp. 42-51, Jun. 2020, doi: 10.47007/komp.v5i01.4615.
- [16] A. Cakir, O. Akin, H. F. Deniz, and A. Yilmaz, "Enabling real time big data solutions for manufacturing at scale," *J. Big Data*, vol. 9, no. 1, p. 118, 2022, doi: 10.1186/s40537-022-00672-6.

- [17] U. Suthakar, L. Magnoni, D. R. Smith, A. Khan, and J. Andreeva, "An efficient strategy for the collection and storage of large volumes of data for computation," *J. Big Data*, vol. 3, no. 1, p. 21, 2016, doi: 10.1186/s40537-016-0056-1.
- [18] A. Vazquez-Ingelmo, F. J. Garcia-Penalvo, and R. Theron, "Information Dashboards and Tailoring Capabilities - A Systematic Literature Review," *IEEE Access*, vol. 7, pp. 109673–109688, 2019, doi: 10.1109/ACCESS.2019.2933472.