Jurnal Ilmiah Teknik Informatika dan Komunikasi
Volume. 6, Nomor. 3, November 2026
E-ISSN : 2827-7945 ; P-ISSN : 2827-8127; Hal. 260-273
DOI: https://doi.org/10.55606/juitik.v6i3.2739
Tersedia: https://journal.sinov.id/index.php/juitik

Perancangan Sistem Informasi Monitoring dan *Controlling* Server Berbasis Web Menggunakan *Framework Express* JS pada PT. Fintek Digital Nusantara

Moh. Khabibur Rokhman [1*], Jarot Dian Susatyono [2], Sumaryanto [3]
[1-3] Program Studi Teknik Informatika, Universitas Sains dan Teknologi Komputer, Indonesia

**Penulis Korespondensi : khabiburrokhman13@gmail.com*

Abstract*. PT. Fintek Digital Nusantara is a financial technology company whose operations depend heavily on the availability of servers and digital services. Server monitoring at the company was still performed manually through terminal commands and conventional recording, so failures were often detected late and troubleshooting became slow. This study aims to design and develop a web-based server monitoring and controlling information system using the ExpressJS framework integrated with Telegram notifications. The system was developed using a descriptive method consisting of requirements analysis, design, implementation, testing, and deployment. It checks service availability every five minutes through HTTP/HTTPS and TCP probes scheduled by node-cron, applies a failure-threshold mechanism to reduce false alarms, stores monitoring history in MongoDB, and automatically sends downtime and recovery alerts through the Telegram Bot API. When a service is unhealthy, a maintenance flag holds new business transactions as pending until recovery, after which they can be synchronized to paid. Black-box testing showed that authentication, master-data management, automatic checking, notification delivery, pending-transaction handling, and reporting worked as intended. The system presents uptime percentage, incident count, and downtime estimation for each service, enabling centralized real-time monitoring and faster incident response than manual checking.*

Keywords: *ExpressJS; MongoDB; Server Monitoring; Telegram Notification; Uptime.*

Abstrak. PT. Fintek Digital Nusantara merupakan perusahaan teknologi finansial yang operasional bisnisnya sangat bergantung pada ketersediaan server dan layanan digital. Proses monitoring server di perusahaan masih dilakukan secara manual melalui perintah terminal dan pencatatan konvensional, sehingga gangguan sering terlambat terdeteksi dan penanganannya menjadi lambat. Penelitian ini bertujuan merancang dan membangun sistem informasi monitoring dan controlling server berbasis web menggunakan framework ExpressJS yang terintegrasi dengan notifikasi Telegram. Sistem dikembangkan dengan metode deskriptif yang meliputi tahapan analisis kebutuhan, desain, implementasi, pengujian, dan deployment. Sistem melakukan pengecekan ketersediaan layanan secara berkala setiap lima menit melalui probe HTTP/HTTPS dan TCP yang dijadwalkan dengan node-cron, menerapkan mekanisme ambang kegagalan (failure threshold) untuk mengurangi false alarm, menyimpan riwayat monitoring pada basis data MongoDB, serta mengirimkan peringatan downtime dan pemulihan secara otomatis kepada administrator dan penanggung jawab melalui Telegram Bot API. Ketika layanan tidak sehat, flag maintenance menahan transaksi bisnis baru sebagai pending, lalu transaksi tersebut dapat disinkronkan menjadi paid setelah layanan pulih. Hasil pengujian black-box menunjukkan seluruh fungsi, meliputi autentikasi, pengelolaan data master, pengecekan otomatis, pengiriman notifikasi, penahanan transaksi pending, dan pelaporan, berjalan sesuai rancangan. Sistem menyajikan persentase uptime, jumlah insiden, dan estimasi downtime setiap layanan, sehingga pemantauan menjadi terpusat, real-time, dan penanganan gangguan lebih cepat dibandingkan pengecekan manual.

Kata kunci: ExpressJS; Ketersediaan Layanan; MongoDB; Monitoring Server; Notifikasi Telegram.

1. LATAR BELAKANG

Sistem monitoring banyak diterapkan dalam dunia industri untuk memantau dan mengetahui kinerja suatu proses, pekerjaan, maupun aplikasi yang sedang berjalan. Seiring dengan perkembangan teknologi yang semakin pesat, sistem monitoring hadir sebagai salah satu solusi yang dapat memberikan kemudahan dalam melakukan pemantauan secara efektif

Naskah Masuk: 20 Agustus 2026; Revisi: 28 September 2026; Diterima: 06 Oktober 2026; Terbit: 01 November 2026

dan efisien (Adeguno & Septiyanti, 2026). Penerapan sistem monitoring dapat dilakukan dengan berbagai metode, salah satunya dengan memanfaatkan jaringan internet melalui konsep Internet of Things (IoT) (Prima et al., 2026).

Perkembangan teknologi informasi dan komunikasi yang semakin pesat telah mendorong munculnya berbagai inovasi digital dalam kehidupan sehari-hari (Sutanto et al., 2026) termasuk dalam cara sistem monitoring diakses dan disajikan. Sistem monitoring kini tidak lagi terbatas pada aplikasi khusus di ruang server, tetapi dapat diakses melalui berbagai perangkat, mulai dari komputer hingga smartphone yang telah menjadi perangkat multifungsi (Muh Sabri et al., 2026). Teknologi berbasis web menjadi pilihan yang banyak digunakan karena bersifat lintas platform sehingga dapat dibuka dari sistem operasi apa pun, termasuk perangkat mobile berbasis Android (Setiawati et al., 2026), tanpa memerlukan proses instalasi khusus. Selain itu, pengembangannya dapat dilakukan dengan berbagai perangkat pengembangan yang tersedia secara gratis (Saragih & Siregar, 2026), sehingga relevan diterapkan pada sistem monitoring server yang perlu diakses dengan cepat oleh tim teknis kapan saja dan di mana saja.

PT. Fintek Digital Nusantara merupakan perusahaan yang bergerak di bidang layanan teknologi finansial yang memanfaatkan sistem berbasis server untuk mendukung operasional bisnis, pengelolaan data transaksi, serta layanan kepada pelanggan. Ketersediaan dan keandalan server menjadi aspek krusial karena berkaitan langsung dengan kontinuitas layanan dan kepercayaan pengguna. Dalam operasionalnya, PT. Fintek Digital Nusantara mengelola berbagai server dan perangkat jaringan yang mendukung aplikasi dan sistem layanan digital. Proses monitoring terhadap kondisi server dan perangkat jaringan menjadi kebutuhan penting guna memastikan sistem berjalan optimal dan meminimalkan risiko downtime. Namun, apabila proses monitoring masih dilakukan secara manual, seperti pengecekan status server melalui perintah ping pada terminal atau pencatatan kondisi server secara konvensional, maka hal tersebut berpotensi menimbulkan berbagai kendala, antara lain proses pengecekan yang memakan waktu, kurangnya efisiensi dalam pendataan, serta keterlambatan dalam mendeteksi gangguan sistem.

Permasalahan tersebut dapat berdampak pada lambatnya penanganan gangguan (troubleshooting), risiko kehilangan data, serta terganggunya layanan kepada pengguna. Dalam industri finansial digital, keterlambatan atau gangguan sistem dapat berimplikasi pada kerugian finansial dan menurunnya tingkat kepercayaan pelanggan. Oleh karena itu, diperlukan suatu sistem monitoring server yang mampu memberikan informasi secara real-time, terpusat, dan mudah diakses oleh tim teknis.

Sistem informasi monitoring server berbasis web menjadi salah satu solusi yang relevan untuk menjawab kebutuhan tersebut. Dengan antarmuka web yang responsif, tim teknis dapat memantau kondisi server kapan saja dan di mana saja melalui berbagai perangkat (Harun & Munaldi, 2026). Sistem ini diharapkan mampu memberikan notifikasi secara langsung apabila terjadi gangguan, menampilkan status server secara real-time, serta menyimpan riwayat monitoring sebagai bahan evaluasi dan pengambilan keputusan. Integrasi kanal pesan instan seperti Telegram juga terbukti mempercepat penyampaian status gangguan kepada teknisi tanpa harus membuka dashboard secara terus-menerus (Wiriaatmadja & Ratama, 2022).

Penelitian (Azizi et al., 2022) yang mengembangkan sistem monitoring server berbasis Android untuk dinas di kantor pemerintahan menunjukkan bahwa aplikasi yang dikembangkan termasuk dalam kategori layak dan dapat digunakan dengan baik dalam memantau server secara real-time. Aplikasi membantu petugas dalam menghadapi permasalahan server yang down dan meningkatkan kinerja pemantauan. Hidayat dan Kurniawan (2020) mengembangkan sistem monitoring jaringan berbasis web yang memanfaatkan protokol ICMP dan HTTP untuk mendeteksi ketersediaan layanan server serta menyimpan data histori gangguan ke dalam basis data. Rahmawati dan Nugroho (2022) menekankan bahwa tingkat availability yang tinggi sangat penting dalam menjaga kepercayaan pengguna pada organisasi yang bergantung pada sistem transaksi daring. Sementara itu, (Arsandi & Syaripudin, 2024) menunjukkan bahwa integrasi monitoring berbasis web dengan notifikasi Telegram memudahkan tim teknis menangani gangguan meskipun berada di luar kantor.

Berdasarkan penelitian terdahulu yan sudah diuraikan peredaann dengan penelitian yang dilakukan yaitu jika dalam penelitian sebelumnya hanya berfokus pada penyajian status perangkat atau bergantung pada perangkat lunak monitoring siap pakai. Hasil pengembangan sistem monitoring yang pernah dikembangkan dalam penelitian terdahulu masih terdapat kesenjangan pada sistem yang dirancang khusus untuk operasional perusahaan, terintegrasi dengan notifikasi proaktif multi-penerima, dilengkapi mekanisme ambang kegagalan untuk menekan false alarm, serta mampu menahan transaksi bisnis ketika layanan tidak sehat. Oleh karena itu, penelitian ini mengembangkan sistem informasi monitoring dan controlling server berbasis web menggunakan framework ExpressJS yang terintegrasi dengan notifikasi Telegram, sehingga gangguan dapat terdeteksi lebih dini, transaksi baru ditahan sebagai pending, dan ditindaklanjuti lebih cepat setelah layanan pulih.

Berdasarkan penjelasan latar belakang di atas maka pertanyaan penelitian dalam penelitian ini adalah 1) bagaimana pengembangan Perancangan Sistem Informasi Monitoring dan Controlling Server Berbasis Web Menggunakan Framework ExpressJS pada PT. Fintek

Digital Nusantara? 2) bagaimana kelayakan hasil pengembangan Perancangan Sistem Informasi Monitoring dan Controlling Server Berbasis Web Menggunakan Framework ExpressJS pada PT. Fintek Digital Nusantara?

2. KAJIAN TEORITIS

Sistem Monitoring Server dan Ketersediaan Layanan

Monitoring server merupakan aktivitas pemantauan kondisi server dan layanan yang berjalan di atasnya secara berkesinambungan, meliputi status hidup atau mati, waktu respons, serta kode status yang dikembalikan oleh layanan. Tujuan utamanya adalah menjaga tingkat ketersediaan (availability), yaitu perbandingan antara waktu layanan beroperasi normal dengan total waktu pengamatan. Semakin tinggi nilai availability, semakin andal layanan tersebut bagi penggunanya (Rahmawati & Nugroho, 2022). Pada organisasi yang bergantung pada transaksi daring seperti perusahaan teknologi finansial, downtime yang tidak segera tertangani berdampak langsung pada kerugian finansial dan menurunnya kepercayaan pelanggan.

Beberapa penelitian terdahulu telah mengembangkan sistem sejenis. Azizi et al. (2022) membangun aplikasi monitoring server berbasis Android untuk instansi pemerintahan dan terbukti layak digunakan untuk pemantauan secara real-time. Hidayat dan Kurniawan (2020) merancang sistem monitoring jaringan berbasis web dengan protokol ICMP dan HTTP yang mampu menampilkan status perangkat serta menyimpan histori gangguan. Adeguno dan Septiyanti (2026) memanfaatkan sistem informasi berbasis web untuk pencatatan dan pemantauan aktivitas pada ruang server. Nugraha dan Ratama (2022) serta Arsandi dan Syaripudin (2024) mengintegrasikan perangkat monitoring jaringan dengan notifikasi Telegram agar tim teknis memperoleh peringatan secara real-time. Dari kajian tersebut terlihat bahwa sebagian besar penelitian berfokus pada penyajian status perangkat atau pada platform monitoring siap pakai, namun belum banyak yang merancang sistem ringan berbasis ExpressJS dengan mekanisme ambang kegagalan dan laporan uptime otomatis. Celah inilah yang diisi oleh penelitian ini.

Teknologi Pendukung

ExpressJS adalah framework aplikasi web minimalis di atas runtime Node.js yang menyediakan mekanisme routing, middleware, dan pengelolaan permintaan HTTP secara ringkas, sehingga cocok untuk membangun aplikasi dashboard dan layanan REST dengan cepat. Pemisahan antara antarmuka pengelolaan dan proses pemantauan yang berjalan di latar belakang sejalan dengan gagasan arsitektur layanan yang longgar terikat, di mana setiap komponen dapat dikembangkan dan dijalankan secara mandiri (Newman, 2021). MongoDB

digunakan sebagai basis data NoSQL berorientasi dokumen yang fleksibel terhadap perubahan skema dan efisien untuk menyimpan data log pengecekan yang bertambah terus-menerus.

Untuk penjadwalan, digunakan pustaka node-cron yang menjalankan tugas berdasarkan ekspresi cron, dalam penelitian ini setiap lima menit. Adapun Telegram Bot API dimanfaatkan sebagai kanal notifikasi karena bersifat gratis, mendukung pengiriman pesan ke banyak penerima, serta dapat diterima secara real-time pada perangkat mobile tanpa perlu membangun aplikasi khusus (Wiriaatmadja & Ratama, 2022). Pendekatan serupa telah digunakan pada sistem monitoring jaringan yang mengirimkan status up dan down langsung ke grup teknisi (Nugraha & Ratama, 2022).

3. METODE PENELITIAN

Penelitian ini menggunakan metode penelitian deskriptif statistik. Deskriptif statistik yaitu mendeskripsikan hasil penelitian selain dengan gambaran umum obyek penelitian juga mendeskripsikan analisis statistik berupa angka-angka dasar seperti dengan persentase. Desrkipsi statistik ini digunakan untuk mendeskripsikan kelayakan dan kesesuaian produk yang dinilai melalui dari validasi ahli. Fakta dan karakteristik objek dan subjek yang diteliti secara tepat. Dalam rekayasa perangkat lunak, pendekatan deskriptif sering dipadukan dengan tahapan rekayasa sistem agar hasil rancangan dapat ditelusuri dari kebutuhan hingga pengujian (Pressman & Maxim, 2020). Tahapan yang digunakan dalam penelitian ini meliputi analisis kebutuhan, perancangan, implementasi, pengujian, dan deployment.

Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk merumuskan data, fungsi, dan alur kerja yang diperlukan sistem. Kebutuhan yang diidentifikasi meliputi data server dan layanan, fitur pengelolaan data master, mekanisme pengecekan berkala, kanal notifikasi, controlling transaksi pending, serta penyajian log dan laporan uptime. Hasil analisis menjadi acuan spesifikasi fungsional pada tahap perancangan.

Perancangan

Tahap perancangan menggambarkan bagaimana sistem dibangun, termasuk arsitektur perangkat lunak, spesifikasi perangkat keras pendukung, dan alur data antara dashboard, worker monitoring, basis data, kanal notifikasi, serta aplikasi bisnis yang membaca flag maintenance. Perancangan juga menetapkan jenis probe (HTTP, HTTPS, dan TCP), interval pengecekan, serta aturan ambang kegagalan.

Implementasi

Implementasi merupakan proses pembuatan sistem berdasarkan hasil analisis dan perancangan. Aplikasi dashboard dibangun dengan ExpressJS dan Pug, autentikasi menggunakan sesi dan bcryptjs, data disimpan pada MongoDB, sedangkan worker monitoring dijalankan terpisah dengan node-cron dan terhubung ke Telegram Bot API. Sebagai pembuktian alur controlling, diimplementasikan aplikasi bisnis contoh yang menahan transaksi baru sebagai pending selama flag maintenance aktif dan menyediakan tombol sinkronisasi setelah maintenance selesai.

Pengujian dan Deployment

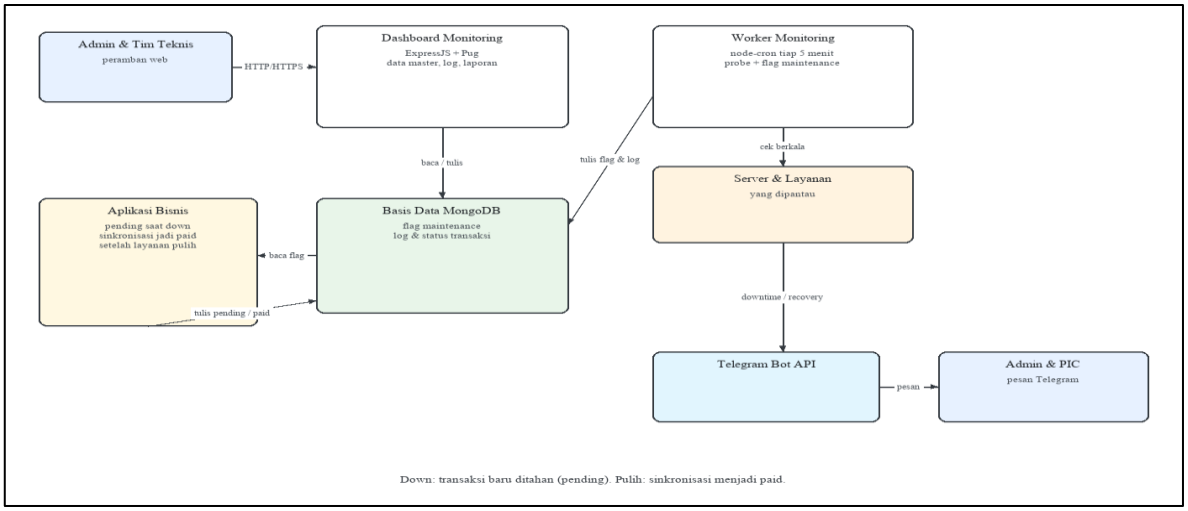
Pengujian dilakukan dengan metode black-box untuk menilai kesesuaian fungsi sistem terhadap kebutuhan tanpa menelusuri kode sumber (Yulianti dkk., 2022). Skenario uji mencakup autentikasi, pengelolaan data, pengecekan otomatis, notifikasi, penahanan transaksi pending, sinkronisasi setelah pemulihan, dan pelaporan. Setelah pengujian fungsional selesai, sistem disiapkan untuk digunakan oleh perusahaan. Tahap pemeliharaan berada di luar batasan penelitian ini.

4. HASIL DAN PEMBAHASAN

Pengembangan dan pengujian sistem dilakukan pada sebuah laptop dengan prosesor multi-core dan RAM 8 GB. Perangkat lunak yang digunakan meliputi Node.js sebagai runtime, framework ExpressJS dengan template engine Pug dan Bootstrap 5 untuk antarmuka, MongoDB sebagai basis data, pustaka node-cron untuk penjadwalan, bcryptjs untuk pengamanan kata sandi, serta Telegram Bot API untuk kanal notifikasi. Data yang digunakan berupa data server dan layanan yang dikelola PT. Fintek Digital Nusantara beserta riwayat pengecekan yang dihasilkan sistem.

Arsitektur Sistem

Sistem terdiri atas dashboard monitoring, worker, basis data MongoDB, dan aplikasi bisnis dapat digambarkan sebagai berikut:



Gambar 1. Arsitektur Sistem Monitoring Server.

Sistem terdiri atas dashboard monitoring, worker, basis data MongoDB, dan aplikasi bisnis, sebagaimana ditunjukkan pada Gambar 1. Dashboard mengelola data master, log, laporan, dan pengaturan. Worker melakukan pengecekan berkala, menulis hasil ke MongoDB, serta mengaktifkan flag maintenance ketika layanan down. Aplikasi bisnis membaca flag tersebut: transaksi baru ditahan sebagai pending selama gangguan, lalu disinkronkan menjadi paid setelah layanan pulih. Worker juga mengirim notifikasi downtime dan recovery melalui Telegram. Komponen pertama adalah aplikasi dashboard berbasis web yang menangani autentikasi pengguna, pengelolaan data master (pengguna, server, dan layanan), penyajian log, laporan, serta konfigurasi bot Telegram. Komponen kedua adalah worker monitoring yang berjalan sebagai proses terpisah dan melakukan pengecekan seluruh layanan aktif setiap lima menit. Pemisahan ini membuat proses pengecekan tetap berjalan meskipun dashboard tidak sedang diakses (Newman, 2021).

Validasi Ahli

Produk yang sudah dikembangkan sebelum dilakukan uji publik atau uji lapangan maka terlebih dahulu dilakukan validasi ahli. Validasi ahli dalam penelitian ini dilakukan pada dosen ahli IT Bapak Sumaryanto, S.Kom., M.Kom Universitas Sains Dan Tekonologi Komputer. Hasil validasi dapat diuraikan sebagai berikut:

Tabel 1. Hasil Valdiasi Ahli.

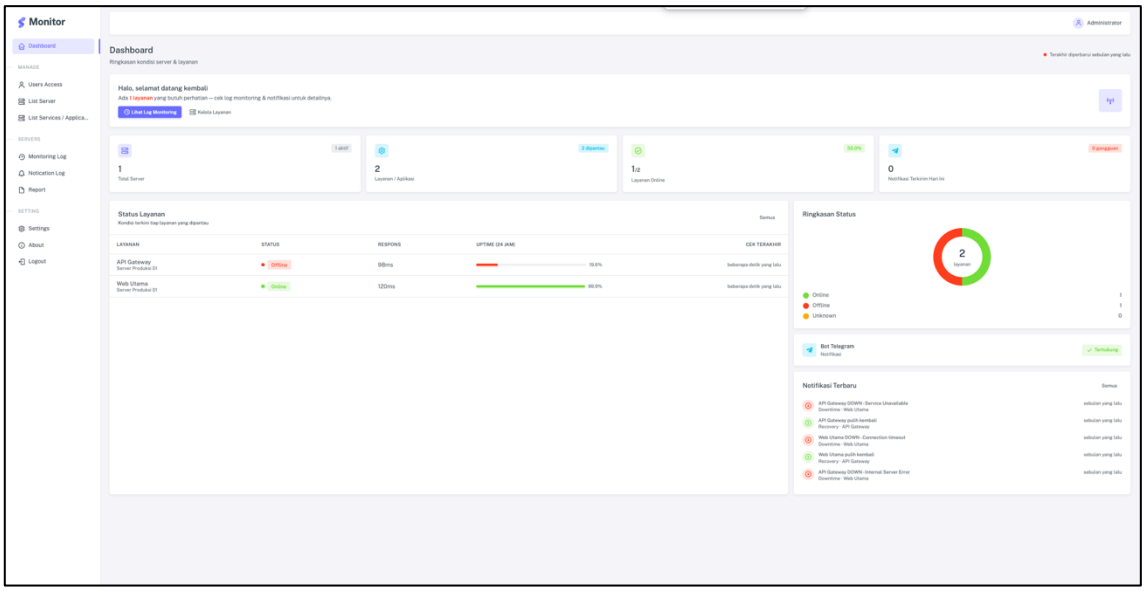
NO	INDIKATOR	Skor
1	Fitur monitoring server dapat berjalan sesuai dengan tujuan pengembangan	3
2	Fitur controlling server dapat menjalankan perintah sesuai fungsi yang dirancang	4
3	Sistem mampu memperbarui data monitoring sesuai mekanisme yang dirancang	3
4	Perintah controlling dapat dijalankan sesuai fungsi yang telah dirancang.	4
5	Pemilihan warna pada sistem sesuai dengan karakteristik aplikasi monitoring server	3
6	Penyajian data monitoring secara visual mudah dipahami.	4
7	Informasi penting pada sistem terlindungi dari akses yang tidak berwenang	3
8	Implementasi sistem mendukung proses controlling secara efektif	3

9	Fitur sistem sesuai dengan kebutuhan monitoring server perusahaan	4
10	Informasi yang ditampilkan relevan dengan kebutuhan pengelolaan server.	3
	JUMLAH SKOR	34/40

Berdasarkan hasil validasi ahli diketahui bahwa produk yang dikembangkan memperoleh skor 34 dari skor total 40. Hasil ini menunjukkan bahwa produk yang divalidasi ahli memperoleh kriteria dengan persentase sebesar 85% yang artinya termasuk dalam kategori sangat layak. Produk yang dikembangkan dalam penelitian ini sesudah dilakukan validasi ahli maka selanjutnya dilakukan uji lapangan.

Implementasi Sistem

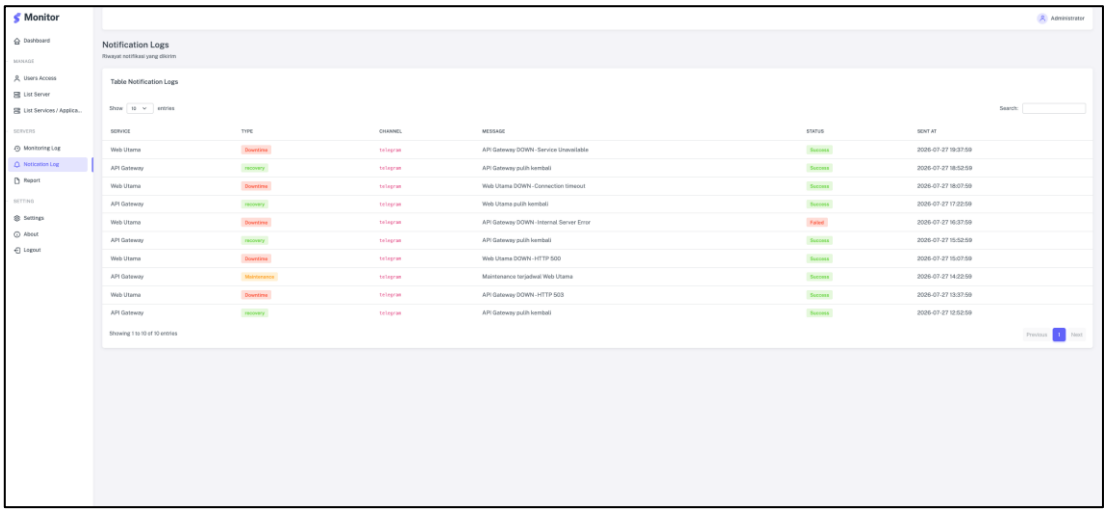
Halaman dashboard (Gambar 2) menyajikan ringkasan kondisi terkini berupa jumlah server dan layanan, persentase layanan online, status tiap layanan beserta uptime 24 jam terakhir, serta notifikasi terbaru yang dikirim ke Telegram. Pengelolaan data pengguna, server, dan layanan dilakukan melalui form modal dengan konfirmasi penghapusan untuk mencegah kesalahan operasional. Setiap layanan dapat dikonfigurasi jenis pengecekannya (HTTP, HTTPS, atau TCP), alamat URL atau port, ambang kegagalan, serta dapat diaktifkan atau dinonaktifkan dari pemantauan.



Gambar 2. Halaman Dashboard Monitoring.

Worker monitoring melakukan probe sesuai jenis layanan: untuk HTTP/HTTPS, layanan dinyatakan normal (up) apabila merespons dengan kode status di bawah 400 dalam batas waktu 10 detik; untuk TCP, keberhasilan pembukaan koneksi ke host dan port tujuan menjadi indikator layanan hidup. Setiap hasil pengecekan dicatat ke basis data beserta waktu respons, kode status, dan pesan gangguan. Layanan baru dinyatakan down setelah gagal berturut-turut sebanyak nilai ambang kegagalan yang dikonfigurasi, sehingga gangguan sesaat pada jaringan

tidak langsung memicu peringatan palsu. Saat layanan dinyatakan down, sistem mengaktifkan flag maintenance, menahan transaksi bisnis baru sebagai pending, dan mengirim notifikasi berisi nama layanan, server, URL, penyebab, serta waktu kejadian ke seluruh chat ID Telegram yang terdaftar. Ketika layanan kembali normal, flag dilepas dan sistem mengirim notifikasi pemulihan (Gambar 3). Seluruh pengiriman notifikasi turut dicatat berikut status berhasil atau gagalnya.

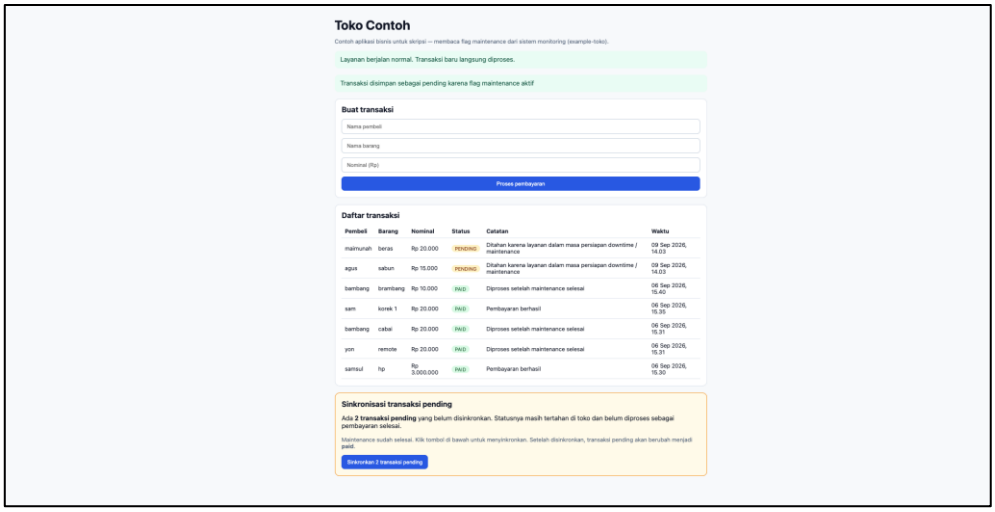


Gambar 3. Notifikasi *Downtime* dan *Recovery* Melalui Telegram.

Penahanan dan Sinkronisasi Transaksi Pending

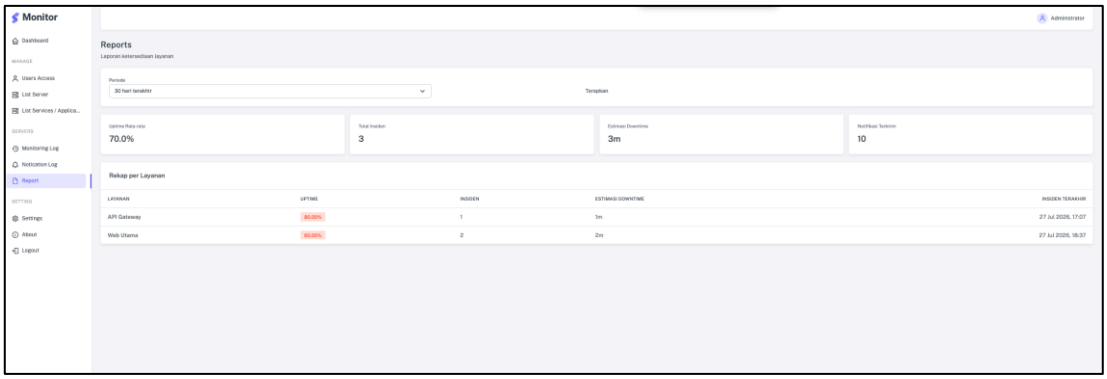
Setelah layanan dinyatakan down, worker tidak hanya mengirim notifikasi, tetapi juga menulis flag maintenance ke dokumen layanan pada MongoDB. Flag ini menandai bahwa aplikasi bisnis masih dapat diakses pengguna, namun kondisi layanan belum sehat. Pada aplikasi bisnis, setiap transaksi baru yang masuk selama flag aktif tidak diproses sebagai pembayaran selesai (paid), melainkan disimpan dengan status pending beserta catatan bahwa transaksi ditahan karena masa persiapan downtime atau maintenance. Pendekatan ini mencegah data transaksi seolah-olah berhasil padahal proses di belakangnya sedang terganggu.

Setelah jeda persiapan berakhir, worker dapat menjalankan pemulihan otomatis (misalnya restart proses melalui PM2) lalu melepas flag ketika health check kembali normal. Selama flag masih aktif, sinkronisasi pending ditolak agar tidak ada transaksi yang dipaksa selesai di tengah gangguan. Setelah maintenance selesai, bagian sinkronisasi transaksi pending ditampilkan pada aplikasi bisnis (Gambar 5). Operator menekan tombol sinkronisasi sehingga seluruh transaksi berstatus pending diubah menjadi paid dengan catatan bahwa transaksi diproses setelah layanan pulih. Dengan alur ini, monitoring tidak berhenti pada peringatan status, tetapi juga mengendalikan dampak gangguan terhadap transaksi.



Gambar 5. Tampilan Transaksi Pending dan Tombol Sinkronisasi pada Aplikasi Bisnis.

Sistem juga menyediakan laporan rekapitulasi per layanan dengan filter periode 7, 30, atau 90 hari (Gambar 4). Tingkat ketersediaan dihitung dengan persamaan Availability = (jumlah pengecekan berstatus up / jumlah seluruh pengecekan) × 100%. Selain persentase uptime, laporan menampilkan jumlah insiden, estimasi total downtime yang dihitung dari jumlah kegagalan dikalikan interval pengecekan, serta waktu insiden terakhir sebagai bahan evaluasi tim teknis.



Gambar 4. Halaman Laporan Uptime Per Layanan.

Pengujian Sistem

Pengujian dilakukan dengan metode black-box terhadap seluruh fungsi utama sistem. Metode ini memeriksa kesesuaian keluaran terhadap skenario masukan tanpa menelusuri struktur internal program (Myers dkk., 2012; Yulianti dkk., 2022). Ringkasan hasil disajikan pada Tabel 1. Skenario yang diuji meliputi autentikasi dengan kredensial valid dan tidak valid, proteksi halaman bagi pengguna yang belum login, operasi tambah, ubah, dan hapus pada data pengguna, server, dan layanan, pengecekan otomatis terjadwal, deteksi down berdasarkan ambang kegagalan, pengiriman notifikasi downtime dan recovery, penahanan transaksi pending, sinkronisasi setelah pemulihan, pencatatan log, penyusunan laporan uptime per

periode, serta uji koneksi bot Telegram dari halaman pengaturan. Seluruh skenario memberikan hasil sesuai yang diharapkan (Muslimin dkk., 2020; Putri dan Prasetiya, 2023).

Tabel 2. Hasil Pengujian Black-Box Sistem.

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil
1	Login dengan kredensial valid	Pengguna berhasil masuk ke dashboard	Sesuai
2	Login dengan kredensial tidak valid	Muncul pesan kesalahan dan akses ditolak	Sesuai
3	Akses halaman tanpa login	Pengguna dialihkan ke halaman login	Sesuai
4	Tambah, ubah, dan hapus data master	Perubahan data pengguna, server, dan layanan tersimpan	Sesuai
5	Pengecekan otomatis terjadwal	Worker memeriksa layanan aktif setiap lima menit	Sesuai
6	Deteksi down dengan ambang kegagalan	Status down ditetapkan setelah gagal sesuai ambang	Sesuai
7	Pengiriman notifikasi downtime	Pesan downtime terkirim ke chat ID Telegram terdaftar	Sesuai
8	Pengiriman notifikasi recovery	Pesan pemulihan terkirim setelah layanan kembali normal	Sesuai
9	Aktivasi flag maintenance saat down	Flag maintenance aktif dan dibaca aplikasi bisnis	Sesuai
10	Transaksi baru saat layanan down	Transaksi disimpan berstatus pending, bukan paid	Sesuai
11	Transaksi baru saat layanan sehat	Transaksi diproses berstatus paid	Sesuai
12	Sinkronisasi saat maintenance masih aktif	Sinkronisasi ditolak dan status pending tetap	Sesuai
13	Sinkronisasi setelah layanan pulih	Transaksi pending berubah menjadi paid	Sesuai
14	Penyusunan laporan uptime per periode	Uptime, jumlah insiden, dan downtime tampil sesuai filter	Sesuai
15	Uji koneksi bot Telegram	Pesan uji diterima pada akun Telegram tujuan	Sesuai

Hasil pengujian menunjukkan bahwa sistem mampu mendeteksi gangguan paling lambat satu siklus pengecekan (lima menit) sejak layanan bermasalah, jauh lebih cepat dan konsisten dibandingkan pengecekan manual yang bergantung pada inisiatif petugas. Penerapan ambang kegagalan terbukti menekan notifikasi palsu akibat gangguan jaringan sesaat, sementara flag maintenance menahan transaksi bisnis agar tidak dipaksa selesai saat layanan tidak sehat. Pencatatan riwayat memberikan jejak audit yang dapat dianalisis untuk perbaikan infrastruktur. Temuan ini menjawab permasalahan keterlambatan deteksi gangguan yang diuraikan pada bagian latar belakang.

Perbandingan dengan Penelitian Terdahulu

Dibandingkan dengan penelitian Azizi dkk. (2022) yang membangun aplikasi monitoring berbasis Android, sistem ini tidak mengharuskan pemasangan aplikasi khusus karena dashboard diakses melalui peramban dan peringatan dikirim melalui Telegram yang telah umum digunakan, sehingga distribusi dan pemeliharaan menjadi lebih sederhana. Terhadap

sistem monitoring web berbasis ICMP/HTTP milik Hidayat dan Kurniawan (2020), kebaruan penelitian ini terletak pada integrasi notifikasi proaktif multi-penerima, mekanisme failure threshold untuk menekan false alarm, serta laporan uptime dan estimasi downtime yang dihitung otomatis dari riwayat pengecekan. Berbeda pula dengan pendekatan yang memanfaatkan Zabbix atau Netwatch sebagai mesin pemantau (Nugraha dan Ratama, 2022; Arsandi dan Syaripudin, 2024; Wiriaatmadja dan Ratama, 2022), sistem ini dirancang sebagai aplikasi ringan yang dibangun sesuai kebutuhan operasional perusahaan.

Selaras dengan kajian Rahmawati dan Nugroho (2022) yang menekankan pentingnya availability bagi layanan digital, penelitian ini memberikan implementasi konkret berupa sistem terpadu yang mencakup deteksi, notifikasi, pencatatan, pelaporan, dan controlling transaksi pending dalam satu kesatuan. Dengan demikian, kontribusi penelitian ini adalah menghadirkan sistem monitoring yang tidak hanya menampilkan status, tetapi juga mendorong respons cepat tim teknis melalui kanal notifikasi yang murah, andal, dan mudah diadopsi perusahaan.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil merancang dan membangun sistem informasi monitoring dan controlling server berbasis web menggunakan framework ExpressJS yang terintegrasi dengan notifikasi Telegram pada PT. Fintek Digital Nusantara. Sistem melakukan pengecekan layanan secara otomatis setiap lima menit, menerapkan ambang kegagalan untuk mengurangi peringatan palsu, menahan transaksi bisnis baru sebagai pending saat layanan tidak sehat, mencatat seluruh riwayat pengecekan dan notifikasi, serta menyajikan laporan persentase uptime, jumlah insiden, dan estimasi downtime. Hasil pengujian black-box menunjukkan seluruh fungsi berjalan sesuai rancangan, dan gangguan dapat terdeteksi paling lambat satu siklus pengecekan disertai notifikasi real-time kepada administrator dan PIC. Dengan demikian, sistem ini menjawab permasalahan pemantauan manual yang lambat dan tidak terpusat, sekaligus membantu perusahaan menjaga ketersediaan layanan digitalnya.

Penelitian ini masih memiliki keterbatasan, antara lain pengecekan jenis ping yang diimplementasikan melalui koneksi TCP (belum ICMP murni), worker monitoring yang berjalan pada satu simpul, serta kanal notifikasi yang terbatas pada Telegram. Penelitian selanjutnya disarankan menambahkan kanal notifikasi lain seperti WhatsApp atau email dengan mekanisme eskalasi bertingkat, menerapkan arsitektur worker yang redundan agar pemantauan tetap berjalan saat terjadi kegagalan simpul, serta memanfaatkan analisis tren waktu respons untuk deteksi anomali secara dini.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada PT. Fintek Digital Nusantara yang telah memberikan izin dan dukungan selama pelaksanaan penelitian, serta kepada dosen pembimbing Jarot Dian Susatyono, S.Kom., M.Kom. dan Sumaryanto, S.Kom., M.Kom. atas arahan dan masukan dalam penyusunan naskah ini. Artikel ini merupakan bagian dari skripsi penulis.

DAFTAR REFERENSI

Adeguno, I. A., & Septiyanti, R. (2026). Sistem Informasi Berbasis Web Guna Pencatatan dan Monitoring Tamu Pada Ruang Server PT. Pertamina Patra Niaga Regional SUMBAGSEL. *Jurnal Surya Informatika*, *16*(1), 63–70. https://doi.org/10.48144/suryainformatika.v16i1.2419

Arsandi, M. R. A., & Syaripudin, A. (2024). Perancangan Sistem Monitoring Jaringan berbasis Web Server Terintegrasi Zabbix dan Notifikasi Telegram Pada PT Time Excelindo. *OKTAL: Jurnal Ilmu Komputer dan Science*, *3*(6), 1553–1561.

Azizi, M. R., Lutfi, S., & Purwanto, A. (2022). Sistem Monitoring Server Berbasis Android Di Dinas Komunikasi Dan Informatika Kabupaten Rembang. *Journal of Information System and Computer*, *2*(1), 53–57. https://doi.org/10.34001/jister.v2i1.287

Harun, A. A., & Munaldi, M. (2026). Pengembangan Sistem Aplikasi Untuk Pelaporan Dan Pemantauan Kebersihan Pekerja Cleaning Service Berbasis Android (Studi Kasus: Pt. Iss Indonesia Bintaro). *Informatika: Jurnal Teknik Informatika dan Multimedia*, *6*(1), 221–235. https://doi.org/10.51903/informatika.v6i1.1602

Hidayat, A., & Kurniawan, R. (2020). Perancangan sistem monitoring jaringan berbasis web menggunakan protokol ICMP dan HTTP. *Jurnal Teknologi Informasi dan Komunikasi*, *8*(2), 45–53.

Muh Sabri, Angriawan, R., & Hariyeni. (2026). Implementasi Simulasi Data Logging Berbasis Iot. *IDENTIK: Jurnal Ilmu Ekonomi, Pendidikan dan Teknik*, *03*(1), 111–116.

Newman, S. (2021). *Building microservices (2nd ed.)*. Sebastopol, CA: O'Reilly Media.

Nugraha, B. P., & Ratama, N. (2022). Implementasi Network Dan Server Monitoring Menggunakan Zabbix Berbasis Linux Integrasi Realtime Notifikasi Telegram. *OKTAL J. Ilmu Komput. dan Sains*, *1*(6), 549–554.

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach (9th ed.)*. New York: McGraw-Hill.

Prima, T., Wardhana, K., Apriani, R., Informatika, T., Insani, U. B., Design, U., Cell, L., Infus, M., & Optocoupler, S. (2026). *Sistem Monitoring Infus Berbasis Internet of Things Untuk Deteksi Volume dan Laju Tetesan Infus*. 408–419.

Rahmawati, L., & Nugroho, S. (2022). Implementasi sistem monitoring server untuk meningkatkan availability layanan digital. *Jurnal Informatika dan Rekayasa Perangkat Lunak*, *10*(1), 67–75.

Saragih, N. E., & Siregar, E. T. (2026). Pengembangan Sistem Informasi Deteksi Stunting Berbasis Android Menggunakan Metode Z-Score. *METHOMIKA Jurnal Manajemen*

Informatika dan Komputerisasi Akuntansi, *10*(1), 127–137. https://doi.org/10.46880/jmika.Vol10No1.pp127-137

Setiawati, C. L., Mahdi, I., & Yusnidar, C. (2026). Pengembangan Sistem Cerdas Berbasis Android dan AI untuk Prediksi Strategi Pemulihan UMKM Pascabencana di Pidie Jaya. *Blend Sains Jurnal Teknik*, *5*(1), 209–218. https://doi.org/10.56211/blendsains.v5i1.1862

Sutanto, A. B., Sugianto, R. A., Ritna Wahyuni, & Andi Prayogi. (2026). Sistem Monitoring Suhu Dan Kelembapan Berbasis Iot Terintegrasi Aplikasi Mobile Di Ruang Server PTPN IV. *Rabit : Jurnal Teknologi dan Sistem Informasi Univrab*, *11*(2), 2868–2881. https://doi.org/10.36341/rabit.v11i2.7912

Wiriaatmadja, M. F. J. S., & Ratama, N. (2022). Sistem Monitoring Jaringan Melalui Notifikasi Telegram Dengan Application Programming Interface (API) Menggunakan Netwatch Mikrotik Pada Jaringan PT. Solusi Trimegah Persada. *Ilmu Komputer Dan Sains*.