

Implementasi Telegram Bot sebagai Media *Monitoring* Gangguan Pelanggan Telkom dengan Pembaruan Berkala Berbasis Mobile

Noor Arief Kurniawan¹, Dyah Ruswanti², Astri Charolina³

^{1,2,3}Program Studi Informatika, Universitas Sahid Surakarta, Surakarta, Indonesia

Email: ¹the.ariep.k@gmail.com, ²dyahruswanti@usahidsolo.ac.id, ³astricharolina@usahidsolo.ac.id

Abstract

In a dynamic operational environment, especially in Telkom customer service, the process of monitoring disturbance reports is a challenge for supervisors. The disturbance reporting applications currently available are web-based, but they have not been designed responsively, so they are less user-friendly when accessed via mobile devices. Meanwhile, supervisors are required to continue conducting direct checks to ensure technician performance in the field. This research aims to develop a mobile-friendly solution in the form of a Telegram Bot that is able to retrieve regularly updated customer disturbance data from a central server, process it, and present it in a concise and easy-to-understand format via the Telegram application. The methods used include periodic data collection by helpdesk officers, data transformation into structured information, and integration of the Telegram API as a user interface. The implementation results indicate that the Telegram bot can provide information on the status of disturbance reports (new, progress, and completed), including the level of urgency, quickly and informatively, according to the latest data available. This solution is proven to support supervisor mobility in decision making and technical supervision without having to rely on web application access via desktop devices. Using Telegram as a platform also increases flexibility in accessing information in various field conditions. This research is a real contribution to the use of bot technology for the efficiency of field monitoring services based on regular updates.

Kata Kunci: *Telegram Bot, Customer Disturbance, Field Monitoring, Periodic Data Collection, Mobile Application.*

Abstrak

Dalam lingkungan operasional yang dinamis, khususnya pada layanan pelanggan Telkom, proses *monitoring* laporan gangguan menjadi tantangan tersendiri bagi *supervisor*. Aplikasi pelaporan gangguan yang tersedia saat ini berbasis *web*, namun belum dirancang secara responsif sehingga kurang ramah saat diakses melalui *mobile*. Sementara itu, *supervisor* dituntut untuk tetap melakukan pengecekan langsung dalam memastikan kinerja teknisi di lapangan. Penelitian ini bertujuan untuk mengembangkan solusi *mobile-friendly* berupa Telegram Bot yang mampu mengambil data gangguan pelanggan yang diperbarui secara berkala dari *server* pusat, mengolahnya, dan menyajikannya dalam format ringkas serta mudah dipahami melalui aplikasi Telegram. Metode yang digunakan meliputi pengambilan data secara periodik oleh petugas *helpdesk*, transformasi data menjadi informasi terstruktur, dan integrasi API Telegram sebagai antarmuka pengguna. Hasil implementasi menunjukkan bahwa *bot* Telegram mampu menyajikan informasi status laporan gangguan (baru, progres, dan selesai), termasuk tingkat urgensi, secara cepat dan informatif sesuai data terakhir yang tersedia. Solusi ini terbukti mendukung mobilitas *supervisor* dalam pengambilan keputusan dan pengawasan teknis tanpa harus bergantung pada akses aplikasi *web* melalui perangkat *desktop*. Penggunaan Telegram sebagai platform juga menambah fleksibilitas akses informasi di berbagai kondisi lapangan. Penelitian ini menjadi kontribusi nyata dalam pemanfaatan teknologi *bot* untuk efisiensi layanan *monitoring* lapangan berbasis pembaruan berkala.

Kata Kunci: *Bot Telegram, Gangguan Pelanggan, Monitoring Lapangan, Pembaruan Berkala, Aplikasi Mobile.*

\

1. PENDAHULUAN

Jaringan internet kini menjadi kebutuhan pokok masyarakat modern yang membawa dampak positif maupun negatif. Akses komunikasi yang luas membuka peluang bagi perkembangan teknologi industri yang mendukung berbagai sektor (Purba et al., 2021). Dalam memilih layanan internet, konsumen cenderung memilih penyedia terpercaya dengan jaringan stabil, di mana PT. Telkom menjadi pilihan utama karena citra merek dan kualitas layanannya. Kualitas produk terbukti memengaruhi keputusan pembelian, sebab konsumen akan menilai terlebih dahulu sebelum memilih (Sebayang & Nst, 2025). Kualitas layanan internet sendiri sangat bergantung pada peran *supervisor* yang berfungsi sebagai pengelola dan pemimpin lapangan (Suharso & Sulaksono, 2022). Namun, sistem pelaporan gangguan pelanggan saat ini hanya dapat diakses melalui aplikasi *web* yang belum optimal di perangkat *mobile*, sehingga *supervisor* kesulitan memperoleh data terbaru di lapangan. Kondisi ini memperlambat respon terhadap gangguan, menimbulkan *miskomunikasi*, dan menurunkan akurasi keputusan, yang pada akhirnya dapat berdampak pada turunnya kualitas dan kepercayaan pelanggan (Siregar & Qonitah, 2022). Untuk mengatasi hal tersebut, penelitian ini mengusulkan pengembangan *Telegram Bot* yang menyajikan informasi gangguan pelanggan secara berkala melalui perangkat *mobile*. Sistem ini berfungsi sebagai antarmuka ringan yang menampilkan data dari *helpdesk* dalam format ringkas dan mudah dipahami, sehingga *supervisor* dapat memantau laporan secara cepat, efisien, dan responsif tanpa bergantung pada aplikasi *web desktop*.

Sejumlah penelitian sebelumnya telah mengembangkan aplikasi serupa dalam ranah *monitoring* berbasis *bot* atau *mobile*, berikut adalah beberapa contoh penelitian tersebut:

Tabel 1. Penelitian *monitoring* berbasis bot

No	Peneliti & Tahun	Judul Penelitian	Kelebihan	Keterbatasan
1	Pasaribu & Ferdiansyah, (2024)	<i>Implementation of Network and Server Monitoring on Mikrotik with Netwatch Via Telegram Bot</i>	Terintegrasi langsung dengan ekosistem Mikrotik; notifikasi cepat	Fokus pada status perangkat, tidak mengelola data manual.
2	Gusrianda et al. (2024)	<i>Monitoring on Solid Foundation Servers Using Suricata Through Telegram Bot Notifications</i>	Implementasi sederhana; bisa dipantau dari mana saja	Berorientasi keamanan <i>server</i> , tidak relevan untuk rekap data manual gangguan pelanggan.
3	Putra et al. (2021)	Sistem Monitoring Server dengan Telegram Bot Menggunakan Zabbix	Integrasi dengan tools monitoring populer (Zabbix)	Tidak dapat mengakomodasi input manual dari petugas <i>helpdesk</i>
4	Lestari (2022)	Monitoring Internet Jaringan dengan Bot API Telegram	Notifikasi cepat kondisi koneksi internet	Informasi terbatas, tidak ada rekap atau <i>filter</i> data gangguan
5	Hidayat & Fikri (2023)	Sistem Notifikasi Server via Telegram Bot dan Cronjob	Mendukung pembaruan berkala menggunakan cronjob	Fokus hanya status server, bukan monitoring layanan pelanggan

Berdasarkan kajian penelitian terdahulu, implementasi *Telegram Bot* umumnya dimanfaatkan untuk *monitoring* teknis seperti pemantauan *server*, jaringan, dan keamanan sistem. Namun, sebagian besar belum memenuhi kebutuhan layanan pelanggan, khususnya pada aspek operasional lapangan. Keterbatasan yang ditemukan

meliputi tidak adanya dukungan terhadap data manual dari *helpdesk*, pembaruan informasi yang belum terstruktur, serta sistem yang belum mendukung mobilitas *supervisor*. Penelitian ini berupaya mengisi kekosongan tersebut dengan mengembangkan *Telegram Bot* sebagai media monitoring gangguan pelanggan yang adaptif terhadap kebutuhan operasional. Sistem yang dibangun mampu mengintegrasikan data manual, menyajikan pembaruan informasi berkala secara ringkas, serta menyediakan akses *mobile* dengan fitur penyaringan data sesuai kebutuhan *supervisor* untuk memantau laporan gangguan secara cepat dan *responsif* tanpa bergantung pada aplikasi web desktop.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini menggunakan metode *Waterfall*, yaitu pendekatan pengembangan sistem yang dilakukan secara bertahap dan terstruktur (Pratama et al., 2024). Pemilihan metode ini didasarkan pada karakteristik penelitian yang memiliki kebutuhan dan tujuan yang jelas sejak awal, sehingga setiap tahapan dapat dikerjakan secara sistematis dan berurutan dan dapat membangun solusi atas permasalahan keterbatasan akses data pelaporan gangguan pelanggan Telkom pada perangkat *mobile* (Hermansyah & Wahyuni, 2024). *Waterfall* dianggap lebih tepat dibandingkan metode *Agile* karena proyek ini tidak menuntut perubahan kebutuhan yang dinamis atau iterasi berulang dalam jangka pendek. Tahapan dalam metode *Waterfall* yang diterapkan meliputi: analisis kebutuhan sistem, perancangan sistem, implementasi *Telegram Bot*, pengujian fungsi sistem, serta evaluasi terhadap hasil akhir sistem. Setiap tahapan diselesaikan secara berurutan untuk memastikan sistem dapat berfungsi dengan baik dan sesuai tujuan.



Gambar 1. Ilustrasi Metode Waterfall

2.2 Analisis Kebutuhan Sistem

Tahap analisis kebutuhan dilakukan untuk mengidentifikasi seluruh *elemen* yang diperlukan dalam pengembangan sistem, baik dari sisi fungsionalitas, alur kerja, data yang digunakan, maupun sumber daya pendukungnya. Informasi kebutuhan sistem diperoleh melalui observasi langsung dan wawancara dengan *supervisor* dan petugas *helpdesk* di lingkungan Telkom Akses Regional 3 Area Solo (Iskandar & Ratnasari, 2021). Dari hasil tersebut, dapat disusun daftar kebutuhan sebagai berikut:

Supervisor saat ini harus melalui beberapa tahapan untuk memperoleh data gangguan pelanggan, salah satunya seperti meminta *file* data secara langsung ke *helpdesk*, ataupun mencari data secara mandiri pada aplikasi yang tidak ramah diakses pada *mobile*. Sistem yang baru harus mampu menyederhanakan alur ini, sehingga *supervisor* cukup memberikan perintah melalui antarmuka tertentu dan langsung menerima data yang sudah terformat dan relevan.

Data yang dibutuhkan berasal dari *file Excel* yang diunggah oleh *helpdesk*, yang berisi informasi mentah seperti nomor layanan, wilayah, jenis gangguan, status laporan, serta tanggal dan waktu laporan.

Data-data ini perlu diolah oleh *script* agar hasil akhirnya lebih ringkas dan mudah dibaca oleh *supervisor*, misalnya dengan menyaring hanya data aktif, mengelompokkan berdasarkan tingkat urgensi, atau menampilkan riwayat terbaru. Berdasarkan kebutuhan fungsional yang telah dipetakan, sistem kemungkinan akan memerlukan sebuah platform berbasis *web* agar dapat diakses kapan saja.

Dari analisis tersebut, dapat disusun kebutuhan sistem yang dibagi menjadi kebutuhan fungsional dan non-fungsional, seperti ditunjukkan pada tabel berikut:

Tabel 2. Kebutuhan Fungsional

No	Kebutuhan Fungsional	Deskripsi
1	Input data gangguan	Sistem harus dapat menerima file Excel dari <i>helpdesk</i> yang berisi data mentah gangguan pelanggan.
2	Pemrosesan data	Sistem harus mampu mengolah data untuk menyaring laporan aktif, mengelompokkan berdasarkan urgensi, dan menampilkan riwayat terbaru.
3	Output laporan	Supervisor dapat menerima hasil olahan data dalam format ringkas dan mudah dibaca melalui Telegram Bot.
4	Pencarian/filter data	Supervisor dapat memfilter data berdasarkan wilayah, status, atau periode waktu tertentu.
5	Integrasi antarmuka Telegram	Sistem harus menyediakan akses data melalui perintah yang dikirim supervisor di Telegram.

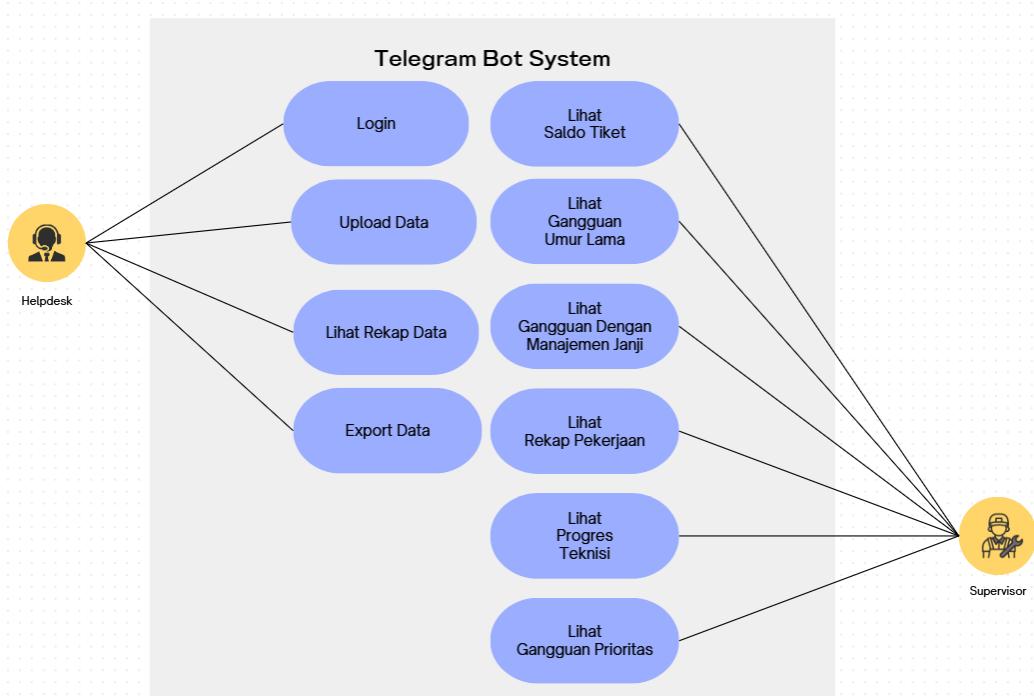
Tabel 3. Kebutuhan Non Fungsional

No	Kebutuhan Non-Fungsional	Deskripsi
1	Aksesibilitas	Sistem dapat diakses supervisor melalui perangkat mobile.
2	Kemudahan penggunaan	Antarmuka sederhana agar supervisor tidak perlu membuka aplikasi yang kompleks.
3	Kinerja	Pemrosesan data harus dilakukan dengan cepat agar supervisor segera memperoleh laporan.
4	Ketersediaan perangkat	Dibutuhkan komputer untuk <i>helpdesk</i> (unggah data) dan perangkat mobile untuk supervisor (akses hasil).

2.3 Perancangan Sistem

Pada tahap ini, sistem mulai dimodelkan untuk menggambarkan bagaimana sistem bekerja serta bagaimana interaksi antara pengguna dan sistem berlangsung. Perancangan dilakukan menggunakan *Use Case Diagram* yang merupakan bagian dari *Unified Modeling Language (UML)* (Arianti et al, 2022). Diagram ini menggambarkan interaksi *supervisor* sebagai aktor utama dengan sistem yang dibangun melalui *platform Telegram*. *Supervisor* dapat melakukan sejumlah aksi, seperti meminta data gangguan pelanggan, menyaring data berdasarkan tingkat urgensi, dan mengakses laporan terbaru. Sistem akan merespons secara otomatis dengan menyajikan informasi dari data yang telah diolah oleh *script PHP*, yang bersumber dari *file raw* berformat Excel (.xls) yang sebelumnya diunggah oleh petugas *helpdesk* melalui antarmuka khusus.

Pendekatan berbasis *use case* ini memudahkan proses perancangan karena mampu menjelaskan fungsi sistem dari sudut pandang pengguna secara sederhana, sistematis, dan mudah dipahami oleh pengembang maupun pihak non-teknis.

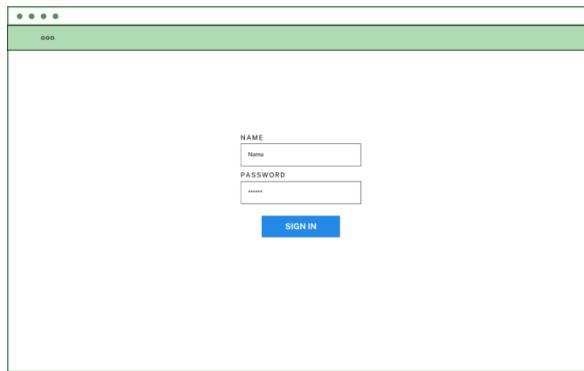


Gambar 2. Use Case

Tabel 4. Penjelasan Teknis

No	Use Case	Aktor	Deskripsi
1	Unggah Data Gangguan	Helpdesk	Helpdesk mengunggah file Excel berisi data gangguan ke sistem. Sistem akan memvalidasi format file, menyimpan data mentah ke database, serta melakukan preprocessing agar data siap dipanggil Telegram Bot.
2	Validasi & Penyimpanan Data	Sistem	Setelah file diterima, sistem memeriksa konsistensi (kolom wajib: ID layanan, wilayah, status, waktu laporan). Jika ada kesalahan, sistem memberi notifikasi error ke helpdesk. Jika valid, data disimpan dan diperbarui.
3	Permintaan Data Gangguan	Supervisor	Supervisor mengetik perintah (misalnya: /saldo atau /tiketlama) melalui Telegram. Bot menerima perintah lalu menghubungkan ke database untuk mengambil data terbaru.
4	Pemrosesan & Penyaringan Data	Sistem	Sistem menyaring data sesuai perintah supervisor (misalnya menampilkan hanya gangguan aktif atau membagi berdasarkan urgensi). Hasilnya diformat agar mudah dibaca di layar mobile.
5	Penyajian Hasil ke Supervisor	Telegram Bot → Supervisor	Sistem mengirimkan hasil olahan berupa ringkasan (jumlah laporan, daftar urgent, riwayat terbaru) ke Telegram Supervisor. Data ditampilkan dengan format tabel sederhana atau daftar pesan.
6	Monitoring Berkala	Supervisor	Supervisor dapat mengaktifkan mode notifikasi berkala. Sistem akan otomatis mengirim update data setiap periode tertentu tanpa harus diminta.

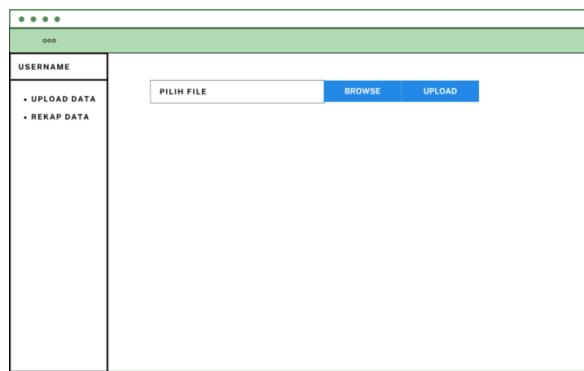
Sebagai bagian dari perancangan yang lebih detail, tahap ini juga mencakup pembuatan *wireframe* sebagai representasi visual awal dari antarmuka pengguna, khususnya untuk petugas *helpdesk*. *Wireframe* ini dirancang untuk menggambarkan alur interaksi dasar seperti *fitur login*, *unggah file Excel*, menampilkan *rekap data*, serta tombol untuk mengekspor data hasil olahan. Pembuatan *wireframe* membantu merancang antarmuka yang sederhana namun fungsional, dengan fokus pada kemudahan penggunaan dan efisiensi kerja (Fatah & Mufarroha, 2022).



Rancangan halaman login. Tampilan ini menampilkan dua input text: 'NAME' dan 'PASSWORD', serta sebuah tombol 'SIGN IN' di bawahnya.

Gambar 3. Rancangan halaman *login*

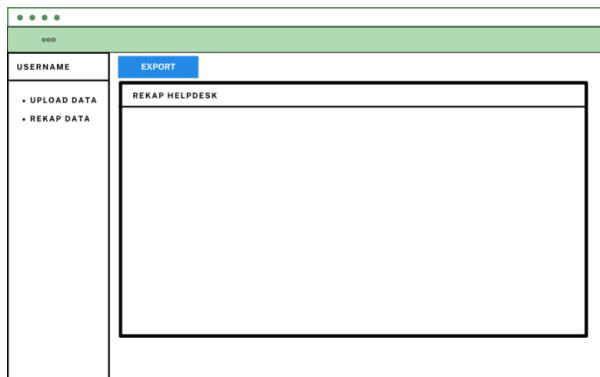
Untuk dapat melakukan proses *upload* data, petugas *helpdesk* diharuskan untuk *login* terlebih dahulu dengan memasukan username dan password



Rancangan halaman upload. Di sisi kiri terdapat sidebar dengan menu 'USERNAME' yang menunjukkan opsi 'UPLOAD DATA' dan 'REKAP DATA'. Di sisi kanan terdapat tiga tombol: 'PILIH FILE', 'BROWSE', dan 'UPLOAD'.

Gambar 4. Rancangan halaman *upload*

Tombol *Browse* digunakan oleh petugas *helpdesk* untuk memilih *file* data yang akan *diupload*, berisi *raw* data dengan ekstensi .xls



Rancangan halaman rekap. Di sisi kiri terdapat sidebar dengan menu 'USERNAME' yang menunjukkan opsi 'UPLOAD DATA' dan 'REKAP DATA'. Di sisi kanan terdapat tombol 'EXPORT' dan bagian tampilan yang bertuliskan 'REKAP HELPDESK'.

Gambar 5. Rancangan halaman rekap

Halaman rekap digunakan petugas *helpdesk* untuk melihat hasil rekapan data gangguan, pada halaman tersebut juga disediakan tombol *export* untuk mengekspor data menjadi *file* dengan ekstensi xls.

Selain itu, dilakukan pula perancangan struktur data, yang mencakup pemetaan informasi penting dari *file Excel*, seperti ID gangguan, wilayah lokasi, status gangguan, waktu laporan, serta tingkat urgensi. Struktur data ini disusun agar dapat langsung diolah oleh sistem, sehingga memudahkan *supervisor* dalam melihat informasi yang telah disaring secara otomatis dan disajikan dengan format yang mudah dipahami (Ginting, 2024).

Tabel 5. Format data unggah

ID	Nama Pelanggan	Workzone	Status	Reported Date	Booking Date	Cust Segment
1	Ahmad	KRT	Open	07/07/2025 07:30:00	07/07/2025 10:00:00	Platinum

Script PHP akan membaca *file* ini menggunakan *library* PhpSpreadsheet, lalu menyaring data berdasarkan: tanggal pelaporan, waktu manajemen janji dan *customer segment*. Berikut adalah struktur data yang digunakan :

Tabel 6. Struktur data

No	Variabel	Tipe Data	Deskripsi
1	id	Integer	Nomor unik untuk mengidentifikasi setiap laporan gangguan
2	nama	varchar	Nama pelanggan yang melaporkan gangguan
3	workzone	char	Area lokasi
4	status	char	Status laporan
5	tgopen	time	Waktu saat gangguan dilaporkan oleh pelanggan
6	tglmanja	time	waktu pelanggan minta dikunjungi
7	csegment	varchar	segment pelanggan (menentukan tingkat urgensi)

Fitur-fitur bot dirancang untuk memberikan informasi penting kepada *supervisor* secara cepat, ringkas, dan kontekstual. Setiap perintah dikembangkan untuk menjawab kebutuhan operasional *monitoring* di lapangan.

Berikut adalah daftar perintah beserta fungsinya :

Tabel 7. Perintah Bot Telegram

Perintah	Deskripsi Fungsi	Keluaran
/saldo	Menampilkan jumlah laporan dengan status <i>open</i> dan <i>close</i>	<i>Output</i> berupa jumlah angka gangguan baik <i>open</i> dan <i>close</i>
/manja	Menampilkan laporan dengan manajemen janji hari H	<i>Output</i> berupa id Area Jam
/tiketlama	Menampilkan laporan dengan durasi waktu diatas 20 Jam	<i>Output</i> berupa id Area Total Durasi
/hvc	Menampilkan laporan pelanggan prioritas	<i>Output</i> berupa id Area Waktu pelaporan Status

2.4 Implementasi Sistem

Tahap ini bertujuan untuk merealisasikan desain yang telah disusun ke dalam bentuk aplikasi nyata yang dapat dijalankan dan diuji. Implementasi dilakukan menggunakan bahasa pemrograman PHP (Damanik, 2021) yang berjalan pada layanan *web hosting* (Huda, 2021), dengan memanfaatkan Telegram API sebagai media interaksi antara sistem dan *supervisor*. Sistem terdiri atas dua antarmuka utama, yaitu antarmuka untuk petugas *helpdesk* dan antarmuka berbasis bot untuk *supervisor* (Fathurrozi, 2021). Pada sisi *helpdesk*, sistem memungkinkan pengguna untuk mengunggah *file* Excel (.xls) berisi data gangguan pelanggan melalui halaman *web* yang telah disediakan. *File* ini kemudian diproses oleh *script* PHP yang akan membaca isi *file*, menyaring data sesuai kebutuhan, dan menyimpannya dalam struktur data terorganisir. Di sisi lain, Telegram Bot merespons perintah dari *supervisor* dengan menyajikan data yang telah diolah dalam format ringkas dan mudah dipahami, seperti daftar gangguan terbaru, gangguan prioritas tinggi, dan filter berdasarkan status.

Struktur implementasi sistem dibangun secara *modular*, dengan memisahkan *script* untuk proses *upload*, pembacaan data, pengolahan data, dan komunikasi dengan Telegram API. Dengan pendekatan ini, proses pengembangan menjadi lebih terstruktur dan memudahkan dalam proses *debugging* maupun pengembangan lanjutan.

Tabel 8. Spesifikasi Implementasi Sistem

Komponen	Spesifikasi
Bahasa Pemrograman	PHP
Web Server	Apache (via CPanel Hosting)
Database (opsional)	MySQL
Format Input Data	Microsoft Excel (.xls)
Platform Bot	Telegram API
Fungsi Utama Helpdesk	Upload file Excel / Lihat riwayat unggahan / Export data olahan
Fungsi Utama Supervisor	Minta data gangguan / Filter data berdasarkan urgensi dan status / Lihat laporan terbaru
Output Telegram Bot	Format teks ringkas berisi daftar gangguan

2.5 Pengujian Sistem

Tahap pengujian sistem bertujuan untuk memastikan bahwa seluruh fungsi yang telah diimplementasikan bekerja sesuai dengan kebutuhan pengguna dan spesifikasi yang telah dirancang. Pengujian dilakukan dengan metode *black-box testing*, yaitu dengan menguji fungsionalitas sistem dari sisi pengguna tanpa melihat struktur internal kode program. Fokus pengujian ditujukan pada apakah sistem dapat merespon input dengan keluaran (*output*) yang sesuai dan bebas dari kesalahan logika (Ismail & Efendi, 2020).

Pengujian dilakukan terhadap dua antarmuka utama, yaitu antarmuka *web* untuk petugas *helpdesk* dan Telegram Bot untuk *supervisor*. Pada antarmuka *helpdesk*, dilakukan pengujian terhadap *fitur* unggah file, hasil unggahan, dan proses ekspor data hasil olahan sistem. Sedangkan pada sisi *supervisor*, diuji respons sistem terhadap berbagai perintah bot seperti permintaan data gangguan, *filter* berdasarkan urgensi, dan permintaan laporan terbaru.

Hasil pengujian menunjukkan bahwa sistem dapat berjalan dengan baik dan seluruh fungsi utama dapat dijalankan sesuai dengan skenario yang direncanakan. Dengan demikian, sistem dianggap telah memenuhi kriteria kelayakan fungsional dan siap digunakan dalam lingkungan operasional.

2.6 Evaluasi Sistem

Evaluasi sistem dilakukan dengan pendekatan komparatif, yaitu membandingkan kondisi proses bisnis sebelum dan sesudah implementasi sistem. Tujuan dari evaluasi ini adalah untuk menilai efektivitas dan efisiensi sistem berdasarkan peningkatan kinerja operasional, terutama dalam hal kecepatan distribusi informasi dan kemudahan akses data oleh pihak terkait (Amelia, 2025).

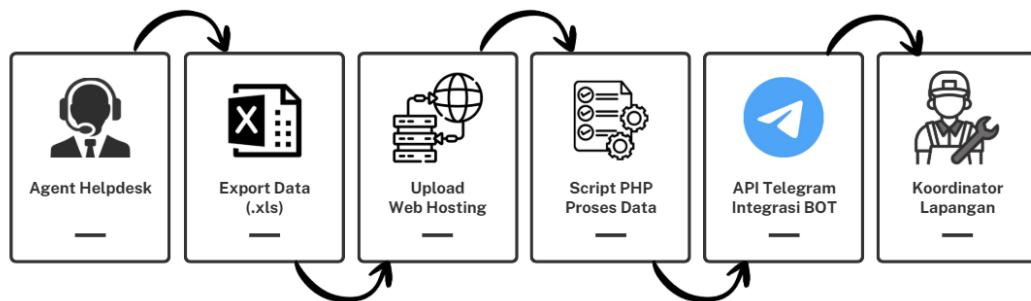
Dari hasil perbandingan tersebut, apakah dapat disimpulkan bahwa sistem yang dikembangkan memberikan dampak positif dalam mempercepat alur kerja dan meningkatkan efisiensi operasional. Evaluasi ini juga menunjukkan apakah sistem berhasil menjawab kebutuhan utama pengguna, sekaligus membuka peluang untuk pengembangan *fitur* lanjutan yang mendukung *otomasi* lebih jauh.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Analisis Kebutuhan

Dari hasil wawancara dan observasi tersebut disimpulkan bahwa dibutuhkan sistem baru yang mampu berjalan secara *online* selama 24 jam melalui layanan *web hosting*, tanpa bergantung pada jaringan *internal* perusahaan (Arman & Meiriyama, 2025). Selain itu, proses kerja menunjukkan bahwa data gangguan pelanggan masih disusun dalam bentuk Excel (.xls) yang diperbarui secara berkala oleh *helpdesk*. Dengan demikian, sistem yang dikembangkan harus mampu membaca dan mengolah data tersebut secara langsung. Dalam hal alur kerja, proses dimulai dari *input* data gangguan oleh *helpdesk*,

kemudian sistem secara otomatis menyajikan informasi tersebut kepada *supervisor* melalui Telegram Bot (Hasibuan & Nasution, 2023). Agar informasi lebih mudah dipahami, bot juga harus mampu menampilkan data yang terstruktur berdasarkan status, urgensi, dan waktu laporan, sehingga dapat membantu *supervisor* dalam menetapkan prioritas penanganan secara cepat dan tepat di lapangan. Untuk mendukung kebutuhan tersebut, sistem dirancang dengan arsitektur *client-server* berbasis *web*. Data disimpan dan diproses di *sisi server* menggunakan *script PHP* yang berjalan pada layanan *web hosting*. Telegram Bot berperan sebagai antarmuka pengguna (*client*) yang mengirim perintah dan menerima respon dari *server* melalui API (Kurniawan at el, 2024). Pendekatan ini memungkinkan sistem untuk berjalan secara efisien, responsif, dan dapat diakses kapan saja oleh pengguna di lapangan melalui aplikasi Telegram di perangkat *mobile*.



Gambar 6. Arsitektur Sistem

3.2 Hasil Perancangan Sistem

Berdasarkan hasil perancangan sistem yang telah dijabarkan pada Bab 2, sistem *monitoring* gangguan pelanggan melalui Telegram Bot telah diimplementasikan menggunakan pendekatan berbasis *web* dan integrasi dengan *platform* pesan *instant*. Implementasi ini mencakup realisasi dari *use case diagram*, desain antarmuka awal (*wireframe*), serta struktur data yang digunakan.

Pada tahap implementasi, setiap fungsi dalam *use case diagram* telah direalisasikan dalam bentuk *fitur* sistem:

Tabel 9. Spesifikasi Implementasi Sistem

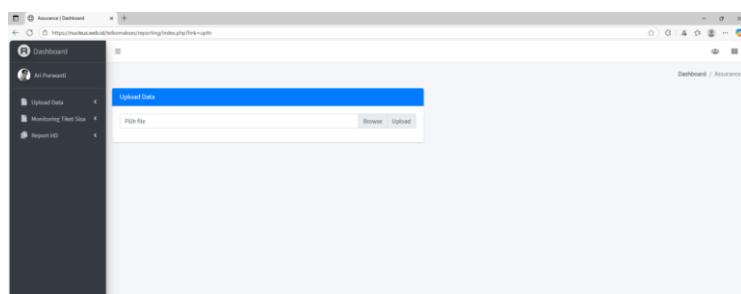
Aktor	Aksi yang Dilakukan	Respons Sistem	Fungsi
<i>Supervisor</i>	Perintah /tiketlama	Bot menampilkan data laporan gangguan dengan durasi waktu diatas 20 Jam	Fitur ini membantu <i>supervisor</i> segera mengidentifikasi laporan yang berisiko terlambat ditangani sehingga dapat mengambil langkah percepatan agar kualitas layanan pelanggan tetap terjaga.
<i>Supervisor</i>	Perintah /saldo	Bot menampilkan data total laporan gangguan dengan status <i>close</i> dan <i>open</i>	Penyajian status gangguan secara <i>real-time</i> memberi <i>supervisor</i> gambaran beban kerja aktual sehingga dapat menentukan prioritas penyelesaian dengan lebih tepat.
<i>Supervisor</i>	Perintah /manja	Bot menampilkan data laporan gangguan dengan manajemen janji khusus	Informasi ini mempermudah <i>supervisor</i> mengawasi laporan yang memerlukan penanganan sesuai kesepakatan khusus dengan pelanggan sehingga mengurangi potensi kelalaian layanan.
<i>Supervisor</i>	Perintah /rekapsto [kodestol]	Bot menampilkan data rekap progres gangguan pada wilayah tertentu	Fitur ini memungkinkan <i>supervisor</i> menganalisis area dengan tingkat gangguan tinggi sehingga dapat segera menambah teknisi atau sumber daya di wilayah yang membutuhkan.

<i>Supervisor</i>	Perintah /cektek [kodesto]	Bot menampilkan data pekerjaan teknisi	Informasi pekerjaan teknisi memberikan <i>visibilitas</i> kepada <i>supervisor</i> terkait progres penyelesaian gangguan di lapangan sehingga distribusi kerja dapat dipantau dan dioptimalkan.
<i>Supervisor</i>	Perintah /hvc	Bot menampilkan data laporan gangguan pelanggan prioritas / urgensi tinggi	Penyaringan otomatis untuk pelanggan prioritas membantu <i>supervisor</i> menjaga kepuasan pelanggan penting yang sangat berpengaruh terhadap citra layanan Telkom.
<i>Helpdesk</i>	Mengunggah file Excel	Sistem menyimpan dan mengolah data untuk dikirim ke Telegram	Fitur ini memudahkan <i>helpdesk</i> dalam memasukkan data gangguan secara cepat dan akurat sehingga <i>supervisor</i> selalu menerima informasi terbaru tanpa risiko keterlambatan atau kesalahan input
<i>Helpdesk</i>	Melihat rekap data	Sistem menampilkan rekapan pekerjaan petugas <i>helpdesk</i>	Dengan adanya rekap data, <i>helpdesk</i> dapat melakukan <i>validasi</i> terhadap informasi yang diunggah sehingga data yang diterima <i>supervisor</i> lebih akurat dan siap digunakan untuk pengambilan keputusan.
<i>Helpdesk</i>	Export Data	Sistem mengekspor data rekapan pekerjaan petugas <i>helpdesk</i>	Fasilitas ekspor data memungkinkan <i>helpdesk</i> menyediakan laporan resmi dalam berbagai format, sehingga mendukung kebutuhan dokumentasi dan pelaporan berkala bagi <i>supervisor</i> .

3.3 Hasil Implementasi Sistem

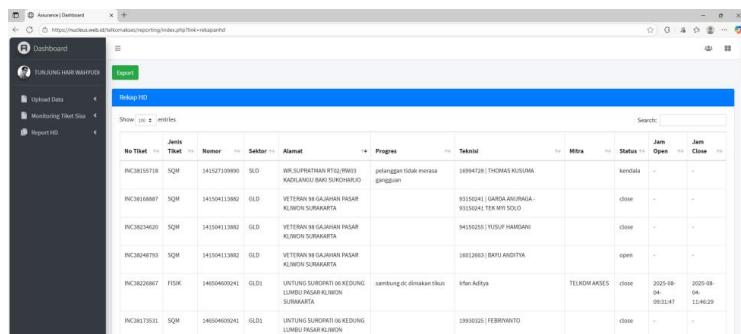
Antarmuka berbasis *web* untuk petugas *helpdesk* diimplementasikan berdasarkan rancangan *wireframe* yang telah dibuat sebelumnya. Halaman ini dirancang dengan fokus pada kemudahan penggunaan dan efisiensi proses kerja, khususnya dalam proses unggah data dan pengelolaan riwayat (Annisa et al, 2024).

Tampilan awal dari halaman web ini memuat komponen utama sebagai berikut:



Gambar 7. Halaman *Upload File*

Memudahkan *helpdesk* dalam memasukkan data gangguan secara cepat tanpa proses manual yang berulang.

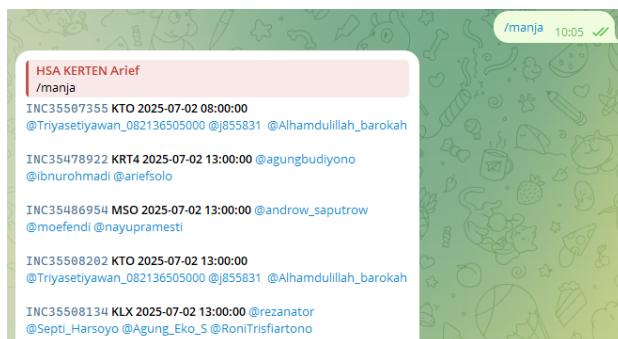


Gambar 8. Halaman Rekap data dan Fungsi *Export*

Memberikan transparansi kepada *helpdesk* atas data yang sudah diunggah.

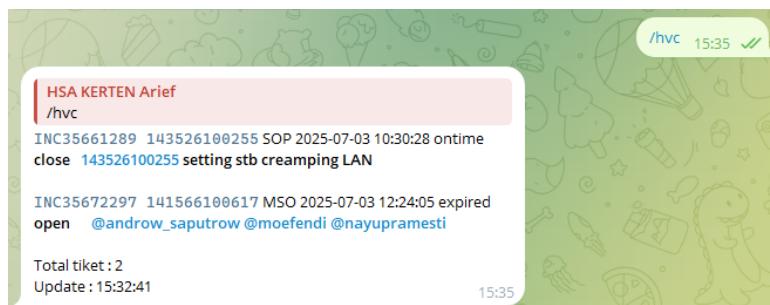
Gambar 9. File hasil fungsi *export*

Menyediakan cadangan dan dokumentasi untuk kebutuhan pelaporan resmi. *Supervisor* berinteraksi dengan sistem melalui Telegram Bot, yang dirancang agar mampu menyajikan data gangguan pelanggan secara ringkas, responsif, dan mudah dipahami, bahkan saat digunakan dalam kondisi *mobilitas* tinggi. Bot ini memberikan beberapa *menu* dan *fitur* utama yang diakses menggunakan perintah dan tombol interaktif. Antarmuka ini dibangun agar *supervisor* tidak perlu membuka aplikasi tambahan atau mengakses sistem *desktop*. Seluruh informasi disajikan langsung melalui pesan di Telegram, dengan struktur data yang telah diolah oleh *script* PHP secara otomatis dari *file* Excel yang diunggah oleh *helpdesk*. Berikut beberapa perintah yang digunakan *supervisor* dalam menjalankan telegram bot ketika sedang berada di lapangan.



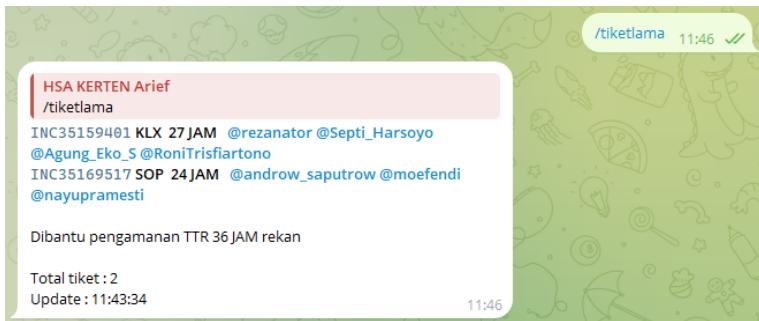
Gambar 10. Hasil perintah /manja

Mengidentifikasi pelanggan yang membutuhkan waktu perbaikan khusus karena sudah ada kesepakatan penanganan



Gambar 11. Hasil perintah /hvc

Membantu supervisor menangani laporan pelanggan prioritas, seperti instansi penting atau pelanggan dengan tagihan tinggi.



Gambar 12. Hasil perintah /tiketlama

Memudahkan supervisor mengidentifikasi laporan yang berisiko lambat ditangani.

3.4 Hasil Pengujian Sistem

Tahap pengujian sistem bertujuan untuk memastikan bahwa seluruh fungsi yang telah diimplementasikan bekerja sesuai dengan kebutuhan pengguna dan spesifikasi yang telah dirancang. Pengujian dilakukan dengan metode *black-box testing*, yaitu dengan menguji fungsionalitas sistem dari sisi pengguna tanpa melihat struktur internal kode program. Fokus pengujian ditujukan pada apakah sistem dapat merespon input dengan keluaran (*output*) yang sesuai dan bebas dari kesalahan logika.

Pengujian dilakukan terhadap dua antarmuka utama, yaitu antarmuka *web* untuk petugas *helpdesk* dan Telegram Bot untuk *supervisor*. Pada antarmuka *helpdesk*, dilakukan pengujian terhadap *fitur* unggah *file*, tampilan rekap data, dan proses ekspor data hasil olahan sistem. Sedangkan pada sisi *supervisor*, diuji respons sistem terhadap berbagai perintah bot seperti permintaan data gangguan, filter berdasarkan urgensi, dan permintaan laporan terbaru (Sakinah et al, 2024).

Hasil pengujian menunjukkan bahwa sistem dapat berjalan dengan baik dan seluruh fungsi utama dapat dijalankan sesuai dengan skenario yang direncanakan. Dengan demikian, sistem dianggap telah memenuhi kriteria kelayakan fungsional dan siap digunakan dalam lingkungan operasional.

Berikut adalah hasil pengujian fungsi *fitur*

Tabel 10. Hasil pengujian *fitur*

No	Fitur	Input	Output yang Diharapkan	Status
1	Upload file	File .xls	File berhasil diunggah dan diproses	Berhasil
2	Data rekap	Klik menu rekap	Menampilkan data rekap	Berhasil
3	Eksport data	Klik tombol export	File hasil pengolahan berhasil diunduh	Berhasil
4	Tiket urgensi	/hvc	Menampilkan data gangguan urgensi tinggi	Berhasil
5	Jumlah saldo	/saldo	Menampilkan jumlah tiket <i>open</i> dan <i>close</i>	Berhasil
6	Tiket lama	/tiketlama	Menampilkan tiket diatas 20 jam	Berhasil
7	Manajemen janji	/manja	Menampilkan laporan gangguan dengan manajemen janji khusus	Berhasil

3.5 Hasil Evaluasi Sistem

Evaluasi sistem dilakukan dengan pendekatan komparatif, yaitu membandingkan kondisi proses bisnis sebelum dan sesudah implementasi sistem. Tujuan dari evaluasi ini adalah untuk menilai efektivitas dan efisiensi sistem berdasarkan peningkatan kinerja operasional, terutama dalam hal kecepatan distribusi informasi dan kemudahan akses data

oleh pihak terkait. Sebelum sistem dikembangkan, proses distribusi data gangguan pelanggan dilakukan secara manual melalui aplikasi *desktop* internal yang sangat sulit diakses menggunakan *mobile*. Hal ini menyebabkan keterlambatan penyampaian informasi ke *supervisor* lapangan yang sering berada di luar kantor. Selain itu, proses penyusunan laporan memerlukan waktu tambahan karena data dari Excel harus disaring dan diringkas secara manual. Setelah implementasi sistem berbasis Telegram Bot dan antarmuka *upload* berbasis *web*, proses tersebut mengalami peningkatan yang *signifikan*. Data gangguan dapat langsung diunggah oleh *helpdesk* melalui *web interface*, kemudian secara otomatis diproses oleh sistem dan dikirimkan ke Telegram Bot yang digunakan *supervisor*. Penyaringan berdasarkan urgensi, status, dan waktu laporan juga dilakukan secara otomatis, sehingga *supervisor* dapat segera menetapkan prioritas penanganan tanpa harus membuka *file* Excel mentah.

Tabel 11. Hasil evaluasi komparatif

Aspek	Sebelum Sistem	Sesudah Sistem	Hasil
Akses Data <i>Supervisor</i>	Hanya melalui <i>desktop internal</i> kantor	Melalui Telegram Bot, bisa diakses kapan saja	Setelah sistem diimplementasikan, <i>supervisor</i> dapat mengakses data melalui Telegram Bot kapan saja dan di mana saja. Perubahan ini secara langsung meningkatkan efisiensi karena <i>supervisor</i> tidak perlu menunggu berada di kantor untuk mendapatkan informasi, sehingga penanganan gangguan dapat dilakukan lebih cepat.
Distribusi Informasi	Manual dan sering tertunda	Otomatis, <i>real-time</i> setelah data diunggah	Dengan sistem baru, informasi terdistribusi secara otomatis dan <i>real-time</i> setelah data diunggah. Hal ini memastikan semua pihak mendapatkan informasi secara serentak, mempercepat koordinasi, dan mengurangi risiko keterlambatan penanganan di lapangan.
Proses <i>Upload</i> Data	Tidak tersedia, <i>input</i> manual	<i>Web interface</i> untuk <i>upload</i> <i>file</i> Excel	Dengan adanya <i>web interface</i> untuk <i>upload file</i> Excel, proses input menjadi lebih mudah, cepat, dan konsisten. Efisiensi meningkat karena <i>helpdesk</i> dapat fokus pada validasi data ketimbang pekerjaan administratif yang berulang.
Penyaringan Data Gangguan	Manual, melalui <i>filter</i> Excel	Otomatis berdasarkan status, urgensi, dan waktu	Sistem baru menyajikan data yang sudah otomatis terfilter berdasarkan status, urgensi, dan waktu. Ini membuat <i>supervisor</i> langsung menerima data yang relevan tanpa perlu proses tambahan, sehingga keputusan prioritas dapat diambil jauh lebih cepat.
Efektivitas Penanganan Lapangan	Terlambat mengambil keputusan prioritas	Cepat, karena data sudah terstruktur dan <i>terfilter</i>	Setelah sistem diterapkan, data yang sudah otomatis terstruktur dan <i>terfilter</i> memungkinkan <i>supervisor</i> segera menentukan tindakan, sehingga pekerjaan teknisi di lapangan menjadi lebih terarah dan efektif.
Waktu Pemrosesan	± 30 menit untuk rekap dan distribusi	< 3 menit dari unggah hingga distribusi ke bot	Dengan sistem baru, waktu pemrosesan turun drastis menjadi kurang dari 3 menit sejak <i>helpdesk</i> mengambil data hingga data terdistribusi ke Telegram Bot. Hal ini secara signifikan meningkatkan efisiensi operasional dan mempercepat siklus penanganan gangguan, sehingga pelanggan tidak perlu menunggu terlalu lama

4. KESIMPULAN

Penelitian ini membuktikan bahwa implementasi Telegram Bot sebagai media *monitoring* gangguan pelanggan mampu meningkatkan efisiensi operasional *supervisor* Telkom dengan menghadirkan sistem yang lebih cepat, mudah diakses, serta menyajikan data yang terstruktur dan relevan. Perubahan mendasar yang ditawarkan sistem ini terletak pada otomatisasi proses distribusi informasi yang sebelumnya memakan waktu

lama, kini dapat dilakukan hanya dalam hitungan menit dengan tingkat akurasi yang lebih baik. Hal ini secara langsung berimplikasi pada percepatan pengambilan keputusan prioritas serta efektivitas koordinasi lapangan, sehingga kualitas layanan terhadap pelanggan dapat ditingkatkan. Lebih jauh, penelitian ini memiliki *implikasi* praktis untuk penerapan di area lain selain Regional 3 Solo, mengingat permasalahan akses data dan keterlambatan distribusi informasi umumnya bersifat serupa di berbagai unit kerja Telkom. Jika sistem ini diperluas cakupannya, potensi peningkatan kinerja operasional dan kepuasan pelanggan akan semakin besar. Sebagai saran pengembangan, penelitian ini dapat ditingkatkan menuju otomatisasi penuh dengan mengintegrasikan sistem secara langsung ke database operasional Telkom agar data gangguan dapat ditarik secara *real-time* tanpa perlu unggahan manual. Namun, langkah tersebut memerlukan persetujuan dan dukungan resmi dari pihak Telkom mengingat keterlibatan infrastruktur dan kebijakan keamanan data perusahaan. Dengan adanya integrasi penuh, sistem ini tidak hanya berfungsi sebagai alat bantu supervisor, tetapi juga dapat berkembang menjadi *platform monitoring* cerdas yang mendukung strategi digitalisasi layanan pelanggan secara menyeluruh.

REFERENCES

- Amelia A. N. (2025). Penerapan Sistem Keuangan Desa Dalam Pengelolaan Keuangan Desa Di Kabupaten Kotawaringin Timur : Studi Komparatif. *Kotawaringin Timur*. 4-6
- Annisa L. H., Ghandi F.M., Nasokha S. N., & Muhammad. S. F. (2024). Perancangan User Interface Website Cloud Storage untuk Meningkatkan Efisiensi Operasional UMKM. *Journal of Digital Business and Management*. 3(1):1-12
- Arianti T., Fa'izi A., Adam S., & Wulandari M. (2022). Perancangan Sistem Informasi Perpustakaan Menggunakan Diagram Uml (Unified Modelling Language). *Jurnal Ilmiah Komputer Terapan dan Informasi*. 1(1):19-25
- Arman, M., & Meiriyama, N. (2025). Rancang bangun web server blog dengan layanan VPS dan navigasi. *Jurnal Ilmiah Betrik*, 16(01), 59–73. <https://doi.org/10.36050/w5xy6d50>
- Damanik, B. (2021). RANCANGAN SISTEM INFORMASI SMP NEGERI 1 TUHEMBERUA KABUPATEN NIAS UTARA MENGGUNAKAN PHP CODEIGNITER. *JURNAL MAHAJANA INFORMASI*, 6(1), 6–15. <https://doi.org/10.51544/jurnalmi.v6i1.1979>
- Fatah, D. A., & Mufarroha, F. A. (2022). PERANCANGAN ANTARMUKA PENGGUNA SISTEM INFORMASI AKADEMIK BERBASIS WIREFRAMING. *Jurnal Simantec*, 11(1), 97–106. <https://doi.org/10.21107/simantec.v11i1.19739>
- Fathurrozi, A., & Karimah, F. (2021). PELAYANAN DAN INFORMASI CUSTOMER SERVICE BERBASIS BOT TELEGRAM DENGAN ALGORITMA FORWARD CHAINING PADA CV.PRIMGUARD INDONESIA. *Journal of Informatics and Information Security*, 2(2). <https://doi.org/10.31599/jiforty.v2i2.884>
- Ginting S. H. N. (2024). Pengantar Struktur Data. Mifandi Mandiri Digital, Sumatera Utara. 2-13.
- Hasibuan, N. Y. M., & Nasution, N. M. I. P. (2023). Penerapan Database Impor dan Ekspor Menggunakan SQL Server 2008. *MAMEN Jurnal Manajemen*, 2(3), 327–334. <https://doi.org/10.55123/mamen.v2i3.2043>
- Hermansyah, Wijaya R. F., & Wahyuni s. (2024). Desain Aplikasi Cinta Mangrove Berbasis Mobile Di Desa Kota Pari Dengan Metode Waterfall. *Senashtek Seminar Nasional Sosial Humaniora & Teknologi*. 2(1):42-48.
- Huda M. (2021). Bisnis Web Hosting: Teknologi Pendukung Untuk Menjalankan Usaha Web Hosting. *Bisakimia*, Jawa Barat. 6-8

- Iskandar A. A., & Ratnasari C. I. (2021). Analisis Kebutuhan Fungsional Sistem Informasi Manajemen Konfensi Berbasis Web (Studi Kasus Shofa Collection Tasikmalaya). *Jurnal Automata*. 2(1):1-9.
- Ismail, I., & Efendi, J. (2020). Black-Box Testing : Analisis kualitas aplikasi Source code bank programming. *Jurnal JTIK (Jurnal Teknologi Informasi Dan Komunikasi)*, 4(2), 1. <https://doi.org/10.35870/jtik.v5i1.148>
- Kurniawan Y., Connie, & Nirwana. (2024). Pengembangan Media Pembelajaran Interaktif dengan Bot Api Media Sosial Telegram pada Materi Gerak Harmonik Sederhana Kelas X SMA. *Kasuari: Physics Education Journal*. 7(1):1-12
- Pratama, M. W., Islami, N. M. a. Q., Melvina, S., Yulianti, S., & Rahayu, S. (2024). Implementasi Metode Waterfall dalam Perancangan Manajemen Proyek Sistem Informasi Penjualan pada Toko Elektronik Jaya Abadi. *Klik - Jurnal Ilmu Komputer*, 5(2), 53–61. <https://doi.org/10.56869/klik.v5i2.599>
- Purba, N., Yahya M., & Nurbaiti. (2021). Revolusi Industri 4.0 : Peran Teknologi Dalam Eksistensi Penguasaan Bisnis Dan Implementasinya. *Jurnal Perilaku dan Strategi Bisnis*. 9(2):91-98. <https://doi.org/10.26486/jpsb.v9i2.2103>
- Sakinah, F. A., Aditiawan, F. P., & Nurlaili, A. L. (2024). PENGUJIAN PADA APLIKASI MANAJEMEN ASET MENGGUNAKAN BLACK BOX TESTING. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(3), 2814–2823. <https://doi.org/10.36040/jati.v8i3.9524>
- Sebayang, F. I. F., & Nst, A. M. U. (2025). Pengaruh kualitas produk, citra merek dan kepercayaan merek terhadap keputusan pembelian layanan internet Telkom Indihome. *RIGGS Journal of Artificial Intelligence and Digital Business*, 4(2), 860–867. <https://doi.org/10.31004/riggs.v4i2.578>.
- Siregar F. A. & Qonitah N. (2022). Perbandingan pengalaman pengguna dalam menggunakan mesin pencari desktop dan mobile. *SINTESIA: Jurnal Sistem dan Teknologi Informasi Indonesia*. 2(1):17-22.
- Soeharso S. Y., & Sulaksono S. (2022). *Supervisor 4.0, Menjawab Tantangan Menjadi Supervisor di Era Digital*. Andi Offset, Yogyakarta. 10-20.