Jurnal Teknik Informatika dan Teknologi Informasi
Volume. 6 Nomor. 2 Agustus 2026
e-ISSN: 2827-9387, p-ISSN: 2827-9379, Hal. 159-175
DOI: https://doi.org/10.55606/jutiti.v6i2.7413
Tersedia: https://journalshub.org/index.php/jutiti

Pengembangan Sistem Pendukung Keputusan Kontinu (*Continuous Decision Support System*) Berbasis Otomatisasi Laporan Konsolidasi menggunakan Google Apps Script dan Telegram API

Jaka Marfenda Satriana[1*], Linda Wahyu Widianti[2]
[1,2]Program Studi Teknologi Indormasi, Sekolah Tinggi Manajemen Informatika dan Komputer (STI&K)
**Penulis Korespondensi: pbmndpdri@gmail.com*

Abstract. *The manual process of consolidated reporting often results in delays in information delivery, data inconsistencies, and low effectiveness in distributing reports to decision-makers. This study aims to develop a Continuous Decision Support System based on automated consolidated reporting using Google Apps Script and the Telegram API, and to analyze it from an Enterprise Architecture perspective. The research employs a Design Science Research approach through the stages of problem identification, architectural design, system development, testing, and evaluation. The system is designed by utilizing Google Sheets as the data source, Google Apps Script as the automation engine, and the Telegram API as the information distribution channel. The results show that the proposed system is capable of integrating data reading, validation, report consolidation, summary generation, and automated notification delivery processes. From an Enterprise Architecture perspective, the system forms a lightweight integration architecture that connects the business, data, application, and technology layers. This study concludes that the integration of Google Apps Script and the Telegram API can support faster, more structured, and continuous information delivery for decision-makers.*

Keywords: *Continuous Decision Support System; Enterprise Architecture; Google Apps Script; Report Automation; Telegram API.*

Abstrak. Proses pelaporan konsolidasi yang masih dilakukan secara manual sering menimbulkan keterlambatan informasi, inkonsistensi data, dan rendahnya efektivitas distribusi laporan kepada pengambil keputusan. Penelitian ini bertujuan mengembangkan Sistem Pendukung Keputusan Kontinu berbasis otomasi laporan konsolidasi menggunakan Google Apps Script dan Telegram API serta menganalisisnya dalam perspektif Enterprise Architecture. Metode penelitian menggunakan pendekatan Design Science Research melalui tahapan identifikasi masalah, perancangan arsitektur, pengembangan sistem, pengujian, dan evaluasi. Sistem dirancang dengan memanfaatkan Google Sheets sebagai sumber data, Google Apps Script sebagai mesin otomasi, dan Telegram API sebagai kanal distribusi informasi. Hasil penelitian menunjukkan bahwa sistem yang diusulkan mampu mengintegrasikan proses pembacaan data, validasi, konsolidasi laporan, pembentukan ringkasan, dan pengiriman notifikasi secara otomatis. Dalam perspektif Enterprise Architecture, sistem ini membentuk arsitektur integrasi ringan yang menghubungkan lapisan bisnis, data, aplikasi, dan teknologi. Penelitian ini menyimpulkan bahwa integrasi Google Apps Script dan Telegram API dapat mendukung penyediaan informasi yang lebih cepat, terstruktur, dan berkelanjutan bagi pengambil keputusan.

Kata Kunci: *Enterprise Architecture*; Google Apps Script; Otomasi Laporan; Sistem Pendukung Keputusan Kontinu; Telegram API.

1. LATAR BELAKANG

Transformasi digital telah mendorong organisasi untuk menata ulang proses bisnis, tata kelola data, aplikasi, dan infrastruktur teknologi agar mampu merespons kebutuhan informasi secara lebih cepat dan akurat. Dalam konteks organisasi modern, pengambilan keputusan tidak lagi cukup bergantung pada laporan periodik yang disusun secara manual, tetapi membutuhkan mekanisme informasi yang terintegrasi, terotomasi, dan dapat diperbarui secara berkelanjutan. Kajian terbaru menunjukkan bahwa transformasi digital tidak hanya berkaitan dengan adopsi teknologi, tetapi juga membutuhkan kombinasi antara perencanaan strategis, konsolidasi sumber daya digital, kesiapan organisasi, dan pemanfaatan teknologi secara terpadu. (Zhou et

Naskah Masuk: 15 April 2026; Revisi: 16 Mei 2026; Diterima: 14 Juni 2026; Tersedia: 04 Juli 2026

al., 2024) menegaskan bahwa tidak ada satu faktor tunggal yang cukup untuk mendorong transformasi digital perusahaan; keberhasilannya ditentukan oleh konfigurasi beberapa faktor organisasi dan teknologi secara simultan.

Salah satu persoalan yang masih sering muncul dalam organisasi adalah proses pelaporan konsolidasi yang terfragmentasi. Laporan dari berbagai unit kerja sering kali dikumpulkan melalui format yang berbeda, diperbarui secara tidak seragam, dan dikonsolidasikan secara manual oleh admin atau pengelola data. Kondisi tersebut berpotensi menimbulkan keterlambatan laporan, kesalahan rekapitulasi, duplikasi pekerjaan, serta rendahnya ketertelusuran informasi. Dalam perspektif sistem informasi, persoalan ini bukan hanya persoalan teknis, melainkan persoalan arsitektural karena menunjukkan belum optimalnya hubungan antara proses bisnis, struktur data, aplikasi pendukung, dan teknologi yang digunakan. Dengan demikian, otomasi laporan konsolidasi perlu dipahami sebagai bagian dari upaya membangun arsitektur informasi organisasi yang lebih terintegrasi.

Enterprise Architecture menjadi pendekatan yang relevan untuk menjawab persoalan tersebut karena menyediakan kerangka untuk menyelaraskan strategi organisasi dengan teknologi informasi. Möhring et al (2023) menjelaskan bahwa banyak organisasi menggunakan Enterprise Architecture Management untuk menyinkronkan strategi korporat dan teknologi informasi, terutama dalam proyek digitalisasi yang mengubah proses bisnis, layanan, dan model kerja organisasi. Namun, kajian tersebut juga menegaskan bahwa proyek digitalisasi sering gagal atau tidak menghasilkan nilai yang diharapkan apabila tantangan arsitektural seperti kompleksitas, dokumentasi arsitektur, fleksibilitas, dan siklus perbaikan berkelanjutan tidak dikelola dengan baik. Oleh karena itu, pengembangan sistem otomasi laporan tidak semestinya hanya diposisikan sebagai pembangunan aplikasi, tetapi sebagai bagian dari rancangan Enterprise Architecture yang menghubungkan arsitektur bisnis, data, aplikasi, dan teknologi.

Dalam penelitian ini, Enterprise Architecture digunakan sebagai perspektif konseptual untuk memetakan sistem pendukung keputusan kontinu ke dalam beberapa lapisan utama. Pada lapisan bisnis, sistem diarahkan untuk memperbaiki proses pelaporan konsolidasi dan distribusi informasi kepada pengambil keputusan. Pada lapisan data, sistem berfungsi menata struktur laporan, validasi data, dan konsolidasi informasi. Pada lapisan aplikasi, Google Apps Script digunakan sebagai mesin otomasi dan penghubung proses, sedangkan Telegram API digunakan sebagai kanal penyampaian informasi. Pada lapisan teknologi, sistem memanfaatkan layanan berbasis cloud dan mekanisme integrasi API. Pemetaan ini sejalan dengan Srisawat et al (2024), yang menunjukkan bahwa Enterprise Architecture dalam

transformasi digital dapat mencakup dimensi business architecture, data architecture, application architecture, technology architecture, security architecture, human capital architecture, dan infrastructure architecture. Kajian tersebut juga menegaskan bahwa EA penting untuk mengintegrasikan teknologi digital dengan alur kerja dan misi organisasi secara sistematis.

Kebutuhan akan sistem pendukung keputusan juga semakin kuat karena organisasi membutuhkan informasi yang tidak hanya tersedia, tetapi juga dapat digunakan dalam setiap tahapan pengambilan keputusan. Szukits & Móricz (2024) menunjukkan bahwa dukungan manajemen terhadap business analytics dan persepsi terhadap kualitas data berperan penting dalam membentuk budaya pengambilan keputusan analitis. Kajian tersebut menempatkan data-driven decision making sebagai proses yang menggunakan data dalam berbagai tahap keputusan, bukan hanya pada tahap pelaporan akhir. Berdasarkan pemikiran tersebut, sistem pendukung keputusan kontinu dalam penelitian ini dipahami sebagai mekanisme yang memungkinkan data laporan dikumpulkan, dikonsolidasikan, diproses, dan didistribusikan secara berulang sehingga pengambil keputusan memperoleh informasi yang lebih cepat dan relevan.

Pemanfaatan Google Apps Script dan Telegram API dipilih karena keduanya dapat membentuk arsitektur integrasi ringan yang sesuai untuk organisasi yang belum memiliki infrastruktur enterprise berskala besar. Google Apps Script dapat digunakan untuk mengotomatisasi proses berbasis Google Workspace, khususnya manipulasi data pada Google Sheets, integrasi proses, dan pengurangan pekerjaan manual. Penelitian Wulan & Dirgahayu (2024) menunjukkan bahwa Google Apps Script mampu digunakan untuk mengotomatisasi proses pendanaan sumber daya manusia, mempercepat pengajuan dan persetujuan, mengurangi kesalahan manusia, serta meningkatkan transparansi dan pemantauan status secara real-time. Sementara itu, penelitian (Bintang et al., 2025) menunjukkan bahwa integrasi Telegram Bot dengan Google Sheets dapat mendukung pelaporan, pengelolaan data, penyimpanan real-time, dan pembuatan laporan otomatis pada layanan pengiriman. Temuan lain dari Alfareza & Aditya (2024) juga memperlihatkan bahwa Google Apps Script, Telegram Bot, dan sumber data operasional dapat diintegrasikan untuk menyajikan laporan performa secara instan dan sistematis.

Meskipun sejumlah penelitian telah membahas Enterprise Architecture dalam transformasi digital, otomasi proses menggunakan Google Apps Script, serta integrasi Telegram Bot dengan Google Sheets, masih terdapat ruang kajian pada pengembangan sistem pendukung keputusan kontinu yang secara eksplisit menggabungkan otomasi laporan

konsolidasi dengan perspektif Enterprise Architecture. Sebagian penelitian sebelumnya cenderung menempatkan Google Apps Script dan Telegram Bot sebagai solusi teknis untuk pelaporan atau notifikasi, tetapi belum banyak yang memosisikannya sebagai artefak arsitektural yang menghubungkan proses bisnis, data, aplikasi, dan teknologi dalam satu rancangan sistem pendukung keputusan. Dengan kata lain, celah penelitian terletak pada belum kuatnya pembahasan mengenai bagaimana arsitektur integrasi ringan berbasis cloud scripting dan messaging API dapat mendukung siklus keputusan organisasi secara kontinu.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk mengembangkan Sistem Pendukung Keputusan Kontinu (Continuous Decision Support System) berbasis otomasi laporan konsolidasi menggunakan Google Apps Script dan Telegram API, serta menganalisis rancangan sistem tersebut dalam perspektif Enterprise Architecture. Penelitian ini diharapkan memberikan kontribusi pada dua aspek. Pertama, kontribusi praktis berupa rancangan sistem otomasi laporan konsolidasi yang dapat mempercepat distribusi informasi kepada pengambil keputusan. Kedua, kontribusi konseptual berupa pemetaan sistem ke dalam lapisan Enterprise Architecture sehingga Google Apps Script dan Telegram API tidak hanya dipahami sebagai teknologi implementatif, tetapi sebagai bagian dari arsitektur enterprise ringan yang mendukung integrasi proses bisnis, tata kelola data, aplikasi, dan infrastruktur teknologi.

2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan Design Science Research (DSR) karena berorientasi pada pengembangan artefak sistem informasi untuk menyelesaikan permasalahan pelaporan konsolidasi dalam organisasi. Artefak yang dikembangkan berupa Sistem Pendukung Keputusan Kontinu (Continuous Decision Support System) berbasis otomasi laporan konsolidasi menggunakan Google Apps Script dan Telegram API.

Pendekatan DSR dipilih karena penelitian ini tidak hanya bertujuan menganalisis permasalahan pelaporan, tetapi juga merancang, membangun, dan mengevaluasi sistem sebagai solusi teknologi informasi. Dalam penelitian ini, sistem yang dikembangkan tidak diposisikan hanya sebagai aplikasi teknis, tetapi juga sebagai artefak arsitektural yang dianalisis melalui perspektif Enterprise Architecture. Oleh karena itu, proses pengembangan sistem memperhatikan keterhubungan antara proses bisnis, struktur data, aplikasi pendukung, dan infrastruktur teknologi.

Penelitian dilakukan melalui lima tahap utama, yaitu identifikasi masalah, perancangan Enterprise Architecture, pengembangan sistem, pengujian sistem, dan evaluasi sistem. Tahap

identifikasi masalah dilakukan dengan menganalisis proses pelaporan konsolidasi yang masih berjalan secara manual, termasuk proses pengumpulan data, rekapitulasi, validasi, dan distribusi laporan kepada pengambil keputusan. Tahap ini bertujuan untuk menemukan kendala utama dalam proses pelaporan, seperti keterlambatan konsolidasi, inkonsistensi format data, risiko kesalahan rekapitulasi, dan belum terintegrasinya proses bisnis dengan teknologi pendukung.

Tahap berikutnya adalah perancangan Enterprise Architecture. Pada tahap ini, sistem dipetakan ke dalam empat lapisan arsitektur, yaitu Business Architecture, Data Architecture, Application Architecture, dan Technology Architecture. Business Architecture digunakan untuk memetakan perubahan alur pelaporan dari proses manual menjadi proses otomatis. Data Architecture digunakan untuk merancang struktur data laporan, validasi data, dan format konsolidasi. Application Architecture digunakan untuk menjelaskan peran Google Apps Script sebagai mesin otomasi dan Telegram API sebagai kanal distribusi informasi. Technology Architecture digunakan untuk menjelaskan penggunaan layanan berbasis cloud, trigger otomatis, dan integrasi API tanpa server khusus.

Tahap pengembangan sistem dilakukan dengan menggunakan Google Sheets sebagai sumber dan media penyimpanan data, Google Apps Script sebagai mesin otomasi, serta Telegram API sebagai kanal pengiriman notifikasi. Google Apps Script digunakan untuk membaca data laporan, melakukan validasi, mengonsolidasikan data, membentuk ringkasan laporan, dan mengirimkan hasil konsolidasi melalui Telegram. Sistem dirancang agar dapat berjalan berdasarkan pemicu tertentu, baik berbasis waktu maupun perubahan data, sehingga proses pelaporan dapat berlangsung secara lebih kontinu.

Setelah sistem dikembangkan, tahap pengujian dilakukan untuk memastikan bahwa fungsi utama sistem berjalan sesuai kebutuhan. Komponen yang diuji meliputi pembacaan data dari Google Sheets, validasi kelengkapan data, konsolidasi laporan, pembentukan pesan ringkasan, pengiriman notifikasi melalui Telegram API, dan eksekusi otomatis melalui trigger Google Apps Script. Tahap terakhir adalah evaluasi sistem, yaitu membandingkan kondisi sebelum dan sesudah sistem dikembangkan berdasarkan indikator efektivitas proses, keandalan notifikasi, pengurangan tahapan manual, konsistensi data, dan kesesuaian sistem dengan perspektif Enterprise Architecture.

Tabel berikut dapat digunakan untuk merangkum tahapan penelitian.

Tabel 1. Tahapan Penelitian Berbasis Design Science Research.

Tahap Penelitian	Kegiatan Utama	Luaran
Identifikasi masalah	Menganalisis proses pelaporan manual dan fragmentasi data	Masalah dan kebutuhan sistem
Perancangan Enterprise Architecture	Memetakan lapisan bisnis, data, aplikasi, dan teknologi	Model arsitektur sistem
Pengembangan sistem	Mengimplementasikan Google Apps Script dan Telegram API	Prototipe sistem
Pengujian sistem	Menguji fungsi konsolidasi dan pengiriman notifikasi	Hasil pengujian sistem
Evaluasi sistem	Membandingkan proses sebelum dan sesudah sistem	Nilai efektivitas sistem

Data penelitian dikumpulkan melalui observasi proses, dokumentasi, dan pengujian sistem. Observasi proses dilakukan untuk mengetahui alur pelaporan sebelum sistem dikembangkan, mulai dari pengumpulan data, rekapitulasi, validasi, hingga distribusi laporan. Dokumentasi digunakan untuk mengidentifikasi struktur laporan, format data, jenis informasi yang dikonsolidasikan, serta kebutuhan informasi bagi pengambil keputusan.

Data teknis diperoleh melalui pengujian sistem, terutama dari log eksekusi Google Apps Script, status pengiriman pesan Telegram, waktu pemrosesan laporan, dan hasil validasi data. Data tersebut digunakan untuk menilai apakah sistem mampu menjalankan fungsi otomasi laporan konsolidasi secara konsisten. Apabila sistem diterapkan kepada pengguna nyata, data tambahan dapat diperoleh melalui wawancara singkat atau kuesioner untuk mengetahui persepsi pengguna terhadap kemudahan penggunaan dan manfaat sistem dalam mendukung proses pengambilan keputusan.

Data dianalisis secara deskriptif kuantitatif dan deskriptif kualitatif. Analisis deskriptif kuantitatif digunakan untuk menilai kinerja sistem berdasarkan indikator teknis, seperti waktu konsolidasi laporan, jumlah pesan yang berhasil dikirim, dan tingkat keberhasilan proses otomasi. Analisis deskriptif kualitatif digunakan untuk menjelaskan kontribusi sistem dalam perspektif Enterprise Architecture, terutama bagaimana sistem menghubungkan proses bisnis, struktur data, aplikasi, dan teknologi.

Evaluasi sistem dilakukan dengan membandingkan kondisi sebelum dan sesudah sistem dikembangkan. Indikator evaluasi meliputi waktu konsolidasi laporan, keberhasilan pengiriman notifikasi Telegram, pengurangan tahapan manual, konsistensi format data, dan kesesuaian sistem dengan lapisan Enterprise Architecture. Efisiensi waktu konsolidasi dihitung dengan membandingkan durasi proses pelaporan sebelum dan sesudah otomasi. Keberhasilan notifikasi dihitung berdasarkan jumlah pesan yang berhasil dikirim dibandingkan dengan jumlah pesan yang seharusnya dikirim.

Secara umum, evaluasi sistem diarahkan untuk menilai dua aspek utama. Pertama, aspek teknis, yaitu kemampuan sistem dalam mengotomatisasi proses konsolidasi laporan dan mengirimkan informasi melalui Telegram. Kedua, aspek arsitektural, yaitu kemampuan sistem dalam membentuk arsitektur enterprise ringan yang mengintegrasikan lapisan bisnis, data, aplikasi, dan teknologi. Dengan demikian, keberhasilan sistem tidak hanya diukur dari berjalannya fungsi teknis, tetapi juga dari kontribusinya dalam memperkuat integrasi arsitektur informasi organisasi.

3. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil pengembangan Sistem Pendukung Keputusan Kontinu berbasis otomasi laporan konsolidasi menggunakan Google Apps Script dan Telegram API. Hasil penelitian difokuskan pada rancangan arsitektur sistem, pemetaan Enterprise Architecture, implementasi alur otomasi, serta pengujian sistem menggunakan data simulasi. Penggunaan data simulasi dilakukan untuk merepresentasikan proses pelaporan konsolidasi dari beberapa unit kerja dan digunakan sebagai dasar pengujian fungsi sistem, bukan untuk menggambarkan kondisi empiris organisasi tertentu.

Kondisi Awal Proses Pelaporan Konsolidasi

Skenario awal dalam penelitian ini menggambarkan proses pelaporan konsolidasi yang masih dilakukan secara manual menggunakan spreadsheet. Setiap unit kerja mengisi laporan pada dokumen masing-masing, kemudian data tersebut direkap oleh pengelola laporan untuk disatukan ke dalam laporan konsolidasi. Proses ini memiliki beberapa kelemahan, antara lain perbedaan format antarunit, keterlambatan pengiriman laporan, risiko kesalahan rekapitulasi, serta distribusi informasi yang belum berlangsung secara otomatis.

Dalam perspektif Enterprise Architecture, kondisi tersebut menunjukkan adanya fragmentasi antara proses bisnis, data, aplikasi, dan teknologi. Pada lapisan bisnis, proses pelaporan belum sepenuhnya terstandar karena masih bergantung pada intervensi manual. Pada lapisan data, format laporan belum sepenuhnya seragam sehingga menyulitkan proses konsolidasi. Pada lapisan aplikasi, spreadsheet hanya digunakan sebagai media input dan penyimpanan data, belum berfungsi sebagai bagian dari sistem otomasi. Pada lapisan teknologi, belum terdapat mekanisme integrasi yang mampu menghubungkan data laporan dengan kanal distribusi informasi secara langsung.

Kondisi tersebut menjadi dasar pengembangan sistem dalam penelitian ini. Sistem yang dikembangkan diarahkan untuk mengubah proses pelaporan dari manual menjadi otomatis melalui integrasi Google Sheets, Google Apps Script, dan Telegram API. Dengan demikian,

otomasi laporan tidak hanya berfungsi untuk mempercepat pekerjaan administratif, tetapi juga membentuk arsitektur integrasi ringan yang mendukung siklus pengambilan keputusan secara kontinu.

Kebutuhan Sistem Pendukung Keputusan Kontinu

Berdasarkan skenario proses pelaporan konsolidasi, sistem yang dikembangkan perlu memenuhi kebutuhan fungsional dan nonfungsional. Kebutuhan fungsional berkaitan dengan fungsi utama sistem dalam membaca data, memvalidasi laporan, mengonsolidasikan informasi, membentuk ringkasan, dan mengirimkan notifikasi kepada pengambil keputusan. Sementara itu, kebutuhan nonfungsional berkaitan dengan kemudahan penggunaan, ketersediaan sistem, efisiensi infrastruktur, dan fleksibilitas pengembangan.

Tabel 2. Kebutuhan Sistem Pendukung Keputusan Kontinu.

Jenis Kebutuhan	Deskripsi Kebutuhan	Implementasi dalam Sistem
Pembacaan data	Sistem mampu membaca data laporan dari Google Sheets	Google Apps Script membaca data berdasarkan rentang dan struktur kolom tertentu
Validasi data	Sistem mampu mendeteksi data yang belum lengkap atau tidak sesuai format	Script memeriksa kelengkapan isian, periode laporan, dan status unit kerja
Konsolidasi laporan	Sistem mampu menggabungkan data dari beberapa unit kerja	Data dari beberapa sheet atau tabel digabungkan ke dalam format ringkasan
Pembentukan ringkasan	Sistem mampu menghasilkan informasi singkat yang relevan bagi pengambil keputusan	Ringkasan berisi status laporan, jumlah unit, indikator utama, dan peringatan
Pengiriman notifikasi	Sistem mampu mengirimkan hasil laporan ke Telegram	Telegram API mengirim pesan ke pengguna atau grup yang telah ditentukan
Eksekusi otomatis	Sistem mampu berjalan tanpa perintah manual	Trigger Google Apps Script menjalankan sistem berdasarkan waktu atau perubahan data

Kebutuhan tersebut menunjukkan bahwa sistem tidak hanya ditujukan untuk menyimpan data, tetapi juga untuk membangun alur informasi yang berulang dan terotomasi. Dalam konteks ini, konsep “kontinu” dipahami sebagai kemampuan sistem untuk menjalankan proses pembacaan data, validasi, konsolidasi, dan distribusi informasi secara berkala atau berdasarkan pemicu tertentu. Dengan mekanisme tersebut, pengambil keputusan tidak harus menunggu laporan manual, tetapi dapat menerima informasi yang telah dikonsolidasikan secara lebih cepat.

Perancangan Enterprise Architecture Sistem

Perancangan Enterprise Architecture dilakukan untuk memastikan bahwa sistem yang dikembangkan selaras dengan kebutuhan organisasi. Pemetaan dilakukan pada empat lapisan utama, yaitu Business Architecture, Data Architecture, Application Architecture, dan Technology Architecture. Pemetaan ini penting karena pengembangan sistem tidak hanya berkaitan dengan pembuatan aplikasi, tetapi juga dengan penyusunan hubungan antara proses kerja, struktur data, aplikasi pendukung, dan infrastruktur teknologi.

Tabel 3. Pemetaan Enterprise Architecture Sistem.

Lapisan EA	Kondisi Awal	Rancangan Sistem
Business Architecture	Proses pelaporan dilakukan manual, bergantung pada rekapitulasi admin, dan distribusi informasi belum otomatis	Alur pelaporan dibuat otomatis melalui pembacaan data, konsolidasi, dan pengiriman informasi ke Telegram
Data Architecture	Data laporan tersebar, format antarunit dapat berbeda, dan validasi dilakukan manual	Struktur data distandarkan pada Google Sheets dengan format kolom dan status validasi yang seragam
Application Architecture	Spreadsheet hanya digunakan sebagai media input dan penyimpanan data	Google Apps Script berfungsi sebagai mesin otomasi, sedangkan Telegram API berfungsi sebagai kanal distribusi informasi
Technology Architecture	Belum terdapat integrasi otomatis antara data laporan dan kanal komunikasi	Sistem menggunakan cloud-based spreadsheet, trigger otomatis, dan integrasi API tanpa server khusus

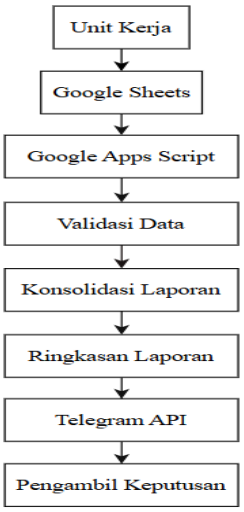
Hasil pemetaan menunjukkan bahwa sistem yang dikembangkan membentuk arsitektur integrasi ringan. Pada lapisan bisnis, sistem memperbaiki alur pelaporan dengan mengurangi ketergantungan pada proses manual. Pada lapisan data, sistem menstandarkan format laporan agar dapat diproses secara otomatis. Pada lapisan aplikasi, Google Apps Script berperan sebagai pengolah logika otomasi, sedangkan Telegram API menjadi media distribusi informasi. Pada lapisan teknologi, sistem memanfaatkan layanan berbasis cloud sehingga tidak membutuhkan infrastruktur server khusus.

Pemetaan tersebut sejalan dengan pandangan bahwa Enterprise Architecture berperan dalam menyelaraskan strategi organisasi dan teknologi informasi, terutama dalam konteks digitalisasi proses bisnis (Möhring et al., 2023). Selain itu, pemetaan pada lapisan bisnis, data, aplikasi, dan teknologi juga mendukung pendekatan Enterprise Architecture yang menekankan integrasi berbagai komponen organisasi dalam transformasi digital (Srisawat et al., 2024).

Desain Arsitektur Sistem

Sistem dirancang dengan alur kerja yang menghubungkan sumber data, mesin otomasi, proses validasi, proses konsolidasi, dan kanal distribusi informasi. Google Sheets digunakan sebagai sumber data laporan, Google Apps Script digunakan sebagai mesin pemrosesan, dan Telegram API digunakan sebagai kanal pengiriman ringkasan laporan.

Alur sistem dapat digambarkan sebagai berikut:



Gambar 1. Desain Arsitektur Sistem Pendukung Keputusan Kontinu.

Gambar tersebut dapat divisualisasikan dalam bentuk diagram alur yang menunjukkan hubungan antara unit kerja, Google Sheets, Google Apps Script, Telegram API, dan pengambil keputusan. Pada tahap awal, unit kerja memasukkan data laporan ke Google Sheets. Selanjutnya, Google Apps Script membaca data tersebut berdasarkan struktur yang telah ditentukan. Sistem kemudian melakukan validasi untuk memastikan bahwa data telah lengkap dan sesuai format. Setelah data dinyatakan valid, sistem melakukan konsolidasi dan membentuk ringkasan laporan. Ringkasan tersebut kemudian dikirimkan secara otomatis kepada pengambil keputusan melalui Telegram API.

Desain arsitektur ini menunjukkan bahwa sistem bekerja sebagai continuous decision loop. Setiap kali data diperbarui atau trigger dijalankan, sistem dapat memproses ulang data dan mengirimkan informasi terbaru. Dengan demikian, pengambil keputusan memperoleh informasi secara lebih cepat dibandingkan dengan proses pelaporan manual.

Implementasi Google Apps Script dan Telegram API

Implementasi sistem dilakukan dengan memanfaatkan Google Apps Script sebagai komponen utama otomasi. Google Apps Script digunakan untuk menjalankan beberapa fungsi, yaitu membaca data dari Google Sheets, memeriksa kelengkapan data, menggabungkan laporan dari beberapa unit, menyusun ringkasan laporan, dan mengirimkan pesan melalui Telegram API.

Pada proses pembacaan data, sistem mengambil data dari sheet yang telah ditentukan. Data yang dibaca meliputi identitas unit, periode laporan, indikator laporan, nilai capaian, status validasi, dan catatan tambahan. Setelah data terbaca, sistem melakukan validasi untuk

mendeteksi data kosong, format yang tidak sesuai, atau laporan yang belum dikirim oleh unit tertentu.

Pada proses konsolidasi, sistem menggabungkan data dari beberapa unit ke dalam satu ringkasan. Ringkasan tersebut dapat berisi jumlah unit yang telah mengirim laporan, jumlah unit yang belum lengkap, nilai indikator tertentu, serta peringatan apabila ditemukan data yang perlu ditindaklanjuti. Setelah ringkasan terbentuk, sistem mengirimkan pesan ke Telegram menggunakan token bot dan chat ID yang telah dikonfigurasi.

Format pesan Telegram dirancang singkat agar mudah dibaca oleh pengambil keputusan. Contoh struktur pesan yang dihasilkan sistem adalah sebagai berikut:

Laporan Konsolidasi Periode: [Periode]

Jumlah unit terlapor: [jumlah unit]

• Data lengkap: [jumlah unit]

• Data perlu validasi: [jumlah unit]

• Indikator utama: [ringkasan indikator]

• Catatan sistem: [peringatan atau rekomendasi tindak lanjut]

Format tersebut mendukung distribusi informasi secara cepat karena pengambil keputusan tidak perlu membuka spreadsheet secara langsung untuk memperoleh ringkasan awal. Apabila diperlukan, pengguna tetap dapat membuka dokumen sumber untuk memeriksa data secara lebih rinci.

Pengujian Sistem Menggunakan Data Simulasi

Pengujian sistem dilakukan menggunakan data simulasi yang merepresentasikan laporan dari beberapa unit kerja. Data simulasi disusun untuk menguji kemampuan sistem dalam membaca data, memvalidasi kelengkapan laporan, mengonsolidasikan informasi, membentuk ringkasan, dan mengirimkan notifikasi melalui Telegram API. Dalam skenario pengujian, data terdiri atas beberapa unit kerja dengan variasi status laporan, yaitu lengkap, belum lengkap, dan perlu validasi.

Pengujian dilakukan pada enam fungsi utama sistem. Hasil pengujian menunjukkan bahwa seluruh fungsi utama dapat berjalan sesuai skenario yang dirancang.

Tabel 4. Hasil Pengujian Fungsional Sistem.

Fungsi yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
Pembacaan data	Sistem membaca data dari Google Sheets	Data terbaca sesuai struktur kolom	Data berhasil dibaca oleh sistem	Berhasil
Validasi data	Sistem memeriksa kelengkapan laporan	Data kosong atau tidak lengkap terdeteksi	Sistem memberikan status perlu validasi	Berhasil
Konsolidasi laporan	Sistem menggabungkan data dari beberapa unit	Ringkasan konsolidasi terbentuk	Ringkasan laporan berhasil dibuat	Berhasil
Pembentukan pesan	Sistem menyusun pesan laporan	Pesan sesuai format yang ditentukan	Pesan terbentuk dengan struktur ringkas	Berhasil
Pengiriman Telegram	Sistem mengirim pesan melalui Telegram API	Pesan diterima pada akun atau grup tujuan	Pesan berhasil diterima di Telegram	Berhasil
Trigger otomatis	Sistem berjalan berdasarkan pemicu waktu	Sistem berjalan tanpa eksekusi manual	Trigger berhasil menjalankan script	Berhasil

Hasil pengujian menunjukkan bahwa integrasi antara Google Sheets, Google Apps Script, dan Telegram API dapat digunakan untuk membangun alur otomasi laporan konsolidasi. Fungsi validasi membantu mengurangi risiko kesalahan akibat data yang belum lengkap, sedangkan fungsi pengiriman Telegram mempercepat distribusi informasi kepada pengambil keputusan.

Evaluasi Efektivitas Sistem

Evaluasi efektivitas sistem dilakukan dengan membandingkan proses pelaporan sebelum dan sesudah otomasi berdasarkan skenario uji. Karena penelitian ini menggunakan data simulasi, hasil evaluasi tidak dimaksudkan untuk mewakili kinerja organisasi tertentu, tetapi untuk menunjukkan perubahan proses yang dihasilkan oleh sistem.

Tabel 5. Perbandingan Proses Sebelum dan Sesudah Otomasi.

Indikator Evaluasi	Sebelum Otomasi	Sesudah Otomasi	Dampak Sistem
Pengumpulan data	Dilakukan melalui input dan pengecekan manual	Data dibaca langsung dari Google Sheets	Proses lebih terpusat
Validasi laporan	Dilakukan oleh admin secara manual	Dilakukan otomatis oleh Google Apps Script	Risiko kelalaian validasi berkurang
Konsolidasi laporan	Dilakukan dengan rekapitulasi manual	Dilakukan otomatis berdasarkan struktur data	Proses lebih cepat dan konsisten
Distribusi informasi	Dikirim manual melalui pesan atau dokumen	Dikirim otomatis melalui Telegram API	Informasi lebih cepat diterima
Ketergantungan pada admin	Tinggi	Berkurang	Proses lebih mandiri
Ketersediaan informasi	Bergantung pada waktu penyusunan laporan	Dapat diperbarui berdasarkan trigger	Mendukung pengambilan keputusan kontinu

Hasil evaluasi menunjukkan bahwa sistem mampu mengurangi tahapan manual dalam proses pelaporan. Sebelum otomasi, proses pelaporan membutuhkan pengumpulan data, pengecekan format, rekapitulasi, penyusunan ringkasan, dan pengiriman laporan secara

manual. Setelah sistem dikembangkan, sebagian besar proses tersebut dapat dilakukan secara otomatis oleh Google Apps Script. Telegram API kemudian mempercepat distribusi informasi karena ringkasan laporan dapat dikirim langsung kepada pengambil keputusan.

Dari sisi efisiensi proses, sistem mengurangi beban kerja administratif pada proses validasi dan konsolidasi. Dari sisi konsistensi data, penggunaan struktur Google Sheets yang terstandar membuat sistem dapat membaca dan memproses data dengan pola yang sama. Dari sisi distribusi informasi, penggunaan Telegram API memungkinkan informasi laporan diterima secara lebih cepat tanpa menunggu pengiriman manual.

Pembahasan dalam Perspektif Enterprise Architecture

Hasil pengembangan sistem menunjukkan bahwa otomasi laporan konsolidasi dapat dipahami sebagai bagian dari Enterprise Architecture, bukan sekadar implementasi teknis. Sistem yang dikembangkan menghubungkan proses bisnis, data, aplikasi, dan teknologi ke dalam satu alur terintegrasi. Hal ini relevan dengan konsep Enterprise Architecture yang menekankan pentingnya penyelarasan antara kebutuhan organisasi dan teknologi informasi (Möhring et al., 2023).

Pada lapisan Business Architecture, sistem mengubah alur pelaporan dari proses manual menjadi proses yang lebih otomatis. Perubahan ini tidak hanya berdampak pada efisiensi waktu, tetapi juga mengubah pola kerja dalam distribusi informasi. Pengambil keputusan tidak lagi menunggu laporan disusun secara manual, melainkan dapat menerima ringkasan laporan melalui kanal notifikasi yang telah terintegrasi.

Pada lapisan Data Architecture, sistem menekankan pentingnya standardisasi struktur data. Data yang sebelumnya berpotensi tersebar dalam berbagai format diarahkan ke dalam struktur yang lebih seragam agar dapat dibaca dan diproses secara otomatis. Standardisasi data ini penting karena pengambilan keputusan berbasis data membutuhkan kualitas data, konsistensi, dan ketersediaan informasi yang memadai (Szukits & Móricz, 2024).

Pada lapisan Application Architecture, Google Apps Script dan Telegram API berfungsi sebagai komponen aplikasi yang saling terintegrasi. Google Apps Script menjalankan fungsi otomasi, sedangkan Telegram API menjalankan fungsi distribusi informasi. Integrasi ini memperluas fungsi spreadsheet dari sekadar media penyimpanan data menjadi bagian dari sistem pendukung keputusan. Temuan ini sejalan dengan penelitian sebelumnya yang menunjukkan bahwa Google Apps Script dapat digunakan untuk mengotomatisasi proses administratif dan meningkatkan efisiensi kerja (Wulan & Dirgahayu, 2024). Selain itu, integrasi Telegram Bot dengan Google Sheets juga telah terbukti dapat mendukung pelaporan dan pengelolaan data secara otomatis (Bintang et al., 2025).

Pada lapisan Technology Architecture, sistem menggunakan pendekatan cloud-based dan serverless. Artinya, sistem dapat berjalan tanpa kebutuhan server lokal atau middleware kompleks. Pendekatan ini sesuai untuk organisasi yang membutuhkan solusi integrasi ringan, terutama ketika sumber daya teknologi terbatas. Penggunaan trigger otomatis pada Google Apps Script juga memungkinkan sistem berjalan secara berkala, sehingga mendukung konsep pengambilan keputusan kontinu.

Dibandingkan dengan penelitian sebelumnya yang lebih banyak menempatkan Google Apps Script atau Telegram Bot sebagai alat otomasi dan notifikasi (Wulan & Dirgahayu, 2024; Bintang et al., 2025), penelitian ini menempatkan integrasi tersebut dalam perspektif Enterprise Architecture. Dengan demikian, kontribusi penelitian tidak hanya terletak pada pengembangan sistem otomasi laporan, tetapi juga pada pemetaan sistem sebagai arsitektur integrasi ringan yang menghubungkan lapisan bisnis, data, aplikasi, dan teknologi.

Continuous Decision Support System sebagai Arsitektur Integrasi Ringan

Sistem yang dikembangkan dapat diposisikan sebagai Continuous Decision Support System karena mendukung siklus informasi yang berulang mulai dari input data, validasi, konsolidasi, pembentukan ringkasan, hingga distribusi informasi. Sistem ini tidak menggantikan keputusan manusia, tetapi menyediakan informasi yang lebih cepat dan terstruktur untuk mendukung proses pengambilan keputusan.

Konsep kontinu dalam sistem ini tidak dimaknai sebagai real-time penuh, melainkan sebagai kemampuan sistem untuk berjalan secara berkala atau berdasarkan pemicu tertentu. Dengan demikian, sistem mampu memperbarui informasi dan mengirimkan laporan secara otomatis ketika data tersedia atau ketika jadwal eksekusi terpenuhi. Pola ini mendukung pengambilan keputusan berbasis data karena informasi dapat diterima lebih cepat, lebih terstruktur, dan lebih mudah diakses oleh pengambil keputusan.

Dalam perspektif Enterprise Architecture, sistem ini dapat disebut sebagai lightweight enterprise architecture karena menggunakan teknologi yang relatif sederhana, mudah diakses, dan tidak membutuhkan infrastruktur kompleks. Meskipun demikian, sistem tetap mampu menjalankan fungsi integrasi antara data, proses, aplikasi, dan teknologi. Model ini relevan bagi organisasi yang belum memiliki sistem enterprise berskala besar, tetapi membutuhkan mekanisme otomasi pelaporan dan distribusi informasi yang lebih efektif.

Implikasi Hasil Penelitian

Hasil penelitian ini memiliki implikasi praktis dan konseptual. Secara praktis, sistem yang dikembangkan dapat membantu organisasi mempercepat proses pelaporan konsolidasi, mengurangi ketergantungan pada proses manual, dan mempercepat penyampaian informasi kepada pengambil keputusan. Sistem ini juga dapat diterapkan pada organisasi yang telah menggunakan Google Sheets sebagai media kerja, sehingga proses adopsi teknologi relatif lebih mudah.

Secara konseptual, penelitian ini menunjukkan bahwa teknologi sederhana seperti Google Apps Script dan Telegram API dapat diposisikan sebagai bagian dari Enterprise Architecture apabila dirancang secara sistematis. Dengan pemetaan pada lapisan bisnis, data, aplikasi, dan teknologi, sistem otomasi laporan dapat dipahami sebagai artefak arsitektural yang mendukung integrasi organisasi. Hal ini memperkuat pandangan bahwa Enterprise Architecture tidak selalu harus diwujudkan melalui infrastruktur kompleks, tetapi juga dapat diterapkan melalui arsitektur ringan yang sesuai dengan kebutuhan dan kapasitas organisasi.

Namun, sistem ini memiliki beberapa keterbatasan. Pertama, sistem masih berbasis aturan sederhana sehingga belum mendukung analitik prediktif atau rekomendasi keputusan berbasis kecerdasan buatan. Kedua, sistem bergantung pada layanan Google Workspace dan Telegram API sehingga ketersediaan sistem dipengaruhi oleh koneksi internet dan stabilitas layanan pihak ketiga. Ketiga, pengujian dalam penelitian ini menggunakan data simulasi sehingga belum menggambarkan variasi kompleksitas data pada organisasi nyata. Oleh karena itu, penelitian lanjutan dapat dilakukan dengan menerapkan sistem pada organisasi tertentu, menggunakan data operasional aktual, serta menambahkan fitur analitik lanjutan untuk memperkuat fungsi pendukung keputusan.

4. KESIMPULAN DAN SARAN

Penelitian ini menghasilkan rancangan Sistem Pendukung Keputusan Kontinu berbasis otomasi laporan konsolidasi menggunakan Google Apps Script dan Telegram API yang mampu mengintegrasikan proses pembacaan data, validasi, konsolidasi laporan, pembentukan ringkasan, dan pengiriman notifikasi secara otomatis. Hasil pengembangan menunjukkan bahwa sistem dapat diposisikan sebagai arsitektur integrasi ringan dalam perspektif Enterprise Architecture karena menghubungkan lapisan bisnis, data, aplikasi, dan teknologi dalam satu alur pelaporan yang lebih terstruktur. Dengan demikian, tujuan penelitian untuk mengembangkan sistem otomasi laporan konsolidasi sekaligus menganalisisnya melalui

perspektif Enterprise Architecture telah tercapai, khususnya dalam mendukung penyediaan informasi yang lebih cepat, konsisten, dan berkelanjutan bagi pengambil keputusan.

Penelitian ini masih terbatas pada pengembangan prototipe dan pengujian menggunakan data simulasi, sehingga penelitian lanjutan perlu dilakukan melalui implementasi pada organisasi nyata dengan data operasional aktual. Pengembangan berikutnya juga dapat diarahkan pada integrasi sistem dengan dashboard analitik, basis data terpusat, serta model rekomendasi berbasis analitik atau kecerdasan buatan agar Sistem Pendukung Keputusan Kontinu tidak hanya berfungsi sebagai otomasi laporan, tetapi juga mampu memberikan dukungan keputusan yang lebih adaptif, prediktif, dan berdaya guna bagi organisasi.

DAFTAR REFERENSI

Alfareza, F., & Aditya, A. (2024). Clockify and BOT Telegram integration in employee performance measurement applications development. *International Journal of Innovation in Enterprise System, 8*(1), 101–111. https://doi.org/10.25124/ijies.v8i01.223

Ansyori, R., Qodarsih, N., & Soewito, B. (2018). A systematic literature review: Critical success factors to implement enterprise architecture. *Procedia Computer Science, 135*, 43–51. https://doi.org/10.1016/j.procs.2018.08.148

Bintang, A., Hananto, A. L., & Hananto, A. (2025). Telegram BOT application development integration with Google Sheets for sending service reporting. *Journal of Artificial Intelligence and Engineering Applications (JAIEA), 4*(3), 2208–2214. https://doi.org/10.59934/jaiea.v4i3.1136

Google Developers. (2026). *Apps Script overview*. Google for Developers. https://developers.google.com/apps-script/overview

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly, 28*(1), 75–105. https://doi.org/10.2307/25148625

Möhring, M., Keller, B., Schmidt, R., Sandkuhl, K., & Zimmermann, A. (2023). Digitalization and enterprise architecture management: A perspective on benefits and challenges. *SN Business and Economics, 3*(2), 1–21. https://doi.org/10.1007/s43546-023-00426-3

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems, 24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Sprague, R. H. (1980). A framework for the development of decision support systems. *MIS Quarterly, 4*(4), 1–26. https://doi.org/10.2307/248957

Srisawat, S., Wannapiroon, P., & Nilsook, P. (2024). Distributed digital enterprise architecture for transformation of educational organizations. *TEM Journal, 13*(2), 1645–1657. https://doi.org/10.18421/TEM132-77

Szukits, Á., & Móricz, P. (2024). Towards data-driven decision making: The role of analytical culture and centralization efforts. *Review of Managerial Science, 18*(10). https://doi.org/10.1007/s11846-023-00694-1

Telegram. (n.d.). *Telegram Bot API*. Retrieved June 23, 2026, from https://core.telegram.org/bots/api

The Open Group. (2022). *The TOGAF Standard* (10th ed.). https://www.opengroup.org/togaf

Verhoef, P. C., Broekhuizen, T., Bart, Y., Bhattacharya, A., Dong, J. Q., Fabian, N., & Haenlein, M. (2021). Digital transformation: A multidisciplinary reflection and research agenda. *Journal of Business Research*, *122*, 889–901. https://doi.org/10.1016/j.jbusres.2019.09.022

Vial, G. (2019). Understanding digital transformation: A review and a research agenda. *The Journal of Strategic Information Systems*, *28*(2), 118–144. https://doi.org/10.1016/j.jsis.2019.01.003

Wulan, S., & Dirgahayu, T. (2024). Pengembangan sumber daya manusia menggunakan Google Apps Script. *Edusaintek: Jurnal Pendidikan, Sains dan Teknologi*, *11*(4), 2238–2254. https://doi.org/10.47668/edusaintek.v11i4.1448

Zhou, Y., Chen, L., Zhou, X., & Hu, S. (2024). Key factors and path selection for enterprise digital transformation: Configuration analysis based on fsQCA. *Frontiers in Sustainability*, *5*. https://doi.org/10.3389/frsus.2024.1366129