

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Penelitian tentang penggunaan *cloud* database untuk Remote Patient Monitoring (RPM) telah berkembang pesat dalam beberapa tahun terakhir. Beberapa penelitian terdahulu yang terkait dengan topik penelitian ini:

Tabel 2.1 Penelitian Terdahulu

Ref.	Jurnal	Peneliti	Judul Penelitian	Metodologi Penelitian	Hasil Penelitian
[6]	Frontiers in Digital Health 2024 Vol. 6	Cao et al. (2024)	Developing remote patient monitoring infrastructure using commercially available <i>cloud</i> platforms	Microsoft Azure dan AWS	Rancangan dan implementasi <i>cloud</i> database untuk RPM menggunakan Azure dan AWS berhasil. Penelitian menunjukkan bagaimana RPM dapat dilakukan menggunakan infrastruktur <i>cloud</i> yang tersedia secara komersial dan memberikan perbandingan kinerja serta biaya implementasi
[12]	International Journal of Telemedicine and Applications 2023	Ko et al. (2023)	Real-time remote patient monitoring and alarming system for noncommunicable lifestyle diseases	MySQL database konvensional	Penelitian lebih memfokuskan pada implementasi RPM, bukan penggunaan sistem basis data tertentu.

					Penelitian memberikan temuan berupa bagaimana RPM dapat meningkatkan kualitas layanan terhadap penyakit sehari-hari yang tidak mudah dikomunikasikan
[13]	Pediatric Nephrology 2023 Vol. 8	Chan et al. (2023)	Outcomes and perception of <i>cloud</i> -based remote patient monitoring in children receiving automated peritoneal dialysis: a prospective study	Platform Sharesource ®	Penelitian lebih memfokuskan pada evaluasi RPM berbasis <i>cloud</i> yang mempunyai hasil positif
[14]	Journal of Big Data 2021 Vol. 8	Iranpak et al. (2021)	Remote patient monitoring and classifying using the internet of things platform combined with <i>cloud</i> computing	<i>Cloud</i> computing	Penelitian membuktikan bahwa <i>cloud</i> computing dapat digunakan untuk RPM, tetapi tidak membahas platform <i>cloud</i> yang digunakan ataupun rancangan <i>cloud</i>
[15]	Diagnostics 2021 Vol. 11	El-Rashidy et al. (2021)	Mobile health in remote patient monitoring for chronic diseases: Principles, trends, and challenges	<i>Cloud</i> computing	Penelitian membahas bagaimana <i>cloud</i> computing mempunyai faktor keberhasilan yang lebih tinggi untuk RPM, serta

					mempunyai rancangan dasar arsitektur <i>cloud</i> yang dapat digunakan
[16]	EAI Endorsed Transactions on Pervasive Health and Technology 2021 Vol. 7	Lakshmi et al. (2021)	<i>Cloud</i> based iot smart healthcare system for remote patient monitoring	PaaS <i>Cloud</i>	Penelitian menunjukkan kelebihan penggunaan <i>cloud</i> dengan layanan PaaS (seperti Microsoft Azure) untuk RPM dengan kemudahan penyimpanan, komputasi, analitik, dan visualisasi
[17]	Bulletin of Electrical Engineering and Informatics 2021 Vol. 10	Majeed J., Aish Q. (2021)	A remote patient monitoring based on wban implementation with internet of thing and <i>cloud</i> server	<i>Cloud</i> server	Data pada wearable sensor dapat dikirimkan ke server <i>cloud</i> yang tersimpan pada Raspberry Pi sederhana, kemudian dapat diakses melalui halaman website
[18]	Indonesian Journal of Electrical Engineering and Computer Science 2021 Vol. 22	Falah et al. (2021)	Comparison of <i>cloud</i> computing providers for development of big data and internet of things application	Alibaba, AWS, Google <i>cloud</i> platform, Microsoft Azure, <i>CloudKilat</i>	Azure dan AWS mempunyai fungsi yang lebih spesifik dibandingkan platform lainnya untuk pengembangan aplikasi internet of things
[19]	Applied Sciences	Uddin, R., Koo, I. (2024)	Real-time remote patient monitoring: A review of	Multi-hop system, <i>Cloud</i> secara umum	Penelitian membahas penggunaan IoT dengan

	(Switzerland) 2024 Vol. 14		biosensors integrated with multi- hop iot systems via <i>cloud</i> connectivity		sistem multi- hop menggunakan integrasi <i>cloud</i> untuk RPM secara <i>real-time</i> . Penelitian tidak memberikan rancangan bagaimana sistem <i>cloud</i> dapat digunakan dalam penelitian.
[20]	Personal and Ubiquitous Computing 2023 Vol. 27	Akhbarifar et al. (2020)	A secure remote health monitoring model for early disease diagnosis in <i>cloud</i> -based IoT environment	<i>Cloud</i> -based sistem secara umum	Penelitian membahas rancangan desain sistem <i>cloud</i> untuk IoT medis yang dapat digunakan. Penelitian lebih memfokuskan ke analisis data yang dapat digunakan untuk meningkatkan kualitas layanan kesehatan, tetapi tidak pada pengembang an model <i>cloud</i> yang dapat digunakan.

Penelitian-penelitian tersebut menunjukkan evolusi teknologi RPM yang semakin matang, dengan menggunakan sistem *cloud* computing dan penyimpanan *cloud*. Penelitian pada tabel mempunyai fokus pada analisis terhadap hasil implementasi RPM menggunakan sistem *cloud* computing (*post-implementation*) dan bukan memberikan rancangan arsitektur *cloud* yang dapat

digunakan untuk RPM. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan rancangan sistem *cloud* yang dapat digunakan untuk RPM dengan menggunakan salah satu layanan *cloud* yang tersedia secara komersil, yaitu Microsoft Azure, dalam upaya untuk meningkatkan kualitas layanan kesehatan di Indonesia.

2.2 Cloud Computing

Cloud computing merupakan model komputasi yang memungkinkan akses jaringan on-demand ke kumpulan sumber daya komputasi yang dapat dikonfigurasi (seperti jaringan, server, penyimpanan, aplikasi, dan layanan) yang dapat dengan cepat disediakan dan dirilis dengan upaya manajemen minimal [21].

Cloud computing merupakan teknologi yang mampu menjalankan server, *database*, jaringan, dan software melalui koneksi *online* melalui internet [8]. Dengan menggunakan sistem *cloud* computing yang tersentralisasi, pengumpulan data dari peralatan IoT dapat dilakukan dengan cepat secara real-time. *Cloud* computing dapat memberikan efisiensi optimal, kecepatan, dan akurasi dalam pengolahan data yang dikumpulkan dari perangkat IoT [14]. Teknologi ini memungkinkan transfer data secara *real-time* ke server *cloud* melalui jaringan, sehingga mendukung pemantauan pasien jarak jauh dengan tingkat responsivitas tinggi.

Penggunaan *cloud* storage sebagai database penyimpanan penting dalam menjalankan *cloud* computing [22]. Database *cloud* yang dapat menyimpan jumlah data yang besar dapat menggunakan *cloud* computing untuk mengoptimasi konsistensi dan efektivitas. Microsoft Azure sebagai salah satu penyedia layanan *Cloud* sering digunakan untuk aplikasi secara umum [22]. Sebagai salah satu *platform cloud* yang paling lama berdiri, Microsoft Azure menyediakan layanan *cloud* computing seperti *cloud* storage, database service, analitik, dan jaringan IoT [23]. Layanan *platform-as-a-service* (PaaS) yang disediakan oleh Microsoft Azure memudahkan implementasi software yang diinginkan, sehingga cocok untuk digunakan dalam RPM.

2.3 Remote Patient Monitoring (RPM)

Remote Patient Monitoring (RPM) merupakan salah satu implementasi sistem IoT yang digunakan untuk memonitor dan mendiagnosa pasien dari jarak jauh [24]. Dengan implementasi RPM, dokter atau keluarga pasien dapat mengawasi pasien dari manapun dan kapanpun. Data yang dapat dikumpulkan oleh RPM meliputi detak jantung, tekanan darah, tingkat saturasi oksigen (SpO₂), suhu tubuh, tingkat glukosa darah, dan sebagainya. Dengan memonitor kondisi pasien secara *real-time*, penyakit dapat diidentifikasi lebih awal dengan langkah-langkah pencegahan [19].

Teknologi RPM menggunakan peralatan seperti sensor dan peralatan medis lainnya untuk mengumpulkan data yang dikumpulkan oleh pihak kesehatan untuk mengambil keputusan [25]. Pengumpulan data yang dilakukan dari RPM dapat dilakukan secara instan, sehingga dapat menghindari penyakit sebelum semakin parah, atau bahkan kematian.

Infrastruktur RPM terdiri dari empat komponen utama: pengumpulan data, transmisi dan penyimpanan data, analisis data, dan presentasi informasi [26]. Data yang dikumpulkan melalui sensor ditransmisikan ke server *cloud* untuk disimpan dan dianalisis. Melalui sebuah interface seperti website, informasi dapat ditampilkan dan divisualisasikan sebagai bahan acuan dalam mengambil keputusan.

Penelitian menunjukkan bahwa program RPM dalam jangka panjang dapat meningkatkan hasil kesehatan pasien yang lebih baik [27]. Penelitian tersebut meneliti program kesehatan untuk penyakit kronis dengan meningkatkan kontrol berdasarkan data RPM. Penggunaan RPM juga mengurangi kunjungan medis dan perawatan yang tidak diperlukan [28].

2.4 Microsoft Azure

Microsoft Azure adalah platform *cloud* computing yang dapat digunakan untuk pengembangan aplikasi, penyimpanan data, dan analisis data yang dikembangkan oleh Microsoft [18]. Azure merupakan sistem yang menyediakan *infrastructure-as-a-service* (IaaS), *platform-as-a-service* (PaaS),

dan juga *software-as-a-service* (SaaS) [29]. Microsoft Azure memberikan beberapa layanan produk yang dapat digunakan untuk sistem IoT. Dalam konteks RPM, Azure memungkinkan pengumpulan, penyimpanan, dan analisis data kesehatan melalui layanan Azure Health Data Services dan Azure Event Hubs, salah satu layanan yang tersedia pada Microsoft Azure [6]. Selain itu, Azure Storage Account (Blob Storage) juga digunakan sebagai tempat penyimpanan awal file CSV (data sintetis) untuk diproses oleh layanan lainnya pada penelitian *Proof of Concept* ini.

Sebagai platform *cloud*, Azure menyediakan integrasi terhadap berbagai perangkat IoT yang digunakan untuk RPM sesuai dengan kebutuhan. Arsitektur Azure yang bersifat *microservice* memecahkan aplikasi RPM menjadi modul-modul kecil yang dapat dikembangkan secara independen [6].

2.4.1 Azure Blob Storage

Azure Blob Storage adalah layanan penyimpanan objek yang dioptimalkan untuk menyimpan data tidak terstruktur dalam jumlah besar, seperti file teks (CSV) atau data biner. Dalam penelitian ini, Blob Storage berfungsi sebagai *Data Lake* atau titik masuk awal (*landing zone*) bagi dataset operasional rumah sakit dan log sensor IoT sebelum diproses lebih lanjut [30].

2.4.2 Azure Data Factory (ADF)

Azure Data Factory merupakan layanan integrasi data berbasis *cloud* yang berfungsi sebagai orkestrator jalur pipa data (*data pipeline*). ADF memungkinkan proses pemindahan data (*Copy Activity*) secara dinamis menggunakan fitur *ForEach* dan pemetaan skema otomatis, yang sangat krusial dalam menyatukan 10 format file heterogen yang digunakan dalam penelitian ini [30].

2.4.3 Azure SQL Database

Azure SQL Database adalah layanan basis data relasional (RDBMS) berbasis Azure SQL yang bersifat cerdas dan terkelola. Layanan ini digunakan sebagai penyimpanan permanen (*Single Source of Truth*) yang mendukung

penegakan integritas data (*Primary* dan *Foreign Key*) serta mampu menangani jutaan baris data sensor pasien secara efisien [30]

2.4.4 Microsoft PowerBI

Power BI merupakan bagian dari ekosistem analitik Microsoft yang terintegrasi erat dengan Azure SQL. PowerBI berfungsi sebagai lapisan presentasi (*Data Visualization*) untuk mengubah sekumpulan data mentah menjadi wawasan visual yang interaktif melalui pembuatan *dashboard* pemantauan pasien.

2.5 Arsitektur Modern Data Warehouse

Secara teoretis, untuk mengatasi fragmentasi informasi atau "silo data", diperlukan sebuah kerangka arsitektur integrasi yang dikenal sebagai *Modern Data Warehouse* (MDW). Berbeda dengan basis data tradisional yang kaku, MDW dirancang untuk beroperasi di lingkungan *cloud* guna menangani data terstruktur maupun tidak terstruktur berskala besar. Arsitektur ini umumnya terdiri dari empat lapisan logis:

1. Lapisan Penyerapan (*Ingestion Layer*)

Bertanggung jawab untuk menyerap data mentah dari berbagai sumber heterogen (seperti log IoT dan dokumen administratif) ke dalam sistem terpusat.

2. Lapisan Penyimpanan Mentah (*Data Lake/Storage Layer*)

Tempat penyimpanan awal data tanpa mengubah format aslinya, memungkinkan retensi data bervolume masif dengan biaya rendah.

3. Lapisan Pemrosesan (*Processing/Transformation Layer*)

Mesin komputasi yang melakukan standardisasi, pembersihan, dan harmonisasi format data.

4. Lapisan Penyajian (*Serving/Presentation Layer*)

Basis data relasional teroptimasi yang menyajikan data bersih ke perangkat intelijen bisnis (*Business Intelligence*) untuk kebutuhan analitik pengguna akhir.

Secara historis, konsep gudang data (*Data Warehouse*) pertama kali didefinisikan sebagai koleksi data yang berorientasi subjek, terintegrasi, memiliki rentang waktu (*time-variant*), dan bersifat tetap (*non-volatile*) untuk mendukung proses pengambilan keputusan manajemen. Namun, dalam perkembangannya, arsitektur gudang data tradisional yang berjalan pada infrastruktur peladen fisik lokal (*on-premise*) mulai menghadapi keterbatasan yang signifikan. Keterbatasan ini sangat terasa ketika sistem dihadapkan pada karakteristik *Big Data*, khususnya dalam menangani volume yang masif dan variasi format dari data log sensor *Internet of Things* (IoT). Oleh karena itu, *Modern Data Warehouse* (MDW) hadir sebagai evolusi arsitektur yang mengandalkan fleksibilitas dan skalabilitas komputasi awan (*cloud computing*). Arsitektur modern ini memisahkan lapisan komputasi (*compute*) dari lapisan penyimpanan (*storage*), sehingga memungkinkan organisasi untuk menyimpan data mentah yang tidak terstruktur dalam skala *petabyte* dengan biaya yang efisien sebelum data tersebut ditransformasikan.

Dalam konteks industri layanan kesehatan, pembangunan gudang data memiliki tingkat kompleksitas yang jauh lebih tinggi dibandingkan sektor industri lainnya [34]. Fasilitas kesehatan tidak hanya mengelola data operasional yang terstruktur (seperti jadwal janji temu, penagihan keuangan, dan data demografi pasien), tetapi juga harus mengonsolidasi data klinis yang sangat bervariasi, termasuk hasil rekam medis dan pencatatan tanda vital seketika dari sistem *Remote Patient Monitoring* (RPM). Ivanovski et al. (2025) menegaskan bahwa arsitektur data kesehatan yang andal harus mampu mengatasi tantangan heterogenitas format ini sekaligus menjaga interoperabilitas informasi lintas departemen [34]. *Modern Data Warehouse* menjawab tantangan ini dengan memposisikan *Data Lake* sebagai zona pendaratan awalan (*landing zone*) yang mampu menampung berbagai format fail (*file*), yang kemudian diorkestrasi melalui jalur pipa data menuju model relasional yang terstruktur.

Puncak dari implementasi *Modern Data Warehouse* pada sektor kesehatan adalah terciptanya *Single Source of Truth* (SSOT) atau sumber kebenaran tunggal bagi seluruh pemangku kepentingan. Dengan adanya kerangka SSOT, fenomena *silo data*—di mana setiap departemen medis maupun administratif memiliki dan mengelola versi datanya masing-masing—dapat dieliminasi sepenuhnya. Dalam ekosistem pemantauan pasien jarak jauh, keberadaan repositori data yang terintegrasi dan telah melalui tahap pembersihan (validasi) ini sangat krusial. Arsitektur ini memberikan jaminan bahwa visualisasi pada antarmuka *dashboard* yang digunakan oleh tenaga medis bersumber dari basis data yang akurat, mutakhir, memiliki integritas referensial yang ketat, dan terbebas dari anomali duplikasi.

2.6 Integrasi Data dan Pipeline

Proses mengalirkan data melewati lapisan-lapisan *Modern Data Warehouse* diatur oleh teori integrasi *Extract, Transform, Load* (ETL). ETL merupakan metodologi rekayasa data (*data engineering*) yang berfungsi untuk:

- *Extract*
Mengidentifikasi dan menarik data dari berbagai sistem operasional yang terisolasi.
- *Transform*
Mengaplikasikan aturan bisnis, konversi tipe data, dan resolusi anomali (seperti sinkronisasi *Primary Key*) agar data heterogen menjadi seragam.
- *Load*
Memuat data yang telah bersih ke dalam wadah penyimpanan target (*Single Source of Truth*). Otomatisasi proses ETL melalui konsep *Data Pipeline* (jalur pipa data) memastikan pembaruan informasi klinis berjalan secara konsisten tanpa intervensi manual yang rentan akan *human error*.

Proses mengalirkan data melewati lapisan-lapisan *Modern Data Warehouse* diatur oleh teori integrasi *Extract, Transform, Load* (ETL). ETL

merupakan metodologi rekayasa data (*data engineering*) yang menjadi tulang punggung dalam arsitektur intelijen bisnis. Secara konseptual, sistem ETL bertugas untuk mengambil data dari lingkungan sumber yang heterogen dan tidak terstruktur, membersihkan serta mengubah data tersebut ke dalam format yang seragam, lalu memuatnya ke dalam pangkalan data target.

Dalam konteks pengolahan data *Remote Patient Monitoring* (RPM), tahap transformasi (*Transform*) merupakan fase yang paling krusial dan memakan beban komputasi paling tinggi. Pada fase ini, fungsi-fungsi pembersihan data (*data cleansing*) diimplementasikan untuk menangani berbagai kualitas data yang buruk, seperti nilai yang hilang (*missing values*), format tanggal yang tidak konsisten, hingga redudansi atribut pengenalan pasien dari perangkat sensor. Valiki (2026) menyatakan bahwa optimalisasi jalur pipa rekayasa data sangat bergantung pada kemampuan sistem untuk melakukan transformasi secara dinamis [35]. Otomatisasi proses ETL melalui konsep *Data Pipeline* (jalur pipa data) memastikan pembaruan informasi klinis berjalan secara presisi dan konsisten, sehingga meminimalisir intervensi manual yang sangat rentan terhadap *human error* (Valiki, 2026) [35].

2.7 Microsoft Azure Well-Architected Network

Pemilihan Microsoft Azure sebagai infrastruktur *cloud* dalam penelitian ini tidak hanya didasarkan pada ketersediaan layanan, melainkan dilandaskan pada kerangka kerja arsitektur yang diakui industri, yaitu *Azure Well-Architected Framework*. Kerangka kerja ini menyediakan lima pilar panduan untuk membangun sistem berbasis *cloud* yang berkualitas tinggi:

1. *Reliability*

Kemampuan sistem untuk pulih dari kegagalan dan terus berfungsi, krusial untuk pemantauan data pasien.

2. *Security*

Perlindungan terhadap data medis sensitif melalui kontrol akses dan isolasi penyimpanan.

3. *Cost Optimization*

Mengeliminasi biaya modal pengadaan peladen fisik (*server*) dengan model *pay-as-you-go*.

4. *Operational Excellence*

Otomatisasi penerapan dan pemantauan sistem, seperti penggunaan *Azure Data Factory*.

5. *Performance Efficiency*

Kemampuan arsitektur untuk diskalakan (*scaling*) guna menangani lonjakan beban data sensor IoT tanpa penurunan kinerja.

Penerapan *Azure Well-Architected Framework* dalam perancangan infrastruktur medis di *cloud* memberikan jaminan bahwa sistem yang dibangun tidak hanya fungsional, tetapi juga berstandar level *Enterprise*. Pilar Keamanan (*Security*) menjadi perhatian paling utama dalam sistem layanan kesehatan. Data pasien yang direkam melalui sensor IoT bersifat sangat rahasia, sehingga arsitektur *cloud* harus mampu menyediakan enkripsi data saat istirahat (*data at rest*) maupun saat dalam proses transit (*data in transit*), serta mengimplementasikan kontrol akses berbasis peran (*Role-Based Access Control*).

Lebih lanjut, penggunaan ekosistem komputasi awan Microsoft Azure memberikan keunggulan kompetitif dibandingkan infrastruktur *on-premise* konvensional dalam hal Skalabilitas dan Pengoptimalan Biaya [23]. Data sensor pasien dari sistem RPM bersifat akumulatif dan akan terus membengkak (*Big Data*). Dengan model layanan *Platform-as-a-Service* (PaaS) dan penyediaan sumber daya *pay-as-you-go*, rumah sakit dapat menekan biaya operasional secara signifikan sekaligus memperoleh performa komputasi yang tinggi tanpa harus memelihara perangkat keras fisik [29]. Integrasi layanan seperti *Azure Data Factory* untuk orkestrasi dan *Azure SQL Database* untuk penyimpanan dinilai mampu memberikan visibilitas analitik yang menyeluruh, menjadikan Azure sebagai platform yang ideal untuk pengembangan arsitektur *Internet of Things* di sektor kesehatan [18].

2.8 Silo Data (*Data Silo*)

Silo data, atau sering disebut sebagai pulau informasi (*information islands*), merujuk pada sekumpulan data yang dipegang oleh satu departemen atau sistem informasi namun terisolasi dan tidak dapat diakses oleh sistem lain di dalam organisasi yang sama. Menurut Laudon [40] dalam literatur *Management Information Systems*, silo data sering kali muncul secara organik seiring dengan pertumbuhan organisasi yang mengadopsi teknologi dan perangkat lunak secara terdesentralisasi untuk memenuhi kebutuhan operasional spesifik masing-masing departemen.

Dalam konteks sistem informasi kesehatan, silo data umumnya terbentuk akibat penggunaan perangkat keras atau perangkat lunak dari vendor yang berbeda, seperti terpisahnya pangkalan data rekam medis administratif (*Electronic Health Records*) dengan log metrik vital dari perangkat *Internet of Things* (IoT). Hal ini memicu masalah interoperabilitas, di mana sistem-sistem tersebut tidak mampu berkomunikasi secara mulus. Keberadaan silo data merupakan hambatan struktural utama karena menghalangi terbentuknya visibilitas informasi yang komprehensif, memicu redundansi data, dan pada akhirnya memperlambat tenaga medis dalam melakukan diagnosis yang akurat berdasarkan rekam jejak pasien secara utuh.

2.9 Jalur Pipa Data (*Data Pipeline*)

Dalam pembahasan rekayasa data modern, jalur pipa data (*data pipeline*) didefinisikan sebagai arsitektur sistematis yang memfasilitasi pergerakan dan transformasi data dari berbagai sumber ke sistem tujuan agar dapat digunakan untuk keperluan analitik. Jalur pipa data merupakan tulang punggung dari siklus hidup rekayasa data (*data engineering lifecycle*) [41]. Jalur pipa yang dirancang dengan baik memastikan bahwa volume data mentah yang masif (seperti data transaksional dan log sensor IoT) diproses secara skalabel, aman, dan memiliki tingkat keandalan yang tinggi.

Metode standar yang digunakan dalam membangun dan mengorkestrasi jalur pipa ini adalah arsitektur *Extract, Transform, Load* (ETL), yang dalam

lingkungan komputasi awan (*cloud computing*) saat ini sangat bertumpu pada otomatisasi proses. Tahapan utamanya meliputi:

1. *Extract* (Ekstraksi)

Proses akuisisi dan pembacaan data dari sistem sumber heterogen (seperti fail CSV atau perangkat pengumpulan data medis) menuju area pendaratan awal (*Data Lake*). Pada infrastruktur awan, proses ini dilakukan secara efisien untuk menyalin data tanpa membebani dan mengganggu kinerja operasional sistem sumber.

2. *Transform* (Transformasi)

Fase pemrosesan inti di mana data mentah dibersihkan, divalidasi, dan diharmonisasi. Reis dan Housley (2022) menekankan bahwa transformasi adalah tahap paling krusial untuk mencegah entropi data (*data entropy*). Proses ini mencakup resolusi inkonsistensi tipe data, penyesuaian format waktu (*timestamp*), serta pemberlakuan aturan bisnis agar skema data struktural menjadi seragam.

3. *Load* (Pemuatan)

Proses penulisan data yang telah ditransformasi ke dalam sistem target akhir, seperti basis data relasional yang berfungsi sebagai *Single Source of Truth* (SSOT). Proses pemuatan ini memastikan bahwa data yang dikirimkan telah memiliki integritas relasional yang kuat (termasuk pencegahan *orphan data*) dan siap dikonsumsi dengan cepat oleh lapisan visualisasi analitik.