

STUDI KASUS SISTEM DYNAMIC BACKEND MIDDLEWARE UNTUK MENGAKOMODASI INTEGRASI DATA DALAM PROSES PENUTUPAN POLIS ASURANSI DI PT XYZ

CASE STUDY OF DYNAMIC BACKEND MIDDLEWARE SYSTEM TO ACCOMMODATE DATA INTEGRATION IN INSURANCE POLICY CLOSING PROCESS AT PT XYZ

Adhi Pradana¹, Jarot S. Suroso²

^{1,2}Universitas Bina Nusantara

adhi.pradana@binus.ac.id

ABSTRACT

The development of the insurance business has a positive impact on the national economy, economic actors, and public welfare. Like other financial products, insurance participation can also be terminated by closing the insurance policy. In other words, the client is no longer protected by the insurance company and cannot enjoy the benefits offered by an insurance product. The insurance policy closing system is still done manually and inefficiently. Therefore, this study was conducted to analyze and design a dynamic system backend middleware for the insurance policy closing process of PT XYZ. The system design was carried out using the Rapid Application Development (RAD) software development methodology. RAD has 3 core phases, namely Planning, Design, and Implementation. Based on the results and discussions, it can be concluded that the solution provided for companies with the resources provided and the company's objective needs is to build a system that has the aim of being a bridge between the partner system and the core system to accelerate transactions from external systems to internal systems and make transactions more precise and faster. The system that was built was named Dynamic System Middleware Backend or has begun to be commonly called MAPS (Mega API System).

Keywords: Insurance, Backend Middleware, Data Integration.

ABSTRAK

Perkembangan bisnis asuransi membawa dampak positif bagi perekonomian nasional, pelaku ekonomi, dan kesejahteraan umum. Seperti halnya produk finansial lainnya, keikutsertaan asuransi juga dapat diakhiri dengan melakukan penutupan polis asuransi. Dengan kata lain, klien tersebut tidak lagi dilindungi oleh perusahaan asuransi dan tidak dapat menikmati manfaat-manfaat yang ditawarkan oleh suatu produk asuransi. Sistem penutupan polis asuransi juga masih dilakukan secara manual dan tidak efisien. Oleh karena itu, penelitian ini dilakukan untuk melakukan analisis dan perancangan *dynamic system backend middleware* untuk proses penutupan polis asuransi PT XYZ. Perancangan sistem dilakukan menggunakan metodologi *software development Rapid Application Development* (RAD). RAD memiliki 3 fase inti yaitu *Planning*, *Design*, dan *Implementation*. Berdasarkan hasil dan pembahasan, dapat disimpulkan bahwa Solusi yang diberikan untuk perusahaan dengan sumber daya yang disediakan dan kebutuhan obyektif perusahaan adalah membangun sebuah sistem yang memiliki tujuan sebagai jembatan antara *partner system* dengan *core system* untuk mempercepat transaksi dari sistem eksternal ke sistem internal dan membuat transaksi menjadi lebih tepat dan cepat. Sistem yang dibangun diberi nama Dynamic System Middleware Backend atau sudah mulai biasa disebut MAPS (Mega API System).

Kata Kunci: Asuransi, Backend Middleware, Integrasi Data.

PENDAHULUAN

Asuransi memainkan peran penting dalam berbagai aspek masyarakat. Ini membantu dalam mendistribusikan kembali risiko dalam sistem ekonomi regional, memperkuat stabilitas rumah tangga dan bisnis, dan memastikan keamanan sosial ekonomi dalam konteks pembangunan berkelanjutan. Asuransi

juga berfungsi sebagai sarana untuk mengumpulkan tabungan dan memberikan pertanggung jawaban risiko kepada publik, menawarkan penggantian dalam situasi kerugian atau ketidakpastian (Pohrishchuk, 2023) Industri asuransi umum menghadapi tantangan dan peluang di dunia bisnis yang kompetitif saat ini. Menerapkan strategi yang efektif sangat penting untuk

keberhasilan dan pertumbuhan perusahaan asuransi (Faza Haifa & Dedy Dewanto, 2022). Asuransi adalah segala sesuatu, tindakan atau cara ganti rugi finansial untuk suatu kerugian yang diterima dan dirasakan oleh individu yang biasa disebut dengan tertanggung. Tertanggung memberikan risiko kerugian sedangkan perusahaan asuransi sebagai penanggung yang akan menanggung risiko kerugian (Mufidah, 2021).

Manajemen data adalah proses pengumpulan, penyimpanan, pengamanan, dan pemanfaatan data organisasi. Ini mencakup semua disiplin ilmu yang berkaitan dengan pengelolaan data sebagai sumber daya yang berharga (Desy, 2022). Manajemen data sangat penting karena data dianggap sebagai sumber daya berharga dalam organisasi modern. Organisasi berinvestasi besar-besaran dalam penyimpanan data dan infrastruktur manajemen ketika mereka memiliki akses ke berbagai jenis data dalam jumlah besar (Maritz, 2003). Integrasi Data adalah proses kombinasi data dari dua sumber atau lebih menjadi satu. Integrasi Data menyatukan komponen yang sesuai dengan suatu sistem sehingga dapat berfungsi dan digunakan pada sistem yang menjadi tujuan. Integrasi data membantu konsolidasi secara signifikan semua tipe data, mempertimbangkan pertumbuhan, volume dan semua format yang bervariasi. Tipe umum integrasi data adalah penyerapan data, maksudnya adalah data dari satu sistem diintegrasikan secara berkala ke sistem lain. Tipe integrasi data ini disebut juga *Extract, Transform, Load* (ETL) (Barbella & Tortora, 2023).

Penutupan Polis Asuransi merupakan salah satu permasalahan yang sering terjadi dalam perusahaan asuransi. Seiring berkembangnya bisnis retail dan mulai banyaknya transaksi serta integrasi dan kolaborasi antar lini bisnis, Saat ini integrasi bisnis masih dilakukan secara manual untuk proses penutupan asuransi, *client* atau *customer* mengirimkan data secara manual dalam *file microsoft excel*

melalui *email* atau SFTP lalu divalidasi dan diproses secara manual oleh bagian *Policy Service*, bagian *Policy Service* dibagi menjadi dua yaitu, *team input* manual satuan dan *team upload* polis *batching* atau *bulking*. Pengajuan penutupan polis asuransi dari *partner* segmen broker ada perusahaan A sepanjang tahun 2022 dengan jumlah 1842 hanya kendaraan bermotor. Sedangkan pada perusahaan B Tahun 2022 dengan jumlah 3140 hanya kendaraan bermotor, dan Perusahaan C dengan jumlah 5219 hanya kendaraan bermotor. Data diambil dari *database* PT XYZ.

Proses manual membutuhkan waktu yang lama dengan SLA H+3 dan tidak dapat diproses secara langsung, sedangkan harapan *client* atau *customer* untuk proses B2B adalah setelah *input* data maka *client* atau *customer* mendapatkan dokumen polis asuransi secara langsung. Jadi masalah yang dihadapi oleh perusahaan adalah proses penutupan polis asuransi dengan skema B2B yang masih dilakukan manual sehingga transaksi tidak dapat dilakukan secara *realtime*. Berdasarkan masalah tersebut ada solusi yang dirasa sangat sesuai dengan kebutuhan untuk mengatasi masalah yang ada, yaitu membuat *middleware* dengan basis *Application Programming Interface* (API).

Sistem *middleware backend* adalah perangkat lunak yang bertindak sebagai jembatan antara aplikasi atau komponen yang berbeda, menyediakan kemampuan komunikasi dan manajemen data (Farizy & Eriana, 2022). Hal ini memungkinkan aplikasi untuk berinteraksi satu sama lain dan dengan *database* atau alat yang awalnya tidak dirancang untuk terhubung dengannya. Sistem *middleware backend* mengidentifikasi aplikasi *middleware* dan menghasilkan artefak penemuan untuk menggambarkan aplikasi ini dan data *input/outputnya*. Klien penemuan kemudian menggunakan informasi ini untuk membuat model REST untuk aplikasi *middleware*, artefak *mapper* untuk memetakan operasi REST, dan artefak

transformator untuk mengonversi format data.

Seiring berkembangnya bisnis retail dan mulai banyaknya transaksi serta integrasi dan kolaborasi antar lini bisnis, PT XYZ dituntut untuk dapat melakukan integrasi Business to Business (B2B). Saat ini integrasi bisnis masih dilakukan secara manual untuk proses penutupan asuransi, klien atau customer mengirimkan data secara manual dalam file microsoft excel melalui email atau SFTP lalu divalidasi dan diproses secara manual oleh bagian *Policy Service*, bagian *Policy Service* dibagi menjadi dua yaitu, tim input manual satuan dan *tim upload polis batching* atau *bulking (Discovery and exposure of transactional middleware server-based applications as consumable service endpoints)*. *Middleware* umumnya berupa *Application Programming Interface (API)* yang dipasang pada sisi klien yang digunakan untuk memanggil sebuah *services*, dan mencakup transmisi permintaan layanan melalui jaringan dan respon yang dihasilkan.

Berdasarkan rumusan masalah yang telah dijelaskan, penelitian ini bertujuan untuk Melakukan analisis dan perancangan *backend system middleware* yang dinamis untuk proses penutupan polis asuransi PT XYZ.

METODE

Penelitian ini dilakukan pada PT. XYZ fokus pada pelayanan langsung kepada nasabah ataupun juga melalui pihak-pihak terkait lainnya. Saat ini terdapat berbagai jenis produk asuransi terbaik yang ditawarkan oleh PT. XYZ. Produk asuransi perjalanan terbaik melalui Mega Travel Care, produk asuransi kesehatan terbaik melalui Mega Medical Care, Mega Proteksi Diri, dan Mega Medical Plus, produk asuransi rumah tinggal melalui Mega Rumah, dan juga produk asuransi kendaraan terbaik melalui Mega Kendaraan.

Perancangan sistem dilakukan menggunakan metodologi *software*

development Rapid Application Development (RAD). RAD memiliki 3 fase inti yaitu, *Planning, Design, Implementation*. Pada fase *Planning* dilakukan untuk melakukan identifikasi kebutuhan berdasarkan masalah yang ada. Setelah melakukan identifikasi kebutuhan dapat dilakukan pengumpulan sumber daya manusia yang dibutuhkan atau membangun tim untuk melakukan *development*. Pada penelitian ini dibutuhkan 1 *analyst*, 1 *developer*, 1 *tester*, dan 1 IT Infrastruktur. Selanjutnya dilakukan persiapan terhadap teknologi yang dibutuhkan. Sampai saat ini teknologi yang dibutuhkan adalah *software development, database, dan cloud environment*. Setelah semua sumberdaya sudah didapatkan, maka *timeline* proyek dapat dibuat. Biaya tambahan yang dibutuhkan belum ada sampai saat ini, biaya *cloud* masuk ke biaya *operational expenditure (Opex)*

Fase selanjutnya yaitu *Design*, pada fase ini dilakukan pembuatan *user requirement* berdasarkan kebutuhan yang sudah diketahui pada fase *planning*. Setelah semua detail kebutuhan sudah diketahui dapat dilakukan perancangan sistem menggunakan UML beserta tampilan *middleware*-nya. Diagram-diagram yang akan dibuat ada *Use Case Diagram* untuk menggambarkan poin-poin dari kebutuhan yang ada beserta aktor-aktor pada proses. Diagram selanjutnya adalah *Activity Diagram* untuk menggambarkan proses kerja *middleware*. Diagram selanjutnya adalah *Sequence Diagram* untuk menggambarkan hubungan antara aktor dan sistem serta menyajikan *flow input* dan *output*. Diagram selanjutnya adalah *Communication Diagram*, diagram ini menggambarkan proses kerja *middleware* beserta aktor dan fungsi yang dapat dilakukan oleh aktor didalamnya. Yang terakhir adalah *Class Diagram, Class Diagram* menggambarkan relasi data berupa tabel-tabel yang ada pada *middleware*.

Fase terakhir adalah fase *Implementation*, pada fase ini dilakukan proses *development*. Setelah proses *development* selesai, dilakukan pengujian terhadap *middleware*, tahap pengujian ada 2, pertama pengujian yang dilakukan oleh tim *testing IT*, yang kedua pengujian dilakukan oleh *user*. Setelah semua pengujian selesai, *middleware* dapat *live* dan *publish*.

Dynamic System Backend Middleware yang akan dibangun berfungsi untuk menerjemahkan atau mengintegrasikan data yang dikirim oleh *partner* menjadi data yang dapat masuk pada *core system* PT XYZ. Cara kerjanya adalah memetakan atau *mapping* antara data *partner* dengan wadah penyimpanan pada PT XYZ, yang dimaksud wadah adalah *core system*. Pengguna yang menggunakan sistem ini hanya memiliki satu *role*, yaitu *administrator*. Fungsi yang diharapkan ada pada sistem ini adalah fungsi *create*, *read*, *update*, *delete master data mapping*, lalu fungsi *mapping* wadah atau *column*, dan terakhir fungsi *setting rule policy* dokumen asuransi yang dibutuhkan *partner* dan akan menjadi *feedback*

HASIL DAN PEMBAHASAN

Perancangan Sistem

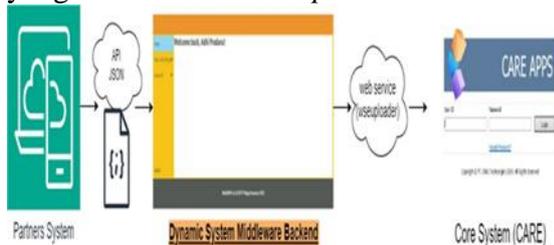
Dynamic System Backend Middleware adalah sistem yang menerjemahkan dan mengintegrasikan data dengan memetakannya ke dalam *core system*. Terdapat dua *role* pengguna yang dapat mengakses sistem ini, diantaranya adalah *user* dan *partner*. *User* berfungsi sebagai *Map Code Setting* yang mana bertugas memetakan data dengan bantuan kolom-kolom yang berasal dari *core system* yang disebut *eUploader*. Selain sebagai *Map Code Setting*, *user* juga berfungsi sebagai *Auto Print* yaitu mendaftarkan semua *class* dokumen yang ada pada *core system* pada menu *AutoPrint Master*. Terdapat 5 diagram yang dihasilkan dari penggunaan UML (Unified Modelling Language) pada

penelitian ini, yaitu *activity diagram*, *use case diagram*, *sequence diagram*, *communication diagram*, dan *class diagram*.

Activity diagram menggambarkan aliran aktifitas dalam sistem yang dirancang sebagaimana setiap aliran dimulai dan diakhiri. Selanjutnya, *use case diagram* menggambarkan fungsionalitas yang diharapkan dari sistem dan merupakan suatu gambaran grafis dari semua aktor serta relasinya pada sebuah sistem dengan berbagai kegiatan. *Use case diagram* tidak menjelaskan secara rinci mengenai penggunaan *use case*, namun hanya memberikan gambaran singkat mengenai hubungan antara aktor dan *user case*. Pada *class diagram* digambarkan kelas-kelas dan atribut-atribut dari seluruh objek pada suatu sistem yang saling berhubungan satu sama lain. Objek-objek tersebut dapat ditemukan dengan cara mengidentifikasi kata benda pada *activity diagram* yang kemudian menambahkan atribut dari kelas dan relasi antar kelas. *Sequence diagram* digunakan untuk menggambarkan interaksi aktor objek di dalam dan di sekitar sistem, berupa pesan-pesan yang digambarkan terhadap waktu. Dalam diagram ini disajikan *flow* informasi *input* dan *output* dari sebuah sistem. Alur dari *sequence diagram* dibuat berdasarkan *flow* dari *activity diagram* yang diusulkan. Selanjutnya adalah *communication diagram*, secara eksplisit menunjukkan hubungan antara jalur kehidupan yang berpartisipasi dalam kolaborasi. Pada *communication diagram*, interaksi digambarkan pada apa yang dasarnya merupakan fragmen dari objek.

Pada rancangan desain *user interface dynamic system middleware backend* memiliki dua menu utama yaitu *Map Code Setting* dan *Autoprint*, dengan enam sub menu yaitu *Master*, *Mapping Column*, *Mapping Source Column*, *Feedback Query*, *Auto Print Master*, dan *Auto Print Rule*. Seluruh menu hanya dapat diakses oleh *role user*. Selanjutnya, desain arsitektur pada *dynamic system*

middleware backend berelasi dengan dua sistem lain, yaitu *partners system* dan *core system* perusahaan sendiri yang bernama *Care*. Relasi dan integrasi pertukaran data dari *partner system* ke *dynamic system middleware backend* menggunakan metode *API* dengan format penyimpanan dan pengiriman data berupa *JSON*. Setelah data dari *partner system* disimpan ke dalam *database dynamic system middleware backend*, selanjutnya diteruskan ke *Care* menggunakan *web service* sebagai perantaranya, *web service* yang ada bernama *wseuploader*.



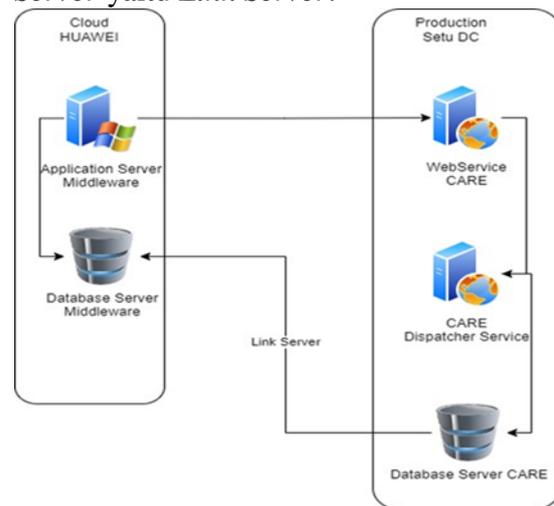
Gambar 1. System Architecture

Dalam proses transaksi, ada momen dimana *dynamic system middleware backend* harus mengambil data dari *core system* tetapi tidak ingin menyimpan data tersebut ke dalam *database dynamic system middleware backend* karena database yang digunakan adalah *SQL Server*. Oleh karena itu, terdapat teknologi yang disediakan yaitu *Link Server*. *Link Server* itulah yang dapat berfungsi sebagai penghubung untuk *query data* antara *dynamic system middleware backend* dengan *core system*.

Desain arsitektur infrastruktur *dynamic system middleware backend* dibagi menjadi dua *environments*, yaitu *Cloud Huawei* dan *Data Centre Production Setu*. *Dynamic system middleware backend* diletakkan pada *Cloud Huawei* bersama dengan *Database Server* sistem tersebut, tujuan dari diletakkan di *Cloud* agar lebih fleksibel dalam berinteraksi dengan sistem lain dan proses *maintenance* lebih mudah.

Seperti yang dijelaskan pada *system architecture*, komunikasi antara *Database Server* pada *Data Centre Production* dengan *Database Server* pada *Cloud*

Huawei menggunakan fitur dari *SQL Server* yaitu *Link Server*.



Gambar 2. Infrastructure Architecture

Proses *Testing* merupakan proses terakhir dari fase implementasi pada metodologi RAD. Pada proses *testing* menghasilkan dua dokumentasi yaitu skenario tes dan dokumen API.

Pembahasan

Dynamic System Middleware Backend meningkatkan efisiensi dalam proses penutupan polis asuransi dengan mengimplementasikan AOP dinamis melalui penggantian kode metode pada saat *runtime*. Pendekatan ini menggunakan titik penggabungan untuk memicu kompilasi ulang kode asli, menghasilkan saran yang sudah dijalin ke dalamnya atau panggilan balik ke saran yang sesuai, sehingga sangat meningkatkan efisiensi proses AOP dinamis. Selain itu, sistem pemrosesan polis asuransi dapat meningkatkan efisiensi sistem manajemen polis asuransi dengan menerima pesan permintaan asuransi dari klien, menyimpan informasi polis asuransi, dan mengirimkan informasi tersebut ke sistem manajemen polis asuransi untuk diproses secara berkala. Sistem ini membuat penerimaan polis asuransi lebih aman, andal, dan efisien, yang pada akhirnya meningkatkan efisiensi proses penjaminan polis asuransi secara keseluruhan.

Dynamic System Middleware Backend membantu dalam meningkatkan akurasi data dengan menyediakan

kerangka kerja keputusan yang secara adaptif memisahkan objek data berdasarkan pola kueri dan pembaruannya. Kerangka kerja ini mengurangi biaya jaringan dengan menyimpan objek data yang sering ditanyakan di *cache* dan objek data yang sering diperbarui di repositori. Selain itu, sistem ini menggunakan metode statistik untuk memastikan bahwa perbaikan yang dilakukan pada data akurat di atas tingkat yang telah ditentukan tanpa interaksi pengguna yang berlebihan. Dengan menggunakan pendekatan ini, sistem dapat menjaga konsistensi dan akurasi dalam data, meminimalkan pelanggaran batasan integritas dan kesalahan dalam *database*. Sistem ini juga mengumpulkan, mengatur, dan mengkorelasikan data sensus pasien dan data transaksi untuk meningkatkan akurasi transaksi farmasi. Secara keseluruhan, *Dynamic System Middleware Backend* menyediakan metode dan algoritma untuk meningkatkan konsistensi dan akurasi data di berbagai domain, seperti repositori ilmiah dan transaksi farmasi.

Dynamic Systems Middleware Backend dapat ditingkatkan untuk menangani lebih banyak data atau jenis asuransi lainnya dengan memanfaatkan *template* yang dapat dikonfigurasi dan diperluas untuk produk asuransi, yang memungkinkan beragam konfigurasi dan perluasan modul, level, dan hubungan berdasarkan kebutuhan yang berbeda. Pendekatan dinamis ini memungkinkan teknisi untuk merancang produk asuransi yang memenuhi persyaratan tertentu dengan cepat dan nyaman tanpa perlu modifikasi atau perubahan kode. Selain itu, penggunaan algoritma penghitungan risiko berdasarkan berbagai variabel, seperti variabel S.M.A.R.T, variabel O.S., variabel Anti-virus, variabel Cadangan, variabel S.N.M.P, variabel Perangkat Keras, dan variabel perilaku pelanggan, memungkinkan terciptanya peraturan baru untuk penghitungan risiko dan pendeteksian otomatis terhadap tren-tren baru. Sistem evaluasi asuransi yang

dinamis juga berkontribusi pada skalabilitas dengan secara otomatis menentukan harga akhir asuransi berdasarkan data yang dikumpulkan dan diproses secara dinamis pada klien dan subjek asuransi.

Dynamic Systems Middleware Backend dapat dikaitkan dengan biaya implementasi dan kompleksitas teknis. Menerapkan dan memelihara *middleware* semacam itu dapat menjadi tantangan karena teknologi yang berbeda yang terlibat, seperti akses ke periferal, sistem operasi *real-time*, dan komunikasi. Selain itu, kompleksitas sistem terdistribusi dan heterogen dapat menyulitkan dalam mendesain, mengimplementasikan, menguji coba, dan memelihara *middleware*. Tantangan-tantangan ini dapat menyebabkan kebingungan mengenai arsitektur dan perlunya penulisan ulang sistem ketika bertransisi ke teknologi baru. Namun, penggunaan teknik pengembangan perangkat lunak berbasis komponen, seperti yang digunakan dalam *middleware* OrocOS, dapat memberikan keuntungan seperti penggunaan ulang kode, eksekusi terdistribusi, dan konfigurasi *runtime*. Selain itu, integrasi *middleware* standar industri dengan arsitektur terbuka memungkinkan kombinasi mekanisme dan kebijakan kualitas layanan yang berbeda untuk mengelola kinerja sistem.

Rekomendasi untuk studi lanjutan mengenai analisis dan desain *dynamic system backend middleware* untuk proses penutupan polis asuransi adalah pengembangan *template* data dasar yang dapat dikonfigurasi dan diperluas untuk produk asuransi, yang memungkinkan konfigurasi dan ekstensi yang beragam untuk memenuhi kebutuhan yang berbeda. Rekomendasi lainnya adalah mengeksplorasi penggunaan penalaran semantik, pesan agen, dan perencanaan dalam pengembangan platform *middleware* untuk sistem TIK industri. Selain itu, mungkin akan bermanfaat untuk menyelidiki faktor-faktor yang

berkontribusi terhadap terbatasnya penggunaan temuan evaluasi dalam desain kebijakan penelitian, pengembangan teknologi, dan inovasi (RTDI), seperti keterlibatan pemangku kepentingan, metodologi, dan perumusan rekomendasi. Terakhir, pendekatan studi kasus dapat digunakan untuk menentukan efek dari tuntutan tugas, kebutuhan pelanggan, dan lingkungan organisasi terhadap rekomendasi desain ulang ergonomis dalam proses penutupan polis asuransi.

SIMPULAN

Penelitian ini telah melakukan analisis terhadap proses bisnis yang menjadi masalah pada organisasi atau perusahaan saat ini, yaitu menunjang bisnis ritel dalam bidang usaha asuransi umum berdasarkan *literature review* yang didapatkan. Dari kumpulan *literature review* yang didapatkan telah dipertimbangkan dan didiskusikan oleh bagian yang akan terkait dengan proyek ini, dan disesuaikan dengan sumber daya yang disediakan oleh perusahaan dan kebutuhan obyektif perusahaan.

Solusi yang diberikan untuk perusahaan dengan sumber daya yang disediakan dan kebutuhan obyektif perusahaan adalah membangun sebuah sistem yang memiliki tujuan sebagai jembatan antara *partner system* dengan *core system* untuk mempercepat transaksi dari sistem eksternal ke sistem internal dan membuat transaksi menjadi lebih tepat dan cepat. Sistem yang dibangun diberi nama *Dynamic System Middleware Backend* atau sudah mulai biasa disebut *MAPS (Mega API System)*. Dari awal bulan Maret, saat sistem ini mulai melakukan transaksi sampai saat ini bulan Oktober, tercatat sistem ini sudah menerima dan melakukan 49.531 transaksi berhasil dengan total 22 *partners*.

DAFTAR PUSTAKA

Barbella, M., & Tortora, G. (2023). A semi-automatic data integration process of heterogeneous databases.

Pattern Recognition Letters, 166, 134–142.

<https://doi.org/10.1016/j.patrec.2023.01.007>

Desy. (2022). *Manajemen Data : Pengertian, Fungsi, Studi Kasus*. Mekari Jurnal.

Farizy, S., & Eriana, E. S. (2022). *Cloud Computing = Komputasi Awan*. Unpam Press.

Faza Haifa, & Dedy Dewanto. (2022). Opportunities & threats in general insurance industry, a case in Indonesia. *Journal of Management and Energy Business*, 2(1), 1–11. <https://doi.org/10.54595/jmeh.v2i1.27>

Maritz, S. G. (2003). Data management: Managing data as an organisational resource. *Acta Commercii*, 3(1). <https://doi.org/10.4102/ac.v3i1.44>

Mufidah, Z. U. (2021). Analisis pengaruh pendapatan premi neto terhadap laba perusahaan sektor asuransi jiwa di indonesia selama pandemi. *Jurnal Keuangan Negara Dan Kebijakan Publik*, 2, 184–195.

Pohrishchuk, B. (2023). The role of insurance in ensuring social and economic security of regions in the context of sustainable development. *Modern Economics*, 37(1), 78–83. [https://doi.org/10.31521/modecon.V37\(2023\)-11](https://doi.org/10.31521/modecon.V37(2023)-11)