BAB 3
PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Pelaksanaan kerja magang dilakukan pada posisi *Technical Consultant* di bawah pimpinan *VP Operation*. Tim tempat bertugas terdiri atas *Project Manager* dan *Team Lead* yang mengarahkan pelaksanaan tugas harian. Posisi ini berada pada lini teknis yang berhubungan langsung dengan pelaksanaan pekerjaan, sedangkan pengambilan keputusan strategis berada pada jalur pimpinan perusahaan.

Peran *Technical Consultant* berjalan berpasangan dengan *Team Lead* dalam kegiatan sehari-hari. *Team Lead* mengatur arah teknis, menetapkan prioritas, dan melakukan pengawasan, sedangkan *Technical Consultant* mengeksekusi pekerjaan utama di lapangan serta mengambil alih koordinasi ketika *Team Lead* berhalangan. Fokus pekerjaan meliputi pemeliharaan sistem ISIM yang telah berjalan, eksplorasi ISVG, penerapan ISVG, serta migrasi dari ISIM ke ISVG. Hambatan ditangani terlebih dahulu pada tingkat *Technical Consultant*, kemudian dieskalasi kepada *Team Lead*, dan diteruskan kepada *Project Manager* apabila diperlukan.

Koordinasi kerja harian dilakukan menggunakan grup WhatsApp internal untuk sinkronisasi tim serta grup *WhatsApp* eksternal untuk komunikasi dengan pihak klien. Surat elektronik (*email*) digunakan untuk korespondensi formal, penjadwalan rapat, konfirmasi keputusan, serta pemberitahuan tugas atau isu baru dari *Project Manager*. Setiap langkah perbaikan dan catatan penyelesaian isu didokumentasikan di Google Drive agar penelusuran perubahan dan alih pengetahuan dapat berjalan secara tertib.

Seluruh penugasan dikelola melalui aplikasi KejarTugas dengan dua jenis tugas, yaitu tugas proyek dan tugas *ad hoc*. Setiap kartu tugas memuat uraian pekerjaan, target, serta catatan kemajuan yang diperbarui secara harian melalui fitur catatan pada akhir jam kerja. Apabila suatu kegiatan pemeliharaan atau perubahan berpotensi memengaruhi operasional Jasindo, koordinasi dilakukan melalui *Project Manager* dengan pihak klien. Proses ini mencakup penetapan ruang lingkup pekerjaan, analisis dampak dan risiko, penyusunan rencana mitigasi dan pemulihan, penentuan jadwal pelaksanaan, pemberitahuan kepada pihak terkait, pemberian persetujuan sebelum eksekusi, pemantauan hasil, serta pencatatan perubahan guna memastikan operasional tetap berjalan dengan baik.

3.2 Tugas yang Dilakukan

Tugas magang berfokus pada uji konsep modernisasi tata kelola identitas untuk klien PT Asuransi Jasindo. Pekerjaan dipecah menjadi empat tahap seperti berikut.

1. Eksplorasi
   Mengumpulkan kebutuhan klien, meninjau kondisi sistem identitas yang berjalan, menetapkan target sistem yang akan diintegrasikan, menyusun rencana alur kerja, dan menyiapkan lingkungan uji.

2. Implementasi
   Mewujudkan alur yang telah direncanakan, melakukan konfigurasi komponen, menyusun skema data dan formulir, mengembangkan konektor ke sistem target, serta menuliskan panduan eksekusi kerja.

3. Pengujian
   Memverifikasi fungsi dan hasil, menguji alur identitas dan kasus batas, memastikan konektivitas ke sistem target, dan mencatat temuan untuk perbaikan.

4. Dokumentasi dan Serah Pengetahuan
   Menyusun dokumentasi cara kerja, bukti uji, dan catatan perubahan, serta melakukan serah pengetahuan kepada pihak terkait agar operasional berjalan konsisten.

3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang Track 2 memiliki proyek yang berbeda. Pada Track 1, kegiatan berupa pengembangan aplikasi internal dengan kontribusi pada frontend, *backend* chatbot AI, serta pengembangan OCR untuk dokumen finansial. Pada Track 2, kegiatan difokuskan pada *proof of concept* (PoC) migrasi platform *Identity and Access Management* dari IBM Security Identity Manager (ISIM) menuju IBM Security Verify Governance (ISVG) untuk kebutuhan PT Asuransi JASINDO, yang mencakup pemetaan komponen, rencana migrasi data dan konfigurasi, serta persiapan skenario *cutover* dan *rollback*. Pelaksanaan kerja magang Track 2 di PT Global Innovation Technology (GIT) untuk setiap minggu diuraikan seperti pada Tabel 3.1.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu ke-	**Pekerjaan yang dilakukan**
1	Penelusuran masalah JWT pada Oracle IDM serta uji peningkatan versi Oracle Identity Management dari 12c ke 14c. Pemeliharaan sistem manajemen identitas Bank Mandiri dilakukan dan dokumentasi peningkatan versi disusun.
2	Penyusunan dan finalisasi draf presentasi uji konsep peningkatan ISIM Jasindo ke ISVG berdasarkan dokumentasi resmi ISVG dan ISVA.
3	Penyusunan panduan instalasi ISVG, penelusuran dokumentasi IBM Db2, serta instalasi Db2 dan ISVG pada server uji Jasindo.
4	*Deployment* awal ISVG dan WebSphere Application Server serta upaya koneksi dengan Db2. Dokumentasi instalasi diperbarui akibat kendala pada komponen SVDI.
5	Instalasi SVDI dan upaya koneksi Db2 melalui ISVD. Penelusuran paket ISVD yang sesuai dilakukan, serta pemeliharaan sistem manajemen identitas Bank Mandiri.
6	Pembuatan *backup* server PoC Jasindo serta penyiapan prasyarat integrasi ISVG, WebSphere, Db2, dan komponen pendukung integrasi, termasuk verifikasi konektivitas antar komponen.
7	Validasi dokumen instalasi di server kantor, penelusuran error Java dan *deployment* ITIM, serta pelaksanaan rapat pembukaan POC ISVG.
8	Penelusuran lanjutan error *deployment* ITIM, pemeliharaan ISIM–ISAM Jasindo, dan verifikasi konfigurasi jaringan akibat ketidaksesuaian IP.
9	Penyiapan dan evaluasi penerapan fix pack Db2 serta penyelarasan paket ISVG terkait kendala *deployment* ITIM. Perbaikan komponen pendukung dilakukan dan laporan eskalasi ke IBM disusun.
	Lanjut pada halaman berikutnya

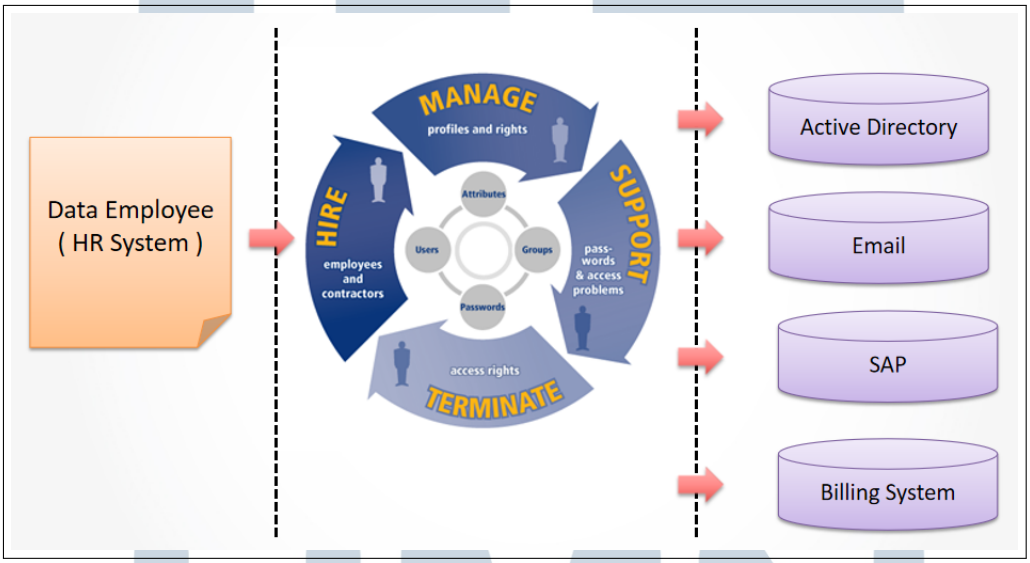
Tabel 3.1 Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang (Lanjutan)

Minggu ke-	Pekerjaan yang dilakukan
10	Implementasi bertahap solusi dari IBM Support dan penelusuran lanjutan terhadap permasalahan *deployment* ITIM.
11	Koordinasi dengan IBM Support dan Jasindo terkait lisensi, penyusunan dokumen komparasi lisensi, serta instalasi Db2 pada server Ubuntu.
12	Rapat koordinasi terkait kesalahan lisensi dan penanganan gangguan ISIM–ISAM Jasindo. Eksplorasi kompatibilitas installer dan pendalaman materi IBM Tivoli dilakukan.
13	Penanganan gangguan koneksi WebSphere–Db2 yang menyebabkan RTO, pembaruan laporan pemeliharaan, dan rapat koordinasi dengan pihak terkait.
14	Pemantauan pasca RTO, pemeriksaan log server, serta rapat dengan principal IBM terkait ketidaksesuaian lisensi peningkatan sistem.
15	Penyusunan dokumen langkah kerja ISVG, pemantauan berkala server Jasindo, dan penghentian layanan server POC sesuai jadwal.
16	Membantu Tim Indosat pada integrasi OIAM dengan MySQL dan Active Directory, serta percobaan rekonsiliasi trusted dan non-trusted.
17	Instalasi Microsoft Active Directory pada Oracle Cloud, restorasi layanan OIAM, dan integrasi Active Directory dengan OIAM.

3.3.1 Gambaran Umum IDM IBM

Identity Management (IDM) adalah cara terpusat untuk mengelola akun dan hak akses pengguna di berbagai aplikasi dan sistem. Sistem ini membantu organisasi dalam pembuatan akun, pemberian hak akses, menyinkronkan perubahan, dan menonaktifkan akun saat tidak lagi dipakai [1]. IBM menyediakan

produk bernama IBM Verify. Produk tersebut menyediakan layanan untuk tata kelola akses dan administrasi identitas, pengelolaan akses untuk login dan *single sign-on*, serta layanan direktori untuk menyimpan data identitas secara andal [21]. Lapisan pengelolaan akses mendukung autentikasi modern dan kebijakan akses terpusat agar penggunaan aplikasi tetap aman [22]. Layanan direktori menyediakan tempat penyimpanan identitas yang skalabel untuk kebutuhan otentikasi dan pencarian atribut [23]. Integrasi ke berbagai sistem tujuan dilakukan melalui *adapter* resmi sehingga proses pemberian akses dan penyelarasan data dapat dikelola dari satu tempat [24].



Gambar 3.1. Ilustrasi alur dasar IDM dari sumber data pegawai ke berbagai *target system* [25]

Gambar 3.1 menunjukkan alur dasar IDM pada umumnya. Data pegawai dari sistem HR menjadi sumber data utama yang memicu proses masuk, perubahan peran, dan keluar [3]. Objek identitas seperti atribut, pengguna, grup, dan kata sandi dikelola dengan kebijakan yang jelas, lalu perubahan tersebut dikirim dan disinkronkan ke berbagai *target system* sesuai kebutuhan [26]. Ketika masa kerja berakhir, sistem mencabut akses secara otomatis agar kondisi tetap aman dan konsisten [26].

A Terminologi Dasar IDM

Untuk menyamakan pemahaman serta memudahkan penelusuran pada bagian-bagian berikutnya, definisi istilah kunci yang digunakan dalam laporan ini

disajikan sebagai berikut:

1. *Target system* adalah sistem tujuan yang menerima atau memakai data akses dari platform IDM, misalnya direktori, email, ERP, atau aplikasi bisnis.
2. *Source of Truth* adalah sumber data utama identitas seperti sistem HR yang memicu pembuatan akun dan menjadi acuan perubahan data.
3. *Flow* adalah rangkaian langkah dari data masuk sampai hak akses diterapkan sesuai kebijakan sehingga proses berjalan teratur dan dapat ditelusuri.
4. *Provisioning* adalah proses memberi atau memperbarui akses ke sistem tujuan sesuai peran dan kebutuhan kerja pengguna.
5. *Deprovisioning* adalah proses mencabut akses ketika status pengguna berubah atau berakhir agar tidak ada hak yang tertinggal.
6. *Adapter* adalah komponen penghubung antara platform IDM dan sistem tujuan sehingga data dan perintah dapat dikirim dengan benar.
7. *Role Based Access Control* (RBAC) adalah model pemberian akses berdasarkan peran di organisasi sehingga pengguna mendapat hak yang sesuai tugasnya.
8. *Reconciliation* adalah penyelarasan data akun di sistem tujuan agar cocok dengan data pada platform IDM dan tetap mutakhir.
9. *Lookup* adalah kolom pilihan yang berisi nilai yang dapat dipilih pengguna untuk memudahkan pengisian data.
10. *Prepopulate* adalah pengisian otomatis kolom berdasarkan data yang sudah tersedia di sistem agar input lebih cepat dan konsisten.

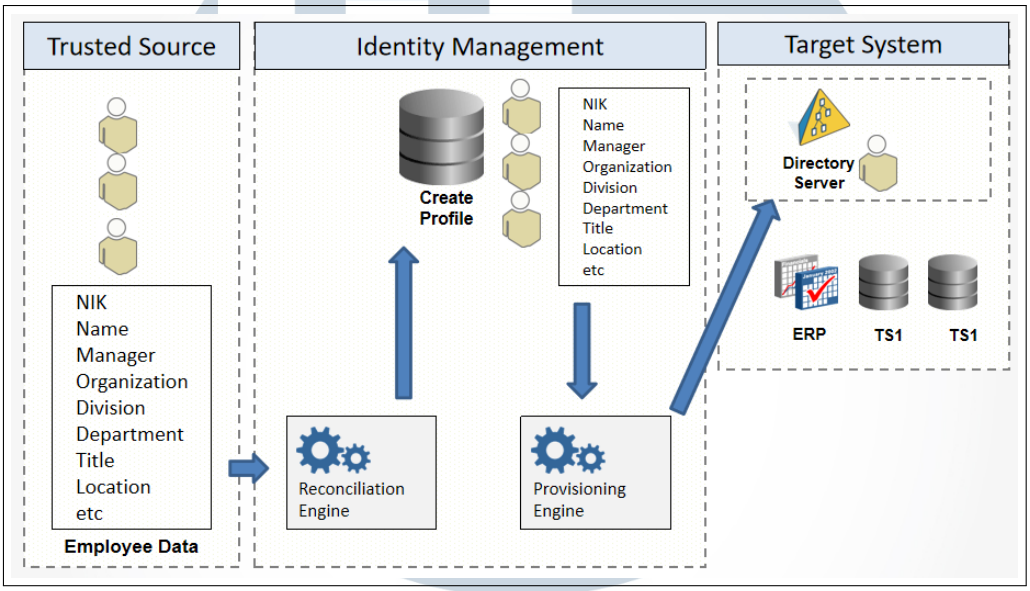
B Arsitektur dan Alur Kerja

Dalam praktik, terdapat empat alur umum, yaitu *onboarding*, *transfer*, *termination*, dan *identification* akun yang sudah ada.

B.1. Onboarding

Pada alur ini, data pegawai baru dari sumber berotoritas dimuat ke platform IDM. Sistem membuat profil pusat yang berisi data kunci, seperti

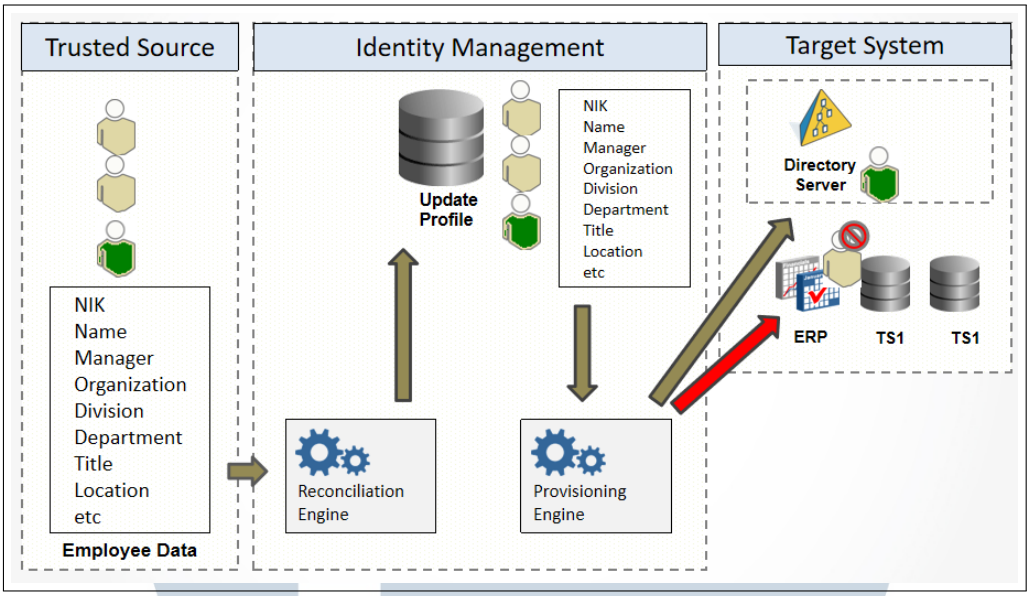
nama, unit, jabatan, atasan, dan lokasi. Setelah itu, akses dasar ke sistem tujuan diberikan sesuai kebijakan yang aktif. Jika terdapat perubahan data, seperti masa percobaan selesai atau peran tambahan, akses disesuaikan agar tetap sesuai kebutuhan kerja. Contoh alur *onboarding* ditunjukkan pada Gambar 3.2.



Gambar 3.2. Contoh alur *onboarding* pengguna pada ekosistem IDM. [25]

B.2. Transfer

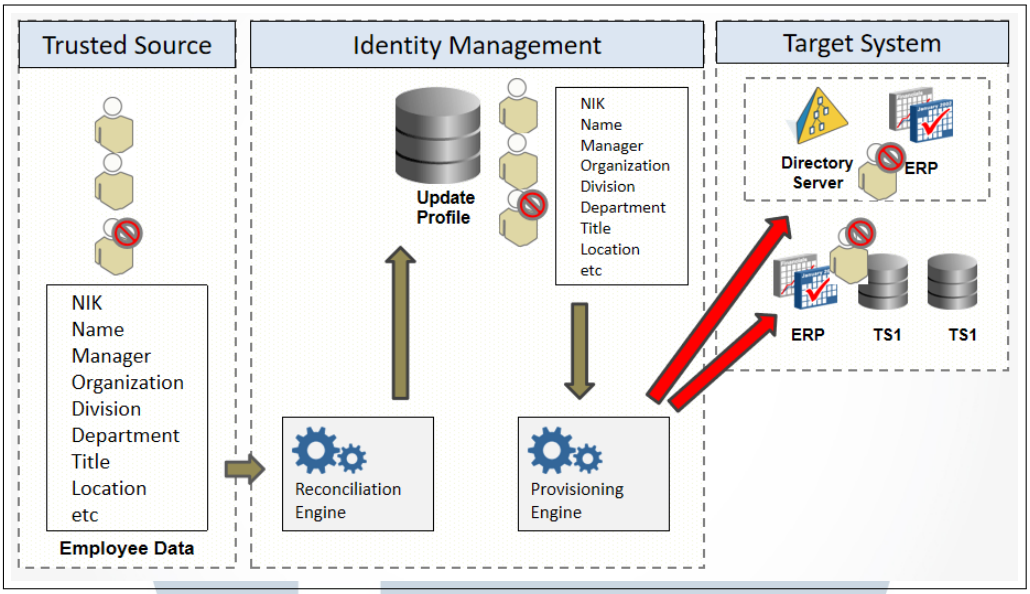
Setelah *onboarding*, perubahan organisasi, jabatan, atau lokasi akan memperbarui profil identitas di pusat. Sistem menata ulang akses agar tetap selaras, dengan cara mempertahankan akses dasar, mencabut akses yang tidak lagi relevan, dan menambah akses yang sesuai peran baru. Pembaruan ini didorong ke direktori, ERP, email, dan aplikasi lain sehingga keanggotaan grup dan kewenangan di setiap sistem tetap konsisten.



Gambar 3.3. Contoh alur *transfer* saat peran atau unit pengguna berubah. [25]

B.3. Termination

Ketika status berhenti diterima dari sumber berotoritas, profil pusat diperbarui untuk mencerminkan kondisi terakhir. Sesuai kebijakan, sistem mencabut seluruh akses pada semua sistem tujuan agar keamanan terjaga dan tidak ada akun yang tertinggal. Bila diperlukan, akun dapat dinonaktifkan sementara untuk masa penutupan administrasi sebelum dihapus.

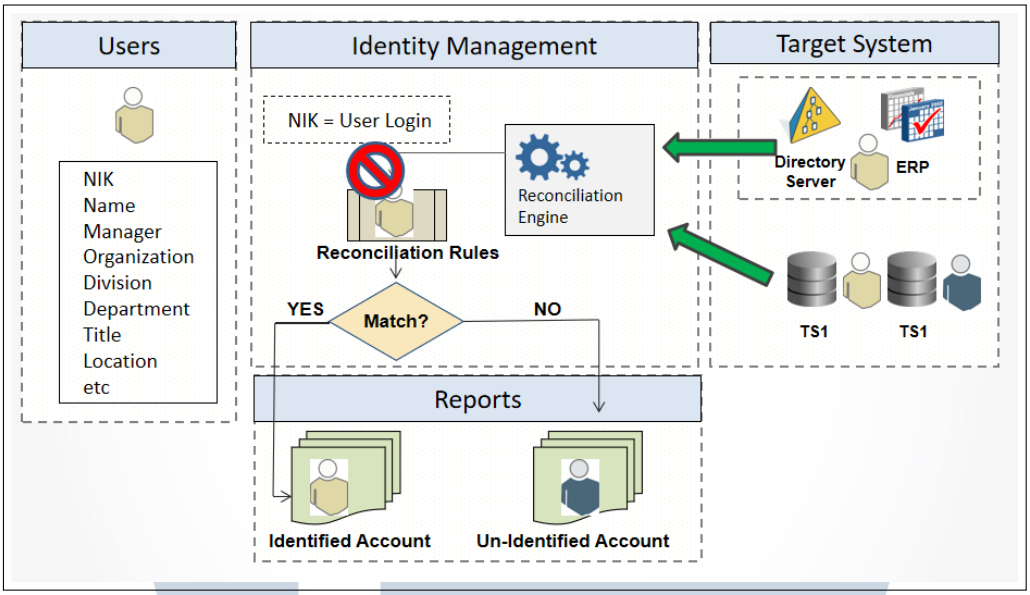


Gambar 3.4. Contoh alur *termination* saat pengguna tidak lagi aktif.

[25]

B.4. Identification

Pada lingkungan yang sudah berjalan, sering ada akun lama yang belum terhubung ke identitas pusat. Alur ini memindai akun pada sistem tujuan, lalu mencocokkannya dengan identitas di platform IDM menggunakan aturan yang disepakati. Akun yang cocok dihubungkan ke pemiliknya sehingga riwayat dan hak akses dapat dikelola secara terpadu. Akun yang tidak jelas kepemilikannya dimasukkan ke daftar tinjauan untuk diputuskan apakah digabungkan, dinonaktifkan, atau dibersihkan.

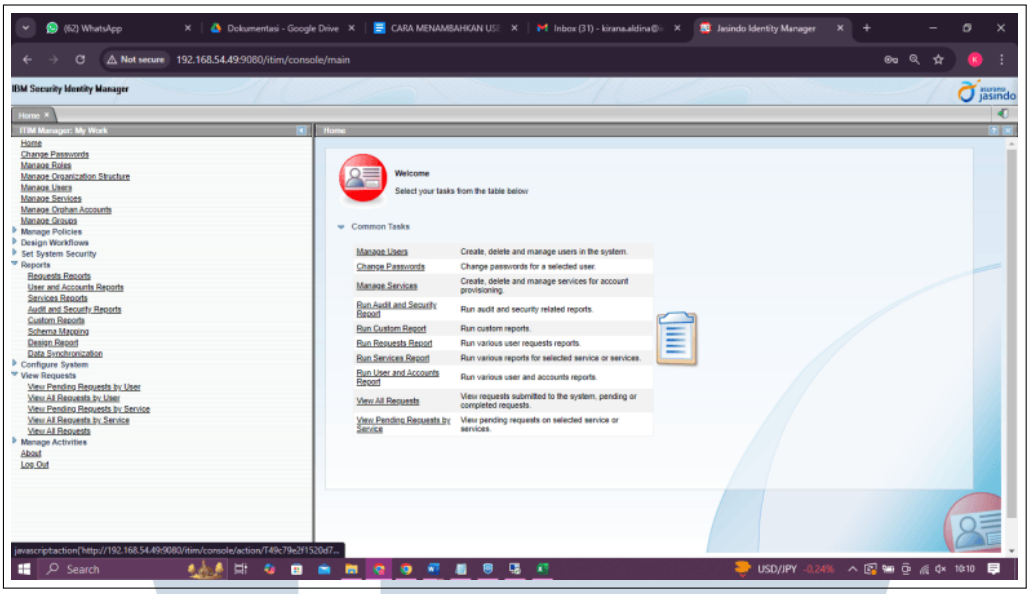


Gambar 3.5. Contoh alur *identification* untuk memasangkan akun yang sudah ada. [25]

3.3.2 Gambaran Umum Sistem Existing IBM Security Identity Manager

Sebelum pelaksanaan *proof of concept* (PoC), IBM Security Identity Manager (ISIM) telah digunakan sebagai platform terpusat untuk pengelolaan identitas dan akses pada PT Asuransi JASINDO. Implementasi ISIM memanfaatkan HR-IS (Star SDM) serta data pengguna *outsource* dalam bentuk berkas CSV sebagai *trusted source* yang memicu proses pengelolaan akun pada berbagai sistem tujuan.

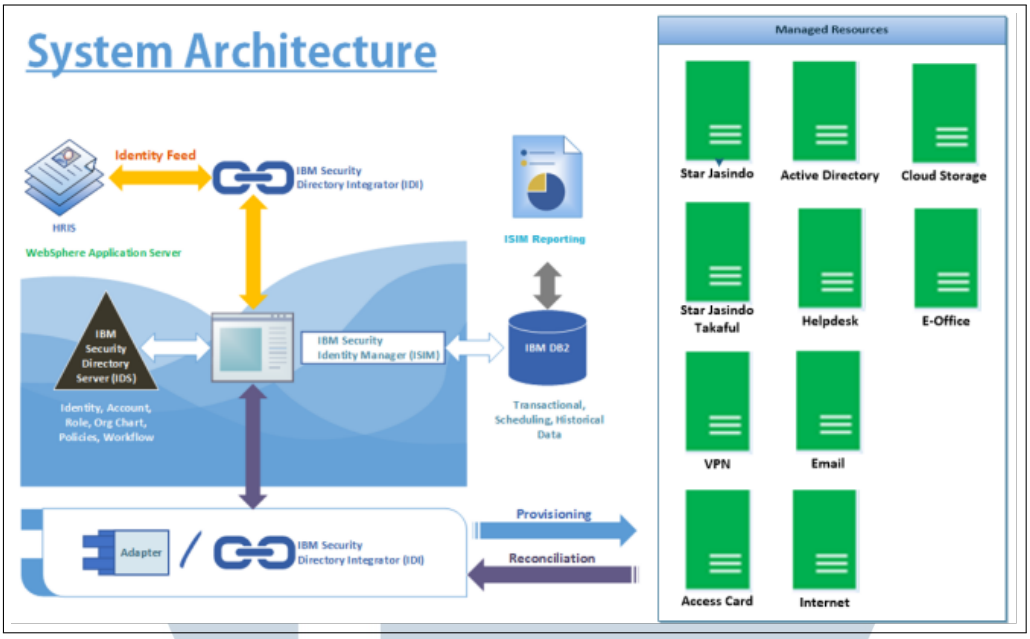
Tampilan awal aplikasi ISIM pada sisi operasional ditunjukkan pada Gambar 3.6. Melalui antarmuka ini, permintaan akses, pelacakan status, serta peninjauan riwayat permintaan dapat dilakukan sesuai alur persetujuan yang berlaku.



Gambar 3.6. Dashboard ISIM

A Arsitektur Sistem Existing

Arsitektur ISIM pada lingkungan *existing* terdiri atas DB2 sebagai penyimpan data transaksi dan riwayat, direktori LDAP sebagai penyimpan keadaan terkini identitas, SDI/TDI sebagai komponen sinkronisasi identitas lintas repositori dan aplikasi, serta WebSphere Application Server (WAS) sebagai *application server*. Integrasi ke sistem tujuan dilakukan melalui *adapter* yang dapat bersifat *agent-based* maupun *agentless*. Arsitektur komponen utama ISIM ditunjukkan pada Gambar 3.7.

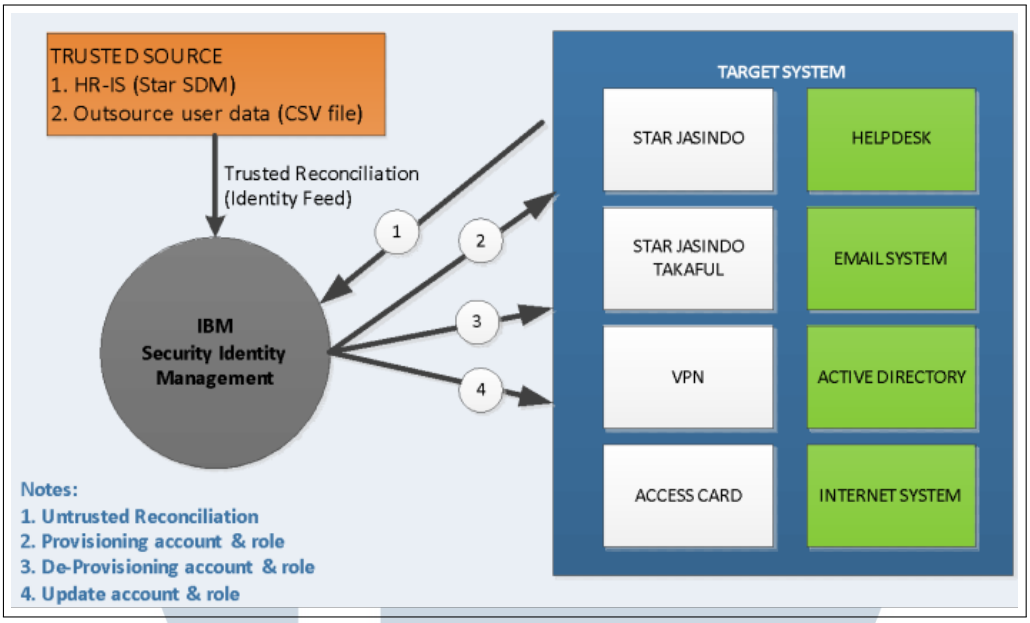


Gambar 3.7. Arsitektur komponen utama ISIM pada lingkungan *existing* [25]

Pada sisi operasional, layanan ISIM bergantung pada layanan pendukung, antara lain layanan DB2, direktori (TDS/LDAP), SDI/TDI, WAS/ISIM, serta *SMTP relay* untuk pengiriman notifikasi. Jika salah satu layanan inti tidak tersedia, proses permintaan akun, persetujuan, sinkronisasi, dan pencatatan aktivitas dapat terhambat.

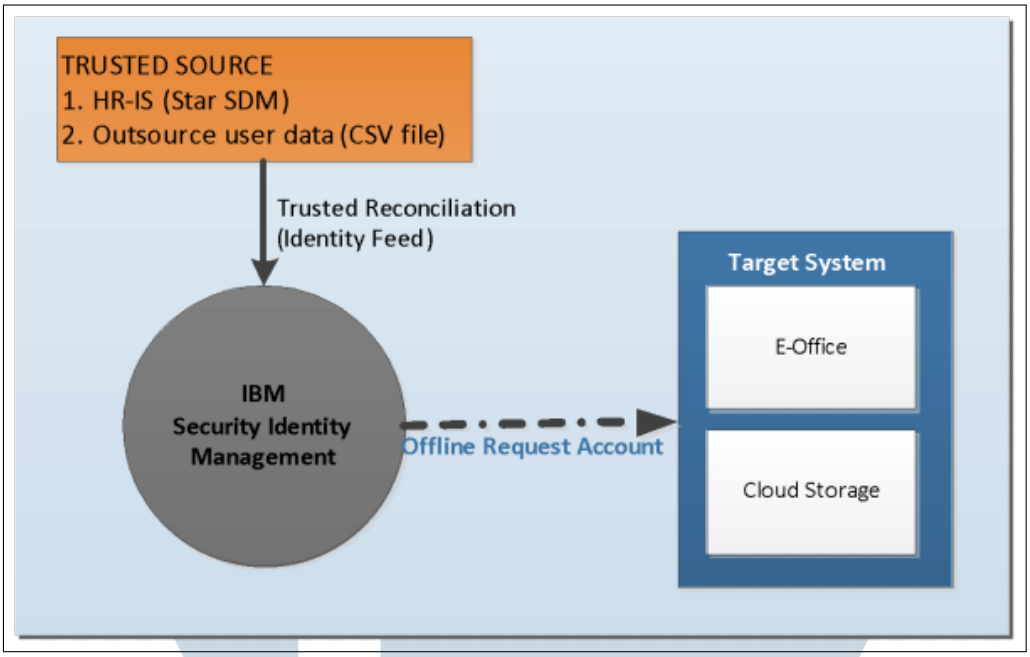
B Alur Kerja Umum Pengelolaan Akun

Pada kategori sistem tujuan *online*, alur kerja umum pengelolaan akun mencakup proses *untrusted reconciliation*, *provisioning account & role*, *de-provisioning account & role*, serta *update account & role*. Sebelum proses tersebut berjalan, data identitas dimuat dari *trusted source* melalui proses *trusted reconciliation* (*identity feed*). Alur umum pada sistem tujuan *online* ditunjukkan pada Gambar 3.8.



Gambar 3.8. Alur umum pengelolaan akun ISIM untuk sistem tujuan *online* [25]

Implementasi *existing* membedakan proses bisnis sistem tujuan menjadi RBAC dan *self-request*. Selain itu, terdapat sistem tujuan *offline* yang tetap menggunakan *self-request*, tetapi pembuatan akun diselesaikan melalui tindak lanjut manual oleh admin sistem tujuan. Alur umum pada sistem tujuan *offline* ditunjukkan pada Gambar 3.9.



Gambar 3.9. Alur umum pengelolaan akun ISIM untuk sistem tujuan *offline* [25]

C Klasifikasi Sistem Tujuan pada Implementasi Existing

Pada implementasi *existing*, sistem tujuan yang dikelola mencakup sepuluh aplikasi, yaitu Star Jasindo Konvensional, Star Jasindo Takaful, Helpdesk, Active Directory, VPN, Sistem E-mail, Sistem Internet, Cloud Storage, E-Office, dan Access Card.

Pada subbab ini, klasifikasi dilakukan berdasarkan dua aspek, yaitu tipe integrasi (*Online*/*Offline*) dan metode pengelolaan akses (RBAC/*self-request*). Tipe *Online* menunjukkan proses pembuatan/perubahan akun dapat dieksekusi melalui mekanisme *provisioning* yang terintegrasi, sedangkan tipe *Offline* memerlukan tindak lanjut manual oleh admin sistem tujuan setelah permintaan disetujui. Metode *Role-Based Access Control* (RBAC) merupakan kontrol akses berbasis peran (role), yaitu pemberian otorisasi ditetapkan melalui peran organisasi sehingga akses dapat distandardisasi dan dikelola lebih konsisten. Sebaliknya, pada metode *self-request*, akun atau hak akses diajukan melalui formulir permintaan dan diproses melalui alur persetujuan sebelum dilakukan *provisioning* ke sistem tujuan. Ringkasan klasifikasi proses ditunjukkan pada Tabel 3.2.

Tabel 3.2. Klasifikasi sistem tujuan pada implementasi *existing* ISIM

Sistem tujuan	**Tipe**	**Metode**	**Ringkasan alur**
Sistem E-mail	Online	RBAC	Akun terbentuk otomatis setelah identitas tercatat pada sumber data berotoritas.
Helpdesk	Online	RBAC	Akun terbentuk otomatis. Pengguna dapat meninjau akun melalui menu peninjauan akun.
Active Directory	Online	RBAC	Akun terbentuk otomatis. Perubahan atribut dan status akun dipelihara oleh kebijakan ISIM.
Sistem Internet (Kantor Pusat)	Online	RBAC	Akun terbentuk otomatis untuk pengguna Kantor Pusat sesuai kebijakan layanan.
Star Jasindo Konvensional	Online	*self-request*	Permintaan akun diajukan melalui formulir. Persetujuan dilakukan berjenjang. Akun diprovisikan ke aplikasi.
Star Jasindo Takaful	Online	*self-request*	Permintaan akun diajukan melalui formulir. Permintaan mencakup pemilihan *mapping role* sesuai kebutuhan.
VPN	Online	*self-request*	Permintaan akun diajukan melalui formulir. Persetujuan dilakukan sesuai alur. Akun dibuat pada sistem tujuan.
Access Card	Online	*self-request*	Permintaan akun diajukan melalui formulir. Persetujuan dilakukan sesuai alur. Akun dibuat pada sistem tujuan.
			Lanjut pada halaman berikutnya

Tabel 3.2 Klasifikasi sistem tujuan pada implementasi *existing* ISIM (Lanjutan)

Sistem tujuan	**Tipe**	**Metode**	**Ringkasan alur**
E-Office	Offline	*self-request*	Setelah disetujui, permintaan menunggu tindak lanjut admin *target system*. Status dapat menampilkan *pending response*. Pembuatan akun dilakukan secara manual.
Cloud Storage	Offline	*self-request*	Setelah disetujui, permintaan menunggu tindak lanjut admin. Status dapat menampilkan *pending response* sebelum selesai.

D Alur *Role Based Access Control*

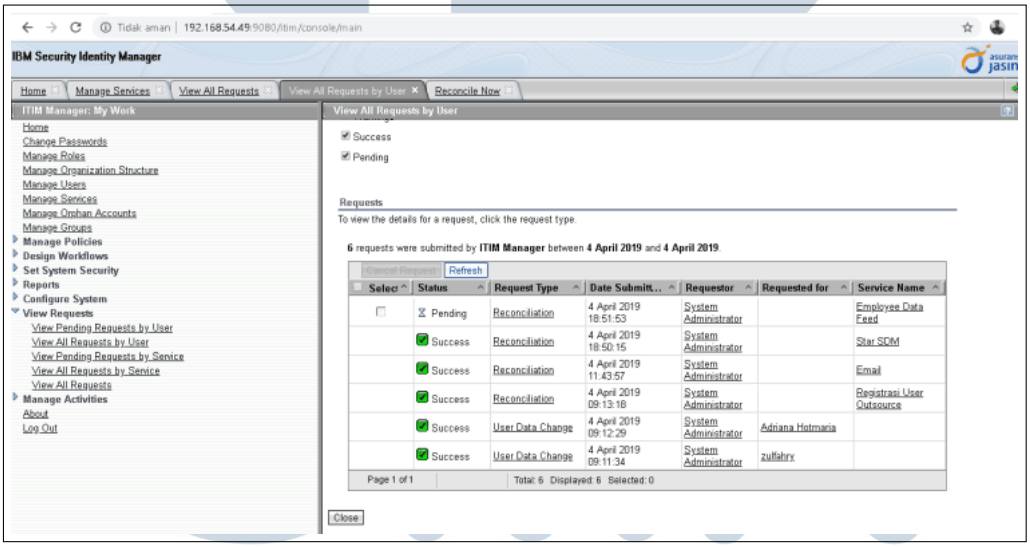
Pada sistem tujuan RBAC, proses pembuatan akun dijalankan secara otomatis ketika terdapat identitas baru pada *trusted source*. Proses ini digunakan pada Sistem E-mail, Helpdesk, Active Directory, serta Sistem Internet (Kantor Pusat). Pada sisi pengguna, akun yang sudah terbentuk dapat ditinjau melalui fitur peninjauan akun, misalnya melalui menu View or Change Account, tanpa melakukan permintaan pembuatan akun terlebih dahulu.

E Alur *Self-request* pada Sistem Tujuan Online

Pada sistem tujuan *self-request* kategori *online*, permintaan akun diawali dari pengguna melalui menu Request Account dan pemilihan sistem tujuan. Pengguna kemudian mengisi formulir sesuai kebutuhan sistem tujuan dan mengirimkan permintaan. Setelah permintaan dikirim, ISIM mengirim notifikasi kepada pihak penyetuju. Penyetuju level pertama umumnya ditentukan dari data HR sebagai atasan langsung, sedangkan penyetuju lain ditentukan melalui peran sesuai sistem tujuan. Penyetuju dapat menyetujui atau menolak permintaan. Selanjutnya, ISIM menampilkan umpan balik status permintaan kepada pengguna. Pada tahap proses, status detail dapat berada pada kondisi seperti *Pending Information*, sedangkan hasil akhir permintaan dapat ditampilkan sebagai *success*, *failed*, atau *in process*. Visualisasi alur ditunjukkan pada Gambar 3.10.

Gambar 3.10. Context diagram untuk *target system* dengan *Self-Request online* [25]

Riwayat permintaan dan penelusuran status dapat ditinjau melalui daftar permintaan, misalnya melalui menu View All Request by User. Contoh riwayat permintaan (*historical request*) ditunjukkan pada Gambar 3.11.



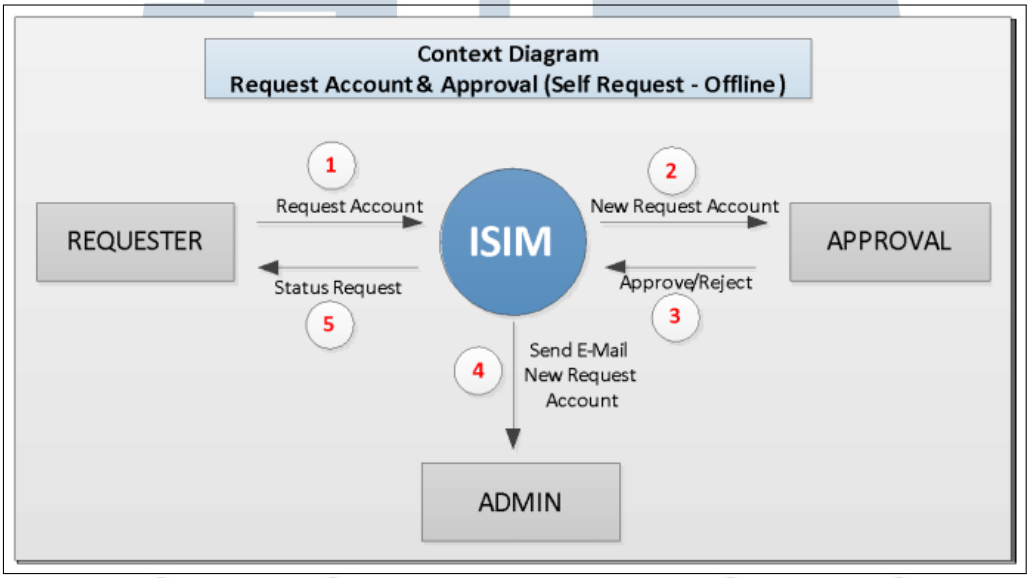
Gambar 3.11. Dahsboard *historical request* pada role ITIM Manager

F Alur *Self-request* pada Sistem Tujuan Offline

Pada sistem tujuan *offline* yaitu E-Office dan Cloud Storage, permintaan akun diawali melalui ISIM dengan pengisian formulir dan persetujuan berjenjang. Setelah permintaan memperoleh persetujuan, proses tidak langsung membuat akun pada aplikasi tujuan. ISIM meneruskan informasi permintaan kepada admin *target system*, termasuk melalui pengiriman formulir permintaan melalui email. Admin

target system kemudian melakukan registrasi akun secara manual pada aplikasi tujuan.

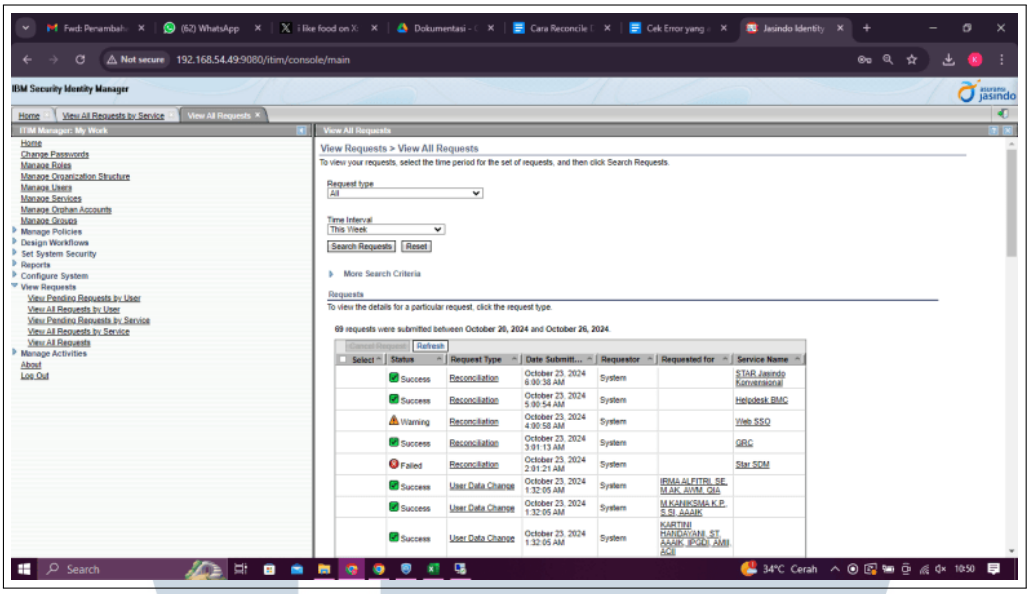
Dalam praktik operasional, status detail permintaan dapat berada pada kondisi *Pending Information* saat menunggu persetujuan. Status detail juga dapat berada pada kondisi *Pending response* saat menunggu tindak lanjut admin *target system*. Setelah proses manual selesai, status permintaan berubah menjadi *Success* dan notifikasi dapat diterima melalui email. Visualisasi alur ditunjukkan pada Gambar 3.12.



Gambar 3.12. Context diagram untuk *target system* dengan *Self-Request offline* [25]

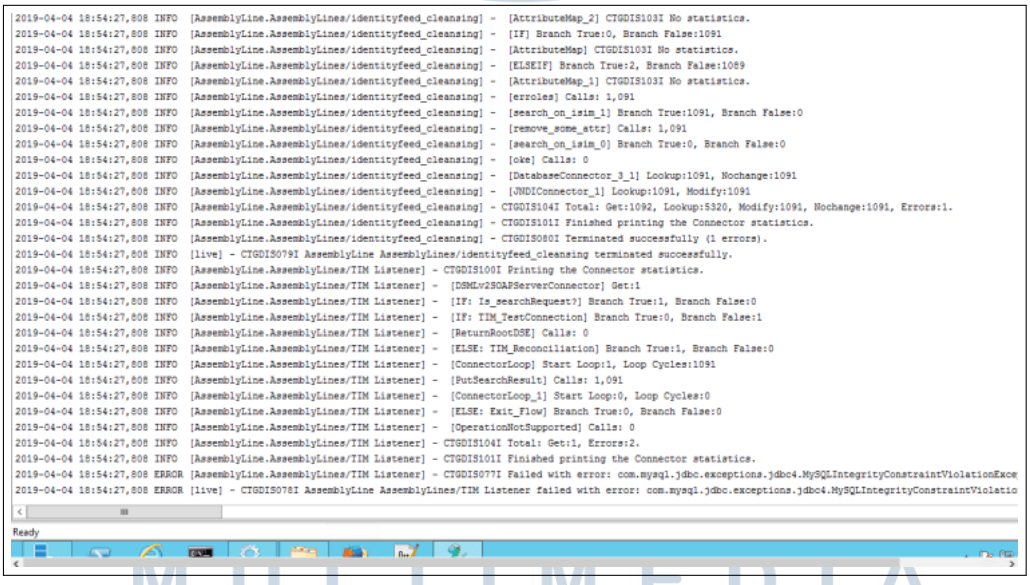
G Pemantauan dan Pemeliharaan Operasional

Pemantauan operasional pada lingkungan *existing* berfokus pada ketersediaan layanan pendukung dan kelancaran eksekusi alur. Aktivitas yang umum dilakukan mencakup pemantauan status layanan DB2, TDS, WAS dan ISIM, TDI, serta SMTP Relay. Pemeriksaan log dilakukan untuk mendukung analisis gangguan dan identifikasi penyebab error pada proses layanan. Selain itu, permintaan yang sedang berjalan ditinjau melalui dashboard untuk memastikan proses *request* berjalan sesuai antrian dan tidak terjadi kegagalan berulang. Dashboard pemantauan permintaan ditunjukkan pada Gambar 3.13.



Gambar 3.13. Dashboard untuk melihat permintaan yang berjalan

Pemeriksaan log runtime dilakukan untuk menganalisis indikasi gangguan dan mengidentifikasi penyebab error pada proses layanan. Contoh log runtime yang digunakan sebagai rujukan analisis ditunjukkan pada Gambar 3.14.



Gambar 3.14. Log runtime ISIM

3.3.3 Rencana Upgrade dari ISIM ke ISVG

Subbab ini menjelaskan rencana peningkatan dari sistem *existing* IBM Security Identity Manager (ISIM) menuju platform IBM Verify Identity

Governance yang sebelumnya dikenal sebagai IBM Security Verify Governance pada fase *proof of concept* (PoC). Rencana ini digunakan sebagai acuan untuk menyiapkan lingkungan PoC, memverifikasi kesiapan komponen, dan menghasilkan dokumentasi temuan teknis sebagai dasar tahap lanjutan.

IBM menyediakan dokumentasi resmi terkait proses *upgrade* dan migrasi dari IBM Security Identity Manager (ISIM) menuju IBM Security Verify Governance (ISVG), termasuk gambaran umum tahapan migrasi serta skenario *separate system upgrade* yang mencakup migrasi data basis data dan direktori ke lingkungan baru. Dokumentasi teknis tersebut bersifat modular dan tersebar pada beberapa bagian dokumentasi IBM, sehingga rujukan implementasi perlu dikompilasi dari beberapa topik yang saling terkait.

Pada praktiknya, kebutuhan implementasi tidak hanya bergantung pada dokumentasi publik, tetapi juga memerlukan referensi tambahan berupa panduan *How To*, *troubleshooting*, serta rujukan *fix pack* melalui portal dukungan IBM, seperti IBM Fix Central dan IBM Entitled Software Support (ESS). Sebagian konten pada portal dukungan IBM menerapkan akses berbasis IBMid dan *entitlement* (hak akses berdasarkan kepemilikan produk/perjanjian dukungan), sehingga tidak dapat diakses tanpa hak akses yang valid. Oleh karena itu, dokumen yang memerlukan akses khusus diperoleh melalui koordinasi dengan PT Asuransi JASINDO sebagai pemilik hak akses portal tersebut.

Dengan kondisi tersebut, kebutuhan referensi teknis pada fase *proof of concept* (PoC) disusun melalui verifikasi silang antara dokumentasi publik IBM dan dokumen dari PT Asuransi JASINDO, kemudian dikondensasikan menjadi dokumen internal sebagai acuan pelaksanaan PoC pada proyek ini.

Pemetaan komponen inti beserta kondisi saat ini dan acuan target ditunjukkan pada Tabel 3.3. Seluruh pemilihan versi pada lingkungan PoC mengacu pada matriks dukungan resmi dan paket instalasi yang digunakan, sehingga proses instalasi, penerapan *fix pack*, dan validasi dapat dijalankan secara konsisten.

Tabel 3.3. Pemetaan komponen dan versi

Komponen	**Versi saat ini**	**Versi target**	**Catatan**
ISIM	6.0	ISVG 10.0.2.x atau 11.0.x	Sesuaikan dengan paket PoC dan matriks dukungan.
Lanjut ke halaman berikutnya			

Komponen	Versi saat ini	Versi target	Catatan
IBM Security Directory Integrator (SDI)	7.2	7.2.x atau yang didukung	Untuk koneksi dan pemuatan identitas.
WebSphere Application Server	8.5	9.0.5.x atau yang didukung	Mengikuti matriks dukungan.
DB2	10.5	11.5.x atau yang didukung	Skema basis data aplikasi.
Directory Server (LDAP)	TDS 6.3	SDS 6.4.0.25 atau yang didukung	Digunakan sebagai direktori LDAP dan memerlukan konfigurasi sesuai kebutuhan sistem.
Database Scripts and Tools (DT Tool)	Tidak digunakan	Sesuai versi paket ISVG	Digunakan untuk instalasi skema database dan alat bantu migrasi.
Adapter Pack ISVG	Tidak tersedia	10.x atau 11.x	Konektor ke sistem tujuan.

Kesesuaian versi dan dependensi pada Tabel 3.3 diverifikasi sebelum pelaksanaan agar tahap instalasi, validasi, dan pendokumentasian hasil PoC dapat dijalankan tanpa konflik dukungan.

Rangkaian tahapan pekerjaan secara end-to-end dirangkum pada Tabel 3.4 untuk menggambarkan tujuan, tindakan utama, serta keluaran dan kriteria lulus pada setiap tahap. Pada fase *proof of concept* (PoC), penilaian keberhasilan ditekankan pada verifikasi kesiapan komponen dan lingkungan, serta tersusunnya dokumentasi temuan teknis. Berdasarkan pelaksanaan PoC, seluruh paket dan komponen pendukung telah terpasang dan dapat diverifikasi, sedangkan instalasi ISVG masih berada pada tahap finalisasi instalasi. Kondisi tersebut berdampak pada tahapan migrasi konfigurasi, rekonsiliasi, dan uji fungsional yang bergantung pada layanan ISVG. Oleh karena itu, keluaran PoC difokuskan pada *baseline*, hasil verifikasi layanan, serta dokumentasi kendala dan rekomendasi tindak lanjut untuk fase berikutnya.

Tabel 3.4. Tahapan upgrade end to end

Tahap	**Tujuan**	**Tindakan utama**	**Keluaran dan kriteria lulus**
Penilaian	Memahami kondisi awal	Inventaris alur, konektor, kebijakan, akun layanan, dan dependensi	*Baseline* konfigurasi tersusun dan tervalidasi, termasuk daftar kesenjangan dan risiko awal
Desain target	Menetapkan rancangan ISVG	Menentukan model peran, sumber identitas, pola sinkronisasi, dan daftar konektor	Rancangan target tersusun dan menjadi acuan konfigurasi pada lingkungan PoC
Persiapan uji	Menyediakan lingkungan uji	Memasang komponen, menyiapkan basis data, menghubungkan ke direktori, menyiapkan data contoh	Komponen pendukung terpasang dan layanan dasar terverifikasi berjalan. Instalasi ISVG berada pada tahap finalisasi instalasi dan temuan teknis terdokumentasi
Migrasi konfigurasi	Menyiapkan proses migrasi konfigurasi	Menyusun pemetaan konfigurasi, skema atribut, kebijakan, dan alur kerja yang akan diterapkan pada ISVG	Dokumen pemetaan migrasi dan rencana eksekusi tersusun. Eksekusi migrasi dijadwalkan pada fase berikutnya setelah ISVG siap
Lanjut ke halaman berikutnya			

Tabel 3.4 Tahapan upgrade end to end (Lanjutan)

Tahap	**Tujuan**	**Tindakan utama**	**Keluaran dan kriteria lulus**
Rekonsiliasi awal	Menyiapkan verifikasi penyelarasan akun	Menyusun skenario rekonsiliasi, aturan adopsi, dan kriteria identifikasi akun yatim	Skenario rekonsiliasi dan kriteria validasi tersusun, termasuk data uji yang dibutuhkan untuk fase berikutnya
Uji fungsional	Menyiapkan verifikasi skenario utama	Menyusun *test case* untuk permintaan akses, persetujuan, perubahan peran, dan pencabutan akses	Daftar skenario uji dan kriteria penerimaan tersusun. Hasil verifikasi prasyarat dan catatan kendala terdokumentasi sebagai keluaran PoC

A Rencana Eksekusi Upgrade dan Migrasi

Rencana eksekusi upgrade disusun dengan pendekatan *parallel run* untuk menjaga kontinuitas layanan. Pada pendekatan ini, lingkungan ISVG dirancang dibangun terpisah dari sistem *existing*. Tahap awal difokuskan pada penyiapan komponen pendukung, verifikasi prasyarat, serta pendokumentasian temuan teknis sebagai dasar penyelesaian instalasi ISVG dan tahap migrasi pada fase berikutnya.

Setelah ISVG siap beroperasi, konfigurasi inti dan konektivitas ke sistem tujuan direncanakan dipindahkan secara bertahap. Tahap ini mencakup rekonsiliasi awal dan pengujian end-to-end sesuai skenario uji yang telah disusun. Apabila hasil pengujian menunjukkan kondisi stabil, *cutover* direncanakan dilakukan secara terjadwal pada *freeze window* agar perubahan akses tetap terkendali.

Untuk mitigasi risiko, mekanisme *rollback* disiapkan melalui pengaktifan kembali alur *provisioning* pada sistem *existing* serta pemulihan data menggunakan *backup* yang dibuat sebelum *cutover*.

B Persiapan *Baseline* dan Pembuatan *Backup*

Sebelum perubahan dilakukan, seluruh komponen pendukung dicatat sebagai *baseline* yang mencakup versi, konfigurasi, dependensi, akun layanan, serta *endpoint*. *Backup* dibuat sebelum perubahan besar, misalnya pada saat *upgrade* DB2, *upgrade* WebSphere Application Server, dan sebelum percobaan instalasi ISVG, agar proses pemulihan dapat dilakukan tanpa kehilangan konfigurasi maupun data audit. Pada fase *proof of concept* (PoC), pembuatan *backup* juga digunakan sebagai bagian dari verifikasi kesiapan lingkungan dan sebagai artefak pendukung untuk fase berikutnya.

Prosedur *backup* basis data DB2 dilakukan menggunakan perintah pada Kode 3.1. *Backup* basis data ISIM dijalankan untuk menjaga konsistensi pemulihan. Untuk basis data ISVG, *backup* dijalankan apabila basis data ISVG telah disiapkan pada lingkungan PoC dan terdaftar pada DB2.

```
# dijalankan sebagai instance owner DB2

# backup database ISIM
db2 backup database ISIM_JASINDO to /backup/db2/ compress include
    logs

# backup database ISVG dijalankan jika database sudah dibuat dan
    terdaftar
# pengecekan sederhana apakah database ISVG terdaftar di DB2
db2 list db directory | grep -i ISVG_JASINDO

# jika database ISVG terdaftar, jalankan backup berikut
db2 backup database ISVG_JASINDO to /backup/db2/ compress include
    logs
```

Kode 3.1: Backup DB2 sebelum perubahan besar

Selain basis data, *backup* konfigurasi WebSphere dilakukan untuk mendukung kebutuhan *rollback*. *Backup* konfigurasi profil dapat dibuat menggunakan utilitas `backupConfig.sh`. Pada lingkungan Linux, *backupConfig* menghasilkan arsip `zip` untuk konfigurasi. Apabila diperlukan pemulihan yang mempertahankan *permission* dan *ownership* secara identik, arsip `tar.gz` juga dapat dibuat terhadap direktori profil. Contoh perintah *backup* ditunjukkan pada Kode 3.2.

```
# backup konfigurasi profil WebSphere menggunakan backupConfig
/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/bin/backupConfig.sh
    \
```

```
  /backup/was_profile_AppSrv01_$(date +%F).zip

# opsi tambahan untuk mempertahankan permission dan ownership pada Linux
tar -czf /backup/was_profile_AppSrv01_$(date +%F).tgz \
  /opt/IBM/WebSphere/AppServer/profiles/AppSrv01

# backup direktori aplikasi atau log yang dipertahankan untuk rollback
# sesuaikan jalur direktori dengan struktur instalasi yang digunakan
tar -czf /backup/isim_runtime_$(date +%F).tgz \
  /opt/IBM/isim/var/itim/var/log/was
```

Kode 3.2: Backup konfigurasi WebSphere dan direktori aplikasi

C Upgrade dan Penyesuaian Komponen Berbasis Java (WAS)

Upgrade WebSphere Application Server dilakukan hingga versi target yang didukung pada lingkungan *proof of concept* (PoC) dan dilanjutkan dengan penerapan *fix pack*. Penerapan *fix pack* dilakukan melalui *repository* internal yang berisi paket dan pembaruan yang telah dikurasi, sehingga proses *patching* dapat dikendalikan sesuai standar dukungan. Proses instalasi atau *upgrade* dan penerapan *fix pack* dilakukan menggunakan IBM Installation Manager melalui perintah *imcl* seperti ditunjukkan pada Kode 3.3. Penggunaan perintah `updateAll` mengacu pada kondisi di mana *repository* yang digunakan berada dalam kendali penuh dan memuat pembaruan yang ditetapkan untuk diterapkan.

```
# WebSphere install atau upgrade dari repository internal
# packageId disesuaikan dengan edisi yang digunakan pada lingkungan PoC
# contoh packageId yang umum digunakan adalah:
# com.ibm.websphere.ND.v90 untuk Network Deployment
# com.ibm.websphere.BASE.v90 untuk Base

/opt/IBM/InstallationManager/eclipse/tools/imcl \
  install com.ibm.websphere.ND.v90 \
  -repositories /repo/ibm/was/9.0.5/ \
  -installationDirectory /opt/IBM/WebSphere/AppServer \
  -acceptLicense

```

```
# penerapan fix pack atau iFix dari repository internal yang telah dikurasi
/opt/IBM/InstallationManager/eclipse/tools/imcl \
  updateAll -installFixes recommended \
  -repositories /repo/ibm/was/fixpack/ \
  -installationDirectory /opt/IBM/WebSphere/AppServer \
  -acceptLicense
```

Kode 3.3: Upgrade WebSphere via IBM Installation Manager (imcl)

Setelah proses *patching* selesai, penyesuaian parameter JVM dilakukan untuk memastikan runtime WAS berada pada konfigurasi yang stabil sebagai prasyarat operasional aplikasi pada fase lanjutan. Penyesuaian dilakukan melalui *wsadmin* (Jython) dengan pengaturan ukuran *heap* dan argumen JVM yang relevan seperti ditunjukkan pada Kode 3.4. Nilai *heap* disesuaikan dengan kapasitas server dan hasil observasi *garbage collection* pada lingkungan PoC.

```
# jalankan dengan:
# /opt/IBM/WebSphere/AppServer/bin/wsadmin.sh -lang jython -f set_jvm_runtime.py

nodeName   = "Node01"
serverName = "server1"

# konfigurasi JVM untuk beban aplikasi pada lingkungan PoC
other = [
  ['initialHeapSize', '2048'],
  ['maximumHeapSize', '4096'],
  ['genericJvmArguments', '-Djava.net.preferIPv4Stack=true -Dhttps.protocols=TLSv1.2']
]

AdminServerManagement.configureJavaVirtualMachine(
  nodeName, serverName, 'false', '', other
)

AdminConfig.save()
```

Kode 3.4: Penyesuaian JVM WAS menggunakan wsadmin (Jython)

Setelah perubahan JVM disimpan, *restart* layanan dilakukan secara terkontrol untuk memastikan konfigurasi baru diterapkan. Pada lingkungan yang menerapkan keamanan administrasi, autentikasi dapat disediakan melalui parameter perintah atau melalui pengaturan pada berkas konfigurasi klien SOAP. Pada

penulisan laporan ini, kredensial ditampilkan dalam bentuk disamarkan seperti ditunjukkan pada Kode 3.5.

```
/opt/IBM/WebSphere/AppServer/bin/stopServer.sh server1 -username
    wasadmin -password REDACTED
/opt/IBM/WebSphere/AppServer/bin/startServer.sh server1
```

Kode 3.5: Restart terkontrol WAS setelah perubahan JVM

D Penyesuaian SDI/TDI dan Konektivitas Adapter

Komponen SDI dipertahankan pada versi yang didukung dan difokuskan pada penyesuaian konektivitas yang mencakup kredensial, *endpoint*, sertifikat, serta pengaturan TLS agar komunikasi dengan *target system* berjalan konsisten. Setelah file properti adapter atau SDI diperbarui, layanan *Dispatcher* perlu dihentikan dan dijalankan kembali agar perubahan diterapkan. Pada fase *proof of concept* (PoC), langkah ini digunakan untuk memuat ulang konfigurasi dan memverifikasi konektivitas adapter sebagai persiapan tahap lanjutan setelah layanan ISVG siap digunakan. Contoh penempatan paket adapter dan prosedur *restart Dispatcher* ditunjukkan pada Kode 3.6.

```
# contoh penempatan library adapter atau connector
# sesuaikan jalur direktori dengan struktur instalasi SDI yang
    digunakan
cp /repo/ibm/isvg/adapters/*.jar /opt/IBM/TDI/V7.2/jars/connectors
    /
cp /repo/ibm/isvg/adapters/patch/*.jar /opt/IBM/TDI/V7.2/jars/
    patch/

# restart dispatcher menggunakan script layanan adapter
cd /opt/IBM/TDI/V7.2/timsol
./ITIMAd stop
./ITIMAd start

# verifikasi proses dispatcher berjalan
ps -ef | grep ibmdisrv
```

Kode 3.6: Deploy paket adapter dan restart Dispatcher

E Upgrade DB2 dan Instalasi Skema Basis Data ISVG

DB2 dipastikan berada pada versi target yang didukung pada lingkungan *proof of concept* (PoC). Pada tahap ini, basis data ISVG dipersiapkan agar proses instalasi aplikasi ISVG dapat melanjutkan tahap inisialisasi skema pada fase berikutnya. Persiapan meliputi aktivasi komunikasi TCP/IP pada instance DB2, serta pemberian *database authority* yang diperlukan kepada *instance owner* yang digunakan untuk pengelolaan basis data ISVG.

Pemberian hak akses dilakukan pada basis data `ISVG_JASINDO` menggunakan perintah pada Kode 3.7. Otorisasi pemberian *authority* mengikuti ketentuan DB2, yaitu pemberi *grant* harus memiliki kewenangan yang sesuai.

```
# dijalankan pada server DB2 (sebagai user yang memiliki otoritas untuk melakukan GRANT)

# pastikan environment instance yang digunakan
export DB2INSTANCE=igiinst
. ~/${DB2INSTANCE}/sqllib/db2profile

# pastikan komunikasi DB2 menggunakan TCP/IP (sesuaikan bila sudah terkonfigurasi)
db2set DB2COMM=tcpip
db2stop
db2start

# pemberian database authority kepada instance owner ISVG
db2 connect to ISVG_JASINDO

db2 "GRANT DBADM  ON DATABASE TO USER igiinst"
db2 "GRANT SECADM ON DATABASE TO USER igiinst"
```

Kode 3.7: Konfigurasi komunikasi DB2 dan pemberian hak untuk instance owner ISVG

Setelah *authority* diberikan, basis data ISVG dipersiapkan menggunakan paket *Database Installation Scripts and Tools* dari *repository* internal. Proses ini dilakukan dengan menjalankan skrip `db2_install.sh` beserta parameter yang diperlukan, termasuk *tablespace path* dan batas ukuran *tablespace*. Nilai *tablespace* ditetapkan agar kebutuhan audit dan proses rekonsiliasi pada fase lanjutan dapat didukung tanpa hambatan kapasitas. Contoh eksekusi ditunjukkan pada Kode 3.8.

```
# lokasi mengikuti paket Database Installation Scripts and Tools yang dipindahkan ke server DB2
cd /repo/ibm/isvg/DB_Tools/db2/__FOR_DBAs__/
```

```
3
4  # parameter instalasi database ISVG
5  # format FQ_IGI_DB: host:port/DBNAME
6  export IGI_DB='ISVG_JASINDO'
7  export FQ_IGI_DB='db2jas01:50050/ISVG_JASINDO'
8  export INSTANCE_OWNER='igiinst'
9  export PASSWORD='REDACTED'
10
11 # tablespace configuration (S kecil, M sedang, L besar sesuai
       dokumentasi paket)
12 export TABLESPACE_SIZE='M'
13 export TABLESPACE_PATH='/db2data/isvg/ISVG_JASINDO'
14 export TABLESPACE_MAXSIZE='16'
15 export TABLESPACE_TEMP_MAXSIZE='4'
16
17 # muat environment DB2 (sesuaikan home instance yang digunakan)
18 export DB2INSTANCE=igiinst
19 . ~/${DB2INSTANCE}/sqllib/db2profile
20
21 # normalisasi EOL bila paket berasal dari Windows
22 dos2unix db2_install.sh
23
24 # jalankan instalasi/persiapan skema
25 ./db2_install.sh
```

Kode 3.8: Persiapan skema database ISVG menggunakan db2_install.sh

F Penyesuaian Per Sistem Tujuan dan Validasi Konektivitas

Penyesuaian dilakukan per sistem tujuan dengan prinsip yang sama, yaitu memastikan akun layanan tersedia, hak akses pada sistem tujuan memadai, sertifikat TLS telah dipercaya, dan *endpoint* dapat dijangkau dari sisi server *Dispatcher* pada lingkungan *proof of concept* (PoC). Validasi konektivitas pada tahap ini ditujukan untuk memastikan prasyarat jaringan dan keamanan komunikasi telah terpenuhi sebagai dasar integrasi pada fase berikutnya.

Validasi konektivitas terhadap Active Directory melalui LDAPS dilakukan dari sisi *Dispatcher* untuk memastikan koneksi TLS dan proses *bind* LDAP berjalan normal. Contoh validasi TLS dan kueri LDAP ditunjukkan pada Kode 3.9.

```
1  # validasi koneksi TLS dan sertifikat pada port LDAPS 636
2  # jika diperlukan, tambahkan -servername sesuai nama DNS untuk
       kebutuhan SNI
```

```
echo | openssl s_client -connect ad01.jasindo.local:636 -showcerts

# validasi bind dan pencarian LDAP melalui LDAPS
ldapsearch -x -H ldaps://ad01.jasindo.local:636 \
  -D "CN=svc_isvg_ad,OU=ServiceAccounts,DC=jasindo,DC=local" -W \
  -b "DC=jasindo,DC=local" "(objectClass=user)" cn sAMAccountName
   -LLL | head
```

Kode 3.9: Validasi konektivitas Active Directory (LDAPS) dari Dispatcher

Validasi *endpoint* aplikasi dilakukan untuk memastikan layanan dapat diakses dari sisi server sesuai jalur jaringan yang berlaku. Pemeriksaan *endpoint* internal menggunakan `curl` dilakukan sebagaimana ditunjukkan pada Kode 3.10.

```
# endpoint internal disesuaikan dengan host aktual
# opsi -k digunakan pada tahap PoC jika trust chain sertifikat
   belum sepenuhnya dikonfigurasi di host penguji
curl -k -I https://helpdesk.jasindo.local/health
curl -k -I https://star-jasindo.jasindo.local/health
curl -k -I https://star-takaful.jasindo.local/health
```

Kode 3.10: Validasi endpoint aplikasi Helpdesk dan Star

Validasi notifikasi email dilakukan untuk memastikan server aplikasi dapat terhubung ke SMTP relay dan mendukung mekanisme STARTTLS sesuai kebijakan keamanan yang diterapkan. Pengujian port dan negosiasi STARTTLS dilakukan sebagaimana ditunjukkan pada Kode 3.11.

```
# uji port SMTP relay
nc -vz smtp-relay.jasindo.local 25

# uji negosiasi STARTTLS SMTP
printf "EHLO poc\r\nQUIT\r\n" | openssl s_client -starttls smtp -
   connect smtp-relay.jasindo.local:25 -crlf
```

Kode 3.11: Validasi notifikasi email (SMTP relay) dari server aplikasi

Validasi akses jaringan ke VPN gateway dilakukan untuk memastikan *reachability* jaringan dan ketersediaan respons layanan pada jalur akses yang digunakan. Pengujian dilakukan melalui *ping* dan pemeriksaan port atau respons HTTPS sebagaimana ditunjukkan pada Kode 3.12.

```
# ping dapat tidak diizinkan pada sebagian perangkat jaringan,
   sehingga verifikasi port juga disertakan
ping -c 3 vpn-gateway.jasindo.local

```

```
# verifikasi port layanan (misalnya HTTPS)
nc -vz vpn-gateway.jasindo.local 443
curl -k -I https://vpn-gateway.jasindo.local/
```

Kode 3.12: Validasi akses jaringan VPN gateway

G Rekonsiliasi Awal, Uji *Provisioning*, dan *Cutover*

Setelah konektivitas dinyatakan siap, tahapan rekonsiliasi awal direncanakan dilakukan untuk setiap sistem tujuan agar akun yang sudah ada dapat terbaca dan dipetakan ke identitas pusat. Pada tahap ini, parameter verifikasi disusun untuk memastikan atribut utama dapat terbaca konsisten, yaitu *identifier*, unit, status, dan *entitlement*. Selain itu, aturan adopsi akun dan identifikasi akun yatim juga dipersiapkan sebagai bagian dari skenario rekonsiliasi.

Selanjutnya, uji *provisioning* terkontrol direncanakan dijalankan menggunakan akun uji migrasi untuk memverifikasi pembuatan akun, pembaruan, dan pencabutan akses sesuai kebijakan yang ditetapkan. Pada fase *proof of concept* (PoC), pelaksanaan rekonsiliasi dan uji *provisioning* yang bergantung pada layanan ISVG belum dijalankan karena instalasi ISVG masih berada pada tahap finalisasi instalasi. Oleh karena itu, keluaran PoC difokuskan pada penyusunan skenario uji, kriteria penerimaan, serta dokumentasi kendala dan rencana tindak lanjut untuk fase berikutnya.

Cutover direncanakan dilakukan setelah layanan ISVG siap digunakan dan hasil pengujian menunjukkan kondisi stabil. Tahap *cutover* mencakup pembekuan perubahan akses pada sistem *existing*, aktivasi alur *provisioning* pada ISVG, serta pemantauan antrian *job* dan log eksekusi sampai kondisi stabil. Untuk mendukung kebutuhan monitoring dan percepatan analisis gangguan, pemantauan log WAS dan log *Dispatcher* dilakukan menggunakan perintah pada Kode 3.13.

```
# WAS log utama pada profile server aplikasi
tail -n 200 -f /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs
    /server1/SystemOut.log
tail -n 200 -f /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs
    /server1/SystemErr.log

# Dispatcher dan adapter berbasis SDI menggunakan log yang
    ditentukan pada konfigurasi log4j
# sesuaikan nama file log dengan implementasi pada server
tail -n 200 -f /opt/IBM/TDI/V7.2/logs/ibmdi.log
```

```
8
9  # opsi pencarian cepat untuk indikasi error berulang
10 grep -i -E "exception|error|severe" /opt/IBM/WebSphere/AppServer/
       profiles/AppSrv01/logs/server1/SystemOut.log | tail
11 grep -i -E "exception|error|severe" /opt/IBM/TDI/V7.2/logs/ibmdi.
       log | tail
```

Kode 3.13: Pemantauan log WAS dan Dispatcher selama validasi dan rencana cutover

3.3.4 Pelaksanaan Upgrade

Pelaksanaan upgrade dilakukan pada lingkungan *proof of concept* (PoC) JASINDO dengan tujuan menyiapkan platform IBM Verify Identity Governance (selanjutnya disebut ISVG) sebagai kandidat pengganti IBM Security Identity Manager (ISIM) pada fase uji konsep. Pada fase PoC, pekerjaan difokuskan pada penyediaan dan verifikasi prasyarat lingkungan, meliputi instalasi serta pemeriksaan layanan pendukung seperti DB2, WebSphere Application Server, dan komponen integrasi SDI/*Dispatcher*, termasuk validasi konektivitas menuju sistem tujuan. Verifikasi pada tahap ini memastikan komponen pendukung berada pada kondisi siap pakai sebagai fondasi untuk proses instalasi dan konfigurasi ISVG.

Keamanan proses migrasi pada fase *proof of concept* (PoC) dijaga dengan menjalankan lingkungan uji secara *parallel* terhadap sistem *existing*. Lingkungan PoC disediakan dalam bentuk server terpisah dengan alamat IP internal, sehingga aktivitas pengujian dilakukan pada jaringan internal dan tidak mengganggu layanan ISIM yang masih berjalan. Pada fase ini, PT Asuransi JASINDO juga menyediakan *adapter existing* yang dikonfigurasi dalam mode *read-only*, sehingga proses yang dilakukan terbatas pada pembacaan dan verifikasi data tanpa melakukan perubahan pada sistem tujuan. Pembatasan hak akses tersebut mengikuti prinsip *least privilege*, yaitu memberikan hak akses minimum yang diperlukan untuk menjalankan kebutuhan pengujian.

Selama *parallel run*, sistem *existing* ISIM tetap aktif dan digunakan untuk operasional harian, sedangkan PoC difokuskan pada validasi konektivitas, konsistensi data, serta pengecekan integrasi pada lingkungan uji. Pemantauan dilakukan secara berkala untuk memastikan tidak terdapat anomali atau *error* pada proses integrasi, serta dilengkapi pemeriksaan rutin bulanan (*monthly check-up*) sesuai prosedur operasional. Dengan pendekatan tersebut, risiko terhadap sistem produksi dapat diminimalkan karena pengujian dilakukan pada lingkungan

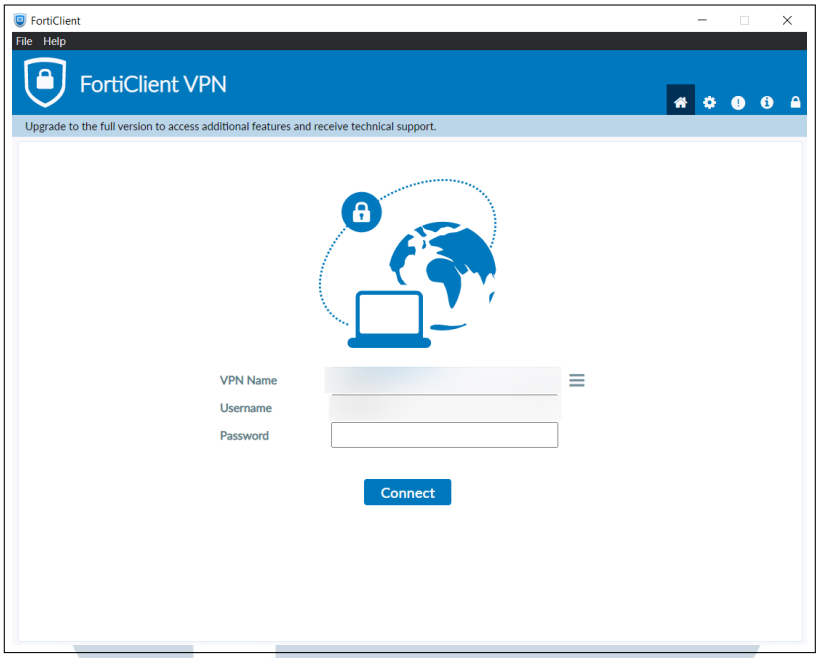
terisolasi dengan akses terbatas, sementara layanan *existing* tetap berjalan normal.

Hingga akhir periode PoC, layanan aplikasi ISVG belum berada pada kondisi siap operasi karena instalasi masih berada pada tahap finalisasi. Kondisi tersebut menyebabkan tahapan yang bergantung pada layanan ISVG, seperti inisialisasi skema aplikasi pada DB2, migrasi konfigurasi, serta pengujian end-to-end, belum dapat dieksekusi pada periode PoC. Oleh karena itu, keluaran PoC difokuskan pada dokumentasi hasil verifikasi prasyarat lingkungan, catatan kendala instalasi, dan rencana tindak lanjut sebagai dasar pelaksanaan pada fase berikutnya.

Lingkungan *proof of concept* ditempatkan pada server milik JASINDO dan diakses dari jaringan kantor. JASINDO menetapkan sistem operasi berbasis keluarga Red Hat sebagai standar server untuk menjaga kesesuaian dengan kebutuhan operasional dan kompatibilitas komponen. Sebelum eksekusi instalasi, dilakukan pemeriksaan awal terhadap matriks dukungan serta verifikasi identitas sistem operasi untuk memastikan platform yang digunakan berada pada keluarga Red Hat dan sesuai dengan prasyarat instalasi. Logo Red Hat Linux ditunjukkan pada Gambar 3.15.

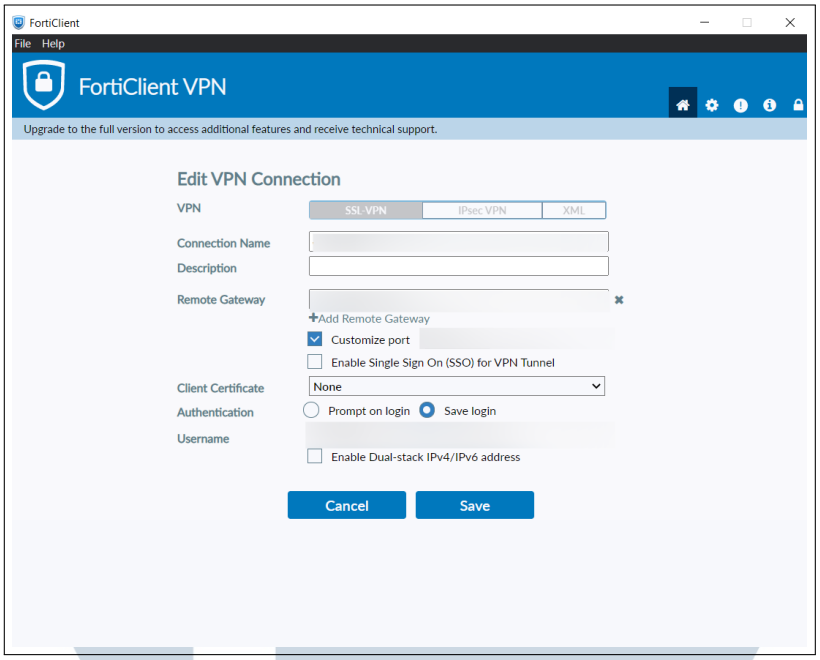
Gambar 3.15. Logo Red Hat Linux

Akses jaringan menuju server PoC dilakukan melalui VPN menggunakan FortiClient. FortiClient digunakan untuk membentuk *tunnel* menuju jaringan internal JASINDO, sehingga alamat IP server PoC hanya dapat dijangkau setelah koneksi VPN aktif dan rute internal tersedia. Aplikasi FortiClient ditunjukkan pada Gambar 3.16.



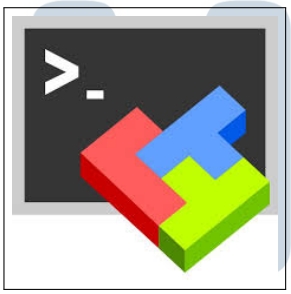
Gambar 3.16. Aplikasi VPN FortiClient

Setelah kredensial VPN diberikan oleh JASINDO, tahap berikutnya adalah melakukan konfigurasi profil koneksi pada FortiClient agar perangkat dapat terhubung ke jaringan internal perusahaan. Pada FortiClient, pengguna membuat koneksi VPN baru, misalnya SSL-VPN, dengan mengisi informasi dasar seperti nama koneksi dan alamat *remote gateway* yang diberikan. Setelah konfigurasi tersimpan, koneksi dapat dijalankan melalui menu *Remote Access* dengan memasukkan *username* dan *password* sesuai kredensial yang diberikan. Lokasi pengaturan koneksi FortiClient ditunjukkan pada Gambar 3.17.



Gambar 3.17. Lokasi konfigurasi koneksi VPN pada FortiClient

Setelah VPN aktif, sesi administrasi server dilakukan melalui SSH menggunakan MobaXterm. MobaXterm digunakan untuk memudahkan eksekusi perintah instalasi, verifikasi layanan, dan penelusuran log selama proses *upgrade*. Tampilan aplikasi MobaXterm ditunjukkan pada Gambar 3.18.



Gambar 3.18. Aplikasi MobaXterm sebagai alat CLI dan SSH

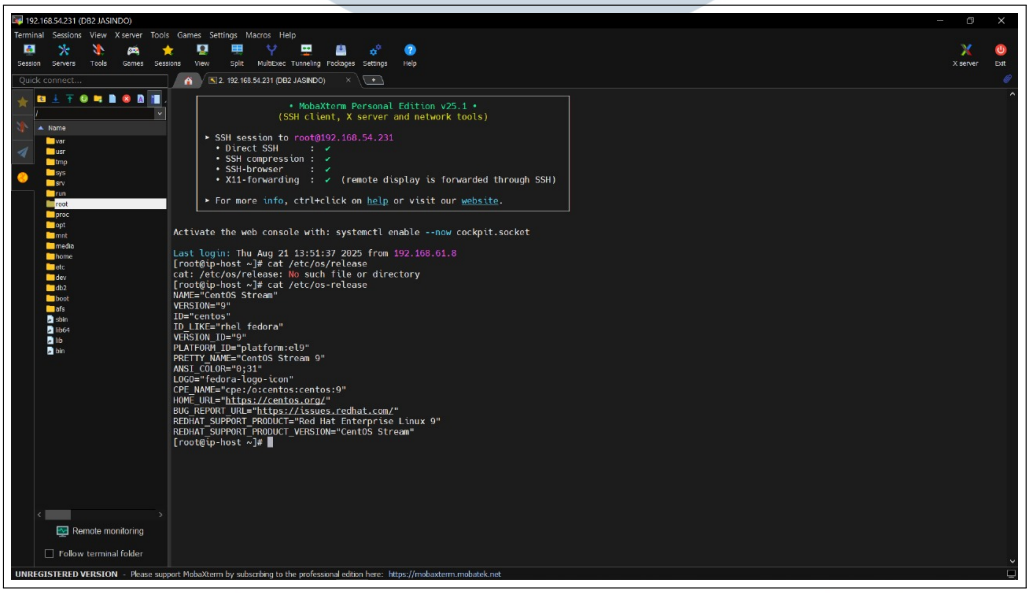
Sebelum memulai instalasi, dilakukan pemeriksaan akses dasar untuk memastikan sesi SSH stabil, identitas sistem sesuai target, serta hak akses operasional tersedia. Pemeriksaan ini juga memastikan server siap menjangkau komponen pendukung yang dibutuhkan pada tahap berikutnya, misalnya host DB2. Contoh perintah akses dan verifikasi awal ditunjukkan pada Kode 3.14.

```
# koneksi ke server uji (dilakukan melalui MobaXterm setelah VPN
    FortiClient aktif)
```

```
ssh <user>@<ip_server>

# verifikasi OS (keluarga Red Hat) dan kernel
cat /etc/redhat-release
cat /etc/os-release
uname -r

# verifikasi identitas host dan waktu
hostnamectl
timedatectl

# verifikasi hak akses operasional untuk instalasi dan
    administrasi layanan
id
sudo -n true
```

Kode 3.14: Akses SSH dan verifikasi awal pada server uji

Hasil verifikasi awal akses, identitas sistem operasi, dan kesiapan operasional sebelum instalasi ditunjukkan pada Gambar 3.19.



Gambar 3.19. Hasil verifikasi awal akses dan identitas server PoC sebelum instalasi

A Instalasi Komponen dan Verifikasi Layanan Dasar

Tahapan instalasi pada lingkungan *proof of concept* (PoC) dilakukan secara berurutan untuk memastikan dependensi antar komponen terpenuhi dan layanan

dasar dapat diverifikasi. Urutan pekerjaan yang dilakukan dirangkum sebagai berikut.

1. Penyiapan *repository* instalasi internal pada IBM Installation Manager, termasuk verifikasi akses *repository* dan pemeriksaan integritas berkas instalasi yang digunakan.
2. Instalasi dan aktivasi layanan DB2 pada server uji hingga instance dapat dijalankan serta menerima koneksi sesuai kebutuhan PoC.
3. Instalasi WebSphere Application Server dan pembuatan profil aplikasi untuk menyiapkan *runtime* yang akan digunakan pada fase berikutnya.
4. Instalasi komponen pendukung integrasi SDI dan *Dispatcher*, serta penyesuaian direktori *connector* dan konfigurasi awal yang diperlukan.
5. Verifikasi layanan dasar dilakukan melalui pemeriksaan status proses atau layanan, pengecekan port, serta validasi konektivitas antar komponen agar jalur integrasi siap digunakan.
6. Setelah prasyarat lingkungan terverifikasi, instalasi ISVG beserta penerapan *fix pack* direncanakan sebagai tahapan lanjutan. Pada pelaksanaan PoC, instalasi ISVG masih berada pada tahap finalisasi instalasi sehingga hasil verifikasi komponen pendukung dan dokumentasi temuan teknis digunakan sebagai dasar tindak lanjut pada fase berikutnya.

B Persiapan DB2 untuk ISVG

Pada fase *proof of concept* (PoC), instalasi DB2 dilakukan melalui terminal agar setiap langkah dapat ditelusuri melalui log instalasi. Installer DB2 secara otomatis membuat berkas log pada direktori `/tmp` dengan pola nama `db2_install.log<PID>`, sehingga hasil instalasi dapat diaudit kembali. Sebelum instalasi dijalankan, pemeriksaan prasyarat dilakukan menggunakan utilitas `db2prereqcheck` untuk memastikan dependensi sistem telah terpenuhi, sebagaimana ditunjukkan pada Kode 3.15.

```
# jalankan dari direktori media instalasi DB2
./db2prereqcheck -v 11.5
```

Kode 3.15: Pemeriksaan prasyarat instalasi DB2 pada server Linux

Proses instalasi DB2 kemudian dijalankan menggunakan skrip `db2_install` agar *file set* DB2 terpasang pada direktori target dan log instalasi terbentuk untuk kebutuhan dokumentasi PoC. Contoh eksekusi instalasi melalui terminal ditunjukkan pada Kode 3.16.

```
# masuk ke direktori media instalasi DB2
cd <db2_media_mount_or_extract_path>

# instalasi DB2 (log instalasi tercatat di /tmp/db2_install.log<
    PID>)
./db2_install

# setelah selesai, periksa log untuk memastikan status instalasi
ls -l /tmp/db2_install.log*
tail -n 50 /tmp/db2_install.log*
```

Kode 3.16: Instalasi DB2 menggunakan db2_install melalui terminal

Pada beberapa kondisi instalasi, komponen terkait dukungan *high availability* dapat memunculkan peringatan pada *file set* tertentu. Pada PoC, peringatan tersebut tetap dicatat sebagai bagian dokumentasi, namun validasi utama difokuskan pada keberhasilan pemasangan *engine* DB2 dan kemampuan menerima koneksi.

Setelah pemasangan *file set* selesai, instance DB2 disiapkan apabila belum tersedia pada server. Karena pada DB2 di Linux instance berada pada level pengguna, pembuatan instance dilakukan menggunakan `db2icrt`, sebagaimana ditunjukkan pada Kode 3.17.

```
# contoh pembuatan akun OS untuk instance owner dan fenced user
sudo useradd -m db2inst1
sudo useradd -m db2fenc1

# pembuatan instance DB2
sudo /opt/ibm/db2/V11.5/instance/db2icrt -u db2fenc1 db2inst1
```

Kode 3.17: Pembuatan instance DB2 (jika instance belum tersedia)

Setelah instance tersedia, layanan DB2 dijalankan dan basis data untuk kebutuhan ISVG dipersiapkan sebagai prasyarat agar proses inisialisasi skema aplikasi dapat dilakukan pada tahap berikutnya. Penyiapan basis data mencakup pembuatan akun layanan skema aplikasi, pembuatan basis data, pembuatan skema, pemberian hak akses, serta uji koneksi dan *query* dasar. Contoh langkah penyiapan tersebut ditunjukkan pada Kode 3.18.

```
# 1) pembuatan akun layanan skema aplikasi (non-interaktif)
sudo useradd -m -r -s /sbin/nologin isvgusr
sudo passwd isvgusr

# 2) start DB2 sebagai instance owner
# gunakan salah satu cara sesuai kebijakan akses pada server
# su - db2inst1 -c "db2start"
db2start

# 3) pembuatan database untuk PoC (penamaan mengikuti standar
    lingkungan)
db2 "create db ISVG_JASINDO using codeset UTF-8 territory ID
    pagesize 32768"
db2 "connect to ISVG_JASINDO"

# 4) pembuatan schema aplikasi
db2 "create schema ISVG authorization isvgusr"

# 5) pemberian hak akses untuk akun layanan
db2 "grant connect on database to user isvgusr"
db2 "grant createtab, bindadd, implicit_schema on database to user
     isvgusr"
db2 "grant dbadm on database to user isvgusr"

# 6) uji koneksi dan query dasar
db2 "connect reset"
db2 "connect to ISVG_JASINDO user isvgusr using ********"
db2 "select current timestamp from sysibm.sysdummy1"
db2 "connect reset"
```

Kode 3.18: Pembuatan akun layanan OS dan penyiapan database DB2 untuk ISVG

Sebagai verifikasi tambahan, validasi instalasi DB2 dapat dijalankan menggunakan `db2val` untuk memastikan instalasi, instance, dan fungsi dasar basis data berada pada kondisi normal. Contoh eksekusi validasi ditunjukkan pada Kode 3.19.

```
/opt/ibm/db2/V11.5/bin/db2val
```

Kode 3.19: Validasi instalasi DB2 menggunakan db2val

Setelah DB2 siap, verifikasi konektivitas dari sisi host aplikasi dilakukan untuk memastikan jalur jaringan menuju port DB2 tersedia sebelum konfigurasi ISVG dijalankan. Pemeriksaan konektivitas jaringan ditunjukkan pada Kode 3.20.

```
# dijalankan pada host aplikasi (server runtime aplikasi)
nc -vz <db2_host> <db2_port>
```

Kode 3.20: Verifikasi konektivitas jaringan dari host aplikasi menuju DB2

C Instalasi ISVG dan Validasi Prasyarat Koneksi DB2

Instalasi ISVG pada fase *proof of concept* (PoC) dipersiapkan menggunakan paket installer berbasis *InstallAnywhere* (`instlinux.bin`) beserta *response files* pada direktori `response_files`. Penyiapan dilakukan dengan memastikan berkas installer dapat dieksekusi dan *response files* untuk platform UNIX tersedia pada direktori kerja, sebagaimana ditunjukkan pada Kode 3.21.

```
# contoh struktur direktori:
# /home/isvg/Downloads/ISVG/
#   - instlinux.bin
#   - response_files/FreshInstallation/unix.zip
#   - response_files/Upgrade/unix.zip

cd /home/isvg/Downloads/ISVG
chmod +x instlinux.bin

# ekstrak response files (pilih sesuai skenario)
mkdir -p /home/isvg/Downloads/ISVG/resp_fresh
unzip -o response_files/FreshInstallation/unix.zip -d /home/isvg/
    Downloads/ISVG/resp_fresh

# isi folder hasil ekstrak umumnya mencakup:
# - installvariables.properties
# - configResponse.properties
# - configResponseCM.properties
ls -lah /home/isvg/Downloads/ISVG/resp_fresh
```

Kode 3.21: Penyiapan media instalasi ISVG dan response files (UNIX)

Setelah *response files* tersedia, instalasi dijalankan dalam mode *silent* agar eksekusi lebih konsisten dan mudah ditelusuri melalui *output log* instalasi. Eksekusi *silent install* dilakukan dengan memanggil `instlinux.bin` dan menunjuk berkas `installvariables.properties`, serta direktori konfigurasi `configResponse.properties` melalui parameter yang relevan, sebagaimana ditunjukkan pada Kode 3.22.

```
# jalankan dari direktori installer
```

```
cd /home/isvg/Downloads/ISVG

# direktori response files hasil ekstrak
export ISVG_RESP_DIR="/home/isvg/Downloads/ISVG/resp_fresh"

# instalasi silent
# catatan: jika response files berada satu folder dengan instlinux.bin,
# parameter -DITIM_CFG_RESP_FILE_DIR dapat dihilangkan
./instlinux.bin -I silent \
  -f "${ISVG_RESP_DIR}/installvariables.properties" \
  -DITIM_CFG_RESP_FILE_DIR="${ISVG_RESP_DIR}" \
  > /var/log/isvg_install_stdout.log 2>&1
```

Kode 3.22: Eksekusi instalasi ISVG secara silent menggunakan instlinux.bin dan response file

Untuk memastikan prasyarat konektivitas basis data tidak menjadi penghambat pada tahap lanjutan, dilakukan verifikasi jalur jaringan dari host aplikasi menuju port DB2 dan uji koneksi dasar DB2 menggunakan kredensial yang telah disiapkan. Pemeriksaan tersebut dilakukan menggunakan perintah pada Kode 3.23.

```
# 1) uji port DB2 dari host aplikasi (ganti host dan port sesuai environment)
nc -vz <db2_host> <db2_port>

# 2) uji koneksi DB2 menggunakan DB2 CLP (jalankan pada host yang memiliki DB2 client/instance)
db2 "connect to <DBNAME> user <db_user> using ********"
db2 "select current timestamp from sysibm.sysdummy1"
db2 "connect reset"
```

Kode 3.23: Validasi konektivitas jaringan dan koneksi dasar DB2 sebelum konfigurasi ISVG

Pada pelaksanaan PoC, apabila installer berhenti pada tahap peluncuran JVM (misalnya muncul pesan kegagalan pembuatan JVM), tahap yang dilakukan adalah mendokumentasikan *output error*, memeriksa variabel lingkungan Java, dan menyiapkan alternatif eksekusi installer agar investigasi dapat dilakukan pada fase berikutnya. Langkah pencatatan dan verifikasi awal tersebut ditunjukkan pada Kode 3.24.

```
# 1) jalankan installer dalam mode console untuk melihat output lebih langsung
```

```
cd /home/isvg/Downloads/ISVG
./instlinux.bin -I console | tee /var/log/isvg_install_console.log

# 2) cek variabel environment yang kadang memengaruhi opsi JVM
printenv | grep -E 'JAVA_TOOL_OPTIONS|_JAVA_OPTIONS|
    JDK_JAVA_OPTIONS' || true

# 3) alternatif: paksa installer menggunakan JVM tertentu (jika
    dibutuhkan dan tersedia)
# catatan: path java disesuaikan dengan JVM yang disiapkan pada
    server
# ./instlinux.bin LAX_VM /opt/java/bin/java -I console | tee /var/
    log/isvg_install_laxvm.log
```

Kode 3.24: Pencatatan error peluncuran installer dan verifikasi lingkungan JVM

D Kendala Stabilisasi Layanan ITIM dan Temuan Teknis pada Fase PoC

Pada fase *proof of concept* (PoC), setelah pemasangan paket dasar dan tahap validasi pasca instalasi dijalankan, ditemukan kendala pada layanan ITIM yang terindikasi dari kemunculan *ITIM servlet error* pada proses *deployment*. Untuk memastikan kendala tidak berasal dari layanan dasar, dilakukan verifikasi ulang terhadap status server WebSphere, status aplikasi yang ter-*deploy*, serta peninjauan log *runtime*.

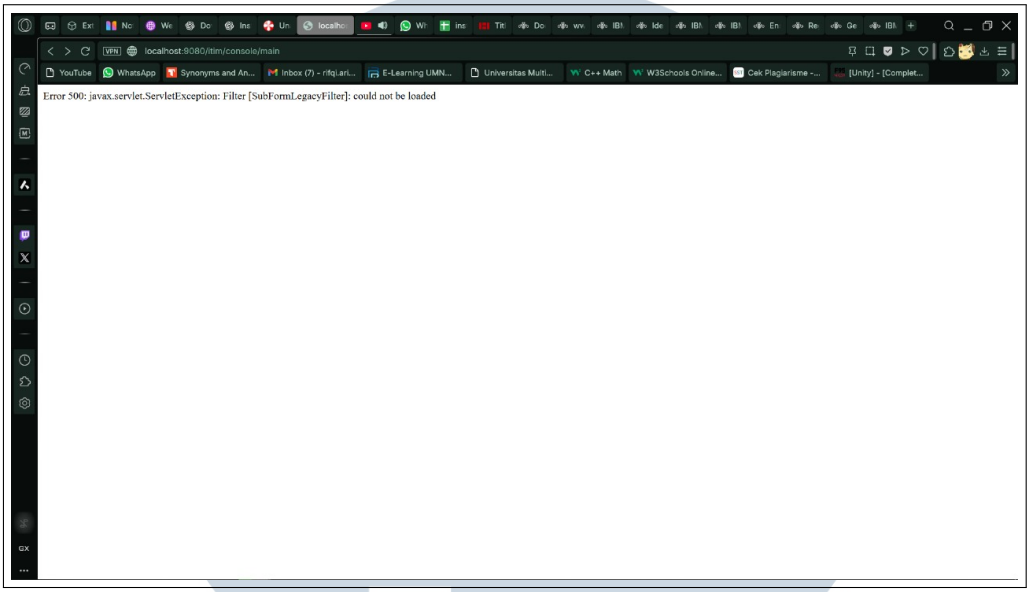
Sebagai upaya mitigasi awal, dilakukan pendekatan bertahap dengan memastikan aplikasi ITIM berada pada kondisi berjalan, membersihkan direktori sementara WebSphere, lalu melakukan *restart* terkontrol. Prosedur mitigasi tersebut dirangkum pada Kode 3.25.

```
# stop dan start WebSphere (kredensial disamarkan)
<was_home>/bin/stopServer.sh server1 -username <was_admin> -
    password ********
rm -rf <was_profile>/wstemp/*
rm -rf <was_profile>/temp/*
rm -rf <was_profile>/tranlog/*
<was_home>/bin/startServer.sh server1
```

Kode 3.25: Restart terkontrol dan pembersihan direktori sementara WebSphere

Hasil verifikasi menunjukkan bahwa mitigasi awal tersebut belum menghasilkan kondisi layanan yang stabil. Indikasi *ITIM servlet error* masih

muncul pada tahap *deployment* pasca instalasi, sebagaimana ditunjukkan pada Gambar 3.20.



Gambar 3.20. Kondisi *ITIM servlet error* pada tahap *deployment* pasca instalasi

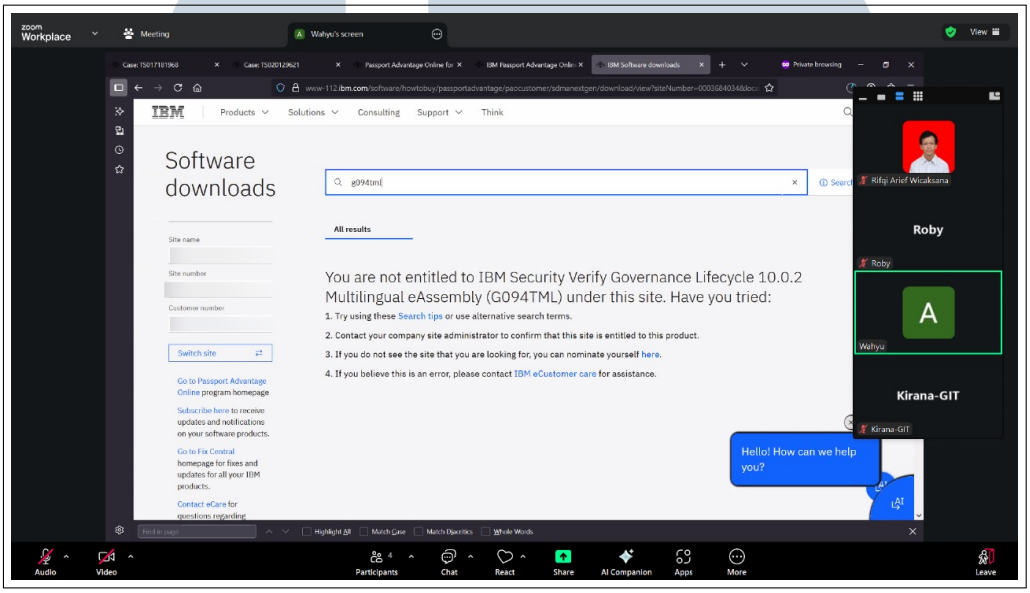
Selain itu, pada beberapa percobaan akses ke URL ITIM juga mengembalikan respons *404 Not Found*. Kondisi ini menunjukkan bahwa layanan aplikasi belum berada pada status siap operasi, sehingga tahapan lanjutan seperti konfigurasi aplikasi dan inisialisasi skema pada DB2 belum dapat dijalankan secara konsisten. Contoh respons *404* ditunjukkan pada Gambar 3.21.

Gambar 3.21. Respons *404 Not Found* pada URL ITIM setelah upaya mitigasi awal

Berdasarkan penelusuran rujukan instalasi dan hasil analisis di lingkungan PoC, penerapan *fix pack* diidentifikasi sebagai kebutuhan untuk menormalkan lapisan aplikasi agar proses stabilisasi layanan dapat dicapai dan tahapan integrasi, termasuk koneksi serta inisialisasi basis data DB2, dapat dilanjutkan pada fase berikutnya. Namun pada pelaksanaan di lapangan, *fix pack* yang tersedia pada *repository* internal belum dapat diterapkan pada instalasi yang sudah ada karena IBM Installation Manager tidak dapat melakukan pembaruan paket pada kondisi

tersebut. Temuan ini mengindikasikan adanya ketidaksesuaian paket instalasi yang terpasang dengan *fix pack* yang disiapkan pada *repository* internal.

Sebagai tindak lanjut, dilakukan penelusuran paket instalasi yang kompatibel melalui akses akun partner agar paket dasar dan *fix pack* berada pada level yang selaras. Dokumentasi penelusuran paket kompatibel ditunjukkan pada Gambar 3.22.



Gambar 3.22. Penelusuran paket ISVG yang kompatibel melalui akun partner

Dengan demikian, fase PoC menghasilkan dokumentasi kendala, bukti kondisi layanan, serta kebutuhan penyelarasan paket dan *fix pack* sebagai dasar pelaksanaan tahap lanjutan.

3.4 Kendala dan Solusi yang Ditemukan

Selama pelaksanaan magang pada kegiatan pemeliharaan IBM Security Identity Manager (ISIM) dan *proof of concept* IBM Security Verify Governance (ISVG) di lingkungan PT Asuransi Jasindo, beberapa kendala utama ditemukan yang memengaruhi alur kerja. Kendala tersebut adalah sebagai berikut.

1. Kendala terjadi karena dokumentasi internal mengenai IBM ISVG belum disusun secara ringkas dan terstruktur, sehingga pemahaman awal terhadap istilah, komponen, alur persetujuan, tata kelola identitas, integrasi *target system*, dan pola rekonsiliasi memerlukan penelusuran rujukan berulang serta pengujian ulang di lingkungan uji.

2. Proses instalasi dan pembaruan komponen bersifat ketat terhadap versi, urutan langkah, dan prasyarat. Ketidaksesuaian pada salah satu tahapan dapat menyebabkan proses instalasi gagal dan lingkungan menjadi tidak stabil sehingga diperlukan pengulangan dari tahap sebelumnya.

3. Pada beberapa kasus, pemilihan paket (misalnya varian *installer*, *fix pack*, atau paket *standalone*) tidak mudah dipastikan kebenarannya karena katalog publik yang dapat ditelusuri secara bebas terbatas. Akibatnya, validasi perlu dilakukan melalui penelusuran dokumen vendor, catatan internal, atau eskalasi dukungan.

Untuk mengatasi kendala-kendala tersebut, beberapa solusi dan pendekatan diterapkan sebagai berikut.

1. Menyusun dokumentasi internal IBM ISVG yang merangkum arsitektur komponen, alur kerja utama, langkah instalasi, serta peta kompatibilitas versi dan *fix pack*, dan perusahaan perlu memastikan ketersediaan dukungan teknis resmi (misalnya melalui support IBM atau principal) agar proses validasi, penanganan *error*, dan pemenuhan paket dapat dilakukan lebih cepat dan konsisten.

2. Pelaksanaan instalasi berbasis *checklist* dan kontrol versi
   Setiap eksekusi instalasi dilakukan berdasarkan *checklist* prasyarat, catatan versi, dan urutan langkah yang terdokumentasi. Ketika terjadi kegagalan, perubahan diisolasi per tahap, prasyarat diverifikasi ulang, lalu instalasi diulang pada kondisi lingkungan yang sudah dipastikan bersih dan konsisten.

3. Penyusunan katalog internal paket dan verifikasi lintas sumber
   Ringkasan internal disusun untuk memuat nama paket, versi, dependensi, dan kecocokan komponen. Verifikasi dilakukan dengan membandingkan beberapa sumber, seperti dokumen vendor, catatan implementasi, dan hasil uji pada lingkungan terkontrol, kemudian temuan dicatat sebagai rujukan untuk proses berikutnya.