

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama magang di PT Asuransi Sinarmas, mahasiswa magang ditempatkan dalam tim yang bertanggung jawab terhadap pengelolaan sistem dan database terkait klaim asuransi. Alur kerja dimulai dengan memahami sistem Pega yang digunakan dalam pengolahan data asuransi. Lalu menganalisa data AI untuk mengetahui tingkat akurasi dari AI yang dimiliki sekarang. Mahasiswa magang berkoordinasi dengan tim pengembang untuk memodifikasi atau mengembangkan AI yang sudah ada agar dapat menjadi lebih akurat.

Koordinasi dilakukan melalui beberapa tahap:

1. Pengenalan Sistem :

Memahami struktur dan alur kerja dalam Pega, termasuk cara kerja fitur seperti *Live UI* dan *Report Definition*.

2. Pengembangan dan Optimalisasi Pega :

Memodifikasi sistem Pega untuk meningkatkan fungsionalitas, seperti menambahkan validasi input pada form agar data kosong tidak dapat tersimpan, serta menyesuaikan format hasil ekspor laporan.

3. Analisis Perbandingan AI :

Menganalisis data klaim untuk mengevaluasi tingkat akurasi AI internal dan membandingkannya dengan Gemini AI guna menentukan keunggulan masing-masing sistem.

4. Pelaporan dan Evaluasi :

Melaporkan hasil pengembangan sistem Pega serta analisis AI kepada supervisor dan tim terkait untuk bahan evaluasi lebih lanjut.

3.2 Tugas yang Dilakukan

Sistem yang digunakan untuk pengolahan data dan analisis AI adalah Google Spreadsheets dan Postman, sedangkan untuk pengembangan web digunakan platform Pegasystems.

Tabel 3.1. Tabel Pekerjaan Selama Magang (Minggu 1-19)

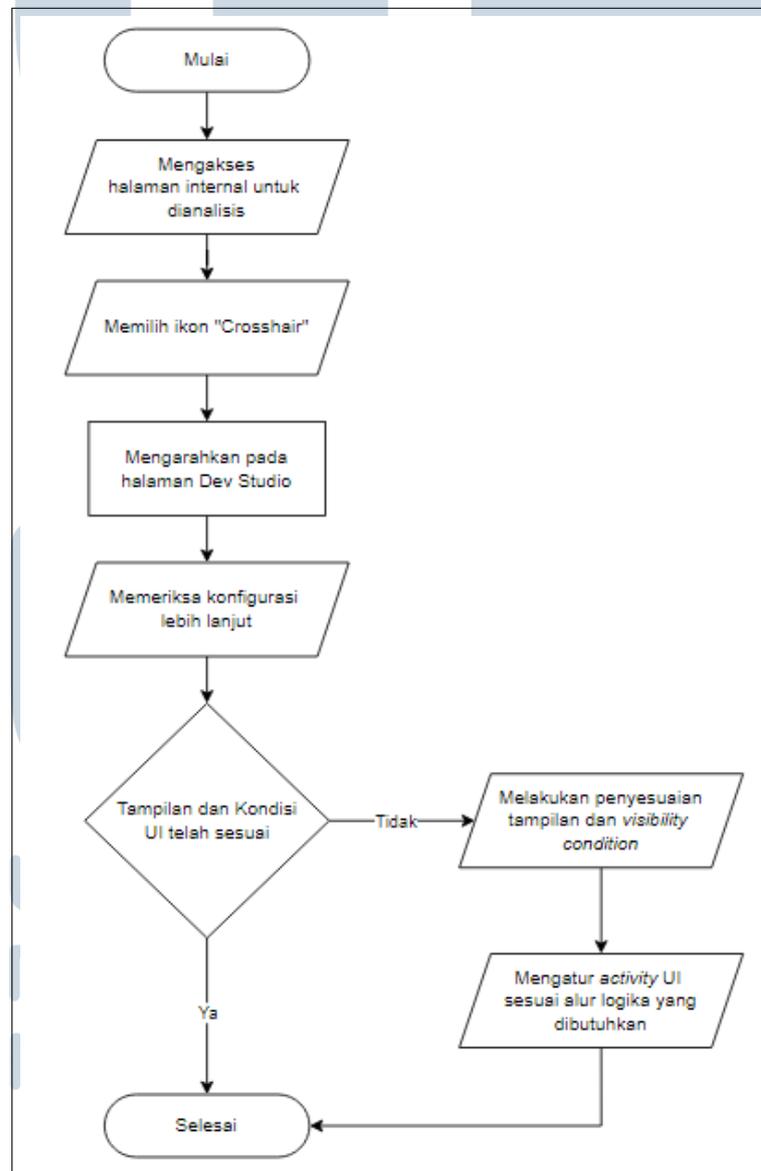
Minggu Ke -	Pekerjaan yang dilakukan
1	Mempelajari Flowchart bagaimana alur kerja pembuatan klaim di PT. Asuransi Sinarmas
2	Mempelajari sistem Pega, dari activities yang digunakan, tipe-tipe cara penarikan data, dan juga cara membuat layout
3	Pembuatan Form untuk penginputan data klaim serta dengan keamanannya untuk menghindari data kosong
4	Mempelajari database klaim kendaraan, serta erd, menentukan Primary Key dan Foreign Key
5	Mempelajari bagaimana cara tim yang bertugas untuk membuat laporan menarik data dari database dan export ke google sheet
6	Membuat Query SQL yang lebih baik dan efisien untuk pengolahan dan penarikan data
7, 8, 9, & 10	Menentukan rentang waktu tarikan data, mempelajari cara data AI menarik data surveyor, menentukan patokan perbandingan data surveyor dan AI, Cara penentuan tipe panel, Ketentuan panel match dan tidak match
11 & 12	Menambahkan Ketentuan match treatment antara surveyor dan AI
13	Melakukan penarikan data AI gemini, dengan penarikan data melalui postman dan menentukan prompt yang baik dan efisien.
14	Membuat perbandingan AI Gemini dan Surveyor dengan ketentuan dan cara yang sama seperti sebelumnya
15 & 16	Melihat dan membandingkan hasil analisa surveyor dengan AI internal dan surveyor dengan AI gemini.
17, 18, & 19	APP Testing untuk mengetahui error dan apa yang dapat di improve untuk APP CLAIMQ.

3.3 Uraian Pelaksanaan Kerja

3.3.1 Pega Systems

A Memahami struktur sistem Pega Seperti Live UI

Proses penyesuaian dan analisis antarmuka pengguna (UI) dalam sistem Pega dimulai dengan mengakses halaman internal aplikasi yang ingin dianalisis. Langkah ini merupakan titik awal untuk mengidentifikasi elemen-elemen UI yang perlu disesuaikan.



Gambar 3.1. Flowchart Live UI

Setelah halaman aplikasi terbuka, pengguna memulai proses dengan memilih ikon *Crosshair* yang tersedia di panel *Live UI*. Ikon ini digunakan untuk memilih secara langsung elemen UI yang sedang tampil di layar. Fungsinya adalah untuk mempermudah pengguna dalam menelusuri struktur serta properti dari komponen yang dipilih, tanpa harus mencarinya secara manual melalui daftar rule.

Setelah ikon *Crosshair* digunakan untuk memilih komponen, sistem secara otomatis akan membuka halaman *Dev Studio*. *Dev Studio* merupakan lingkungan pengembangan di Pega yang menampilkan detail konfigurasi dari elemen yang dipilih, seperti rule yang digunakan, properti yang terkait, serta hubungan antar elemen lainnya.

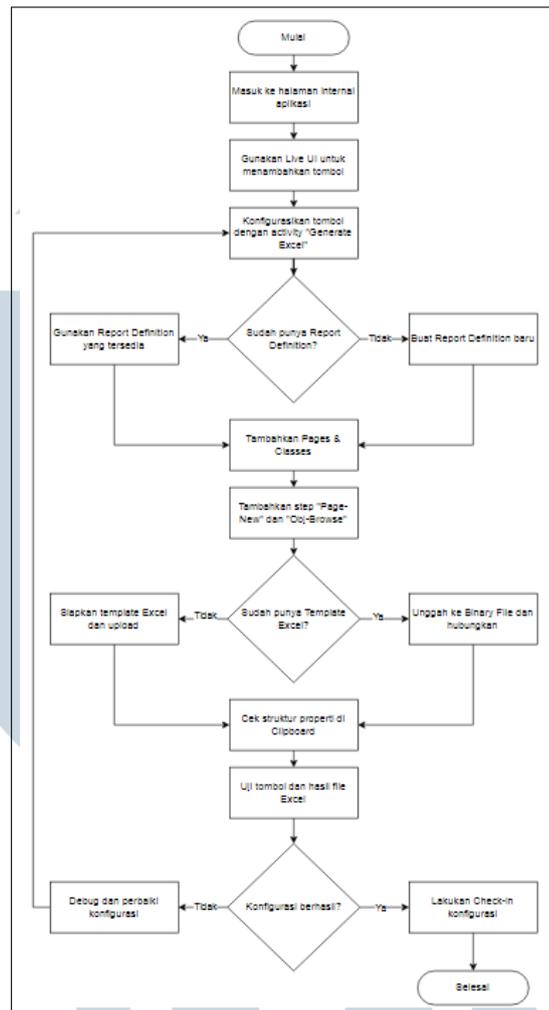
Jika dari hasil pemeriksaan ditemukan bahwa tampilan atau logika dari elemen tersebut belum sesuai dengan yang diharapkan, maka dilakukan penyesuaian. Penyesuaian ini dapat mencakup pengaturan tampilan visual seperti layout, warna, dan ukuran, maupun pengaturan *visibility condition*, yaitu logika yang menentukan kapan suatu elemen akan ditampilkan atau disembunyikan berdasarkan kondisi tertentu.

Setelah tampilan dan kondisi visibilitas disesuaikan, pengguna juga dapat mengatur aktivitas yang berhubungan dengan elemen tersebut. Contohnya, jika pengguna mengklik sebuah tombol, maka sistem harus menjalankan proses tertentu sesuai kebutuhan. Pengaturan ini dapat dilakukan melalui *Activity*, *Data Transform*, atau rule lainnya yang relevan.

Jika semua pengaturan sudah dilakukan dan hasilnya telah sesuai baik dari sisi tampilan maupun logika interaksi, maka proses konfigurasi UI dianggap selesai.

B Membuat Tombol Generate Excel File

Pembuatan tombol *Generate Excel File* dilakukan sebagai bagian dari inisiatif untuk mempermudah proses ekstraksi data klaim dari sistem Web Pega. Tombol ini difungsikan untuk menarik data yang mencakup daftar klaim kendaraan bermotor beserta informasi operator yang menangani masing-masing kasus tersebut. Proses ini memungkinkan pengguna untuk memperoleh data dalam format spreadsheet secara otomatis, sehingga memudahkan analisis dan pelaporan. Fitur ini juga mendukung efisiensi kerja tim operasional dalam memantau status klaim. Alur pembuatan tombol ini ditunjukkan pada flowchart berikut :



Gambar 3.2. Flowchart Button Generate Excel

Flowchart ini menggambarkan alur konfigurasi tombol *Generate Excel* di aplikasi internal berbasis Pega. Proses dimulai dengan *mengakses halaman internal aplikasi* melalui Dev Studio, di mana pengguna akan menggunakan fitur *Live UI* untuk menambahkan sebuah tombol baru ke antarmuka pengguna. Tombol ini dikonfigurasi untuk menjalankan sebuah *activity* bernama *Generate Excel*, yang akan memicu proses pengambilan dan ekspor data klaim ke file Excel.

Selanjutnya, sistem akan mengecek apakah *Report Definition* yang dibutuhkan untuk mengambil data sudah tersedia. Jika sudah ada, pengguna dapat langsung menggunakannya. Namun jika belum tersedia, pengguna perlu membuat *Report Definition* baru terlebih dahulu. Setelah itu, konfigurasi dilanjutkan dengan menambahkan entri di *Pages & Classes* agar *activity* dapat mengakses struktur data yang sesuai. Tahap berikutnya adalah menambahkan langkah-langkah ke dalam *activity*, termasuk method *Page-New* untuk membuat halaman baru, dan *Obj-*

Browse untuk mengambil data dari class yang ditentukan.

Setelah pengambilan data disiapkan, sistem akan mengecek apakah template file Excel sudah tersedia. Jika belum, pengguna harus menyiapkan dan *mengunggah template* tersebut terlebih dahulu. Jika sudah tersedia, template dapat langsung diunggah ke *Binary File* dan dikaitkan dengan *activity* melalui pengaturan seperti *TemplateRFB* dan *FSFileName*. Template ini berisi format dan struktur data yang akan diisi saat file diunduh oleh pengguna. Struktur data dalam template harus sesuai dengan yang muncul di *Clipboard*, seperti referensi properti menggunakan *.pxResults().pyNamaProperti*.

Setelah semua langkah konfigurasi selesai, pengguna dapat *menguji tombol Generate Excel* dan memeriksa apakah file hasil ekspor sudah sesuai harapan. Jika hasilnya sesuai, maka konfigurasi dianggap berhasil dan pengguna dapat melanjutkan untuk melakukan *Check-in* konfigurasi ke dalam sistem. Namun jika hasilnya belum sesuai, maka perlu dilakukan *debug dan perbaikan konfigurasi* sebelum proses bisa diselesaikan dengan benar.

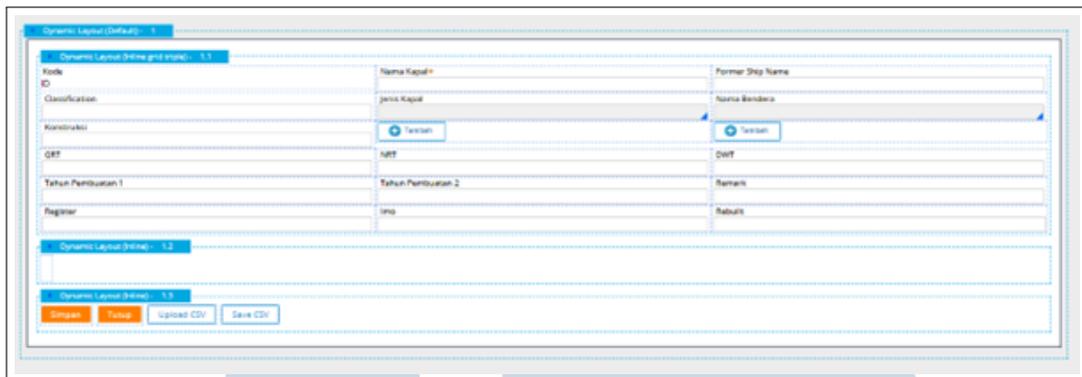
Flowchart ini menampilkan beberapa cabang keputusan, yaitu untuk memeriksa keberadaan Report Definition, ketersediaan Template Excel, dan keberhasilan hasil akhir. Hal ini membuat alur tidak hanya lurus, namun mencerminkan proses teknis yang adaptif terhadap kondisi sistem saat konfigurasi dijalankan.

C Membuat Form

Berikut merupakan langkah-langkah pembuatan formulir klaim kapal sebagai bagian dari proses digitalisasi layanan klaim di PT Asuransi Sinar Mas. Formulir ini dibuat untuk memudahkan penginputan data klaim

1. Menyiapkan Komponen Formulir

Buka halaman aplikasi menggunakan fitur *Launch User* untuk meninjau contoh formulir yang akan dibuat. Gunakan *Live UI*, klik ikon *Aim* pada komponen kolom yang ingin dimodifikasi. Masuk ke *Dev Studio*, pilih menu *Data Capture*, seret elemen *Text Input*, lalu konfigurasi properti sebagai berikut: Nama variabel (Property), Label tampilan, Status editable, dan action: *Post Value* saat *Change*

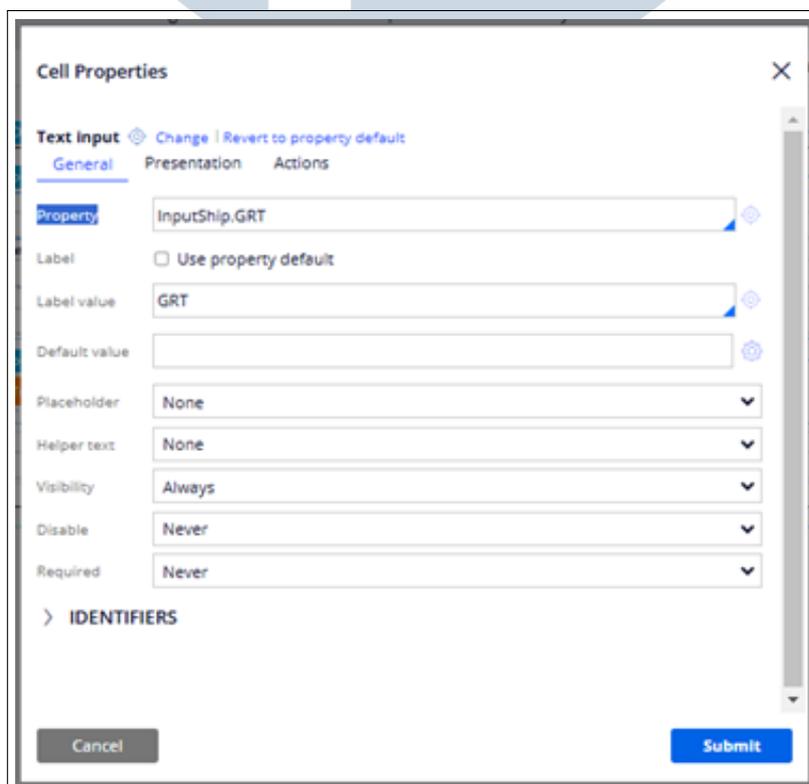


Gambar 3.3. Tampilan awal konfigurasi form

Sumber: Web Perusahaan

2. Mengonfigurasi Tombol Simpan

Klik dua kali pada tombol Simpan yang telah tersedia di antarmuka. Pada bagian konfigurasi action, atur agar perintah *Click* menjalankan activity *SaveShip_Act*.

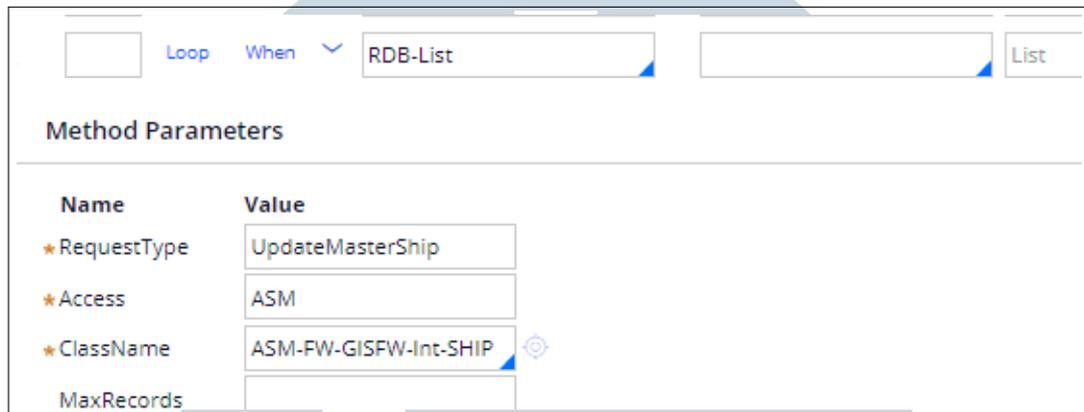


Gambar 3.4. Pengaturan tombol simpan pada form

Sumber: Web Perusahaan

3. Menghubungkan dengan Kelas Data

Gunakan metode *RDB-List* untuk mencari dan mengakses class (kelas data) yang diperlukan. Pastikan isi dari *Browse* telah sesuai dengan class tujuan.

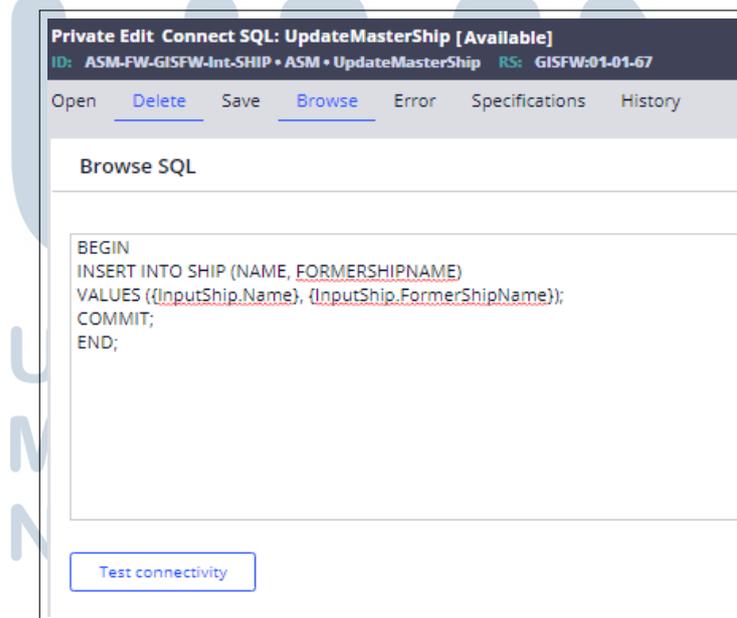


Gambar 3.5. Pengaturan pemanggilan kelas data

Sumber: Web Perusahaan

4. Menyusun Input untuk Tabel

Pada bagian *UpdateMasterShip*, masukkan nama kolom (header) dari tabel target. Isi bagian *Values* dengan nama properti (property cell) yang sebelumnya telah dibuat dan dikonfigurasi.

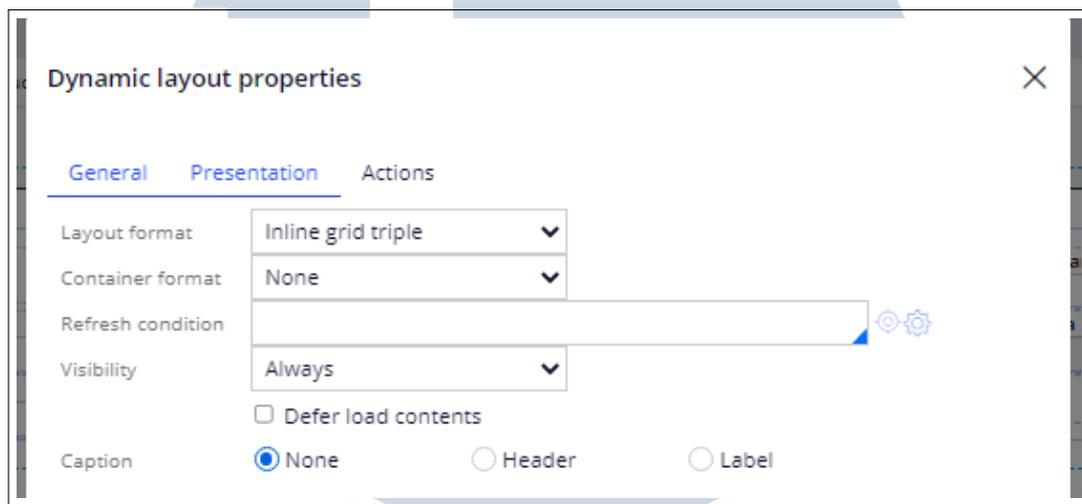


Gambar 3.6. Pengaturan input data ke dalam tabel

Sumber: Web Perusahaan

5. Mengatur Tampilan Tata Letak Formulir

Klik dua kali pada komponen *Dynamic Layout* untuk membuka pengaturannya. Ubah nilai pada *Layout Format* menjadi *Inline grid triple*, dan pada tab *Presentation*, atur *Alignment* menjadi *Auto*. Simpan perubahan dengan mengklik *Save* untuk melakukan pengujian tampilan secara lokal.



Gambar 3.7. Pengaturan format dan perataan layout

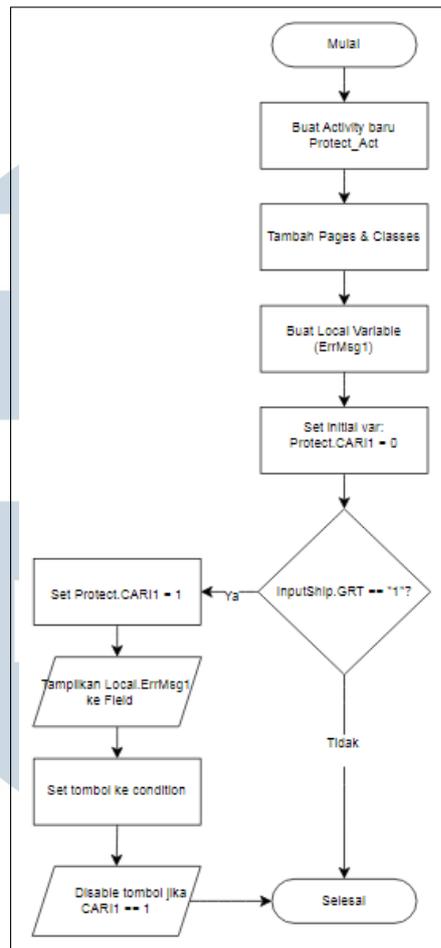
Sumber: Web Perusahaan

D Membuat Proteksi Berdasarkan Nilai Variabel Tertentu

Proteksi berdasarkan nilai tertentu ini digunakan untuk menghindari data-data yang tidak seharusnya, seperti contohnya apabila nama diisi dengan angka dan hal-hal lainnya, yang dapat merusak database karena menjadi data kotor dimana akan terdapat *char* dan *int* di column yang sama.

Langkah pertama adalah membuat *Activity* baru dengan nama *Protect_Act* melalui proses *Save As* dari *activity* simpan yang asli. Setelah itu, dilakukan konfigurasi pada bagian *Pages & Classes* dengan menambahkan entri seperti *Protect* untuk nama halaman dan *ASM-FW-GISW-Data-Search* untuk kelasnya.

Selanjutnya, dibuat *Local Variable* dengan nama *ErrMsg1* di tab *Parameters*. Variabel ini nantinya akan digunakan untuk menampilkan pesan kesalahan pada field input tertentu. Setelah itu, nilai awal dari variabel proteksi diinisialisasi dengan perintah *Protect.CAR11 = 0*.



Gambar 3.8. Flowchart Button Generate Excel

Flow kemudian memasuki percabangan kondisi (decision), yang memeriksa apakah nilai dari properti *InputShip.GRT* adalah "1". Jika kondisi ini terpenuhi, maka nilai *Protect.CAR11* diubah menjadi 1 menggunakan method *Property-Set*, yang menandakan bahwa proteksi aktif.

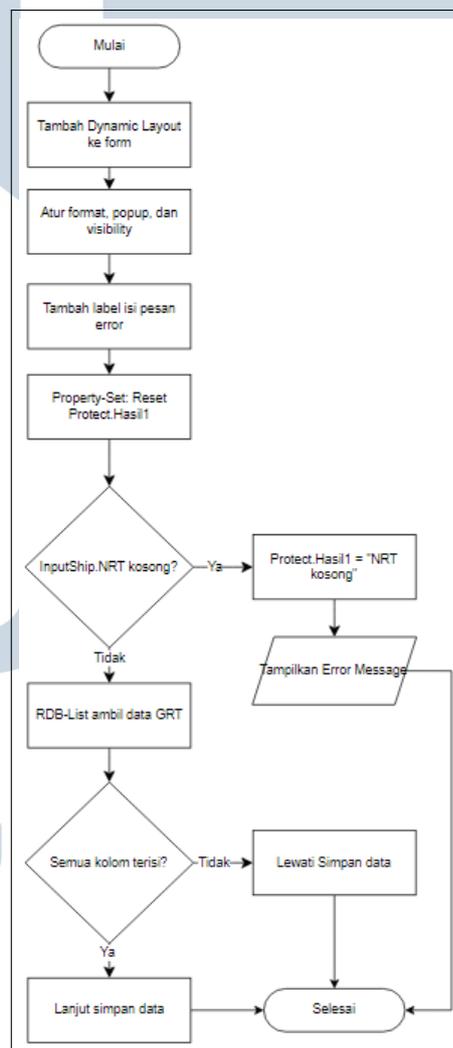
Setelah proteksi aktif, pesan kesalahan ditampilkan menggunakan method *Property-Set-Messages*, dengan mengambil nilai dari *Local.ErrMsg1* dan menampilkannya pada field yang relevan. Setelah itu, tombol disesuaikan agar dapat dikendalikan dengan kondisi tertentu, yakni hanya aktif jika *Protect.CAR11* tidak sama dengan 1.

Jika *Protect.CAR11 == 1*, maka tombol akan dinonaktifkan menggunakan ekspresi kondisi pada properti *Disable*. Proses kemudian berakhir pada node *Selesai*.

E Membuat Deteksi Jika Field Belum Terisi Semua

Proteksi field kosong ini dibuat untuk menghindari, data kosong yang terupload ke dalam database, seperti lupa mengisi nama, nomor klaim, dan lain-lainnya.

Proses dimulai dengan menambahkan *Dynamic Layout* ke dalam form. Layout ini digunakan sebagai wadah untuk menampilkan pesan kesalahan apabila validasi input gagal. Setelah layout ditambahkan, dilakukan pengaturan properti seperti *container format*, jenis *popup* (misalnya *Information* atau *Error*), dan *visibility* yang diubah menjadi *Condition (expression)* dengan kondisi *Protect.Hasill != ""*. Artinya, layout hanya akan muncul jika nilai *Protect.Hasill* tidak kosong.



Gambar 3.9. Flowchart Deteksi Field Belum Terisi

Setelah itu, sebuah *label* ditambahkan ke dalam *Dynamic Layout* dan diatur agar menampilkan isi dari variabel *Protect.Hasill*. Variabel ini akan menyimpan pesan kesalahan yang akan ditampilkan ke pengguna.

Pada tahap berikutnya, sistem mereset terlebih dahulu nilai dari *Protect.Hasill* menggunakan *Property-Set* agar nilai lama tidak terbawa ke validasi yang baru. Kemudian, sistem memeriksa apakah kolom *InputShip.NRT* kosong. Jika kosong, maka *Protect.Hasill* diisi dengan pesan "NRT kosong" dan layout error akan ditampilkan. Jika tidak kosong, sistem melanjutkan proses dengan mengambil data *GRT* menggunakan *RDB-List*.

Data yang berhasil diambil akan dicek lagi apakah semua kolom telah terisi. Jika ada kolom yang masih kosong, maka sistem melewati proses simpan dan langsung mengakhiri proses. Namun jika semua kolom terisi dengan benar, sistem melanjutkan ke proses penyimpanan data.

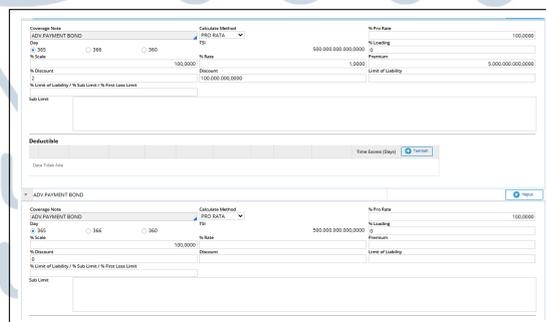
F Membuat Pesan Error dengan *Page-Set-Message*

Error message dibuat untuk memberitahu pengguna mengenai penyebab kesalahan, seperti muncul pesan error bahwa nama belum terisi.

1. Konfigurasi Tombol dan Akses Activity

Pada UI yang ingin ditambahkan tombol, buat tombol melalui *Actions* → *Click* → *Refresh* → *Create new action* (atau *Save As*).

Untuk memanggil activity dari class lain, gunakan sintaks: *Call NAMACLASS.NAMAACTIVITY*. Tambahkan entri *Pages & Classes* dengan page name dan class. Contohnya: *pyWorkPage* diisi dengan class tempat tombol berada.

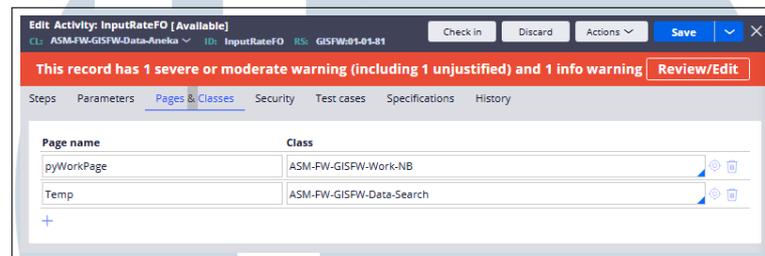


Gambar 3.10. Penambahan tombol dan pemanggilan activity

Sumber: Web Perusahaan

2. Menyiapkan Parameter Lokal

Tambahkan local parameter untuk menentukan jenis error yang akan ditangani, seperti validasi kosong, format tidak sesuai, atau nilai di luar batas. Selain itu, pastikan juga untuk menentukan lokasi field tempat pesan error akan ditampilkan agar pengguna langsung mengetahui letak kesalahan *input*.

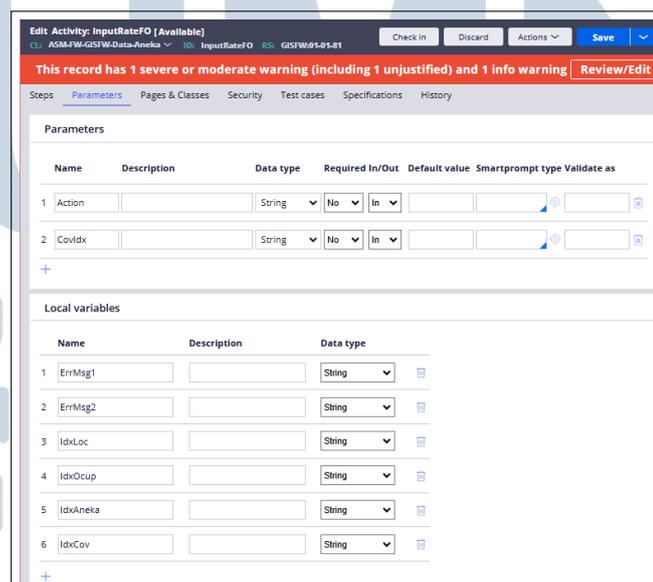


Gambar 3.11. Pengaturan parameter untuk error message

Sumber: Web Perusahaan

3. Inisialisasi Nilai Default

Gunakan step pertama untuk mengatur nilai awal variabel-variabel yang akan digunakan selama proses validasi. Inisialisasi ini penting agar setiap parameter memiliki nilai *default* yang konsisten, sehingga proses validasi dapat berjalan secara terstruktur dan menghindari *error* akibat nilai yang tidak terdefinisi.

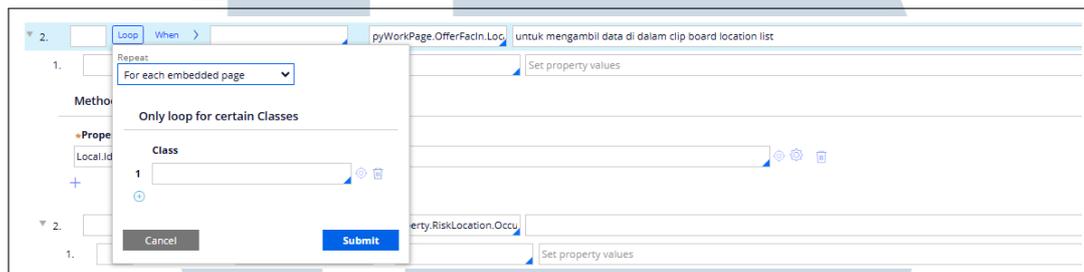


Gambar 3.12. Inisialisasi nilai awal variabel

Sumber: Web Perusahaan

4. Menangkap Indeks Array

Gunakan loop untuk menelusuri data array, misalnya *LocationList(1)*, agar setiap elemen dalam list dapat diproses satu per satu. *Step Page* diisi dengan lokasi array yang sesuai, yaitu *pyWorkPage.OfferFacIn.LocationList*, untuk memastikan konteks data yang diakses berada pada level elemen array yang tepat.

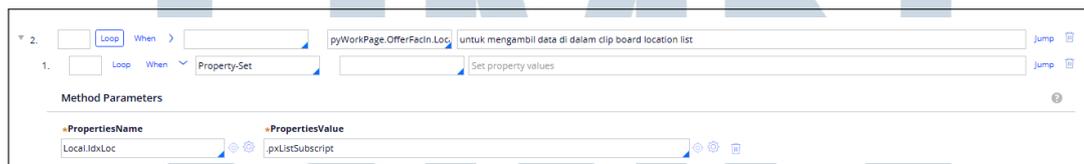


Gambar 3.13. Menyimpan indeks array menggunakan *pxListSubscript*

Sumber: Web Perusahaan

5. Menelusuri Sub-Array Tambahan

Klik *Add Child* lalu tetapkan *Local.idxLoc* → *.pxListSubscript* untuk menyimpan indeks array yang sedang diproses. Langkah ini memungkinkan kita untuk melacak posisi elemen dalam array saat iterasi berlangsung. Lanjutkan loop untuk menelusuri struktur array lebih dalam



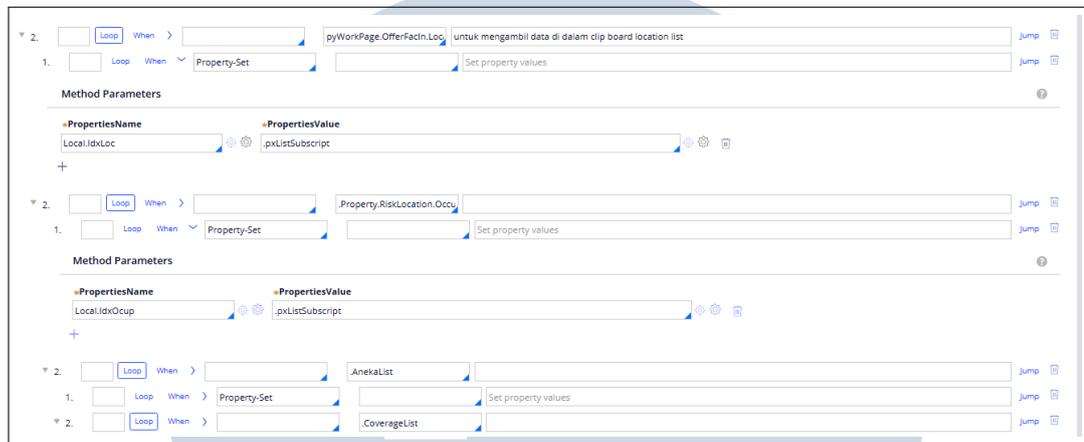
Gambar 3.14. Loop lanjutan untuk *OccupationList*

Sumber: Web Perusahaan

6. Melanjutkan Looping Struktur Data

.Property.RiskLocation.OccupationList.Data ini bisa dicek melalui Clipboard karena *OccupationList* merupakan array. Lanjutkan perulangan untuk menjangkau array berikutnya jika data bersifat *nested* (bertumpuk), misalnya ketika setiap elemen dalam *OccupationList* memiliki sublist atau struktur data lanjutan. Gunakan pendekatan yang sama seperti sebelumnya, yaitu dengan

menentukan *Step Page* yang sesuai dan menyimpan indeks dengan *Local variable*, agar proses iterasi tetap terstruktur dan mudah ditelusuri.

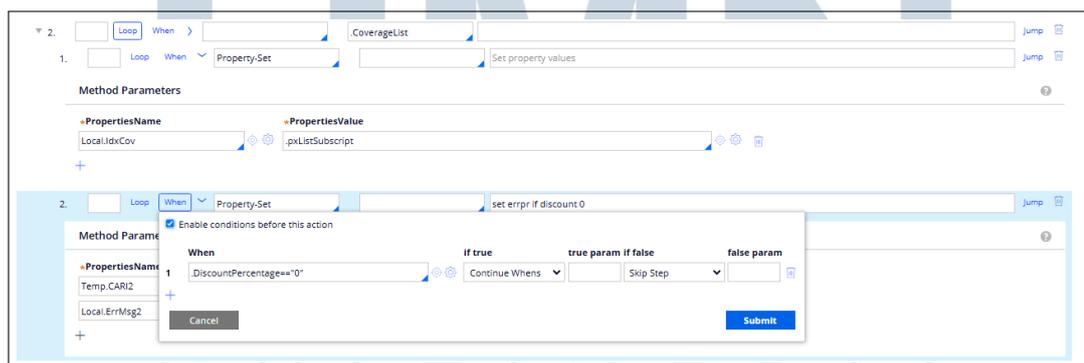


Gambar 3.15. Langkah lanjutan dalam loop nested

Sumber: Web Perusahaan

7. Menentukan Field yang Divalidasi

Gunakan *Property-Set* untuk menetapkan kondisi validasi, misalnya kolom *DiscountPercentage* tidak boleh bernilai 0. Langkah ini bertujuan untuk menyimpan status atau informasi validasi yang akan digunakan pada langkah selanjutnya, misalnya untuk menampilkan pesan error atau menghentikan proses jika data tidak sesuai.



Gambar 3.16. Validasi isi kolom tertentu, misalnya Discount

Sumber: Web Perusahaan

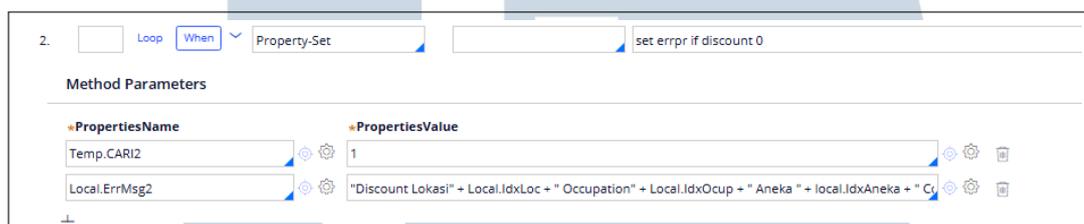
8. Menyusun Pesan Error Dinamis

Jika kondisi validasi terpenuhi, isi *Temp.CARI2* dengan nilai 1 sebagai penanda bahwa telah terjadi pelanggaran aturan. Selanjutnya, susun pesan

error secara dinamis dengan format sebagai berikut:

*"Discount Lokasi " + Local.IdxLoc + " Occupation " +
Local.IdxOcup + " Aneka " + Local.IdxAneka + " Coverage "
+ Local.IdxCov + " Tidak Boleh 0"*

Format pesan ini memungkinkan pelacakan lokasi kesalahan dalam struktur data yang kompleks atau bertingkat (nested).



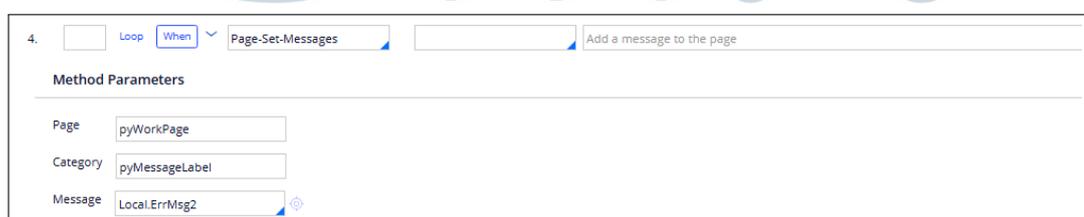
Gambar 3.17. Konstruksi pesan error berdasarkan posisi array

Sumber: Web Perusahaan

9. Menampilkan Pesan Error

Gunakan *Page-Set-Message* untuk menetapkan pesan error ke halaman tertentu. Langkah ini penting agar pesan kesalahan dapat ditampilkan pada UI yang sesuai dengan konteks data.

- Tentukan *page* tujuan, seperti *pyWorkPage*, agar pesan disimpan dan ditampilkan pada level kerja utama.
- Tentukan isi pesan, misalnya menggunakan variabel *Local.ErrMsg2* yang telah disusun secara dinamis sebelumnya.

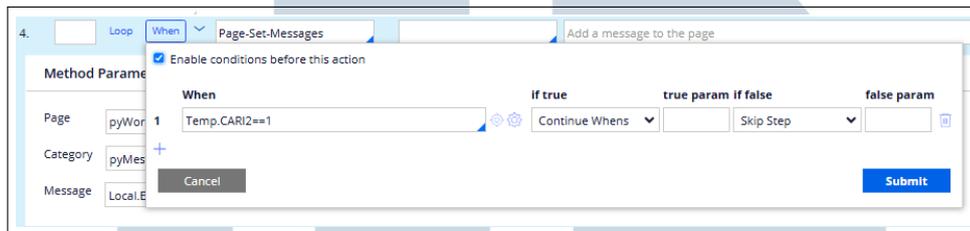


Gambar 3.18. Menampilkan error dengan *Page-Set-Message*

Sumber: Web Perusahaan

10. Menentukan Kondisi Pemicu Pesan

Pesan akan ditampilkan jika kondisi $Temp.CARI2 = 1$ terpenuhi. Nilai ini berfungsi sebagai indikator bahwa setidaknya satu validasi telah gagal selama proses pemeriksaan data.



Gambar 3.19. Pemicu tampilan pesan berdasarkan flag

Sumber: Web Perusahaan

3.3.2 Oracle SQL dan Google Sheets

A Evaluasi hasil AI, perbandingan AI Gemini melalui Postman dengan AI Perusahaan

1. Query Pengambilan Gambar Kendaraan :

Query ini digunakan untuk mengambil dokumen gambar (foto kerusakan) terbaru dari klaim kendaraan yang sudah dinyatakan selesai (status *Resolved-Completed*). Dokumen yang dipilih terbatas pada jenis foto kerusakan atau foto full kerusakan, dan hanya diambil dari klaim dalam periode tertentu.

Cuplikan kode SQL dapat dilihat pada Lampiran.

2. Logika CTE (*Common Table Expression*) :

Untuk mengambil hanya satu dokumen gambar terbaru per klaim, digunakan pendekatan *Common Table Expression* (CTE) dengan fungsi *ROW_NUMBER()*. Logika ini memungkinkan pemfilteran data untuk memastikan hanya dokumen terakhir yang diproses per klaim yang ditampilkan.

Dalam query ini juga dilakukan penggabungan (*JOIN*) ke berbagai tabel lain untuk mendapatkan informasi terkait kendaraan (seperti merk dan tipe), serta dokumen gambar dalam format BLOB (*binary large object*).

5. Pencatatan ke Google Sheets :

Hasil dari AI kemudian dimasukkan ke dalam Spreadsheets. Data ini dikompilasi bersama hasil penilaian manual oleh surveyor untuk kemudian dibandingkan menggunakan fitur *Pivot Table*.

BUMPER DEPAN	-	GANTI	2502240245881800	BUMPER DEPAN	-	30	PERBAIKAN
COVER FOG LAMP	KIRI	GANTI	2502240245881808	COVER FOG LAMP	KIRI	70	PERBAIKAN
PINTU BELAKANG	KANAN	PERBAIKAN	2502240245881807	PINTU BELAKANG	KANAN	40	PERBAIKAN
TRIPLANG / ROCKER PANEL	KANAN	PERBAIKAN	2502240245881806	-	-	-	-
-	-	-	2502240245881805	PANEL BAWAH PINTU PEN	-	20	PERBAIKAN
-	-	-	2502240245881804	DEFLEKTOR BUMPER	-	20	PERBAIKAN
-	-	-	2502240245881803	COVER BUMPER	-	25	PERBAIKAN
-	-	-	2502240245881802	FOGLAMP	-	80	GANTI
-	-	-	2502240245881801	LIST CHROME COVER FOX	-	20	PERBAIKAN
-	-	-	-	COVER FOG LAMP SET	-	80	GANTI
-	-	-	-	FOGLAMP KIRI	-	20	PERBAIKAN
-	-	-	-	QUARTER PANEL	KANAN	25	PERBAIKAN
-	-	-	-	PANEL BAWAH PINTU PEN KANAN	-	30	PERBAIKAN
BUMPER DEPAN	-	PERBAIKAN	2502240245835000	BUMPER DEPAN	-	20	PERBAIKAN

Gambar 3.22. Tabel sheets berisi panel kerusakan

Sumber: Google Sheets Perusahaan

6. Analisis dan Perbandingan Akurasi :

Langkah akhir adalah membuat laporan komparasi antara hasil AI internal dengan AI eksternal (Gemini AI). Perbandingan ini dilakukan berdasarkan kesesuaian hasil terhadap penilaian surveyor dan ditampilkan dalam bentuk visualisasi serta tabel analitik.

Panel Match Postman 970 Panel	Claim akurasi = 100% (Postman) 1.55% 10 dari 645 Claim	Panel Match AI 1084 Panel	Claim akurasi = 100% (AI) 1.4% 9 dari 645 Claim
Persentase Panel Match Postman 10.88%	Claim akurasi > 50% (Postman) 2.79% 18 dari 645 Claim	Persentase Panel Match AI 19.44%	Claim akurasi > 50% (AI) 11.16% 72 dari 645 Claim

Gambar 3.23. Hasil Analisis dari perbandingan 2 AI

Sumber: Google Sheets Perusahaan

Dari hasil laporan diatas, AI internal terbukti lebih unggul dan konsisten dibanding Postman, dengan jumlah panel match yang lebih tinggi dimana AI Internal mendeteksi 1.084 Panel (19.44%) dibanding Gemini 970 Panel (10.88%) serta klaim dengan akurasi di atas 50% yang jauh lebih banyak AI Internal mendeteksi 72 Klaim (11.16%) dibanding Gemini 18 Klaim (2.79%). Meskipun Postman sedikit lebih unggul dalam klaim dengan akurasi 100% 10 Klaim (1.55%) dibanding AI Internal 9 Klaim (1.4%), proporsi tersebut jauh lebih kecil jika dibandingkan dengan total klaim yang melebihi 50%, yaitu hanya 18 klaim pada Postman dibanding 72 klaim pada AI. Hal ini

menunjukkan bahwa AI internal lebih stabil dalam mendeteksi klaim dengan akurasi tinggi, sehingga lebih direkomendasikan untuk digunakan dalam analisis klaim secara umum.

Performa AI internal yang lebih stabil disebabkan oleh pelatihan model yang berbasis pada data historis perusahaan, sehingga AI ini lebih mampu mengenali pola, format, dan standar kerusakan yang sesuai dengan kebutuhan operasional internal. Sebaliknya, Gemini AI memiliki karakteristik yang lebih umum karena tidak dikembangkan secara khusus untuk mengikuti SOP perusahaan, sehingga tingkat konsistensinya dalam konteks operasional harian menjadi lebih rendah. Selain itu, kualitas gambar yang dikirimkan oleh nasabah juga menjadi variabel penting yang turut memengaruhi akurasi kedua model.

Laporan perbandingan akurasi AI dapat dilihat melalui tautan berikut:

Klik di sini untuk melihat Laporan Akurasi Gemini AI dan AI Internal

B Pengolahan data perbandingan AI Perusahaan yang sudah di Mapping dan Labelling dengan Surveyor

1. Tabel list kendaraan

Pada tabel ini ditampilkan daftar lengkap semua merek dan model kendaraan yang telah melalui proses pelatihan (training). Data yang ditampilkan sudah mencakup proses *labelling* dan *mapping* untuk setiap panel kendaraan.

	Kendaraan	Status Training Panel	PERSENTASE AKURASI DETEKSI PANEL	PERSENTASE AKURASI DETEKSI KLAIM	TAB
SUV	DAIHATSU NEW TERIOS TX 1.5 M/T HC	Done	-	0.00%	
HATCHBACK	HONDA JAZZ 1.5 E CVT	Done	-	0.00%	# DATA REPORT KENDARAAN
MV	HONDA MOBILITY RS CVT	Done	30.77%	0.00%	# DATA REPORT KENDARAAN
HATCHBACK	SUZUKI OPLASH GL 1.2 M/T	Done	24.24%	0.00%	# DATA REPORT KENDARAAN
HATCHBACK	TOYOTA YARIS 1.5 S A/T	Done	28.57%	0.00%	# DATA REPORT KENDARAAN
KEMUDI	WALIND COMFORT 1.5 STD DOUBLE BLOWER	Done	17.95%	0.00%	# DATA REPORT KENDARAAN
HATCHBACK	DAIHATSU AVILA 4 A/T	Done	38.11%	0.00%	# DATA REPORT KENDARAAN
KEMUDI	TOYOTA SIENNA 1.5 V CVT	Done	15.38%	0.00%	# DATA REPORT KENDARAAN
HATCHBACK	TOYOTA YARIS 1.5 E M/T	Done	13.04%	0.00%	# DATA REPORT KENDARAAN
HATCHBACK	HONDA BRIO E A/T	Done	47.06%	0.00%	# DATA REPORT KENDARAAN
HATCHBACK	TOYOTA YARIS 1.5 E S/T	Done	40.91%	0.00%	# DATA REPORT KENDARAAN
KEMUDI	DAIHATSU ALL NEW HENIA 1.5 L R M/T	Done	24.59%	0.00%	# DATA REPORT KENDARAAN
HATCHBACK	TOYOTA ETIOS 1.5 G M/T	Done	15.52%	0.00%	# DATA REPORT KENDARAAN
HATCHBACK	DAIHATSU AVILA 4 M/T	Done	43.39%	40.00%	# DATA REPORT KENDARAAN
HATCHBACK	HONDA ALL NEW BRIO S 1.2 HD	Done	13.85%	0.00%	# DATA REPORT KENDARAAN
HATCHBACK	HONDA BRIO 1 M/T	Done	40.00%	0.00%	# DATA REPORT KENDARAAN
SUV	HONDA HR-V 1.5 S M/T	Done	22.50%	20.00%	# DATA REPORT KENDARAAN
HATCHBACK	HONDA JAZZ 1.5 S M/T	Done	25.00%	0.00%	# DATA REPORT KENDARAAN
MV	HONDA MOBILITY E CVT PRESTIGE	Done	8.89%	0.00%	# DATA REPORT KENDARAAN
MV	HONDA MOBILITY S A/T PRESTIGE	Done	33.33%	0.00%	# DATA REPORT KENDARAAN
SUV	DAIHATSU TERIOS TX ADVENTURE 1.5 M/T	Done	16.67%	0.00%	# DATA REPORT KENDARAAN

Gambar 3.24. List merk dan model kendaraan yang sudah di training

Sumber: Google Sheets Perusahaan

2. Mencari data klaim untuk model kendaraan tertentu

Dari database, dilakukan penarikan data berdasarkan model kendaraan sebagai acuan awal untuk mengetahui nomor klaim (*claim number*) yang

terkait. Nomor klaim ini kemudian digunakan untuk memeriksa apakah data surveyor atau petugas lapangan sudah tersedia atau telah diisi dengan lengkap.

NO_KLAIM	JENIS_MODEL	DESC_MERK	DESC_MODEL
CLM-311519	MINIBUS	DAIHATSU	ALL NEW XENIA 1.3 L X STD M/T
CLM-312100	MINIBUS	DAIHATSU	ALL NEW XENIA 1.3 L X STD M/T
CLM-312184	MINIBUS	DAIHATSU	ALL NEW XENIA 1.3 L X STD M/T
CLM-312897	MINIBUS	DAIHATSU	ALL NEW XENIA 1.3 L X STD M/T
CLM-313194	MINIBUS	DAIHATSU	ALL NEW XENIA 1.3 L X STD M/T
CLM-313401	MINIBUS	DAIHATSU	ALL NEW XENIA 1.3 L X STD M/T

Gambar 3.25. Mencari no klaim pada database

Sumber: Google Sheets Perusahaan

3. Melakukan sorting dan pencocokan data dengan oracle sql

Proses selanjutnya adalah menggabungkan dua tabel, yaitu tabel hasil analisis AI dan tabel hasil survei lapangan (surveyor), berdasarkan nomor klaim (*claim number*). Setelah kedua tabel digabung, dilakukan pencocokan data berdasarkan nama panel dan sisi kerusakan (*damage side*) pada kendaraan.

SVY_PANEL	SVY_SIDE	AI_PANEL	AI_SIDE	SORT_ORDER
BREKET BUMPER DEPAN	KANAN	-	-	0
BUMPER DEPAN	-	BUMPER DEPAN	-	0
-	-	-	-	1
-	-	KAP BAGASI / HATCHBACK	-	1
-	-	PINTU BELAKANG	KIRI	1

Gambar 3.26. Pencocokan data dengan oracle sql

Sumber: Google Sheets Perusahaan

4. Masukkan data kerusakan yang sudah di cocokkan ke Sheets

Setelah proses pencocokan dilakukan, salin data yang telah digabung ke dalam database untuk dianalisis lebih lanjut. Setiap baris data kemudian diberi label berdasarkan hasil kecocokan antara data surveyor dan data AI. Sebagai contoh, apabila kombinasi *panel* + *sisi* dari surveyor sama persis dengan *panel* + *sisi* dari AI, maka baris tersebut diberi label *panel match*.

JENS MOBIL	MERK MOBIL	TIPE MOBIL	NO_KLAIM	PANEL_SURVEYOR	SISI_PANEL_SURVEYOR	PANEL_AI	SISI_PANEL_AI	STATUS_MATCH_PANEL
MINIBUS	DAIHATSU	ALL NEW XENIA 1.3 L X STD MT	CLM-311519	BUMPER BELAKANG	-	BUMPER BELAKANG	-	PANEL MATCH
MINIBUS	DAIHATSU	ALL NEW XENIA 1.3 L X STD MT	CLM-311519	STOP LAMP UNIT	KANAN	-	-	PANEL TIDAK TERDETEKSI
MINIBUS	DAIHATSU	ALL NEW XENIA 1.3 L X STD MT	CLM-312100	BUMPER DEPAN	-	-	-	PANEL TIDAK TERDETEKSI
MINIBUS	DAIHATSU	ALL NEW XENIA 1.3 L X STD MT	CLM-312100	COVER FOG LAMP	KIRI	COVER FOG LAMP	KIRI	PANEL MATCH
MINIBUS	DAIHATSU	ALL NEW XENIA 1.3 L X STD MT	CLM-312100	ENGSEL PINTU DEPAN BAWAH KIRI	-	-	-	PANEL TIDAK TERDETEKSI
MINIBUS	DAIHATSU	ALL NEW XENIA 1.3 L X STD MT	CLM-312100	FENDER	KIRI	FENDER	KIRI	PANEL MATCH
MINIBUS	DAIHATSU	ALL NEW XENIA 1.3 L X STD MT	CLM-312100	LINER QUARTER	KIRI	-	-	PANEL TIDAK TERDETEKSI
MINIBUS	DAIHATSU	ALL NEW XENIA 1.3 L X STD MT	CLM-312100	PINTU BELAKANG	KIRI	PINTU BELAKANG	KIRI	PANEL MATCH
MINIBUS	DAIHATSU	ALL NEW XENIA 1.3 L X STD MT	CLM-312100	PINTU DEPAN	KIRI	PINTU DEPAN	KIRI	PANEL MATCH
MINIBUS	DAIHATSU	ALL NEW XENIA 1.3 L X STD MT	CLM-312100	VELG	-	-	-	PANEL TIDAK TERDETEKSI
MINIBUS	DAIHATSU	ALL NEW XENIA 1.3 L X STD MT	CLM-312100	QUARTER PANEL	KIRI	-	-	PANEL TIDAK RUSAK
MINIBUS	DAIHATSU	ALL NEW XENIA 1.3 L X STD MT	CLM-312100	TRIPLANG / ROOBER PANEL	KIRI	-	-	PANEL TIDAK RUSAK
MINIBUS	DAIHATSU	ALL NEW XENIA 1.3 L X STD MT	CLM-312100	TUTUP TANGKI BAHAN BAKAR	-	-	-	PANEL TIDAK RUSAK
MINIBUS	DAIHATSU	ALL NEW XENIA 1.3 L X STD MT	CLM-312184	BUMPER DEPAN	-	BUMPER DEPAN	-	PANEL MATCH
MINIBUS	DAIHATSU	ALL NEW XENIA 1.3 L X STD MT	CLM-312184	-	-	COVER FOG LAMP	KANAN	PANEL TIDAK RUSAK
MINIBUS	DAIHATSU	ALL NEW XENIA 1.3 L X STD MT	CLM-312184	-	-	COVER FOG LAMP	KIRI	PANEL TIDAK RUSAK
MINIBUS	DAIHATSU	ALL NEW XENIA 1.3 L X STD MT	CLM-312897	BUMPER DEPAN	-	BUMPER DEPAN	-	PANEL MATCH

Gambar 3.27. Memasukkan data kedalam google sheets
Sumber: Google Sheets Perusahaan

5. Membuat laporan dalam bentuk pivot tabel

Dari tabel yang telah dibuat, gunakan *pivot table* untuk menganalisis datanya. Analisis ini mencakup berapa banyak kesalahan yang terjadi, ada berapa perbedaan antara data AI dan surveyor, serta panel-panel apa saja yang sulit dideteksi oleh AI dan lainnya.

JUMLAH PANEL AI DAN SURVEYOR PER CLAIM				TINGKAT AKURASI AI SECARA GARIS BESAR				
NO_KLAIM	PANEL_AI	NO_KLAIM	PANEL_SURVEYOR	STATUS_MATCH_PANEL	JUMLAH_PANEL	PERSENTASE		
CLM-311519		CLM-311519		PANEL MATCH	9	38.10%		
CLM-312100		CLM-312100		PANEL TIDAK RUSAK	7	33.33%		
CLM-312184		CLM-312184		PANEL TIDAK TERDETEKSI	6	28.87%		
CLM-312897		CLM-312897						
CLM-315184		CLM-315184						
Grand Total		15 Grand Total			21	100.00%		
JUMLAH PANEL YANG TERMATCH DAN TIDAK				PANEL YANG SERING TIDAK TERDETEKSI				
NO_KLAIM	PANEL_MATCH	PANEL_TIDAK_RUSAK	PANEL_TIDAK_TERDETEKSI	Grand Total	MAKNA_PANEL_SURVEYOR	PANEL_TIDAK_TERDETEKSI	Grand Total	
CLM-311519	1		2	3	BREKET BUMPER DEPAN	1	1	
CLM-312100	4	3	4	11	BUMPER DEPAN	1	1	
CLM-312184	1	2	3	6	ENGSEL PINTU DEPAN BAWAH	1	1	
CLM-312897	1		1	2	LINER QUARTER	1	1	
CLM-315184		2	1	3	STOP LAMP UNIT	1	1	
Grand Total	8	7	6	21	VELG	1	1	
Grand Total					Grand Total	6	6	
TINGKAT AKURASI AI PER CLAIM								
NO_KLAIM	PANEL_MATCH	PANEL_TIDAK_RUSAK	PANEL_TIDAK_TERDETEKSI	Grand Total				
CLM-311519	50.00%		50.00%	100.00%				
CLM-312100	30.30%	21.27%	30.30%	100.00%				
CLM-312184	33.33%	66.67%		100.00%				
CLM-312897	100.00%			100.00%				
CLM-315184	20.00%	60.00%	20.00%	100.00%				
Grand Total	38.10%	33.33%	28.57%	100.00%				
				Claim Accuracy	Panel Accuracy	Sulit di Deteksi	Kesimpulan	53.85% Kesalahan
				20.00%	38.10%	BREKET BUMPER DEPAN BUMPER DEPAN ENGSEL PINTU DEPAN BAWAH LINER QUARTER STOP LAMP UNIT VELG	Kurang Akurat	Panel Tidak Rusak

Gambar 3.28. Membuat laporan dalam Pivot Table
Sumber: Google Sheets Perusahaan

3.4 Kendala dan Solusi yang Ditemukan

3.4.1 Kendala yang Ditemukan

1. Kesulitan dalam Memahami Sistem Pega :

Sistem Pega memiliki arsitektur yang cukup kompleks dan memerlukan waktu untuk mempelajari berbagai fiturnya, seperti penggunaan *Obj-Browse*, *Report Definition*, dan fitur lainnya.

2. Kesalahan dalam Penulisan Query SQL :

Beberapa query yang digunakan dalam proses pengolahan data mengalami kesalahan karena penggunaan alias yang kurang tepat atau sintaks SQL yang tidak sesuai.

3. Menentukan Informasi yang Relevan dalam Laporan :

Sering kali grafik atau data yang ditampilkan dalam presentasi atau laporan tidak diperlukan atau kurang memberikan informasi yang bermakna.

4. Kesulitan dalam Mengekspor Data ke Spreadsheet :

Proses ekspor data ke Excel mengalami kendala apabila template yang digunakan tidak sesuai dengan struktur data yang tersedia dalam sistem.

5. Akurasi AI yang dipengaruhi oleh kualitas gambar :

Sering kali AI gagal untuk mendeteksi panel dengan benar dikarenakan gambar yang buram, banyak pantulan pad bodi mobil, panel kerusakan yang tertutup tangan atau benda, dan yang lainnya.

3.4.2 Solusi atas Kendala yang Ditemukan

1. Memperdalam Pemahaman tentang Sistem Pega :

Mencatat setiap fitur yang telah digunakan serta melakukan peninjauan kembali (review) sebelum memulai pekerjaan, agar pemahaman terhadap sistem semakin meningkat.

2. Lebih Teliti dalam Penulisan Query SQL :

Memeriksa setiap baris dalam query secara menyeluruh, termasuk memastikan kesesuaian nama tabel dan penggunaan tanda baca seperti titik koma (;) agar query dapat dijalankan dengan benar.

3. Menentukan Topik atau Tujuan Laporan Sejak Awal :

Menetapkan fokus laporan sejak awal akan membantu penyusunan konten yang lebih terarah dan informatif. Selain itu, pendekatan pembelajaran dari kesalahan sebelumnya ("learning from your mistakes") juga diterapkan untuk memperbaiki kualitas presentasi dan laporan berikutnya.

4. Menggunakan Bantuan Google Collab dan Formula Spreadsheet :

Data diekstraksi dan diformat ulang dengan bantuan Google Collab menjadi bentuk tabel yang sesuai, kemudian diproses kembali menggunakan formula spreadsheet untuk meningkatkan efisiensi dan akurasi dalam pengolahan data.

5. Perbaiki SOP dan guideline untuk mekanisme foto panel kerusakan :

Untuk menghindari pengambilan gambar yang tidak bagus kualitasnya, di berikan contoh pengambilan foto yang baik, angle yang baik, dan segi lighting.

