

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

Implementasi *PEGA Systems* pada aplikasi ini, mengacu pada beberapa penelitian sebelumnya:

Penelitian yang dilakukan oleh (Listya Dwi Ariadi. 2018) yang berjudul *Perancangan dan Implementasi Modul Car Verification pada Warranty Application Menggunakan Platform PEGA System: Studi Kasus PT. Asuransi Sinar Mas*. Kerjasama merupakan suatu cara untuk memecahkan suatu masalah dalam perusahaan. Salah satu kerjasama dalam pembuatan aplikasi, PT. PGI bekerjasama dengan PT. Asuransi Sinar Mas dalam pembuatan *modul car verification* pada *warranty application*. Penelitian ini menggunakan *Platform PEGA System* dalam membangun *modul car verification* pada *warranty application*. *Platform PEGA System* dipilih karena platform ini berorientasi *pada Business Process Management (BPM)*. *Platform PEGA System* sangat memudahkan development dalam membangun sistem. Semua fungsi dasar dalam membuat sebuah *user interface* dan fungsi lainnya sudah tersedia dalam *PEGA System*. Penelitian ini menghasilkan aplikasi *modul car verification* yang memudahkan *user Dealer, Adjuster, dan Manajer* dalam melakukan proses bisnis verifikasi mobil PT. PGI.

Penelitian yang dilakukan oleh (Anggi Wahyu Desando. 2020) yang berjudul Implementasi dan Pengujian Pinjaman Baru Kartu Kredit *PEGA* (PT. BANK MANDIRI, TBK). Pada era teknologi informasi ini peranan komputer sangat penting dalam mendukung kinerja suatu perusahaan. Salah satu bidang perusahaan yang membutuhkan teknologi informasi adalah perbankan. Bank Mandiri adalah Badan Usaha Milik Negara (BUMN) beroperasi, Sebagai penyedia jasa keuangan di Indonesia layanan bank mandiri meliputi pembiayaan perdagangan, Valuta asing, Pengolahan kas, Proses pembayaran Kartu kredit dan debit, Produk unggulan dari bank mandiri salah satunya adalah Kartu Kredit.

Penelitian yang dilakukan oleh (Kevin Alexander Harjanto. 2020) yang berjudul Perancangan Sistem Pengajuan Cuti Izin Sakit (CIS) Menggunakan *Platform PEGA System*. PT Asuransi Sinar Mas merupakan perusahaan yang bergerak pada bidang layanan asuransi dan PT Asuransi Sinar Mas memiliki karyawan sekitar 2873 dan akan terus bertambah. Dengan banyaknya karyawan di dalam PT Asuransi Sinar Mas maka dibuatlah aplikasi pengajuan CIS yang bertujuan untuk lebih memudahkan dalam melakukan pengajuan izin / cuti kerja. Cuti / izin sakit dapat digunakan atau diajukan ketika karyawan tidak dapat bekerja karena memiliki berbagai halangan yang memungkinkan untuk tidak bekerja. Aplikasi ini dibuat menggunakan *Pega System 8* yang berfokus

pada pengelolaan *Business Process Management* (BPM) sebagai perantara antara komunitas bisnis dan metode yang digunakan dalam pembuatan aplikasi ini adalah *Agile*. Setelah dibuatnya aplikasi ini dan melakukan pengujian, aplikasi ini menghasilkan pengaruh terhadap sistem pengajuan yang menjadi lebih efisien karena tidak memakan waktu yang lama. Dengan adanya sistem ini, karyawan dapat melakukan pengajuan kapan saja dan dimana saja tanpa harus dilakukan secara manual atau tatap muka secara langsung

Penelitian yang dilakukan oleh (Trivena Andriani, 2020) yang berjudul Perancangan Aplikasi Perekrutan Karyawan Berbasis WEB Menggunakan *Platform PEGA*. Perekrutan karyawan merupakan proses penting untuk mendapatkan tenaga kerja yang berkualitas. Salah satu kendala yang dihadapi dalam proses perekrutan karyawan adalah pengelolaan data calon karyawan. Dalam penelitian ini dirancang sistem perekrutan karyawan berbasis *web* untuk pengelolaan data calon karyawan. Model proses yang dipergunakan dalam penelitian ini adalah model *prototype*. Platform yang dipergunakan adalah *Pega*. Penelitian ini menggunakan studi kasus di PT. Asuransi Sinar Mas. Berdasarkan hasil pengujian yang dilakukan didapatkan bahwa aplikasi yang dibuat mudah untuk digunakan serta membantu dalam melakukan rekrutmen.

Penelitian yang dilakukan oleh (Patrick Levin Sunjaya. 2022) yang berjudul Perancangan Sistem Renovasi Menggunakan *Platform PEGA System* Pada PT. Asuransi Sinar Mas. Perusahaan Asuransi Sinar Mas (ASM) adalah perusahaan nasional yang berfokus pada bidang asuransi yang mencakup seluruh aspek kebutuhan hidup sampai kebutuhan antariksa. Dengan banyaknya pegawai di dalam Perusahaan ASM yang tersebar di seluruh Indonesia, tentu banyak juga gedung yang digunakan. Dan dengan penggunaan gedung tersebut, wajar jika terjadi kerusakan, sehingga pekerjaan renovasi pasti dibutuhkan. Maka dari itu, di buatlah aplikasi pengajuan renovasi gedung yang bertujuan memudahkan dan mempercepat proses renovasi. Aplikasi ini dibuat dengan menggunakan *Pegasystem*, yaitu sebuah alat *Business Process Management* yang terpusat pada proses bisnis. Dengan dibuatnya sistem ini, karyawan dapat mengajukan pengajuan renovasi gedung dari kota manapun, dan pengajuan tidak perlu dikirimkan secara manual kepada seluruh pihak yang bersangkutan.

Perbandingan tinjauan pustaka disajikan pada tabel 2.1. Tabel tersebut menyajikan informasi mengenai peneliti, judul penelitian, metode, dan hasil dari penelitian terdahulu yang relevan dengan penelitian ini.

Tabel 2. 1 Tabel Tinjauan Pustaka

No	Sumber	Objek	Tool	Hasil
1	Listy Dwi Ariadi, 2018	Perancangan dan implementasi Modul <i>Car Verification</i> Pada <i>Warranty Application</i> menggunakan Platform PEGA System. Studi kasus : PT ASURANSI SINARMAS	PEGA SYSTEM	Modul <i>car verification</i> yang dirancang dan diimplementasikan dapat memudahkan pengguna, yaitu <i>dealer, adjuster</i> , dan manajer, dalam melakukan proses bisnis verifikasi mobil. Modul ini juga dapat meningkatkan efisiensi dan efektivitas proses bisnis verifikasi mobil di PT. Asuransi Sinar Mas

No	Sumber	Objek	Tool	Hasil
2	Anggi Wahyu Desando, 2020	Implementasi dan Pengujian Pinjaman Baru Kartu Kredit <i>PEGA</i> (PT. BANK MANDIRI, TBK)	<i>PEGA System</i>	Modul pinjaman baru kartu kredit <i>PEGA</i> dapat membantu meningkatkan kepuasan nasabah. Modul ini dapat memberikan pengalaman yang lebih baik bagi nasabah dalam mengajukan pinjaman baru kartu kredit.
3	Kevin Alexandre Harjanto. 2020	Perancangan Sistem Pengajuan Cuti Izin Sakit (CIS) Menggunakan <i>Platform Pega System</i>	<i>PEGA System</i>	Sistem ini dapat memberikan pengalaman yang lebih baik bagi karyawan dalam mengajukan cuti izin sakit. Sistem ini dapat memberikan informasi yang lebih lengkap dan akurat kepada karyawan, sehingga karyawan dapat membuat keputusan yang lebih tepat dalam mengajukan cuti izin sakit.

No	Sumber	Objek	Tool	Hasil
4	Trivena Andriani. 2020	Perancangan Sistem Renovasi Menggunakan Platform <i>PEGA System</i> pada PT. Asuransi Sinar Mas	<i>PEGA System</i>	Hasil dari skripsi perancangan Sistem Renovasi menggunakan Platform <i>PEGA System</i> pada PT. Asuransi Sinar Mas adalah positif. Sistem renovasi yang dirancang dan diimplementasikan dapat memudahkan pengguna, yaitu karyawan dan sistem, dalam melakukan bisnis renovasi

No	Sumber	Objek	Tool	Hasil
5	Patrick Levin Sunjaya. 2020	Perancangan Sistem Renovasi Menggunakan Platform PEGA System Pada PT. Asuransi Sinar Mas	Pega System	Sistem renovasi PEGA dapat membantu meningkatkan kepuasan karyawan. Sistem ini dapat memberikan pengalaman yang lebih baik bagi karyawan dalam mengajukan permohonan renovasi. Sistem ini dapat memberikan informasi yang lebih lengkap dan akurat kepada karyawan, sehingga karyawan dapat membuat keputusan yang lebih tepat dalam mengajukan permohonan renovasi. Selain itu, sistem ini juga dapat memberikan proses yang lebih cepat dan mudah bagi karyawan dalam mengajukan permohonan renovasi.

No	Sumber	Objek	Tool	Hasil
6	Felisia Maria Waas Mascarenhas (2023)	Implementasi Pega System dalam pembuatan aplikasi reservasi booking hotel	Pega System 8.7	Hasil yang direncanakan, Diharapkan aplikasi ini dapat menjalankan fitur reservasi dengan baik dan dapat meningkatkan booking hotel.

2.1. Dasar Teori

2.2.1. Pega Systems

Pega Systems adalah perusahaan perangkat lunak yang terkenal dalam pengembangan platform otomatisasi proses bisnis (Business Process Automation, BPA) dan pengelolaan keputusan bisnis (Business Decision Management, BDM). Mereka dikenal karena produk utama mereka, yaitu Pega Platform, yang digunakan oleh organisasi bisnis untuk mengotomatisasi berbagai aspek proses bisnis mereka.

Pega Platform memungkinkan perusahaan untuk merancang, mengotomatisasi, dan mengelola berbagai jenis proses bisnis, seperti manajemen kasus, manajemen keputusan, manajemen interaksi pelanggan, manajemen tugas, dan sebagainya. Ini dilakukan dengan menggunakan

teknologi berbasis aturan dan algoritma cerdas yang dapat mengotomatisasi tindakan dan keputusan bisnis berdasarkan aturan yang ditetapkan.

Pega Systems juga dikenal dengan pendekatan Low-Code/No-Code yang memungkinkan pengembangan aplikasi dengan lebih cepat dan efisien tanpa memerlukan pengetahuan pemrograman yang mendalam. Platform ini sering digunakan untuk mengotomatisasi proses bisnis yang kompleks dan berhubungan dengan pengalaman pelanggan, pengelolaan kasus, manajemen risiko, kepatuhan hukum, dan banyak lagi.

Pega Systems telah digunakan dalam berbagai industri, termasuk perbankan, asuransi, pelayanan kesehatan, perawatan pelanggan, dan lain-lain. Produk-produk mereka membantu perusahaan meningkatkan efisiensi, mengurangi biaya, meningkatkan kepatuhan, dan meningkatkan pengalaman pelanggan.

2.2.2. Low-Code

Low-Code adalah sebuah pendekatan dalam pengembangan perangkat lunak yang dirancang untuk mempercepat dan menyederhanakan proses pembuatan aplikasi dengan mengurangi sejumlah kode yang harus ditulis secara manual. Pendekatan ini memungkinkan pengembang perangkat lunak, bahkan yang mungkin

tidak memiliki latar belakang pemrograman yang kuat, untuk membuat aplikasi dengan lebih cepat dan lebih mudah. Berikut beberapa karakteristik utama dari pendekatan low code:

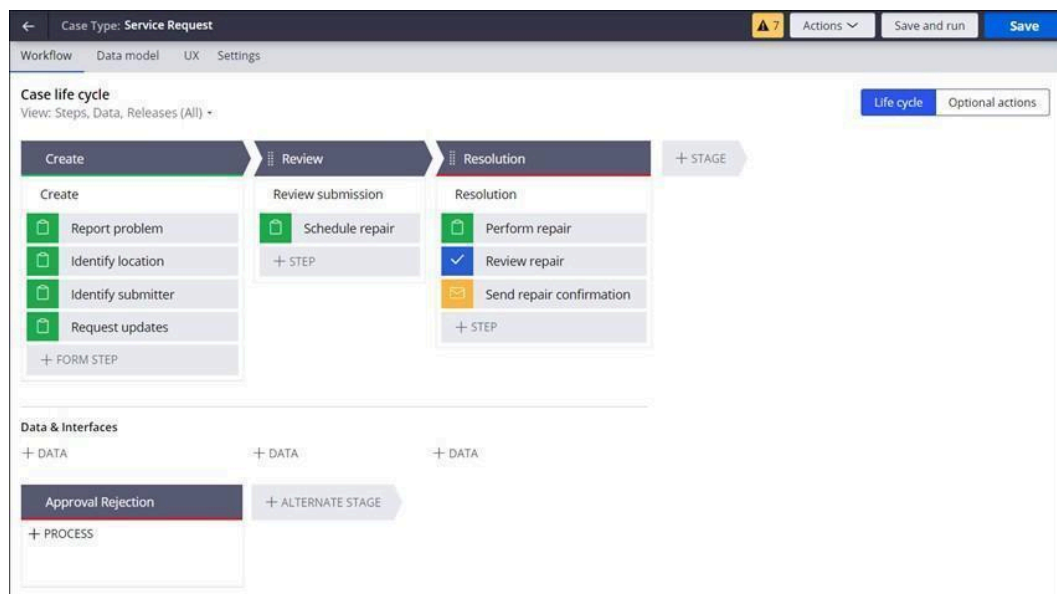
1. Antarmuka Visual: Low code biasanya memiliki antarmuka visual yang memungkinkan pengguna untuk mendesain aplikasi dengan menggambarkan logika bisnis dan aliran data dengan menggunakan elemen-elemen grafis seperti blok-blok fungsi atau logika alur.
2. Kode Otomatisasi: Meskipun dinamakan "low code," masih memerlukan penulisan kode, tetapi lebih banyak tugas pemrograman yang dilakukan secara otomatis oleh platform low code. Pengembang harus menambahkan kode hanya ketika perlu menangani kasus khusus atau logika yang lebih kompleks.
3. Komponen Siap Pakai: Low code sering menyertakan kumpulan komponen atau modul siap pakai, seperti integrasi dengan database, sistem pihak ketiga, atau layanan web. Ini memungkinkan pengembang untuk dengan cepat menggabungkan fungsionalitas yang diperlukan tanpa membangun semuanya dari nol.

4. Kustomisasi yang Dapat Diterapkan: Meskipun low code dapat mempercepat pengembangan, platform biasanya memungkinkan pengguna untuk menyesuaikan aplikasi sesuai kebutuhan bisnis mereka. Ini memungkinkan aplikasi yang dibangun dengan low code untuk tetap memenuhi persyaratan khusus perusahaan.
5. Percepatan Pengembangan: Salah satu tujuan utama low code adalah mempercepat siklus pengembangan aplikasi. Hal ini terutama bermanfaat dalam situasi di mana bisnis membutuhkan aplikasi dengan cepat untuk merespons perubahan pasar atau kebutuhan pelanggan.

Low code telah menjadi solusi yang populer untuk perusahaan yang ingin mengembangkan aplikasi dengan cepat, seperti aplikasi manajemen alur kerja, aplikasi internal, atau solusi khusus. Ini juga membantu mengurangi biaya pengembangan dan memungkinkan orang tanpa latar belakang pemrograman yang mendalam untuk berkontribusi dalam proses pengembangan perangkat lunak.

Saat developer memanipulasi dan memperluas model visual, alat pengembangan Low-code akan membuat kode untuk developer. Fitur antarmuka pengguna seperti drag-and-drop, alur proses, dan alat visual memungkinkan siapa saja membuat perangkat lunak transformasional terlepas

dari kemampuan teknisnya. Pendekatan ini memungkinkan peningkatan produktivitas, karena tugas pengembangan aplikasi sehari-hari disederhanakan, sehingga mengurangi keterlibatan TI. Dengan tool Low-code membuat pengembangan aplikasi lebih sederhana.



Gambar 2. 1 Case Life Cycle

2.2.3. Case Type

Case Type adalah sebuah teknik yang digunakan dalam pengembangan sebuah software atau sistem informasi untuk menangkap kebutuhan fungsional dari sistem yang bersangkutan. Case type menjelaskan interaksi yang terjadi antara 'aktor' — inisiator dari interaksi sistem itu sendiri dengan sistem yang ada, sebuah Case type

direpresentasikan dengan urutan langkah yang sederhana secara umum, case type terdiri dari tiga komponen utama, yaitu:

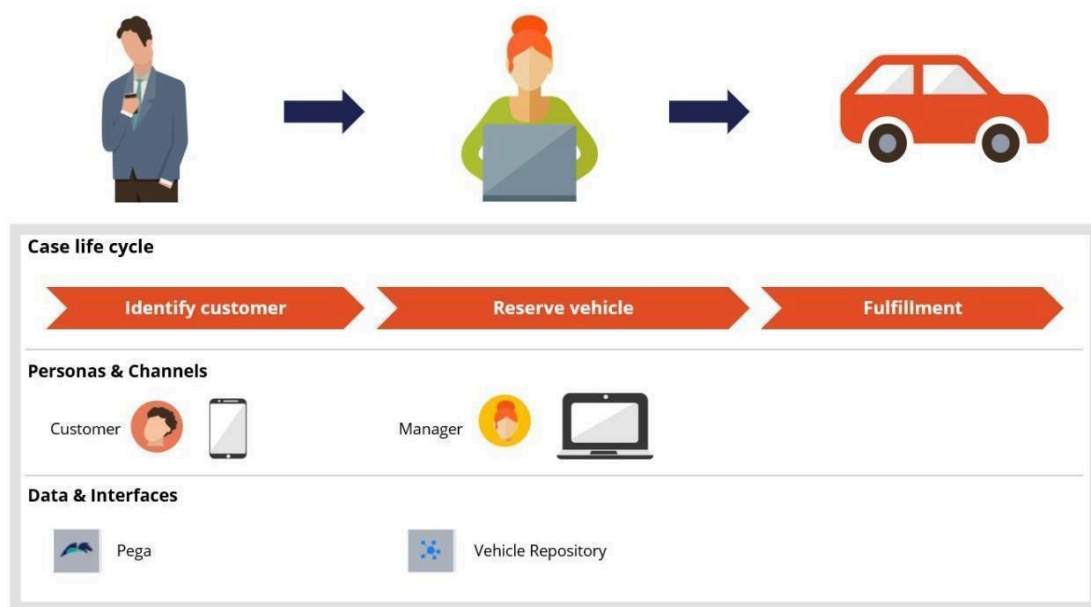
- a. Aktor: Pihak yang berinteraksi dengan sistem.
- b. Trigger: Kejadian yang memicu interaksi antara aktor dan sistem.
- c. Aktivitas: Langkah-langkah yang dilakukan sistem untuk memenuhi permintaan aktor.

Dalam konteks sistem pemesanan hotel, case type dapat digunakan untuk menggambarkan interaksi antara pelanggan, resepsionis, dan sistem. Berikut adalah beberapa contoh case type sistem pemesanan hotel:

1. Pemesanan Kamar: Case type ini menggambarkan interaksi antara pelanggan dan sistem untuk melakukan pemesanan kamar.
2. Pembatalan Pemesanan: Case type ini menggambarkan interaksi antara pelanggan dan sistem untuk membatalkan pemesanan kamar.
3. Perubahan Pemesanan: Case type ini menggambarkan interaksi antara pelanggan dan sistem untuk mengubah pemesanan kamar.
4. Check-In: Case type ini menggambarkan interaksi antara pelanggan dan resepsionis untuk melakukan check-in.

Case type merupakan salah satu alat yang penting dalam pengembangan sistem informasi. Case type dapat membantu untuk memahami kebutuhan

fungsional dari sistem, mengkomunikasikan kebutuhan tersebut kepada pemangku kepentingan, dan sebagai dasar untuk perancangan sistem.



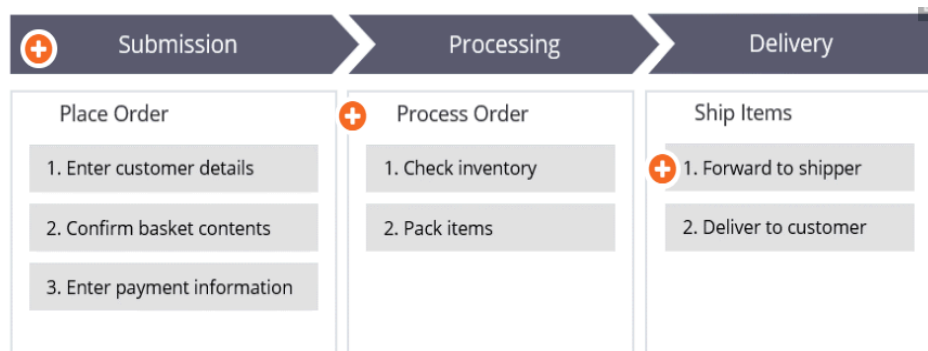
Gambar 2. 2 Case Type

2.2.4. Case Life Cycle

Case Life Cycle adalah serangkaian fase yang dilalui oleh sebuah case type dalam pengembangan sistem informasi. Fase-fase tersebut adalah sebagai berikut:

1. Fase 1: Identifikasi: Fase ini bertujuan untuk mengidentifikasi case type yang diperlukan oleh sistem.
2. Fase 2: Spesifikasi: Fase ini bertujuan untuk mendefinisikan secara detail case type yang telah diidentifikasi.

3. Fase 3: Implementasi: Fase ini bertujuan untuk mengimplementasikan case type ke dalam sistem.
4. Fase 4: Pengujian: Fase ini bertujuan untuk menguji case type untuk memastikan bahwa case type tersebut berfungsi dengan benar.
5. Fase 5: Deployment: Fase ini bertujuan untuk menyebarkan case type ke lingkungan produksi.



Gambar 2.3 Case Life Cycle

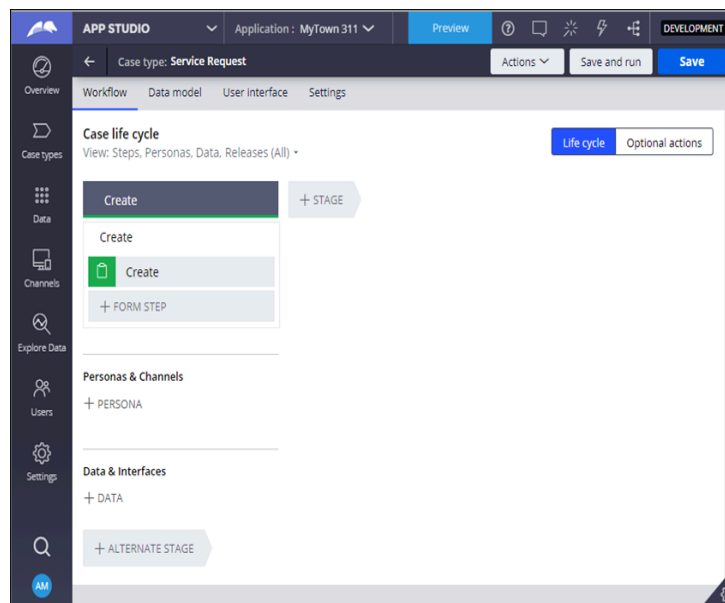
Pada gambar 2.3, dapat diketahui bahwa pada *Case Life Cycle* terdapat beberapa bagian, pada kolom pertama dinamakan dengan *Stage* yang memiliki fungsi sebagai tahapan yang mewakili perpindahan dari kasus dari pekerjaan ke pekerjaan yang lainnya, Misalnya, dalam pemesanan online, Anda mengharapkan tiga *Stage* berbeda untuk berpartisipasi: pelanggan yang melakukan pemesanan, karyawan gudang

yang memproses pesanan, dan karyawan layanan pengiriman yang mengantarkan pesanan. Oleh karena itu, kita membuat tiga tahap dalam jenis kasus pesanan online: *Submission*, *Processing*, dan *Delivery*. Pada kolom kedua dinamakan dengan *Process* yang memiliki fungsi sebagai serangkaian tugas, atau langkah, yang diselesaikan pengguna saat mereka mengerjakan kasus tersebut. Setiap tahap dapat berisi satu atau lebih proses. Misalnya, *Place Order*, *Process Order*, dan *Ship Items* adalah bagian dari *Process*. Pada kolom ketiga dinamakan dengan *Step* yang memiliki fungsi sebagai tindakan pengguna atau tindakan otomatis dalam proses yang dilakukan oleh aplikasi. Langkah-langkah yang dilakukan oleh sistem disebut sebagai langkah otomatisasi. Misalnya, Masukkan detail pelanggan adalah langkah dalam proses Pemesanan yang mengharuskan pengguna memasukkan informasi.

2.2.5. App Studio

App Studio menyediakan fitur inti untuk pengembangan aplikasi, seperti desain kasus, manajemen data, dan pengalaman pengguna. *App Studio* dirancang untuk pengguna *Low-Code Developer* umumnya termasuk pengembang aplikasi, pengembang *front-end*, *insinyur data*, *Bisnis Analisis*, dan arsitek sistem. Gunakan *App Studio* agar aplikasi Anda beroperasi dengan cepat.

Pada *App Studio* membantu *developer* untuk memvisualisasikan faktor kunci dari proses bisnis yang diinginkan. Saat pengguna menguraikan prosesnya, Developer membangun hubungan antara tahapan proses, persona yang berpartisipasi, saluran komunikasi, dan data yang diperlukan untuk penyelesaian proses. Model visual yang dibangun di *App Studio* membantu *Developer* merencanakan dan mengelola beban kerja tim pengembangan.



Gambar 2. 4 Tampilan dari App Studio

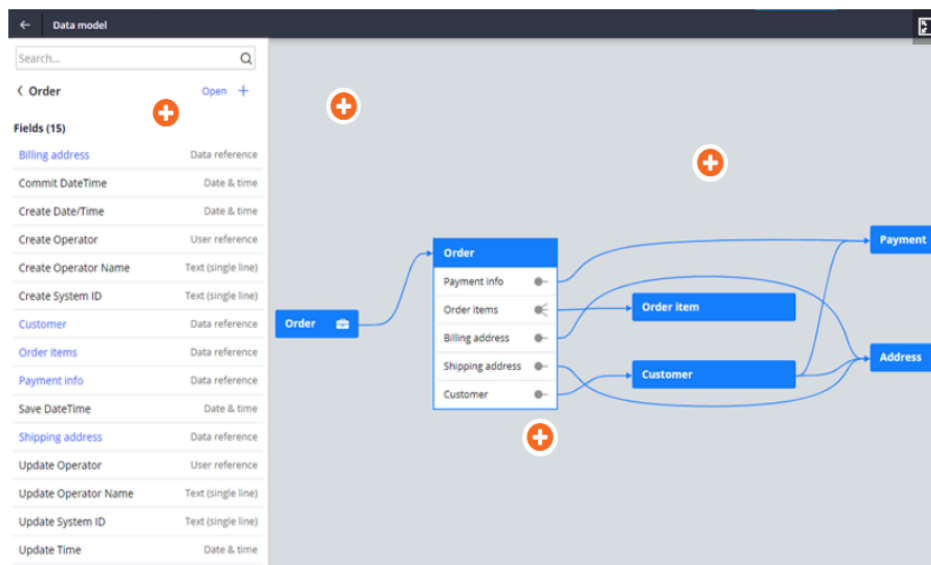
2.2.6. Data Modeling

Data Modeling adalah adalah proses dimana elemen data masuk ke aplikasi dalam format yang sesuai untuk bisnis dan kemudian

diproses, dilaporkan, dan disimpan. *Data modeling* menentukan tipe dan struktur data dalam aplikasi dan menstandarkan bagaimana elemen data berhubungan satu sama lain. *Data modeling* juga merupakan alat penting untuk komunikasi antara pemangku kepentingan bisnis yang menentukan persyaratan data yang diperlukan dan dibuat oleh proses bisnis, dan arsitek sistem yang bertindak berdasarkan persyaratan untuk membangun aplikasi. Misalnya, model data yang dibuat mungkin mewakili data yang digunakan oleh aplikasi untuk mengambil data tentang buku teks. Aplikasi ini meneruskan ISBN buku teks ke layanan yang mengembalikan daftar judul dan edisi dan kemudian memformat hasilnya sehingga perwakilan penjualan dapat memproses pesanan menggunakan judul dan dengan mudah beralih antar edisi. Jika Developer mengimpor data untuk tipe data Buku Teks secara rutin, maka kita dapat menentukan pemetaan default antara bidang ISBN di file .csv dan bidang ISBN di tipe data.

Anda memerlukan komponen berikut untuk memodelkan data:

1. field: Properti yang menyimpan dan memformat data dalam aplikasi Anda
2. Objek data: Kategori data yang memiliki bidang, pemetaan bidang, dan koneksi ke sumber data.



Gambar 2. 5 Tampilan dalam menu Data Model

Pada gambar Gambar 2.5 terdapat 4 bagian, pertama adalah Objek data dalam tipe kasus yang dipilih Daftar objek data yang berisi tipe kasus yang dipilih muncul di panel kiri. Yang kedua adalah *Data modeling* visual mengilustrasikan komponen data yang terdapat dalam suatu pilihan dan memperlihatkan hubungannya dengan objek data lainnya. Yang ketiga adalah *Relationship indicators*, setiap komponen dalam tipe kasus yang dipilih memiliki ikon yang menunjukkan tipe hubungan yang dimiliki komponen tersebut dengan objek data lainnya. Lingkaran dengan satu garis menandakan hubungan *one-to-one*. Lingkaran dengan banyak garis menunjukkan hubungan *one to many*. Yang keempat adalah *Related data objects* dengan mengklik objek data

terkait mana pun akan mengisi daftar di panel kiri untuk memperlihatkan bidang apa yang terdapat dalam objek data.

2.2.7. Dev Studio

Aplikasi *Pega Platform*TM yang dibangun pada perusahaan akan semakin sempurna tergantung pada kerja sama antar dua kelompok utama pengembang aplikasi. Pakar domain — analis bisnis, pengembang warga, dan pengembang *front-end* — memberikan wawasan berharga mengenai proses dan kebutuhan pengguna. Pakar implementasi — arsitek sistem, pengembang *full-stack*, administrator *database*, dan administrator keamanan — memberikan keahlian yang diperlukan untuk menangani kasus penggunaan penting yang memerlukan konfigurasi kompleks.

Di *App Studio*, pakar domain dapat mengakses fitur pengembangan aplikasi inti (desain kasus, manajemen data, dan pengalaman pengguna) dan menerapkan pengetahuan mereka untuk meningkatkan hasil pengembangan. Untuk mendukung konfigurasi aturan tingkat lanjut dalam aplikasi, *pega Platform* menyediakan lingkungan pengembangan kedua bagi pakar implementasi. Di *Dev Studio*, pakar implementasi mengakses formulir aturan secara langsung untuk mengatasi persyaratan konfigurasi yang kompleks atau kurang umum. Selain itu, *Dev Studio* menyediakan fitur untuk mengkonfigurasi izin keamanan dan kontrol akses, mengelola aturan untuk mendorong

penggunaan kembali, dan mengatasi keterbatasan kinerja aplikasi. Dengan menyediakan dua lingkungan pengembangan, *Pega Platform* mendukung setiap kelompok pengembang aplikasi dengan lingkungan yang disesuaikan dengan tingkat keahlian mereka dan dioptimalkan untuk tugas yang mereka lakukan. *Developer* aplikasi dapat beralih antara *App Studio* dan *Dev Studio* sesuai kebutuhan untuk mengkonfigurasi perilaku Aturan. Sudut kiri atas setiap studio berisi menu yang mencantumkan semua studio yang tersedia bagi pengguna. Untuk beralih ke studio lain, pilih nama studio dari menu.