

Spesifikasi Tugas Besar 2: O.W.C.A Logistics

IF2010 Pemrograman Berorientasi Objek

Revisi Terakhir:

27.05.25: Penambahan data pada pengiriman internasional

30.05.25: Penambahan nomor resi pada pengiriman

03.06.25: Penjelasan field *status* pada manajemen parsel, penambahan informasi tentang settings untuk plugin, dan penjelasan tentang data store state management

05.06.25: Tambahan penjelasan state pengiriman dan penambahan info soft-delete kurir

Daftar Isi

Daftar Isi	2
Deskripsi Persoalan	3
Spesifikasi Sistem	5
GUI	5
IDE dan Build Tools	5
Ketentuan Teknis	5
Mekanisme	7
● Manajemen Pengiriman	7
● Tracking Pengiriman	8
● Manajemen Kurir	9
● Manajemen Parsel	10
● Ganti Akun	11
● Halaman Kurir	12
● Halaman Setting	13
● Threading	14
● Exception	14
● Plugin	14
○ Analytics	15
● Data Store	16
Bonus	18
GUI Responsive	18
Unit Testing	18
Pemindaian Barcode	18
Plugin Driver Simulator	19
Reports Generation	19
Notifikasi Parsel	19
Panduan Pengeraaan	20
Panduan Laporan	20
Asistensi & QnA	21
Pengumpulan dan Deadline	21

Deskripsi Persoalan



Sumber: Gambar dibuat menggunakan DALL·E, model AI oleh OpenAI

Setelah sekian lama menjalankan operasi rahasia di balik layar, OWCA kini menghadapi tantangan yang berbeda: urusan logistik. Bukan lagi menyelamatkan dunia dari penjahat super, melainkan mengatur ratusan pengiriman peralatan agen, surat rahasia, dan parcel berisi perlengkapan misi ke berbagai titik di seluruh dunia—mulai dari pusat pelatihan di Antartika hingga markas bawah laut di Samudra Pasifik.

Sayangnya, sistem logistik OWCA masih bergantung pada catatan manual, spreadsheet tersebar, dan komunikasi via walkie-talkie. Akibatnya, banyak pengiriman yang terlambat, parcel tertukar, bahkan laporan yang tidak bisa dipertanggungjawabkan. Kondisi ini membuat Mr. Monogram geram. “Kita bisa melacak penjahat dengan teknologi satelit, tapi kirim pelindung lutut saja bisa nyasar tiga benua!”

Maka dibentuklah satuan tugas baru: membangun sistem manajemen logistik modern yang bisa mengatur pengiriman, kurir, parcel, hingga pelacakan status pengiriman secara efisien dan real-time. Sistem ini harus bisa berjalan lancar di markas OWCA, mendukung operasional lapangan, dan cukup fleksibel untuk menangani kondisi darurat yang seringkali muncul tiba-tiba.

Namun tentu saja, membangun sistem ini bukan sekadar soal tampilan. Dibutuhkan perencanaan arsitektur yang rapi, pengolahan data yang fleksibel, serta kemampuan integrasi dengan plugin untuk menghadapi skenario-skenario yang tak terduga di dunia OWCA.

Misi ini tidak mengandalkan laser atau jetpack, tapi tetap sama pentingnya: menciptakan sistem logistik yang andal untuk mendukung para agen dalam menjalankan tugasnya—tanpa ada lagi parcel yang salah kirim ke markas musuh.

Sayangnya, kemampuan OWCA sangat terbatas dalam hal pemrograman sistem manajemen logistik. Mr. Monogram pun kembali meminta bantuan Purry, yang—seperti biasa—tidak tahu perbedaan antara public class dan public enemy. Oleh karena itu, OWCA meminta bantuanmu untuk membangun OWLMS, sistem logistik modern yang akan menjadi tulang punggung operasi rahasia para agen. Tanpa sistem ini, satu paket yang tersesat bisa berarti kehancuran sebuah misi. Jadilah bagian dari tim penyelamat OWCA—bukan dengan senjata, tapi dengan baris-baris kode!

Spesifikasi Sistem

GUI

- Aplikasi **wajib menggunakan JavaFX** sebagai framework GUI.
- Desain antarmuka dibebaskan, namun harus tetap mendukung seluruh fungsionalitas sistem.
- Seluruh fitur aplikasi harus dapat diakses melalui GUI.
- CSS diperbolehkan. HTML dan Web Rendering dilarang.

IDE dan Build Tools

- **Disarankan** menggunakan **IntelliJ IDEA** karena mendukung pengelolaan JDK, library, dan integrasi build tools dengan baik.
- Aplikasi **wajib menggunakan Maven** sebagai build automation tool.
- Build tools digunakan untuk:
 - Menambahkan library eksternal seperti JSON/XML parser, PDF generator, dll.
 - Otomatisasi proses kompilasi dan build.
 - Pengelolaan struktur proyek secara modular.

Ketentuan Teknis

Berikut adalah hal-hal yang **minimal** wajib diimplementasikan di aplikasi yang Anda buat.

Pemrograman Berorientasi Objek

1. Inheritance
2. Composition
3. Interface
4. Polymorphism (overriding dan overloading)
5. Gunakan Java Collection API (List, Map, dll) sesuai kebutuhan.
6. Terapkan prinsip SOLID

Design Pattern

Wajib menggunakan minimal 5 design pattern:

1. Adapter Pattern untuk Data Store
2. 4 design pattern lainnya bebas sesuai kebutuhan sistem ([referensi](#))

Reflection: Wajib digunakan dalam sistem plugin.

Threading

1. Wajib menampilkan jam digital di halaman utama menggunakan thread.

2. Nilai tambah diberikan jika ada penggunaan lain selain jam digital di halaman utama dan yang ditulis dispesifikasi.

Notes

- Hindari menggunakan konsep OOP diatas hanya karena diwajibkan saja! Gunakanlah konsep OOP yang sesuai dengan kasusnya!
- DRY (Don't Repeat Yourself), tidak memiliki kode duplikat, pindahkan ke fungsi/class.
- Memiliki struktur kelas yang mudah dipahami.
- Dekomposisi yang baik dan implementasi yang tidak terlalu kompleks (sebuah method tidak terlalu panjang). Pecah-pecah dan buat method baru agar tidak terlalu kompleks.
- Maksimalkan SOLID terutama pada open-closed principle dan dependency inversion. Karena tugas besar ini memiliki sistem plugin.
- Usahakan memaksimalkan penggunaan konsep OOP, terutama konsep polymorphism, inheritance, composition dan sebagainya.

Mekanisme

Program yang Anda buat harus memenuhi beberapa mekanisme. Berikut merupakan mekanisme yang harus Anda implementasikan di program

- **Manajemen Pengiriman**

Terdapat suatu halaman pada aplikasi untuk mengelola seluruh aktivitas terkait pengiriman barang, mulai dari pembuatan pengiriman baru, pencarian pengiriman, dan pengeditan/penghapusan entri. Setiap entri pengiriman dapat memiliki beberapa parsel. Setidaknya pengiriman memiliki atribut sebagai berikut.

- **ID Pengiriman:** format bebas
- **No Resi:** fomat bebas, contoh: **SPX00001**
- **Tujuan:** String, alamat tujuan
- **Tipe:** 2 tipe: Domestik, Internasional
- **Status:** Enum
 - 1. Menunggu Konfirmasi – nunggu konfirmasi dari kurir (*default*)
 - 2. Diproses – setelah di-confirm oleh kurir tapi belum diambil
 - 3. Menunggu Kurir – mau diambil (di Courier Dashboard → ambil paket)
 - 4. Dikirim – kurir baru jalan (di Courier Dashboard → paket dalam perjalanan)
 - 5. Tiba di Tujuan – paket sampai (*end state*)
 - 6. Gagal – paket gagal sampai

Kalau misalnya paket belum sampai di tujuan, dan kurir di *soft-delete*, maka state pengiriman akan menjadi *gagal* dan pengiriman bisa di-reassign ke kurir lain.

- **Tanggal Pembuatan:** Tanggal entri dibuat
- **Tanggal Diperbarui:** Tanggal entri diperbarui
- **Nama pengirim:** string nama pengirim atau *customer* yang mendaftarkan parsel, minimal 3 karakter.
- **Nomor telepon pengirim:** string digit nomor telepon (karakter 0-9) dengan panjang 10-15 karakter.
- **Nama penerima:** string nama penerima, minimal 3 karakter.
- **Nomor telepon penerima:** string digit nomor telepon (karakter 0-9) dengan panjang 10-15 karakter.
- **List Parsel:** List dari parsel yang dikirimkan

Selain menampilkan tabel dari pengiriman, pada halaman ini juga terdapat beberapa fitur sebagai berikut.

1. Search & Filter

- Kotak pencarian teks untuk mencari berdasarkan ID Pengiriman, alamat tujuan, atau catatan lain
- Filter tipe: All, Domestik, Internasional
- Filter status

2. Actions

- Add: Domestik, Internasional
 - Saat pembuatan pengiriman ada pembuatan list parsel juga, barang-barang apa saja yang akan dikirim, silahkan lihat pada bagian [Manajemen Parsel](#)
- Edit
- Delete

3. Dapat melihat detail parsel (modal kecil)

Untuk fitur search pada halaman ini, implementasikan **ShipmentSearcher<T extends Shipment>** dengan **method search(SearchCriteria criteria)** yang mengembalikan **Collection<? extends T>** (Wildcard), memungkinkan pencarian menghasilkan berbagai subtipe pengiriman (seperti *DomesticShipment* atau *InternationalShipment*) sesuai dengan kriteria yang dimasukkan.

Penerapan wildcard generics ini memungkinkan sistem pencarian tetap *type-safe* saat menampilkan hasil yang bervariasi pada tabel pengiriman, tanpa perlu melakukan casting yang berulang dan berisiko error.

Hal yang membedakan pengiriman domestik dan internasional adalah, penambahan field pada pengiriman internasional. Biasanya, pengiriman internasional memerlukan dokumen pengiriman untuk perpajakan. Sediakan sebuah data **CustomsInformation** yang berisi kode pajak (string), dokumen pajak (tipe pdf, disimpan di-lokal dan yang dicatat adalah path filenya saja pada data store). Kedua informasi ini akan disimpan untuk pengiriman.

Manajemen Pengiriman

ID Pengiriman	Tujuan	Tipe	Status	Tgl Pembuatan	Tgl Diperbarui	Action
1001	123 Elm st	Domestik	Menunggu Konfirmasi Diproses	1 Mei 2025	1 Mei 2025	
1002	123 Elm st	Domestik	Menunggu Kurir Dikirim	1 Mei 2025	1 Mei 2025	
1003	123 Elm st	Domestik	Tiba di Tujuan Gagal	1 Mei 2025	1 Mei 2025	
1004	123 Elm st	Domestik	Menunggu Konfirmasi	1 Mei 2025	1 Mei 2025	

1 of 2 Previous Next

Ilustrasi UI (hanya sebagai contoh, silahkan kreasikan sendiri)

● Tracking Pengiriman

Setiap entri pada tabel manajemen pengiriman memiliki halaman detail yang dapat dibuka. Halaman ini berisi detail pengiriman sebagai berikut.

1. Status pengiriman
2. List parsel
3. Informasi pengiriman
 - Kurir
 - Informasi pengirim
 - Informasi penerima
 - Log status pengiriman

Contoh: <Datetime> - <Status> - <Message>

- a. 01/01/2025 10.00 - Menunggu Konfirmasi
- b. 01/01/2025 12.00 - Diproses - Paket sedang diproses oleh kurir
- c. 01/01/2025 13.00 - Dikirim - Paket sedang dikirim oleh kurir
- d. 02/01/2025 15.00 - Tiba di Tujuan - Paket sudah sampai tujuan

The screenshot shows a user interface for managing a delivery tracking system. At the top, there is a navigation bar with tabs: 'Manajemen Pengiriman' (selected), 'Manajemen Kurir', and 'Manajemen Parsel'. On the right side of the header, there is a user profile icon labeled 'CN Admin' with a dropdown arrow.

Tracking Pengiriman: 1001

Status: Diproses

Kurir: Danny

Pengirim: John / 082142

Penerima: Jane / 082146

List Parsel

ID Parsel	Dimensi	Berat	Jenis Barang
1001	20x30	5 kg	Pakaian
1002	20x30	5 kg	Pakaian
1003	20x30	5 kg	Pakaian

Log Status

Datetime	Status	Pesan
01/01/2025 10.00	Menunggu Konfirmasi	Menunggu Konfirmasi
01/01/2025 12.00	Diproses	Paket sedang diproses oleh kurir

Ilustrasi UI (hanya sebagai contoh, silahkan kreasikan sendiri)

● Manajemen Kurir

Terdapat suatu bagian aplikasi untuk manajemen kurir yang terdaftar pada aplikasi. Kurir setidaknya memiliki atribut sebagai berikut.

- **Nama:** String, berupa nama lengkap kurir, minimal 3 karakter
- **Jenis kelamin:** Enum laki-laki atau perempuan
- **Foto:** gambar untuk profil kurir, mungkin untuk kosong atau tidak ada. Berkas gambar dapat disimpan pada *filesystem*, sehingga yang tersimpan di data store adalah *path* dari file.
- **Tanggal lahir:** format tanggal atau date bebas

Aplikasi ini perlu mengimplementasikan CRUD (Create, Read, Update, Delete) untuk kurir. Perlu diperhatikan, untuk delete, apabila kurir yang dihapus sudah memiliki pengiriman yang terkait, pengiriman tersebut **tidak** ikut terhapus. Untuk itu, kalian dapat mengimplementasikan *soft delete* untuk penghapusan kurir. Artinya, kurir yang terhapus tetap ditampilkan pada bagian pengiriman.

Courier Management					+ Add New Courier
Photo	Name	Gender	Date of Birth	Actions	
	John Doe	Male	May 15, 1990		
	Jane Smith	Female	October 22, 1988		
	Alex Johnson	Non-binary	March 8, 1995		
	Sarah Williams	Female	December 17, 1992		
	Michael Brown	Male	July 30, 1985		

Ilustrasi UI (hanya sebagai contoh, silahkan kreasikan sendiri)

● Manajemen Parsel

Parsel merupakan bentuk objek yang dikirimkan pada setiap pengiriman. Artinya, setiap pengiriman akan memiliki satu atau lebih parsel. Bagian ini mirip seperti pada bagian manajemen kurir, tetapi data yang diolah adalah data parsel. Setiap parsel setidaknya memiliki atribut berikut.

- **ID Parsel:** format bebas
- **Status:** Registered / Unregistered
- **Dimensi:** berupa p x l x t. Penyimpanan dibebaskan (array atau masing-masing atribut terpisah)
- **Berat:** berupa massa parsel dalam satuan gram.
- **Jenis barang:** string berupa jenis barang yang dikirim, misalnya pakaian, elektronik, dsb.

Terdapat halaman yang dapat melihat daftar parsel yang terdaftar pada sistem, beserta statusnya. Status parsel dapat ditentukan berdasarkan pengiriman yang terkait. Jika belum ditetapkan pada suatu pengiriman, status parsel adalah registered atau terdaftar, jika belum ditetapkan pada suatu pengiriman maka statusnya adalah unregistered.

Dalam manajemen parsel, ketika meng-edit parsel pastikan hanya parsel yang berstatus unregistered yang dapat diubah, jika parsel sudah ditetapkan ke pengiriman (registered), maka detail parsel tidak bisa diubah.

Parsel yang belum ditetapkan pada suatu pengiriman dapat dipilih untuk ditambahkan pada pengiriman (manajemen pengiriman).

- **Ganti Akun**

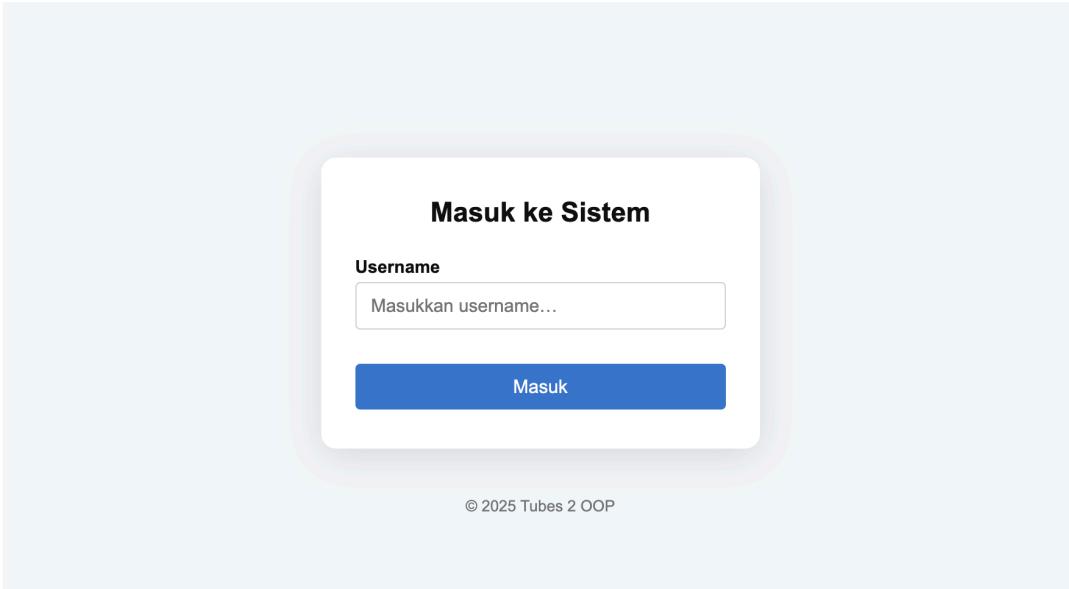


Ilustrasi UI (hanya sebagai contoh, silahkan kreasikan sendiri)

Selain menjadi Admin yang dapat melakukan manajemen terhadap kurir dan parsel, pengguna juga dapat melakukan Login sebagai Kurir.

Langkah	Deskripsi
1. Buka Menu Profil	Pengguna menekan avatar/foto profil di bagian kanan-atas.
2. Pilih Ganti Akun	Pengguna dapat menekan sebuah tombol untuk melakukan pergantian akun.
3. Masukkan Nama Kurir	Tampilkan halaman Login dan sebuah kolom untuk memasukkan Nama Kurir.
4. Autentikasi	Jika nama ditemukan, maka autentikasi sukses, lalu akan ditujukan ke Halaman Kurir Jika nama tidak ditemukan, maka tampilkan pesan <i>error</i>
5. Kembali ke Admin	Dari menu profil yang sama pada kurir, terdapat opsi untuk ganti akun untuk kembali Login. Pengguna dapat Login kembali menjadi Admin dari halaman

	yang sama. Pastikan pada saat Autentikasi username, dilakukan validasi apakah username tersebut Admin atau Kurir.
--	---



Ilustrasi UI (hanya sebagai contoh, silahkan kreasikan sendiri)

- **Halaman Kurir**

Sebagai kurir, terdapat beberapa fitur yang didukung pada aplikasi ini, yaitu melakukan Update status parsel yang di-assign kepada kurir tersebut dan dapat mengecek apakah ada parsel baru yang di-assign kepada kurir tersebut.

Pastikan pengiriman yang masuk ke dalam Dashboard hanyalah yang sudah dalam status "Menunggu Kurir" dan di-assign kepada atas nama kurir tersebut. Juga lakukan pagination untuk halaman tersebut, yakni setiap halaman dapat menampilkan sejumlah X baris, dan pengguna bisa berpindah halaman untuk mendapatkan data dari X-2X (seperti pagination pada umumnya), jumlah baris yang ditampilkan disesuaikan dengan ukuran screen.

No. Resi	Penerima	Alamat	Status	Aksi
INV001	John Doe	Jl. Mawar No.10, Bandung	Menunggu Kurir	Menunggu Kurir <input type="button" value="▼"/> <input type="button" value="Simpan"/>
INV002	Jane Smith	Jl. Melati No.5, Cimahi	Dikirim	Dikirim <input type="button" value="▼"/> <input type="button" value="Simpan"/>
INV003	Alex Johnson	Jl. Kamboja No.2, Bandung	Tiba Di Tujuan	Tiba Di Tujuan <input type="button" value="▼"/> <input type="button" value="Simpan"/>
INV004	Alice Brown	Jl. Kenanga No.8, Jakarta	Gagal	Gagal <input type="button" value="▼"/> <input type="button" value="Simpan"/>

Ilustrasi UI (hanya sebagai contoh, silahkan kreasikan sendiri)

- Update Status Parsel

Selain menampilkan informasi, kurir diberi kemampuan untuk memperbarui status pengiriman secara langsung.

Status yang dapat diubah oleh kurir adalah:

1. Menunggu Kurir
2. Dikirim
3. Tiba Di Tujuan
4. Gagal

Pastikan status ini tidak *reversible*, misalnya jika status sudah diubah menjadi "Tiba Di Tujuan" tidak bisa diubah menjadi Dikirim.

State dari status: Menunggu Kurir -> Dikirim -> Tiba Di Tujuan / Gagal

- Halaman Setting

Terdapat halaman setting yang menyediakan opsi untuk mengubah tempat penyimpanan file data serta format. Layout halaman dibebaskan. Jika data file disimpan lebih dari 1 file maka basis penyimpanan adalah memilih folder.

Untuk menghapus/mencopot plugin (akan dijelaskan di spesifikasi bagian plugin).

Jika ada hal-hal lain yang bisa dikonfigurasi atau disetting, dapat diletakkan pada halaman ini.

- **Threading**

Implementasi *threading* lain selain notifikasi penerimaan parsel pada kurir (bonus) adalah tampilan jam pada dashboard atau bagian pinggir diaplikasi.

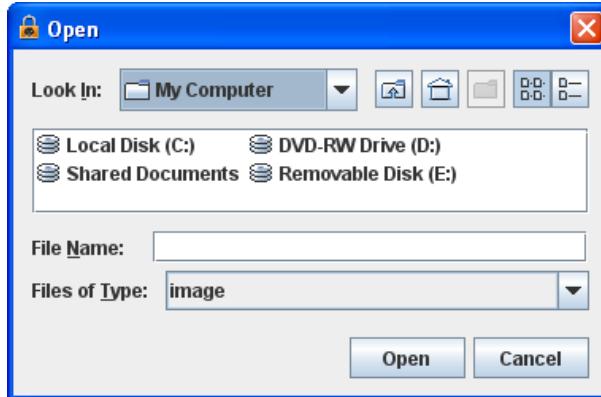
- **Exception**

Gunakan exception untuk menangani kasus-kasus seperti pengagalan penghapusan pengiriman ketika status pengiriman sudah bersama driver. Gunakan pola yang sama untuk tindakan lain yang dirasa diperlukan penggunaan untuk exception.

- **Plugin**

Plugin adalah perangkat lunak yang menambahkan fitur tertentu pada suatu program tanpa harus mengubah program itu sendiri. Biasanya, plugin digunakan agar pengguna dapat memilih komposisi fitur pada program yang tersedia. Sebagai contoh, wordpress adalah salah satu perangkat lunak berbasis website yang menggunakan sistem plugin, sehingga pengguna dapat memilih plugin yang dibutuhkan dan memungkinkan developer untuk membuat plugin baru tanpa mengubah wordpress itu sendiri.

Tugas anda adalah membuat sistem plugin pada tugas besar ini. Secara umum, terdapat sebuah menu plugin yang dapat membuka sebuah panel baru untuk membaca file. Tampilan memilih file adalah sebagai berikut (yang ada pada halaman setting). Diimplementasikan sebagai sebuah button untuk melakukan loading plugin.



Ilustrasi UI (hanya sebagai contoh, silahkan kreasikan sendiri)

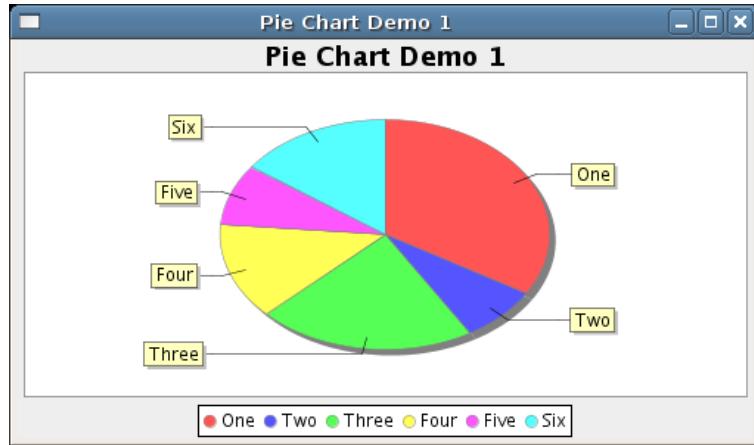
File yang bisa dibaca hanyalah file .jar yang spesifikasinya dijelaskan pada bagian berikutnya. Untuk menjalankan JAR yang dibaca secara dinamis, dapat memanfaatkan Java Reflection dan Java Class Loader. Berikut adalah bacaan yang mungkin dapat membantu anda untuk mengerjakan bagian ini:

1. [Java Reflection: Dynamic Class Loading](#)
2. [Load JAR File Dynamically](#)
3. [Class Loader: How it works](#)
4. [Java Class Loader](#)
5. [Building Simple Class Loader \(Part 1\)](#)
6. [Building Simple Class Loader \(Part 2\)](#)

Apabila program di restart, program harus bisa membaca plugin yang sebelumnya di load. Untuk menghapus/mencopot plugin, bisa dilakukan pada halaman settings. Boleh diasumsikan, JAR plugin tidak pernah dipindahkan atau dihapus setelah di load dan plugin tidak bisa di unload kecuali direset (hapus total semua data yang ada).

- Analytics

Salah satu penggunaan dari plugin adalah untuk menambah kegunaan dari sistem utama. Salah satunya adalah *analytics*. Pada halaman dashboard, tambahkan beberapa “widget” untuk *analytics* ketika plugin berhasil di *load*. Hal yang dimaksud dengan *analytics* dapat berupa (bisa menggunakan <https://www.jfree.org/jfreechart/samples.html>):



1. Pie chart sederhana untuk melihat berapa banyak jumlah pengiriman yang berhasil atau gagal.

Penjelasan tentang halaman setting: settings pada program dapat berupa bentuk **JSON**, disimpan pada sebuah file misalnya seperti pemilihan bentuk penyimpanan dan loading dari data store. Pada halaman setting dibuatkan sebuah dropdown pemilihan bentuk format dari data store (XML/JSON) dan selector pilihan tempat penyimpanan atau loading dari plugin.

Kapan plugin di-load? Jika sebelumnya setting belum diset plugin apa saja yang di-load maka tidak perlu dilakukan loading, tetapi ketika menjalankan aplikasi, dan ada beberapa plugin yang di-load, maka ketika aplikasi ditutup dan dibuka kembali, pluginnya masih terpasang, state dari loaded plugin disimpan pada settings.

● Data Store

Dalam pengembangan aplikasi Logistic Management System (LMS) untuk manajemen pengiriman paket, komponen Data Store berperan sebagai lapisan penyimpanan utama bagi data penting seperti informasi pengiriman, kurir, parcel, serta pengaturan aplikasi. Setiap jenis data harus disimpan terpisah sesuai konteksnya – misalnya, data kurir disimpan dalam file yang berbeda dari data parcel.

Data disimpan dalam format JSON atau XML sesuai preferensi. Format penyimpanan wajib menggunakan representasi nilai asli seperti string, integer, dan objek bersarang. Tidak diperbolehkan menyimpan data dalam bentuk byte array hasil serialisasi objek Java, agar file tetap terbaca, mudah diedit, dan kompatibel lintas sistem.

Sistem ini menerapkan [Adapter Pattern](#) untuk memisahkan logika aplikasi dari detail teknis penyimpanan. Library pihak ketiga boleh digunakan untuk parsing JSON/XML,

namun harus dibungkus dalam adapter internal yang mengikuti antarmuka standar dan tidak boleh diakses langsung oleh komponen utama.

Selain itu, untuk setiap objek yang disimpan pada data store, gunakan **Repository Pattern** serta **Generics** untuk mengakses data, sehingga tidak perlu mengimplementasikan operasi CRUD secara berulang untuk setiap jenis *entity*. Repository Pattern berfungsi sebagai abstraksi yang menyediakan interface standar untuk operasi dasar seperti find, save, update, dan delete.

Dengan menggabungkan **Repository Pattern** dan **Generics**, dapat dibuat class seperti `Repository<T>` yang dapat digunakan untuk berbagai jenis entity (Kurir, Parcel, Pengiriman) tanpa duplikasi kode. Implementasi konkret seperti `KurirRepository`, `ParcelRepository`, dan `PengirimanRepository` dapat meng-extend generic repository dan menambahkan metode spesifik yang dibutuhkan untuk masing-masing entity.

Seluruh method pada repository wajib menggunakan **Assertions** untuk memvalidasi parameter input dan kondisi pra-syarat, seperti `assert id != null : "ID tidak boleh null";` atau `assert entity instanceof Kurir : "Entity harus bertipe Kurir";`

Seluruh akses baca dan tulis file hanya dilakukan melalui komponen Data Store. Komponen lain seperti GUI, plugin, maupun business logic hanya boleh berinteraksi dengan data melalui lapisan ini. Pendekatan ini menjaga modularitas sistem, memudahkan pemeliharaan, dan mendukung pengembangan jangka panjang.

Selain itu, ketika program ditutup, semua data yang ada dimemori wajib disimpan dalam bentuk pilihan XML/JSON yang diatur pada halaman settings. Jadi ketika program dibuka kembali tetap memiliki data yang sama seperti sebelumnya.

Bonus

GUI Responsive

Antarmuka aplikasi sebaiknya dirancang responsif untuk menyesuaikan tampilan dengan ukuran layar pengguna, baik saat jendela aplikasi diperbesar, diperkecil, maupun berpindah tab. Komponen seperti tabel, form input, tombol, dan navigasi harus tetap terbaca, proporsional, dan tidak saling bertumpuk. Responsivitas juga mencakup:

- Penyesuaian tata letak (layout) secara otomatis.
- Scroll otomatis untuk konten yang tidak muat di layar.
- Elemen yang fleksibel dan tetap mudah diakses dalam kondisi layar terbatas.

Hal ini akan memberikan pengalaman pengguna yang lebih baik dan profesional, serta menunjukkan perhatian terhadap detail dalam perancangan antarmuka.

Unit Testing

Unit test dibuat untuk menguji kelas-kelas dalam sistem secara terpisah. Framework unit testing yang digunakan dibebaskan. Gunakan coverage tool seperti JaCoCo atau tools serupa untuk menampilkan nilai cakupan (coverage) unit test. Screenshot hasil coverage wajib dilampirkan dalam laporan.

Pemindaian Barcode

Sistem mendukung fitur pemindaian barcode menggunakan kamera untuk mempercepat proses pencatatan barang ke dalam parsel. Hasil pemindaian akan langsung menentukan jenis barang pada parsel.

Beberapa ketentuan teknis:

- Kamera yang digunakan bebas, termasuk kamera laptop maupun kamera eksternal, selama dapat diakses melalui program.
- Library yang digunakan untuk pemrosesan video atau decoding barcode juga dibebaskan, selama dapat membaca barcode secara akurat dari input kamera.
- Format barcode yang didukung adalah EAN-13 atau Code 128, dan Anda dapat memilih salah satu sesuai kebutuhan. Pilihan format harus didemokan saat presentasi.

Plugin Driver Simulator

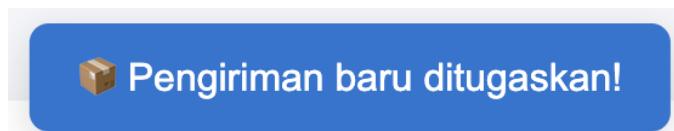
Rancang sebuah plugin yang bisa melakukan simulasi pengiriman, yakni sebuah sistem otomatis yang bisa mengalokasikan pengiriman ke kurir yang tersedia dan menggunakan beberapa delay yang dirandom bisa mengubah status pengiriman, untuk memudahkan, implementasikan pengalokasian sehingga pengiriman berhasil dikirim (statusnya bukan gagal). Cara mengaktifkan plugin ini dengan menggunakan sebuah button yang bisa ditekan pada dashboard atau pada halaman tersendiri.

Reports Generation

Salah satu penggunaan lain, adalah *reports generation*. Ketika plugin ini berhasil di load, maka aplikasi dapat menghasilkan suatu **file PDF** untuk melakukan print sebuah tabel yang merupakan data dari halaman tersebut. Umumnya, karena penghasilan **PDF** memakan waktu yang cukup lama, gunakan **threading** untuk melakukan operasi ini, PDF yang dihasilkan disimpan pada **folder output**. Berikut adalah beberapa yang wajib untuk diimplementasikan:

1. Halaman kurir, tampilkan tabel semua nama gender, dan date of birth.
2. Halaman pengiriman yang bisa di-filter berdasarkan satu kurir atau semua, yang berisi semua field dari pengiriman yang ada.

Notifikasi Parsel



Sistem notifikasi sederhana digunakan agar kurir tidak melewatkkan sebuah tugas baru yang di-assign kepadanya. Akan ditampilkan notifikasi bahwa sebuah parsel baru telah di-assign kepada kurir tersebut. Parsel baru yang ditampilkan notifikasinya adalah parsel yang baru di-assign sejak kurir terakhir kali melakukan Log-In. Dapat menggunakan *background thread* untuk cek apakah ada pengiriman baru yang diassign ke kurir.

Panduan Pengerjaan

Tugas besar ini dilakukan secara berkelompok menggunakan kelompok yang sebelumnya dibuat. Tidak perlu membuat kelompok lagi karena sudah di-assign menggunakan kelompok sebelumnya pada GH classroom. Untuk mengerjakan assignment silahkan gunakan link berikut:

<https://classroom.github.com/a/V9aFwY4v>

Pada repository tersebut, sudah disediakan template yang bisa di **RUN** langsung dengan panduannya untuk menggunakan JavaFX sebagai GUI. Semoga ini dapat membantu mempercepat pengerjaan tugas besar.

Semua link kebutuhan praktikum ada di drive ini:  Public

Adapun ketentuan seperti berikut.

1. JAR File Wajib bisa dijalankan di sistem operasi **LINUX**.
 - a. Akan ada **pengurangan nilai** jika terjadi anomali.
 - b. Berhak tidak dinilai jika program **tidak jalan total** atau gagal mensimulasikan **majoritas kondisi** untuk test case yang dibuat asisten.
 - c. Windows Subsystem Linux atau **WSL** diperbolehkan untuk substitusi sistem operasi **LINUX**.
 - d. Dihimbau untuk setiap kelompok mengerjakan di **WSL** atau **Linux** daripada diakhir saat demo tidak jalan sama sekali/ada kasus unik.
2. Diperbolehkan memakai versi java **8** atau **diatasnya** (11, 17, 19, dsb.)
 - a. Harus jalan di **salah satu versi Java** 8/9/11/15/17/19. Tuliskan versi Java yang kalian pakai sebagai pada java-version.txt pada zip pengumpulan.
3. Aplikasi GUI harus desktop app **non web based** tidak boleh diakali seperti menggunakan *webview* dari Java untuk menampilkan tampilan HTML & CSS.
4. Tidak boleh plagiat dari internet maupun kelompok lain.
5. Jangan asal spesifikasi selesai atau program jalan. Nilai bisa hancur meskipun selesai namun desain jelek dan tidak memaksimalkan OOP. Terlebih lagi, nilai individu bisa jelek kalau tidak paham apa-apa mengenai hal yang dikerjakan (bantuan *copilot*, *chatgpt*, dan teman diperbolehkan selama tidak berlebihan dan masih paham).

Panduan Laporan

Sebagai programmer yang baik, Anda dilatih tidak hanya untuk membuat kode, tetapi juga merancang sehingga program kalian *maintainable*. Pembuatan laporan dapat mengikuti format berikut  IF2010_TB2_Laporan_XXX .

Asistensi & QnA

Asistensi bersifat **optional** namun sangat dianjurkan untuk melakukan asistensi untuk melakukan *alignment* terkait arsitektur kelas atau aplikasi yang masih kurang jelas. Asistennya tetap sama seperti sebelumnya, komunikasikan jadwal dengan asisten terlebih dahulu jika mau melakukan asistensi. Pertanyaan terkait dengan spesifikasi yang belum jelas atau hal-hal yang berkaitan dengan masalah teknis pada tugas besar dapat ditanyakan pada borang QnA

 IF2010 OOP - QnA Tugas Besar 2

Pengumpulan dan Deadline

DEADLINE: Minggu, 8 Juni 2025 23.59

Laporan dan kode program dikumpulkan bersamaan dengan rilis di Github Classroom sesuai deadline.

Semangat, ini tubes terakhir semester ini 🔥

- Asisten Labpro 24/25