

# **TUGAS 3**

## **CLASS DIAGRAM**

**SIMADA**  
**(Sistem Manajemen Sumber Daya Wakanda)**

untuk:

Auralea Alvinia S

Kelompok 11

Ferdinand Gabe Tua Sinaga	13523051
Muhammad Aufa Farabi	13523023
Buege Mahara Putra	13523037
Ferdin Arsenarendra Purtadi	13523117
Muhammad Iqbal Haidar	13523111

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**  
**INSTITUT TEKNOLOGI BANDUNG**  
**JL. GANESA 10, BANDUNG 40132**

**2024**

## **Daftar Isi**

<b>Daftar Isi</b>	<b>2</b>
<b>Daftar Perubahan</b>	<b>3</b>
<b>Daftar Tabel</b>	<b>4</b>
<b>Daftar Gambar</b>	<b>5</b>
<b>1 Deskripsi Umum</b>	<b>6</b>
1.1 Deskripsi Umum Sistem	6
1.2 Deskripsi Pengguna Perangkat Lunak	6
<b>2 Deskripsi Kebutuhan Perangkat Lunak</b>	<b>7</b>
2.1 Kebutuhan Fungsional	7
2.2 Kebutuhan Non Fungsional	8
<b>3 Model Use Case</b>	<b>9</b>
3.1 Identifikasi Actor	9
3.2 Identifikasi Use Case	9
3.3 Diagram Use Case	10
3.4 Skenario Use Case	10
3.4.1 Skenario Use Case 01	10
3.4.2 Skenario Use Case 02	11
3.4.3 Skenario Use Case 03	13
3.4.4 Skenario Use Case 04	13
3.4.5 Skenario Use Case 05	14
<b>4 Diagram Kelas</b>	<b>15</b>
4.1 Use Case Melihat Sumber Daya	15
4.1.1 Identifikasi Kelas Use Case 01	15
4.1.2 Diagram Kelas Use Case 01	15
4.2 Use Case Mengelola Rekaman Sumber Daya	16
4.2.1 Identifikasi Kelas Use Case 02	16
4.2.2 Diagram Kelas Use Case 02	16
4.3 Use Case Melihat Penggunaan Sumber Daya	17
4.3.1 Identifikasi Kelas Use Case 3	17
4.3.2 Diagram Kelas Use Case 03	17
4.4 Use Case Melihat Inventaris Sumber Daya	18
4.4.1 Identifikasi Kelas Use Case 04	18
4.4.2 Diagram Kelas Use Case 04	18
4.5 Use Case Mengelola Laporan Sumber Daya	19
4.5.1 Identifikasi Kelas Use Case 05	19
4.5.2 Diagram Kelas Use Case 05	19
4.6 Diagram Kelas Keseluruhan	20
<b>5 Traceability</b>	<b>24</b>

## Daftar Perubahan

Revisi	Deskripsi
A	Merging use case scenario, merging use case
B	Memperbaiki diagram penamaan diagram kelas dan tipe dari atribut yang dimiliki oleh kelas
C	
D	
E	

## Daftar Tabel

Tabel 1.2 Tabel Jenis Pengguna P/L	6
Tabel 2.1. Tabel Kebutuhan Fungsional	7
Tabel 2.2. Tabel Kebutuhan Non Fungsional	8
Tabel 3.1. Identifikasi Actor	9
Tabel 3.2. Identifikasi Use Case	9
Tabel 3.4.1 Skenario Use Case 01	10
Tabel 3.4.2 Skenario Use Case 02	11
Tabel 3.4.3 Skenario Use Case 03	13
Tabel 3.4.4 Skenario Use Case 04	13
Tabel 3.4.5 Skenario Use Case 05	14
Tabel 4.1.1 Kelas Use Case 01	15
Tabel 4.1.2 Tabel Diagram Kelas Use Case 01	15
Tabel 4.2.1 Kelas Use Case 02	16
Tabel 4.2.2 Tabel Diagram Kelas Use Case 02	16
Tabel 4.3.1 Kelas Use Case 03	17
Tabel 4.3.2 Tabel Diagram Kelas Use Case 03	18
Tabel 4.4.1 Kelas Use Case 04	18
Tabel 4.4.2 Tabel Diagram Kelas Use Case 04	19
Tabel 4.5.1 Kelas Use Case 05	19
Tabel 4.5.2 Tabel Diagram Kelas Use Case 05	20
Tabel 4.6 Tabel Diagram Kelas Keseluruhan	21
Tabel 5.1 Traceability Kelas dengan Kebutuhan Fungsional dan UCT	24

## **Daftar Gambar**

Gambar 3.3 Diagram Use Case	10
Gambar 4.1.2 Diagram Kelas Use Case 01	15
Gambar 4.2.2 Diagram Kelas Use Case 02	16
Gambar 4.3.2 Diagram Kelas Use Case 03	17
Gambar 4.4.2 Diagram Kelas Use Case 04	18
Gambar 4.5.2 Diagram Kelas Use Case 05	19
Gambar 4.6 Diagram Kelas Keseluruhan	20

# 1 Deskripsi Umum

## 1.1 Deskripsi Umum Sistem

Aplikasi SIMADA (Sistem Manajemen Sumber Daya Wakanda) merupakan suatu solusi yang dirancang untuk menangani masalah pengelolaan dan pemantauan alokasi sumber daya di Wakanda. Aplikasi ini diharapkan dapat memenuhi kebutuhan pengguna utama, yaitu admin, agar ia dapat memantau dan mengelola seluruh sumber daya strategis Wakanda, seperti vibranium. Melalui aplikasi ini, diharapkan manajemen sumber daya Wakanda menjadi lebih efisien, terstruktur, dan dapat diawasi dengan lebih baik.

SIMADA menyediakan berbagai fitur yang mempermudah proses manajemen sumber daya, mulai dari pencatatan, pelacakan, hingga pelaporan secara real-time. Dengan antarmuka yang intuitif, admin dapat mengelola inventaris sumber daya, memastikan alokasinya tetap efisien dan aman. Aplikasi ini juga memungkinkan admin untuk memantau perubahan jumlah dan lokasi penyimpanan, serta menghasilkan laporan yang komprehensif mengenai penggunaan dan distribusi sumber daya. Akses 24/7 ke aplikasi ini memastikan bahwa sumber daya strategis Wakanda selalu berada dibawah pengawasan yang ketat, meningkatkan efisiensi pengelolaan, serta meminimalkan kesalahan manusia.

Selain itu, aplikasi ini dirancang untuk penggunaan pribadi dengan akses offline tanpa memerlukan registrasi atau login sehingga keamanan dan privasi data dapat terjamin dengan baik. Dengan pendekatan ini, setiap langkah pengelolaan sumber daya dilakukan secara lebih efektif dan terorganisir. Diharapkan SIMADA akan meningkatkan efisiensi dan akurasi dalam pengelolaan sumber daya Wakanda dan SIMADA akan memastikan setiap aspek manajemen berjalan dengan lebih baik untuk mendukung stabilitas dan kesejahteraan Wakanda.

## 1.2 Deskripsi Pengguna Perangkat Lunak

Tabel 1.2 Tabel Jenis Pengguna P/L

Pengguna	Deskripsi
Admin user	Admin memiliki hak akses penuh terhadap semua fitur dan seluruh entri data untuk melakukan pemantauan dan pemeliharaan sumber daya di Wakanda.

## 2 Deskripsi Kebutuhan Perangkat Lunak

### 2.1 Kebutuhan Fungsional

Tabel 2.1 Tabel Kebutuhan Fungsional

ID	Kebutuhan	Penjelasan
F01	Dapat membuat catatan atau rekaman mengenai sumber daya baru termasuk jumlah dan lokasi.	Pengguna dapat membuat rekaman baru untuk sumber daya dengan memasukkan informasi seperti jumlah dan lokasi penyimpanan sumber daya tersebut. Aplikasi harus menyediakan antarmuka yang intuitif untuk memfasilitasi input data secara efisien.
F02	Dapat melihat penggunaan sumber daya	P/L dapat memberi informasi tentang sejarah penggunaan suatu sumber daya sejak awal dimasukkan, meliputi perubahan jumlah dan lokasinya.
F03	Dapat memperlihatkan inventaris	User dapat melihat secara detail jumlah dan lokasi sumber daya apa saja yang masih tersimpan.
F04	Dapat memberikan laporan mengenai sumber daya	P/L memberikan sebuah text field yang dapat digunakan oleh user untuk melakukan pencatatan atau melaporkan penggunaan sumber daya.
F05	Dapat menghapus rekaman data alokasi sumber daya	Pengguna diberi kemampuan untuk menghapus rekaman sumber daya yang sudah tidak relevan atau tidak diperlukan lagi dari sistem inventaris, dengan proses konfirmasi penghapusan untuk mencegah kesalahan.
F06	Dapat memperbaiki data alokasi sumber daya yang sudah ada dengan mengedit jumlah dari sumber daya yang dialokasikan	User dapat memperbaiki informasi pada data sumber daya yang sudah ada, seperti mengubah jumlah dari penyimpanan sumber daya. Aplikasi harus menyediakan mekanisme untuk melakukan pembaruan secara aman dan efisien.
F07	Dapat mengubah data lokasi sumber daya.	User dapat memperbaiki informasi perihal data lokasi penyimpanan sumber daya yang sudah ada menuju tempat lain.
F08	Dapat menghapus laporan sumber daya	User dapat menghapus laporan yang sudah ada mengenai sumber daya jika diinginkan.
F09	Dapat memperbaiki laporan sumber daya	User dapat memperbaiki data-data laporan yang sudah ada mengenai sumber daya jika memang diperlukan perubahan agar pemantauan menjadi lebih akurat.
F10	Dapat menampilkan riwayat penggunaan sumber daya	P/L dapat menampilkan apa saja yang telah dilakukan terhadap sumber daya spesifik, misalnya lokasi penyimpanan, waktu pengecekan, stok dan kondisi barang, dan jenis aktivitas terakhir yang terjadi pada barang(pemasukan, pengeluaran, distribusi).

## 2.2 Kebutuhan Non Fungsional

Tabel 2.2 Tabel Kebutuhan Non Fungsional

ID	Parameter	Kebutuhan
NF01	Availability	Aplikasi dirancang agar dapat diakses selama 24/7.
NF02	Reliability	Aplikasi dirancang memiliki tingkat keberhasilan minimal 95% dalam melaksanakan perintah pengguna.
NF03	Ergonomy	Penggunaan warna, tata letak, dan ukuran teks harus disesuaikan untuk meminimalkan kelelahan pengguna, terutama saat digunakan dalam jangka waktu yang lama. Selain itu, aplikasi harus mendukung aksesibilitas yang baik, memastikan bahwa berbagai kelompok pengguna dapat berinteraksi dengan sistem tanpa hambatan.
NF04	Portability	Aplikasi dirancang untuk dapat dijalankan secara offline serta dapat dioperasikan minimal pada 2 device berbeda.
NF05	Memory	Aplikasi dirancang untuk mengelola penggunaan memori agar tetap bisa berjalan pada perangkat dengan spesifikasi rendah (minimal RAM 4GB).
NF06	Response time	Aplikasi dirancang agar dapat menampilkan entri kurang dari 4 detik.
NF07	Security	Aplikasi dirancang untuk mengenkripsi entri data yang tersimpan dalam database menjadi cipher text.
NF08	Robustness	Semua input dari pengguna harus divalidasi untuk mencegah kesalahan atau data yang tidak valid memasuki sistem. Input yang tidak valid harus ditolak dengan pesan kesalahan.
NF09	Bahasa komunikasi	Semua komunikasi dan perintah dalam aplikasi dalam bahasa Indonesia.



### 3 Model Use Case

#### 3.1 Identifikasi Actor

Tabel 3.1 Identifikasi Actor

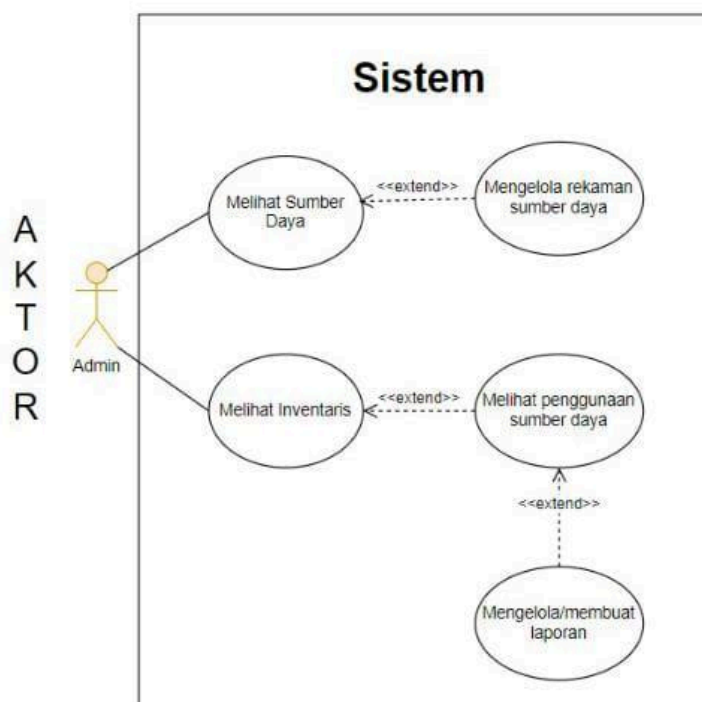
Kode Actor	Nama	Keterangan
A01	Admin	Memiliki akses penuh untuk melakukan semua tindakan dalam sistem, termasuk: <ul style="list-style-type: none"><li>• Menambah, mengedit, dan menghapus data sumber daya.</li><li>• Mengelola inventaris dan alokasi sumber daya.</li><li>• Membuat dan meninjau laporan penggunaan dan distribusi sumber daya.</li><li>• Menerima notifikasi terkait status inventaris</li></ul>

#### 3.2 Identifikasi Use Case

Tabel 3.2. Identifikasi Use Case

No	Use Case	Keterangan
UC01	Melihat Sumber Daya	Admin dapat melihat semua sumberdaya yang sedang tersedia secara realtime.
UC02	Mengelola rekaman sumber daya	Admin dapat menambah, menghapus, mengubah jumlah, mengubah lokasi dan mengalokasikan rekaman sumber daya untuk kemudian diberikan kepada sistem untuk disimpan dalam basis data..
UC03	Melihat penggunaan sumber daya	Sistem menampilkan riwayat penggunaan atau riwayat distribusi sumber daya terkait sedari awal.
UC04	Melihat inventaris sumber daya	Sistem menampilkan informasi jumlah dan lokasi penyimpanan sumber daya terkait.
UC05	Mengelola laporan sumber daya	Admin dapat membuat, menghapus, dan mengubah isi laporan untuk kemudian disimpan oleh sistem.

### 3.3 Diagram Use Case



Gambar 3.3 Diagram Use Case

### 3.4 Skenario Use Case

#### 3.4.1 Skenario Use Case 01

Nama Use Case: Melihat Sumber Daya

Skenario:

Tabel 3.4.1 Skenario Use Case 01

Aksi Actor	Reaksi Sistem
Skenario Normal	
1. Masuk ke dalam Homepage	
	2. Menampilkan seluruh sumber daya yang tersedia

### 3.4.2 Skenario Use Case 02

Nama Use Case: Mengelola Rekaman Sumber Daya  
Skenario:

Tabel 3.4.2 Skenario Use Case 02

Aksi Actor	Reaksi Sistem
Skenario Normal [TAMBAH]	
1. Memilih menu Tambah Sumber daya	
	2. Menampilkan form yang akan diisi dengan nama dari sumber daya
3. Memasukkan nama sumber daya dan menekan tombol tambah pada form tersebut	
	4. Menampilkan form yang akan diisi dengan jumlah dari sumber daya yang diinginkan beserta tempat penyimpanannya.
5. Memasukkan informasi yang dibutuhkan dan menekan tambah sumber daya	
	6. Mengupdate informasi sumber daya baru pada basis data
Skenario Alternatif (1)	
	4. Mengeluarkan pesan “Sumber Daya yang Anda maksud sudah ada. Silahkan cek kembali Sumber Daya yang anda inginkan”
Skenario Alternatif (2)	
5. Menekan tombol batal	
	6. Mengeluarkan pesan “Proses Penambahan Jenis Sumber Daya Dibatalkan”
Skenario Alternatif (3) [HAPUS]	
1. Memilih tombol hapus di home page	
	2. Menampilkan form dimana user dapat memilih sumber daya mana saja yang ingin dihapus
3. User memilih sumber daya yang ingin dihapus	
4. User menekan tombol hapus	
	5. Menghapus rekaman data sumber daya yang dipilih pada basis data
Skenario Alternatif (4)	
	2. Menampilkan pesan “Anda Tidak memiliki Sumber Daya apapun “
Skenario Alternatif (5)	
3. Menekan tombol batal	

	4. Mengeluarkan pesan “Proses Penghapusan Sumber Daya Dibatalkan”
	4. Mengeluarkan pesan “Proses Penghapusan Sumber Daya Dibatalkan”
Skenario Alternatif (6) [UBAH JUMLAH]	
1. Memilih tombol ubah jumlah di sumber daya yang diinginkan	
	2. Menampilkan form untuk memasukkan jumlah yang akan ditambah atau dikurangi
3. Menuliskan berapa jumlah yang akan ditambah/dikurang	
4. User memilih tombol tambah/kurang	
	5. Mengupdate jumlah terbaru dari sumber daya pada basis data dan menampilkan jumlah terbarunya di homepage
Skenario Alternatif (7)	
	5. Menampilkan Pesan “Jumlah barang yang akan dihapus melebihi jumlah sumber daya saat ini”
Skenario Alternatif (8)	
3. Menekan tombol batal	
	4. Sistem mengeluarkan pesan “Proses Perubahan Jumlah Sumber Daya Dibatalkan”
Skenario Alternatif (9) [UBAH LOKASI]	
1. Memilih sumber daya yang ingin diubah lokasinya, lalu menekan tombol Ubah Lokasi Penyimpanan	
	2. Menampilkan form lokasi penyimpanan yang baru
3. Mengisi form dengan lokasi penyimpanan yang baru	
	4. Mengupdate lokasi penyimpanan sumber daya terkait pada database
Skenario Alternatif (10)	
3. Menekan tombol batal	
	4. Sistem mengeluarkan pesan “Proses Perubahan Lokasi Penyimpanan Sumber Daya Dibatalkan”
Skenario Alternatif (11) [ALOKASI]	
1. Memilih tombol alokasi di sumber daya yang diinginkan	
	2. Menampilkan form untuk memasukkan jumlah yang akan dialokasi atau dealokasi
3. Menuliskan berapa jumlah yang akan dialokasi atau dealokasi	
4. User memilih tombol dialokasi atau dealokasi	

	5. Mengupdate alokasi terbaru dari sumber daya pada basis data dan menampilkannya di homepage
Skenario Alternatif (12)	
	5. Menampilkan Pesan “Jumlah barang yang akan dialokasi melebihi jumlah sumber daya saat ini”
Skenario Alternatif (13)	
3. Menekan tombol batal	
	4. Sistem mengeluarkan pesan “Proses Alokasi Sumber Daya Dibatalkan”

### 3.4.3 Skenario Use Case 03

Nama Use Case: Melihat Penggunaan Sumber Daya

Skenario:

Tabel 3.4.3. Skenario Use Case 03

Aksi Actor	Reaksi Sistem
Skenario Normal	
1. Memilih Sumber Daya yang diinginkan di Halaman Utama, lalu menekan pilihan riwayat	
	2. Menampilkan Page Riwayat Sumber Daya
3. Menekan tombol navigasi prev	
	4. Menampilkan Homepage

### 3.4.4 Skenario Use Case 04

Nama Use Case: Melihat Inventaris Sumber Daya

Skenario:

Tabel 3.4.4 Skenario Use Case 04

Aksi Actor	Reaksi Sistem
Skenario Normal	
1. Memilih Sumber Daya yang diinginkan di Halaman Utama, lalu menekan pilihan inventaris	
	2. Menampilkan Page Inventaris Sumber Daya
3. Menekan tombol navigasi prev	
	4. Menampilkan Homepage

### 3.4.5 Skenario Use Case 05

Nama Use Case: Mengelola Laporan Sumber Daya  
Skenario:

Tabel 3.4.5 Skenario Use Case 05

Aksi Actor	Reaksi Sistem
Skenario Normal [TAMBAH]	
1. Menekan tombol tambah laporan	
	2. Menampilkan Form yang bisa diisi dengan laporan yang diinginkan
3. Menulis isi laporan pada form	
4. Menekan tombol tambah	
	5. Memunculkan pesan “Anda berhasil menambahkan laporan”
Skenario Alternatif (1)	
3. Menekan tombol batal	
	4. Memunculkan pesan “Laporan batal ditambahkan”
Skenario Alternatif (2) [HAPUS]	
1. Menekan tombol hapus laporan	
	2. Menampilkan laporan yang dapat dipilih dan dihapus oleh user
3. Memilih laporan yang ingin dihapus	
4. Menekan tombol hapus	
	5. Menghapus laporan yang dipilih pada basis data
Skenario Alternatif (3)	
	2. Menampilkan pesan “Anda belum memiliki laporan apapun “
Skenario Alternatif (4)	
3. Menekan tombol batal	
	4. Mengeluarkan pesan “Proses Penghapusan Laporan Dibatalkan”
Skenario Alternatif (5) [UBAH]	
1. Memilih menu Ubah Laporan	
	2. Menampilkan daftar laporan dari basis data ke layar
3. Memilih laporan yang ingin diubah	
	4. Menampilkan laporan terkait dalam form
5. Melakukan perubahan terhadap isi laporan	
	6. Menyimpan isi laporan terbaru ke dalam basis data
Skenario Alternatif (6)	

	2. Menampilkan pesan “Anda belum memiliki laporan”
Skenario Alternatif (7)	
3. Menekan tombol batal	
	4. Mengeluarkan pesan “Proses Perubahan Laporan Dibatalkan”

## 4 Diagram Kelas

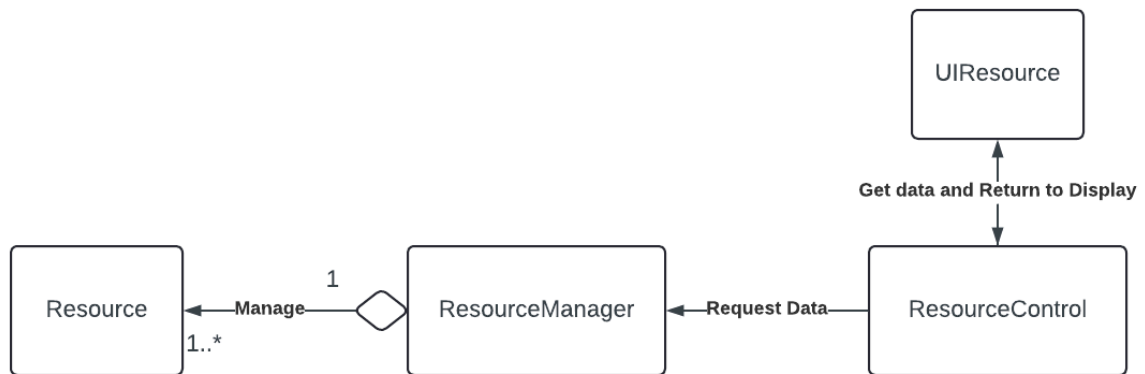
### 4.1 Use Case Melihat Sumber Daya

#### 4.1.1 Identifikasi Kelas Use Case 01

Tabel 4.1.1 Kelas Use Case 01

No	Nama Kelas	Jenis (Boundary, Controller, Entity)
1	<i>UIResource</i>	<i>Boundary</i>
2	<i>ResourceControl</i>	<i>Controller</i>
3	<i>ResourceManager</i>	<i>Controller</i>
4.	<i>Resource</i>	<i>Entity</i>

#### 4.1.2 Diagram Kelas Use Case 01



Gambar 4.1.2 Diagram Kelas Use Case 01

Tabel 4.1.2 Tabel Diagram Kelas Use Case 01

ID Kelas	Nama Kelas	Atribut	Metode
C-01	<i>UIResource</i>	<ul style="list-style-type: none"> <li>• <i>-ResourceControl</i></li> </ul>	<i>+RequestAllResource</i> <i>+ShowAllResource</i>
C-02	<i>ResourceControl</i>	<ul style="list-style-type: none"> <li>• <i>-ResourceManager</i></li> </ul>	<i>+GetAllResource(manager: ResourceManager)</i>

C-03	<i>ResourceManager</i>	<ul style="list-style-type: none"> <li>• <i>-ResourceList</i></li> </ul>	<i>+GetResourceList(ResourceList)</i>
C-04	<i>Resource</i>	<ul style="list-style-type: none"> <li>• <i>-Resource</i></li> </ul>	<i>+GetResourceQuantity</i>

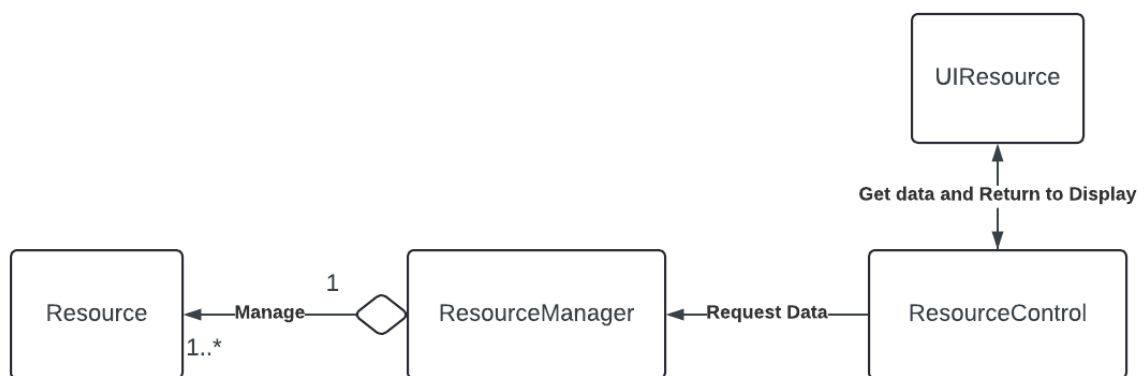
## 4.2 Use Case Mengelola Rekaman Sumber Daya

### 4.2.1 Identifikasi Kelas Use Case 02

Tabel 4.2.1 Kelas Use Case 02

No	Nama Kelas	Jenis (Boundary, Controller, Entity)
1	<i>UIResource</i>	<i>Boundary</i>
2	<i>ResourceControl</i>	<i>Controller</i>
3	<i>Resource</i>	<i>Entity</i>
4.	<i>ResourceManager</i>	<i>Controller</i>

### 4.2.2 Diagram Kelas Use Case 02



Gambar 4.2.2 Diagram Kelas Use Case 02

Tabel 4.2.2 Tabel Diagram Kelas Use Case 02

ID Kelas	Nama Kelas	Atribut	Metode
C-01	<i>UIResource</i>	<ul style="list-style-type: none"> <li>• <i>-formData</i></li> </ul>	<i>+formCreateResource</i> <i>+formReduceQuantity</i> <i>+formAddQuantity</i> <i>+RequestStockDetails</i>
C-02	<i>ResourceControl</i>	<ul style="list-style-type: none"> <li>• <i>-InputValidation</i></li> <li>• <i>-ResourceManager</i></li> </ul>	<i>+AddResource</i> <i>+RemoveResource</i> <i>+UpdateResourceLocation</i> <i>+ShowCurrentStockForResource</i> <i>+Alloc</i>



C-03	Resource	<ul style="list-style-type: none"> <li>• -resourceID</li> <li>• -nama</li> <li>• -jumlah</li> </ul>	+UpdateQuantity +UpdateLoc
C-04	ResourceManager	<ul style="list-style-type: none"> <li>• -ResourceByLoc</li> </ul>	+CreateResource +UpdateResource +GetLoc +DeleteResource +getStockDetails +allocResource +deallocResource

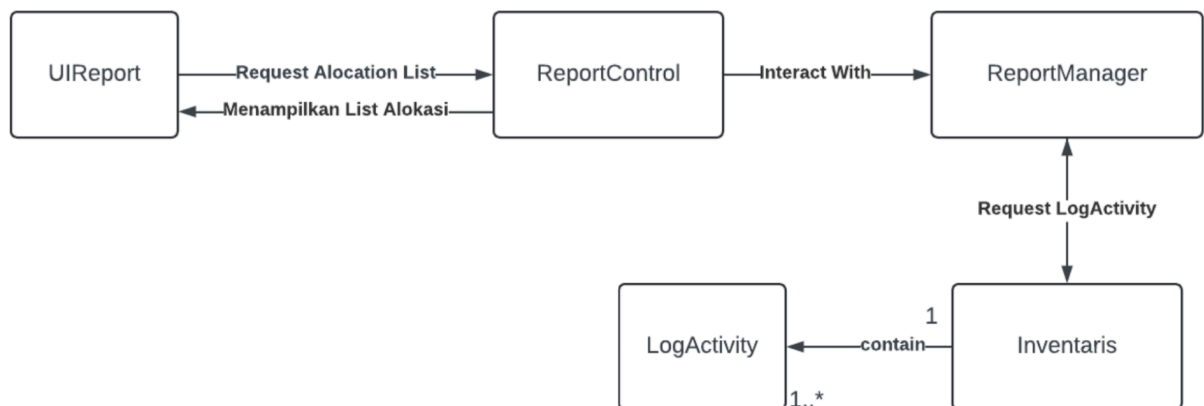
### 4.3 Use Case Melihat Penggunaan Sumber Daya

#### 4.3.1 Identifikasi Kelas Use Case 03

Tabel 4.3.1 Kelas Use Case 03

No	Nama Kelas	Jenis (Boundary, Controller, Entity)
1	UIResource	Boundary
2	ResourceControl	Controller
3	ResourceManager	Controller
4.	Inventaris	Entity
5.	LogActivity	Entity

#### 4.3.2 Diagram Kelas Use Case 03



Gambar 4.3.2 Diagram Kelas Use Case 03

Tabel 4.3.2 Tabel Diagram Kelas Use Case 03

ID Kelas	Nama Kelas	Atribut	Metode
C-05	UIReport	<ul style="list-style-type: none"> <li>• -ResourceControl</li> </ul>	+SeeResourceActivity +ViewDetailAllocationLog
C-02	ResourceControl	<ul style="list-style-type: none"> <li>• -ResourceManager</li> </ul>	+getResourceActivityLog
C-04	ResourceManager	<ul style="list-style-type: none"> <li>• -Inventaris</li> </ul>	+requestLogActivity +updateLogActivity
C-06	Inventaris	<ul style="list-style-type: none"> <li>• -LogActivity</li> </ul>	+GetActivityLogsForResource
C-09	LogActivity	<ul style="list-style-type: none"> <li>• -ResourceName</li> <li>• -ActionType (string)</li> <li>• -Timestamp</li> <li>• -Jumlah</li> <li>• -Lokasi</li> </ul>	+logNewActivity

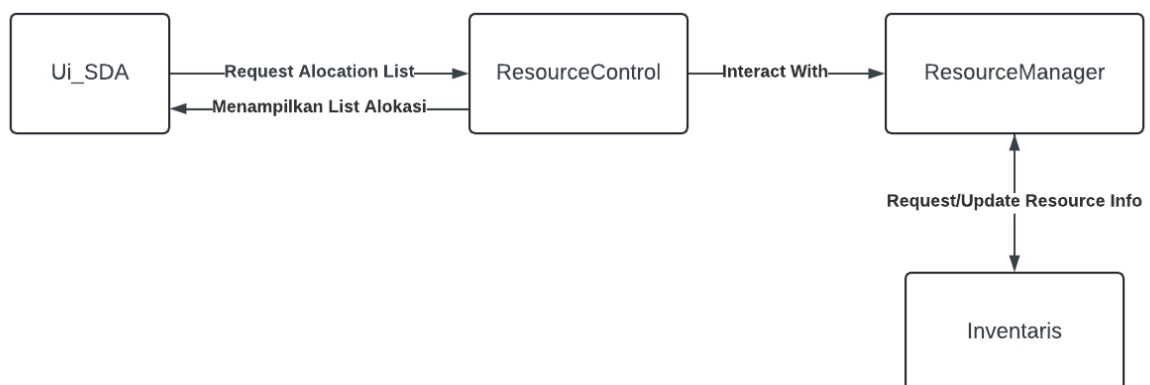
## 4.4 Use Case Melihat Inventaris Sumber Daya

### 4.4.1 Identifikasi Kelas Use Case 04

Tabel 4.4.1 Kelas Use Case 04

No	Nama Kelas	Jenis (Boundary, Controller, Entity)
1	UIResource	Boundary
2	ResourceControl	Controller
3	ResourceManager	Controller
4.	Inventaris	Entity

### 4.4.2 Diagram Kelas Use Case 04



Gambar 4.4.2 Diagram Kelas Use Case 04

Tabel 4.4.2 Tabel Diagram Kelas Use Case 04

ID Kelas	Nama Kelas	Atribut	Metode
C-01	UIResource	<ul style="list-style-type: none"> <li>-ResourceControl</li> </ul>	+RequestStockDetails
C-02	ResourceControl	<ul style="list-style-type: none"> <li>-ResourceManager</li> </ul>	+ShowCurrentStockForResource
C-03	ResourceManager	<ul style="list-style-type: none"> <li>-Inventaris</li> </ul>	+GetStockDetails
C-06	Inventaris	-Allocation	+ LogAllocation +getCurrentStockForResource

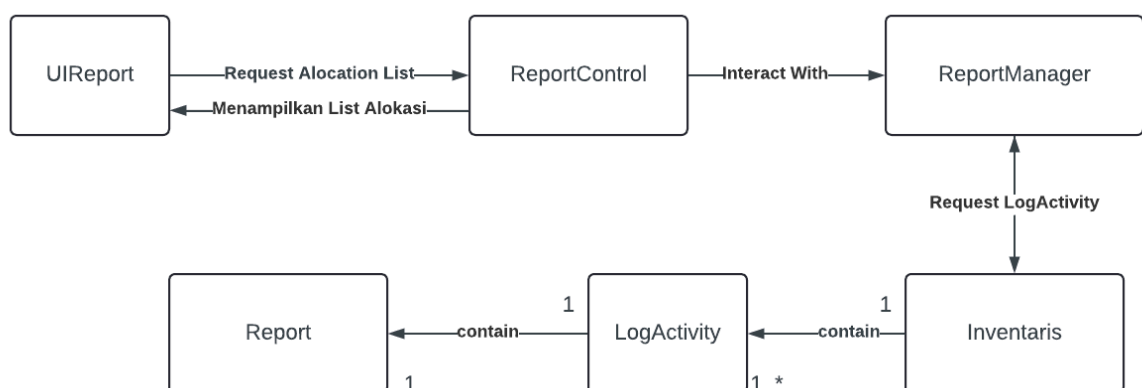
## 4.5 Use Case Mengelola Laporan Sumber Daya

### 4.5.1 Identifikasi Kelas Use Case 05

Tabel 4.5.1 Kelas Use Case 05

No	Nama Kelas	Jenis (Boundary, Controller, Entity)
1	UIReport	Boundary
2	ReportControl	Controller
3	LogActivy	Entity
4.	ReportManager	Controller
5.	Report	Entity
6.	Inventaris	Entity

### 4.5.2 Diagram Kelas Use Case 05

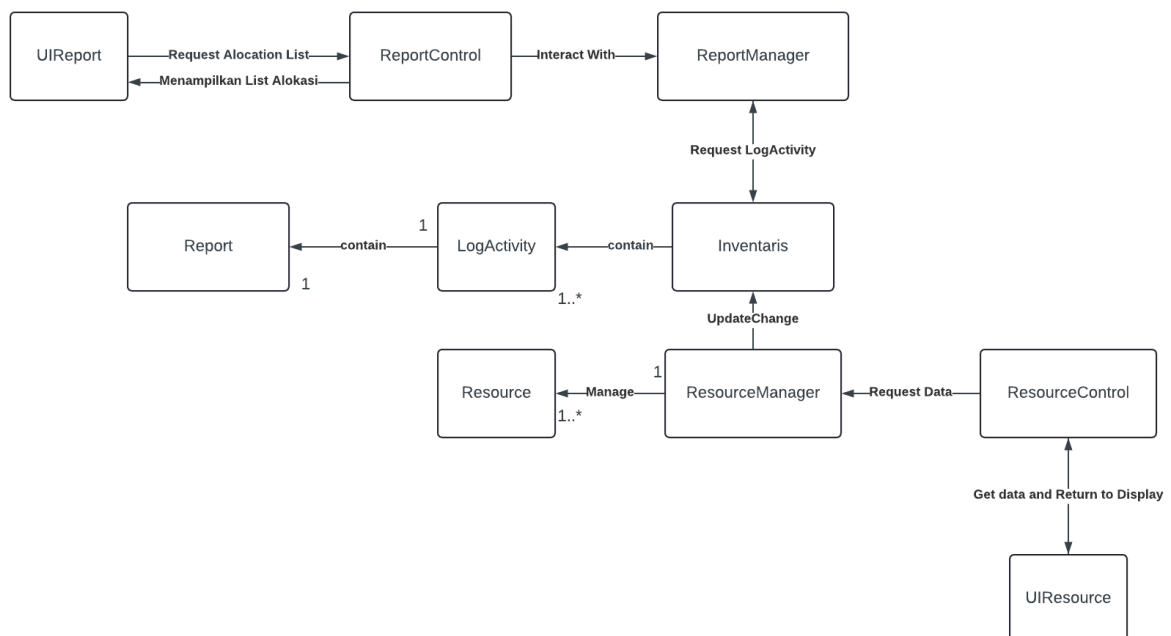


Gambar 4.5.2 Diagram Kelas Use Case 05

Tabel 4.5.2 Tabel Diagram Kelas Use Case 05

ID Kelas	Nama Kelas	Atribut	Metode
C-05	UIReport	-ReportControl	+TampilkanFormLaporan +TampilkanLaporan
C-06	Inventaris	-LogActivity	+GetActivityLogsForResource
C-07	ReportControl	-ReportManager	+MakeReport +GetReport +DeleteReport
C-08	ReportManager	-LogActivity	+SaveReport +DeleteReportfromLog
C--09	LogActivity	<ul style="list-style-type: none"> <li>• -ResourceName</li> <li>• -ActionType (string)</li> <li>• -Timestamp</li> <li>• -Quantity</li> <li>• -Location</li> <li>• -Report</li> </ul>	+AddReport +EditReport +DeleteReport +SeeReport
C-10	Report	<ul style="list-style-type: none"> <li>• -ReportDetail</li> </ul>	+Report +GetReportDetail +DeleteReport

#### 4.6 Diagram Kelas Keseluruhan



Gambar 4.6 Diagram kelas keseluruhan

Tabel 4.6 Tabel Diagram Kelas Keseluruhan

ID Kelas	Nama Kelas	Atribut	Metode
C-01	UIResource	<ul style="list-style-type: none"> <li>• -formData</li> <li>• -ResourceControl</li> </ul>	<ul style="list-style-type: none"> <li>• +formCreateResource</li> <li>• +formReduceQuantity</li> <li>• formAddQuantity <ul style="list-style-type: none"> <li>• +RequestStockDetails</li> </ul> </li> </ul>
C-02	ResourceControl	<ul style="list-style-type: none"> <li>• -InputValidation</li> <li>• -ResourceManager</li> </ul>	<ul style="list-style-type: none"> <li>• +AddResource(resourceName: String, resourceQuantity: int, resourceLocation: String) → ResourceTerdefinisi</li> <li>• +RemoveResource(resourceName: String) → ResourceTerhapus</li> <li>• +UpdateResourceQuantity(resourceName: String, newQuantity: int) → QuantityResourceBerkurang <ul style="list-style-type: none"> <li>• +UpdateResourceLocation(resourceName: String, resourceLocation: string) → ResourceLocationBerubah</li> </ul> </li> <li>• +ListResources()</li> <li>• +GetResourceActivityLog(ResourceManager, resourceName: String) → AllResourceLog</li> </ul>
C-03	Resource	<ul style="list-style-type: none"> <li>• -resourceID</li> <li>• -nama</li> <li>• -jumlah</li> </ul>	<ul style="list-style-type: none"> <li>• +UpdateQuantity(newQuantity: int) → QuantityResourceTerupdate</li> </ul>

			<ul style="list-style-type: none"> <li>• +UpdateLoc(<i>newLocation</i>: String)→ LocationResourceTerupdate</li> </ul>
C-04	ResourceManager	<ul style="list-style-type: none"> <li>• -ResourceByLoc</li> </ul>	<ul style="list-style-type: none"> <li>• +CreateResource(<i>resourceName</i>: String, <i>resourceQuantity</i>: int)→ ResourceTedefinisi</li> <li>• +GetResourcebyLoc(<i>resourceName</i>: String, <i>resourceLocation</i>: String)→ AllResourceLoc</li> <li>• +AllocateTo(<i>location</i>: String, <i>quantity</i>: int)→ ResourceAllocated</li> <li>• +DistributeTo(<i>destination</i>: String, <i>quantity</i>: int)→ ResourceDistributed</li> <li>• +DeallocateResourceIn(<i>location</i>: String, <i>quantity</i>: int)→ ResourceDeallocated</li> <li>• +UpdateResource(<i>resourceName</i>: String, <i>resourceQuantity</i>: int)→ ResourceQuantityUpdated</li> <li>• +DeleteResource(<i>resourceName</i>: String)</li> <li>• +requestLogActivity(<i>resourceName</i>: String, <i>Inventaris</i>)→ AllResourceLog</li> <li>• +updateLogActivity(<i>resourceName</i>: String, <i>activityDetails</i>: String)→ ListLogUpdated</li> </ul>
C-05	UIReport	<ul style="list-style-type: none"> <li>• -ResourceControl</li> </ul>	<ul style="list-style-type: none"> <li>• +SeeResourceActivity</li> </ul>

			<ul style="list-style-type: none"> <li>• -ViewDetailAllocationLog</li> <li>• +GenerateReport</li> </ul>
C-06	Inventaris	<ul style="list-style-type: none"> <li>• -LogActivity</li> </ul>	<ul style="list-style-type: none"> <li>• +GetActivityLogsForResource(resourceName: String)→ AllResourceLog</li> </ul>
C-07	ReportControl	<ul style="list-style-type: none"> <li>• -ReportManager</li> </ul>	<ul style="list-style-type: none"> <li>• +MakeReport(logActivity: LogActivity, reportDetails: String)→ Report</li> <li>• +GetReport(logActivity: LogActivity)→ AllResourceLog</li> <li>• +DeleteReport(logActivity: LogActivity)</li> </ul>
C-08	ReportManager	<ul style="list-style-type: none"> <li>• -LogActivity</li> </ul>	<ul style="list-style-type: none"> <li>• +SaveReport(logActivity: LogActivity, reportDetails: String)</li> <li>• +DeleteReportfromLog(logActivity: LogActivity)</li> <li>• +GetReportFromLog(logActivity: LogActivity)</li> </ul>
C-09	LogActivity	<ul style="list-style-type: none"> <li>• -ResourceName</li> <li>• -ActionType (string)</li> <li>• -Timestamp</li> <li>• -Jumlah</li> <li>• -Lokasi</li> </ul>	<ul style="list-style-type: none"> <li>• +LogNewActivity(resource: Resource, actionType: String, timestamp: String, jumlah: int, lokasi: String)→ Log</li> <li>• +AddReport(report: Report)→ Report</li> <li>• +EditReport(report: Report)</li> <li>• +DeleteReport(report: Report)</li> <li>• +SeeReport(logactivity: LogActivity, report: Report)</li> </ul>
C-10	Report	<ul style="list-style-type: none"> <li>• -ReportDetail</li> </ul>	<ul style="list-style-type: none"> <li>• +Report(reportDetails: String)→ Log</li> <li>• +GetReportDetail→ Detail</li> <li>• +DeleteReport</li> <li>• +ConfirmDeletion</li> </ul>

## 5 Traceability

*Tabel 5.1 Traceability Kelas dengan Kebutuhan Fungsional dan UC*

Kode Kebutuhan Fungsional	Kode Use Case	Kode Kelas
<i>F01</i>	<i>UC01</i>	<i>C-01, C-02, C-03</i>
<i>F02</i>	<i>UC02</i>	<i>C-01, C-02, C-03, C-04</i>
<i>F03</i>	<i>UC03</i>	<i>C-02, C-04, C-05, C-06,C-09</i>
<i>F04</i>	<i>UC04</i>	<i>C-01, C-02, C-03, C-06</i>
<i>F05</i>	<i>UC05</i>	<i>C-05, C-06, C-07, C-08, C-09, C-10</i>