

TUGAS 7

Deskripsi Perancangan Perangkat Lunak

SIMADA
(Sistem Manajemen Sumber Daya Wakanda)

untuk:

Auralea Alvinia S

Dipersiapkan oleh:

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 Perubahan

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	
G	

Daftar Isi

Daftar Gambar.....	5
1. Pendahuluan.....	6
1.1 Tujuan Penulisan Dokumen.....	6
1.2 Lingkup Masalah.....	6
1.3 Definisi dan Istilah.....	6
1.4 Aturan Penamaan dan Penomoran.....	6
1.5 Referensi.....	6
1.6 Ikhtisar Dokumen.....	7
2 Deskripsi Perancangan Global.....	8
2.1 Rancangan Lingkungan Implementasi.....	8
2.2 Deskripsi Arsitektural.....	8
2.3 Deskripsi Komponen.....	8
3 Perancangan Rinci.....	9
3.1 Realisasi Use Case.....	9
3.1.1 Use Case Melihat Sumber Daya (UC01).....	9
3.1.1.1 Identifikasi Kelas.....	9
3.1.1.2 Sequence Diagram.....	9
3.1.1.3 Diagram Kelas.....	9
3.1.2 Use Case Mengelola Sumber Daya (UC02).....	9
3.1.2.1 Identifikasi Kelas.....	10
3.1.2.2 Sequence Diagram.....	10
3.1.2.3 Diagram Kelas.....	12
3.1.3 Use Case Melihat Penggunaan Sumber Daya (UC03).....	12
3.1.3.1 Identifikasi Kelas.....	12
3.1.3.2 Sequence Diagram.....	12
3.1.3.3 Diagram Kelas.....	13
3.1.4 Use Case Melihat Inventaris Sumber Daya (UC04).....	13
3.1.4.1 Identifikasi Kelas.....	13
3.1.4.2 Sequence Diagram.....	13
3.1.4.3 Diagram Kelas.....	14
3.1.5 Use Case Mengelola Laporan Sumber Daya (UC05).....	14
3.1.5.1 Identifikasi Kelas.....	14
3.1.5.2 Sequence Diagram.....	15
3.1.5.3 Diagram Kelas.....	15
3.2 Perancangan Detil Kelas.....	16
3.2.1 Kelas UIResource.....	16
3.2.2 Kelas ResourceControl.....	17
3.2.3 Kelas ResourceManager.....	17
3.2.3.1 Algoritma/Query.....	18
3.2.3.2 Perancangan Antarmuka.....	19
3.2.4 Kelas Resource.....	21
3.2.5 Kelas Report.....	21

3.2.6 Kelas ReportManager.....	21
3.2.6.1 Algoritma/Query.....	22
3.2.6.2 Perancangan Antarmuka.....	23
3.2.7 Kelas Inventaris.....	23
3.2.7.1 Algoritma/Query.....	24
3.2.7.2 Perancangan Antarmuka.....	25
3.2.8 Kelas LogActivity.....	25
3.3 Perancangan Representasi Persistensi Kelas.....	27
3.4 Diagram Kelas Keseluruhan.....	28
4 Matriks Keruntutan.....	28

Daftar Gambar

Gambar 2.2. Diagram Komponen	8
Gambar 3.1.1.2 Sequence Diagram Use Case 01	9
Gambar 3.1.1.3 Class Diagram Use Case 01	9
Gambar 3.1.2.2 Sequence Diagram Use Case 02	11
Gambar 3.1.2.3 Class Diagram Use Case 02	12
Gambar 3.1.3.2 Sequence Diagram Use Case 03	12
Gambar 3.1.3.3 Class Diagram Use Case 03	13
Gambar 3.1.4.2 Sequence Diagram Use Case 04	13
Gambar 3.1.4.3 Class Diagram Use Case 04	14
Gambar 3.1.5.2 Sequence Diagram Use Case 05	25
Gambar 3.1.5.3 Class Diagram Use Case 05	15
Gambar 3.2.3.2 Antarmuka Daftar Resource SIMADA	19
Gambar 3.2.6.2 Antarmuka Log Activity SIMADA	23
Gambar 3.2.7.1 Antarmuka Inventaris Daftar Resource SIMADA	25

1. Pendahuluan

1.1 Tujuan Penulisan Dokumen

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) ini disusun untuk menjelaskan rancangan perangkat lunak yang akan dikembangkan. Dokumen ini berfungsi sebagai alat komunikasi antar pemangku kepentingan, panduan pengembangan, acuan quality assurance, dan referensi pemeliharaan sistem. Dengan adanya dokumen ini, tim pengembang memiliki acuan yang jelas dalam membangun perangkat lunak sesuai spesifikasi yang diharapkan.

1.2 Lingkup Masalah

SIMADA (Sistem Manajemen Sumber Daya Wakanda) adalah sebuah aplikasi yang dirancang untuk membantu admin atau pengguna dalam mengelola sumber daya strategis secara lebih efektif. Dengan adanya SIMADA, admin dapat lebih mudah memantau, mencatat, dan mengelola alokasi sumber daya, seperti vibranium, untuk memastikan penggunaannya tetap efisien dan optimal. Aplikasi ini juga mendukung admin dalam pengambilan keputusan dengan menyediakan informasi real-time terkait ketersediaan dan distribusi sumber daya. Selain itu, fitur offline dan enkripsi data memungkinkan aplikasi diakses kapan saja dengan jaminan keamanan informasi yang baik.

1.3 Definisi dan Istilah

Singkatan, Akronim, atau Istilah	Penjelasan
SIMADA	Sistem Manajemen Sumber Daya Wakanda, sebuah aplikasi untuk mengelola dan memantau sumber daya strategis,
Admin	Pengguna utama yang bertanggung jawab atas pengelolaan sumber daya, termasuk pencatatan dan pelaporan.
Enkripsi	Proses mengubah data menjadi format yang tidak dapat dibaca tanpa kunci dekripsi, untuk menjaga kerahasiaan informasi.
Real-time Monitoring	Pemantauan secara langsung terhadap data dan aktivitas sumber daya, seperti inventaris dan alokasi.
Use Case	Deskripsi interaksi antara pengguna dan sistem untuk mencapai tujuan tertentu.
Inventaris	Tempat menyimpan data alokasi sumberdaya tertentu.
Log Activity	Catatan atau rekaman yang mencatat setiap aktivitas atau peristiwa yang terjadi pada suatu sumber daya
Akses Offline	Fitur yang memungkinkan penggunaan aplikasi tanpa koneksi internet.

1.4 Aturan Penamaan dan Penomoran

Tabel 1.4. Tabel Aturan Penomoran

Hal/Bagian	Penomoran	Keterangan
Identifikasi Use Case	UCXX	Penomoran identifikasi use case diberi prefiks “UC” dan XX adalah nomor urutan.
Identifikasi Query	Q-XXX	Penomoran identifikasi query diberi prefiks “Q-” dan XXX adalah nomor urutan.
Identifikasi Algoritma	Algo-XXX	Penomoran identifikasi Algoritma diberi prefiks “Algo-” dan XXX adalah nomor urutan.

1.5 Referensi

K01_G11_Final_SKPLOO

Roger S. Pressman; *Software Engineering: A Practitioner's Approach (8th Ed.)*; McGraw-Hill, 2015.

Ian Sommerville, *Software Engineering (10th Ed.)*, Pearson, 2016

1.6 Ikhtisar Dokumen

Dokumen DPPL ini terdiri dari enam bab. Bab 1, Pendahuluan, menjelaskan tujuan penulisan, lingkup masalah, definisi istilah dan singkatan yang digunakan, aturan penomoran, referensi yang dipakai, serta ikhtisar mengenai dokumen DPPL ini. pada Bab 2, Deskripsi Umum Global, menjelaskan rancangan lingkungan implementasi sistem, deskripsi arsitektur yang dibangun, serta deskripsi komponen. Bab 3 menguraikan perancangan rinci dari sistem, yakni identifikasi kelas, sequence diagram, dan diagram kelas dari tiap-tiap use case, diagram kelas secara keseluruhan, serta perancangan detail kelas yang berisi detail operasi dan atribut kelas, algoritma atau query tiap kelas, diagram statechart, perancangan antarmuka. dan Perancangan Representasi Persistensi untuk setiap kelas. Terakhir, Bab 4 berisi matrix kerunutan, yang memperlihatkan hubungan antara kelas dan use case yang terkait.

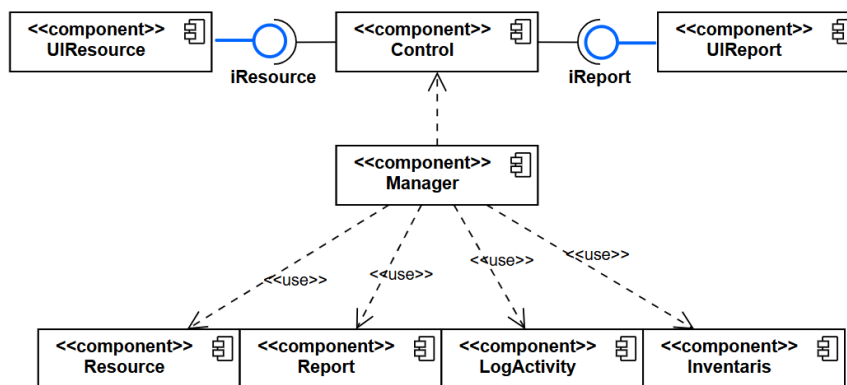
2 Deskripsi Perancangan Global

2.1 Rancangan Lingkungan Implementasi

OS : Windows 11
DBMS : SQLCipher
Development tools : VSCode, Git, Github
Filing system : NTFS (Windows)
Bahasa pemrograman : Python

2.2 Deskripsi Arsitektural

Style/Pattern Arsitektur yang dijadikan acuan adalah *Layered Architecture*. Pemilihan tersebut karena arsitektur ini memungkinkan pemisahan tanggung jawab yang jelas antar lapisan sehingga memudahkan pengelolaan dan pengembangan sistem. Berikut dibawah adalah gambar diagram komponennya.



Gambar 2.2. Diagram Komponen

2.3 Deskripsi Komponen

No	Nama Komponen	Keterangan
1.	UIResource	Untuk Tampilan seluruh Aplikasi baik dari tampilan form, homepage dan navigasi nya
2.	UIReport	Untuk display laporan dan untuk menampilkan form untuk membuat/mengupdate laporan
3.	Control	Menghubungkan antara UI dan logic dari sistem
4.	Manager	Tempat untuk operasi pengelolaan komponen dan koordinasi antar komponen lainnya.
5.	Resource	Komponen tempat menyimpan 1 buah Sumber daya Tunggal beserta data data seperti jumlahnya
6.	Report	Komponen yang digunakan untuk menampung data laporan dan memproses informasi yang diperlukan untuk menghasilkan laporan
7.	LogActivity	Komponen yang bertugas mencatat dan menyimpan semua aktivitas atau log dari pengguna, termasuk tindakan yang dilakukan, waktu kejadian, serta data relevan lainnya untuk keperluan audit dan pemantauan.
8.	Inventaris	Komponen yang berfungsi untuk menampilkan dan mengelola seluruh alokasi sumber daya yang tersedia dalam sistem, termasuk pelacakan perubahan, pemindahan, atau penggunaan sumber daya.

3 Perancangan Rinci

3.1 Realisasi Use Case

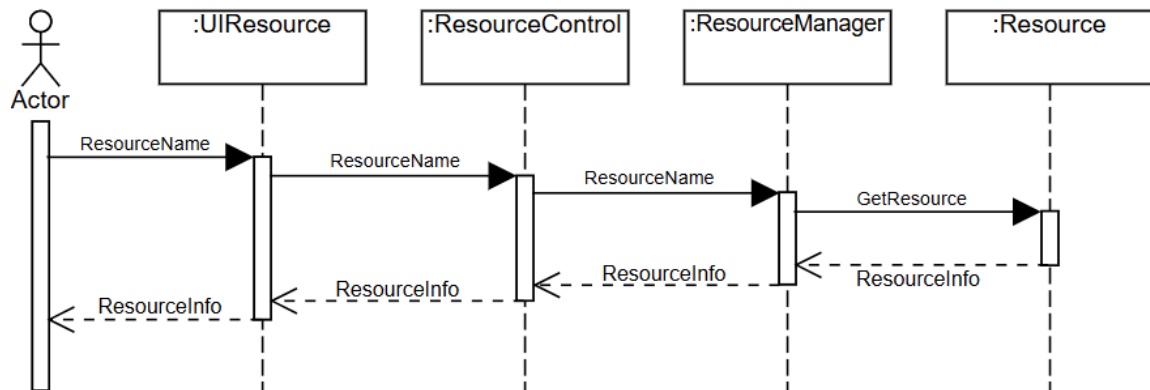
3.1.1 Use Case Melihat Sumber Daya (UC01)

Use Case ini digunakan apabila pengguna perangkat lunak ingin melihat detail informasi terkait dengan sumber daya yang diinginkan.

3.1.1.1 Identifikasi Kelas

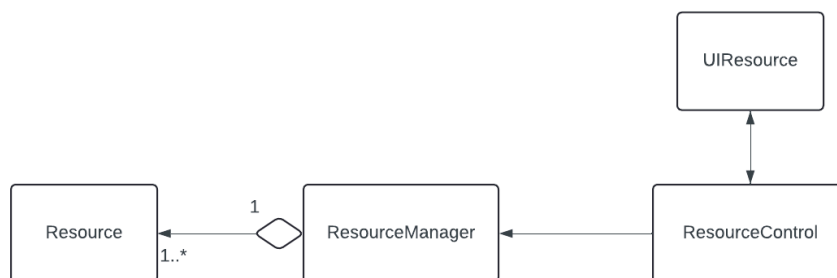
No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait
1.	<i>UIResource</i>	<i>UIResource</i>
2.	<i>ResourceControl</i>	<i>ResourceControl</i>
3.	<i>ResourceManager</i>	<i>ResourceManager</i>
4.	<i>Resource</i>	<i>Resource</i>

3.1.1.2 Sequence Diagram



Gambar 3.1.1.2 Sequence Diagram Use Case 01

3.1.1.3 Diagram Kelas



Gambar 3.1.1.3 Class Diagram Use Case 01

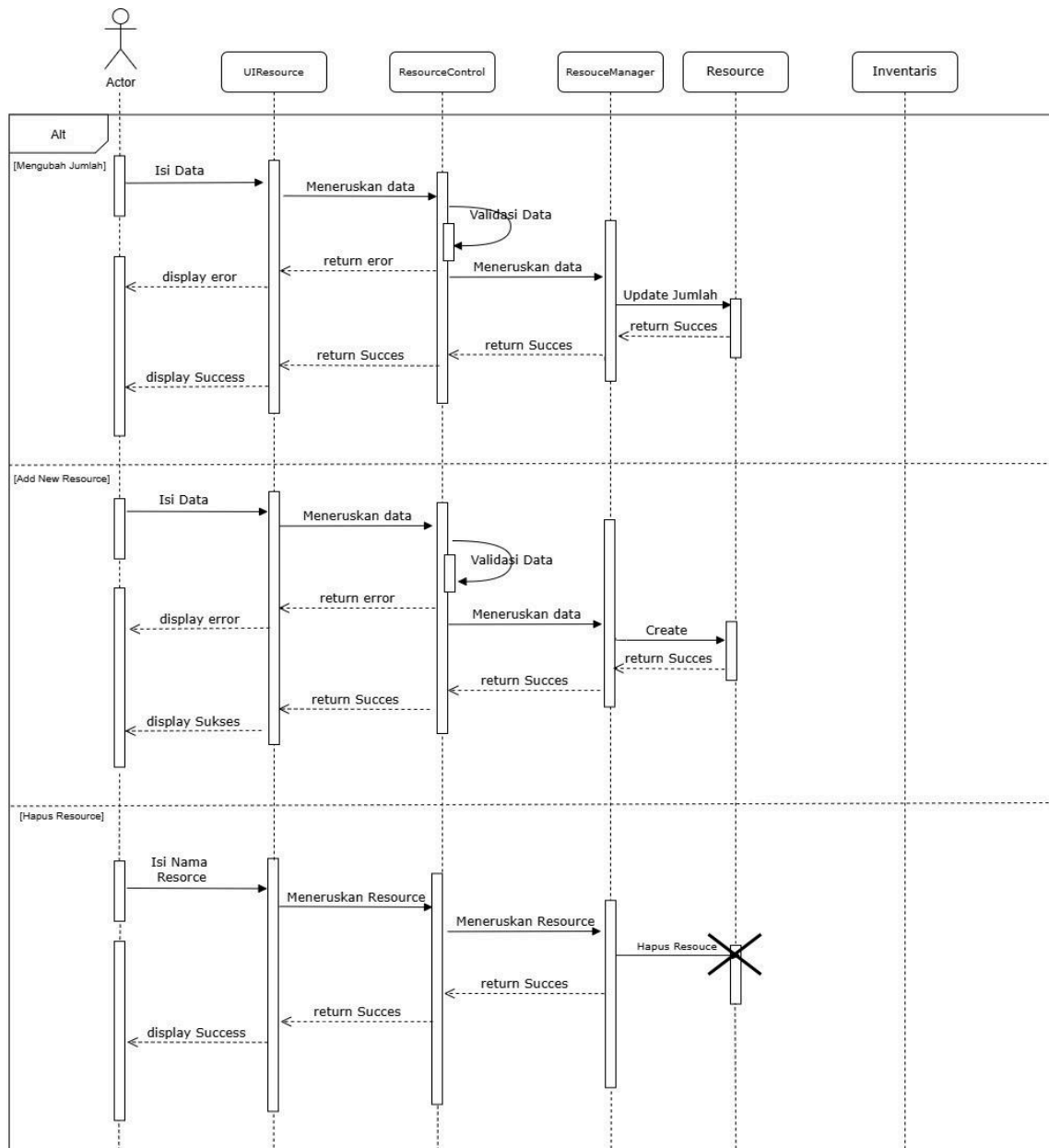
3.1.2 Use Case Mengelola Sumber Daya (UC02)

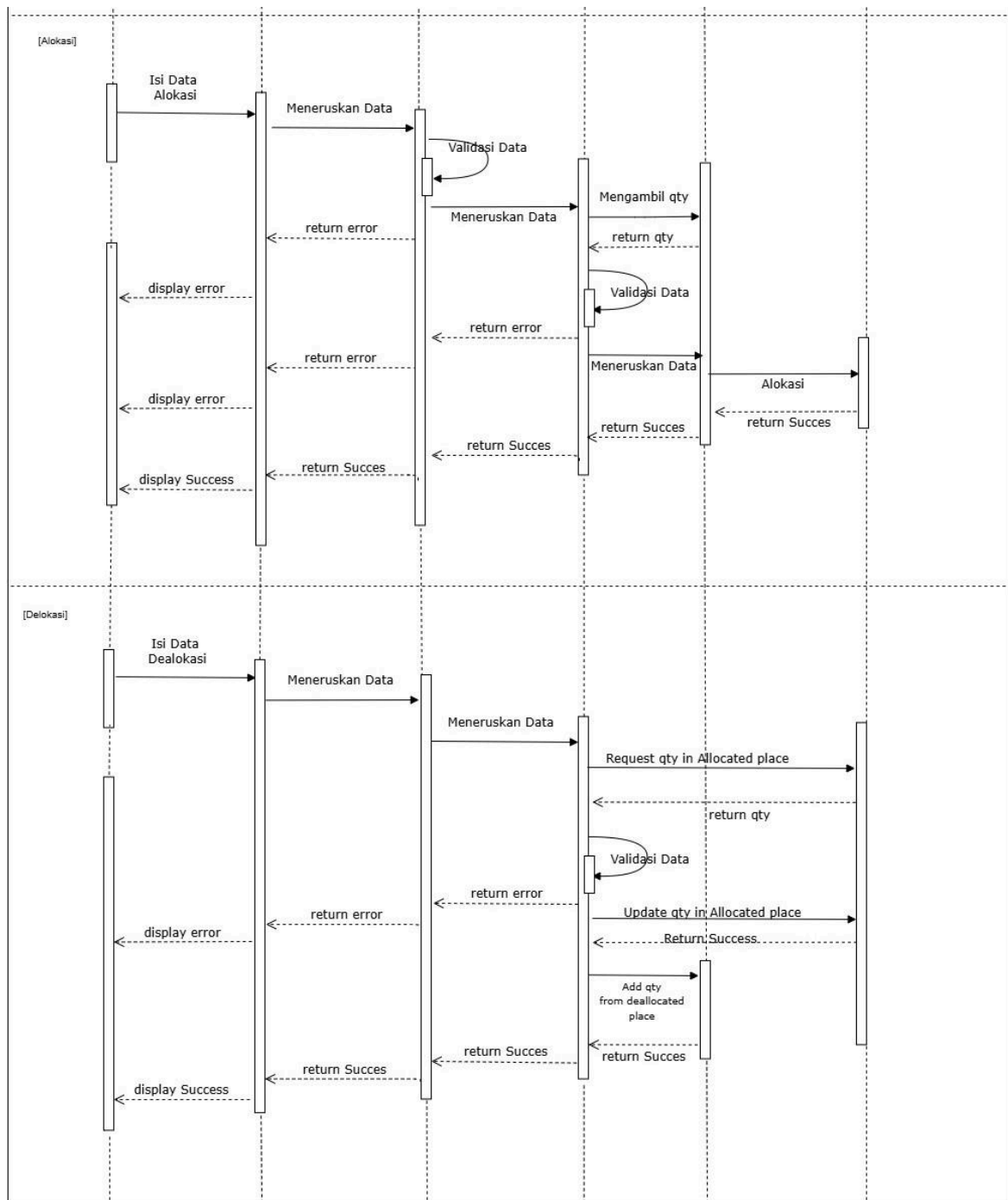
Use Case ini digunakan apabila pengguna perangkat lunak ingin menambahkan, menghapus, mengubah jumlah, mengalokasikan, dan mendealokasikan terkait dengan sumber daya yang diinginkan.

3.1.2.1 Identifikasi Kelas

No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait
1.	<i>UIResource</i>	<i>UIResource</i>
2.	<i>ResourceControl</i>	<i>ResourceControl</i>
3.	<i>ResourceManager</i>	<i>ResourceManager</i>
4.	<i>Resource</i>	<i>Resource</i>
5.	<i>Inventaris</i>	<i>Inventaris</i>

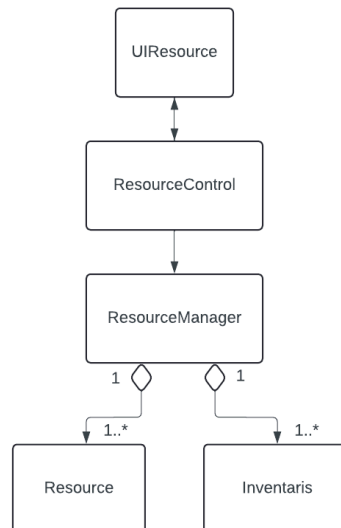
3.1.2.2 Sequence Diagram





Gambar 3.1.2.2 Sequence Diagram Use Case 02

3.1.2.3 Diagram Kelas



Gambar 3.1.2.3 Class Diagram Use Case 02

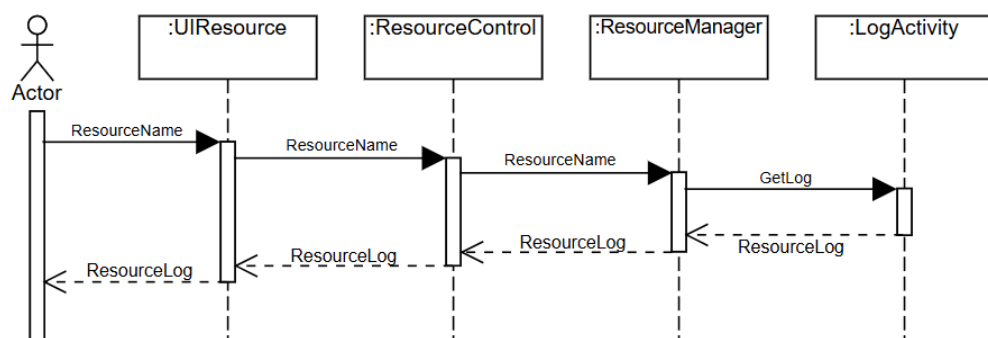
3.1.3 Use Case Melihat Penggunaan Sumber Daya (UC03)

Use Case ini digunakan apabila pengguna perangkat lunak ingin melihat histori/loj penggunaan terkait dengan sumber daya yang diinginkan.

3.1.3.1 Identifikasi Kelas

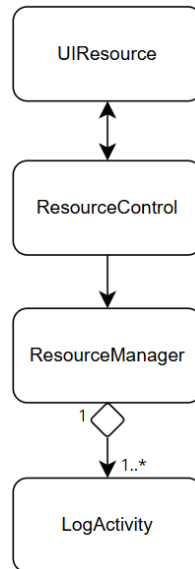
No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait
1.	<i>UIResource</i>	<i>UIResource</i>
2.	<i>ResourceControl</i>	<i>ResourceControl</i>
3.	<i>ResourceManager</i>	<i>ResourceManager</i>
4.	<i>LogActivity</i>	<i>LogActivity</i>

3.1.3.2 Sequence Diagram



Gambar 3.1.3.2 Sequence Diagram Use Case 03

3.1.3.3 Diagram Kelas



Gambar 3.1.3.3 Class Diagram Use Case 03

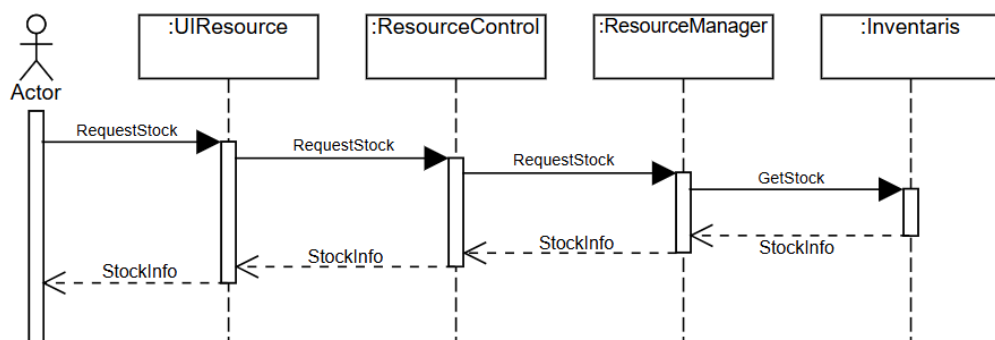
3.1.4 Use Case Melihat Inventaris Sumber Daya (UC04)

Use Case ini digunakan apabila pengguna perangkat lunak ingin melihat inventaris terkait dengan sumber daya yang tersedia saat ini.

3.1.4.1 Identifikasi Kelas

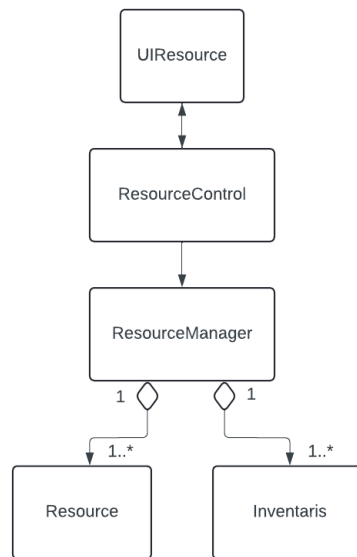
No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait
1.	<i>UIResource</i>	<i>UIResource</i>
2.	<i>ResourceControl</i>	<i>ResourceControl</i>
3.	<i>ResourceManager</i>	<i>ResourceManager</i>
4.	<i>Resource</i>	<i>Resource</i>
5.	<i>Inventaris</i>	<i>Inventaris</i>

3.1.4.2 Sequence Diagram



Gambar 3.1.4.2 Sequence Diagram Use Case 04

3.1.4.3 Diagram Kelas



Gambar 3.1.4.3 Class Diagram Use Case 04

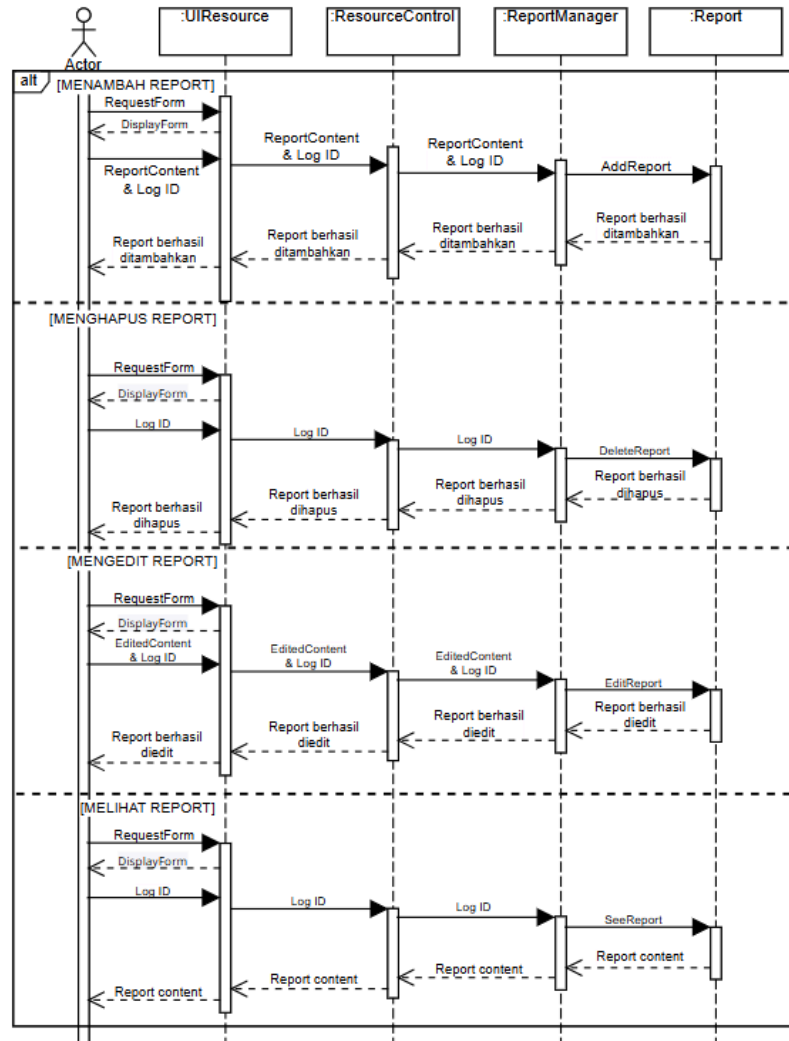
3.1.5 Use Case Mengelola Laporan Sumber Daya (UC05)

Use Case ini digunakan apabila pengguna perangkat lunak ingin membuat, melihat, mengedit, atau menghapus laporan terkait dengan sumber daya yang diinginkan.

3.1.5.1 Identifikasi Kelas

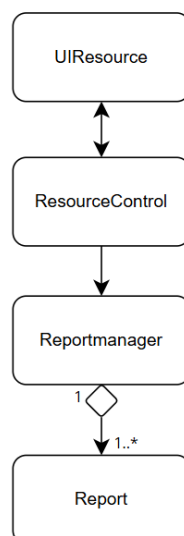
No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait
1.	<i>UIResource</i>	<i>UIReport</i>
2.	<i>ResourceControl</i>	<i>ReportControl</i>
3.	<i>ReportManager</i>	<i>ReportManager</i>
4.	<i>Report</i>	<i>Report</i>

3.1.5.2 Sequence Diagram



Gambar 3.1.5.2 Sequence Diagram Use Case 05

3.1.5.3 Diagram Kelas



Gambar 3.1.5.3 Class Diagram Use Case 05

3.2 Perancangan Detil Kelas

No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait
1.	<i>UIResource</i>	<i>UIResource</i>
2.	<i>ResourceControl</i>	<i>ResourceControl, ReportControl</i>
3.	<i>ResourceManager</i>	<i>ResourceManager</i>
4.	<i>ReportManager</i>	<i>ReportManager</i>
4.	<i>Resource</i>	<i>Resource</i>
5.	<i>Report</i>	<i>Report</i>
6.	<i>Inventaris</i>	<i>Inventaris</i>
7.	<i>LogActivity</i>	<i>LogActivity</i>

3.2.1 Kelas UIResource

Nama Kelas : *UIResource*

Nama Operasi	Visibility (private, public)	Keterangan
<i>+formCreateResource</i>	<i>public</i>	Fungsi untuk menampilkan form antarmuka pengguna guna membuat sumber daya baru.
<i>+formUpdateResource</i>	<i>Public</i>	Fungsi menampilkan antarmuka untuk mengupdate informasi mengenai sumber daya
<i>+RequestStockDetails</i>	<i>public</i>	Fungsi untuk mengajukan permintaan detail stok dari backend dan menampilkan data kepada pengguna tanpa melakukan manipulasi langsung pada basis data.
Nama Atribut	Visibility (private, public)	Tipe
<i>-formData</i>	<i>private</i>	<i>Map<String, Object></i>
<i>-ResourceControl</i>	<i>private</i>	<i>Object</i>

3.2.2 Kelas ResourceControl

Nama Kelas : ResourceControl

Nama Operasi	Visibility (private, public)	Keterangan
+AddResource(resourceName: String, resourceQuantity: int, resourceLocation: String) → ResourceTerdefinisi	private	Menambahkan sumber daya baru dengan nama, jumlah, dan lokasi tertentu lalu mengembalikan detail sumber daya yang telah dibuat.
+RemoveResource(resourceName: String) → ResourceTerhapus	public	Menghapus sumber daya yang ada berdasarkan nama yang telah ditentukan.
+UpdateResourceQuantity(resourceName: String, newQuantity: int) → QuantityResourceBerkurang	public	Melakukan perubahan pada jumlah sumber daya yang telah ditentukan dengan jumlah yang baru lalu mengembalikan hasil pembaruannya.
+ListResources()	public	Memberikan daftar sumber daya apa saja yang tersedia.
+GetResourceActivityLog(ResourceManager, resourceName: String) → AllResourceLog	public	Mengambil aktivitas sumber daya dari ResourceManager berdasarkan nama lalu mengembalikan log semua sumber daya.
+MakeReport(logActivity: LogActivity, reportDetails: String) → Report	public	Membuat laporan yang berisi log aktivitas dan juga detail laporan kemudian mengembalikan laporan secara lengkap.
+GetReport(logActivity: LogActivity) → AllResourceLog	public	Mengambil laporan dari log aktivitas yang telah ditentukan.
+DeleteReport(logActivity: LogActivity)	public	Menghapus laporan dari sistem berdasarkan log aktivitas yang telah ditentukan.
Nama Atribut	Visibility (private, public)	Tipe
-InputValidation	private	boolean
-ResourceManager	private	class ResourceManager
-ReportManager	private	class ReportManager

3.2.3 Kelas ResourceManager

Nama Kelas : ResourceManager

Nama Operasi	Visibility (private, public)	Keterangan
---------------------	---	-------------------

+CreateResource(resourceName: String, resourceQuantity: int)→ ResourceTedefinisi	public	Membuat sumber daya baru dengan nama dan jumlah tertentu, lalu mengembalikan detail sumber daya yang telah dibuat.
+GetResourcebyLoc(resourceName: String, resourceLocation: String)→ AllResourceLoc	public	Mengambil semua data sumber daya yang tersedia di lokasi tertentu berdasarkan nama sumber daya.
+AllocateTo(location: String, quantity: int)→ ResourceAllocated	public	Mengalokasikan sejumlah sumber daya ke lokasi tertentu dan mengembalikan data hasil alokasi.
+DistributeTo(destination: String, quantity: int)→ ResourceDistributed	public	Mendistribusikan sejumlah sumber daya ke tujuan tertentu dan mengembalikan data hasil distribusi.
+DeallocateResourceIn(location: String, quantity: int)→ ResourceDeallocated	public	Menghapus alokasi sejumlah sumber daya dari lokasi tertentu dan mengembalikan data hasil dealokasi.
+UpdateResource(resourceName: String, resourceQuantity: int)→ ResourceQuantityUpdated	public	Memperbarui jumlah sumber daya tertentu dengan nilai baru dan mengembalikan hasil pembaruan.
+DeleteResource(resourceName: String)	public	Menghapus sumber daya dari sistem berdasarkan nama sumber daya.
+requestLogActivity(resourceName: String, Inventaris)→ AllResourceLog	public	Mengambil semua log aktivitas yang terkait dengan sumber daya tertentu dan inventaris yang terkait.
+updateLogActivity(resourceName: String, activityDetails: String)→ ListLogUpdated	public	Memperbarui log aktivitas untuk sumber daya tertentu dengan detail aktivitas baru
Nama Atribut	Visibility (private, public)	Tipe
-ResourceByLoc	private	List
-LogActivityList	private	List

3.2.3.1 Algoritma/Query

Nama Kelas : ResourceManager

Nama Operasi : CreateResource

Algoritma : (Algo-001)

1. Terima resourceName, resourceQuantity
2. Validasi resourceQuantity harus lebih besar dari nol.
3. Eksekusi Q-001
4. Jika berhasil, kembalikan pesan keberhasilan: "Sumber daya berhasil ditambahkan."
5. Jika gagal, kembalikan pesan kesalahan: "Gagal menambahkan sumber daya."

Nama Kelas : ResourceManager

Nama Operasi : UpdateResource

Algoritma : (Algo-002)

1. Terima resourceName, resourceQuantity
2. Validasi resourceQuantity harus lebih besar dari nol.

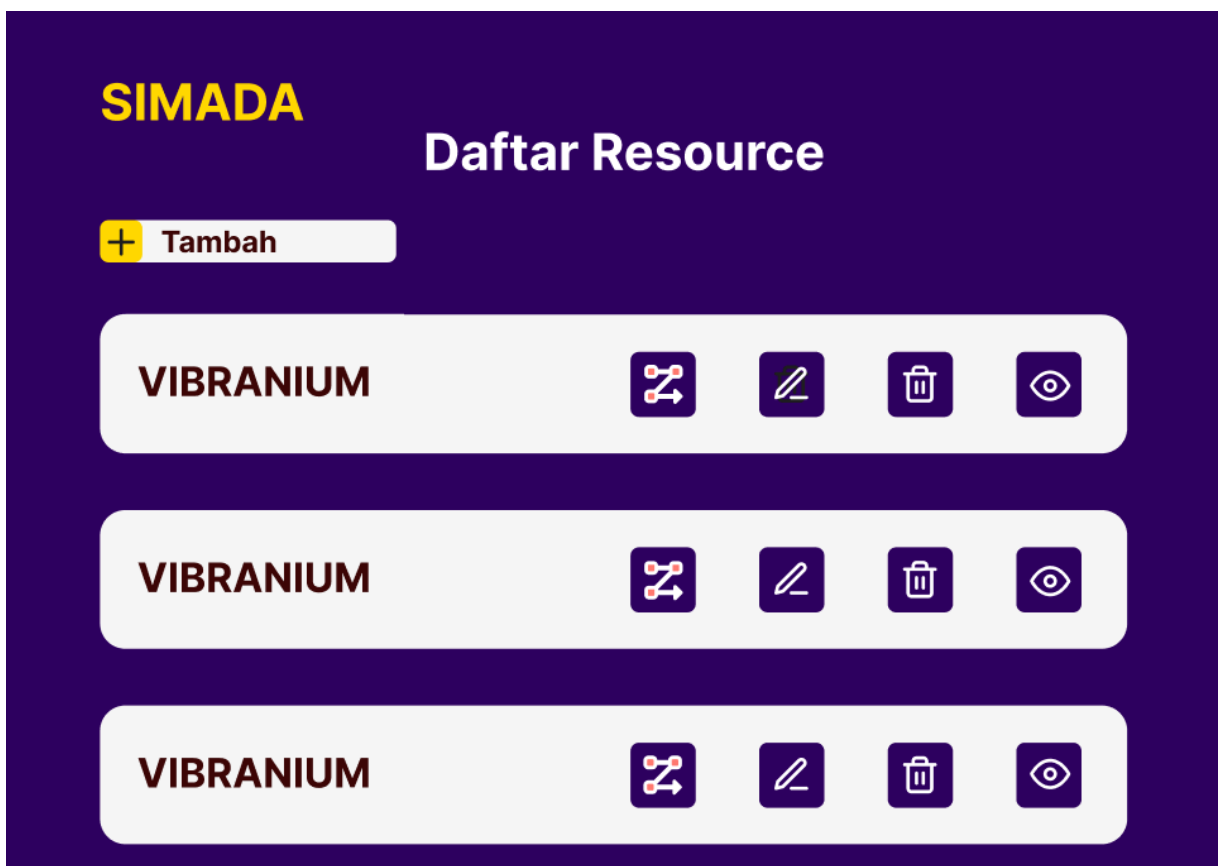
3. Eksekusi Q-003
 4. Jika berhasil, kembalikan pesan keberhasilan: "Jumlah sumber daya berhasil diperbarui."
 5. Jika gagal, kembalikan pesan kesalahan: "Gagal memperbarui jumlah sumber daya."

Nama Kelas : ResourceManager
 Nama Operasi : DeleteResource
 Algoritma : (Algo-003)

1. Terima resourceName,
 2. Eksekusi Q-002
 3. kembalikan pesan keberhasilan: "Sumber daya berhasil dihapus."

No Query	Query	Keterangan
Q-001	INSERT INTO resources (nama, jumlah) VALUES (resourceName, resourceQuantity);	Menambahkan sumberdaya baru ke tabel resources dengan nama dan jumlah tertentu
Q-002	DELETE FROM resources WHERE nama = resourceName;	Menghapus sumberdaya yang telah dipilih
Q-003	UPDATE resources SET jumlah = resourceQuantity WHERE nama = resourceName;	Untuk menambahkan/ mengurangi jumlah dari sumberdaya misal ia di alokasi/dealokasi

3.2.3.2 Perancangan Antarmuka



Gambar 3.2.3.2 Antarmuka Daftar Resource SIMADA

Tabel 3.2.1.1 Objek Daftar Resource

Id Objek	Jenis	Nama	Keterangan
<i>Add_Button</i>	<i>Button</i>	<i>Tambah</i>	<i>Algo-001.</i>
<i>Del_Button</i>	<i>Button</i>	<i>Delete(Trash-Icon)</i>	<i>Algo-003</i>
<i>Update_button</i>	<i>Button</i>	<i>Edit(Writing-Icon)</i>	<i>Algo-002</i>

3.2.4 Kelas Resource

Nama Kelas : Resource

Nama Operasi	Visibility (private, public)	Keterangan
+SetQuantity(newQuantity: int)→ QuantityResourceTerupdate	public	Mengatur jumlah baru dari resource.
+SetLoc(newLocation: String)→ LocationResourceTerupdate	public	Mengatur lokasi baru dari resource.
Nama Atribut	Visibility (private, public)	Tipe
-resourceID	private	integer
-nama	private	string
-jumlah	private	integer

3.2.5 Kelas Report

Nama Kelas : Report

Nama Operasi	Visibility (private, public)	Keterangan
+Report(reportDetails: String)	public	Membuat laporan baru berdasarkan detail laporan
+GetReportDetail	public	Mengambil detail laporan yang ada dalam sistem
+SetReportDetail	public	Mengubah atau memperbarui detail laporan tertentu.
+SetReport	public	Menyetel atau mengganti laporan secara keseluruhan.
+ConfirmDeletion	public	Memastikan penghapusan laporan setelah konfirmasi dari pengguna.
Nama Atribut	Visibility (private, public)	Tipe
-ReportDetail	private	string

3.2.6 Kelas ReportManager

Nama Kelas : Report

Nama Operasi	Visibility (private, public)	Keterangan
+AddReport(reportDetails: String)	public	Menambahkan laporan baru ke database dan mengembalikan status berhasil/tidak.
+GetReportById	public	Mengambil laporan tertentu berdasarkan ID dari database.
+UpdateReportById	public	Memperbarui detail laporan tertentu berdasarkan ID.
+DeleteReportById	public	Menghapus laporan tertentu dari database berdasarkan ID
Nama Atribut	Visibility (private, public)	Tipe
-reports	private	List

3.2.6.1 Algoritma/Query

Nama Kelas : ReportManager

Nama Operasi : AddReport

Algoritma : (Algo-001)

1. Terima reportId, reportDetails.
2. Buat objek laporan baru
3. Tambahkan objek laporan ke dictionary reports
4. Simpan laporan ke basis data menggunakan Q-001

Nama Kelas : ReportManager

Nama Operasi : UpdateReportById

Algoritma : (Algo-002)

1. Terima reportId, reportDetails.
2. Simpan reportDetails ke basis data menggunakan Q-003

Nama Kelas : ReportManager

Nama Operasi : DeleteReportById

Algoritma : (Algo-003)

1. Terima reportId
2. Hapus laporan dari basis data menggunakan Q-004

Nama Kelas : ReportManager

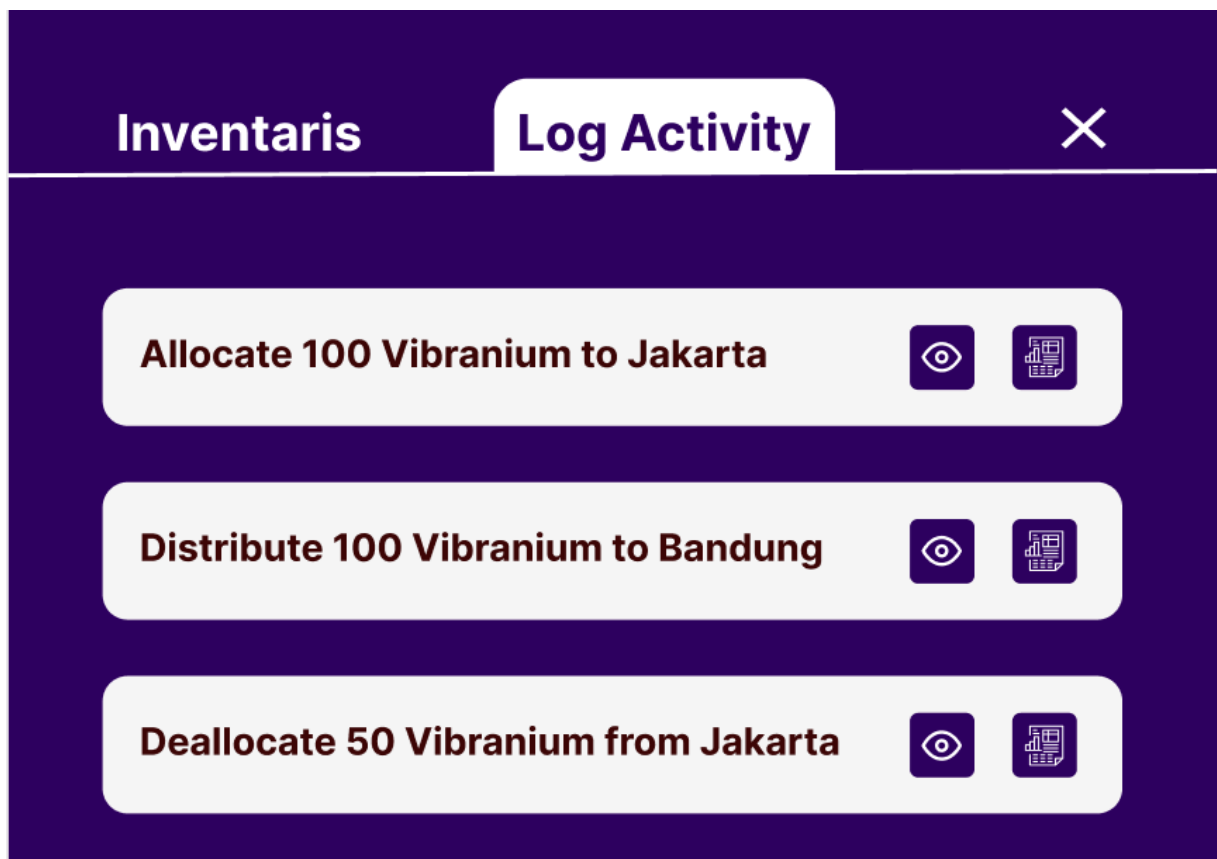
Nama Operasi : GetReportById

Algoritma : (Algo-004)

1. Terima reportId
2. Jika ada akan Mengambil laporan dari basis data menggunakan Q-002 jika tidak kembalikan teks "Tidak Ada Laporan yang Dibuat"
3. Menampilkan reportDetails dari laporan

Query :

No Query	Query	Keterangan
Q-001	INSERT INTO reports (report_id, details) VALUES (reportId, reportDetails);	Simpan laporan ke basis data
Q-002	SELECT report by ID FROM reports	Mengambil laporan tertentu berdasarkan ID.
Q-003	UPDATE reports SET details = reportDetails WHERE report_id = reportId;	Memperbarui detail laporan tertentu berdasarkan ID.
Q-004	DELETE FROM reports WHERE report_id = reportId;	Menghapus laporan tertentu berdasarkan ID.



Gambar 3.2.6.2 Antarmuka Log Activity SIMADA

Id Objek	Jenis	Nama	Keterangan
<i>detail_Button</i>	<i>Button</i>	<i>(eye-icon)</i>	<i>Algo-004</i>
<i>Write report</i>	<i>Button</i>	<i>(report-Icon)</i>	<i>Algo-001 dan Algo-002</i>

3.2.7 Kelas Inventaris

Nama Kelas : *Inventaris*

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
<i>+AddResourceToLocation(resourceName: String, location: String, quantity: int)</i>	<i>public</i>	<i>Menambahkan sejumlah sumber daya ke lokasi tertentu. Jika lokasi belum ada, buat entri baru.</i>
<i>+AllocateFromLocation(resourceName: String, sourceLocation: String, targetLocation: String, quantity: int)</i>	<i>public</i>	<i>Mengalokasikan sumber daya dari lokasi A ke lokasi B, memperbarui alokasi kedua lokasi.</i>
<i>+DeallocateToSource(resourceName: String, location: String, quantity: int)</i>	<i>public</i>	<i>Mengembalikan sejumlah sumber daya dari lokasi ke sumber utama.</i>

+UpdateResourceAtLocation(resourceName: String, location: String, quantityChange: int)	public	Memperbarui jumlah sumber daya di lokasi tertentu tanpa perubahan struktur.
+GetAllResourceAndLocation(resourceName: String)	public	Mengambil daftar lokasi dan jumlah sumber daya yang dialokasikan di setiap lokasi untuk sumber daya tertentu.
Nama Atribut	Visibility (private, public)	Tipe
-Allocations	private	Dictionary
-locations	private	List

3.2.7.1 Algoritma/Query

Nama Kelas : Inventaris

Nama Operasi : AddResourceToLocation

Algoritma : (Algo-001)

1. Terima resourceName, location, quantity.
2. jika lokasi sudah ada eksekusi Q-002
3. jika belum ada eksekusi Q-003

Nama Kelas : Inventaris

Nama Operasi : AllocateFromLocation

Algoritma : (Algo-002)

1. Terima resourceName, sourceLocation, targetLocation, quantity.
2. jika jumlah mencukupi eksekusi Q-004 dan update quantity di target dengan Q-002
3. jika belum cukup kembalikan pesan belum cukup

Nama Kelas : Inventaris

Nama Operasi : GetAllResourceAndLocation

Algoritma : (Algo-003)

1. Terima resourceName.
2. Eksekusi Q-001
3. Mengembalikan data hasil dari query tersebut

Nama Kelas : Inventaris

Nama Operasi : Deallocate

Algoritma : (Algo-004)

1. Terima resourceName, location, quantity..
2. Eksekusi Q-004 dan Q-005
3. Mengembalikan data hasil dari query tersebut

Query :

No Query	Query	Keterangan
Q-001	SELECT location, quantity FROM allocations WHERE resource_id = (SELECT resource_id FROM resources WHERE nama = resourceName);	Mengambil semua lokasi dan jumlah alokasi untuk sumber daya tertentu.
Q-002	UPDATE inventaris SET quantity = quantity + quantity WHERE resource_id = (SELECT resource_id FROM resources WHERE nama = resourceName) AND location = location;	Menambahkan jumlah sumber daya di lokasi tertentu jika entri sudah ada.
Q-003	INSERT INTO inventaris (resource_id, location, quantity) VALUES ((SELECT resource_id FROM resources WHERE nama = resourceName), location, quantity);	Membuat entri alokasi baru di lokasi tertentu untuk sumber daya.

Q-004	<i>UPDATE inventaris SET quantity = quantity - quantity WHERE resource_id = (SELECT resource_id FROM resources WHERE nama = resourceName) AND location = location;</i>	<i>Mengurangi jumlah sumberdaya tertentu yang ada di lokasi yang dipilih</i>
Q-005	<i>UPDATE resources SET total_quantity = total_quantity + quantity WHERE nama = resourceName;</i>	<i>Menambahkan jumlah resource yang ada di tempat utama</i>

3.2.7.2 Perancangan Antarmuka



Gambar 3.2.7.1 Antarmuka Inventaris Daftar Resource SIMADA

Id Objek	Jenis	Nama	Keterangan
<i>Deallocate_Button</i>	<i>Button</i>	<i>Delete(Trash-Ico n)</i>	<i>Algo-004</i>
<i>Distribute_button</i>	<i>Button</i>	<i>Distribute(truck-I con)</i>	<i>Algo-002,</i>

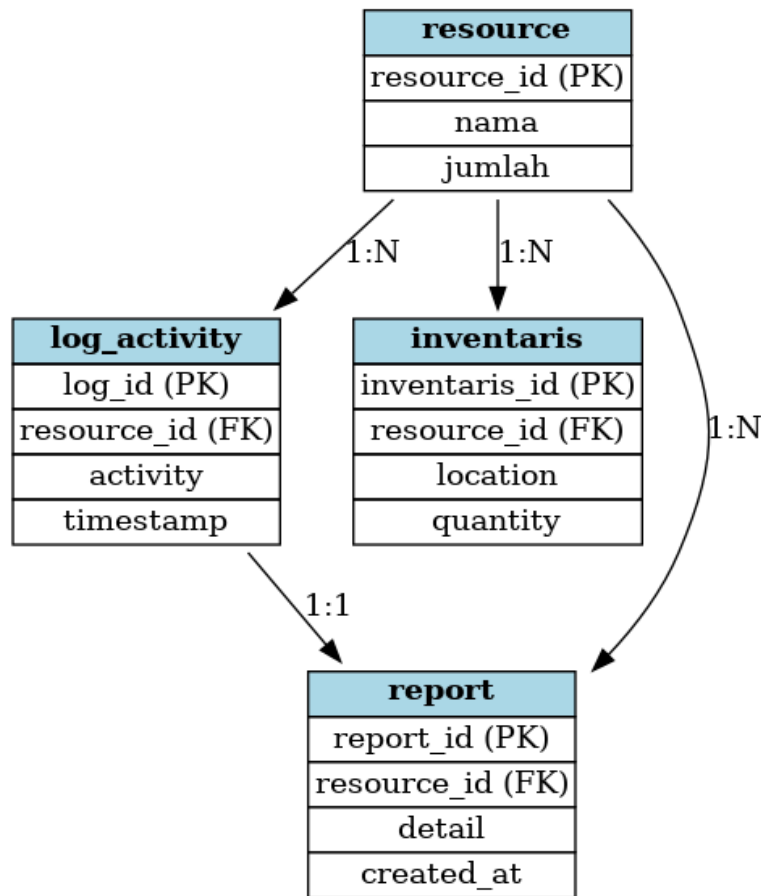
3.2.8 Kelas LogActivity

Nama Kelas : *Inventaris*

Nama Operasi	Visibility (private, public)	Keterangan
<i>+LogNewActivity(resource: Resource, actionType: String, timestamp: String, jumlah: int, lokasi: String)→ Log</i>	<i>public</i>	<i>Menambahkan log activity sesuai dengan data pada sumber daya terkait</i>

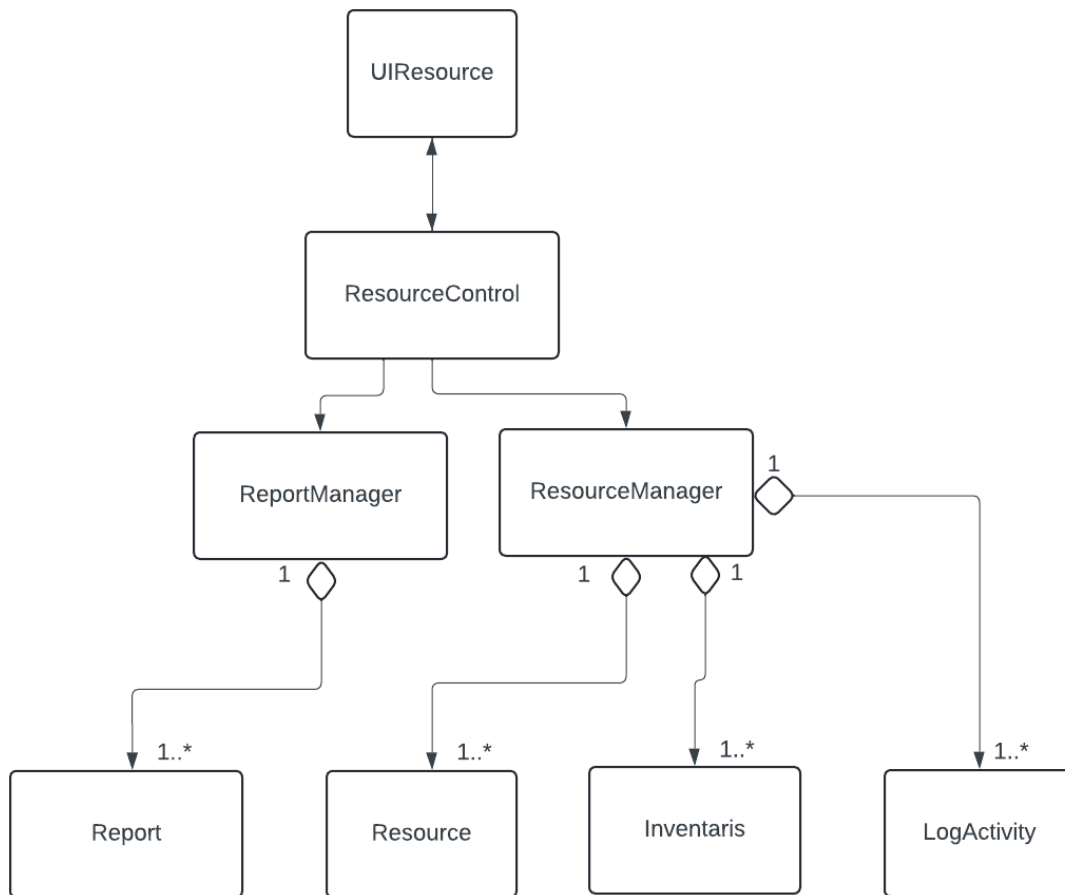
+ <i>GetLogActivity(resource: Resource)</i>	<i>Public</i>	<i>Mengambil data log activity pada sumber daya yang diinginkan</i>
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
- <i>ResourceName</i>	<i>private</i>	<i>String</i>
- <i>ActionType</i>	<i>private</i>	<i>String</i>
- <i>Timestamp</i>	<i>private</i>	<i>String</i>
- <i>Jumlah</i>	<i>private</i>	<i>Integer</i>
- <i>Lokasi</i>	<i>private</i>	<i>String</i>

3.3 Perancangan Representasi Persistensi Kelas



<i>Nama Tabel</i>	<i>Primary Key</i>	<i>Kelas Entity Terkait</i>	<i>Deskripsi</i>
<i>resource</i>	<i>resource_id</i>	<i>Resource</i>	Merupakan entitas yang merepresentasikan sumber daya seperti nama dan jumlah sumber daya.
<i>log_activity</i>	<i>log_id</i>	<i>LogActivity</i>	Menyimpan catatan aktivitas log terkait sumber daya, seperti alokasi atau perubahan.
<i>inventaris</i>	<i>inventaris_id</i>	<i>Inventaris</i>	Menyimpan lokasi inventaris sumber daya beserta jumlah sumber daya yang tersimpan di lokasi.
<i>report</i>	<i>report_id</i>	<i>Report</i>	Menyimpan detail laporan yang dibuat berdasarkan log aktivitas dan sumber daya tertentu.

3.4 Diagram Kelas Keseluruhan



4 Matriks Kerunutan

Kelas	Use Case Terkait
UIResource	UC01, UC02, UC03, UC04, UC05
ResourceControl	UC01, UC02, UC03, UC04, UC05
ResourceManager	UC01, UC02, UC03, UC04
ReportManager	UC05
Resource	UC01, UC02
Report	UC05
Inventaris	UC02, UC04
LogActivity	UC03