

TUGAS 7

Deskripsi Perancangan Perangkat Lunak

ReBuilt

(Perangkat Lunak untuk Renovasi Rumah)

untuk:

Kayla Namira M.

Dipersiapkan oleh:

Grup 9:

- | | | |
|----|------------------------------|----------|
| 1. | Abrar Abhirama Widyadhana | 13523058 |
| 2. | Noumisyifa Nabila Nareswari | 13523011 |
| 3. | Diyah Susan Nugrahani | 13523080 |
| 4. | Muhammad Edo Raduputu Aprima | 13523096 |
| 5. | Athian Nugraha Muarajuang | 13523106 |

**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

1. Pendahuluan.....	6
1.1 Tujuan Penulisan Dokumen.....	7
1.2 Lingkup Masalah.....	7
1.3 Definisi dan Istilah.....	7
1.4 Aturan Penamaan dan Penomoran.....	7
1.5 Referensi.....	8
1.6 Ikhtisar Dokumen.....	8
2 Deskripsi Perancangan Global.....	8
2.1 Rancangan Lingkungan Implementasi.....	9
2.2 Deskripsi Arsitektural.....	9
2.3 Deskripsi Komponen.....	9
3 Perancangan Rinci.....	10
3.1 Realisasi Use Case.....	10
3.1.1 Use Case Mengelola Seluruh Data Proyek Renovasi.....	10
3.1.1.1 Identifikasi Kelas.....	10
3.1.1.2 Sequence Diagram.....	11
3.1.1.3 Diagram Kelas.....	12
3.1.2 Use Case Mengelola Seluruh Data Tugas dari Setiap Proyek Renovasi.....	13
3.1.2.1 Identifikasi Kelas.....	13
3.1.2.2 Sequence Diagram.....	14
3.1.2.3 Diagram Kelas.....	15
3.1.3 Use Case Mengelola Seluruh Data Biaya pada Setiap Tugas.....	16
3.1.3.1 Identifikasi Kelas.....	16
3.1.3.2 Sequence Diagram.....	17
3.1.3.3 Diagram Kelas.....	18
3.1.4 Use Case Mengelola Seluruh Data Inspirasi Renovasi.....	19
3.1.4.1 Identifikasi Kelas.....	19
3.1.4.2 Sequence Diagram.....	20
3.1.4.3 Diagram Kelas.....	21
3.2 Perancangan Detil Kelas.....	22
3.2.1 Kelas Proyek.....	22
3.2.1.1 Algoritma/Query.....	23
3.2.1.2 Diagram Statechart.....	24
3.2.1.3 Perancangan Antarmuka.....	24
3.2.1.4 Perancangan Representasi Persistensi Kelas.....	24
3.2.2 Kelas PengelolaDataProyek.....	25
3.2.2.1 Algoritma/Query.....	25
3.2.2.2 Diagram Statechart.....	26
3.2.2.3 Perancangan Antarmuka.....	26
3.2.2.4 Perancangan Representasi Persistensi Kelas.....	26
3.2.3 Kelas DisplayDataProyek.....	26
3.2.3.1 Algoritma/Query.....	27
3.2.3.2 Perancangan Antarmuka.....	28
3.2.3.3 Perancangan Representasi Persistensi Kelas.....	29
3.2.4 Kelas TugasProyek.....	29
3.2.4.1 Algoritma/Query.....	29

3.2.4.2 Diagram Statechart.....	30
3.2.4.3 Perancangan Antarmuka.....	30
3.2.4.4 Perancangan Representasi Persistensi Kelas.....	30
3.2.5 Kelas PengelolaTugasProyek.....	30
3.2.5.1 Algoritma/Query.....	31
3.2.6.1 Diagram Statechart.....	31
3.2.6.2 Perancangan Antarmuka.....	31
3.2.6.3 Perancangan Representasi Persistensi Kelas.....	31
3.2.6 Kelas DisplayTugasProyek.....	32
3.2.6.1 Algoritma/Query.....	32
3.2.6.2 Diagram Statechart.....	32
3.2.6.3 Perancangan Antarmuka.....	33
3.2.6.4 Perancangan Representasi Persistensi Kelas.....	33
3.2.7 Kelas InspirasiRenovasi.....	33
3.2.7.1 Algoritma/Query.....	34
3.2.7.2 Diagram Statechart.....	34
3.2.7.3 Perancangan Antarmuka.....	34
3.2.7.4 Perancangan Representasi Persistensi Kelas.....	35
3.2.8 Kelas Pengelola Inspirasi Renovasi.....	35
3.2.8.1 Algoritma/Query.....	35
3.2.8.2 Diagram Statechart.....	37
3.2.8.3 Perancangan Antarmuka.....	37
3.2.9 Kelas Display Inspirasi Renovasi.....	37
3.2.9.1 Algoritma/Query.....	37
3.2.9.2 Diagram Statechart.....	38
3.2.9.3 Perancangan Antarmuka.....	38
3.2.10 Kelas DisplayBiaya.....	40
3.2.10.1 Algoritma/Query.....	40
3.2.10.2 Perancangan Antarmuka.....	41
3.2.10.3 Perancangan Representasi Persistensi Kelas.....	41
3.2.11 Kelas Biaya.....	41
3.2.11.1 Algoritma/Query.....	42
3.2.11.2 Perancangan Antarmuka.....	43
3.2.11.3 Perancangan Representasi Persistensi Kelas.....	43
3.2.12 Kelas PengelolaBiaya.....	43
3.2.12.1 Algoritma/Query.....	44
3.3 Diagram Kelas Keseluruhan.....	45
4 Matriks Kerunutan.....	47

Daftar Tabel

Tabel 1.3. Singkatan, Akronim atau istilah	6
Tabel 1.4. Aturan Penomoran	6
Tabel 2.3. Daftar Modul	8
Tabel 3.1.1.1. Daftar Kelas pada Use Case 01	9
Tabel 3.1.1.3. Tabel Kelas Use Case 01	11
Tabel 3.1.2.1. Daftar Kelas pada Use Case 02	12
Tabel 3.1.2.3. Tabel Kelas Use Case 02	14
Tabel 3.1.3.1. Daftar Kelas Use Case 03	15
Tabel 3.1.3.3. Tabel Kelas Use Case 03	17
Tabel 3.1.4.1. Tabel Kelas Use Case 04	18
Tabel 3.1.4.3. Tabel Kelas Use Case 04	20
Tabel 3.2. Daftar Kelas pada Rebuilt	21
Tabel 3.2.1. Daftar Operasi dan Atribut Kelas Proyek	21
Tabel 3.2.1.1. Daftar Operasi Query Kelas Proyek	23
Tabel 3.2.2. Daftar Operasi dan Atribut Kelas PengelolaDataProyek	25
Tabel 3.2.2.1. Daftar Operasi Query Kelas PengelolaDataProyek	25
Tabel 3.2.3. Daftar Operasi dan Atribut Kelas DisplayDataProyek	26
Tabel 3.2.3.1. Daftar Objek AntarMuka DisplayProyek	28
Tabel 3.2.4. Daftar Operasi dan Atribut Kelas TugasProyek	28
Tabel 3.2.4.1. Daftar Operasi Query Kelas TugasProyek	29
Tabel 3.2.5. Daftar Operasi dan Atribut Kelas PengelolaTugasProyek	30
Tabel 3.2.5.1. Daftar Operasi Query Kelas PengelolaTugasProyek	30
Tabel 3.2.6. Daftar Operasi dan Atribut Kelas DisplayTugasProyek	31
Tabel 3.2.6.1. Daftar Objek AntarMuka DisplayTugasProyek	32
Tabel 3.2.7. Daftar Operasi dan Atribut Kelas InspirasiRenovasi	33
Tabel 3.2.7.1. Daftar Operasi Query Kelas InspirasiRenovasi	34
Tabel 3.2.8. Daftar Operasi dan Atribut Kelas PengelolaInspirasiRenovasi	36
Tabel 3.2.8.1. Daftar Operasi Query Kelas PengelolaInspirasiRenovasi	36
Tabel 3.2.9. Daftar Operasi dan Atribut Kelas DisplayInspirasiRenovasi	39
Tabel 3.2.9.1. Daftar Objek AntarMuka DisplayInspirasi	39
Tabel 3.2.10. Daftar Operasi dan Atribut Kelas DisplayBiaya	39
Tabel 3.2.10.1. Daftar Objek AntarMuka DisplayBiaya	40
Tabel 3.2.11. Daftar Operasi dan Atribut Kelas Biaya	41
Tabel 3.2.11.1 Daftar Operasi Query Kelas Biaya	42
Tabel 3.2.12. Daftar Operasi dan Atribut Kelas PengelolaBiaya	43
Tabel 3.2.12.1. Daftar Operasi Query Kelas PengelolaBiaya	43
Tabel 3.3.. Daftar Operasi dan Atribut Kelas Keseluruhan	44
Tabel 4. Matriks Keruntutan	46

Daftar Gambar

Gambar 1. Deskripsi Arsitektural	08
Gambar 2. Diagram Sequence Proyek Renovasi	10
Gambar 3. <i>Diagram Kelas Proyek Renovasi</i>	11
Gambar 4. <i>Diagram Sequence Pengelola Proyek</i>	13
Gambar 5. <i>Diagram Kelas Pengelola Proyek</i>	14
Gambar 6. <i>Diagram Sequence Pengelola Biaya</i>	16
Gambar 7. <i>Diagram Kelas Pengelola Biaya</i>	17
Gambar 8. <i>Diagram Sequence Pengelola Inspirasi Renovasi</i>	19
Gambar 9. <i>Diagram Kelas Pengelola Inspirasi Renovasi</i>	20
Gambar 10. <i>Diagram Statechart Proyek</i>	23
Gambar 11. <i>Perancangan Representasi Persistensi Kelas Proyek</i>	23
Gambar 12. <i>Diagram Statechart Pengelola Data Proyek</i>	25
Gambar 13. <i>Perancangan Antarmuka displayAllProyek</i>	26
Gambar 14. <i>Diagram Statechart Tugas Proyek</i>	28
Gambar 15. <i>Perancangan Representasi Persistensi Kelas Tugas</i>	29
Gambar 16. <i>Perancangan Display Tugas Proyek</i>	31
Gambar 17. <i>Diagram Statechart Inspirasi Renovasi</i>	33
Gambar 18. <i>Perancangan Representasi Persistensi Kelas Inspirasi</i>	33
Gambar 19. <i>Diagram Statechart Pengelola Inspirasi Renovasi</i>	36
Gambar 20. <i>Diagram Statechart Inspirasi Renovasi</i>	37
Gambar 21. <i>Perancangan Antarmuka displayInspirasi</i>	38
Gambar 22. <i>Perancangan Antarmuka displayEditInspirasi</i>	38
Gambar 23. <i>Perancangan Antarmuka displayDeleteInspirasi</i>	39
Gambar 24. <i>Perancangan Antarmuka displayBiaya</i>	40
Gambar 25. <i>Diagram State Chart Pengelola Biaya</i>	44
Gambar 26. <i>Diagram Kelas Keseluruhan</i>	44

1. Pendahuluan

1.1 Tujuan Penulisan Dokumen

Tujuan dari pembuatan dokumen Dokumen Perancangan Perangkat Lunak (DPPL) ini adalah sebagai bahan acuan dalam pengembangan perangkat lunak renovasi rumah ReBuilt. Dokumen ini menjadi acuan dalam mengimplementasikan semua rancangan yang ingin direalisasikan agar sesuai dengan kebutuhan dan keinginan user. Dokumen ini juga menjadi batasan agar dalam proses pengembangannya sesuai dengan rencana awal dan tidak keluar dari batasan-batasan yang ditentukan. Selain itu, dokumen ini juga dapat mempermudah *developer* dalam pengerjaan karena semua spesifikasi yang dibutuhkan disusun secara lengkap dan sistematis sehingga dapat dengan mudah dilakukan revisi apabila diminta oleh *client*. Dokumen ini juga dapat mempermudah proses pengembangan aplikasi jika dalam pengerjaannya ada pergantian SDM agar SDM yang baru bergabung dalam proyek dapat memahami seluruh spesifikasi dan alur pengerjaan proyek melalui dokumen ini karena semua hal yang dibutuhkan sudah tercantum di dalamnya.

1.2 Lingkup Masalah

Lingkup masalah yang ReBuilt ingin tuntaskan utamanya adalah meningkatkan efisiensi dan efektivitas sistem manajemen proyek-proyek renovasi rumah dengan mengembangkan suatu P/L pengelolaan yang terpusat. Selain itu, agar pengelolaan proyek renovasi dapat lebih optimal, ReBuilt memungkinkan penggunaannya agar suatu proyek terbagi menjadi beberapa tugas utama. Rincian yang dapat ditambahkan pada setiap proyek dan tugas diberikan dengan cukup rinci sehingga pengguna dapat menggunakan ReBuilt sebagai acuan manajemen proyek yang optimal. ReBuilt juga memasukkan permasalahan manajemen biaya proyek ke dalam lingkup masalah yang ingin diatasi dengan menambahkan fitur yang memungkinkan pengguna untuk mencatat pengeluaran yang kemudian akan dikalkulasi secara otomatis. Lingkup masalah terakhir yang ReBuilt ingin atasi adalah penampungan inspirasi proyek renovasi rumah dengan mengintegrasikan ReBuilt dengan fitur pengelolaan inspirasi-inspirasi renovasi rumah.

1.3 Definisi dan Istilah

Tabel 1.3. Singkatan, Akronim atau istilah

Singkatan, Akronim, atau Istilah	Penjelasan
Deskripsi Perancangan Perangkat Lunak (DPPL)	Dokumen yang menjelaskan
Perangkat Lunak (P/L)	Produk digital yang dijalankan di hardware
ReBuilt	Aplikasi manajemen proyek renovasi rumah
Datebase Management System (DBMS)	Perangkat lunak yang memungkinkan pembuatan, pengelolaan, dan manipulasi basis data.
Operating System(OS)	Sistem yang mengelola perangkat keras dan perangkat lunak
Hardware (HW)	Komponen fisik komputer
Structured Query Language (SQL)	Bahasa pemrograman untuk mengelola dan memanipulasi basis data relasional.
Actor	Pengguna atau sistem yang berinteraksi dengan P/L

1.4 Aturan Penamaan dan Penomoran

Berikut adalah aturan penomoran dari hal/bagian yang ada dalam dokumen ini.

Tabel 1.4. Aturan Penomoran

Hal/Bagian	Penomoran	Keterangan
Use Case	UC<XX>	Huruf 'UC' melambangkan kata <i>use case</i> dan <XX> merepresentasikan urutan penomoran <i>use case</i> yang ada pada P/L.
Kelas	C-<XX>	Huruf 'C' melambangkan kata <i>class</i> yang merupakan terjemahan bahasa Inggris dari kelas dan <XX> merepresentasikan urutan penomoran kelas yang ada pada P/L.
Algoritma	Algo-XXYY	'Algo' melambangkan kata algoritma dan XX merepresentasikan nomor kelas dan YY merepresentasikan

		nomor metode ke berapa dalam kelas XX algoritma tersebut merepresentasikan.
Query	Q-<XX>-<YY>	Huruf 'Q' melambangkan kata query dan <XX> merepresentasikan urutan penomoran kelas dan <YY> merepresentasikan urutan query tersebut di dalam kelas tersebut.

1.5 Referensi

Tim Pengajar IF2250. 2022. Slide perkuliahan Analisis Struktur dan Penulisan SKPL.

1.6 Ikhtisar Dokumen

Dokumen ini terbagi ke dalam 6 bagian. Pada bagian pertama, yaitu pendahuluan, berisi tujuan penulisan dokumen, lingkup masalah, definisi istilah dan singkatan, aturan penomoran, serta referensi yang digunakan. Bagian kedua mencakup deskripsi umum sistem dan perangkat lunak, yang meliputi karakteristik pengguna, batasan perangkat lunak, dan lingkungan operasional perangkat lunak tersebut.

Pada bagian ketiga, dokumen merinci kebutuhan perangkat lunak, baik dari sisi fungsional maupun non-fungsional, yang akan menjadi dasar pengembangan aplikasi. Bagian keempat mendeskripsikan pemodelan use case, yang meliputi identifikasi aktor, use case, dan skenario yang terkait dengan interaksi antara pengguna dan sistem. Bagian kelima menyajikan pemodelan kelas, yang memuat struktur perangkat lunak berdasarkan use case yang telah diidentifikasi. Terakhir, bagian keenam membahas traceability, yang menghubungkan setiap kebutuhan dengan implementasi yang sesuai.

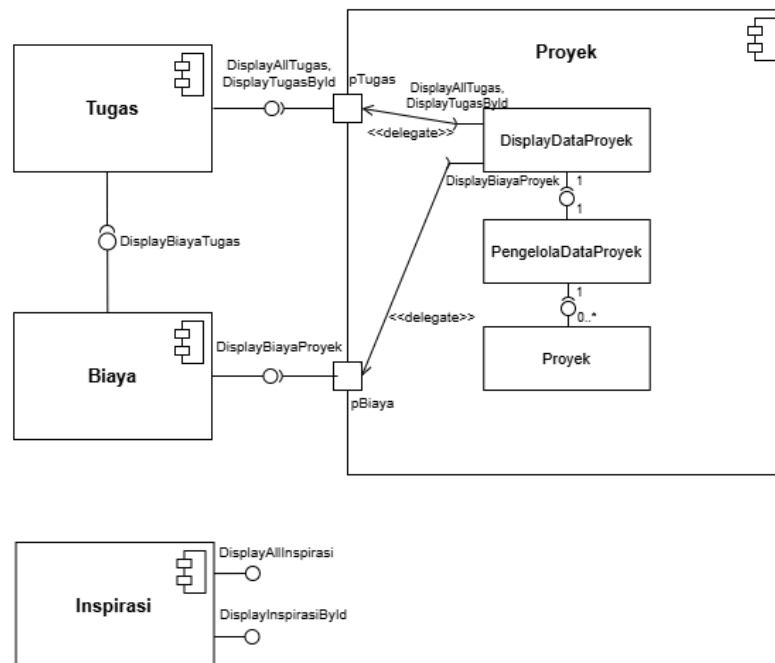
2 Deskripsi Perancangan Global

2.1 Rancangan Lingkungan Implementasi

- OS : Windows
- DBMS : MySQL
- Dev tools : Visual Studio Code
- Bahasa : Python

2.2 Deskripsi Arsitektural

Arsitektur perangkat lunak ini dirancang dengan pendekatan modular yang terdiri dari empat komponen utama, yaitu komponen Proyek, komponen Tugas, komponen Biaya, dan komponen Inspirasi. Komponen proyek bertindak sebagai komponen sentral yang terhubung dengan komponen Tugas, dan kedua komponen tersebut terhubung dengan komponen Biaya. Terpisah dari ketiga komponen tersebut, terdapat komponen Inspirasi yang berdiri sendiri.



Gambar 1. Deskripsi Arsitektural

2.3 Deskripsi Komponen

Tabel 2.3. Daftar Modul

No	Nama Komponen	Keterangan
1.	Proyek	Komponen yang mengelola dan menampung data setiap proyek dari pengguna.
2.	Tugas	Komponen yang mengelola dan menampung data setiap tugas yang terikat dengan suatu proyek.
3.	Biaya	Komponen yang menampung dan mengelola data pembiayaan setiap tugas dan proyek.

No	Nama Komponen	Keterangan
4.	Inspirasi	Komponen yang mengelola dan menampung data inspirasi renovasi yang dimiliki pengguna.

3 Perancangan Rinci

3.1 Realisasi Use Case

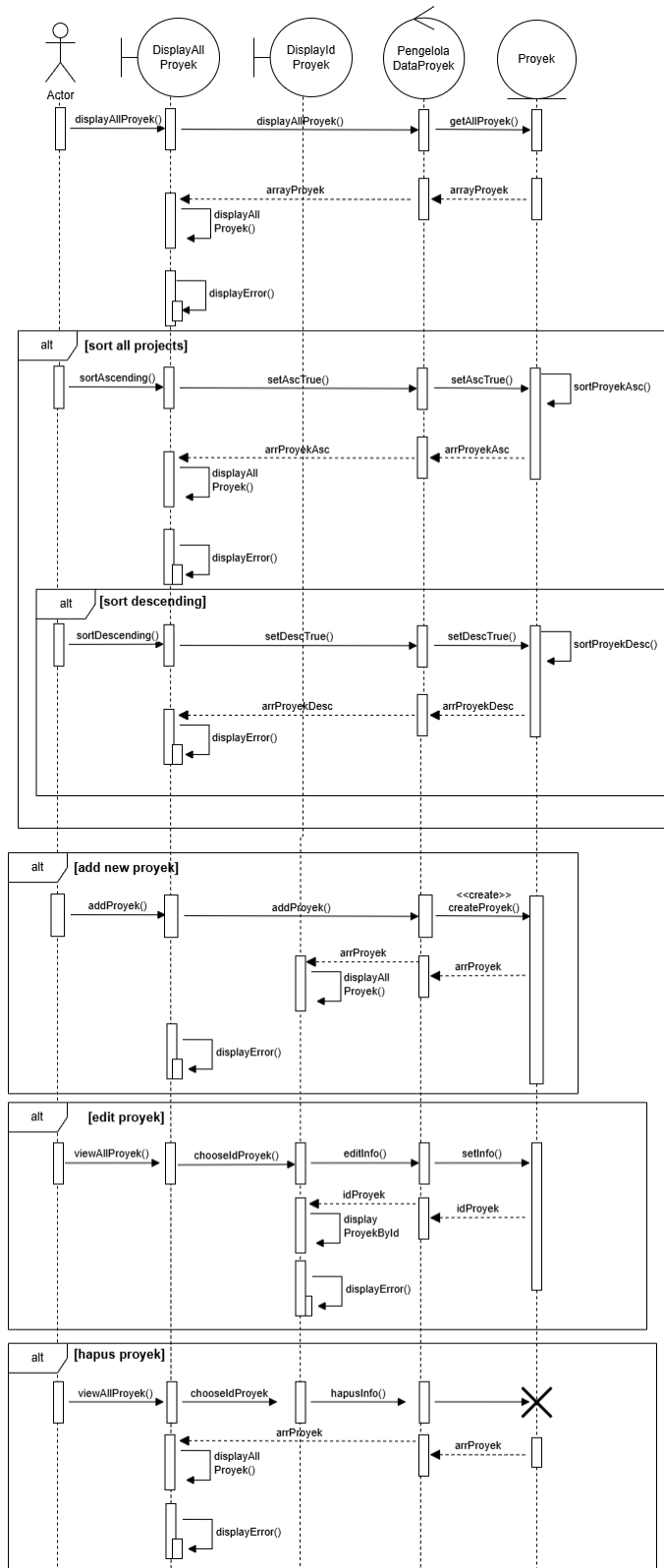
3.1.1 Use Case Mengelola Seluruh Data Proyek Renovasi

3.1.1.1 Identifikasi Kelas

Tabel 3.1.1.1. Daftar Kelas pada Use Case 01

No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait
1	DisplayDataProyek	DisplayDataProyek
2	PengelolaDataProyek	PengelolaDataProyek
3	Proyek	Proyek

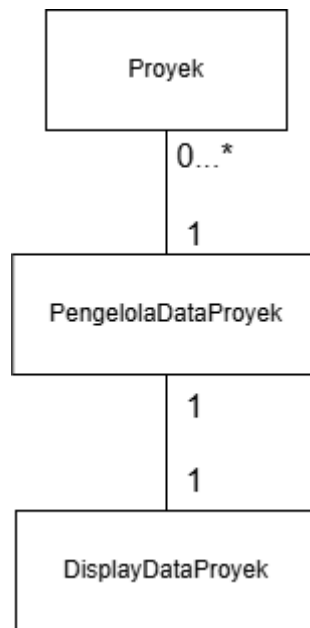
3.1.1.2 Sequence Diagram



Gambar 2. Diagram Sequence Proyek Renovasi

Untuk keterbacaan yang lebih baik, berikut adalah link [sequence diagram mengelola seluruh data proyek](#).

3.1.1.3 Diagram Kelas



Gambar 3. Diagram Kelas Proyek Renovasi

Tabel 3.1.1.3. Tabel Kelas Use Case 01

ID Kelas	Nama Kelas	Atribut	Metode
C-01	Proyek	proyek_id	get_proyek_id(<i>proyek_id</i>)
			set_proyek_id(<i>proyek_id</i>)
		proyek_nama	get_proyek_nama(<i>proyek_id</i>)
			set_proyek_nama(<i>proyek_id</i>)
		proyek_deskripsi	get_proyek_deskripsi(<i>proyek_id</i>)
			set_proyek_deskripsi(<i>proyek_id</i>)
		proyek_tanggal	get_proyek_tanggal_mulai(<i>proyek_id</i>)
			set_proyek_tanggal_mulai(<i>proyek_id</i>)
			set_proyek_tanggal_selesai(<i>proyek_id</i>)
			get_proyek_tanggal_selesai(<i>proyek_id</i>)
		proyek_progress	get_proyek_progress(<i>proyek_id</i>)
		proyek_status	get_status(<i>proyek_id</i>)
			set_status(<i>proyek_id</i>)

C-02	PengelolaDataProyek		add_proyek (proyek_id, proyek_nama, proyek_deskripsi, proyek_tanggal_mulai)
			delete_proyek(proyek_id)
			update_proyek (proyek_id,proyek_nama, proyek_deskripsi, proyek_tanggal_mulai, proyek_progress)
			get_all_proyek(,)
			filter_proyek (proyek_progress)
			sort_proyek (ascending/descending)
C-03	DisplayDataProyek		display_proyek (proyek_id)

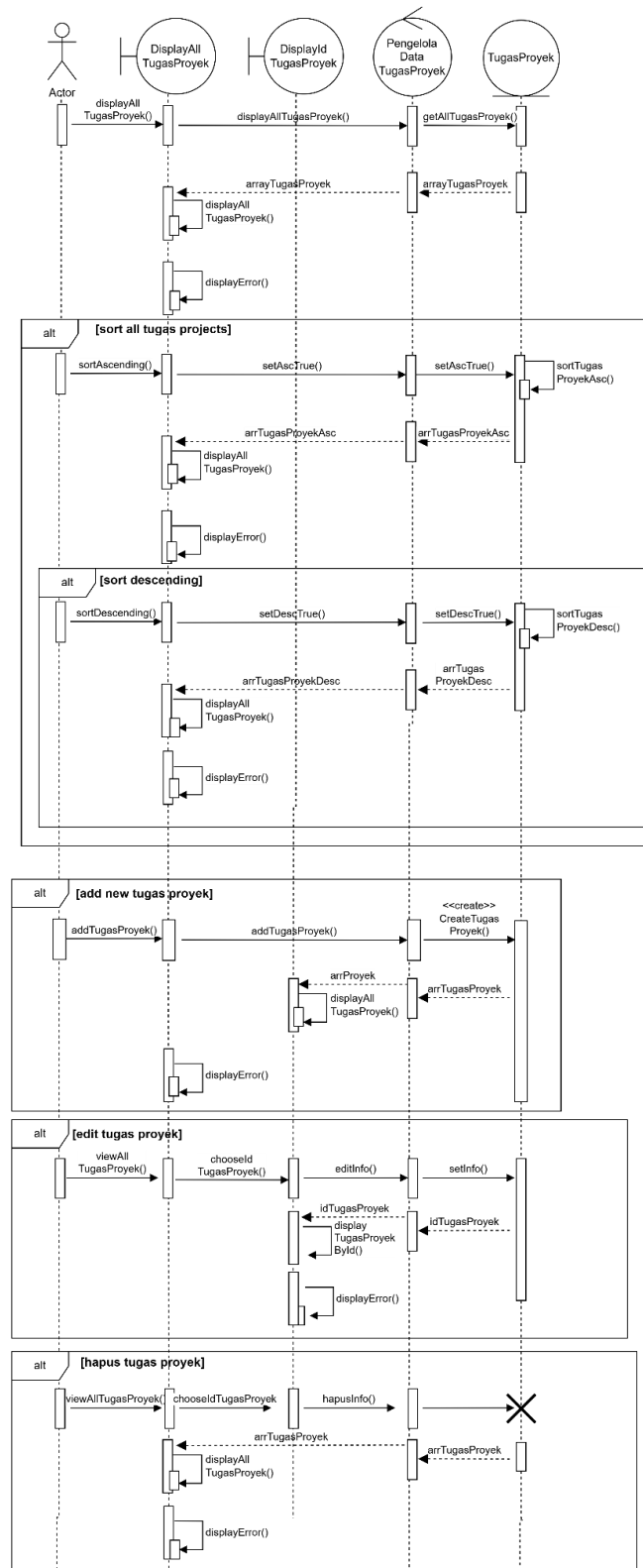
3.1.2 Use Case Mengelola Seluruh Data Tugas dari Setiap Proyek Renovasi

3.1.2.1 Identifikasi Kelas

Tabel 3.1.2.1. Daftar Kelas pada Use Case 02

No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait
1	DisplayTugas	DisplayTugas
2	PengelolaTugasProyek	PengelolaTugasProyek
3	Tugas	Tugas

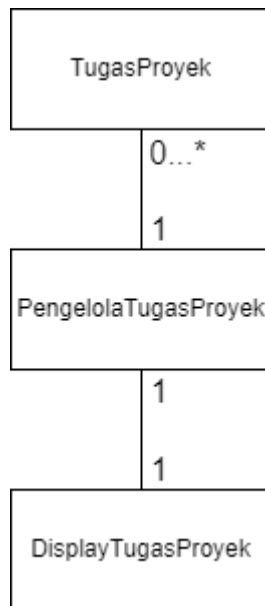
3.1.2.2 Sequence Diagram



Gambar 4. Diagram Sequence Pengelola Proyek

Untuk keterbacaan yang lebih baik, berikut adalah [link sequence diagram mengelola seluruh data tugas dari setiap proyek](#).

3.1.2.3 Diagram Kelas



Gambar 5. Diagram Kelas Pengelola Proyek

Tabel 3.1.2.3. Tabel Kelas Use Case 02

ID Kelas	Nama Kelas	Atribut	Metode
C-04	Tugas Proyek	tugas_id	get_tugas_id(<i>tugas_id</i>)
			set_tugas_id(<i>tugas_id</i>)
		tugas_nama	get_tugas_nama(<i>tugas_id</i> , <i>tugas_nama</i>)
			set_tugas_nama(<i>tugas_id</i> , <i>tugas_nama</i>)
		tugas_deskripsi	get_tugas_deskripsi(<i>tugas_id</i> , <i>tugas_deskripsi</i>)
			set_tugas_deskripsi(<i>tugas_id</i> , <i>tugas_deskripsi</i>)
		tugas_status	get_tugas_status(<i>tugas_id</i> , <i>tugas_status</i>)
			set_tugas_status(<i>tugas_id</i> , <i>tugas_status</i>)
		proyek_id	get_proyek_id(<i>proyek_id</i>)
			set_proyek_id(<i>proyek_id</i>)
C-05	Pengelola Tugas Proyek		add_tugas (<i>tugas_id</i> , <i>tugas_nama</i> , <i>tugas_deskripsi</i> , <i>tugas_status</i>)
			delete_tugas (<i>tugas_id</i> , <i>tugas_nama</i> , <i>tugas_deskripsi</i> , <i>tugas_status</i>)
			update_tugas (<i>tugas_id</i> , <i>tugas_nama</i> , <i>tugas_deskripsi</i> , <i>tugas_status</i>)
			get_all_tugas (<i>tugas_id</i> .)
C-06	Display Tugas Proyek		display_tugas (<i>tugas_id</i>)

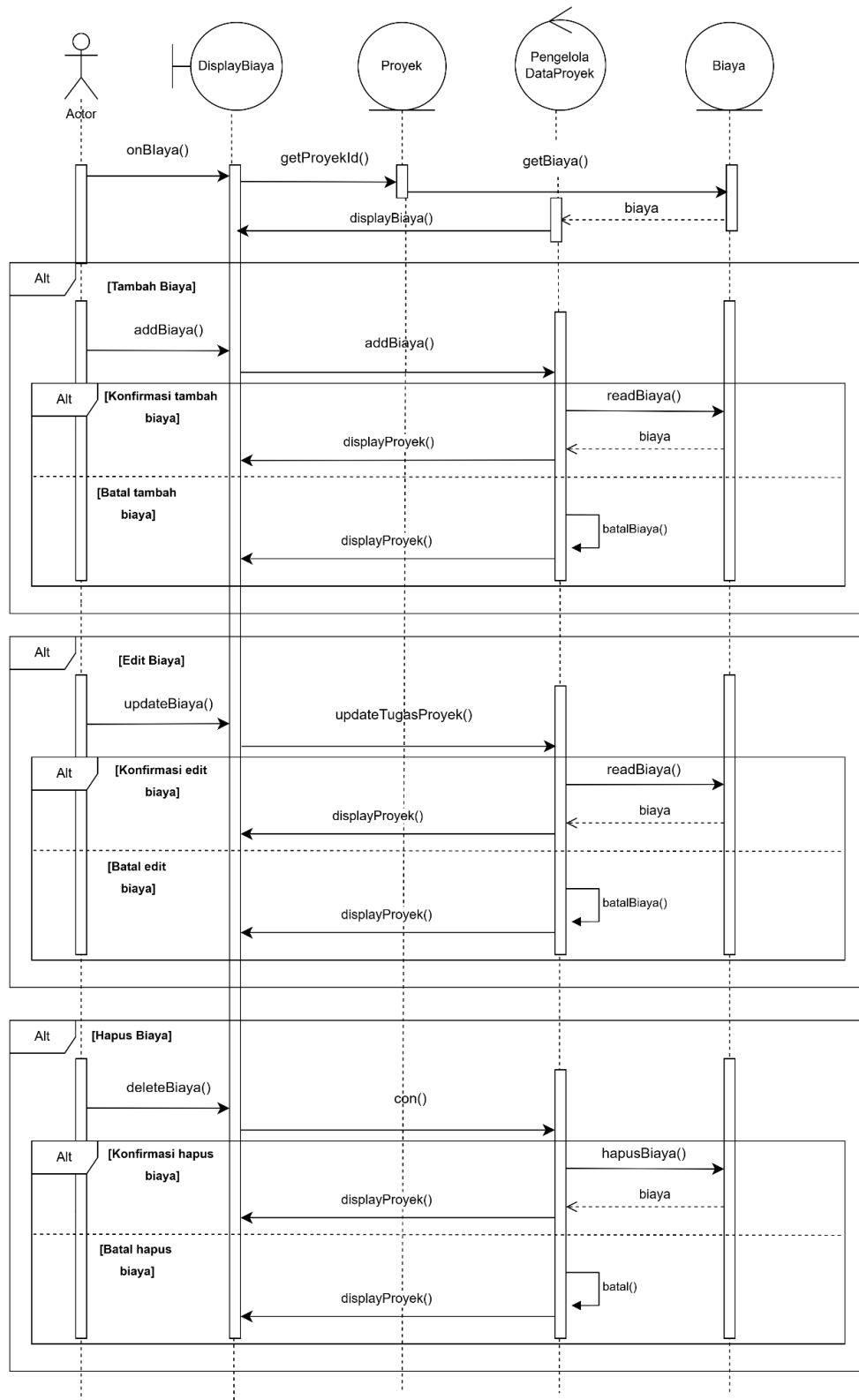
3.1.3 Use Case Mengelola Seluruh Data Biaya pada Setiap Tugas

3.1.3.1 Identifikasi Kelas

Tabel 3.1.3.1. Daftar Kelas Use Case 03

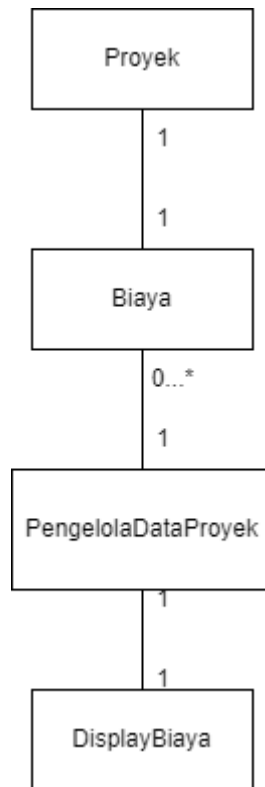
No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait
1	DisplayBiaya	DisplayBiaya
2	PengelolaDataProyek	PengelolaDataProyek
3	Proyek	Proyek
4	Biaya	Biaya

3.1.3.2 Sequence Diagram



Gambar 6. Diagram Sequence Pengelola Biaya

3.1.3.3 Diagram Kelas



Gambar 7. Diagram Kelas PengelolaBiaya

Tabel 3.1.3.3. Tabel Kelas Use Case 03

ID Kelas	Nama Kelas	Atribut	Metode
C-07	Biaya	biaya_id biaya_nilai proyek_id tugas_id	getBiaya(displayBiaya(p royek_id) readBiaya(displayBiaya (proyek_id) akumulasiBiaya(display Biaya(proyek_id)
C-01	Proyek	proyek_id	get_proyek_id(proyek_id) set_proyek_id(proyek_id)
		proyek_nama	get_proyek_nama(proyek _id) set_proyek_nama(proyek _id)
		proyek_deskripsi	get_proyek_deskripsi(pr oyek_id)
C-02	Pengelola Data Proyek		add_proyek (proyek_id, proyek_nama, proyek_deskripsi, proyek_tanggal_mulai)
			delete_proyek(proyek_id)
			update_proyek (proyek_id,proyek_nama, proyek_deskripsi, proyek_tanggal_mulai, proyek_progress)

			get_all_proyek(<i>proyek_id</i>)
C-08	Display Biaya		displayBiaya(<i>proyek_id</i>)

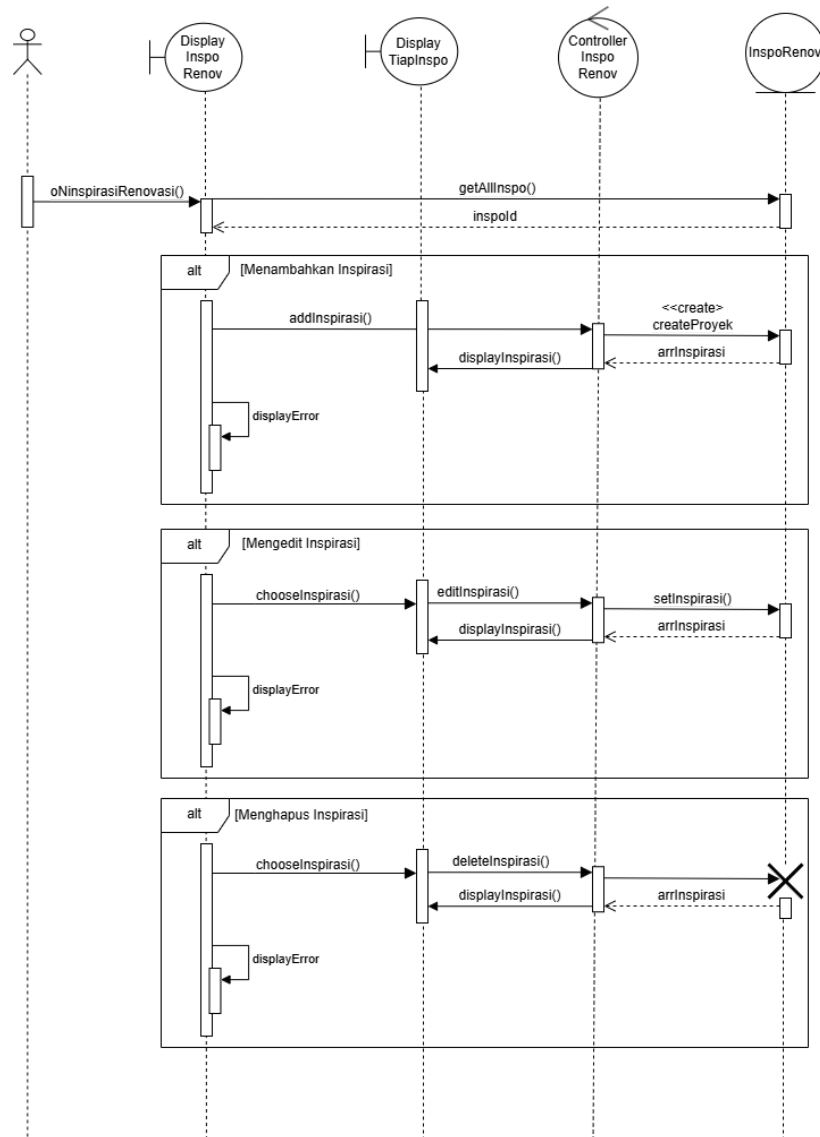
3.1.4 Use Case Mengelola Seluruh Data Inspirasi Renovasi

3.1.4.1 Identifikasi Kelas

Tabel 3.1.4.1. Tabel Kelas Use Case 04

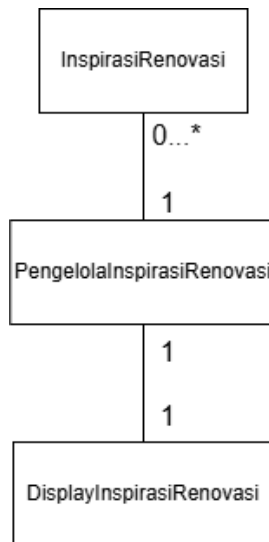
No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait
1	DisplayRenovasiRumah	DisplayRenovasiRumah
2	PengelolaRenovasiRumah	PengelolaRenovasiRumah
3	InspirasiRumah	InspirasiRumah

3.1.4.2 Sequence Diagram



Gambar 8. Diagram Sequence Pengelola Inspirasi Renovasi

3.1.4.3 Diagram Kelas



Gambar 9. Diagram Kelas Pengelola Inspirasi Renovasi

Tabel 3.1.4.3. Tabel Kelas Use Case 04

ID Kelas	Nama Kelas	Atribut	Metode
C-09	InspirasiRenovasi	inspirasi_id	get_inspirasi_id (inspirasi_id)
			set_inspirasi_id(inspirasi_id)
		nama_inspirasi	get_inspirasi_nama(inspirasi_id)
			set_inspirasi_nama(inspirasi_id)
		detail_inspirasi	get_inspirasi_detail (inspirasi_id)
			set_inspirasi_detail(inspirasi_id)
C-10	PengelolaInspirasiRenovasi		add_inspirasi (inspirasi_id, nama_inspirasi, detail_inspirasi)
			delete_inspirasi (inspirasi_id, nama_inspirasi, detail_inspirasi)
			update_inspirasi (inspirasi_id, nama_inspirasi, detail_inspirasi)
			get_all_inspirasi (inspirasi_id)
C-11	DisplayInspirasiRenovasi		display_inspirasi (inspirasi_id)

3.2 Perancangan Detil Kelas

Berikut adalah daftar seluruh kelas yang ada dalam program Rebuilt.

Tabel 3.2. Daftar Kelas pada Rebuilt

No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait
1.	Proyek	Proyek
2.	PengelolaDataProyek	PengelolaDataProyek
3.	DisplayDataProyek	DisplayDataProyek
4.	Tugas	Tugas
5.	PengelolaDataTugas	PengelolaDataTugas
6.	DisplayDataTugas	DisplayDataTugas
7.	InspirasiRenovasi	InspirasiRenovasi
8.	PengelolaInspirasiRenovasi	PengelolaInspirasiRenovasi
9.	DisplayInspirasiRenovasi	DisplayInspirasiRenovasi
10.	DisplayBiaya	DisplayBiaya
11.	Biaya	Biaya
12.	PengelolaBiaya	PengelolaBiaya

3.2.1 Kelas Proyek

Nama Kelas : Proyek

Tabel 3.2.1. Daftar Operasi dan Atribut Kelas Proyek

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
setNamaProyek(namaProyek)	<i>private</i>	mengubah atribut nama menjadi namaProyek
setDescProyek(descProyek)	<i>private</i>	mengubah atribut desc menjadi descProyek
setProgressProyek(progressProyek)	<i>private</i>	mengubah atribut progress menjadi progressProyek
setBiayaProyek(biayaProyek)	<i>private</i>	mengubah atribut biaya menjadi biayaProyek
setEstimasiBiayaProyek(estimasiBiayaProyek)	<i>private</i>	mengubah atribut estimasiBiaya menjadi estimasiBiayaProyek
setProyekTanggal(tanggalProyek)	<i>private</i>	mengubah atribut tanggal menjadi tanggalProyek
setProyekStatus(statusProyek)	<i>private</i>	mengubah atribut status menjadi statusProyek
setProyekId(idProyek)	<i>private</i>	mengubah atribut id menjadi idProyek
getNamaProyek(namaProyek)	<i>private</i>	mereturn atribut nama
getDescProyek(descProyek)	<i>private</i>	mereturn atribut desc
getProgressProyek(progressProyek)	<i>private</i>	mereturn atribut progress
getBiayaProyek(biayaProyek)	<i>private</i>	mereturn atribut biaya
getEstimasiBiayaProyek(estimasiBiayaProyek)	<i>private</i>	mereturn atribut estimasiBiaya
getProyekTanggal(tanggalProyek)	<i>private</i>	mereturn atribut tanggal
getProyekStatus(statusProyek)	<i>private</i>	mereturn atribut status
getProyekId(idProyek)	<i>private</i>	mereturn atribut id
addBiayaProyek(cost)	<i>private</i>	menambah atribut biaya sebesar cost

subtractBiayaProyek(cost)	<i>private</i>	mengurangi atribut biaya sebesar cost
incrementId()	<i>private</i>	menambah atribut id sebanyak 1
decrementId()	<i>private</i>	mengurangi atribut id sebanyak 1
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
initialized	<i>private</i>	boolean
proyek id	<i>private</i>	integer
nama	<i>public</i>	string
desc	<i>public</i>	string
progress	<i>public</i>	float
biaya	<i>public</i>	integer
estimasi biaya	<i>public</i>	integer
tanggal mulai	<i>public</i>	ADT tanggal
tanggal selesai	<i>public</i>	ADT tanggal
status	<i>public</i>	string

3.2.1.1 *Algoritma/Query*

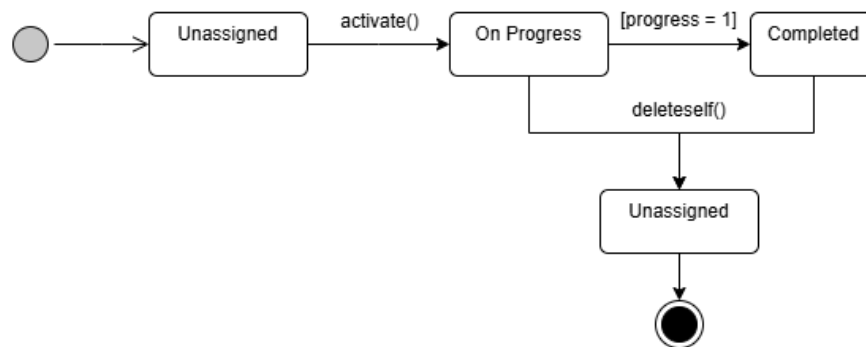
Query : *Proyek*

Tabel 3.2.1.1. Daftar Operasi Query Kelas Proyek

<i>No Query</i>	<i>Query</i>	<i>Keterangan</i>
<i>Q-01-01</i>	SELECT nama FROM t_proyek WHERE proyek_id = id_input	Mengambil data nama proyek dengan proyek_id = id_input
<i>Q-01-02</i>	SELECT desc FROM t_proyek WHERE proyek_id = id_input	Mengambil data desc proyek dengan proyek_id = id_input
<i>Q-01-03</i>	SELECT progress FROM t_proyek WHERE proyek_id = id_input	Mengambil data progress proyek dengan proyek_id = id_input
<i>Q-01-04</i>	SELECT biaya FROM t_proyek WHERE proyek_id = id_input	Mengambil data biaya proyek dengan proyek_id = id_input
<i>Q-01-05</i>	SELECT estimasi_biaya FROM t_proyek WHERE proyek_id = id_input	Mengambil data estimasi_biaya proyek dengan proyek_id = id_input
<i>Q-01-06</i>	SELECT tanggal_mulai FROM t_proyek WHERE proyek_id = id_input	Mengambil data tanggal_mulai proyek dengan proyek_id = id_input
<i>Q-01-07</i>	SELECT tanggal_selesai FROM t_proyek WHERE proyek_id = id_input	Mengambil data tanggal_selesai proyek dengan proyek_id = id_input
<i>Q-01-08</i>	SELECT status FROM t_proyek WHERE proyek_id = id_input	Mengambil data status proyek dengan proyek_id = id_input
<i>Q-01-01</i>	UPDATE t_proyek SET nama = nama_input WHERE proyek_id = id_input	Mengubah data nama proyek dengan proyek_id = id_input
<i>Q-01-02</i>	UPDATE t_proyek SET desc= desc_input WHERE proyek_id = id_input	Mengubah data desc proyek dengan proyek_id = id_input
<i>Q-01-03</i>	UPDATE t_proyek SET progress = progress_input WHERE proyek_id = id_input	Mengubah data progress proyek dengan proyek_id = id_input
<i>Q-01-04</i>	UPDATE t_proyek SET biaya = biaya_input	Mengubah data biaya proyek dengan proyek_id = id_input

	WHERE proyek_id = id_input	
Q-01-05	UPDATE t_proyek SET estimasi_biaya = estimsi_biaya_input WHERE proyek_id = id_input	Mengubah data estimasi_biaya proyek dengan proyek_id = id_input
Q-01-06	UPDATE t_proyek SET tanggal_mulai = tanggal_mulai_input WHERE proyek_id = id_input	Mengubah data tanggal_mulai proyek dengan proyek_id = id_input
Q-01-07	UPDATE t_proyek SET tanggal_selesai = tanggal_selesai_input WHERE proyek_id = id_input	Mengubah data tanggal_selesai proyek dengan proyek_id = id_input
Q-01-08	UPDATE t_proyek SET progress= progress_input WHERE proyek_id = id_input	Mengubah data status proyek dengan proyek_id = id_input

3.2.1.2 *Diagram Statechart*

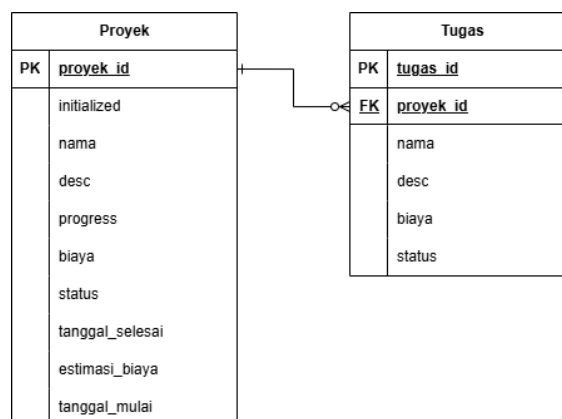


Gambar 10 Diagram Statechart Proyek

3.2.1.3 *Perancangan Antarmuka*

Kelas bertipe *entity* tidak memiliki antarmuka.

3.2.1.4 *Perancangan Representasi Persistensi Kelas*



3.2.2 Kelas PengelolaDataProyek

Nama Kelas : *PengelolaDataProyek*

Tabel 3.2.2. Daftar Operasi dan Atribut Kelas PengelolaDataProyek

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
addProyek (id, nama, desc, tanggal, estimasi_biaya)	<i>public</i>	Untuk membuat objek/ <i>entity</i> proyek baru dengan rincian atribut sesuai argumen.
getProyekById (id)	<i>public</i>	Untuk mengambil data proyek dengan id proyek = id input.
deleteProyek (id)	<i>public</i>	Menghapus obyek proyek dengan atribut id proyek = id.
editProyek (id, newNama, newDesc, newEstimasiBiaya)	<i>public</i>	Mengedit atribut obyek dengan id_proyek = id sesuai dengan input pada argumen.
getAllProyek ()	<i>public</i>	Mengambil data semua obyek proyek yang ada pada <i>database</i>
sortProyek (order)	<i>public</i>	Mengurutkan data seluruh proyek berdasarkan <i>progress</i> sesuai dengan <i>order</i> . <i>Order</i> dapat <i>ascending</i> atau <i>descending</i> .
filterProyek (status)	<i>public</i>	Memfilter data seluruh proyek berdasarkan status yang diminta.
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
-	-	-

3.2.2.1 Algoritma/Query

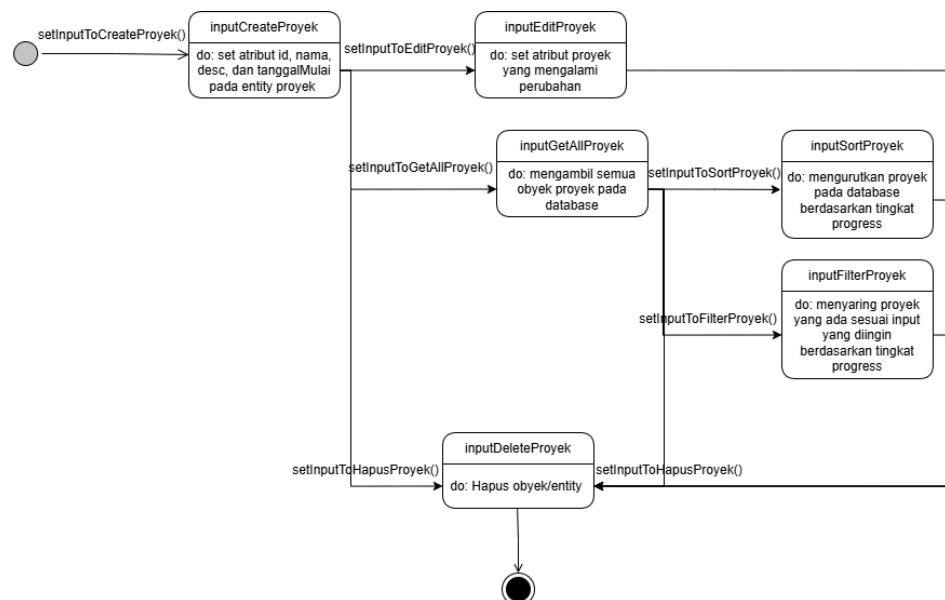
Query : *PengelolaDataProyek*

Tabel 3.2.2.1. Daftar Operasi Query Kelas PengelolaDataProyek

<i>No Query</i>	<i>Query</i>	<i>Keterangan</i>
<i>Q-02-01</i>	SELECT * FROM t_proyek ORDER BY proyek id DESC	Mengambil semua data proyek secara descending berdasarkan proyek id
<i>Q-02-02</i>	INSERT INTO t_proyek(nama_proyek, desc_proyek, estimasi_biaya_proyek, tanggal_mulai_proyek) VALUES(nama,desc,estimasi_biaya,tanggal_mulai)	Menambahkan proyek baru ke dalam t_proyek berisikan rincian sesuai input pada argumen
<i>Q-02-03</i>	DELETE FROM t_proyek WHERE proyek_id = proyek_id_dihapus	Menghapus proyek dari t_proyek dengan proyek_id = proyek_id_dihapus

Q-02-04	UPDATE t_proyek SET nama = new_nama, desc = new_desc, progress = new_progress, biaya = new_biaya, tanggal_selesai = new_tanggal_selesai, status = new_status WHERE proyek_id = proyek_id_edited	Mengedit atribut pada data proyek dengan proyek_id = proyek_id_edited sesuai dengan masukan yang diberikan
Q-02-05	SELECT * FROM t_proyek ORDER BY progress DESC	Mengambil semua data proyek secara descending berdasarkan atribut progress
Q-02-06	SELECT * FROM t_proyek ORDER BY progress ASC	Mengambil semua data proyek secara ascending berdasarkan atribut progress
Q-02-07	SELECT proyek FROM t_proyek WHERE proyek_id = proyek_input	Mengambil data proyek dengan id tertentu dimana proyek_id = proyek_input
Q-02-08	SELECT * FROM t_proyek WHERE status = input_status	Mengambil data semua proyek dengan status = input_status

3.2.2.2 Diagram Statechart



Gambar 12. Diagram Statechart PengelolaDataProyek

3.2.2.3 Perancangan Antarmuka

Kelas bertipe *controller* tidak memiliki antarmuka.

3.2.2.4 Perancangan Representasi Persistensi Kelas

Kelas bertipe *controller* sehingga tidak dapat disusun ERD.

3.2.3 Kelas DisplayDataProyek

Bagian ini diisi dengan daftar operasi dan atribut Buat untuk setiap kelas.

Nama Kelas : *DisplayDataProyek*

Tabel 3.2.3. Daftar Operasi dan Atribut Kelas DisplayDataProyek

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
displayAllProyek	public	Menampilkan semua proyek yang ada
displayProyek(id_proyek)	public	Menampilkan halaman data tiap proyek dengan id_proyek = id input
displayProyekEdit(id_proyek)	public	Menampilkan halaman edit proyek
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
id input	public	integer

3.2.3.1 Algoritma/Query

Nama Kelas : DisplayDataProyek

Nama Operasi : displayDataProyek()

Algoritma : (Algo-201)

Initial State (IS)

Terdapat array of proyek yang berisi semua data yang ada dengan ID masing-masing

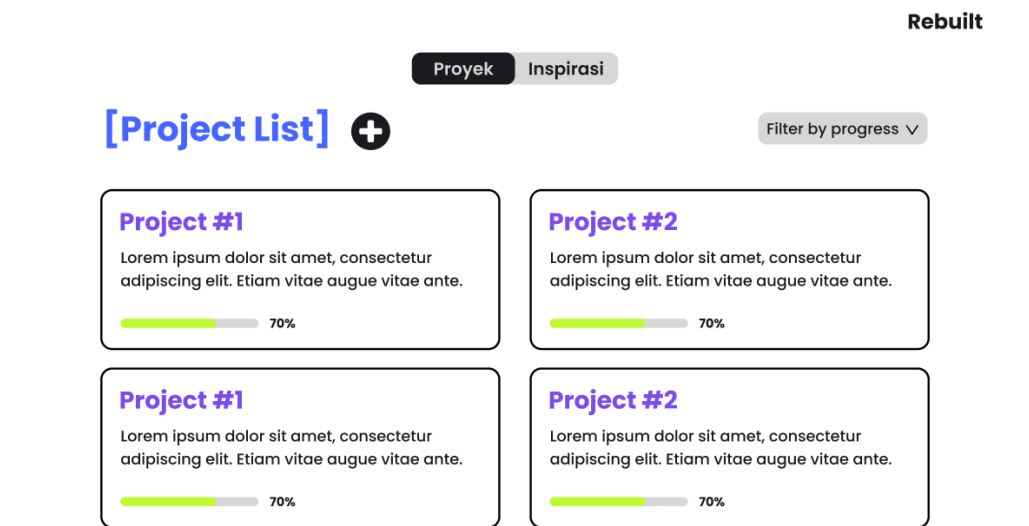
Final State (FS)

Menampilkan proyek sesuai dengan ID yang diminta

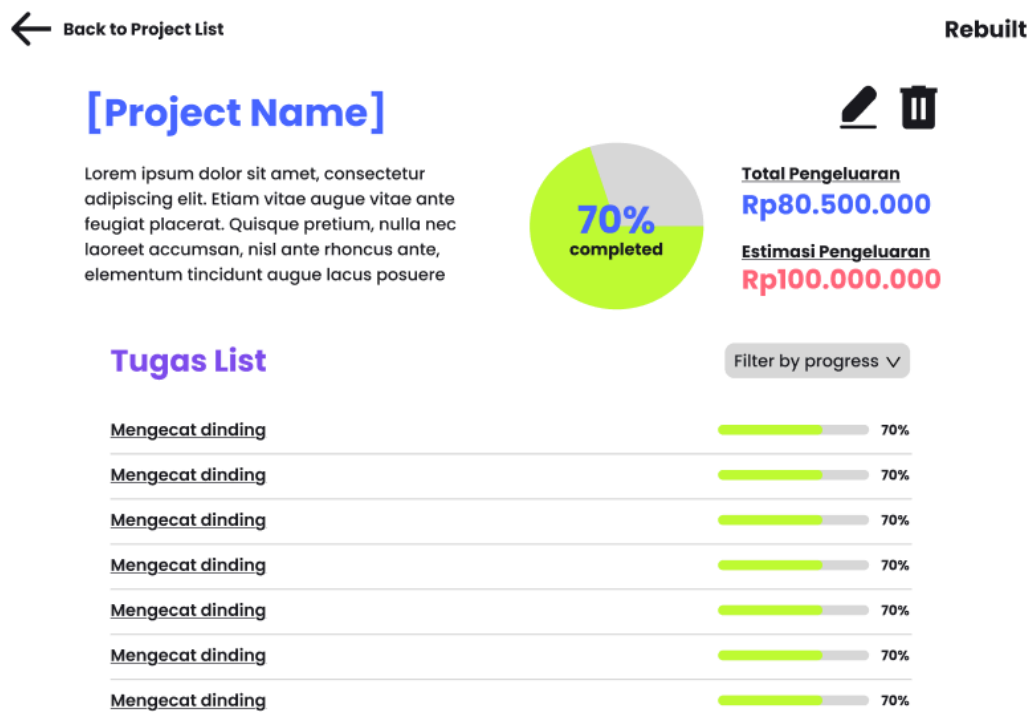
Spesifikasi proses/algoritma :

1. User menekan proyek yang ingin dilihat (memilih proyekId mana yang ingin dilihat)
2. Controller mengambil data proyek dari database by ID
3. Menampilkan proyek sesuai dengan ID

3.2.3.2 Perancangan Antarmuka



Gambar 13. Perancangan Antarmuka displayAllProyek



Gambar 13. Perancangan Antarmuka displayProyek

Tabel 3.2.3.1. Daftar Objek AntarMuka DisplayProyek

Id Objek	Jenis	Nama	Keterangan
CardProject	Box	Project	Jika diklik akan mengarahkan ke tampilan project by ID
ButtonAdd	Button	(icon plus)	Jika diklik, akan menambahkan project yang baru

Id Objek	Jenis	Nama	Keterangan
<i>ButtonEdit</i>	<i>Button</i>	<i>(icon pensil)</i>	<i>Jika diklik akan menampilkan halaman edit proyek</i>
<i>ButtonDelete</i>	<i>Button</i>	<i>(icon tempat sampah)</i>	<i>Jika diklik akan menghapus proyek</i>

3.2.3.3 Perancangan Representasi Persistensi Kelas

Kelas bertipe boundary sehingga tidak dapat disusun ERD.

3.2.4 Kelas TugasProyek

Nama Kelas : Tugas Proyek

Tabel 3.2.4. Daftar Operasi dan Atribut Kelas TugasProyek

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
setNamaTugas(namaTugas)	<i>private</i>	mengubah atribut nama menjadi namaTugas
setDescTugas(descTugas)	<i>private</i>	mengubah atribut desc menjadi descTugas
setBiayaTugas(biayaTugas)	<i>private</i>	mengubah atribut biaya menjadi biayaTugas
setTugasId(idTugas)	<i>private</i>	mengubah atribut id menjadi idTugas
setTugasStatus(statusTugas)	<i>private</i>	mengubah atribut status menjadi statusTugas
getNamaTugas(namaTugas)	<i>private</i>	me-return atribut nama
getDescTugas(descTugas)	<i>private</i>	me-return atribut desc
getBiayaTugas(biayaTugas)	<i>private</i>	me-return atribut biaya
getTugasId(idTugas)	<i>private</i>	me-return atribut id
getTugasStatus(statusTugas)	<i>private</i>	me-return atribut status
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
nama	<i>public</i>	<i>string</i>
desc	<i>public</i>	<i>string</i>
biaya	<i>public</i>	<i>integer</i>
proyek id	<i>private</i>	<i>integer</i>
tugas id	<i>public</i>	<i>integer</i>
status	<i>public</i>	<i>string</i>

3.2.4.1 Algoritma/Query

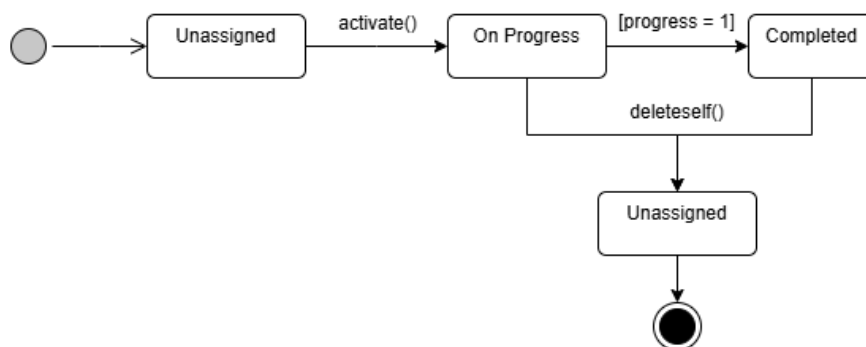
Query :Tugas Proyek

Tabel 3.2.4.1. Daftar Operasi Query Kelas TugasProyek

<i>No Query</i>	<i>Query</i>	<i>Keterangan</i>
<i>Q-04-01</i>	SELECT nama FROM t_tugas WHERE tugas_id = id_input	Mengambil data nama tugas dengan tugas_id = id_input
<i>Q-04-02</i>	SELECT desc FROM t_tugas WHERE tugas_id = id_input	Mengambil data desc tugas dengan tugas_id = id_input
<i>Q-04-03</i>	SELECT status FROM t_tugas WHERE tugas_id = id_input	Mengambil data status tugas dengan tugas_id = id_input
<i>Q-04-04</i>	SELECT biaya FROM t_tugas WHERE tugas_id = id_input	Mengambil data biaya tugas dengan tugas_id = id_input

Q-04-05	UPDATE t_tugas SET nama = nama_input WHERE tugas_id = id_input	Mengubah data nama tugas dengan tugas_id = id_input
Q-04-06	UPDATE t_tugas SET desc = desc_input WHERE tugas_id = id_input	Mengubah data desc tugas dengan tugas_id = id_input
Q-04-07	UPDATE t_tugas SET status = status_input WHERE tugas_id = id_input	Mengubah data status tugas dengan tugas_id = id_input
Q-04-08	UPDATE t_tugas SET biaya = biaya_input WHERE tugas_id = id_input	Mengubah data biaya tugas dengan tugas_id = id_input

3.2.4.2 Diagram Statechart



Gambar 14. Diagram Statechart TugasProyek

3.2.4.3 Perancangan Antarmuka

Kelas bertipe entity tidak memiliki antarmuka.

3.2.4.4 Perancangan Representasi Persistensi Kelas

Tugas	
PK	<u>tugas_id</u>
FK	<u>proyek_id</u>
	nama
	desc
	biaya
	status

Gambar 15. Perancangan Representasi Persistensi Kelas Tugas

3.2.5 Kelas PengelolaTugasProyek

Bagian ini diisi dengan daftar operasi dan atribut Buat untuk setiap kelas.

Nama Kelas : PengelolaTugasProyek

Tabel 3.2.5. Daftar Operasi dan Atribut Kelas PengelolaTugasProyek

Nama Operasi	Visibility	Keterangan
--------------	------------	------------

	<i>(private, public)</i>	
add_tugas (tugas_id, tugas_nama, tugas_deskripsi, tugas_status)	public	Menambahkan tugas ke proyek
delete_tugas (tugas_id, tugas_nama, tugas_deskripsi, tugas_status)	public	menghapus tugas dari proyek
update_tugas (tugas_id, tugas_nama, tugas_deskripsi, tugas_status)	public	memperbaharui isi tugas proyek
get_all_tugas (tugas_id,)	public	me-return list tugas-tugas pada proyek

3.2.5.1 Algoritma/Query

Query : PengelolaTugasProyek

Tabel 3.2.5.1. Daftar Operasi Query Kelas PengelolaTugasProyek

No Query	Query	Keterangan
Q-05-01	SELECT * FROM t_tugas ORDER BY tugas_id DESC	Mengambil semua data tugas secara descending berdasarkan tugas_id
Q-05-02	INSERT INTO t_tugas(nama_tugas, desc_tugas, estimasi_biaya_tugas,desc_tugas) VALUES(nama,desc,estimasi_biaya,tanggal_mulai)	Menambahkan tugas baru ke dalam t_tugas berisikan rincian sesuai input pada argumen
Q-05-03	DELETE FROM t_proyek WHERE tugas_id = tugas_id_dihapus	Menghapus tugas dari t_tugas dengan tugas_id = tugas_id_dihapus
Q-05-04	UPDATE t_tugas SET nama = new_nama, desc = new_desc, biaya = new_biaya, status = new_status WHERE proyek_id = proyek_id_edited	Mengedit atribut pada data proyek dengan proyek_id = proyek_id_edited sesuai dengan masukan yang diberikan
Q-05-05	SELECT * FROM t_tugas ORDER BY progress DESC	Mengambil semua data proyek secara descending berdasarkan atribut progress
Q-05-06	SELECT * FROM t_tugas ORDER BY progress ASC	Mengambil semua data proyek secara ascending berdasarkan atribut progress
Q-05-07	SELECT proyek FROM t_tugas WHERE tugas_id = proyek_input	Mengambil data tugas proyek dengan id tertentu dimana tugas_id = tugas_input
Q-05-08	SELECT * FROM t_tugas WHERE status = input_status	Mengambil data semua proyek dengan status = input_status

3.2.6.1 Diagram Statechart

Bagian ini hanya diisi jika ada kelas yang kompleks. Perubahan status kelas tersebut harus digambarkan dalam bentuk diagram statechart. Boleh dibuat subba per kelas.

3.2.6.2 Perancangan Antarmuka

Kelas bertipe controller tidak memiliki antarmuka.

3.2.6.3 Perancangan Representasi Persistensi Kelas

Kelas bertipe controller sehingga tidak dapat disusun ERD.

3.2.6 Kelas DisplayTugasProyek

Bagian ini diisi dengan daftar operasi dan atribut Buat untuk setiap kelas.

Nama Kelas : DisplayTugasProyek

Tabel 3.2.6. Daftar Operasi dan Atribut Kelas DisplayTugasProyek

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
display_tugas (<i>tugas_id</i>)	<i>public</i>	<i>meng-display tugas kepada user</i>

3.2.6.1 Algoritma/Query

Nama Kelas : DisplayTugasProyek

Nama Operasi : display_tugas(*tugas_id*)

Algoritma : (Algo-0601)

Initial State (IS)

Display tugas proyek belum ditunjukkan

Final State (FS)

Display tugas telah ditunjukkan

Spesifikasi proses/algoritma :

1. User menekan salah satu proyek

3.2.6.2 Diagram Statechart

Bagian ini hanya diisi jika ada kelas yang kompleks. Perubahan status kelas tersebut harus digambarkan dalam bentuk diagram statechart. Boleh dibuat subba per kelas.

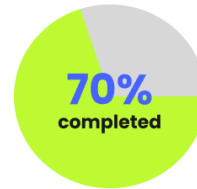
3.2.6.3 Perancangan Antarmuka

← Back to Project Name

Rebuilt

[Tugas Name]

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam vitae augue vitae ante feugiat placerat. Quisque pretium, nulla nec laoreet accumsan, nisl ante rhoncus ante, elementum tincidunt augue lacus posuere



Total Pengeluaran
Rp80.500.000

Estimasi Pengeluaran
Rp100.000.000

Biaya List

Barang	Harga	Qty	Total	Keterangan	Aksi
Semen	Rp9.000	12	Rp108.000		 
Pasir	Rp9.000	12	Rp108.000		 
Pegawai	Rp9.000	12	Rp108.000		 
Cat dinding	Rp9.000	12	Rp108.000		 

Gambar 16. Perancangan DisplayTugasProyek

Tabel 3.2.6.1. Daftar Objek AntarMuka DisplayTugasProyek

Id Objek	Jenis	Nama	Keterangan
ButtonEdit	Button	(icon pensil)	Jika diklik akan menampilkan halaman edit tugas
ButtonDelete	Button	(icon tempat sampah)	Jika diklik akan menghapus tugas
ButtonEditBiaya	Button	(icon pensil)	Jika diklik akan menampilkan halaman edit biaya
ButtonDeleteBiaya	Button	(icon tempat sampah)	Jika diklik akan menghapus biaya
BiayaTable	Table	(tabel)	Menampilkan data biaya

Jika objek dikaitkan ke File lain (misalnya file gambar, file teks), berikan nama file terkait dan deskripsi ringkas dalam kolom keterangan

3.2.6.4 Perancangan Representasi Persistensi Kelas

Kelas bertipe boundary sehingga tidak dapat disusun ERD.

3.2.7 Kelas InspirasiRenovasi

Nama Kelas : InspirasiRenovasi

Tabel 3.2.7. Daftar Operasi dan Atribut Kelas InspirasiRenovasi

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
<i>getInspirasiId(arrayInspiras)</i>	<i>public</i>	<i>mengembalikan id inspirasi renovasi</i>
<i>getDetailInspo(inspirasiId)</i>	<i>public</i>	<i>mengembalikan detail inspirasi renovasi berupa gambar dan deskripsi</i>
<i>setDetailInspo(inspirasiId)</i>	<i>public</i>	<i>mengatur detail inspirasi renovasi berupa gambar dan deskripsi</i>
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>inspirasiId</i>	<i>public</i>	<i>integer</i>
<i>arrayInspirasi</i>	<i>public</i>	<i>array</i>

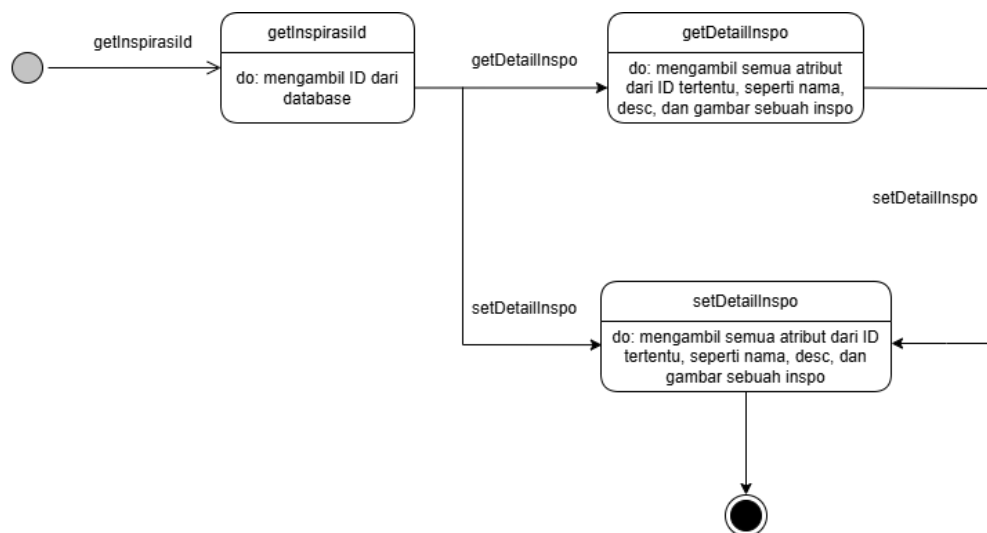
3.2.7.1 Algoritma/Query

Query : InspirasiRenovasi

Tabel 3.2.7.1. Daftar Operasi Query Kelas InspirasiRenovasi

No Query	Query	Keterangan
Q-07-01	SELECT * FROM tabelInspo WHERE inspirasiId = ID	Mengambil data id inspirasi renovasi
Q-07-02	SELECT nama, deskripsi, gambar FROM tabelInspo WHERE inspirasiId = ID	Mengambil data nama, deskripsi, dan gambar inspo dengan id = ID dari inspirasi renovasi
Q-07-03	INSERT INTO tabelInspo VALUES (nama, deskripsi, gambar) WHERE inspirasiId = ID	Menginput data nama, deskripsi, dan gambar inspo dengan id = ID ke tabel inspirasi renovasi

3.2.7.2 Diagram Statechart



Gambar 17. Diagram Statechart InspirasiRenovasi

3.2.7.3 Perancangan Antarmuka

Kelas InspirasiRenovasi tidak memiliki antarmuka.

3.2.7.4 Perancangan Representasi Persistensi Kelas

TabellInspirasi	
PK	<u>inspirasiId</u>
	nama
	deskripsi
	gambar

Gambar 18. Perancangan Representasi Persistensi Kelas Inspirasi

3.2.8 Kelas Pengelola Inspirasi Renovasi

Bagian ini diisi dengan daftar operasi dan atribut Buat untuk setiap kelas.

Nama Kelas : PengelolaInspirasiRenovasi

Tabel 3.2.8. Daftar Operasi dan Atribut Kelas PengelolaInspirasiRenovasi

Nama Operasi	Visibility (private, public)	Keterangan
createInspirasi(inspirasiId)	public	membuat inspirasi renovasi baru
editInspirasi(inspirasiId)	public	mengedit inspirasi renovasi baru
deleteInspirasi(inspirasiId)	public	menghapus inspirasi renovasi baru
Nama Atribut	Visibility (private, public)	Tipe
inspirasiId	public	integer
arrInspirasi	public	array

3.2.8.1 Algoritma/Query

Nama Kelas : PengelolaInspirasiRenovasi

Nama Operasi : createInspirasi()

Algoritma : (Algo-0801)

Initial State (IS)

Inspirasi renovasi belum terbentuk

Final State (FS)

Inspirasi renovasi dengan ID baru telah terbentuk

Spesifikasi proses/algoritma :

4. User menekan add inspirasi
5. Inspirasi renovasi yang baru telah terbuat
6. User memasukkan nama, deskripsi, dan gambar
7. User menekan tombol simpan, database dengan ID tertentu dibuat

Nama Kelas : PengelolaInspirasiRenovasi

Nama Operasi : editInspirasi()

Algoritma : (Algo-0802)

Initial State (IS)

Inspirasi renovasi sudah ada

Final State (FS)

Inspirasi renovasi diperbarui dengan data yang baru

Spesifikasi proses/algoritma :

1. User menekan edit inspirasi
2. User memasukkan data yang baru dan ingin diubah
3. User menekan tombol simpan, database dengan ID tertentu diperbarui

Nama Kelas : *PengelolaInspirasiRenovasi*

Nama Operasi : *deleteInspirasi()*

Algoritma : *(Algo-0803)*

Initial State (IS)

Inspirasi renovasi sudah ada

Final State (FS)

Inspirasi renovasi dengan ID tertentu dihapus

Spesifikasi proses/algoritma :

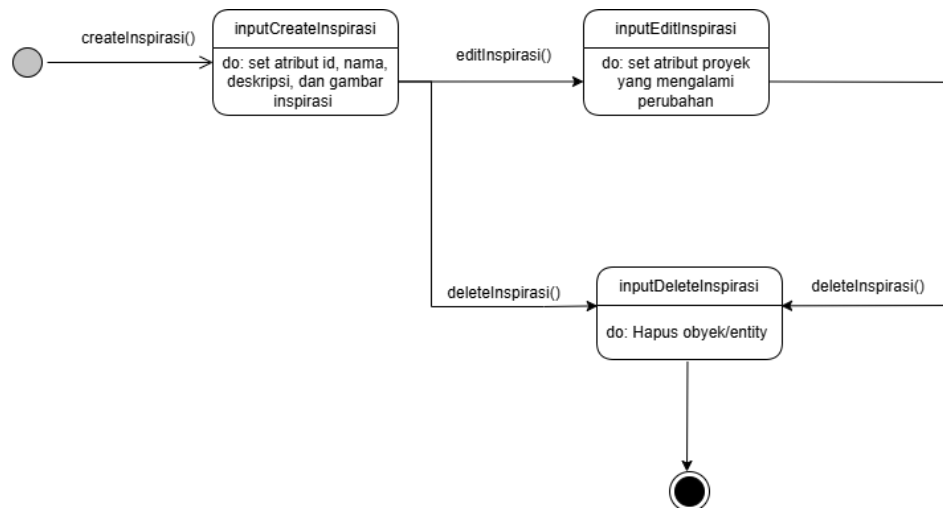
1. User menekan delete inspirasi
2. Muncul pop up konfirmasi delete
3. Ketika user memilih yes, maka database dengan ID tertentu akan dihapus

Query : *PengelolaInspirasiRenovasi*

Tabel 3.2.8.1. Daftar Operasi Query Kelas PengelolaInspirasiRenovasi

No Query	Query	Keterangan
Q-0801	INSERT INTO tabelInspo VALUES (nama, deskripsi, gambar) WHERE inspirasiId = ID	Menambah inspirasi renovasi baru
Q-0802	UPDATE tabelInspo SET (nama, deskripsi, gambar) WHERE inspirasiId = ID	Mengedit inspirasi renovasi yang sudah ada
Q-0803	DELETE FROM tabelInspo VALUES (nama, deskripsi, gambar) WHERE inspirasiId = ID	Menghapus inspirasi renovasi yang sudah ada

3.2.8.2 Diagram Statechart



Gambar 19. Diagram Statechart PengelolaInspirasiRenovasi

3.2.8.3 Perancangan Antarmuka

Kelas PengelolaInspirasiRenovasi tidak memiliki antarmuka.

3.2.9 Kelas Display Inspirasi Renovasi

Bagian ini diisi dengan daftar operasi dan atribut Buat untuk setiap kelas.

Nama Kelas : DisplayInspirasiRenovasi

Tabel 3.2.9. Daftar Operasi dan Atribut Kelas DisplayInspirasiRenovasi

Nama Operasi	Visibility (private, public)	Keterangan
displayAllInspirasi(inspirasiId)	public	Menampilkan halaman berisi seluruh inspirasi
displayInspirasi(inspirasiId)	public	Menampilkan halaman tiap inspirasi renovasi by ID
displayEditInspirasi(inspirasiId)	public	Menampilkan halaman edit inspirasi renovasi
displayKonfirmasiDelete(inspirasiId)	public	Menampilkan halaman pop up konfirmasi delete
displayKonfirmasiSimpan(inspirasiId)	public	Menampilkan halaman pop up konfirmasi perubahan
Nama Atribut	Visibility (private, public)	Tipe
inspirasiId	public	integer

3.2.9.1 Algoritma/Query

Nama Kelas : DisplayInspirasiRenovasi

Nama Operasi : displayInspirasi()

Algoritma : (Algo901)

Initial State (IS)

Terdapat array of inspirasi yang berisi semua inspirasi renovasi yang ada dengan ID masing-masing

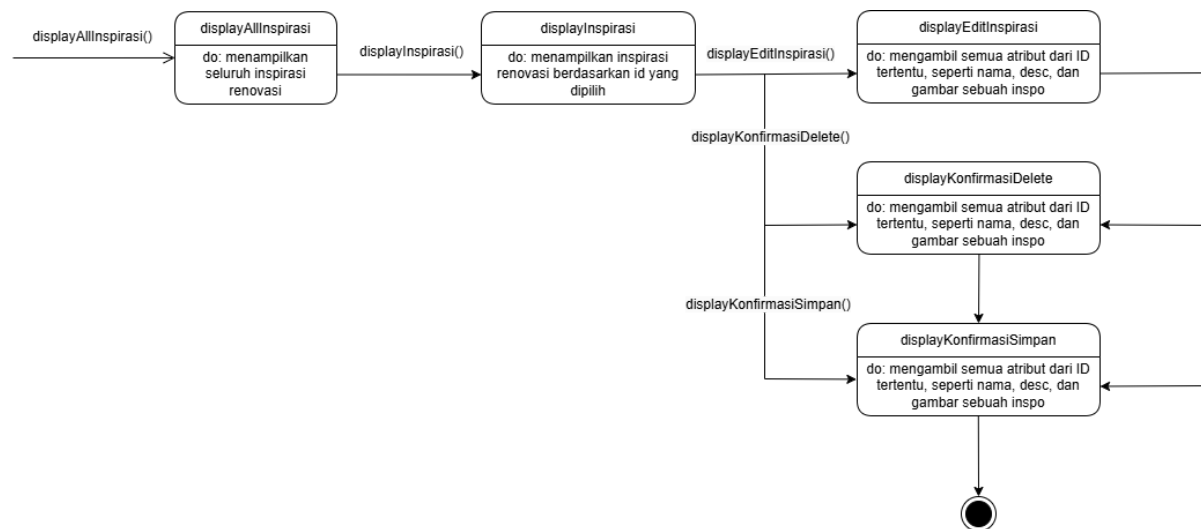
Final State (FS)

Menampilkan inspirasi renovasi sesuai dengan ID yang diminta

Spesifikasi proses/algoritma :

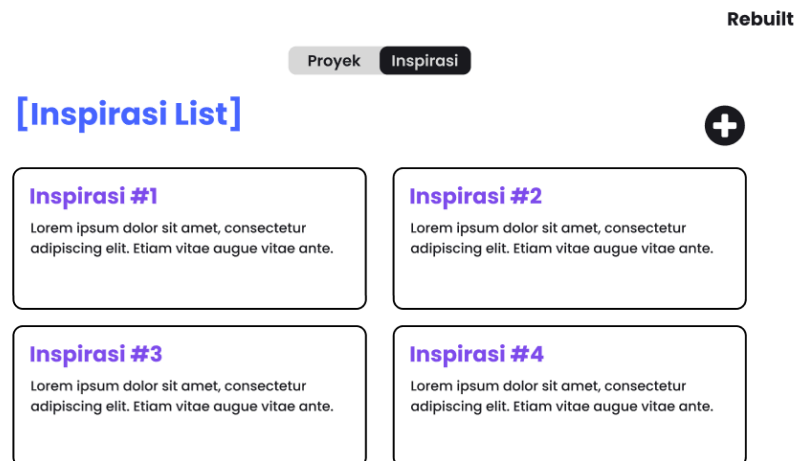
8. User menekan inspirasi renovasi yang ingin dilihat (memilih inspirasiId mana yang ingin dilihat)
9. Controller mengambil data inspirasi renovasi dari database by ID
10. Menampilkan inspirasi renovasi sesuai dengan ID

3.2.9.2 Diagram Statechart

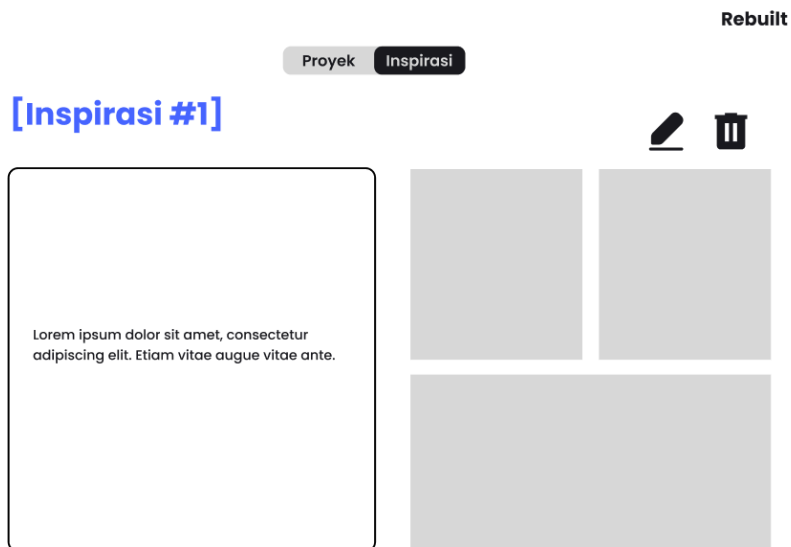


Gambar 20. Diagram Statechart InspirasiRenovasi

3.2.9.3 Perancangan Antarmuka



Gambar 21. Perancangan Antarmuka displayInspirasi



Gambar 22. Perancangan Antarmuka *displayEditInspirasi*



Gambar 23. Perancangan Antarmuka *displayDeleteInspirasi*

Tabel 3.2.9.1. Daftar Objek AntarMuka DisplayInspirasi

Id Objek	Jenis	Nama	Keterangan
<i>Card Inspo</i>	<i>Box</i>	<i>Inspirasi</i>	<i>Jika diklik akan mengarahkan ke tampilan inspirasi renovasi by ID</i>
<i>Button Add</i>	<i>Button</i>	<i>(icon plus)</i>	<i>Jika diklik akan menambahkan inspirasi renovasi yang baru</i>
<i>Button Edit</i>	<i>Button</i>	<i>(icon pensil)</i>	<i>Jika diklik akan menampilkan halaman edit inspirasi renovasi</i>
<i>Button delete</i>	<i>Button</i>	<i>(icon tempat sampah)</i>	<i>Jika diklik akan menghapus inspirasi renovasi</i>
<i>Pop up konfirmasi</i>	<i>Box</i>	<i>Ingin menghapus inspirasi?</i>	<i>Jika pilih iya, maka inspirasi akan dihapus, jika pilih tidak maka inspirasi batas dihapus</i>

3.2.10 Kelas DisplayBiaya

Bagian ini diisi dengan daftar operasi dan atribut Buat untuk setiap kelas.

Nama Kelas : Display Biaya

Tabel 3.2.10. Daftar Operasi dan Atribut Kelas DisplayBiaya

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
displayBiaya(<i>proyek_id</i>)	<i>public</i>	<i>menampilkan halaman data tiap biaua proyek</i>
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>Proyek_id</i>	<i>public</i>	<i>integer</i>

3.2.10.1 Algoritma/Query

Nama Kelas : DisplayBiaya

Nama Operasi : displayBiaya()

Algoritma : (Algo-1001)

Initial State (IS)

Display biaya belum di tunjukan

Final State (FS)

Display Biaya di tunjukkan

Spesifikasi proses/algoritma :

2. User menekan salah satu proyek
3. User menekan biaya

3.2.10.2 Perancangan Antarmuka

← Back to Project Name

Rebuilt







[Tugas Name]

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam vitae augue vitae ante feugiat placerat. Quisque pretium, nulla nec laoreet accumsan, nisl ante rhoncus ante, elementum tincidunt augue lacus posuere



Total Pengeluaran
Rp80.500.000
Estimasi Pengeluaran
Rp100.000.000

Biaya List

Barang	Harga	Qty	Total	Keterangan	Aksi
Semen	Rp9.000	12	Rp108.000		 
Pasir	Rp9.000	12	Rp108.000		 
Pegawai	Rp9.000	12	Rp108.000		 
Cat dinding	Rp9.000	12	Rp108.000		 

Gambar 24. Perancangan Antarmuka displayBiaya

Tabel 3.2.10.1. Daftar Objek AntarMuka DisplayBiaya

Id Objek	Jenis	Nama	Keterangan
ButtonEditBiaya	Button	(icon pensil)	Jika diklik akan menampilkan halaman edit biaya
ButtonDeleteBiaya	Button	(icon tempat sampah)	Jika diklik akan menghapus biaya
BiayaTable	Table	(tabel)	Menampilkan data biaya

Jika objek dikaitkan ke File lain (misalnya file gambar, file teks), berikan nama file terkait dan deskripsi ringkas dalam kolom keterangan

3.2.10.3 Perancangan Representasi Persistensi Kelas

Bagian ini diisi dengan rancangan skema basis data dan traceability-nya terhadap kelas entity.

3.2.11 Kelas Biaya

Nama Kelas : Biaya

Tabel 3.2.11. Daftar Operasi dan Atribut Kelas Biaya

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

getBiaya(displayBiaya(projek_id)	private	me-return atribut biaya
readBiaya(displayBiaya(projek_id)	private	meneruma masukan biaya
akumulasiBiaya(displayBiaya(projek_id)	private	mengakumulasi total biaya
Nama Atribut	Visibility (private, public)	Tipe
biaya ID	private	integer
biaya nilai	public	integer

3.2.11.1 Algoritma/Query

Bagian ini hanya diisi untuk kerangka algoritma untuk proses-proses yang dianggap cukup penting. Implementasi skeleton code juga sudah dapat dilakukan untuk kelas-kelas yang terdefinisi pada bahasa pemrograman tertentu. Boleh dibuat subbab per kelas.

Nama Kelas : Biaya

Nama Operasi : getBiaya()

Algoritma : (Algo-1101)

Initial State (IS)

Atribut biaya belum dikembalikan.

Final State (FS)

Atribut biaya berhasil dikembalikan sesuai dengan projek_id.

Spesifikasi proses/algoritma :

1. Sistem menerima masukan projek_id dari user.
2. Sistem mencari data biaya berdasarkan projek_id di database.
3. Sistem mengembalikan atribut biaya_ID dan biaya_nilai terkait projek_id.

Nama Kelas : Biaya

Nama Operasi : readBiaya()

Algoritma : (Algo-1102)

Initial State (IS)

Sistem belum menerima masukan biaya dari user.

Final State (FS)

Sistem menerima dan menyimpan data biaya baru sesuai projek_id.

Spesifikasi proses/algoritma :

1. User memasukkan data biaya_ID, biaya_nilai, dan projek_id.
2. Sistem memvalidasi data yang dimasukkan.
3. Sistem menyimpan data baru ke dalam tabel biaya.

Nama Kelas : Biaya

Nama Operasi : AkumulasiBiaya()

Algoritma : (Algo-1103)

Initial State (IS)

Total biaya proyek belum dihitung.
Final State (FS) Total biaya untuk suatu proyek berhasil dihitung dan ditampilkan.
Spesifikasi proses/algoritma : <ol style="list-style-type: none"> 1. Sistem menerima proyek_id dari user. 2. Sistem menjumlahkan seluruh nilai biaya (biaya_nilai) berdasarkan proyek_id. 3. Sistem menampilkan total biaya yang dihitung.

{Jika mengacu query tertentu, lengkapi tabel query di bawah}

Query : Biaya

Tabel 3.2.11.1 Daftar Operasi Query Kelas Biaya

No Query	Query	Keterangan
Q-1101	SELECT biaya_ID, biaya_nilai FROM tabelBiaya WHERE proyek_id = 'parameter_proyek_id';	Mengambil atribut biaya berdasarkan proyek ID
Q-1102	INSERT INTO tabelBiaya (biaya_ID, biaya_nilai, proyek_id) VALUES ('parameter_biaya_ID', 'parameter_biaya_nilai', 'parameter_proyek_id');	Menambahkan data biaya baru ke database
Q-1103	SELECT SUM(biaya_nilai) AS total_biaya FROM tabelBiaya WHERE proyek_id = 'parameter_proyek_id';	Menghitung total biaya proyek berdasarkan ID

3.2.11.2 Perancangan Antarmuka

Kelas bertipe entity tidak memiliki antarmuka.

3.2.11.3 Perancangan Representasi Persistensi Kelas

Bagian ini diisi dengan rancangan skema basis data dan traceability-nya terhadap kelas entity.

3.2.12 Kelas PengelolaBiaya

Nama Kelas : PengelolaBiaya

Tabel 3.2.12. Daftar Operasi dan Atribut Kelas PengelolaBiaya

Nama Operasi	Visibility (private, public)	Keterangan
addBiaya(id, nama, desc, tanggal, estimasi_biaya)	public	Untuk membuat objek/entity proyek baru dengan rincian atribut sesuai argumen.
getBiayaById(id_biaya)	public	Untuk mengambil data biaya dengan id biaya = id input.

deleteBiaya(id)	<i>public</i>	Menghapus obyek proyek dengan atribut id biaya = id.
editBiaya(id, newNama, newDesc, newEstimasiBiaya)	<i>public</i>	Mengedit atribut obyek dengan id_biaya = id sesuai dengan input pada argumen.
getAllBiayaInProyek(id_proyek)	<i>public</i>	Mengambil data semua obyek biaya yang ada pada <i>database</i> id_proyek = id input.
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
-	-	-

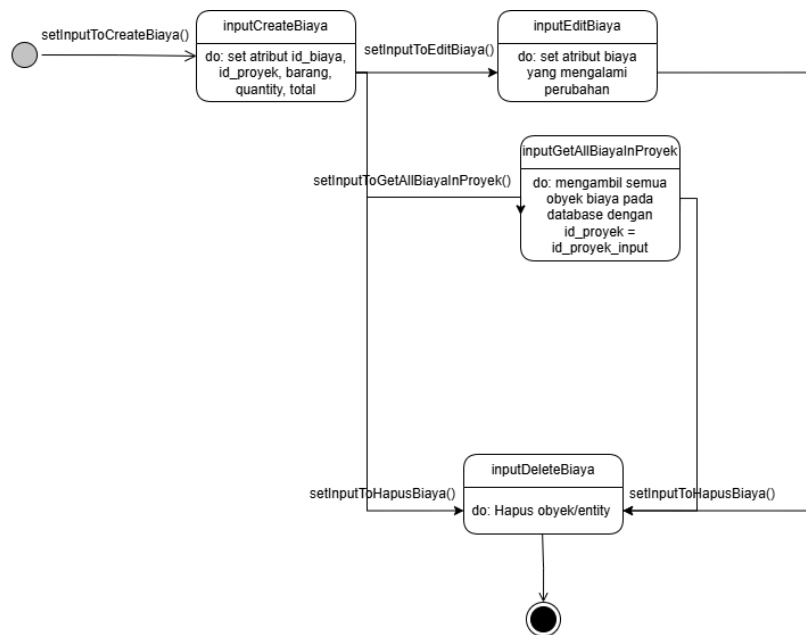
3.2.12.1 Algoritma/Query

Query : PengelolaDataProyek

Tabel 3.2.12.1. Daftar Operasi Query Kelas PengelolaBiaya

<i>No Query</i>	<i>Query</i>	<i>Keterangan</i>
<i>Q-12-01</i>	SELECT * FROM t_biaya WHERE id_proyek = id_proyek_input ORDER BY biaya_id DESC	Mengambil semua data biaya secara descending berdasarkan dengan proyek_id = proyek_id_input
<i>Q-12-02</i>	INSERT INTO t_biaya(id_proyek, barang, quantity, total) VALUES(id_proyek, barang, quantity, total)	Menambahkan proyek baru ke dalam t_proyek berisikan rincian sesuai input pada argumen
<i>Q-12-03</i>	DELETE FROM t_biaya WHERE biaya_id = biaya_id_dihapus	Menghapus proyek dari t_proyek dengan proyek_id = proyek_id_dihapus
<i>Q-12-04</i>	UPDATE t_biaya SET barang = new_barang, quantity= new_quantity, total= new_total WHERE biaya_id = biaya_id_edited	Mengedit atribut pada data proyek dengan proyek_id = proyek_id_edited sesuai dengan masukan yang diberikan
<i>Q-12-05</i>	SELECT proyek FROM t_biaya WHERE biaya_id = biaya_input	Mengambil data proyek dengan id tertentu dimana proyek_id = proyek_input

3.2.12.2 Diagram Statechart



Gambar 25. Diagram State Chart Pengelola Biaya

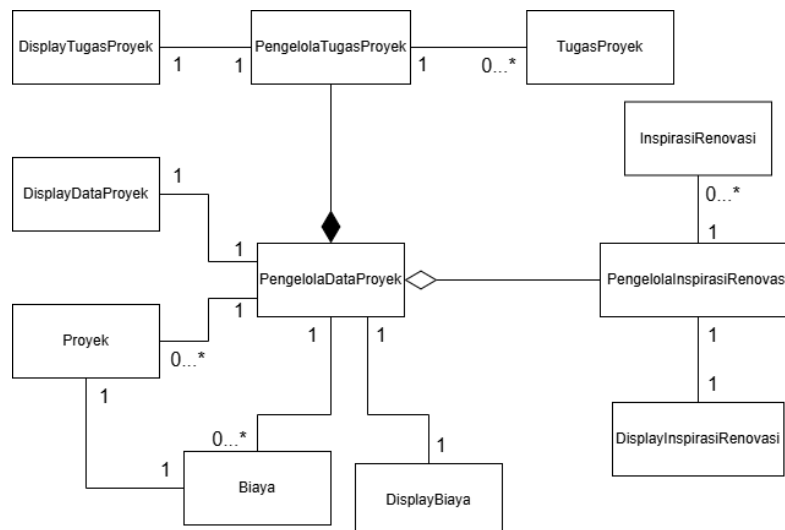
3.2.12.3 Perancangan Antarmuka

Kelas bertipe controller tidak memiliki antarmuka

3.2.12.4 Perancangan Representasi Persistensi Kelas

Kelas bertipe controller sehingga tidak memiliki ERD.

3.3 Diagram Kelas Keseluruhan



Gambar 26. Diagram Kelas Keseluruhan

Tabel 3.3.. Daftar Operasi dan Atribut Kelas Keseluruhan

ID Kelas	Nama Kelas	Atribut	Metode
C-01	Proyek	proyek_id	get_proyek_id(<i>proyek_id</i>)
			set_proyek_id(<i>proyek_id</i>)

		proyek_nama	get_proyek_nama(<i>proyek_id</i>)
			set_proyek_nama(<i>proyek_id</i>)
		proyek_deskripsi	get_proyek_deskripsi(<i>proyek_id</i>)
			set_proyek_deskripsi(<i>proyek_id</i>)
C-02	PengelolaDataProyek		add_pryoeks (<i>proyek_id</i> , <i>proyek_nama</i> , <i>proyek_deskripsi</i> , <i>proyek_tanggal_mulai</i>)
			delete_proyek(<i>proyek_id</i>)
			update_proyek (<i>proyek_id</i> , <i>proyek_nama</i> , <i>proyek_deskripsi</i> , <i>proyek_tanggal_mulai</i> , <i>proyek_progress</i>)
			get_all_proyek(.)
			filter_proyek (<i>proyek_progress</i>)
			sort_proyek (<i>ascending/descending</i>)
C-03	DisplayDataProyek		display_proyek (<i>proyek_id</i>)
C-04	Tugas Proyek	tugas_id	get_tugas_id(<i>tugas_id</i>)
			set_tugas_id(<i>tugas_id</i>)
		tugas_nama	get_tugas_nama(<i>tugas_id</i> , <i>tugas_nama</i>)
			set_tugas_nama(<i>tugas_id</i> , <i>tugas_nama</i>)
		tugas_deskripsi	get_tugas_deskripsi(<i>tugas_id</i> , <i>tugas_deskripsi</i>)
			set_tugas_deskripsi(<i>tugas_id</i> , <i>tugas_deskripsi</i>)
		tugas_status	get_tugas_status(<i>tugas_id</i> , <i>tugas_status</i>)
			set_tugas_status(<i>tugas_id</i> , <i>tugas_status</i>)
		proyek_id	get_proyek_id(<i>proyek_id</i>)
			set_proyek_id(<i>proyek_id</i>)
C-05	PengelolaTugasProyek		add_tugas (<i>tugas_id</i> , <i>tugas_nama</i> , <i>tugas_deskripsi</i> , <i>tugas_status</i>)
			delete_tugas (<i>tugas_id</i> , <i>tugas_nama</i> , <i>tugas_deskripsi</i> , <i>tugas_status</i>)
			update_tugas (<i>tugas_id</i> , <i>tugas_nama</i> , <i>tugas_deskripsi</i> , <i>tugas_status</i>)
			get_all_tugas (<i>tugas_id</i> ,.)
C-06	DisplayTugasProyek		display_tugas (<i>tugas_id</i>)

C-07	Biaya	biaya_Id biaya_Nilai	getBiaya(displayBiaya(<i>proyek_id</i>) readBiaya(displayBiaya(<i>proyek_id</i>) akumulasiBiaya(displayBiaya(<i>proyek_id</i>)
C-08	<i>Display</i> Biaya		displayBiaya(<i>proyek_id</i>)
C-09	InspirasiRenovasi	inspirasi_id	get_inspirasi_id(<i>inspirasi_id</i>) set_inspirasi_id(<i>inspirasi_id</i>)
		nama_inspirasi	get_inspirasi_nama(<i>inspirasi_id</i>)
C-10	PengelolaInspirasiRenovasi		add_inspirasi (<i>inspirasi_id</i> , <i>nama_inspirasi</i> , <i>detail_inspirasi</i>)
			delete_inspirasi (<i>inspirasi_id</i> , <i>nama_inspirasi</i> , <i>detail_inspirasi</i>)
			update_inspirasi (<i>inspirasi_id</i> , <i>nama_inspirasi</i> , <i>detail_inspirasi</i>)
			get_all_inspirasi (<i>inspirasi_id</i>)
C-11	<i>Display</i> InspirasiRenovasi		display_inspirasi (<i>inspirasi_id</i>)

4 Matriks Kerunutan

Tabel 4. Matriks Kerunutan

Kelas	Use Case Terkait
C-01	UC01
C-02	
C-03	
C-04	UC02
C-05	
C-06	
C-01	UC03
C-02	
C-07	
C-08	
C-08	UC04
C-09	
C-11	