

WELCOME TO BOOKLET OF

SPAKBOR HILLS

FARMER GAME



182223005
ANDHIKA M.
ADDIPUTRA



182223029
KEVIN AZRA



182223091
JASON SAMUEL



182223105
ZHEANNETTA
APPLE H.

K04 - KELompok 12





TABLE OF CONTENTS

SPAKBOR HILLS

GAME

1 Table of Contents

6 Our Journey

2 Game Farmer Works

7 Implemented Design Pattern

3 Bridging of User Manual

8 Log Activity

4 Gameplay

5 Class Diagram

SPAKBOR HILLS

GAME



GAME FARMER WORKS

Contribution of our Team!

KELOMPOK 12 KELAS K04



182223005

ANDHIKA M.
ADIPUTRA

005

1. Menggerjakan Code model entity and controller
2. Menggerjakan Class Character beserta turunannya
3. Membuat UI dan mengintegrasikannya
4. Asosiasi Class Diagram
5. Finishing Code



029



1. Melakukan UI Integration
2. Membuat booklet
3. Menggerjakan Player dan mencari asset nya
4. Menggerjakan bagian Menu
5. UI Game keseluruhan
6. Asosiasi Class Diagram
7. Testing



182223029

KEVIN AZRA

091



182223091

JASON SAMUEL

1. Code model entity and controller
2. Class Map dan turunannya
3. Asosiasi Class Diagram



105



1. Melakukan UI Integration
2. Mengisi booklet
3. Menggerjakan Code View & Controller
4. Asosiasi Class Diagram
5. Testing



182223105

ZHEANNETTA
APPLE H.

SPAKBOR HILLS

GAME

BRIDGING OF USER MANUAL

"Spakbor Hills" adalah permainan berbasis **GUI** yang dibuat menggunakan bahasa pemrograman Java. Permainan ini terinspirasi dari game populer "**Stardew Valley**" dan mengisahkan seorang *player* yang dapat melakukan kegiatan bercocok tanam, berinteraksi dengan tetangga, kegiatan jual beli, dan kegiatan lainnya yang dilakukan pada *Farm Map*. Pemain akan memiliki pengalaman bermain dengan **GUI yang interaktif** dan dapat melakukan berbagai kegiatan yang seru dan menakjubkan.

HOW TO PLAY

The Steps: Terminal

Clone our Repository

- Buka terminal atau *Command Prompt*
- Jalankan perintah berikut untuk meng-clone repository:
`git clone https://github.com/AndhikaAddiputra/IF2010_TB_04_12`
- Masuk ke direktori hasil clone:
`cd IF2010_TB_04_12`

Kompilasi dan Menjalankan Program

- Jalankan perintah berikut untuk membangun *project*:
`javac -d bin -cp src src/**/*java`
- Jalankan program dengan perintah berikut:
`java -cp bin App`



SYSTEM

Persyaratan Sistem

- Java Development Kit 8 or newer**
- Terminal** atau **Command Prompt**
- VSCode** (opsional)

Tujuan Permainan

Tujuan dari "Spakbor Hills" adalah melakukan berbagai macam aktivitas yang ingin dilakukan, seperti memasak, menanam, tidur, atau bahkan berinteraksi dengan NPC lain.

OUR GAME MENU



START
MEMULAI PERMAINAN



COMMAND
DAFTAR PERINTAH YANG DAPAT
DIGUNAKAN SELAMA BERMAIN.



ACTIVITY

MENAMPILKAN DAFTAR
AKTIVITAS YANG TERSEDIA
DALAM PERMAINAN
DAN DESKRIPSI NYA.



MENAMPILKAN DAFTAR NPC
YANG TERSEDIA
DALAM PERMAINAN
DAN DESKRIPSI MEREKA.



EXIT

KELUAR DARI PERMAINAN

SPAKBOR HILLS

GAME

USER MANUAL GAMEPLAY!

Inisialisasi Permainan:

Pada aplikasi Spakbor Hills, proses inisialisasi permainan dimulai dari kelas **App** yang menjalankan metode main. Pada metode ini, aplikasi mengatur tampilan UI agar sesuai dengan sistem operasi, lalu menampilkan jendela utama melalui objek **MainMenuWindow**. Inisialisasi lebih lanjut, seperti pembuatan peta permainan (**WorldMap**), Inventory, dan objek-objek lain yang diperlukan, dilakukan pada tahap berikutnya ketika pemain memulai permainan dari menu utama. Dengan demikian, file **App.java** berfungsi sebagai entry point aplikasi yang menyiapkan tampilan awal, sedangkan logika inisialisasi detail permainan diatur di kelas-kelas lain yang terkait dengan gameplay.

WELCOME TO THE GAME



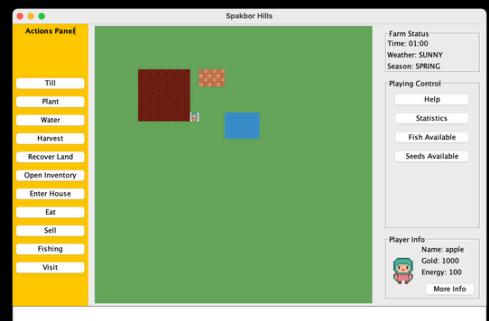
Pemain diminta untuk memilih menu yang tersedia!

**YOU CHOOSE TO
START THE GAME!**

INSERT YOUR PLAYER NAME AND FARM NAME



WELCOME TO YOUR FARM MAP

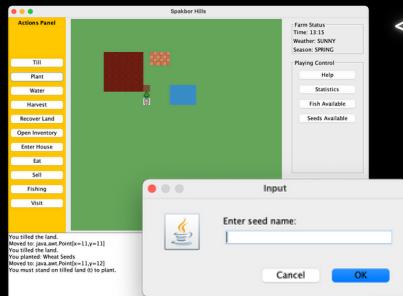


SPAKBOR HILLS

GAME

USER MANUAL GAMEPLAY!

GAME HAS STARTED



<<<TILL, PLANT, WATER>>>

Planting, Watering, dan Harvesting adalah aksi utama dalam bercocok tanam. Setelah mengolah tanah (tilling), pemain dapat menanam seed (planting) di atasnya, menyiram tanaman setiap hari (watering) agar tumbuh, dan memanen hasilnya (harvesting) saat tanaman siap. Ketiga aksi ini saling terhubung: planting memulai siklus tanam, watering mempercepat pertumbuhan, dan harvesting menghasilkan crop yang dapat digunakan atau dijual.

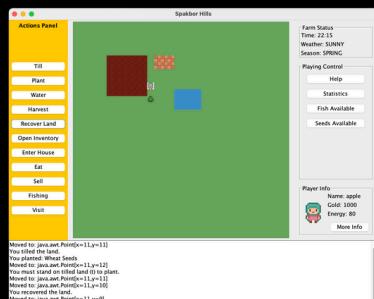
<<<HARVEST>>>

You harvested: 3 unit(s) of Wheat

Aksi Harvest dilakukan ketika player berdiri di area plant yang sudah harvestable/siap di panen. Aksi ini akan mengurangi 5 energi untuk setiap crop yang dipanen dan melakukan skip waktu 5 menit untuk setiap plant yang diharvest. plant yang diharvest akan masuk ke inventory.

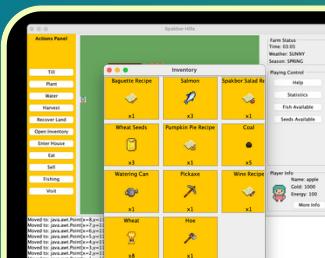
<<<RECOVER LAND>>>

Aksi Recover land akan mengembalikan tanah yang sudah dibajak/tilled kembali menjadi land. Aksi ini mengurangi 5 energi dan melakukan skip waktu 5 menit untuk setiap tile yang direcover.



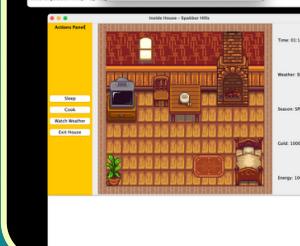
<<<OPEN INVENTORY>>>

Aksi open inventory akan membuka panel inventory dan memperlihatkan isi inventory player.



<<<ENTER HOUSE>>>

Aksi enter house digunakan untuk memasuki rumah pemain. Aksi ini hanya bisa dilakukan ketika pemain berada di luar rumah. Aksi ini akan menampilkan interior rumah dan memungkinkan aksi baru untuk dilakukan, yaitu sleep, cook, watch weather dan exit house.

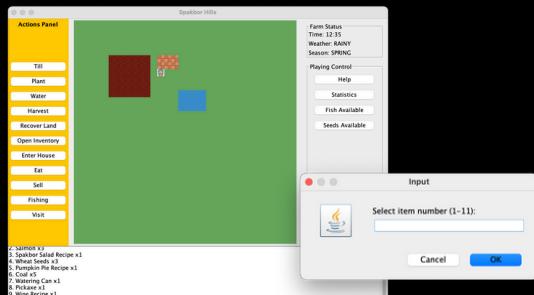
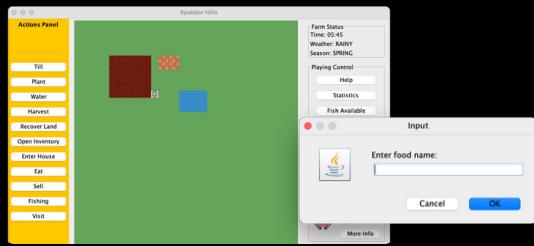


SPAKBOR HILLS

GAME

USER MANUAL GAMEPLAY!

GAME HAS STARTED

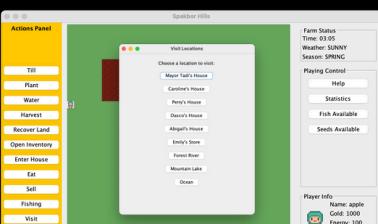


<<< EAT >>>

Aksi Eating digunakan untuk memulihkan energi pemain dengan memakan item edible seperti crop, fish, atau food. Pemain memilih item yang ingin dimakan dari inventory. Setiap aksi makan memberikan energi tambahan sesuai dengan nilai item yang dimakan dan memajukan waktu permainan 5 menit.

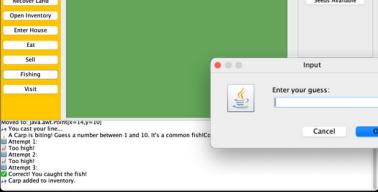
<<< SELL >>>

Aksi Selling dilakukan di dekat shipping bin. Pemain memilih item dari inventory dan memasukkannya ke shipping bin untuk dijual. Setelah aksi selesai, waktu maju 15 menit, dan gold akan langsung ditambahkan ke pemain. Item yang bisa dijual antara lain crop, fish, food, dan equipment.



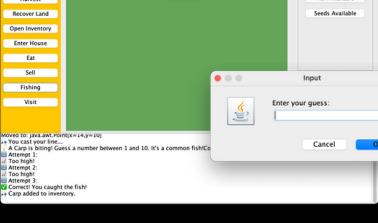
<<< VISIT >>>

Aksi Visit memungkinkan pemain pergi ke lokasi di luar farm, seperti rumah NPC, store, atau area fishing. Pemain harus berdiri di tepi map untuk melakukan visit, lalu memilih lokasi tujuan. Setelah sampai, pemain bisa melakukan interaksi seperti chat, gift, propose, marry di rumah NPC, atau membeli barang di store.



<<< FISHING >>>

Aksi Fishing dilakukan di lokasi fishing spot (seperti pond, river, lake, atau ocean). Pemain menggunakan fishing rod dan akan memainkan mini-game tebak angka dengan RNG untuk menangkap ikan. Jenis ikan yang ditangkap dipengaruhi oleh lokasi, waktu, musim, dan cuaca. Memancing mengurangi energi pemain 5 poin, menambahkan waktu 15 menit, dan jika berhasil, ikan akan ditambahkan ke inventory.

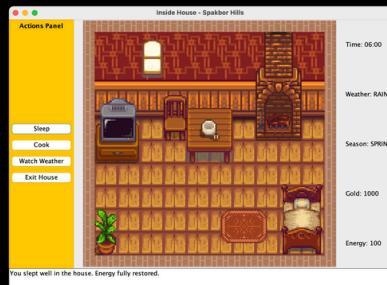


SPAKBOR HILLS

GAME

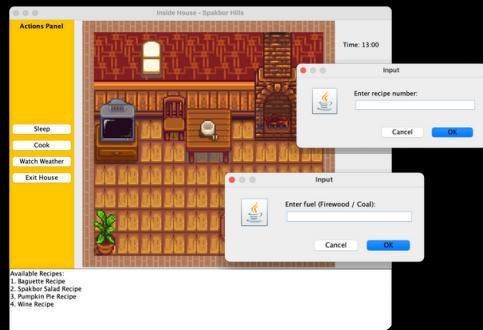
USER MANUAL GAMEPLAY!

ENTER HOUSE



<<< SLEEP >>>

Aksi Sleeping digunakan untuk mengisi ulang energi pemain sepenuhnya. Pemain harus berdiri di dekat house untuk tidur secara optimal, atau jika tidur di tempat lain, pemulihan energi hanya sebagian. Jika pemain tidak tidur hingga lewat pukul 02:00 pagi, pemain akan otomatis tidur di tempat terakhir dengan penalti energi. Aksi tidur memajukan waktu ke pukul 06:00 keesokan harinya.



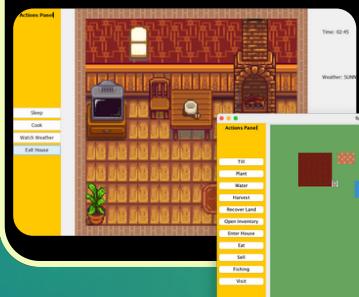
<<< COOK >>>

Aksi Cooking memungkinkan pemain mengolah bahan-bahan dari inventory menjadi makanan siap saji. Pemain harus berada di dekat house, memilih resep yang sudah di-unlock, menyiapkan bahan-bahan dan bahan bakar (firewood/coal), lalu memasak. Proses memasak memakan waktu 60 menit dalam permainan dan bersifat pasif, setelah selesai makanan akan otomatis ditambahkan ke inventory.



<<< WATCH WEATHER >>>

Aksi watch weather digunakan untuk mendapatkan informasi mengenai cuaca hari ini. Cuaca dapat memengaruhi berbagai aktivitas seperti menyiram tanaman, menangkap ikan tertentu.



<<< EXIT HOUSE >>>

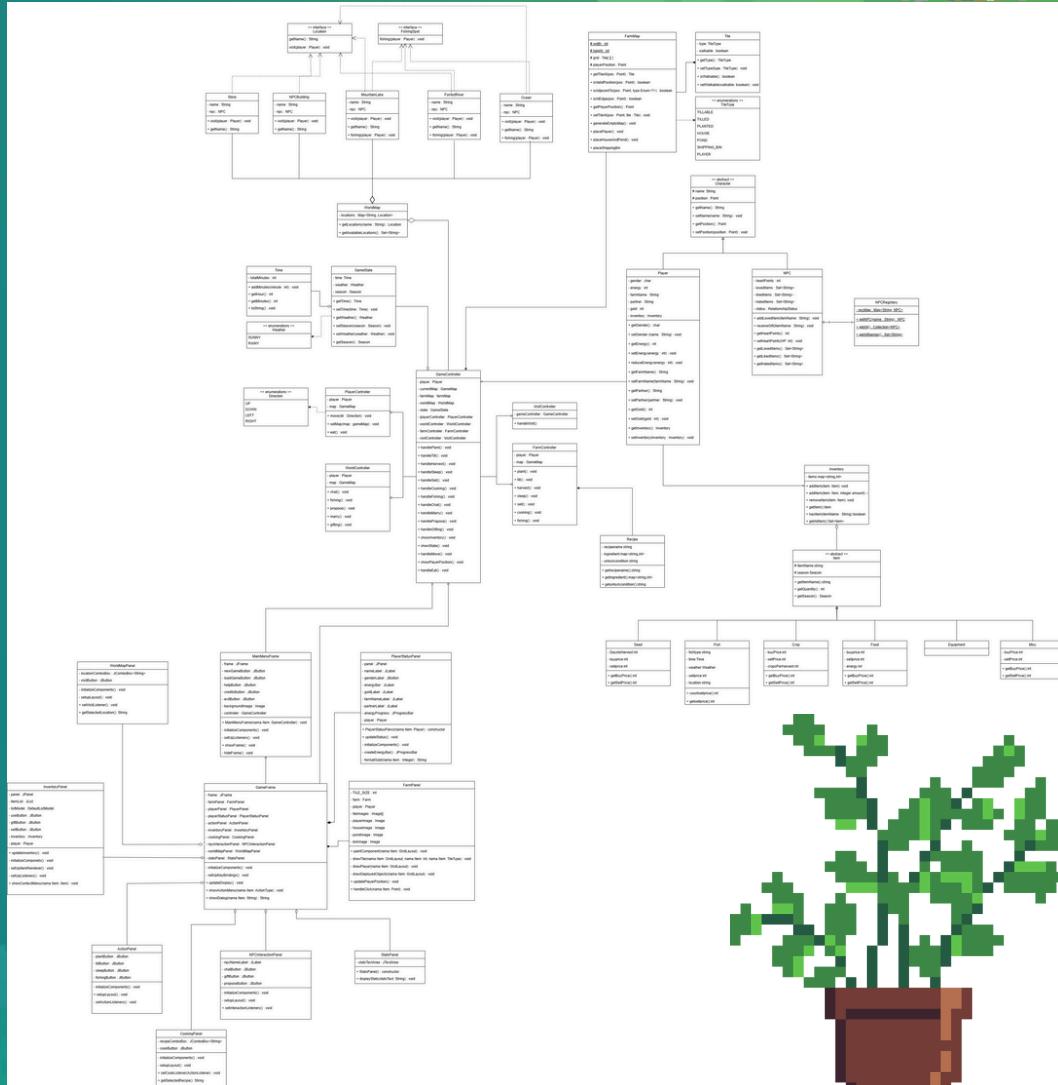
Aksi exit house digunakan untuk keluar dari rumah. Aksi ini hanya bisa dilakukan ketika pemain berada di dalam rumah.



SPAKBOR HILLS

GAME

OUR CLASS DIAGRAM



Download the Image !

[Click here to a better Quality!](#)

SPAKBOR HILLS

WHAT CHANGED?

GAME

FROM MILESTONE 1 TO MILESTONE 2

Banyak perbedaan yang kami lakukan dari rencana awal di **milestone 1** dan realisasi di **milestone 2**. Pada awalnya, implementasi untuk **static file** akan menggunakan **csv atau txt** untuk menyimpan. Namun, setelah dipikir-pikir sepertinya akan mengalami rawan **bug**. Jadi, kami menggunakan **class registry** atau **class factory** untuk **instance static** objeknya.. Hal tersebutlah yang paling banyak kami ubah

Lalu, untuk **UI** kami membuatnya berdasarkan **map**. Jadi, panel-panelnya berdasarkan **map**, bukan lagi berdasarkan **aksi**. Kalau di **class diagram** masih berdasarkan aksi dan panggil di **controller**nya itu juga berubah

Handle inventory tidak langsung memasukkan item dan quantitynya, tapi memanggil satu kelas lagi yaitu **inventory entry**. Kelas tersebut tugasnya mirip seperti **registry** untuk memasukkan, untuk **instance** secara otomatis

MILESTONE 1



Selama pengerjaan **Milestone 1**, kami melakukan diskusi sebanyak **3 kali** (Pada tanggal **21 April** untuk diskusi pertama, **24 April** untuk Asistensi, dan **27 April** untuk Finalisasi Akhir).

Kami melakukan pengerjaan tugas sesuai dengan yang kami bagi berdasarkan **chat** di grup.

Dengan **finalisasi** dilakukan bersama-sama. Kami juga selesai mengerjakan Milestone 1 dengan **BAIK!**

Kendala yang kami alami selama **Milestone 1** mungkin hanya kebingungan saja dalam membuat **class diagram** dan sulitnya menentukan hubungan antar kelasnya.

Berikut dokumentasi asistensi kami saat asistensi 1 pada **tanggal 24 April 2024**:



MILESTONE 2



Jurur pada saat **Milestone 2** tantangan terberat kami adalah untuk mengintegrasikan UI dengan code yang telah kami buat. Kami berusaha untuk mencoba berbagai cara agar dapat mengintegrasikan, dengan cara mencari asset terlebih dahulu dan menonton video tutorial

Namun, pada akhirnya kami berhasil menyelesaikan Tubes ini dengan Baik dan telah berusaha semampu kami

Berkat bantuan teman-teman kelompok, asisten, dan Pak Catur selaku pengajar OOP, kami bisa menyelesaikan Tugas Besar kali ini. Terkait bonus, kami berhasil menuntaskannya sehingga Game yang kami buat, dibuat dengan GUI.

Pengerjaan juga telah dilakukan semaksimal yang kami bisa dan kami harap game ini dapat diimplementasikan dengan baik :D



FACTORY DESIGN PATTERN

Factory Pattern digunakan secara luas dalam game ini, terutama untuk pembuatan berbagai jenis objek item seperti *equipment*, *seeds*, *crop*, *fish*, *food*, *misc item*, dan *recipe*. Implementasi *pattern* ini dapat ditemukan pada *class-class* seperti *EquipmentRegistry*, *SeedsRegistry*, *CropRegistry*, *FishRegistry*, *FoodRegistry*, *MiscRegistry*, dan *RecipeRegistry*. Dengan *Factory Pattern*, proses instansiasi objek-objek tersebut tidak dilakukan secara langsung menggunakan *keyword new*, melainkan melalui *method factory* seperti *getEquipment("Hoe")* atau *getSeeds("Pumpkin Seeds")*. Hal ini memberikan fleksibilitas tinggi, karena penambahan tipe item baru cukup dilakukan di *registry/factory* terkait tanpa perlu mengubah kode di bagian lain. Selain itu, pattern ini juga mendukung prinsip *Open/Closed Principle* dari SOLID, karena logic pembuatan objek terenkapsulasi di dalam *factory*, sehingga kode lain tidak perlu mengetahui detail implementasinya.

EVENT NOTIFICATION DESIGN PATTERN

Observer Pattern (atau *Event Notification Pattern*) diterapkan untuk mengatur komunikasi antar komponen, khususnya antara *controller/model* dan *view*. Penerapan *pattern* ini dapat dilihat pada *interface* seperti *MessageListener* dan *UserInputListener* di *package utility*. Misalnya, *class FarmWindow* mengimplementasikan *MessageListener* agar dapat menerima notifikasi atau pesan dari *controller* atau *model*, sehingga UI dapat menampilkan informasi terbaru kepada *user*. Dengan *Observer Pattern*, komponen pengirim *event* (seperti *controller*) tidak perlu mengetahui detail implementasi penerima *event* (seperti *view*), sehingga tercipta *loose coupling* dan kode menjadi lebih modular serta mudah dikembangkan. Penambahan *listener* baru pun dapat dilakukan tanpa mengubah kode pengirim *event*.

SPAKBOR HILLS

GAME

LOG ACTIVITY

005



18223005

ANDHIKA
MAULANA A.

Kegiatan	Waktu [Tanggal]
Mengikuti diskusi pertama	21 April 2025
Membuat <i>Repository GitHub</i>	21 April 2025
Membuat struktur kelas sementara	21 April 2025
Mengimplementasi kode model	22 - 27 April 2025
Mengikuti, bertanya, dan melakukan notulensi Asistensi 1	24 April 2025
Mengimplementasi kode controller dan view	25 April - 29 Mei 2025
Mengikuti dan bertanya Asistensi 2	27 Mei 2025
Finishing Code	29 - 31 Mei 2025

SPAKBOR HILLS

GAME

LOG ACTIVITY

029



18223029

KEVIN AZRA

Kegiatan	Waktu [Tanggal]
Membuat grup koordinasi	16 April 2025
Mengikuti diskusi pertama	21 April 2025
Membuat Class Diagram View	22 - 27 April 2025
Mengikuti, bertanya, dan melakukan notulensi Asistensi 1	24 April 2025
Mencari Asset dan design	28 April - 29 Mei 2025
Integrasi GUI	26 April 2025
Mengikuti, bertanya, dan melakukan notulensi Asistensi 2	27 Mei 2025
Mendesign, merancang, dan mengisi Buklet	13 - 31 Mei 2025
Testing	30 - 31 Mei 2025

SPAKBOR HILLS

GAME

LOG ACTIVITY

091



18223091

JASON
SAMUEL

Kegiatan	Waktu [Tanggal]
Mengikuti diskusi pertama	21 April 2024
Membuat Class Diagram Model	22 - 27 April 2025
Mengikuti, bertanya, dan melakukan notulensi Asistensi 1	24 April 2025
Membuat beberapa Asset	27 April - 29 Mei 2025
Mengikuti Asistensi 2	27 Mei 2025

SPAKBOR HILLS

GAME

LOG ACTIVITY

105



18223105

ZHEANNETTA
APPLE H.

Kegiatan	Waktu [Tanggal]
Mengikuti diskusi pertama	21 April 2024
Membuat Class Diagram View	22 - 27 April 2025
Mengikuti, bertanya, dan melakukan notulensi Asistensi 1	24 April 2025
Membuat Code Gambaran Kasar Integrasi GUI	25 April - 15 Mei 2025
Mengimplementasikan code pada Controller, Model Store dan Asset	15 - 29 Mei 2025
Mengikuti, bertanya, dan melakukan notulensi Asistensi 2	27 Mei 2025
Mengisi Buklet	30-31 Mei 2025