

SPAKBOR HILLS

Created by:

KOS-603 / OOPTimisa

 Muhammad Farhan
(18223004)

 Alghan Pridanusuta
(18223058)

 Fhatika Adhalisman R.
(18223062)

 Hans Joseph B. W. S
(18223072)

 Joan Melkior Silaen
(18223102)

USER MANUAL

Spakbor Hills adalah game CLI di mana pemain menjadi petani yang menanam tanaman, mengelola inventori, memancing, memasak, dan berinteraksi dengan NPC untuk mencapai milestone seperti mengumpulkan gold dan menikah. Setiap tanaman punya karakteristik unik, dan pemain harus mengelola energi serta waktu dengan baik agar berhasil. Game ini dibuat dengan Java menggunakan konsep OOP, concurrency, exceptions, dan design pattern.



GAME CONTROL

Berikut ini penjelasan tentang cara bermain game. Gunakan keyboard handler W, A, S, D untuk navigasi ketika memilih menu Moving, dan tekan Enter pada setiap ketikannya. Setelah permainan dimulai, masukkan input dalam bentuk angka/kata sesuai perintah. Terdapat aturan sistem musim dan waktu yang memengaruhi alur permainan, serta terdapat tujuan utama permainan.



HOW TO PLAY

1. Download zip dan extract
2. Buka terminal dan ubah directory dengan command “cd <path Folder yang berisikan kode utama program>”
3. Jalankan program dengan mengetik ./gradlew build lalu ketik java -jar build/libs/Tubes-OOP-K02-G03.jar

HAPPY PLAYING !

GAMEPLAY

Pada awal permainan, player diberikan 9 menu utama.

==== SPAKBOR HILLS ====

1. New Game
2. Help
3. View Player Info
4. Statistics
5. Actions (Lanjutkan Permainan)
6. Credits
7. Exit
8. Save Game
9. Load Game

> |

KETERANGAN MENU

- 1: Memulai game baru
- 2: Menampilkan cara bermain
- 3: Melihat info player (setelah New Game)
- 4: Melihat statistik player (setelah New Game)
- 5: Melanjutkan permainan (setelah New Game)
- 6: Melihat kredit
- 7: Keluar dari permainan
- 8: Menyimpan game
- 9: Memuat game yang telah disimpan

Input berupa angka , misalnya player memasukkan angka 2 maka akan muncul menu Help seperti ini

>> 2

==== BANTUAN ===

Selamat datang di Spakbor Hills! Game simulasi pertanian oleh Dr. Asep Spakbor. Tujuan utama: Capai 17.209g atau Menikah dengan salah satu NPC. Gunakan berbagai aksi untuk bertani, memancing, memasak, dan berinteraksi. Setiap aksi membutuhkan energi dan waktu. Atur energimu dengan baik! Musim berganti setiap 10 hari, perhatikan tanaman dan ikan yang tersedia. Cuaca bisa hujan atau cerah, hujan akan menyirami tanamamu secara otomatis. Untuk informasi lebih lanjut, lihat spesifikasi atau coba berbagai aksi!

Untuk memulai permainan masukkan input 1. Setelah itu masukkan nama pemain, gender (Laki-laki/Perempuan), dan nama Farm

>> 1

==== MULAI GAME BARU ===

Masukkan nama pemain: Kelompok 3

Masukkan gender (Laki-laki/Perempuan): Laki-laki

Masukkan nama Farm: ITB

Game baru berhasil dimulai!

Lalu, game akan dimulai dengan menampilkan Waktu, Hari, Musim, Cuaca, Energi, Gold, Peta Farm yang dirandom setiap New Game (beserta legendanya), Lokasi, dan Aksi yang dapat dipilih player.

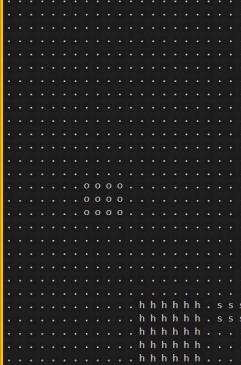
GAMEPLAY

Selamat datang di Spakbor Hills! 🎮

Kamu sudah memasuki dunia pertanian milik Dr. Asep Spakbor.

= APA YANG AKAN KAMU LAKUKAN? =
Waktu: 06:00 | Hari: 1 | Musim: SPRING | Cuaca: SUNNY
Energi: 100/100
Gold: 0.0g

--- PETA FARM ---



Legenda: p (Player), h (House), o (Pond), s (Shipping Bin)
. (Fillable), t (Filled), l (Planted)
Lokasi: Farm (15,29)

Pilih Aksi:

1. Moving	11. Proposing
2. Tilling	12. Marry
3. Planting	13. Watching TV
4. Watering	14. Visiting
5. Harvesting	15. Chatting
6. Recover Land	16. Gifting
7. Eating	17. Open Inventory
8. Sleeping	18. Show Time
9. Cooking	19. Show Location
10. Fishing	20. Selling (to Shipping Bin)
21. Exit to Main Menu	

TUJUAN UTAMA PERMAINAN

- ⌚ Mencapai gold sebanyak 11.209g
- 💍 Menikahi salah satu NPC yang tersedia

SISTEM WAKTU & MUSIM

- ⌚ 1 detik nyata = 3 menit game
- 🕒 Hari dimulai jam 06.00 dan malam dimulai jam 18.00
- 🕒 Terdapat 4 musim {Spring, Summer, Fall, Winter} dan 2 jenis cuaca {Sunny & Rainy}
- 🌧 Saat Rainy, kamu tidak perlu menyiram tanaman

KETERANGAN AKSI

No.	Aksi	Keterangan
1	Moving	Bergerak ke arah atas/bawah/kiri/kanan dalam Farm Map
2	Tilling	Menabur lahan ('.') menjadi lahan siap tanam ('t') menggunakan Hoe
3	Planting	Menanam benih di lahan ('t' menjadi 'l') yang sudah dibajak
4	Watering	Mengairi tanaman agar bisa tumbuh
5	Harvesting	Memanen tanaman matang untuk dijual atau dimakan
6	Recover Land	Mengembalikan tanah yang sudah dibajak atau ditanami menjadi tanah kosong
7	Eating	Mengonsumsi makanan dari inventory untuk memulihkan energi
8	Sleeping	Tidur untuk memulihkan energi dan melanjutkan ke hari berikutnya
9	Cooking	Memasak resep di dalam rumah jika bahan dan bahan bakar tersedia
10	Fishing	Memancing ikan sesuai lokasi, waktu, musim, dan cuaca
11	Proposing	Menikahi NPC jika heart point mencukupi dan cincin ikarannya tersedia
12	Marry	Menikahi dengan NPC jika ikarannya telah diterima
13	Watching TV	Menonton TV di rumah untuk mendapatkan tips atau informasi cuaca/musim
14	Visiting	Mengunjungi rumah NPC atau lokasi di luar Farm (misal: Store, Ocean)
15	Chatting	Mengobrol dengan NPC untuk meningkatkan heart point
16	Gifting	Memberikan hadiah ke NPC, respon tergantung jenis item
17	Open Inventory	Melihat isi inventory dan jumlah item yang dimiliki
18	Show Time	Menampilkan waktu, hari, musim, dan cuaca saat ini
19	Show Location	Menampilkan lokasi pemain dalam dunia dan koordinat di map Farm
20	Selling	Menjual hasil panen ke Shipping Bin untuk ditukarkan menjadi Gold
21	Exit to Main Menu	Keluar dari permainan ke menu utama

END GAME STATISTICS

Jika kamu mencapai salah satu milestone {Gold ≥ 11.209g atau sudah menikah}, game akan menampilkan statistik

- 💰 Total income & expenditure
- 📊 Rata-rata per musim
- 💬 Status dengan NPC
- 🎣 Crops & Fish yang sudah kamu panen/tangkap

PEMBAGIAN TUGAS



**Muhammad
Farhan
(13223004)**

- Membuat repository GitHub
- Membuat class diagram bagian Items, Maps, dan Menu Game
- Membuat README
- Mensetting gradle
- Membuat code Farm, FarmMap, WorldMap, House, Tile, Bonus (Let's Go Gambling & Keyboard/Mouse Input), Main, Game
- Memperbaiki bug dan error
- Melakukan testing



**Alghan
Priidanusuta
(13223053)**

- Membuat class diagram bagian Farm, House, dan Action
- Membuat code Action (Action, Cooking, Eating, Fishing, Harvesting, Planting, Proposing, RecoverLand, Sleeping, Tiling, Watering), House
- Memperbaiki bug dan error
- Melakukan testing



**Fhatika
Adhalisman R.
(13223062)**

- Membuat class diagram bagian Time, Season, Weather, dan Shipping Bin
- Membuat code Time, Season, Weather, Shipping Bin, dan GameTime.
- Membuat buklet
- Memperbaiki bug dan error
- Melakukan testing

PEMBAGIAN TUGAS

- Membuat class diagram bagian Inventory dan Endgame
- Membuat code Action (Chatting, Gifting, Marry, Moving, OpenInventory, Selling, ShowLocation, ShowTime, Visiting, Watching), Inventory, Main, Game
- Memperbaiki bug dan error
- Melakukan testing



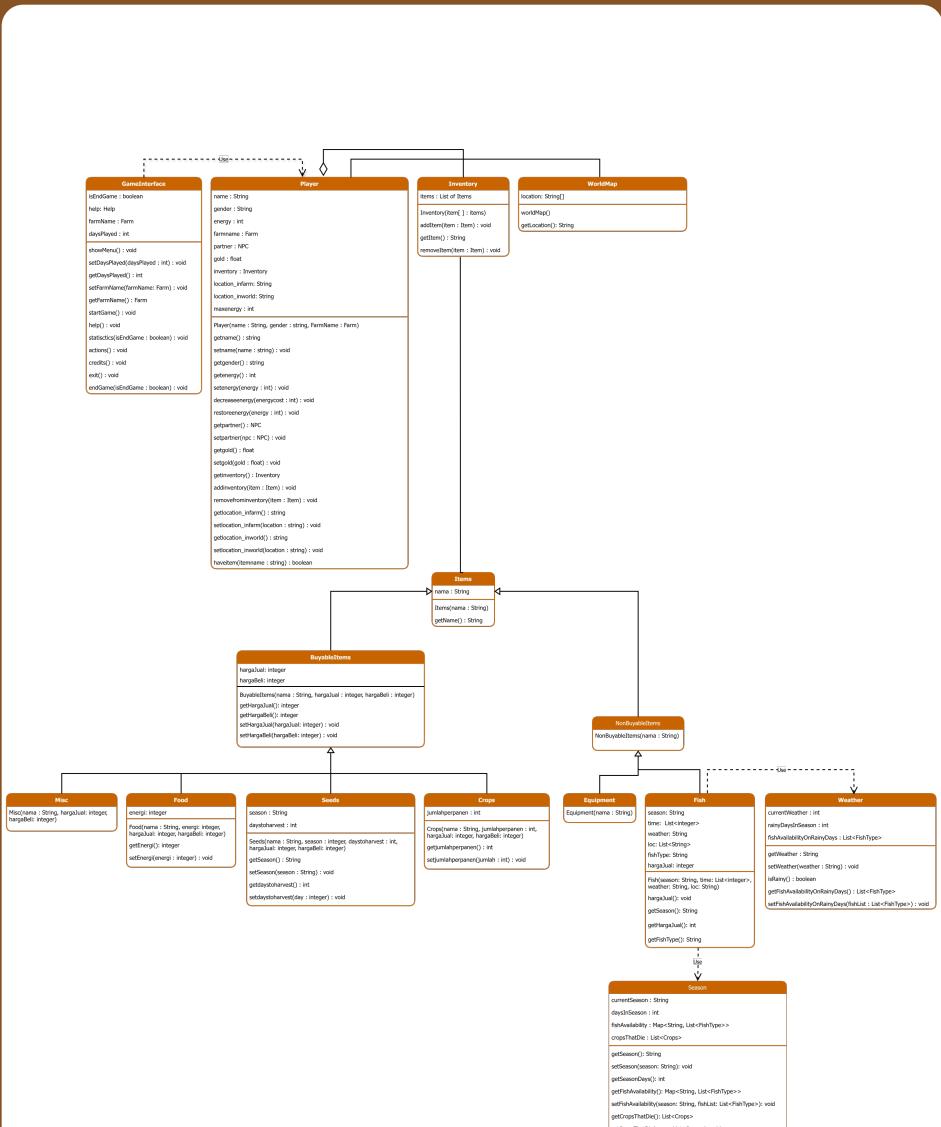
Hans Joseph
B. W. S.
(18223072)



**Joan Melkior
Silaen**
(18223102)

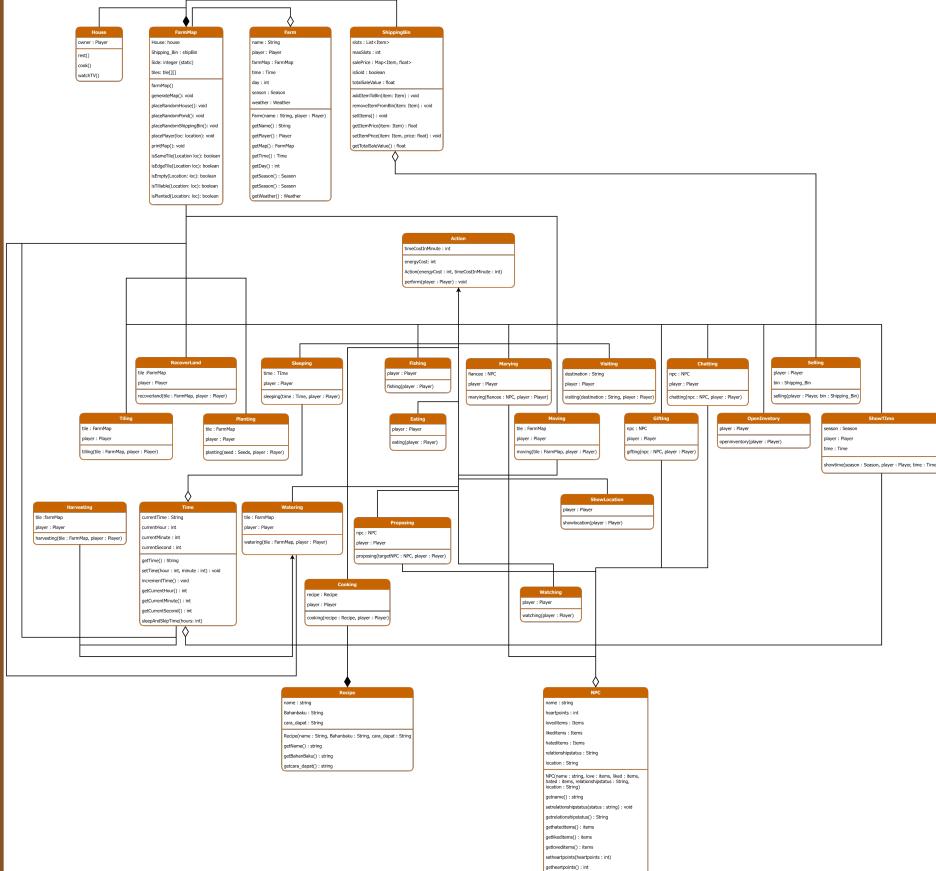
- Membuat class diagram bagian Player, NPG, dan Items
- Membuat code Action, Items (Buyable, Drops, Equipment, Fish, FishDatabase, Food, Items, Misc, NonbuyableItems, Recipe, Seeds), Pond, Player, NPG, Bonus (Save & Load), Main, Game
- Memperbaiki bug dan error
- Melakukan testing

CLASS DIAGRAM AWAL



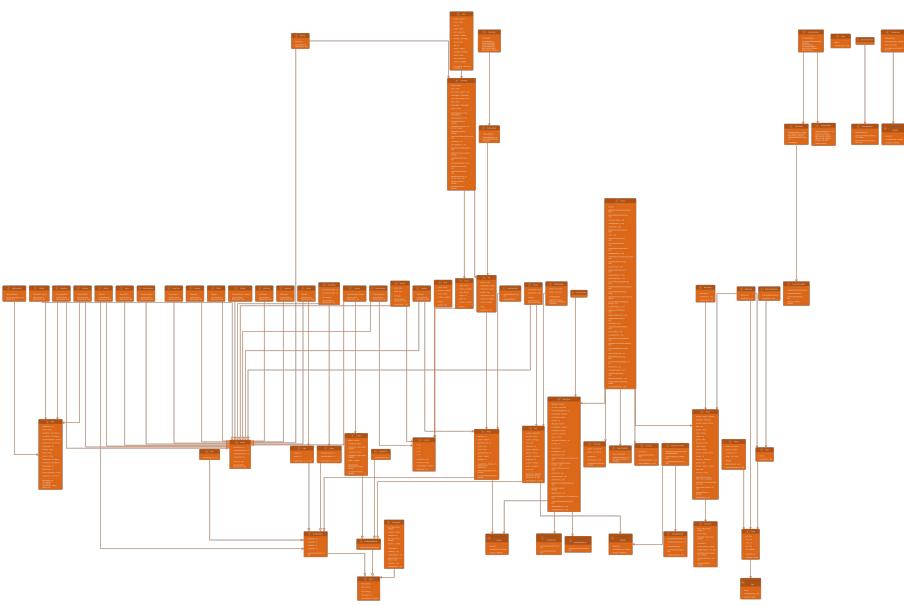
[bit.ly/ClassPage.com/eval-kos-sns](https://classpage.com/eval-kos-sns)

CLASS DIAGRAM AWAL



bit.ly/ClassDiagramJavaWithKos-gcs

CLASS DIAGRAM AKHIR



Region / Sistem	Bantangan Read	Implementasi Metrik	Pembuktian	Aksi Teknis / Fungsional
Pengendali Utama	-	Stone (loop OLE, menu, int, endgame, atska)	Dikembangkan	Operasikan setiap controller kerusakan untuk mengeksekusi file permission dan mengontrol objek-objek di game.
Time System	GameTime (non-Horizon- dotted)	GameTime * TimeController; SecondController; MechanicsController; TimeMeu	HWI-Speech	Pemisahan model dari logika update & komponen → memudahkan unit-test dan concurrence (tambah thread).
Store / Economy	Toko-koko	Store (seed/feel seeds; recipe, mice)	Dikembangkan	Logika harga, stok & update resep tanaman dengan teknologi AI. Batasi pricng dengan teknologi AI.
Data Base	Faktur-satuan	Faktur-satuan (filter, bg session, sekuhan, lokasi, time)	Dikembangkan	Perbaikan permission & pemeliharaan data DB (tanpa kerusakan menutup stok Fishing).
Action Hierarchi	Superclasses + 4-5 contoh oksi	1 superkelas + 21 subkelas (Moving, Tilting...; Matching, Visting)	Dikembangkan & dimulai	Penghindari pular-bean; memudahkan pricng Open-Based (tambah oksi = tambah hasil).
Player	1. Player- (move, ponder, energy, Form, Inventory)	+ statistik agent, AIbehavior, displayPlayer, checkRightSeat, (red) blood, guard, recess , helper, (houseless, coldwinter, etc.)	Pembuktian strukturn & method	Pembuktian requirement makelone (Y189 g3 muncul) + end-game statistic.
Inventory	Map(item, qtg), metode dester	Metode removeItem, addItemInventory, removeItem	Dikembangkan	Atasi Cooking, Planting, Browsing Kembalikan validitas S metoda inventori ganti item.
Resep	Resep ingredient, result	+ statik (calories), calculate(m), getIngredient(m)	Dikembangkan	Kembalikan statistik resep untuk respi respi (tambahkan metoda resep).
World Map	Type enum / list objek di dunia	WorldMap (GameWorld) * The-Entity-and- entity / planted, add toPlayer, remove , set edge, preCheckPlayer set edgeToWorld	Dikembangkan	Atasi WorldMap perlu list dunia S. validasi input世紀。
Runway & The	Excavator * atribut atribut	PlayMap (GameWorld) * The-Entity-and- entity / planted, add toPlayer, remove , set edge, preCheckPlayer set edgeToWorld	Pembuktian strukturn & detail	Perbaikan dari Flying, Planting, Watering, Harvesting, Resove Lent serta gerak player drimp.
NPC	name, HealthPoint, level/level/hp/ status	+ counter visit, shot, gift: get toPlayer getLeaveFrom...	Pembuktian data	Operasikan untuk statistik dan efek akibat Shooting, Gifting, Watering, Harvesting, Hunting.
ShippingBin	Komposit di Excavator	+ static %, sellIn(glober)	Dikembangkan method	Pembuktian horion & perhitungan gold hasil tanpa pada Inventory.
Denonverifikasi	Galon tangki	Thread pool! GameTime.updateTask() & Thread pool! Cooking (1 min Imagine or sleep 10 sec)	Dikembangkan	Pembuktian ketepatan teknik "timbangan" S spesifikasi Cooking pool.
Resep Item (Food)	Unsur-alkali/Rasa dan	Volibel (play) di Cooking (Firewood)	Logika khusus	Logikal Cooking resulabilitas konsumsi Firewood.
Matematik Olah	Universonal di Bentuk/Bentuk	Implementasi detail	Atasi anomali data responing permission & trigger-reaction ekstrim	

BONUS YANG DIKERJAKAN:

- Keyboard/Mouse Input
- Save & Load
- Let's Go Gambling

bit.ly/ClassDiagramAkhir-KOS-GOS

DESIGN PATTERN

Observer Pattern adalah behavioral pattern yang memungkinkan satu objek (Subject) memberi tahu sekumpulan objek lain (Observer) apabila terjadi perubahan keadaan tanpa membuat ketergantungan ketat (tight-coupling).

```
TimeObserver.java ×  
Tubes.yey > app > src > GameCalendar > Observer > J TimeObserver.java > Language Support for Java(TM) by Red Hat > {} Game  
1 package GameCalendar.Observer;  
2  
3 import GameCalendar.Model.GameTime;  
4  
5 public interface TimeObserver {  
6     /**  
7      * Dipanggil setiap kali ada update waktu yang signifikan dari GameTime.  
8      * @param currentTime instance GameTime saat ini.  
9      */  
10    void onTimeUpdate(GameTime currentTime);  
11  
12    /**  
13     * Dipanggil secara spesifik ketika waktu game mencapai pukul 02:00.  
14     * @param currentTime instance GameTime saat ini (yang seharusnya menunjukkan 02:00).  
15     */  
16    void onTwoAM(GameTime currentTime);  
17 }
```

Permainan Spakbor Hills memakai Observer Pattern dengan menjadikan GameTime sebagai subject yang hanya mengelola hitungan waktu, sementara kelas-kelas yang membutuhkan info waktu (mis. Game) mendaftar lewat interface TimeObserver.

```
Game.java ×  
Tubes.yey > app > src > main > java > J Game.java > Language Support for Java(TM) by Red Hat > {} Game > S handleAutoSleepOrTermination  
67 public class Game implements TimeObserver{  
68  
69     @Override  
70     public synchronized void onTimeUpdate(GameTime currentTime) {  
71         // Jika (player != null && player.getEnergy() <= 20 && isRunning) {  
72             // System.out.println("ALERT OBSERVER: Energi habis karena waktu berjalan pasif!");  
73         }  
74  
75         @Override  
76     public synchronized void onTwoAM(GameTime currentTime) {  
77         // Metode ini dipanggil secara spesifik saat jam 2 AM.  
78         if (!isRunning) return; // Jika game sudah dalam proses berhenti, jangan lakukan apa-apa.  
79  
80         System.out.println("ALERT OBSERVER: Sudah jam 2 pagi! Waktunya tidur atau game berakhir.");  
81         if (player != null && actionMap != null && actionMap.containsKey("key:sleeping")) {  
82             handleAutoSleepOrTermination();  
83         } else {  
84             System.out.println("Game akan dihentikan karena sudah jam 2 pagi dan tidak bisa tidur otomatis.");  
85             this.isRunning = false; // Fallback jika tidak bisa tidur  
86         }  
87     }  
88  
89     private void handleAutoSleepOrTermination() {  
90         if (!this.isRunning) return; // Sudah dalam proses terminasi  
91  
92         // Cek apakah player masih ada (belum null karena kondisi lain)  
93         if (player == null) {  
94             System.out.println("Pemain tidak ditemukan, game akan dihentikan.");  
95             this.isRunning = false;  
96             return;  
97         }  
98  
99         if (player.getEnergy() <= 20) {  
100             System.out.println("Energi habis! Kamu pinggang dan otomatis tidur.");  
101         } else {  
102             System.out.println("Waktu sudah menunjukkan pukul 02:00; Kamu otomatis tidur.");  
103         }  
104     }  
105  
106     Action sleepingAction = actionMap.get("key:sleeping");  
107  
108     if (sleepingAction != null) {  
109         sleepingAction.perform(player, gameTime); // Jadi Tidur akan memanggil gameTime.timeSkip()  
110         updateEndGameConditions(forceShowStats:false); // Cek kondisi akhir game  
111         checkEndGameConditions(forceShowStats:false); // Cek kondisi akhir game  
112         // Game loop akan berlanjut ke hari berikutnya jika isRunning masih true.  
113     } else {  
114         System.out.println("ERROR: Aksi 'sleeping' tidak ditemukan! Game tidak bisa lanjut.");  
115         this.isRunning = false; // Hentikan game jika tidak bisa tidur  
116     }  
117 }
```

Tepat pukul 02.00, GameTime menyiarkan notifikasi. Game menanggapi sinyal itu untuk memaksa pemain tidur atau menghentikan permainan tanpa perlu logika waktu tersimpan di kelas jam. Pola ini menjaga kode tetap longgar terikat, mudah diperluas, dan mematuhi prinsip SOLID karena logika waktu dan aturan gameplay dipisahkan rapi.

PROSES PENGEMBANGAN

MEET 1

24 April 2023



- Menetapkan kelas apa saja yang akan dibuat
- Membagi tugas ke setiap anggota
- Membuat class diagram

MEET 2

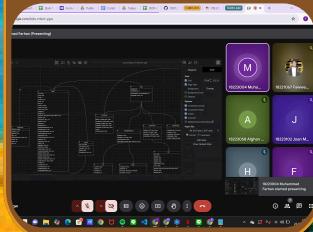
29 April 2023



- Membuat class diagram
- Menetapkan hubungan antar kelas

ASISTENSI 1

29 April 2023



- Melaporkan progress Milestone 1
- Bertanya dan meminta saran mengenai class diagram yang telah dibuat

MEET 3

30 Mei 2023



- Membuat code sesuai kelas yang sudah dibagi
- Mempersiapkan lingkungan build otomatis menggunakan Gradle

MEET 4

32 Mei 2023



- Membuat code sesuai kelas yang sudah dibagi ke masing-masing anggota

MEET 5

26 Mei 2023



- Membuat main untuk menguji program
- Memperbaiki bug dan error

PROSES PENGEMBANGAN

MEET 6

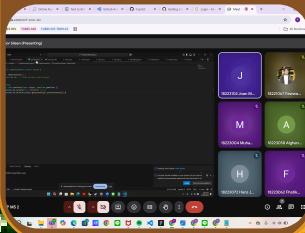
28 Mei 2023



- Memperbaiki bug dan error
- Menambahkan spesifikasi yang belum diimplementasikan

ASISTENSI 2

28 Mei 2023



- Melaporkan progres Milestone 2
- Menguji Atur utama program
- Bertanya dan meminta saran mengenai masalah yang dialami

MEET 7

30 Mei 2023



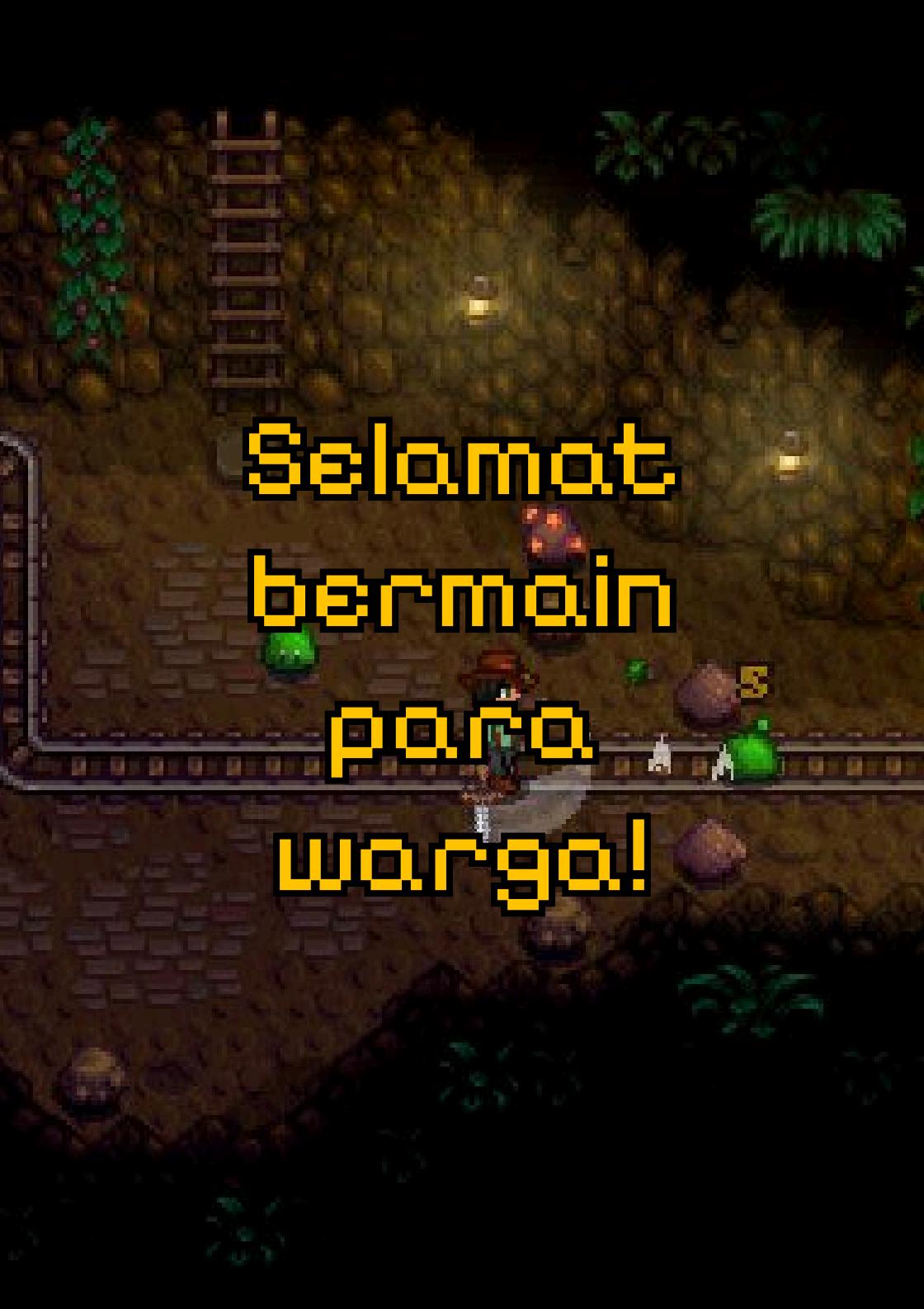
- Melakukan testing semua Atur yang ada
- Memperbaiki bug dan error

MEET 8

31 Mei 2023



- Melakukan testing semua Atur yang ada
- Memperbaiki bug dan error
- Finalisasi program
- Merilis program ke GitHub



**Selamat
bermain
para
warga!**