

SiMPLICITY





Table Of Content



User Manual

Bagian ini berisi penjelasan serta deskripsi detail mengenai jalannya permainan



List Aksi Tambahan

Bagian ini berisi keterangan lebih lanjut mengenai aksi tambahan yang sudah dibuat



Class Diagram

Bagian ini berisi penjelasan struktur final dari objek yang terdapat pada game



Implementasi Konsep OOP

Bagian ini berisi pemaparan source code yang sesuai dengan spesifikasi konsep OOP



Implementasi Bonus

Bagian ini berisi penjelasan implementasi bonus yang sudah diterapkan pada game Simplicity



Proses Pengembangan

Bagian ini berisi proses pengembangan selama pembuatan tugas besar

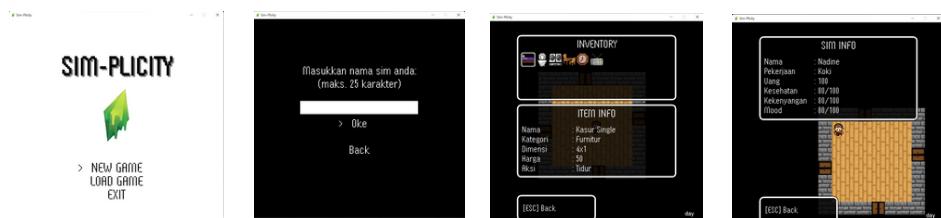


Pembagian Tugas

Bagian ini berisi pemaparan tugas yang sudah dilakukan oleh tiap anggota kelompok

Awal Permainan

1. Pilih opsi "New Game" untuk memulai permainan
2. Masukkan nama sims untuk memulai permainan
3. Sims memiliki info kekenyangan, mood, dan kesehatan yang di generate pada angka 80 dan uang pada angka 100
4. Game akan men-generate rumah dengan 1 rumah dan beberapa objek basic
5. Game akan menyimpan catatan hari dan waktu yang sudah berjalan



Help Center

1. Untuk mempermudah user dalam mengakses permainan, tekan "esc" pada keyboard
2. Pilih opsi help untuk melihat semua command yang dapat dilakukan pada game



Melakukan Pekerjaan

1. Untuk melakukan aksi kerja, tekan tombol "k" pada keyboard lalu pilihlah pekerjaan dan lama pekerjaan yang ingin dilakukan
2. Ketik durasi (dalam detik) lama pekerjaan yang ingin dilakukan
3. Tunggu hingga pekerjaan selesai dilakukan oleh sims



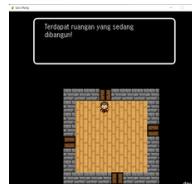
Aksi Aktif

1. Mendekatlah ke salah satu furnitur yang ada di dalam rumah
2. Klik tombol "enter" pada keyboard
3. Masukkan durasi aksi yang ingin dilakukan atau klik "esc" jika sims tidak jadi melakukan aksi
4. Tunggu sampai aksi selesai



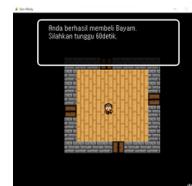
Aksi Upgrade

1. Pastikan sims ada di dalam rumah
2. Pencet tombol "U" pada keyboard
3. Pilih lokasi ruangan yang ingin diupgrade
4. Masukkan nama ruangan yang ingin ditambah
5. Tunggu 18 menit hingga ruangan selesai diupgrade



Membeli Barang

1. Pencet tombol "B" pada keyboard
2. Klik opsi buy
3. Pilih barang yang ingin dibeli, lalu tunggu beberapa saat sampai waktu pengiriman habis dan barang muncul di inventory

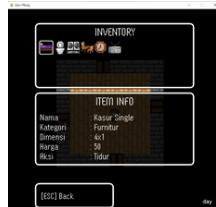


Melihat inventory

User Manual

1. Klik tombol "I" pada keyboard

2. Isi inventory akan bertambah jika sims membeli barang/bahan makanan



Aksi Menambah Sims

1. Pencet tombol "N" pada keyboard

2. Ketik nama sims baru serta koordinat rumah sims baru

3. Kalian dapat melihat list sims yang ada dengan menekan tombol "G" pada keyboard

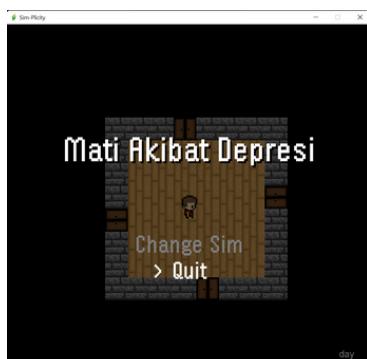
4. Kalian juga dapat berkunjung ke rumah sims lain



Game Over

1. Apabila mood, kekenyangan, dan kesehatan sims mencapai angka 0, sims akan mati

2. Saat sims mati, game akan berakhir





Aksi Tambahan



Bermain Game

- Mood +5/30 s
- Kesehatan -5/30 s
- Kenyangan -5/30 s



Menonton Televisi

- Mood +10/60 s
- Kesehatan -5/60 s
- Kenyangan -5/60 s



Menyiram Tanaman

- Mood +1/10 s
- Kesehatan +1/10 s
- Kenyangan -1/10 s



Shalat

- Mood +5/5 s
- Kenyangan +5/5 s
- Kesehatan +5/5 s



Mendengar Radio

- Mood +1/10 s
- Kenyangan -1/30 s



Mandi

- Mood +10/30 s
- Kesehatan +10/30 s
- Kenyangan -5/30 s



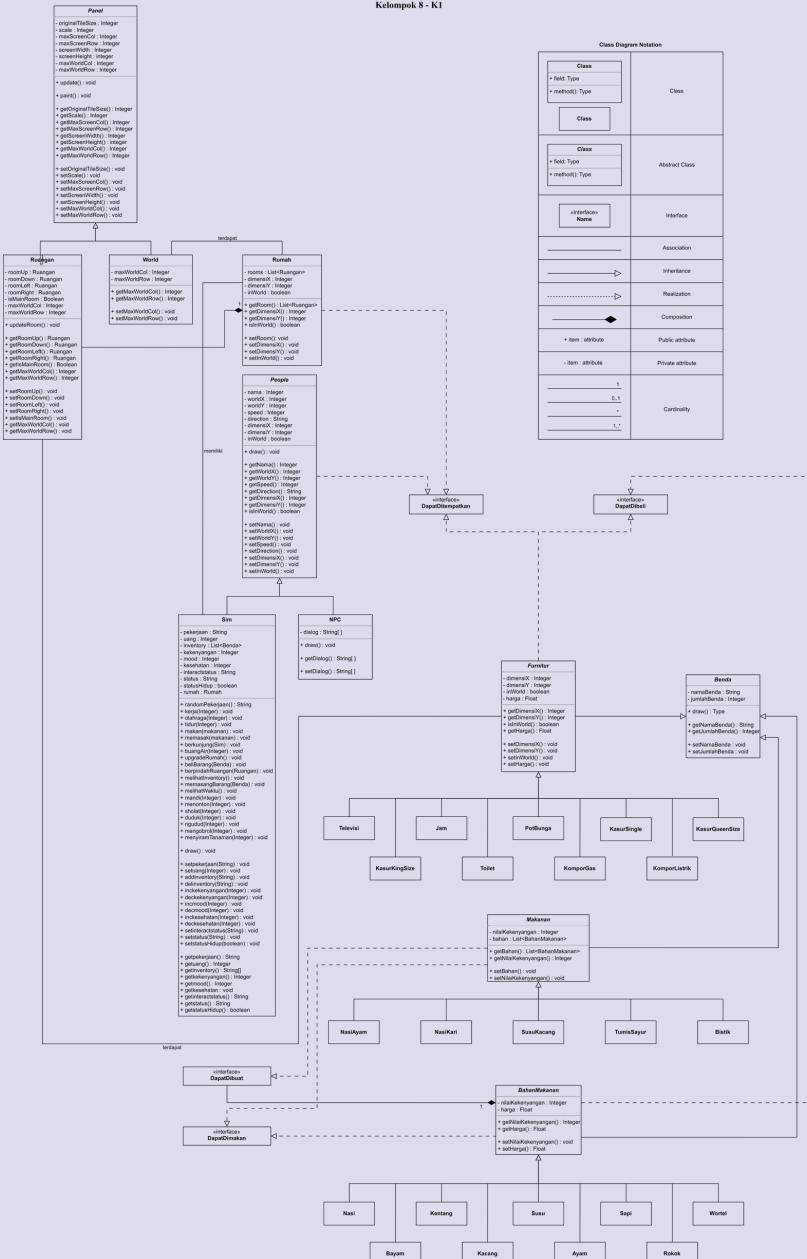
Membaca Buku

- Mood +3/10 s
- Kenyangan -3/10 s

Class Diagram Awal

Class Diagram

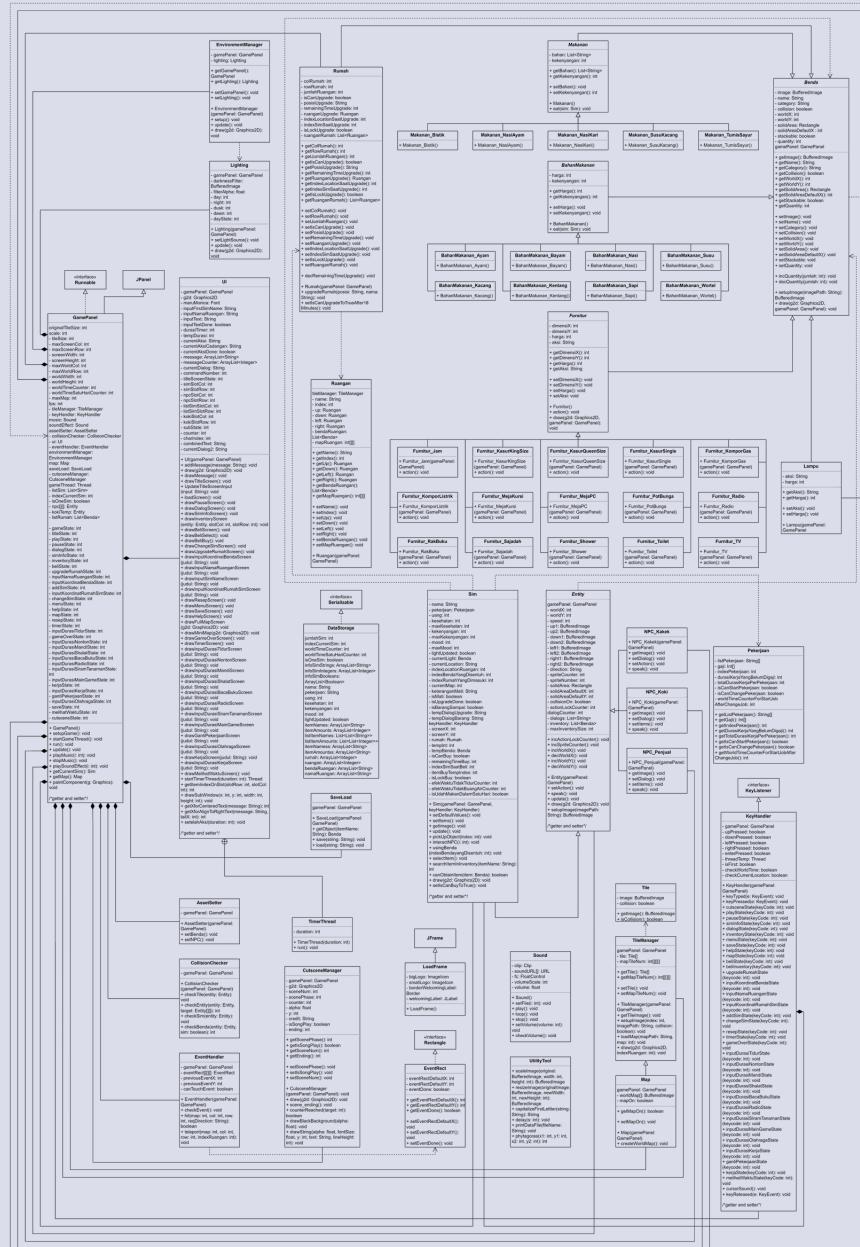
Kelompok 8, KI



Class Diagram Final

Class Diagram

Kelompok 8



Penjelasan Class Diagram



Pada rancangan awal, terdapat beberapa interface diantaranya DapatDibuat, DapatDimakan, DapatDitempatkan, dan DapatDibeli. Interface tersebut rencananya akan diimplementasikan oleh beberapa class terkait. Namun pada implementasi akhir, kami tidak menggunakan interface tersebut dalam kode program.

Selain itu, pada awalnya method terkait aksi-aksi yang dilakukan diimplementasikan pada Class Sim. Namun, pada implementasi akhir, method terkait aksi yang dapat dilakukan Sim diimplementasikan pada setiap Benda(Furnitur). Pada class Furnitur terdapat method action yang akan di-override oleh keseluruhan furnitur yang ada. Perubahan tersebut dilakukan karena Sim hanya dapat melakukan aksi ketika berinteraksi dengan benda terkait.



Kemudian, pada proses membuat rancangan awal, belum terdapat gambaran terkait implementasi GUI yang akan digunakan. Sehingga pada rancangan akhir terdapat perubahan menyesuaikan dengan implementasi GUI yang dibuat. Implementasi GUI menyebabkan penambahan beberapa class, method, dan attribute. Selain GUI, pada rancangan awal belum terdapat gambaran terkait save dan load pada permainan. Sementara itu, pada implementasinya digunakan save dan load sehingga dapat menyimpan data atau progress pada permainan.

Pada implementasi program juga ditambahkan beberapa Benda(Furnitur) menyesuaikan dengan penambahan aksi yang dapat dilakukan oleh Sim. Selain itu, kami juga menambahkan beberapa fitur lain yang tidak termasuk pada spesifikasi seperti penambahan music, map, dan beberapa NPC serta beberapa bagian lainnya pada permainan. Terakhir perubahan dilakukan menyesuaikan dengan konsep OOP pada perancangan aplikasi diantaranya Inheritance, Abstract Class, Polymorphism, Generics, Exceptions, Concurrency, dan penerapan Design Pattern.



IMPLEMENTASI KONSEP OOP

INHERITANCE

```
public class BahanMakanan_Ayam extends BahanMakanan{  
    public BahanMakanan_Ayam(){  
        name = "Ayam";  
        image = setupImage("benda/bahan_makanan/ayam");  
        harga = 10;  
        kekenyangan = 8;  
    }  
}
```

```
public abstract class Furnitur extends Benda{  
    public int dimensiX;  
    public int dimensiY;  
    public int harga;  
    public String aksesoris;  
    public Furnitur(){  
        category = "Furnitur";  
        stackable = true;  
    }  
}
```

ABSTRACT CLASS

POLYMORPHISM

```
public void setAction() {  
}  
}
```

Entity.java

```
@Override  
public void setAction(){  
    actionLockCounter++;  
  
    if (actionLockCounter > 128){  
        actionLockCounter = 0;  
        Random random = new Random();  
        int i = random.nextInt(100)+1;  
  
        if (i <= 25){  
            direction = "up";  
        } else if (i <= 50){  
            direction = "down";  
        } else if (i <= 75){  
            direction = "left";  
        } else {  
            direction = "right";  
        }  
    }  
}
```

NPC_Kakek.java

```
public void startGameThread() { // start the game  
    thread = new Thread(this);  
    gameThread.start();  
}  
  
@Override  
public void run() { // the  
    // start timer thread  
    public Thread startTimerThread(int duration) {  
        TimerThread timer = new TimerThread(duration);  
        Thread thread = new Thread(timer);  
        thread.start();  
        return thread;  
    }  
  
    // inner class for timer thread  
    public class TimerThread implements Runnable {  
        ...  
    }  
}
```

CONCURRENCY

GENERIC

```
public List<Benda>[] getListRumah() {  
    return listRumah;  
}  
  
public void setListRumah(List<Benda>[] listRumah) {  
    this.listRumah = listRumah;  
}
```

```
public UI(GamePanel gamePanel) {  
    this.gamePanel = gamePanel;  
  
    try {  
        maruMonica = Font.createFont(Font.TRUETYPE_FONT,  
            new FileInputStream(new File("data/fonts/x12y16pxMaruMonica.ttf")));  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

EXCEPTION

IMPLEMENTASI BONUS

● Graphical User Interface



● Fitur Save and Load

```
public void load(String string) {
    try {
        ObjectInputStream ois = new ObjectInputStream(new FileInputStream(new File(string)));
        DataStorage ds = (DataStorage) ois.readObject();
        // Load Data
        gamePanel.setWorldTime(ds.getWorldTime());
        gamePanel.setSim(ds.getSim());
        gamePanel.setWorldTimeCounter(ds.worldTimeCounter);
        gamePanel.setWorldTimeSatuhar(counter.ds.worldTimeSatuhar(counter));
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

```
public void saveString(string) {
    try {
        ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(new File(string)));
        DataStorage ds = new DataStorage();
        // Save data
        ds.jumlahSim = gamePanel.getListSim().size();
        ds.dsNeSim = gamePanel.getListOneSim();
        ds.worldTimeCounter = gamePanel.getWorldTimeCounter();
        ds.worldTimeSatuhar(counter.ds.worldTimeSatuhar(counter));
        gamePanel.getWorldTimeSatuhar(counter);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```



● Design Pattern

- Design Pattern: State Pattern
- State pattern digunakan untuk membagi game menjadi 32 state.
- State Pattern kami pilih karena cocok dalam pembuatan game yang cukup kompleks ini. Terutama untuk membedakan Key dan aksi yang dapat dilakukan di masing-masing state

```
private final int titleState = 0;
private final int playState = 1;
private final int pauseState = 2;
private final int dialogState = 3;
private final int simInfoState = 4;
private final int inventoryState = 5;
private final int buyState = 6;
private final int upState = 7;
private final int downState = 8;
private final int leftState = 9;
private final int rightState = 10;
private final int infoState = 11;
private final int exitState = 12;
private final int helpState = 13;
private final int settingsState = 14;
private final int aboutState = 15;
private final int creditsState = 16;
private final int exitToTitleState = 17;
private final int exitToPlayState = 18;
private final int exitToPauseState = 19;
private final int exitDialogTitleState = 20;
private final int exitToSimInfoState = 21;
private final int exitToInventoryState = 22;
private final int exitToBuyState = 23;
private final int exitToUpState = 24;
private final int exitToLeftState = 25;
private final int exitToRightState = 26;
private final int exitToDownState = 27;
private final int exitToHelpState = 28;
private final int exitToSettingsState = 29;
private final int exitToAboutState = 30;
private final int exitToCreditsState = 31;
private final int exitToExitState = 32;
```

```
private final int inputDurasiStramTanamanState = 25;
private final int inputDurasiMainGameState = 26;
private final int kerjaState = 27;
private final int inputDurasiKerjaState = 28;
private final int gantiPekerjaanState = 29;
private final int inputDurasiolahragaState = 30;
private final int tuState = 31;
private final int ateState = 32;
private final int exitState = 33;
```



PROSES PENGEMBANGAN



Meet Diskusi Ke-1

12 April 2023

Membuat class diagram dan menentukan penanggung jawab sementara.
Hasil diskusi juga berupa spreadsheet dan list pertanyaan untuk asistensi.



15 April 2023

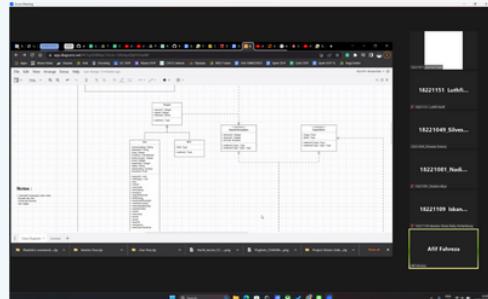
Asistensi Ke-1



Pada asistensi pertama ini, kami menanyakan banyak hal mengenai class diagram dan juga isi spesifikasi yang kami rasa masih membingungkan.

15 April 2023

Asistensi Ke-1



Meet Diskusi Ke-2

17 April 2023

Setelah asistensi, kami lanjut diskusi. Kami merevisi class diagram sesuai dengan masukan asistensi dan membagi tugas sesuai PJ sekaligus rencana mendatang

24 April 2023

Meet Diskusi Ke-3



Diskusi progress dan membuat rencana berikut-berikutnya. Pembagian tugas dan juga revisi jika tidak sesuai spesifikasi



Asistensi Ke-2

28 April 2023

Demo progress yang sudah ada sekaligus meminta pendapat Kak Afif apa saja yang harus diperbaiki. Kemudian dilanjutkan pembagian tugas hingga akhir.

Masukkan Proses Pengembangan



PROSES PENGEMBANGAN



Meet Diskusi Ke-4 28 April 2023

EDIT ROOM	amriz	DL : MINGGU PAGI
GAME OVER	amriz	
BURUNG TIME	amriz	
BUAH AIR	amriz	
KERJAS	amriz	
ULAM RASA	amriz	
NONTON	hr	SIL
NYIRAM TANAMAN	pot_bunga	LUTHFI
SHOP	kasir	NADINE
NYETEL RADIO	radio	KANDAR
MANDI	shower	LUTHFI
BACA BUKU	rak_buku	NADINE
MAIN GAME	map_pt	SIL

SAVE IN LOAD	KANDAR
+ TAMBAHIN EFEK KE AKSI YANG UDAH JADI	

Kerjakan AMRIZ	
PICK UP OBJECT	• buah
GAMBAR BURUNG	• rotan
BUAH AIR	• berukung
KERJAS	• buah
IMPLEMENT MIKAN DI MEJA MAKAN	• buah buahan
	• buah buang
	• buah tidak tur
	• buah tidak buang air

Hasil diskusi berupa pembagian tugas pada spreadsheet

1 Mei 2023

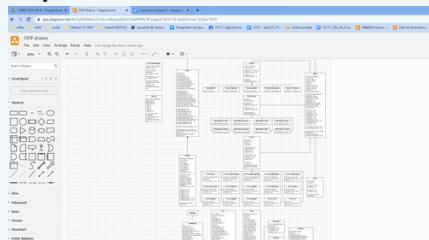
Asistensi Ke-3

Melakukan revisi pada cara kerja world time, saat itu penggunaan waktu pasive masih kurang sesuai. Berikutnya kami mendiskusikan kapan akan menyelesaikan dan merapikan code.



Finalisasi dan Merapikan Code

4-5 Mei 2023



Kami melakukan finalisasi dan mengubah atribut yang masih public menjadi private, sekaligus merapikan source code.

5 Mei 2023

Asistensi Ke-4 dan Diskusi

Kami melakukan asistensi terakhir kami untuk finalisasi dan bertanya mengenai buklet dan requirement lainnya. Kemudian kami diskusi untuk membahas progress melakukan testing terakhir sebelum dikumpulkan.



Masukkan Proses Pengembangan





Pembagian Tugas



Silvester Kresna - 18221049

PJ Buklet, Membuat Class Diagram milestone 1, Membuat aksi taruh barang versi awal, Membuat class Furnitur PC dan pot, Membuat aksi Bermain game dan Menyiram tanaman, mendesain asset furnitur dan asset yang dibutuhkan, Melakukan revisi atribut seluruh class pada folder main selain GamePanel serta class lain yang terkait, mengisi dan mendesign laporan buklet



Ahmad Rizki - 18221071

PJ Source Code, Membuat Class Diagram, Membuat kerangka awal GUI, Memegang kendali branch main GitHub, Membuat aksi tidur, olahraga, masak, makan, melihat waktu, upgrade rumah, naruh barang, pindah ruangan/rumah, kerja, dan ganti kerja, beli barang, melihat inventory, menambah/mengganti sim, Membuat fitur Game Over, NPC, dst.



Nadine Aliya - 18221081

PJ Buklet, Membuat Class Diagram milestone 1, Membuat entity pekerjaan, Membuat class Furnitur Rak Buku dan Sajadah, Membuat aksi Shalat dan membaca buku, Mengedit sound yang dipakai pada game, Melakukan revisi atribut pada folder object , folder map, dan file entity beserta class yang terkait, Mengisi dan mendesign laporan buklet



Iskandar Muda - 18221109

PJ Source Code, Membuat Class Diagram milestone 1, Membuat options menu dan help, Menambahkan sound effect dan musik ke dalam game, Membuat class Furnitur Radio, Membuat aksi menyetel radio, Membuat minimap dan fullmap, Membuat save and load, Mencari bug dan melakukan revisi terhadap bug



Luthfi Hanif - 18221151

PJ Class Diagram, Membuat Class Diagram milestone 1, Menjadwalkan asistensi, Membuat class Furnitur TV dan Shower, Membuat aksi nonton dan mandi, Melakukan revisi atribut pada class SIM, Pekerjaan, dan GamePanel serta class lain yang terkait, Menambahkan beberapa sound, Membuat class diagram final dan penjelasan terkait perubahan yang dilakukan