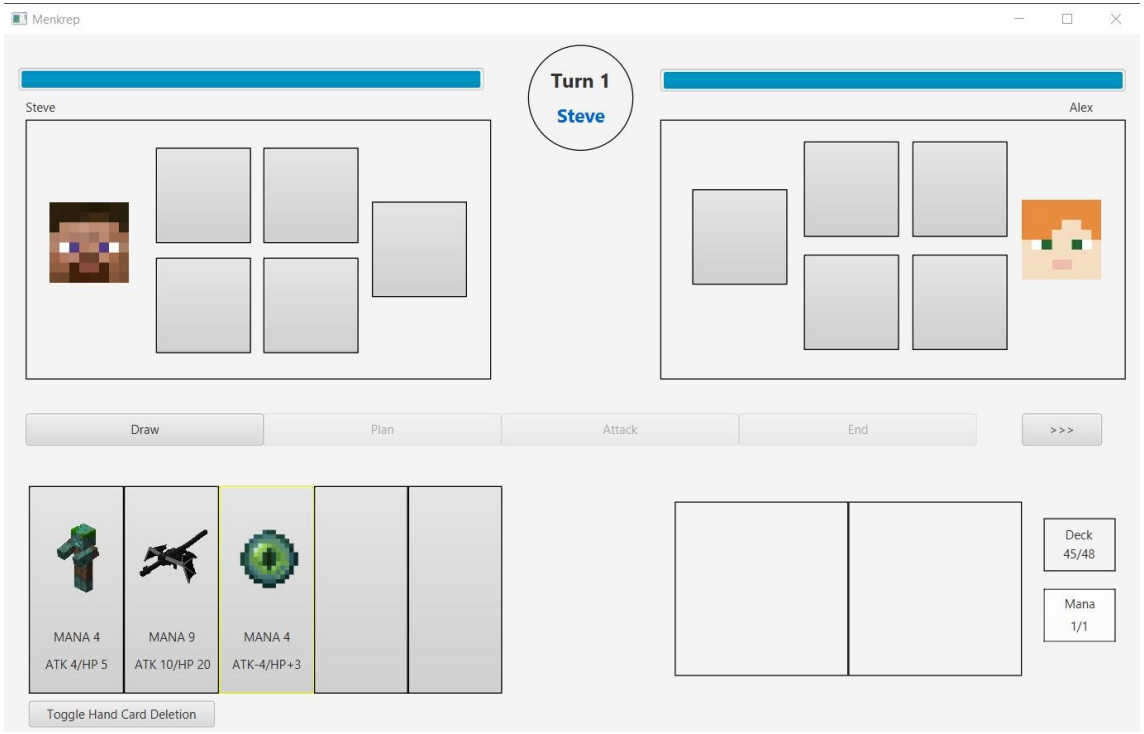
Kelas : 03
Kelompok : 02
Nama Kelompok : Menkrep

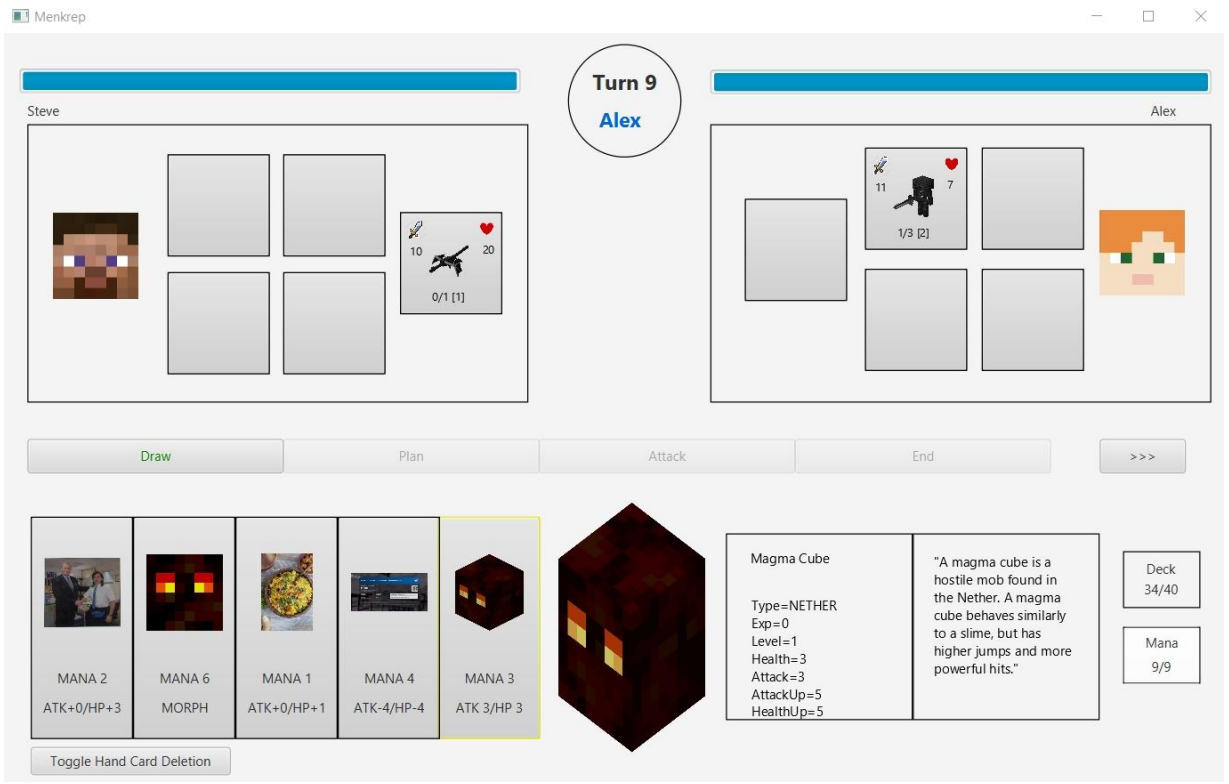
1. 13520057 / Marcellus Michael Herman K
2. 13520066 / Putri Nurhaliza
3. 13520084 / Adelline Kania Setiyawan
4. 13520087 / Dimas Shidqi Parikesit
5. 13520111 / Rizky Akbar Asmaran
6. 13520153 / Vito Ghifari

Asisten Pembimbing : 13518035 / Matthew Kevin Amadeus

1. Deskripsi Umum Aplikasi

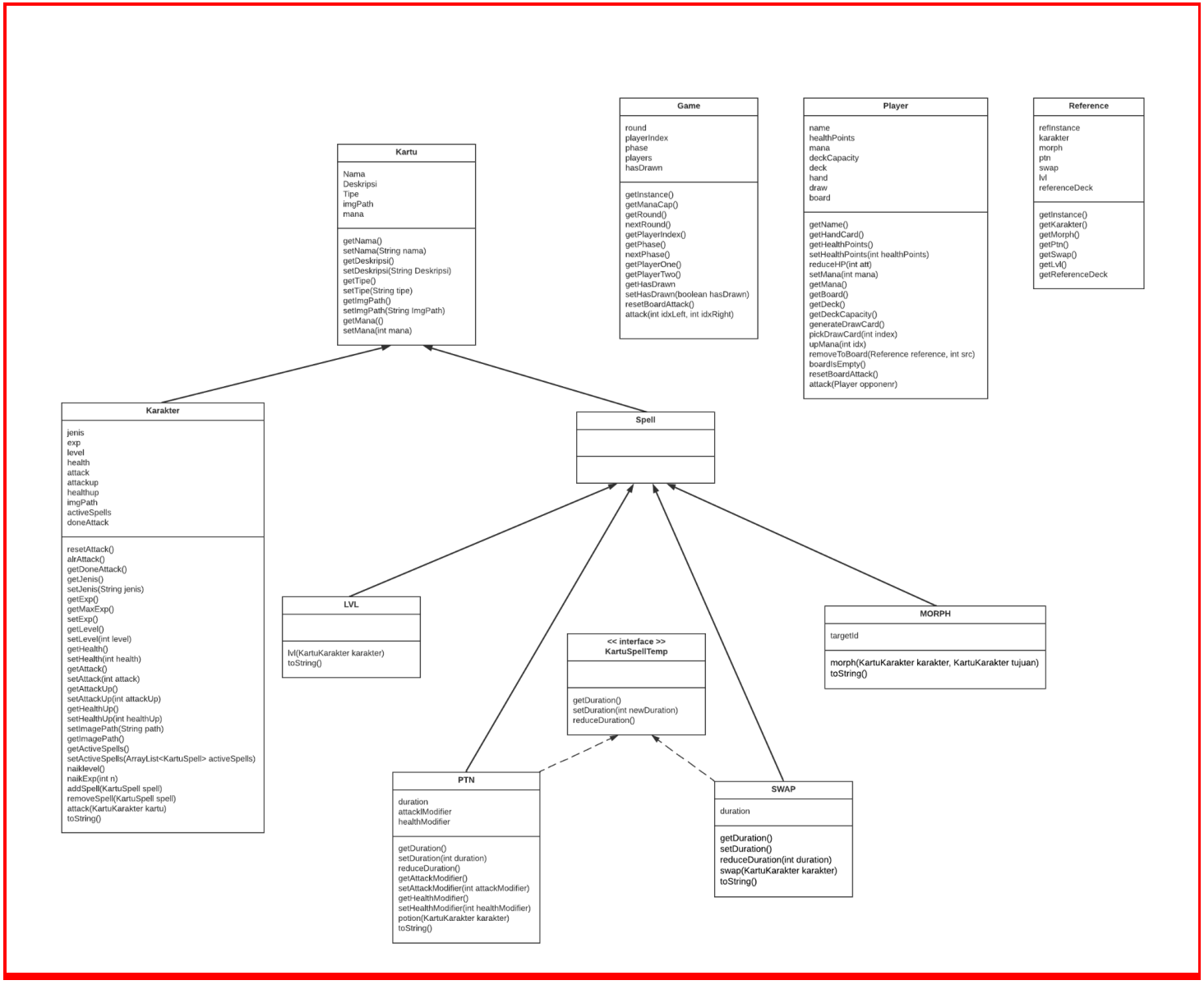


Pada Tugas Besar 2 kali ini, kelompok kami membuat aplikasi game yang bernama Minecraft: Aether Realm Wars. Game ini merupakan sebuah game kartu *turn based* untuk dua orang pemain saja. Kedua pemain akan bermain secara bergantian pada satu layar yang sama. Game ini bertujuan untuk menghabiskan HP dari pemain lawan dengan melakukan serangan dari kartu karakter yang diletakkan di board permainan. Pada game ini memiliki dua jenis kartu, yaitu kartu spell dan kartu karakter. Kartu spell dan kartu karakter memiliki tipenya masing masing. Pada kartu spell memiliki 2 sifat kartu, Permanen atau Sementara, dan 4 tipe, Potion (sementara), Level (permanen), Swap (sementara), dan Morph (permanen). Sedangkan, Pada kartu karakter memiliki 3 tipe, yaitu Overworld, Nether, atau End. Tipe kartu karakter akan mempengaruhi damage yang diberikan maupun yang diterima oleh karakter.



Pada Game ini pemain akan diberikan kumpulan kartu (Deck) yang berisi 40 sampai 60 kartu. Pada Game ini, pemain hanya dapat memiliki kartu hand sejumlah maksimal 5 kartu, apabila lebih maka pemain harus membuang kartu hingga hanya 5 kartu di hand. Mana pada game ini akan bertambah jumlahnya di setiap gilirannya. Pemain akan menerima Exp hanya dengan dua cara. Pertama, membunuh musuh. Karakter akan mendapatkan exp sesuai dengan level musuh. Kedua, menggunakan mana. Karakter akan mendapatkan exp sejumlah mana yang digunakan. Exp yang didapatkan karakter akan dapat dimanfaatkan untuk meningkatkan level karakter yang dapat meningkatkan health dan attack karakter. Pada Game ini, setiap giliran pemain akan dibagi menjadi 4 phase, yaitu phase draw, planning phase, attack phase, dan end phase. Pemain akan dinyatakan menang apabila HP lawan mencapai 0 atau kurang, atau pada Draw phase lawan, deck lawan kosong.

2. Konsep OOP


Kartu
Nama
Deskripsi
Tipe
imgPath
mana
getNama()
setNama(String nama)
getDeskripsi()
setDeskripsi(String Deskripsi)
getTipe()
setTipe(String tipe)
getImgPath()
setImgPath(String ImgPath)
getMana(()
setMana(int mana)
Game
round
playerIndex
phase
players
hasDrawn
getInstance()
getManaCap()
getRound()
nextRound()
getPlayerIndex()
getPhase()
nextPhase()
getPlayerOne()
getPlayerTwo()
getHasDrawn
setHasDrawn(boolean hasDrawn)
resetBoardAttack()
attack(int idxLeft, int idxRight)
Player
name
healthPoints
mana
deckCapacity
deck
hand
draw
board
getName()
getHandCard()
getHealthPoints()
setHealthPoints(int healthPoints)
reduceHP(int att)
setMana(int mana)
getMana()
getBoard()
getDeck()
getDeckCapacity()
generateDrawCard()
pickDrawCard(int index)
upMana(int idx)
removeToBoard(Reference reference, int src)
boardIsEmpty()
resetBoardAttack()
attack(Player opponenr)
Reference
refInstance
karakter
morph
ptn
swap
lvl
referenceDeck
getInstance()
getKarakter()
getMorph()
getPtn()
getSwap()
getLvl()
getReferenceDeck
Karakter
jenis
exp
level
health
attack
attackup
healthup
imgPath
activeSpells
doneAttack
resetAttack()
alrAttack()
getDoneAttack()
getJenis()
setJenis(String jenis)
getExp()
getMaxExp()
setExp()
getLevel()
setLevel(int level)
getHealth()
setHealth(int health)
getAttack()
setAttack(int attack)
getAttackUp()
setAttackUp(int attackUp)
getHealthUp()
setHealthUp(int healthUp)
setImagePath(String path)
getImagePath()
getActiveSpells()
setActiveSpells(ArrayList<KartuSpell> activeSpells)
naikdevel()
naikExp(int n)
addSpell(KartuSpell spell)
removeSpell(KartuSpell spell)
attack(KartuKarakter kartu)
toString()
Spell
LVL
lvl(KartuKarakter karakter)
toString()
<< interface >>
KartuSpellTemp
getDuration()
setDuration(int newDuration)
reduceDuration()
MORPH
targetId
morph(KartuKarakter karakter, KartuKarakter tujuan)
toString()
PTN
duration
attackModifier
healthModifier
getDuration()
setDuration(int duration)
reduceDuration()
getAttackModifier()
setAttackModifier(int attackModifier)
getHealthModifier()
setHealthModifier(int healthModifier)
potion(KartuKarakter karakter)
toString()
SWAP
duration
getDuration()
setDuration()
reduceDuration(int duration)
swap(KartuKarakter karakter)
toString()

2.1. Inheritance

Pada Program, kami menerapkan konsep inheritance untuk mengimplementasikan Kartu dengan membuat kelas Kartu sebagai parentnya. Lalu dari kelas kartu dibuat 2 kelas turunan, yaitu KartuKarakter dan KartuSpell. Selanjutnya, pada KartuSpell juga terdapat 4 kelas turunan, yaitu KartuSpellLvl, KartuSpellMorph, KartuSpellPotion dan KartuSpellSwap. Keuntungan dari dibuatnya kelas turunan adalah mempermudah membuat objek kartu berdasarkan jenisnya, yaitu kartu karakter atau kartu spell, memanfaatkan penggunaan kode kelas yang terdapat pada parent, meningkatkan readability kode dan memudahkan perubahan pada jenis kartu yang mungkin terdapat atribut atau method yang unik. Berikut merupakan daftar dari kelas yang menggunakan konsep inheritance.

- Kelas KartuKarakter (app/src/main/java/Menkrep/Model/Kartu/KartuKarakter.java) merupakan anak dari kelas Kartu (app/src/main/java/Menkrep/Model/Kartu/Kartu.java)
- Kelas KartuSpell (app/src/main/java/Menkrep/Model/Kartu/KartuSpell.java) merupakan anak dari kelas Kartu (app/src/main/java/Menkrep/Model/Kartu/Kartu.java)
- Kelas KartuSpellLvl (app/src/main/java/Menkrep/Model/Kartu/KartuSpellLvl.java) merupakan anak dari kelas KartuSpell (app/src/main/java/Menkrep/Model/Kartu/KartuSpell.java), Kelas KartuSpell merupakan anak dari kelas Kartu (app/src/main/java/Menkrep/Model/Kartu/Kartu.java)
- Kelas KartuSpellPotion (app/src/main/java/Menkrep/Model/Kartu/KartuSpellPotion.java) merupakan anak dari kelas KartuSpell (app/src/main/java/Menkrep/Model/Kartu/KartuSpell.java), Kelas KartuSpell merupakan anak dari kelas Kartu (app/src/main/java/Menkrep/Model/Kartu/Kartu.java)
- Kelas KartuSpellSwap (app/src/main/java/Menkrep/Model/Kartu/KartuSpellSwap.java) merupakan anak dari kelas KartuSpell (app/src/main/java/Menkrep/Model/Kartu/KartuSpell.java), Kelas KartuSpell merupakan anak dari kelas Kartu (app/src/main/java/Menkrep/Model/Kartu/Kartu.java)
- Kelas KartuSpellMorph (app/src/main/java/Menkrep/Model/Kartu/KartuSpellMorph.java) merupakan anak dari kelas KartuSpell (app/src/main/java/Menkrep/Model/Kartu/KartuSpell.java), Kelas KartuSpell merupakan anak dari kelas Kartu (app/src/main/java/Menkrep/Model/Kartu/Kartu.java)

Berikut merupakan cuplikan kode dari kelas KartuKarakter dan KartuSpell yang merupakan kelas turunan dari kelas Kartu.

```
public class KartuKarakter extends Kartu {
```

```
    private String jenis; // OVERWORLD END NETHER
    private int exp;
    private int level;
    private int health;
    private int attack;
    private int attackUp;
    private int healthUp;
    private ArrayList<KartuSpell> activeSpells;
    private boolean doneAttack;

    private ArrayList<Integer> healthTemp;
    private ArrayList<Integer> attackTemp;
    private ArrayList<Integer> duration;
    private int swapDuration;

    public KartuKarakter(String nama, String deskripsi, String jenis, int exp, int level, int health,
            int attack, int attackUp, int healthUp, String imgPath, int mana) {
        super(nama, deskripsi, "KARAKTER", mana, imgPath);
        this.jenis = jenis;
        this.exp = exp;
        this.level = level;
        this.health = health;
        this.attack = attack;
        this.attackUp = attackUp;
        this.healthUp = healthUp;
```

```
        this.activeSpells = new ArrayList<KartuSpell>();
        this.healthTemp = new ArrayList<>();
        this.attackTemp = new ArrayList<>();
        this.duration = new ArrayList<>();
        this.doneAttack = false;
        this.doneAttack = false;
        this.swapDuration = 0;
        super.setImgPath(imgPath);
        super.setMana(mana);
    }
```

```
public class KartuSpell extends Kartu {

    public KartuSpell(String nama, String deskripsi, String tipe, int mana, String imgPath) {
        super(nama, deskripsi, tipe, mana, imgPath);
    }
```

2.2. Interface

Pada program ini, kami mengimplementasikan konsep interface dengan membuat 2 interface, yaitu Attackable dan KartuSpellTemp. Kedua interface tersebut diimplementasikan oleh beberapa kelas yang ada. Untuk interface KartuSpellTemp diimplementasikan pada kelas KartuSpellSwap dan KartuSpellPotion. Sedangkan pada interface Attackable diimplementasikan pada kelas kelas Player. Berikut merupakan Daftar beserta contoh interface yang terdapat pada program:

- Interface Attackable (app/src/main/java/Menkrep/Model/Player/Attackable.java)

```
public interface Attackable {
    void attack(Player opponent);
}
```

- Interface KartuSpellTemp(app/src/main/java/Menkrep/Model/Kartu/KartuSpellTemp.java)

```
public interface KartuSpellTemp {
    int getDuration();
    void setDuration(int newDuration);
    void reduceDuration();
}
```

2.3. Method Overriding

Method overriding adalah method yang dilakukan saat ingin membuat ulang sebuah method pada class turunan/anak. Untuk mengimplementasikan Method overriding ini, Kami menambahkan notasi @Override sebelum pembuatan method untuk menandakan penggunaan Method Overriding. Pada program ini, method toString() dapat di override oleh kelas yang membutuhkan seperti pada kelas KartuKarakter, KartuSpell dan lainnya. Berikut merupakan Daftar penggunaan konsep Method Overriding pada program yang telah kami buat:

- Method attack pada kelas KartuKarakter (app/src/main/java/Menkrep/Model/Kartu/KartuKarakter.java) dan Player (app/src/main/java/Menkrep/Model/Player/Player.java)
- Method start pada kelas App (app/src/main/java/Menkrep/App.java)

- Method toString() pada kelas KartuKarakter (app/src/main/java/Menkrep/Model/Kartu/KartuKarakter.java)
- Method toString() pada kelas KartuSpellLvl (app/src/main/java/Menkrep/Model/Kartu/KartuSpellLvl.java)
- Method toString() pada kelas KartuSpellMorph (app/src/main/java/Menkrep/Model/Kartu/KartuSpellMorph.java)
- Method toString() pada kelas KartuSpellPotion (app/src/main/java/Menkrep/Model/Kartu/KartuSpellPotion.java)
- Method toString() pada kelas KartuSpellSwap (app/src/main/java/Menkrep/Model/Kartu/KartuSpellSwap.java)
- Method toString() pada kelas KartuKarakter (app/src/main/java/Menkrep/Model/Kartu/KartuKarakter.java)

```java
@Override
    public String toString() {
        return "Type=" + jenis +
                "\nExp=" + exp +
                "\nLevel=" + level +
                "\nHealth=" + health +
                "\nAttack=" + attack +
                "\nAttackUp=" + attackUp +
                "\nHealthUp=" + healthUp;
    }
```

```java
@Override
    public String toString() {
        return "Mana = " + getMana() +
                "\nDuration=" + duration +
                "\nAtk=" + attackModifier +
                "\nHp=" + healthModifier;
    }
```

2.4. Polymorphism

Konsep polymorphism dapat diimplementasikan pada method di dalam program ini. Pada program ini fungsi yang memanfaatkan konsep polymorphism adalah method getDisplayString(). Method ini tidak memiliki parameter dan berfungsi untuk menampilkan sebuah string yang merupakan keterangan dari masing masing kelas. Keuntungan dari penggunaan konsep ini adalah mengurangi coupling, meningkatkan readability kode yang telah dibuat, dan menggunakan fungsi yang sama untuk tipe data yang berbeda. Berikut merupakan daftar penggunaan konsep polymorphism:

- Method getDisplayString() pada kelas Kartu yang di-*override* pada kelas turunan KartuKarakter, KartuSpell, KartuSpellMorph, KartuSpellPotion, KartuSpellSwap.

```java
// Kartu.java
    public String getDisplayString() {
        return "";
    }
```

```java
// KartuSpellMorph.java
    @Override
    public String getDisplayString() {
        return "MORPH";
    }
```

2.5. Java API

Berikut merupakan daftar konsep Java API yang terdapat pada program ini:

- Kelas Player (app/src/main/java/Menkrep/Model/Player/Player.java) memiliki Array List of deck, Array List of hand, dan Array List of board.
- Kelas Reference (app/src/main/java/Menkrep/Model/Reference/Reference.java) memiliki list of string karakter, list of string morph, list of string ptn, dan list of string swap.
- Penggunaan kelas Random dengan method nextInt yang digunakan untuk melakukan pengacakan deck untuk pemain.
- Method nextInt pada kelas Random untuk melakukan pengacakan pengembalian dari 2 kartu pada draw phase agar kembali ke deck.
- Kelas KartuKarakter (java/Menkrep/Model/Kartu/KartuKarakter.java) memiliki ArrayList of KartuSpell activeSpells, ArrayList of Integer healthTemp, ArrayList of Integer attackTemp, dan ArrayList of Integer duration.

```
import java.util.ArrayList;
import java.util.Random;

public class Player {
    // Atribut
    private final String name;
    private int healthPoints;
    private int mana;
    private final int deckCapacity;
    private final ArrayList<Kartu> deck;
    private final ArrayList<Kartu> hand;
    private ArrayList<Kartu> draw;
    private final ArrayList<KartuKarakter> board;
```

3. Pembagian Tugas

- 13520057 / Marcellus Michael Herman K

    - Mana
    - Apply Spell
    - GUI gambar dan pola teks, nama variabel, attack dan spell
    - Debugging
- 13520066 / Putri Nurhaliza
    - Membuat class KartuSpell, Lvl, Potion, Morph, Swap
    - GUI: Bind Deck, Mana, Hand Card, dan Hover Card Details (description dan stats)
    - Testing spell, character
- 13520084 / Adelline Kania Setiyawan
    - Membuat class Player dan implementasinya
    - GUI: Bind Turn, Health Progress, Phase
    - Membuat logic pemindahan kartu dari hand ke board pada GUI
- 13520087 / Dimas Shidqi Parikesit
    - Setup gradle
    - Membuat class Game, Reference
    - Membuat interface KartuSpellTemp
    - Testing class Game, Reference
    - Membuat logic attack
    - Membuat logic spell
    - Membuat converter mana to exp
    - GUI next phase bind,
    - Debugging spell, attack, GUI, dll
- 13520111 / Rizky Akbar Asmaran
    - Membuat kelas Kartu
    - Membuat kelas KartuKarakter
    - Laporan
- 13520153 / Vito Ghifari
    - Membuat GUI kerangka tampilan utama, tampilan draw phase, dan tampilan game finish
    - Membuat fungsionalitas draw phase
    - Membuat kelas KartuSpellMorph dan KartuSpellSwap

- Membuat Test kelas Player

4. Lampiran

Kelas : 03
Nomor Kelompok : 02
Nama Kelompok : Menkrep

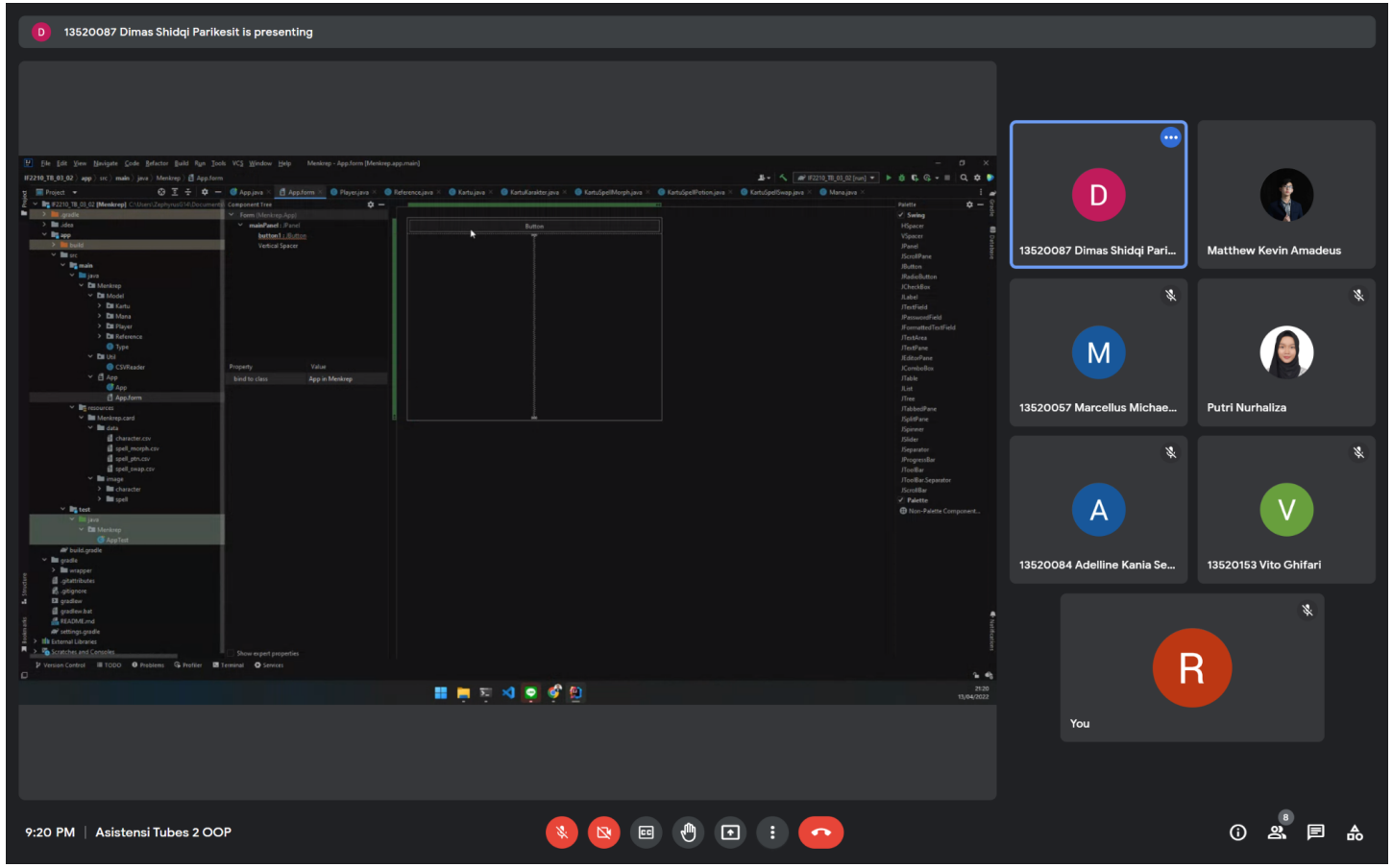
1. 13520057/Marcellus Michael Herman K
2. 13520066/Putri Nurhaliza
3. 13520084/Adelline Kania Setiyawan
4. 13520087/Dimas Shidqi Parikesit
5. 13520111/Rizky Akbar Asmaran
6. 13520153/Vito Ghifari

Asisten Pembimbing : Matthew Kevin Amadeus

1. Konten Diskusi

- Bikin GUI itu drag and drop dan tahun lalu pakai aplikasi lain (Scene Builder). GUI itu masalah dependency aja.
- Kelas diagram kita belum ada “Game”, “Deck” (ga terlalu wajib katanya) isinya id dari cardnya
- Mana tidak perlu dijadikan class, cek apakah suatu bagian itu perlu diimplementasikan menjadi kelas. Contohnya mana itu type ke pa
- Kelas summon: Kalian punya deck di player. Method take() pada deck buat ngambil kartu dari deck. Card itu hanya untuk objek yang ada di kelas Deck. Spell card juga, apa yang mau ditaroh di deck (?).
- https://refactoring.guru/design-patterns/singleton, cuma extension sekali. Extension nya disimpen di static variabel di class itu(?)
- Bisa manfaatin instance of, apakah objek ini dari class ini. Bisa consider buat inheritance. Misall spell card, kalau dia morph kan type nya sama dengan morph. Bisa di pass instance nya morph, gaperlu cek type nya morph. Buat rapiin doang

2. Screenshot Bukti


13520087 Dimas Shidqi Parikesit is presenting
13520087 Dimas Shidqi Pari...
Matthew Kevin Amadeus
13520057 Marcellus Michae...
Putri Nurhaliza
13520084 Adelline Kania Se...
13520153 Vito Ghifari
You
9:20 PM | Asistensi Tubes 2 OOP