

Kelas : 01

Nomor Kelompok : 02

Nama Kelompok : K111111

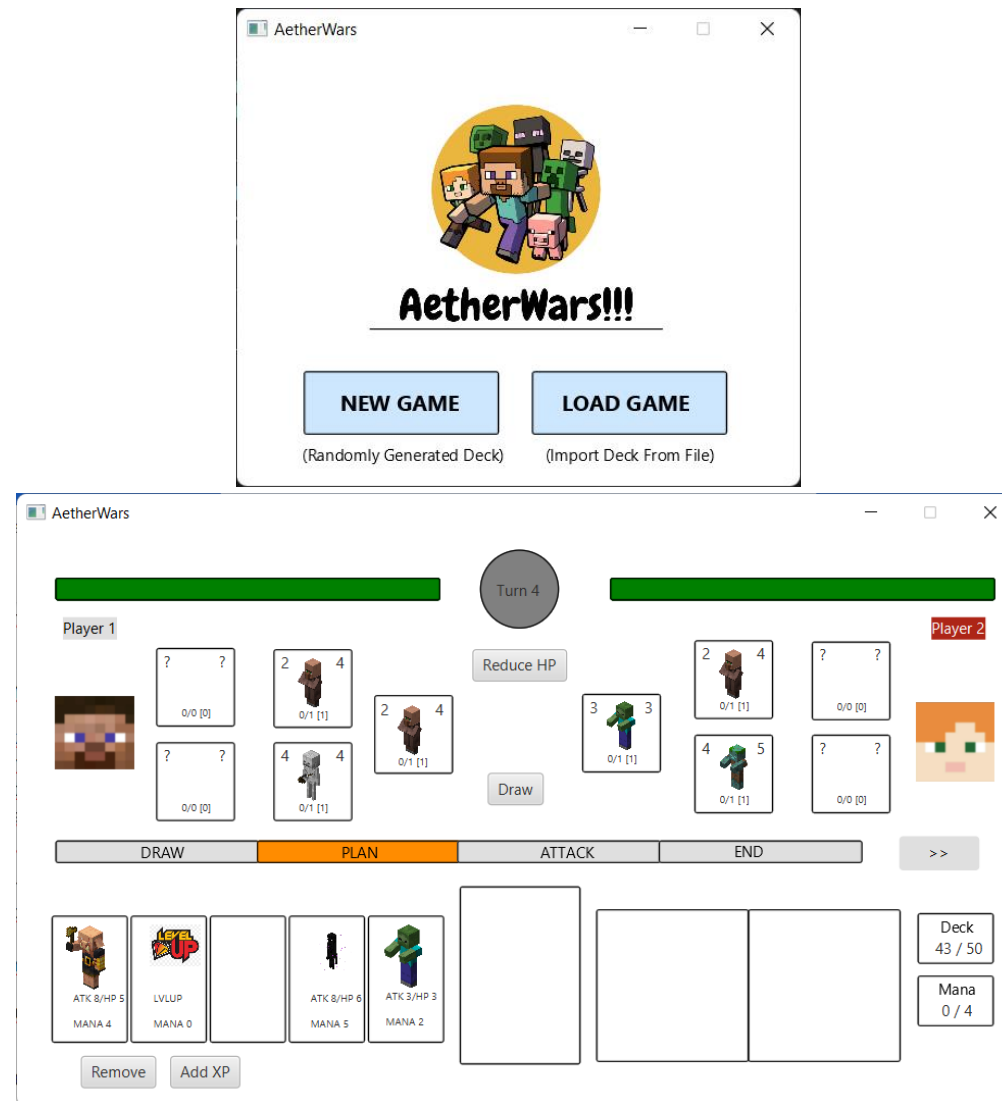
1. 13520016 / Gagas Praharsa Bahar
2. 13520085 / Ubaidillah Ariq Pratama
3. 13520094 / Saul Sayers
4. 13520109 / Patrick Amadeus Irawan
5. 13520130 / Nelsen Putra
6. 13520142 / Rania Dwi Fadhilah

Asisten Pembimbing : Muhammad Fauzan Rafi Sidiq Widjonarto

## 1. Deskripsi Umum Aplikasi

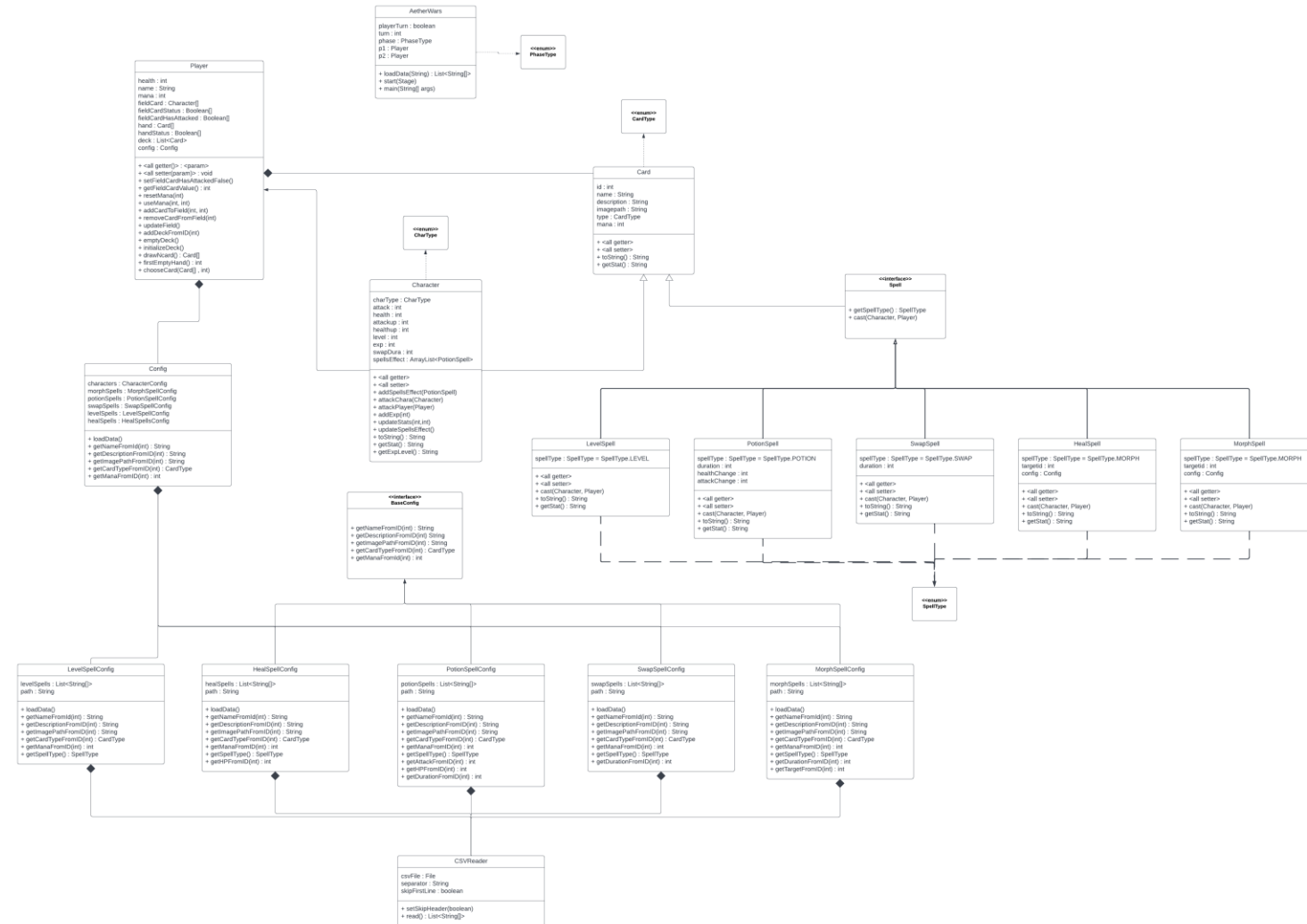
Aplikasi AetherWars merupakan aplikasi permainan game kartu *turn based* untuk 2 pemain. Pemain akan bermain secara bergantian pada 1 layar yang sama hingga tercapai tujuan akhir dari game. Tujuan akhir dari game adalah menghabiskan *health points* (HP) dari musuh. HP ini dapat berkurang jika mendapat serangan dari kartu karakter yang diletakkan pada *field* pertandingan. Harus diperhatikan pula bahwasanya pemain juga kalah ketika isi *deck*-nya kosong. Permainan sendiri dapat dimulai dengan menggunakan kartu *deck* baru maupun melanjutkan permainan dengan meng-*import* kartu *deck* yang lama.

Permainan ini dibuat dengan menggunakan bahasa pemrograman Java dan memanfaatkan Permainan ini dibuat dengan menggunakan bahasa pemrograman Java dan memanfaatkan library GUI yaitu JavaFX. Terdapat banyak kelas yang dibuat pada permainan ini. Kelas-kelas tersebut dibuat dengan mengimplementasikan konsep OOP yang dipelajari pada mata kuliah IF2210 Pemrograman Berorientasi Objek. Untuk menjalankan permainan ini, pemain hanya perlu membuka terminal dari folder permainan ini dan mengetik **`./gradlew run`**



Contoh screenshot aplikasi permainan kami

## 2. Diagram Kelas



### 3. Konsep OOP

#### 3.1. Inheritance

- Kelas PotionSpell (src/main/java/com/aetherwars/model/Spell/java) merupakan anak dari kelas Spell (src/main/java/com/aetherwars/model/Spell/java)
- Kelas LevelSpell (src/main/java/com/aetherwars/model/Spell/java) merupakan anak dari kelas Spell (src/main/java/com/aetherwars/model/Spell/java)
- Kelas HealSpell (src/main/java/com/aetherwars/model/Spell/java) merupakan anak dari kelas Spell (src/main/java/com/aetherwars/model/Spell/java)
- Kelas Spell (src/main/java/com/aetherwars/model/Spell/java) merupakan anak dari kelas Card (src/main/java/com/aetherwars/model/Card.java)
- Kelas Character (src/main/java/com/aetherwars/model/Character.java) merupakan anak dari kelas Card (src/main/java/com/aetherwars/model/Card.java)

#### 3.2. Composition

- Kelas Player (src/main/java/com/aetherwars/model/Player.java) memiliki kelas Character (src/main/java/com/aetherwars/model/Character.java), kelas Card (src/main/java/com/aetherwars/model/Card.java), dan kelas Config (src/main/java/com/aetherwars/util/Config.java)
- Kelas AetherWars (src/main/java/com/aetherwars/AetherWars.java) memiliki kelas Player (src/main/java/com/aetherwars/model/Player.java) dan kelas PhaseType (src/main/java/com/aetherwars/model/type/PhaseType.java)
- Kelas Spell (src/main/java/com/aetherwars/model/Spell.java) memiliki kelas SpellType (src/main/java/com/aetherwars/model/type/SpellType.java)
- Kelas Character (src/main/java/com/aetherwars/model/Character.java) memiliki kelas CharType (src/main/java/com/aetherwars/model/type/CharType.java)

- Kelas Config (src/main/java/com/aetherwars/util/Config.java) memiliki kelas CharactersConfig, PotionSpellsConfig, SwapSpellsConfig, HealSpellsConfig, MorphSpellsConfig, LevelSpellsConfig (src/main/java/com/aetherwars/util/config/)

### **3.3. Interface**

- Interface BaseConfig (src/main/java/com/aetherwars/util/config/BaseConfig.java)

### **3.4. Method Overriding**

- Method cast pada kelas PotionSpell (src/main/java/com/aetherwars/model/Spell.java)
- Method toString pada kelas PotionSpell (src/main/java/com/aetherwars/model/Spell.java)
- Method getStat pada kelas PotionSpell (src/main/java/com/aetherwars/model/Spell.java)
- Method toString pada kelas Character (src/main/java/com/aetherwars/model/Character.java)
- Method getStat pada kelas Character (src/main/java/com/aetherwars/model/Character.java)

### **3.5. Polymorphism**

- Kelas Player (src/main/java/com/aetherwars/model/Player.java) memiliki atribut hand berupa Array of Card, tetapi pada method chooseCard di kelas Player atribut ini diisi dengan kelas Character (src/main/java/com/aetherwars/model/Character.java) dan Spell (src/main/java/com/aetherwars/model/Spell/java)
- Kelas Player (src/main/java/com/aetherwars/model/Player.java) memiliki atribut deck berupa List of Card, tetapi pada method initializeDeck dan addDeckFromID di kelas Player atribut ini diisi dengan kelas Character Character (src/main/java/com/aetherwars/model/Character.java) dan Spell (src/main/java/com/aetherwars/model/Spell/java)
- Method getMana, getID, getName pada kelas Card dipanggil menggunakan kelas Card, tapi pada method-method ini pada kelas Spell (src/main/java/com/aetherwars/model/Spell/java), dipanggil menggunakan kelas Potion, PotionSpell, SwapSpell, HealSpell, MorphSpell, LevelSpell

- Method `getMana`, `getID`, `getName` pada kelas `Card` dipanggil menggunakan kelas `Card`, tapi pada method-method ini pada kelas `Character` `Character` (`src/main/java/com/aetherwars/model/Character.java`), dipanggil menggunakan kelas `Character`

### 3.6. Java API

- Kelas `AetherWars` (`src/main/java/com/aetherwars/AetherWars.java`) memiliki List of Array of String
- Kelas `Controller` (`src/main/java/com/aetherwars/Controller.java`) memiliki banyak `ArrayList` dan banyak sekali penggunaan List
- Kelas `Player` (`src/main/java/com/aetherwars/model/Player.java`) memiliki `ArrayList` of `Card`
- Kelas `CSVReader` (`src/main/java/com/aetherwars/util/config/CSVReader.java`) memiliki Array List of Array of String
- Method `initializeDeck` pada kelas `Player` memanfaatkan java API `Collections` `shuffle`

### 3.7. SOLID

- S - Single-responsibility Principle
  - Semua kelas turunan `BaseConfig` hanya bertugas untuk menangani satu file konfigurasi saja, seperti `CharacterConfig` yang akan menyimpan konfigurasi karakter, `PotionSpellConfig` yang akan menyimpan konfigurasi `PotionSpell`, dan lain-lain.
  - Kelas `Character` (`model/Character.java`) bertugas untuk menjadi container dari entity `Character`, sedangkan kelas `Spell` berlaku sebagai container dari entity `Spell` yang memiliki tugas dan fungsionalitas berbeda pada permainan.
  - Enum `CardType`, `CharType`, `PhaseType`, dan `SpellType` (`model/type/*.java`) semuanya menggambarkan enumerasi tipe yang berbeda dan hanya dipakai untuk menjelaskan tipe terkait.
- O - Open-closed Principle
  - Pada kelas `Character` (`model/Character.java`) yang di extend dari kelas `Card` (`model/Card.java`), diimplementasikan Open-closed Principle karena `Character` dapat mengakses atribut pada `Card` tanpa mengubahnya.
  - Pada kelas `Spell` (`model/Spell.java`) yang di extend dari kelas `Card` (`model/Card.java`), diimplementasikan Open-closed Principle karena `Spell` dapat mengakses atribut pada `Card` tanpa mengubahnya.
  - Pada kelas `PotionSpell`, `SwapSpell`, `MorphSpell`, `LevelSpell`, `HealSpell` (`model/Spell.java`) yang di extend dari kelas `Spell` (`model/Spell.java`), diimplementasikan Open-closed Principle karena `PotionSpell`, `SwapSpell`, `MorphSpell`, `LevelSpell`, `HealSpell` dapat mengakses atribut pada `Spell` tanpa mengubahnya.
- L - Liskov Substitution Principle

- Kelas Player (src/main/java/com/aetherwars/model/Player.java) memiliki atribut hand berupa Array of Card, tetapi pada method chooseCard di kelas Player atribut ini diisi dengan kelas Character (src/main/java/com/aetherwars/model/Character.java) dan Spell (src/main/java/com/aetherwars/model/Spell.java). Hal ini menunjukkan bahwa Character dapat berlaku sebagai Card, yaitu kelas parentnya.
- Method getMana, getID, getName pada kelas Card dipanggil menggunakan kelas Card, tapi pada method-method ini pada kelas Spell (src/main/java/com/aetherwars/model/Spell.java), dipanggil menggunakan kelas Potion, PotionSpell, SwapSpell, HealSpell, MorphSpell, LevelSpell
- Kelas Player (src/main/java/com/aetherwars/model/Player.java) memiliki atribut deck berupa List of Card, tetapi pada method initializeDeck dan addDeckFromID di kelas Player atribut ini diisi dengan kelas Character Character (src/main/java/com/aetherwars/model/Character.java) dan Spell (src/main/java/com/aetherwars/model/Spell.java)
- I - Interface Segregation Principle
  - Pada kelas-kelas Config (CharacterConfig, PotionSpellConfig, dll) semua mengimplement interface BaseConfig dan ManaConfig, kecuali LevelSpellConfig yang tidak mengimplementasikan getMana karena tidak ada data mana pada konfigurasinya. Dengan segregasi ini, kelas-kelas tersebut tidak mengimplementasikan method atau interface yang tidak diperlukan.
- D - Dependency Inversion Principle
  -

### 3.8. Design Pattern

- Digunakan design pattern Decorator dengan Spell (src/main/java/com/aetherwars/model/Spell.java) dan turunannya, yaitu kelas PotionSpell, LevelSpell, SwapSpell, MorphSpell, dan HealSpell. Kelima kelas berlaku sebagai *wrapper* dari kelas spell, menambah atribut dan kegunaan baru kepada kelas Spell.
- Digunakan design pattern Prototype dengan Config (src/main/java/com/aetherwars/util/Config.java). Config pada program ini berlaku sebagai prototype untuk Card, sehingga untuk membuat instance card baru, hanya dibutuhkan data yang dapat diambil langsung dari kelas config.
- Digunakan design pattern behavioral State pada Controller.java pada pengontrolan Phase (DRAW, PLAN, ATTACK, END). Tombol-tombol dan juga game itu sendiri akan memiliki fungsi yang berbeda saat dijalankan pada fase yang berbeda pula.

## 4. Bonus Yang dikerjakan

### 4.1. Bonus yang diusulkan oleh spek

#### 4.1.1. Import Deck

Pendekatan kami dalam membuat bonus fitur import deck menggunakan sebuah CSV reader. Pada tampilan pertama aplikasi, pengguna dapat memilih untuk men-generate deck secara random, atau menggunakan fitur import deck agar mendapatkan kartu – kartu yang sudah pasti. Pada halaman import deck, terdapat 2 form yang dapat diisi yakni masing – masing nama file untuk player 1 dan player 2 (dengan extensionnya juga). Jenis *file* yang dapat dibaca baik .txt file ataupun .csv. *File* konfigurasi isi deck tersebut terletak pada direktori src\main\resources\com\aetherwars\deck. Baris pertama merupakan banyaknya kartu yang diisi, dan seluruh baris setelahnya merupakan id kartu. CSV reader akan membaca isi file tersebut dan mengembalikan sebuah list of array of string. Pada kasus ini, tiap baris hanya terdiri dari satu kolom sehingga sudah dapat dipastikan dimensi array of string adalah 1. Apabila nama file yang diinput tidak ditemukan untuk salah satu form, maka aplikasi akan menampilkan pesan bahwa nama file tidak ditemukan, silahkan mengisi ulang nama file. Apabila nama file ditemukan, isi *file* akan dibaca dan langsung menambahkannya ke dalam deck player, lalu berpindah ke halaman utama aplikasi untuk dapat dimainkan. Berikut adalah cuplikan kode yang relevan dengan pendekatan kami :



```

String filename1 = "./com/aetherwars/deck/" + player1box.getText();
String filename2 = "./com/aetherwars/deck/" + player2box.getText();

boolean file1ada;
boolean file2ada;

if (AetherWars.class.getClassLoader().getResource(filename1) != null) {
    file1ada = true;
} else {
    file1ada = false;
}

if (AetherWars.class.getClassLoader().getResource(filename2) != null) {
    file2ada = true;
} else {
    file2ada = false;
}

if (!file1ada && !file2ada){
    errorText.setText(value: "Filename 1 and 2 not found!");
    return;
}

if (!file1ada) {
    errorText.setText(value: "Filename 1 not found!");
    return;
}

if (!file2ada){
    errorText.setText(value: "Filename 2 not found!");
    return;
}

try {
    List<String[]> tempdeck1 = loadCSV(filename1);
    List<String[]> tempdeck2 = loadCSV(filename2);
    for (String[] baris : tempdeck1){
        AetherWars.p1.addDeckFromID( Integer.parseInt(baris[0]));
    }
    for (String[] baris : tempdeck2){
        AetherWars.p2.addDeckFromID( Integer.parseInt(baris[0]));
    }
    Stage popupwindow = (Stage) gobackBox.getScene().getWindow();
    Pane throwPane = FXMLLoader.load(AetherWars.class.getClassLoader().getResource(name: "./Sample.fxml"));
    popupwindow.setScene(new Scene(throwPane));
    popupwindow.centerOnScreen();
}
catch (URISyntaxException e) {
    errorText.setText(value: "FILE NOT FOUND");
}
catch (IOException e) {
    errorText.setText(value: "FILE NOT FOUND");
}

public List<String[]> loadCSV(String filename) throws URISyntaxException, IOException {
    File characterCSVFile = new File(AetherWars.class.getClassLoader().getResource(filename).toURI());
    CSVReader characterReader = new CSVReader(characterCSVFile, separator: "\t");
    characterReader.setSkipHeader(isSkippingHeader: true);
    return characterReader.read();
}

```

#### 4.1.2. Spell Baru

Spell baru yang diimplementasikan oleh kelompok kami adalah spell Heal. Terdapat 3 spell yang kami buat, yaitu Green Herbs, Indomi Goreng, dan juga Dukungan Gebetan. Jumlah health dan mana yang dihasilkan dan dibutuhkan bervariasi. Berikut adalah cuplikan csv spell Heal :

src > main > resources > com > aetherwars > card > data >  spell\_heal.csv

```

1 "id name      description imagepath  health  mana"
2 "501  Green Herbs Plant that can heal player  card/image/spell/heal/greenherbs.jpg  10  3"
3 "502  Indomi goreng  Bisa dimakan dan nambah nyawa  card/image/spell/heal/indomigoreng.png  25  5"
4 "503  Dukungan Gebetan  Nyawa langsung naik drastis!!  card/image/spell/heal/gebetan.png  50  8"
5

```

Spell Heal ini berfungsi untuk memberikan *heal* kepada karakter, dimana *health* dari si karakter akan bertambah sejumlah *health* dari spellnya. Berikut adalah cuplikan kode dari implementasi spell heal :

```

1 class HealSpell extends Spell {
2     private final SpellType spellType = SpellType.HEAL;
3     private int health;
4
5     public HealSpell(){
6         super();
7     }
8
9     public HealSpell(Card card, int health){
10        super(card);
11        this.health = health;
12    }
13
14    public HealSpell(int health) {this.health = health; }
15
16    public int getHealth(){
17        return this.health;
18    }
19
20    @Override
21    public void cast(Character c, Player p){
22        if(p.getMana() >= this.getMana()){
23            int temp = c.getHealth();
24            c.setHealth(temp + this.getHealth());
25            p.setMana(p.getMana() - this.getMana());
26        }
27    }
28
29    public SpellType getSpellType(){
30        return this.spellType;
31    }
32
33    @Override
34    public String toString() {
35        return "HEAL" + " (" + Integer.toString(this.getHealth()) + ")";
36    }
37
38    public String getStat(){
39        return this.getName() + "\nHEALTH: " + this.getHealth() + "\nTYPE: " + this.getSpellType();
40    }
41 }

```

## 5. Library External

- Java.io : Digunakan untuk membuat objek File untuk mengetahui apakah file tersebut ada (dalam import deck).
- Java.net : Digunakan untuk mendapatkan URL dari sebuah path dalam loading file user interface (FXML). Selain itu digunakan untuk mendapatkan jenis
- Java.util : Digunakan untuk mendapatkan List, ArrayList, serta Array untuk menyimpan kumpulan objek.
- Javafx.scene : Digunakan untuk menciptakan scene serta fungsi2nya dalam pembuatan user interface. Di antaranya adalah pengaturan button, label, image, event, shape, color, layout, pane, serta anchorpane.
- Javafx.stage : Digunakan untuk menciptakan stage serta fungsi2nya dalam pembuatan user interface. Selain itu juga digunakan untuk mengatur modality dari sebuah window.
- Javafx.fxml : Digunakan untuk mendapatkan method loader sebuah file user interface (FXML). Selain itu juga digunakan untuk melakukan initialize dari sebuah window.
- Javafx.application : Digunakan untuk meluncurkan aplikasinya secara keseluruhan..
- Javafx.lang.Math : Digunakan untuk mendapatkan fungsi min dan max.

## 6. Pembagian Tugas

- 13520016 / Gagah Praharsa Bahar
  - Attack Phase
  - Game Phase state
  - Struktur Data
  - Config
- 13520085 / Ubaidillah Ariq Pratama
  - Plan Phase
  - Implementasi Spell
  - Implementasi Character
  - Struktur Data
- 13520094 / Saul Sayers
  - Draw Phase

- Opening/Ending Screen
- Spell Baru
- Import Deck
- Struktur Data
- 13520109 / Patrick Amadeus Irawan
  - Base UI
  - Spell Baru
  - Plan Phase
  - Struktur Data
- 13520130 / Nelsen Putra
  - Implement Kelas
- 13520142 / Rania Dwi Fadhilah
  - Attack Phase
  - Struktur Data
  - Game Phase State