

Kelas : 02

Kelompok : 07

1. 13520005 / Christine Hutabarat

2. 13520050 / Felicia Sutandijo

3. 13520077 / Rava Naufal Attar

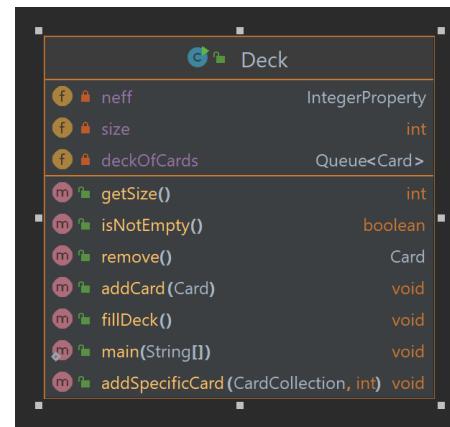
4. 13520140 / Febryola Kurnia Putri

5. 13520164 / Hilda Carissa Widelia

Asisten Pembimbing : 13520084 / Jonathan Yudi Gunawan

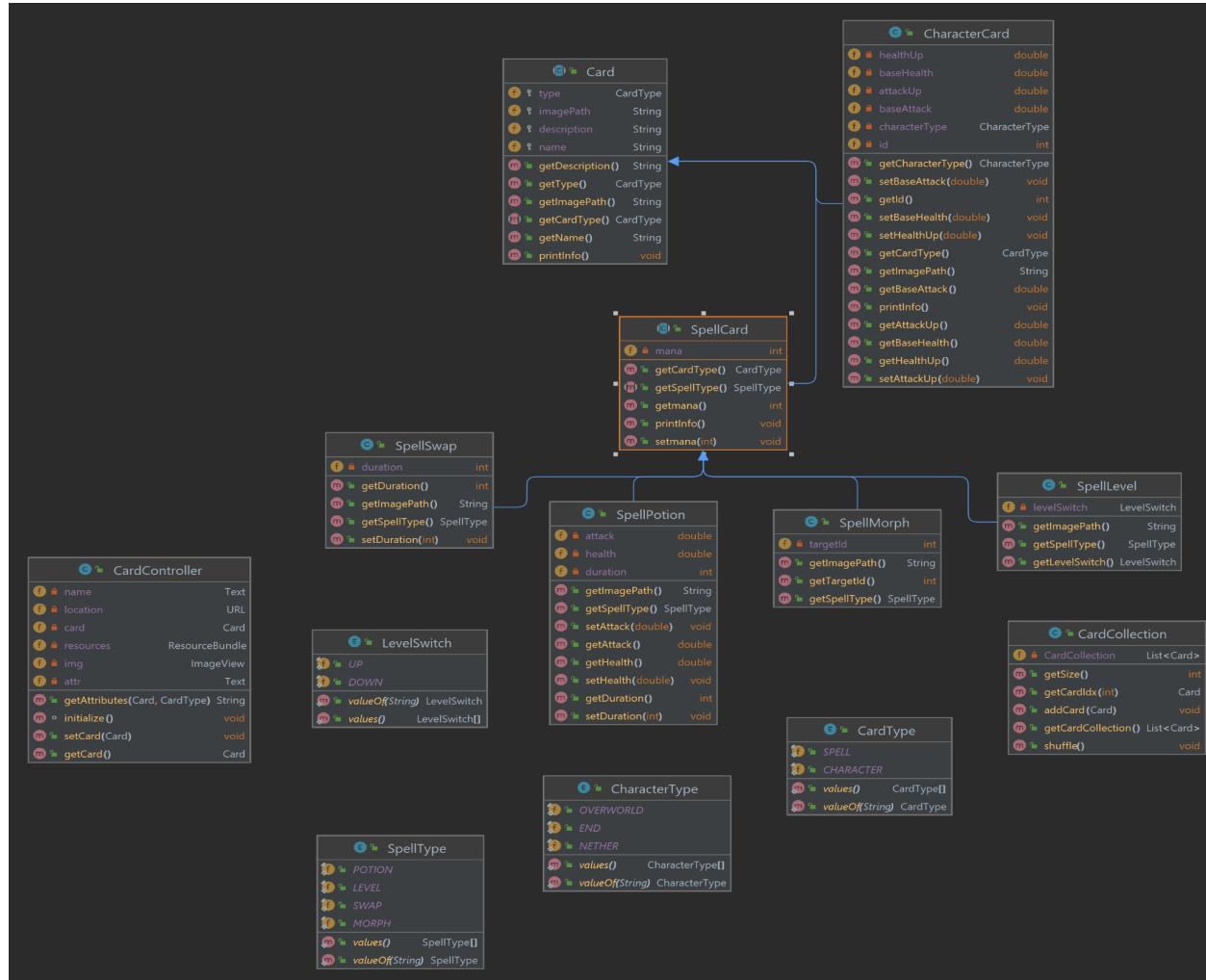
## 1. Class Diagram

### a. Diagram pada folder deck



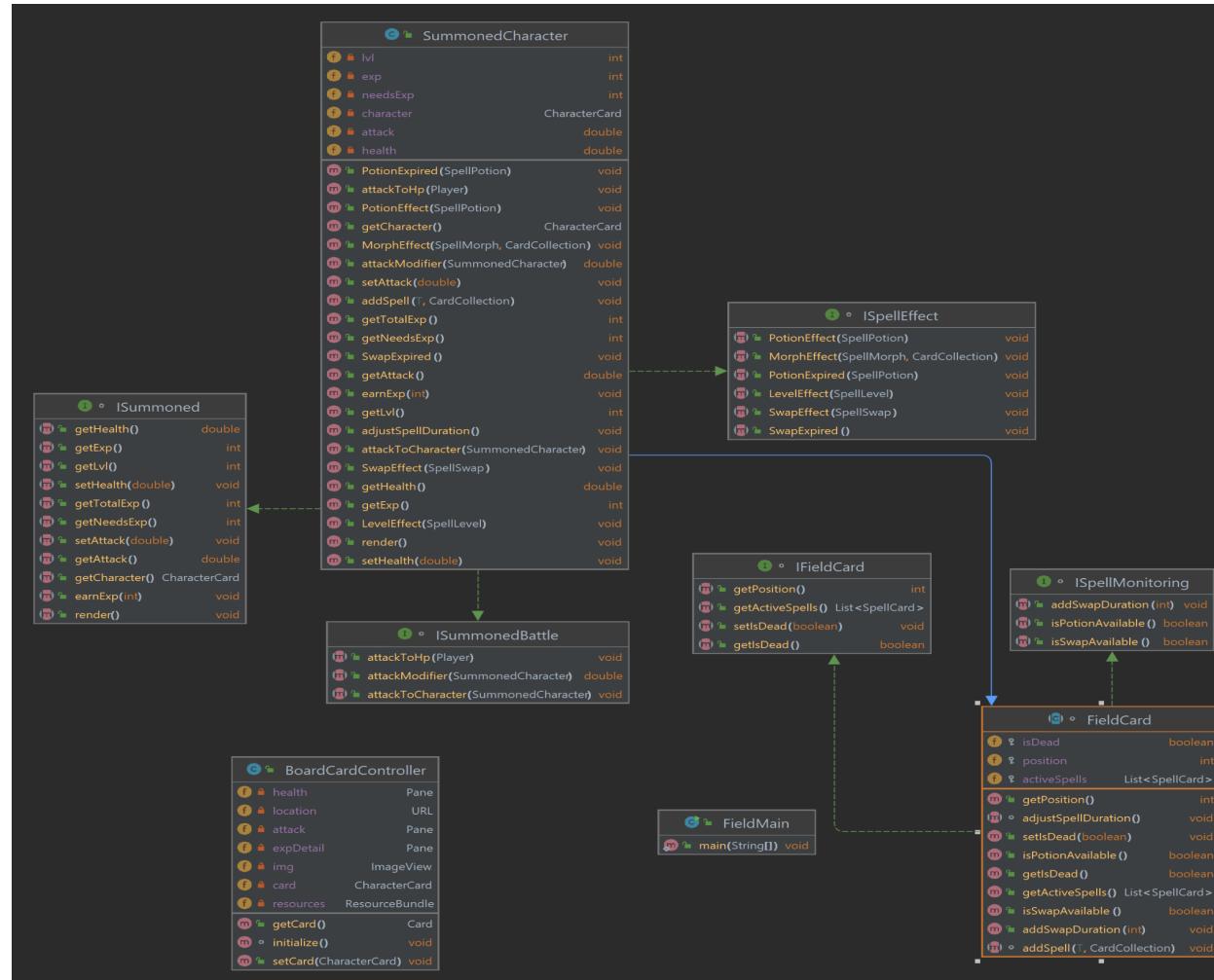
Gambar 1.1 Diagram Kelas Deck

## b. Diagram pada folder card



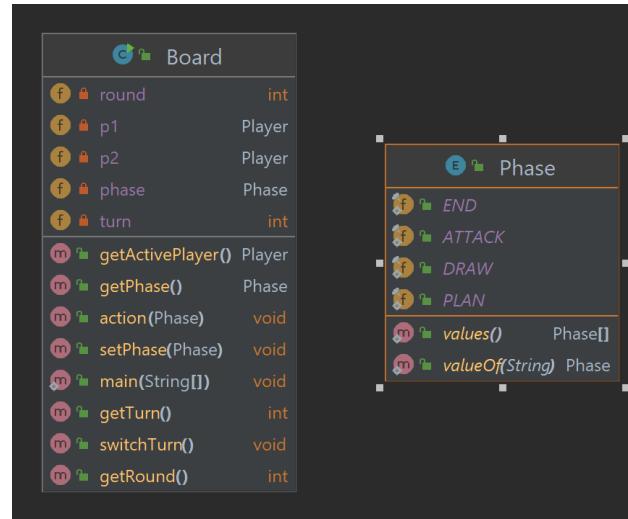
Gambar 1.2 Diagram Kelas pada folder card

### c. Diagram pada folder fieldcard



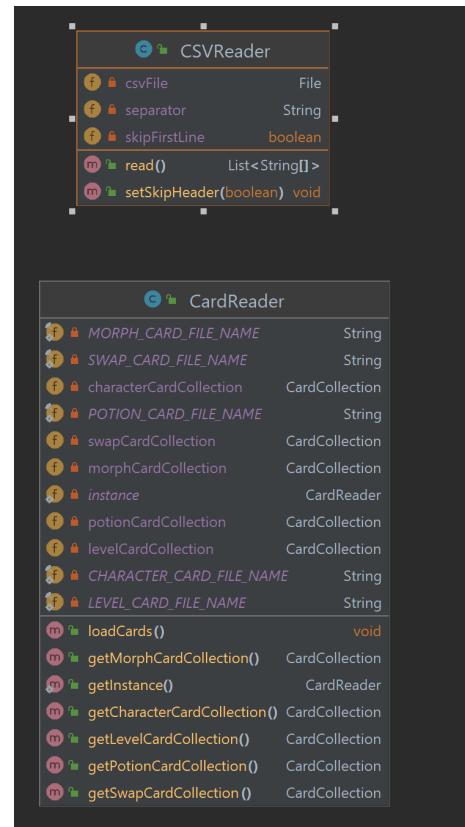
Gambar 1.3 Diagram kelas pada folder fieldcard

d. Diagram pada folder board



Gambar 1.4 Diagram Kelas pada folder board

e. Diagram pada folder util



Gambar 1.5 Diagram Kelas pada folder util

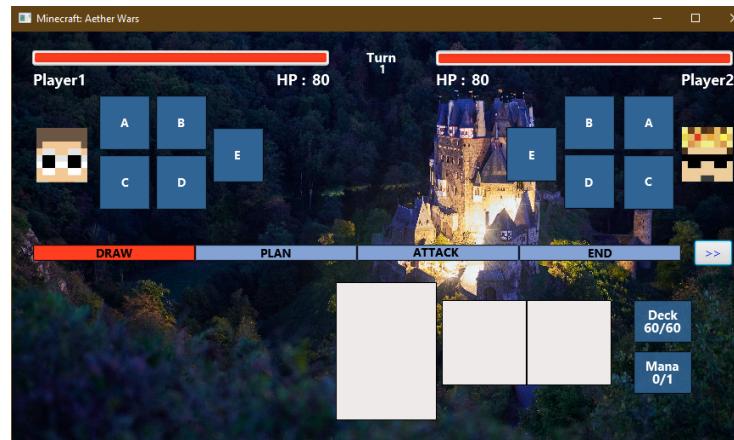
**f. Diagram pada folder player**



Gambar 1.6 Diagram Kelas Player

## 2. Deskripsi Umum Aplikasi

Pada tugas besar ini, dibuat sebuah aplikasi bernama AetherWars. AetherWars merupakan sebuah permainan kartu yang melibatkan 2 orang pemain, dimainkan pada satu layar yang sama secara bergantian. Pemain awalnya memiliki 80 *health points* (HP) dan pemain yang berhasil menghabiskan HP lawannya-lah yang menang. Pemain dapat saling menyerang menggunakan kartu yang dimilikinya pada setiap *turn*.



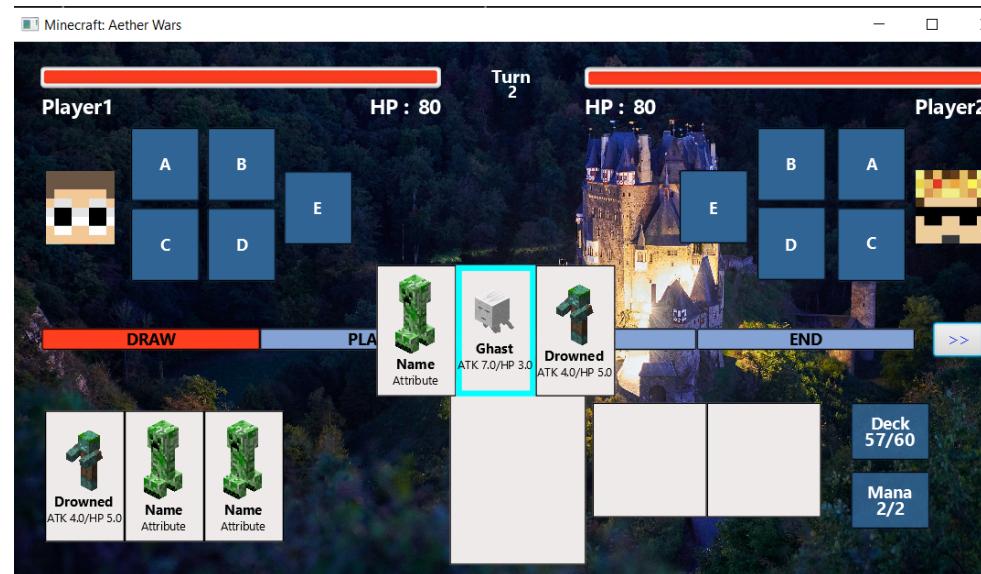
Gambar 2.1. Tampilan Permainan Awal

Permainan AetherWars ini melibatkan beberapa komponen, di antaranya *player* (pemain), *deck*, kartu, HP, dan mana. Setiap pemain memiliki *deck*, kartu di tangan, kartu di papan, HP, serta mana masing-masing yang dapat digunakan pada giliran pemain tersebut. Sebuah giliran atau *turn* pemain terdiri dari 4 *phase*, yaitu *draw*, *plan*, *attack*, dan *end*. Pada awal permainannya, 3 kartu dari *deck* akan dipindahkan ke *hand card* sehingga kartu pada *deck* bersisa 57. Kartu yang ditampilkan dilakukan secara random.

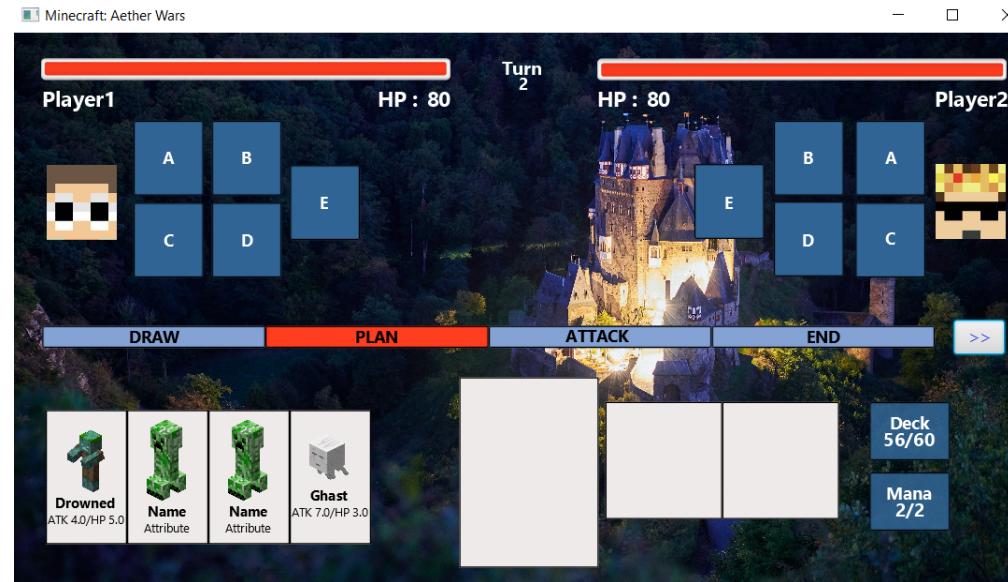


Gambar 2.2. Tampilan Permainan saat Turn Player 1

Pada saat *phase draw* awal, 3 buah kartu random yang berasal dari Deck akan diletakkan pada hand card. Lalu pada saat turn berikutnya, akan ditampilkan tiga kartu Pemain yang sedang bermain akan mengambil **3** kartu teratas dari deck, lalu memilih **1** kartu untuk dimasukkan ke *hand* dan mengembalikan **2** kartu yang lain ke *deck* secara acak. Apabila kartu di *hand* melebihi **5** kartu, pemain **harus membuang** kartu hingga berjumlah 5 kartu. Berikut contoh tampilannya:



Gambar 2.3. Tampilan fase draw saat memilih kartu



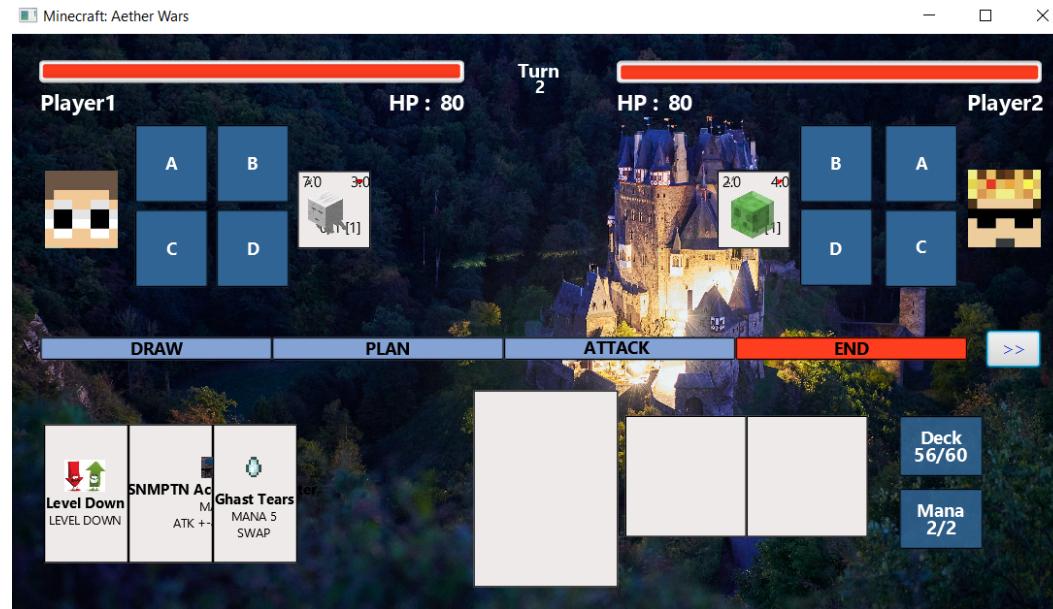
Gambar 2.4 Kartu yang telah dipilih dan sisanya dikembalikan ke deck

Saat Phase plan, kartu karakter akan dimasukkan ke board card sesuai dengan turn masing-masingnya. Berikut tampilan phase plan pada aplikasi ini:



Gambar 2.5. Tampilan saat fase plan dan attack dimulai

Pada saat phase end, maka aplikasi tidak melakukan apa-apa, berikut tampilannya pada aplikasi ini:



Gambar 2.5. Tampilan saat fase end

### 3. Konsep OOP

#### 3.1. Inheritance

Berikut penerapan konsep inheritance pada aplikasi:

- Kelas CharacterCard (src/main/java/com.aetherwars/card/CharacterCard.java) dan SpellCard (src/main/java/com.aetherwars/card/SpellCard.java) merupakan anak dari kelas Card (src/main/java/com.aetherwars/card/Card.java)
- Kelas SpellLevel (src/main/java/com.aetherwars/card/SpellLevel.java), SpellMorph (src/main/java/com.aetherwars/card/SpellMorph.java), SpellPotion (src/main/java/com.aetherwars/card/SpellPotion.java), dan SpellSwap (src/main/java/com.aetherwars/card/SpellSwap.java) merupakan anak dari kelas SpellCard (src/main/java/com.aetherwars/card/SpellCard.java)

### 3.2. Composition

Berikut composition yang diterapkan pada aplikasi:

- Kelas Player (src/main/java/com.aetherwars/player/Player.java) memiliki tepat satu kelas Deck (src/main/java/com.aetherwars/deck/Deck.java).
- Kelas SummonedCharacter (src/main/java/com.aetherwars/fieldcard/SummonedCharacter.java) memiliki tepat satu kelas CharacterCard(src/main/java/com.aetherwars/card/CharacterCard.java)

### 3.3. Interface

Berikut penerapan konsep interface pada aplikasi sebagai berikut:

- Interface IFieldCard (src/main/java/com.aetherwars/fieldcard/IFieldCard.java)
- Interface ISpellMonitoring(src/main/java/com.aetherwars/fieldcard/IFieldCard.java)
- Interface ISummoned(src/main/java/com.aetherwars/fieldcard/ISummoned.java)
- Interface ISpellEffect(src/main/java/com.aetherwars/fieldcard/ISummoned.java)
- Interface ISummonedBattle(src/main/java/com.aetherwars/fieldcard/ISummoned.java)

### 3.4. Method Overriding

Berikut penerapan method overriding pada aplikasi ini:

- Method printInfo() pada kelas Card (src/main/java/com.aetherwars/card/Card.java)
- Method getCardType() pada kelas Card (src/main/java/com.aetherwars/card/Card.java)
- Method getImagePath() pada kelas Card (src/main/java/com.aetherwars/card/Card.java)
- Method getSpellType() pada kelas SpellCard (src/main/java/com.aetherwars/card/SpellCard.java)

### 3.5. Polymorphism

Berikut penerapan polymorphism pada aplikasi ini:

- Method `addSpell()` pada Kelas `SummonedCharacter` yang merupakan turunan dari Kelas `FieldCard` (`src/main/java/com.aetherwars/fieldcard/SummonedCharacter.java`) menerima *generic type inherit* dari `SpellCard`, tapi spell yang ditambahkan akan ditampung pada `List<SpellCard> activeSpells` yang merupakan attribut dari Kelas `FieldCard`

### 3.6. Java API

- Kelas `FieldCard` (`src/main/java/com.aetherwars/fieldcard/FieldCard.java`) memiliki `List` of `SpellCard`
- Method `adjustSpellDuration()` pada Kelas `SummonedCharacter` (`src/main/java/com.aetherwars/fieldcard/SummonedCharacter.java`) memanfaatkan Java API stream filter untuk *filtering* Spell yang memiliki durasi sama dengan nol maka akan dihapus dari list `activeSpells`.

### 3.7. SOLID

- **Single Responsibility Principle:**
  - Kelas `Card` (`src/main/java/com.aetherwars/card/Card.java`) hanya bertugas untuk mengurus nama, deskripsi, tipe, dan `imagepath` dari suatu `Card`, sedangkan `Card` yang dapat di-*summon* ataupun `Card` yang berupa `Spell` dan memiliki mana untuk memakainya diatur oleh Kelas `CharacterCard` (`src/main/java/com.aetherwars/card/CharacterCard.java`) dan Kelas `SpellCard` (`src/main/java/com.aetherwars/card/SpellCard.java`)
  - Kelas `SpellCard` hanya mengurus kebutuhan mana dari `Spell` untuk digunakan, sedangkan tipe-tipe `Spell` dengan efek yang berbeda-beda diatur oleh tiap Kelas `SpellPotion` (`src/main/java/com.aetherwars/card/SpellPotion.java`), `SpellLevel` (`src/main/java/com.aetherwars/card/SpellLevel.java`), `SpellSwap` (`src/main/java/com.aetherwars/card/SpellSwap.java`), `SpellMorph` (`src/main/java/com.aetherwars/card/SpellMorph.java`)
- **Open Closed Principle:**
  - Pada Kelas `SpellCard` (`src/main/java/com.aetherwars/card/Spellcard.java`) dapat melakukan ekstensi (method `printInfo()` dan `getCardType()`) dari Kelas `Card` (`src/main/java/com.aetherwars/card/Card.java`) tanpa merubah abstraksi dari kelas tersebut

- Pada masing-masing SpellPotion, SpellLevel, SpellSwap, dan SpellMorph merupakan ekstensi dari abstraksi kelas SpellCard yang mana pada setiap Spell tersebut terdapat ekstensi method getSpellType()
- **Liskov Substitution Principle:**
  - Pada tiap-tiap Kelas SpellPotion, SpellLevel, SpellSwap, dan SpellMorph memiliki override method getImagePath() yang memiliki return type yang sama seperti parent-nya
  - Pada Kelas SummonedCharacter (src/main/java/com.aetherwars/fieldcard/SummonedCharacter.java) memiliki method generic *addSpell* dan *adjustSpellDuration* dari abstraksi kelas FieldCard yang memiliki tipe parameter yang sama
- **Interface Segregation Principle:**
  - Kelas FieldCard (src/main/java/com.aetherwars/fieldcard/FieldCard.java) mengimplementasi beberapa interface yang sudah di-segregasi menjadi IFieldCard dan ISpellMonitoring
  - Kelas SummonedCharacter (src/main/java/com.aetherwars/fieldcard/SummonedCharacter.java) mengimplementasi beberapa interface yang sudah di-segregasi menjadi ISummoned, ISpellEffect, dan ISummonedBattle
- **Dependency Inversion Principle:**
  - Method generic *addSpell* dari abstract Kelas FieldCard (src/main/java/com.aetherwars/fieldcard/FieldCard.java) menerima tipe-tipe spell inherit dari SpellCard, sehingga tidak perlu membuat method *addSpell* untuk masing-masing tipe spell yang ingin ditambahkan
  - Pada Kelas CardCollection (src/main/java/com.aetherwars/card/CardCollection.java) memiliki atribut List of Card yang mana dapat digunakan untuk menampung kartu-kartu karakter ataupun spell tanpa harus menambah atribut-atribut baru sesuai tipe dari kartu tersebut

### 3.8. Design Pattern

- Digunakan design pattern Factory pada kelas Deck (src/main/java/com.aetherwars/deck/Deck.java), dengan method addSpecificCard(CardCollection x, int j) untuk menghasilkan berbagai card yang akan dimasukkan ke deck.

#### 4. Pembagian Tugas

- 13520005 / Christine Hutabarat
  - Membuat dokumentasi untuk kelas Board
  - Bertanggung jawab untuk tampilan dan penggunaan javafx
- 13520050 / Felicia Sutandijo
  - Membuat kelas Player
  - Membuat mekanisme DRAW phase
  - Membuat mekanisme bertarung
- 13520077 / Rava Naufal Attar
  - Bertanggung jawab untuk Kelas FieldCard
  - Bertanggung jawab untuk Kelas SummonedCharacter beserta Spell yang diberikan dan *effect*-nya
  - Bertanggung jawab untuk SummonedCharacter Battle
- 13520140 / Febryola Kurnia Putri
  - Membuat kelas Card beserta turunannya yaitu CharacterCard dan SpellCard, serta turunan dari SpellCard itu sendiri, yaitu SpellLevel, SpellPotion, SpellSwap, dan SpellMorph
  - Membuat Card Reader
- 13520164 / Hilda Carissa Widelia
  - Membuat dokumentasi untuk kelas Deck
  - Bertanggung jawab untuk tampilan dan penggunaan javafx