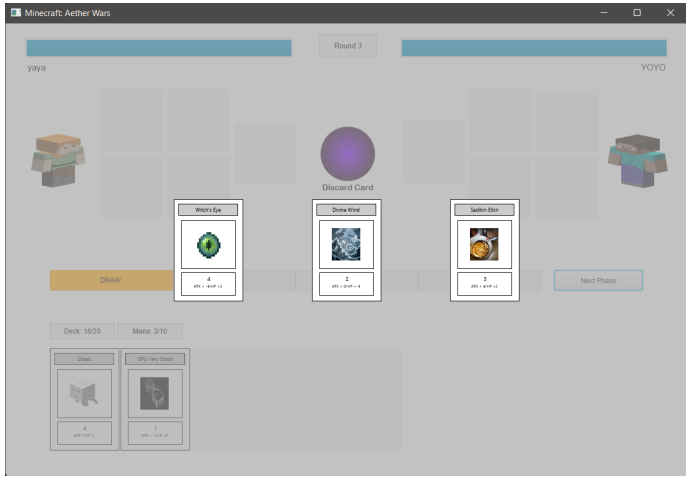
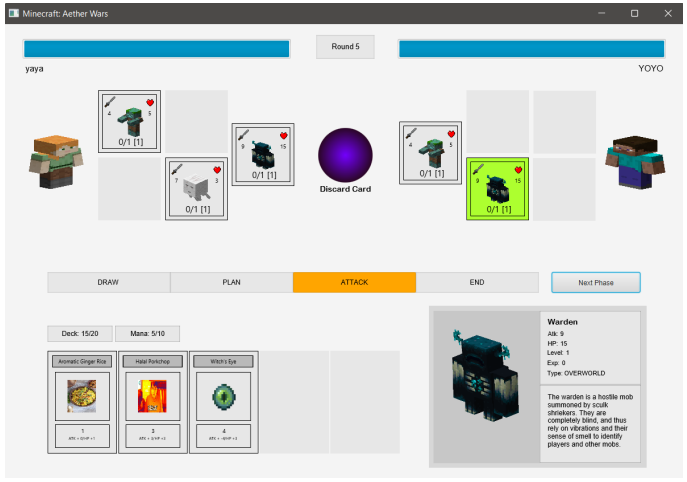
Kelas : 02
Nomor Kelompok : 08
Nama Kelompok : F.A.A.A.K
1. 13520023 / Ahmad Alfani Handoyo
2. 13520044 / Adiyansa Prasetya Wicaksana
3. 13520056 / Fikri Khoiron Fadhila
4. 13520065 / Rayhan Kinan Muhannad
5. 13520083 / Sarah Azka Arief
Asisten Pembimbing : Morgen Sudyanto

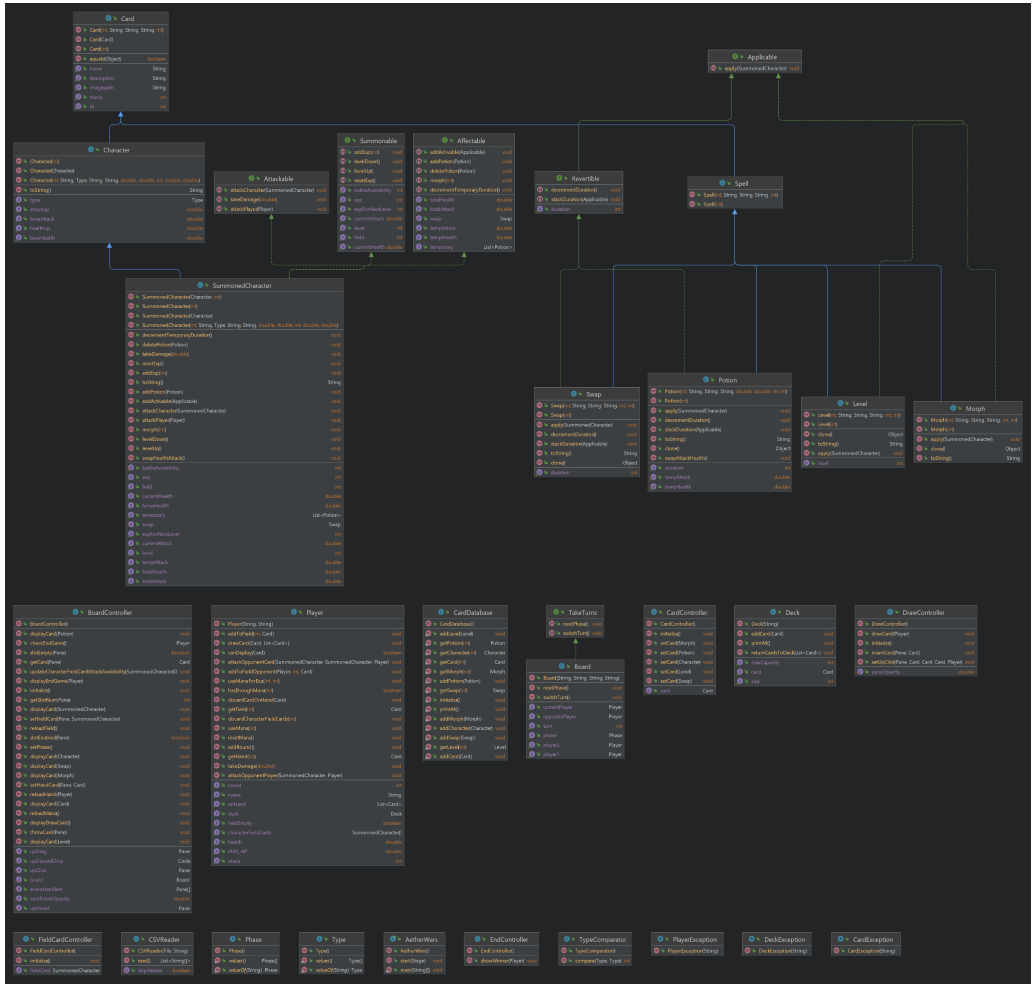
1. Deskripsi Umum Aplikasi

Aplikasi ini merupakan sebuah aplikasi *game* kartu turn based untuk 2 pemain. Pemain bermain secara bergantian pada 1 layar yang sama. Tujuan dari game ini adalah menghabiskan health points (HP) musuh. HP dapat berkurang apabila terkena serangan dari kartu karakter yang diletakkan di board.





2. Konsep OOP


Card
Applicable
Character
Summonable
Affectable
Attackable
Reversible
Spell
SummonedCharacter
Swap
Potion
Level
Morph
BoardController
Player
CardDatabase
TakeTurns
Board
CardController
Deck
DrawController
FieldCardController
CSVReader
Phase
Type
AetherWars
EndController
TypeComparator
PlayerException
DeckException
CardException

2.1. Inheritance

- Kelas SummonedCharacter (src/main/java/com/aetherwars/model/card/character/SummonedCharacter.java) merupakan anak dari kelas Character (src/main/java/com/aetherwars/model/card/character/Character.java), kelas Character merupakan anak dari kelas Card (src/main/java/com/aetherwars/model/card/Card.java)
- Kelas Level (src/main/java/com/aetherwars/model/card/spell/level/Level.java) merupakan anak dari kelas Spell (src/main/java/com/aetherwars/model/card/spell/Spell.java), kelas Spell merupakan anak dari kelas Card (src/main/java/com/aetherwars/model/card/Card.java)
- Kelas Morph (src/main/java/com/aetherwars/model/card/spell/Morph/Morph.java) merupakan anak dari kelas Spell (src/main/java/com/aetherwars/model/card/spell/Spell.java), kelas Spell merupakan anak dari kelas Card (src/main/java/com/aetherwars/model/card/Card.java)
- Kelas Potion (src/main/java/com/aetherwars/model/card/spell/Morph/Morph.java) merupakan anak dari kelas Spell (src/main/java/com/aetherwars/model/card/spell/Spell.java), kelas Spell merupakan anak dari kelas Card (src/main/java/com/aetherwars/model/card/Card.java)
- Kelas Swap (src/main/java/com/aetherwars/model/card/spell/Swap/Swap.java) merupakan anak dari kelas Spell (src/main/java/com/aetherwars/model/card/spell/Spell.java), kelas Spell merupakan anak dari kelas Card (src/main/java/com/aetherwars/model/card/Card.java)

2.2. Composition

- Kelas Deck (src/main/java/com/aetherwars/model/deck/Deck.java) memiliki banyak kelas Card (src/main/java/com/aetherwars/model/card/Card.java).
- Kelas Player (src/main/java/com/aetherwars/model/player/Player.java) memiliki banyak kelas Card (src/main/java/com/aetherwars/model/card/Card.java), tepat lima kelas SummonedCharacter (src/main/java/com/aetherwars/model/card/character/SummonedCharacter.java), tepat satu kelas Deck (src/main/java/com/aetherwars/model/deck/Deck.java).
- Kelas Character (src/main/java/com/aetherwars/model/card/character/Character.java) tepat memiliki satu kelas Type (src/main/java/com/aetherwars/model/card/character/Type.java).

- Kelas SummonedCharacter (src/main/java/com/aetherwars/model/card/character/SummonedCharacter.java) memiliki banyak kelas Potion (src/main/java/com/aetherwars/model/card/spell/potion/Potion.java) dan memiliki tepat satu kelas Swap (src/main/java/com/aetherwars/model/card/spell/swap/Swap.java).
- Kelas Board (src/main/java/com/aetherwars/model/board/Board.java) memiliki tepat dua kelas Player (src/main/java/com/aetherwars/model/player/Player.java) dan tepat satu kelas Phase (src/main/java/com/aetherwars/model/board/Phase.java).

2.3. Interface

- Interface Affectable (src/main/java/com/aetherwars/model/card/character/Affectable.java)
- Interface Attackable (src/main/java/com/aetherwars/model/card/character/Attackable.java)
- Interface Summonable (src/main/java/com/aetherwars/model/card/character/Summonable.java)
- Interface Applicable (src/main/java/com/aetherwars/model/card/spell/Applicable.java)
- Interface Revertible (src/main/java/com/aetherwars/model/card/spell/Revertible.java)

2.4. Method Overriding

- Method getExpForNextLevel() pada kelas SummonedCharacter (src/main/java/com/aetherwars/model/card/character/SummonedCharacter.java)
- Method decrementDuration() pada kelas Potion (src/main/java/com/aetherwars/model/card/spell/potion/Potion.java)
- Method stackDuration() pada kelas Swap (src/main/java/com/aetherwars/model/card/spell/swap/Swap.java)
- Method nextPhase() pada kelas Board (src/main/java/com/aetherwars/model/board/Board.java)
- Method apply() pada kelas Level (src/main/java/com/aetherwars/model/card/spell/level/Level.java)

2.5. Polymorphism

- Method stackDuration pada interface Revertible.java (src/main/java/com/aetherwars/model/card/spell/Revertible.java) menerima parameter dengan kelas Applicable, namun pada method setSwap dan addPotion pada kelas SummonableCharacter.java, kelas ini dipanggil dengan tipe parameter Swap dan Potion.
- Method getCard pada kelas CardDatabase.java (src/main/java/com/aetherwars/model/card/CardDatabase.java) mengembalikan kelas Card, namun pada setiap return value dari method tersebut mengembalikan beberapa kelas turunan Card (Character, Potion, Swap, Morph, dan Level) sesuai dengan parameter method tersebut.
- Method addActivable pada interface Affectable.java (src/main/java/com/aetherwars/model/card/character/Affectable.java) menerima parameter dengan kelas Applicable, tetapi pemain dapat menambahkan beberapa *spell* dengan jenis yang berbeda-beda pada *character* permainan.

2.6. Java API

- Kelas BoardController (src/main/aetherwars/controller/BoardController.java) memanfaatkan Object untuk menyimpan userData pada Pane
- Method compare pada kelas TypComparator (src/main/java/com/aetherwars/model/card/character/TypeComparator.java) mengimplementasikan Comparator<Type>
- Kelas Player (src/main/java/com/aetherwars/model/player/Player.java) memiliki ArrayList of Card
- Kelas Player (src/main/java/com/aetherwars/model/player/Player.java) memiliki List of SummonedCharacter
- Method DrawCard pada kelas Player (src/main/java/com/aetherwars/model/player/Player.java) mengimplementasikan Java.Random

2.7. SOLID

- S: Kelas Board (src/main/java/com/aetherwars/model/board/Board.java) hanya bertugas untuk mengatur turn, round dan phase, sedangkan untuk membuild sebuah deck dan mengaturnya diatur oleh kelas Deck (src/main/java/com/aetherwars/model/deck/Deck.java), sedangkan Kelas yang merupakan anak dari Card dan mengimplemen Affectable, Attackable, Summonable dan beserta perilakunya diatur dalam kelas SummonedCharacter (src/main/java/com/aetherwars/model/card/character/SummonedCharacter.java)

- O: Pada program kami, penerapan open close principle yang sangat terlihat ialah hubungan antara kelas Character dengan SummonedCharacter. Kelas Character terbuka terhadap ekstensi fungsionalitas. Hal ini dapat dilihat pada kelas SummonedCharacter yang menambahkan fungsionalitas-fungsionalitas tambahan pada Character. Namun, kelas SummonedCharacter tidak memodifikasi. kelas Character tersebut karena kelas Character tertutup terhadap modifikasi dari luar. Selain itu berlaku juga untuk kelas Spell dengan Level, Spell dengan Potion, Spell dengan Morph, Spell dengan Swap
- L: Pada program ini, penerapan dari prinsip ini banyak diterapkan pada Super-Class Character dan kelas Sub-Classnya SummonedCharacter. Kedua kelas tersebut menyimpan instantiasi objek-objek dari kelas turunan Card, seperti Character, Spell pada kelas Card. Ketika fungsionalitas-fungsionalitas pada kelas Card untuk bekerja untuk suatu kasus, misalkan untuk mendapatkan nama, imagepath, deskripsi, dll, kita tau bahwa instantiasi kelas-kelas turunan Card tersebut dapat menggantikannya.
- I: kelas SummonableCharacter (src/main/java/com/aetherwars/model/card/character/SummonedCharacter.java) mengimplemen beberapa interface yang sudah di-segregasi menjadi Summonable, Attackable, Affectable
- D : Pada program ini, prinsip ini diterapkan pada beberapa update yang dilakukan pada kelas-kelas Character. Kelas-kelas Character bergantung terhadap interface abstrak dan interface abstrak tersebut diimplementasikan oleh objek-objek model. Hal ini dapat ditemukan misalnya pada kelas Character yang bergantung pada interface Summonable, sementara Summonable diimplementasikan oleh SummonableCharacter. Contoh lain ialah Spell yang bergantung pada interface Applicable, sementara interface Applicable di-implementasikan oleh Level.

2.8. Design Pattern

- Digunakan design pattern Builder dengan Board (src/main/java/com/aetherwars/controller/BoardController.java) sebagai kelas Director dan Concretenya adalah Card (CardController.java) , FieldCard (FieldCard.java), dan End (EndController.java).
- Digunakan design pattern Proxy dengan BoardController sebagai proxy yang menerima input dari Client (user) sebelum merubah databasenya (model).
- Digunakan design pattern Facade pada program ini, kami menggunakan pattern facade pada kelas. Kelas BoardController merupakan kelas yang bertanggungjawab atas *event handlers* dari button dan juga *clickable components* dari *interface* sehingga dapat membalut fungsionalitas GUI lainnya dibalik balutan *interface* yang sederhana.
- Digunakan design pattern Mediator dengan BoardController sebagai mediator yang menghubungkan beberapa Controller lain serta modelnya.
- Digunakan design pattern Decorator pada package Controller untuk menambahkan/merubah views FXML.

- Digunakan design pattern Singleton dengan Board (src/main/java/com/aetherwars/model/board/Board.java) sebagai kelas Board yang diinstansiasi satu kali objek dalam runtime/eksekusi program.
- Digunakan design pattern Publisher-Subscriber pada program ini, kami menggunakan design pattern ini pada objek yang perlu mengirimkan data ke objek lain yang men-subscribe ke dia. Kami menggunakan pattern ini pada kelas Deck yang perlu mengirimkan data 3 card ke GUI untuk ditampilkan saat pengguna memilih kartu.
- Digunakan design pattern observer - Pada program ini, kami menggunakan design pattern Observer pada model sebagai observable (object yang diamati) dan GUI sebagai observer (object yang mengamati). Model yang memakai design pattern ini ialah model yang dapat berganti datanya, dan perlu ditampilkan pada aplikasi GUI ketika datanya berganti. Design pattern ini berguna agar pada aplikasi GUI tidak perlu menyimpan objek yang akan ditampilkannya, tetapi akan mendapatkan informasi jika objek yang ditampilkannya berubah datanya. Penggunaan design pattern ini bermanfaat agar tampilan program akan selalu tersinkronisasi dengan model / data yang disimpan.

3. Bonus Yang dikerjakan

3.1. Bonus yang diusulkan oleh spek

3.1.1. Import Deck

```
public Deck(String deckFilename) throws IOException, URISyntaxException, CardException {
    this.buffer = new ArrayList<>();

    String deckPath = String.format("data/%s", deckFilename);
    File deckCSVFile = new File(Objects.requireNonNull(Deck.class.getResource(deckPath)).toURI());
    CSVReader deckReader = new CSVReader(deckCSVFile,  separator: "\t");
    deckReader.setSkipHeader(true);
    List<String[]> deckRows = deckReader.read();
    for (String[] row : deckRows) {
        int id = Integer.parseInt(row[0]);
        this.addCard(CardDatabase.getCard(id));
    }

    Collections.shuffle(this.buffer);

    this.maxCapacity = this.buffer.size();
}
```

Import deck menggunakan CSVReader dan penggunaannya dengan menambahkan deck .csv pada folder resources/com/aetherwars/model/deck/data .

3.1.2. Spell Baru

Penambahan tipe spell baru yaitu level untuk merubah level (menambahkan atau mengurangi) dengan implementasi Applicable dan Cloneable.

```java
@Override
public void apply(SummonedCharacter c) throws CardException {
    if (this.level >= 0) {
        int N = this.level;
        while (N > 0) {
            c.resetExp();
            c.levelUp();
            N--;
        }
    } else {
        int N = this.level;
        while (N < 0) {
            c.resetExp();
            c.levelDown();
            N++;
        }
    }
}
```

- **Bottle O’ Enchanting dan Bottle O' Disenchanting**

```
id  name    description imagepath   level   mana
401 Bottle O' Enchanting    TING TING TING  Bottle O' Enchanting.png    1   5
402 Bottle O' Disenchanting TUNG TUNG TUNG  Bottle O' Disenchanting.png -1  5
```

Spell level yang terdapat pada csv.

4. Pembagian Tugas

- 13520023 / Ahmad Alfani Handoyo
    - Bertanggung jawab atas user interface, menghubungkan dengan model, dan testing program
- 13520044 / Adiyansa Prasetya Wicaksana
    - Bertanggung jawab atas user interface, menghubungkan dengan model, dan testing program
- 13520056 / Fikri Khoiron Fadhila
    - Bertanggung jawab atas modul player, modul board dan phase dan testing modul
- 13520065 / Rayhan Kinan Muhannad
    - Bertanggung jawab atas modul card, modul deck, serta testing modul
- 13520083 / Sarah Azka Arief
    - Bertanggung jawab atas user interface, menghubungkan dengan model, dan testing program