



# **TUBES OOP KEL-6**

STEPHANIE HUTAGALUNG  
18220001

FATIH DARIELMA GAIKTA  
18220009

TANIA JUNARDI CHUA  
18220067

RACHITA CARONICA JONUR  
18220091

# GAME PLAY



Game ini dapat dimainkan oleh 2 orang (PvP). Pada awal permainan, mereka mendapatkan masing-masing 5 monster (dapat di-setting sendiri, lihat bagian Konfigurasi!), tetapi hanya satu monster saja yang akan digunakan pada giliran pertama. Setelah itu, akan berlanjut pada fase pertarungan. Saat giliran salah satu pemain, mereka dapat memilih antara dua opsi, yaitu menggunakan move yang dimiliki oleh monster atau mengganti monster, di mana hanya satu opsi saja yang dapat dijalankan pada satu giliran.

Urutan move dijalankan sesuai priority dari move dan speed yang dimiliki monster. Monster juga mungkin tidak dapat mengeksekusi move apabila terkena dampak status condition sleep atau paralyze. Untuk sleep, monster tidak dapat bergerak sesuai dengan jumlah giliran yang di-random, sedangkan paralyze memungkinkan monster tidak dapat bergerak selama satu giliran sebesar 25%.

Setelah setiap move dijalankan, akan dilakukan perhitungan damage sesuai dengan tipe move (normal / special) yang mengenai sasaran. Lalu, akan dicek pula status dari monster lawan, di mana apabila terkena burn atau poison, maka akan terkena damage lagi.

Jika monster yang sedang digunakan mati karena damage yang dilakukan oleh lawan, maka monster tersebut tidak bisa dipakai lagi dan player harus menggantinya dengan monster lain yang masih hidup. Permainan ini akan berlanjut terus sampai ada salah satu pemain yang sudah tidak memiliki monster untuk bertarung (semua monster mati). Maka secara otomatis, lawannya dinyatakan sebagai pemenang dalam permainan ini.

# USER MANUAL

1. Pindah ke directory src (lokasi Main.java).

```
\Tubes00P>cd src
```

2. Compile program dengan command berikut.

```
\Tubes00P\src>javac Main.java
```

3. Jalankan program Main dengan command berikut.

```
\Tubes00P\src>java Main
```

```
[1] Start Game
[2] About
[0] Exit
-----
>>>
```

4. Tekan 1 untuk memulai game.

```
>>> 1
Monster Saku starts in 3...2...1...
=====
WELCOME TO MONSTER SAKU
=====
Loading configuration...
Generating monsters...
Fetching data...
DONE
```

Tekan 2 untuk melihat trademark.

```
>>> 2
<C> Monsaku Inc.
```

Tekan 0 untuk keluar dari game.

```
>>> 0
Exit..
```

# USER MANUAL

5. Jika memulai game, masukkan nama kedua player.

```
<PLAYER 1>'S Name: Blue
<PLAYER 2>'S Name: Red
```

6. Kemudian, akan ditampilkan daftar monster dari masing-masing player dan list pilihan instruksi yang dapat dipilih monster.

```
-[0] Help
-[1] Move
-[2] Switch Monster
-[3] View Monster Info
-[4] View Game Info
-[-] Exit
```

[0] Help

Perintah ini akan menampilkan list pilihan instruksi

[1] Move

Perintah ini akan menampilkan move dari monster yang dapat dipilih pemain. Setelah memilih move, giliran pemain berikutnya untuk memilih move

[2] Switch Monster

Perintah ini akan menampilkan pilihan monster yang dapat dipilih pemain. Memilih monster hanya dapat dilakukan sekali sebelum memilih move.

[3] View Monster Info

Perintah ini akan menampilkan informasi mengenai monster yang sedang bermain.

[4] View Game Info

Perintah ini akan menampilkan informasi mengenai list monster dari tiap pemain dan monster yang sedang bertarung.

[-] Exit

Keluar dari game.

7. Setelah kedua pemain memilih move, akan ditampilkan HP dari kedua monster setiap eksekusi move dilakukan.



# ISTILAH



Istilah	Keterangan
Nama	Nama dari Monster
Health Point	Nyawa yang dimiliki monster
Attack	Besar serangan monster yang diberikan saat menggunakan normal move
Defense	Besar pertahanan monster untuk menahan damage dari normal move
Sp. Attack (Special Attack)	Besar serangan monster yang diberikan saat menggunakan special move
Sp. Defense (Special Defense)	Besar pertahanan monster untuk menahan damage dari special move
Speed	Kecepatan monster dalam menyerang
Elmt (Element type)	Elemen yang dimiliki monster
Move	Skill yang bisa digunakan monster dalam pertarungan

# DEVELOPMENT PROCESS

Proses pengembangan dimulai dengan kami membuat class **Monster**, **Stats**, dan **ElementType**. Awalnya, Stats dibuat class terpisah. Namun, setelah melewati beberapa pertimbangan kami memutuskan untuk menjadikan stats sebagai inner class dalam class monster. Hal ini dilakukan untuk memperjelas hubungan “has a” antara Monster dan Stats karena Stats akan ada jika ada Monster. Selanjutnya, kami membuat bagian ElementType. Awalnya, kami membuat ElementType dengan mengimplementasikan “list<string>”, tapi kemudian kami membuatnya dengan menggunakan enum.

Monster, Stats, dan ElementType sudah, selanjutnya kami membuat **Move**. Pada Move, kami berpikir untuk membuat subclass StatusMove, DefaultMove, NormalMove, dan SpecialMove. Semuanya berjalan lancar sampai ditemukan masalah saat membuat **Controller**, yaitu saat membuat skema serang menyerang. Lebih tepatnya, kami kesulitan dalam mengimplementasikan method loadMonster dan loadMove. Untungnya, beberapa hari setelahnya kami mempelajari reflection di kelas. Jadi, kami menggunakan reflection untuk membuat dynamic instancing pada loadMove dan loadMonster. Akibatnya, subclass dari Move (StatusMove, NormalMove, dan SpecialMove) kami satukan ke dalam file Move kecuali **DefaultMove** karena cara eksekusinya berbeda dengan move yang lain.

# DEVELOPMENT PROCESS

Terkhir, kami membuat **Main** untuk menjalankan game. Untuk alur permainannya sendiri kami buat kurang lebih mirip dengan sebuah game pertarungan antarninja, yaitu Ninja Saga. Berikut alur permainannya.

Player 1 memilih move -> player 1 menyerang player 2 -> ganti pemain ke player 2 -> player 2 memilih move -> player 2 menyerang player 1 -> ganti pemain ke player 1 -> repeat

Namun, setelah memahami ulang spesifikasi, akhirnya kami melakukan sedikit modifikasi, yaitu pada bagian pemilihan move. Monster dari player 1 dan 2 akan memilih move terlebih dahulu kemudian dari kedua move tersebut akan ditentukan move yang mana yang akan dieksekusi. Berikut alur fixed-nya.

Player 1 memilih move/switch -> Player 2 memilih move/switch -> ditentukan move mana yang akan dieksekusi -> repeat

# DEVELOPMENT PROCESS

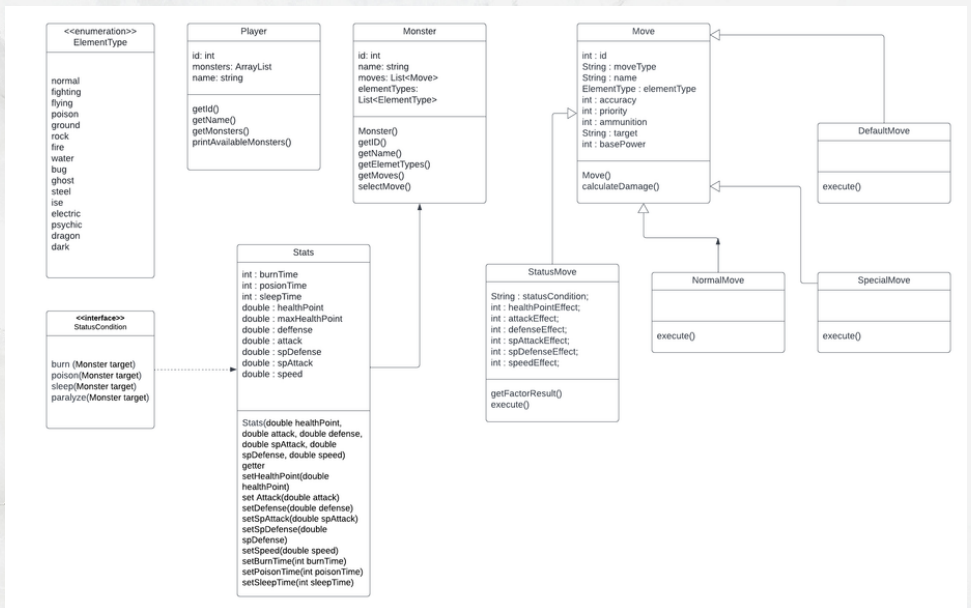
Berikut merupakan timeline pengerjaan kami

Timeline	Deskripsi
21 Maret 2022	Meet membahas konsep tugas besar serta pembagian tugas
26 Maret 2022	Progress report pengerjaan tugas
27 Maret - 2 April 2022	Proses pengembangan program serta pembuatan file main
2 April - 10 April 2022	Membuat booklet serta proses debugging



# DEVELOPMENT PROCESS

Berikut merupakan coretan kami untuk membantu menggambarkan class dan method yang perlu dibuat



**NOTE:** Coretan class diagram di atas menggambarkan proses brainstorming saat PERTAMA KALI mendesain game. Beberapa penggunaan class, atribut, dan method pada diagram tersebut mungkin tidak sesuai dengan yang ada di source code karena telah terjadi beberapa modifikasi.

# DEVELOPMENT PROCESS

Berikut merupakan salah satu meet yang kami lakukan untuk proses brainstorming dan debugging. Selain meet, kami juga menggunakan line untuk berdiskusi

The screenshot displays a Zoom meeting interface. The main window shows a UML class diagram titled "OOP Pokemon Lucidchart". The diagram includes the following classes and their attributes:

- Player**:
  - id: int
  - monsters: ArrayList
  - name: string
  - Methods: `getId()`, `getName()`, `getMonsters()`, `printAvailableMonsters()`
- Monster**:
  - id: int
  - name: string
  - moves: List<Move>
  - elementTypes: List<ElementType>
  - Methods: `Monster()`, `getId()`, `getName()`, `getElementTypes()`, `getMoves()`, `selectMove()`
- Move**:
  - id: int
  - String: moveType
  - String: name
  - ElementType: elementType
  - int: accuracy
  - int: priority
  - int: ammunition
  - String: target
  - int: basePower
  - Methods: `Move()`, `calculateDamage()`
- Stats**:
  - int: burnTime
- StatusMove**: (Association class between Monster and Move)

The diagram also shows associations between Player and Monster, and between Monster and Move. A "StatusMove" association class is connected to the Monster-Move association.

On the right side of the Zoom window, a list of participants is visible:

- 18220009 Fatih Darielma Gaizta
- 18220091 Rachita Caronica Jonur (highlighted)
- 18220001 Stephanie Hutagalung
- You

# KONFIGURASI

Orang yang ingin memainkan game ini dapat mengubah data konfigurasi pada ketiga file berikut dalam direktori **config**.

**MonsterPool.csv**

Id	name	elementTypes	baseStats	moves
----	------	--------------	-----------	-------

**MovePool.csv**

Id	move Type	name	element Type	accuracy	priority	ammunition	target	effect
----	-----------	------	--------------	----------	----------	------------	--------	--------

**ElementTypesEffectivity.csv**

source	target	effectivity
--------	--------	-------------

**NOTE:** Perhatikan contoh konfigurasi yang sudah dibawakan dalam source code karena dalam suatu field, mungkin terdiri dari beberapa nilai yang dipisahkan oleh tanda koma!

Perhatikan contoh pada field berikut:

- baseStats: health point, attack, defense, sp. attack, sp.defense, speed
- effect: basePower, atau efek (statusCondition) dengan faktor dari status buff (healthPoint, attack, defense, spAttack, spDefense, speed)

# CONSTRAINT

- 1.Orang yang ingin memainkan permainan ini dapat melakukan konfigurasi sendiri, seperti menciptakan monster baru dengan menambahkannya di MonsterPool.csv dan move baru di MovePool.csv (NOTE: Pastikan masukan konfigurasi harus sesuai format yang ada pada bagian Konfigurasi).
- 2.Setiap monster yang ada harus memiliki move yang berbeda dengan monster lain. Misal, jika monster A sudah memiliki move X, monster B tidak boleh memiliki move X lagi. Hal tersebut dilarang karena dilakukan instancing sendiri-sendiri pada setiap move sehingga jika monster B memiliki move X, maka move X yang dimiliki monster A dan monster B refer ke object yang sama.
- 3.Apabila monster pertama yang mengeksekusi move mati (akibat burned, poisoned, atau mati karena default move), monster kedua akan tetap mengeksekusi move kepada monster yang sudah mati tersebut.
- 4.Player yang memilih switch tidak akan melakukan move dengan monster yang baru di-switch. Setiap switch akan dilakukan terlebih dahulu sebelum monter lawan mengeksekusi move-nya.
- 5.Jika ingin menambah monster dan move dalam konfigurasi, pastikan hal-hal berikut terpenuhi:
  - a.Tidak ada monster yang memiliki move yang sama dengan monster lain.
  - b.Monster memiliki move dengan element type yang sesuai dengan element type yang dikuasai oleh monster (Jika tidak, move tidak akan bisa dieksekusi).



# PEMBAGIAN TUGAS



- Membuat Move.java
- Membantu proses pembuatan Monster.java
- Membantu proses pembuatan booklet

**STEPHANIE HUTAGALUNG - 18220001**



- Membuat Main.java
- Membuat Controller.java
- Membuat Player.java
- Membuat Configuration.java
- Debugging

**FATIH DARIELMA GAIZTA - 18220009**



- Membuat ElementType.java
- Membuat StatusCondition.java
- Membantu proses pembuatan Monster.java
- Membantu proses pembuatan booklet

**TANIA JUNARDI CHUA - 18220067**



- Membuat stats pada monster.java
- Membantu membuat monster.java
- Membantu proses pembuatan booklet

**RACHITA CARONICA JONUR - 18220091**

THANK YOU  
THANK YOU  
**THANK YOU**  
THANK YOU  
THANK YOU

