

# Spesifikasi Tugas Besar 1

## IF2210 Pemrograman Berorientasi Objek

Revision I 15 Maret 2021

### Willy Wangky and the Engimon Factory

Setelah Willy Wangky membangun taman bermainnya yang bernama Willy Wangky's World, ia menjadi orang yang sangat sukses. Untuk meningkatkan kekayaannya Willy Wangky memiliki ide untuk membuat wahana baru di taman bermainnya, yakni Engimon Factory. Di wahana tersebut Willy Wangky akan menyebarkan engimon-engimon liar, para pemain dapat melawan dan menangkap engimon liar tersebut menggunakan engimon mereka sendiri. Willy Wangky meminta bantuan anda untuk membuat sebuah game yang dapat mensimulasikan wahana Engimon Factory milik Willy Wangky, agar ia dapat membangun wahananya dengan lebih baik.

### Fungsionalitas

#### 1. Engimon

- a. Engimon memiliki bermacam-macam spesies
- b. Setiap spesies memiliki beberapa informasi wajib:
  - i. Name
  - ii. Parent Names & Species
  - iii. Skill/kemampuan(maksimal 4)
  - iv. Elements (bisa lebih dari 1)
  - v. Level
  - vi. Experience
  - vii. Cumulative Experience
  - viii. ~~Atribut lain seperti Nature dan Ability (berbeda dengan skill)~~
- c. Ketika experience mencapai nilai kelipatan 100, maka engimon akan level up
- d. Ketika engimon telah mencapai nilai maksimum cumulative experience tertentu, maka engimon akan mati (dihapus dari program)
- e. **Minimal dibuat 1 jenis spesies untuk setiap elemen dan untuk setiap kombinasi elemen (detail ada di poin 5.e)**

#### 2. Skill

- a. Skill memiliki berbagai macam
- b. Setiap skill memiliki informasi wajib:
  - i. Nilai Numerik Base Power
  - ii. Mastery Level
  - iii. **Elements yang dapat mempelajari skill tersebut (bisa majemuk)**
- c. Setiap spesies engimon memiliki satu skill bawaan yang unik

### 3. Player:

- a. Player memiliki beberapa *commands* yang bisa dipilih pemain pada suatu waktu:
  - i. Bergerak ke satu petak ke kiri, kanan, atas, atau bawah
  - ii. Menampilkan list engimon yang dimiliki
  - iii. Menampilkan data lengkap suatu engimon (setiap atribut kelas)
    1. Juga harus menampilkan nama *parent* beserta *spesies* mereka
  - iv. Mengecek dan mengganti *active* engimon
  - v. Menampilkan *list* skill item yang dimiliki
  - vi. Menggunakan skill item pada suatu engimon
  - vii. Melaksanakan *breeding* antara 2 engimon
  - viii. Melakukan *battle* dengan suatu engimon yang berada didekatnya (*adjacent tiles*) yaitu satu petak di sebelah kiri, kanan, atas, dan bawah.
    1. Ketika ingin melakukan *battle*, tampilkan detail engimon lawan ke layar

Untuk *flow* UI dan tampilan UI dibebaskan asalkan dapat mendukung delapan kebutuhan diatas. Beberapa *command* akan dijelaskan lebih detail di poin-poin selanjutnya

- b. Player memiliki ***inventory*** yang bertugas untuk menyimpan informasi mengenai benda-benda yang dimiliki player. Adapun ketentuan mengenai *inventory* adalah sebagai berikut:
  - i. Merupakan class *generic* yang minimal menampung objek-objek hasil instansiasi 2 buah kelas berikut:
    1. Engimon
      - a. Merupakan kelas yang digunakan untuk mewakili **satu** engimon yang dimiliki oleh *player*.
      - b. Boleh terdapat engimon dengan nama yang sama dalam *inventory*.
      - c. Gunakan kelas *engimon* yang digunakan untuk mewakili engimon lain di *factory*.
    2. Skill Item
      - a. Ekuivalen dengan TM (*Technical Machine*) dan HM (*Hidden Machine*) di game Engimon.
      - b. Memiliki atribut jumlah. Dengan demikian, **tidak boleh** terdapat skill dengan nama yang **berbeda sama (duplikat)** dalam *inventory*.
      - c. **Anda dapat menggunakan kelas *skill* yang sama dengan kelas *skill* pada poin spek 2.**
      - d. **Memiliki method *learn*. Engimon hanya dapat mempelajari *skill* yang kompatibel dengan elemennya.**
  - ii. PERHATIAN: satu instansi *inventory* tidak harus mampu menampung keduanya sekaligus

- iii. Dapat menggunakan *Container* dari *Standard Template Library* (STL) untuk mengimplementasikan *inventory*. Pilihlah *container* yang paling sesuai menurut anda.
- iv. Terdapat atribut **max capacity**. Jumlah engimon + skill items dalam *inventory* tidak boleh melebihi kapasitas maksimum tersebut.
- c. Player memiliki **satu active engimon** dengan ketentuan sebagai berikut:
  - i. Engimon tersebut mengikuti gerakan *player* dari belakang.
  - ii. Active engimon tidak diperbolehkan **keluar dari map** atau **menabrak objek lain**
  - iii. Jika aturan poin ii dilanggar, lemparlah sebuah kasus *exception*. Kemudian dalam *catch* blok berikanlah pesan ke layar (isi pesan bebas) dan lakukanlah reposisi agar posisi engimon tidak melanggar poin ii (mekanisme pemilihan posisi baru dibebaskan).
  - iv. Dapat **berinteraksi** dengan *player*:
    - 1. Ketika *player* berinteraksi dengan *active* engimon, maka mereka akan mengeluarkan teks unik berdasarkan nama spesies engimonnya. Teks yang dikeluarkan sesuai dengan template:

[<NAMA\_ENGIMON>]: <teks bebas>

<NAMA\_ENGIMON> = nama engimon yang diinteraksikan

<teks bebas> = slogan bebas dan unik per spesies engimon
  - v. *Player* dapat melakukan **switch out active engimon**.
    - 1. *Player* dapat menggantikan **active engimon** dengan **engimon lain dari inventory**.

#### 4. Battle

Battle antara 2 engimon hanya berupa *status comparison*:

- a. Hitung power:  $\text{level} * \text{element advantage} + \text{SUM}(\text{every skill's base power} * \text{Mastery Level})$
- b. Element advantage chart sebagai berikut:



	Fire	Water	Electric	Ground	Ice
Fire	1	0	1	0.5	2
Water	2	1	0	1	1
Electric	1	2	1	0	1.5
Ground	1.5	1	2	1	0
Ice	0	1	0.5	2	1

Contoh : Engimon player dengan elemen fire battle dengan engimon water, nilai element advantage untuk perhitungan power engimon player adalah 0.5, sedangkan untuk engimon yang dilawan adalah 2.

- c. Jika engimon player kalah, engimon player akan mati. Kemudian, player dapat memilih engimon active baru lalu memiliki opsi untung battle lagi.
- d. Jika engimon player menang, player akan mendapatkan engimon yang menjadi lawan jika inventory masih cukup. Active engimon juga akan menerima *experience points* dengan besaran yang bebas (boleh statik atau menggunakan rumus tertentu). **Player juga akan mendapatkan random Skill Item yang kompatibel dengan elemen engimon musuh.**

- e. Jika battle antara multiple element engimon :

Contoh :

Engimon player = fire/water

Engimon wild = water/electric

Nilai element advantage diambil dari kombinasi perhitungan terbesar dengan nilai maksimal sama dengan 2.

Perhitungan element advantage untuk engimon player adalah :

- Fire > water = 0
- Fire > electric = **1 (Diambil yang paling besar)**
- Water > water = **1 (Diambil yang paling besar)**
- Water > electric = 0

Perhitungan element advantage untuk engimon player adalah **2** :

- Water > fire = **2 (Diambil yang paling besar)**
- Water > water = 1
- Electric > fire = 1
- Electric > water = **2 (Diambil yang paling besar)**

## 5. Breeding

- a. Untuk start *breeding*, kedua *parent* harus memiliki level  $\geq 30$ .
- b. Setelah melakukan breeding, level kedua *parent* berkurang sebesar 30
- c. Player dapat memberikan nama kepada anak hasil *breeding*
- d. Inherit Skill
  - i. Skill yang dimiliki oleh anak engimon akan mengambil skill dari kedua engimon induknya. Prioritas pengambilan skill adalah (diurutkan dari aturan yang paling signifikan ke bawah):
    1. Skill yang memiliki *mastery level* tertinggi
    2. Jika ada *mastery level* yang sama, skill dari Parent A diutamakan
    3. Pilihlah skill pada urutan slot yang lebih terdepan
    4. Ambil skill sebanyak mungkin hingga mencapai maksimal 4 skill
  - ii. Jika skill yang dipilih dimiliki oleh kedua parent:
    1. Kedua parent memiliki mastery level yang sama, maka skill yang di-inherit memiliki mastery level bernilai  $\text{mastery level parent A} + 1$
    2. Kedua parent memiliki mastery level yang berbeda, maka skill yang di-inherit memiliki mastery level bernilai  $\max(\text{mastery level parent A}, \text{mastery level parent B})$
- e. *Resulting Child Species & Element*

Kasus yang mungkin:

- i. Jika elemen kedua *parent* sama, anak akan memiliki elemen yang sama dengan kedua *parent*. Spesies anak dipilih dari *parent A* atau *parent B* secara bebas (boleh random atau aturan spesifik tertentu)
- ii. Jika elemen kedua *parent* berbeda maka anak akan memiliki elemen dan spesies dari elemen yang memiliki *element advantage* yang lebih tinggi.
- iii. Jika elemen kedua *parent* berbeda dan kedua elemen memiliki *element advantage* yang sama, maka anak akan memiliki spesies berbeda dari kedua *parent* yang memiliki kedua elemen *parent* (boleh dipilih random atau *hardcoded*).

Sesuai dengan tabel *element advantage* diatas, **kombinasi dual element engimon yang mungkin:**

1. Fire x Electric -> Fire/Electric
2. Water x Ice -> Water/Ice
3. Water x Ground -> Water/Ground

**TIPS:** Waspada dengan kasus ***diamond problem!***

- iv. **[BONUS]** Breeding yang melibatkan engimon dengan lebih dari satu elemen tidak wajib untuk dikerjakan (anda dapat menampilkan *exception* saja). Jika ingin dikerjakan speknya:
- i. Engimon anak kemungkinan hanya memiliki 1 elemen atau dua elemen saja (tidak lebih)
  - ii. Elemen yang mewakilkan seorang parent hanya perlu satu saja. Misal: jika parent berelemen fire/electric, anda dapat menganggap parent sebagai elemen fire atau elemen electric saja dalam *breeding* tersebut

Hati-hati kasus ***operator overloading resolution!***

## 6. Peta

Berikut merupakan contoh ilustrasi peta Engimon Factory (warna pada ilustrasi hanya untuk mempermudah ilustrasi)

-	-	-	-	-	-	o	o	o	o	o	o		Keterangan:  W/w: Water engimon l/i : Ice engimon F/f: Fire engimon G/g: Ground engimon E/e: Electric engimon L/l: Fire/Electric engimon S/s: Water/Ice engimon N.n: Water/Ground engimon  P: Player  - : Grassland o : Sea
-	-	-	-	-	-	o	o	o	o	o	o		
-	-	F	-	-	-	o	o	l	o	i	o		
-	G	-	-	-	-	o	o	W	o	o	o		
-	-	-	-	-	-	o	o	o	o	w	o		
-	-	g	-	-	-	o	o	o	o	o	o		
-	-	-	-	-	-	-	-	-	-	-	-		
-	-	-	-	P	-	-	-	f	-	-	-		
-	-	-	-	-	-	-	-	-	-	-	-		
-	-	-	-	E	-	-	-	e	-	-	-		

Peta Engimon Factory memiliki beberapa aturan sebagai berikut:

- a. Engimon dengan level minimal X (bebas), dikeluarkan dalam peta dengan huruf besar.
- b. Engimon dengan level kurang dari X (ditentukan pemain), dikeluarkan dalam peta dengan huruf kecil.

- c. Huruf dari engimon yang ditampilkan tergantung elemennya (W/w : water engimon, F/f: Fire engimon, dst).
- d. Peta berbentuk tile dengan dibagi menjadi 2 lingkungan, *sea* dan *grassland*. Engimon liar hanya dapat berjalan di area dengan elemen yang sesuai (Fire, Ground, dan Electric engimon **hanya** dapat berjalan di *grassland*, Water dan Ice engimon **hanya** dapat berjalan di *sea*).
- e. Player dapat menempati tile apapun
- f. Engimon liar bergerak secara random setiap turn, tapi hanya dapat berjalan dalam tile yang bisa dijalani saja sesuai elemennya (aturan d).
- g. Dalam sebuah peta ada batas maksimum engimon yang di-*spawn* (jumlah maksimum engimon yang dapat di-*spawn* ditentukan pemain).
- h. Engimon di-*spawn* secara random:
  - i. Algoritma untuk spawning engimon liar dibebaskan
    - 1. Boleh menggunakan spawn point sebagai titik spawning
    - 2. Boleh melakukan spawn secara periodik (periode bebas)
    - 3. Spesies dan atribut engimon yang di-*spawn* boleh dibuat secara **[BONUS]** *pure random* atau statik
  - i. Engimon bergerak secara random di peta namun hanya dalam wilayah yang bisa dilewatinya saja sesuai dengan poin d
  - j. Load peta dilakukan melalui file eksternal

## Panduan Pengerjaan

Program yang Anda buat harus menggunakan pemrograman berorientasi objek dan menggunakan bahasa C++. Desain UI dan program dibebaskan, namun harus mencakup 5 konsep pemrograman orientasi objek seperti:

- Inheritance & Polymorphism
- Method/Operator Overloading
- Template & Generic Classes
- Exception
- C++ *Standard Template Library*

Selain penilaian dari tes fungsional, akan dilakukan penilaian dari segi desain program. Untuk mendapatkan nilai yang maksimal dari segi desain, terapkanlah prinsip:

- **Hindari menggunakan 5 konsep OOP diatas hanya karena diwajibkan saja! Gunakanlah konsep OOP yang sesuai dengan kasusnya!**
- DRY (Don't Repeat Yourself), tidak memiliki kode duplikat
- Memiliki struktur kelas yang mudah dipahami
- Dekomposisi yang baik dan implementasi yang tidak terlalu kompleks (sebuah method tidak terlalu panjang)

- Mengikuti prinsip SOL di [SOLID](#)

[BONUS]:

1. Dual element breeding
2. Pure Randomly generated wild engimons

Selain dari 2 poin bonus yang ditetapkan diatas, anda diperbolehkan untuk bebas berkreasi menambahkan fitur baru asal tidak mengubah fitur wajib. Setiap kreasi yang anda tambahkan akan menjadi **pertimbangan** untuk **nilai bonus**.

## Panduan Laporan dan Dokumentasi

Sebagai programmer yang baik, Anda dilatih tidak hanya untuk membuat kode, tapi juga membuat dokumentasi dan tes.

Anda wajib membuat sebuah laporan dengan format **pdf** yang **wajib** berisi:

1. Class diagram program anda (lengkap dengan header method dan atribut kelas secara singkat)
2. Dimana anda menggunakan 5 konsep OOP (selain yang diwajibkan) beserta alasannya
3. Pada spek diatas, ada beberapa poin yang detail implementasinya dibebaskan ke anda. Tuliskanlah teknik implementasi yang ada pilih beserta alasannya!
4. Bonus yang dikerjakan
5. Pembagian tugas