

Spesifikasi Tugas Besar 2

IF2210 Pemrograman Berorientasi Objek

Willy Wangky and the Engimon Factory in JAVA

Setelah Engimon Factory berhasil dibuat oleh anda, Willy Wangky berpikir bahwa bahasa C++ terlalu mudah dan tidak mampu memadai cita rasa sang Willy Wangky. Dengan hati yang tersinggung, beliau ingin anda mengerjakan ulang tugas besar anda sekali lagi dalam bahasa **Java**.

I. Modifikasi pada Spesifikasi Tugas Besar I

Spesifikasi *engimon factory* secara garis besar mirip dengan spesifikasi pada tugas besar I, namun ada beberapa perubahan spesifikasi yang diberi indikator *highlight* berwarna hijau

1. Engimon

- a. Engimon memiliki bermacam-macam spesies
- b. Suatu spesies memiliki tipe elemen yang konsisten (misal spesies 'Ratatamon' hanya bertipe Ground saja, tidak mungkin ada yang bertipe Water)
- c. Setiap spesies memiliki beberapa informasi wajib:
 - i. Name
 - ii. Life (*by default*, Active Engimon memiliki 3 lives, Wild Engimon memiliki 1 life)
 - iii. Parent Names & Species
 - iv. Skill/kemampuan(maksimal 4)
 - v. Elements (bisa lebih dari 1)
 - vi. Level
 - vii. Experience
 - viii. Cumulative Experience
- d. Ketika experience mencapai nilai kelipatan 100, maka engimon akan level up
- e. Ketika engimon telah mencapai nilai maksimum cumulative experience tertentu (bebas nilai maksnya), maka engimon akan mati (dihapus dari program)
- f. **Minimal dibuat 1 jenis spesies untuk setiap elemen dan untuk setiap kombinasi elemen (detail ada di poin 5.e)**
- g. Setiap X turn, *engimon liar* akan mendapat 100 exp atau X exp yang cukup untuk naik level (sesuai dengan ketentuan anda) secara otomatis! Pastikan

bahwa engimon liar juga bisa meninggal setelah melewati batas *cumulative exp* maksimum!

2. Skill

- a. Skill memiliki berbagai macam (banyak jenis skill)
- b. Setiap skill memiliki informasi wajib:
 - i. Nilai Numerik Base Power
 - ii. Mastery Level (MAKSIMAL Lvl. 3)
 - 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. Menampilkan *list* dari *commands* yang tersedia
 - ii. Bergerak ke satu petak ke kiri, kanan, atas, atau bawah
 - iii. Menampilkan list engimon yang dimiliki
 - iv. Menampilkan data lengkap suatu engimon di *inventory* (setiap atribut kelas dan setiap atribut skillnya)
 1. Juga harus menampilkan nama *parent* beserta *spesies* mereka
 - v. Mengecek dan mengganti *active* engimon
 - vi. Menampilkan *list* skill item yang dimiliki beserta informasi mengenai atribut skill tersebut (base power dan elemen yang bisa learn skill tersebut)
 - vii. Menggunakan skill item pada suatu engimon
 - viii. Melaksanakan *breeding* antara 2 engimon
 - ix. 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

- x. Membuang X *amount* dari suatu *skill item* atau melepaskan engimon *inventory*
 - xi. Mengganti nama dari suatu *engimon* yang ada di *inventory*
 - xii. *Save game*
- 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*.
- d. Daftar engimon di *inventory* memiliki aturan:
 - i. *Grouped by* elementnya
 - ii. Dalam 1 *group*, *sorted by level* dari yang tertinggi dahulu

Contoh:

1. EngimonA/Fire/Lv.30
2. EngimonB/Fire/Lv.6
3. EngimonC/Water/Lv.15
4. EngimonD/Ground/Lv.17

2. Skill Item

- a. (TRIVIA) Mekanisme mirip dengan **one-time use skill scroll**
- 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. *Skill* yang di-*learn* dari *skill item* memiliki *mastery level* 1
- e. **Engimon tidak bisa *learn skill* yang sudah dimiliki oleh engimon tersebut**
- f. Jika engimon sudah penuh *skill slot*-nya, maka *player* akan diberi opsi untuk *replace* salah satu *skill* dari engimon tersebut
- g. Daftar *skill item* di *inventory* memiliki aturan:
 - i. *Sorted by* base power dari yang tertinggi dahulu

- 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.
 1. Jadi objek *inventory* untuk engimon dan *skill item* boleh dipisah tapi *max capacity* dari kedua *inventory* dihitung sebagai satu kesatuan
- c. *Player* memiliki **satu active engimon** dengan ketentuan sebagai berikut:

- i. Engimon tersebut mengikuti gerakan *player* dari belakang. Active Engimon boleh melewati tile apapun.
- ii. Active engimon tidak diperbolehkan **keluar dari map** atau **menabrak objek/engimon 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**.

d. *Player* akan kalah (permainan selesai) jika sudah kehabisan *active engimon* di *inventory*

4. Battle

Battle antara 2 engimon hanya berupa *power level comparison*. Engimon yang memiliki *power level* lebih tinggi adalah pemenang dari battle. *Jika sama besar maka engimon player akan menang*. Detail lebih lanjut:

- a. *Player* tidak bisa melakukan *battle* ketika sedang tidak ada *engimon* yang aktif
- b. Hitung power: $\text{level} * \text{element advantage} + \text{SUM}(\text{every skill's base power} * \text{Mastery Level})$
- c. Sebelum *battle*, tampilkanlah *status lengkap engimon musuh* dan *total power level* dari kedua engimon yang sedang bertarung ke layar. Setelah itu, berilah opsi bagi *player* untuk *proceed ke battle* atau *cancel battle*
- d. 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, sedangkan untuk engimon yang dilawan adalah 2.

- Jika engimon player kalah, engimon player akan kehilangan 1 life. Jika life dari engimon mencapai 0, engimon akan mati. Kemudian player dapat memilih *command* selanjutnya seperti biasa.
- 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 Skill Item yang berada skill di slot pertama dari engimon musuh.
- 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 1 :

- 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 wild adalah 2 :

1. Water > fire = **2 (Diambil yang paling besar)**
2. Water > water = 1
3. Electric > fire = 1
4. Electric > water = **2 (Diambil yang paling besar)**

5. Breeding

- a. Untuk start *breeding*, kedua *parent* harus memiliki level ≥ 4 .
- b. Setelah melakukan breeding, level kedua *parent* berkurang sebesar 3
- c. Player dapat memberikan nama kepada anak hasil *breeding*
- d. Inherit Skill
 - i. 1 skill slot diutamakan untuk skill unik spesies anak (kalau misalnya hasil inherit skill dari parent menghasilkan skill tersebut, utamakan pengambilan skill yang *inherited* dari *parents* terlebih dahulu)
 - ii. Karena pada proses *breeding skill* di-inherit (tidak dipelajari) maka bisa **bypass constraint atribut elemen yang mampu mempelajari skill tersebut** (2.b.iii). *Breeding skill* juga dapat mewariskan skill unik dari setiap spesies.
 - iii. 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
 - iv. 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 $\text{mastery level parent A} + 1$. **Mastery level tidak bisa melebihi level 3!**
 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 speknnya:
 - i. Engimon anak kemungkinan hanya memiliki 1 elemen atau dua elemen saja (tidak lebih)
 - ii. Elemen yang mewakili 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

Peta Engimon Factory memiliki beberapa aturan sebagai berikut:

- a. Peta dibagi menjadi 3 lingkungan!
 - i. **Mountains:** Hanya Fire engimon saja yang mampu melintasinya
 - ii. **Sea:** Hanya Water engimon saja yang mampu melintasinya
 - iii. **Grassland:** Hanya Ground & Electric engimon saja yang mampu melintasinya
 - iv. **Tundra:** Hanya Ice engimon saja yang mampu melintasinya

- b. Player dapat menempati tile apapun
- c. Engimon liar bergerak secara random **setiap X turn** (X bernilai bebas), tapi hanya dapat berjalan dalam tile yang bisa dijalani saja sesuai elemennya (aturan d)
- d. Dalam sebuah peta ada batas maksimum engimon yang di-*spawn* (jumlah maksimum engimon yang dapat di-*spawn* ditentukan pemain).
- e. Engimon di-*spawn* dengan cara:
 - i. Algoritma untuk posisi spawning engimon liar dibebaskan
 - 1. Boleh menggunakan *spawn points* sebagai titik spawning
 - 2. WAJIB melakukan spawn secara periodik (periode bebas)
 - ii. Algoritma atribut *spawned engimon*:
 - 1. *Spawned engimon* harus memiliki level \geq level tertinggi dari *engimon* di inventory *player*
 - 2. Spesies dan setiap atribut engimon yang di-*spawn* boleh dibuat secara statik atau **[BONUS]** *pure random*
- f. Engimon bergerak secara random di peta namun hanya dalam wilayah yang bisa dilewatinya saja sesuai dengan poin d
- g. Load peta dilakukan melalui file eksternal
- h. Kasus tepi: pada kasus player dan *wild* engimon bergerak bersamaan ke petak yang sama, maka yang harus reposisi (*exception handling*) adalah *wild* engimon

II. Spesifikasi Tambahan untuk Tugas Besar II

1. Graphical User Interface

- a. Buatlah *Graphical User Interface* untuk aplikasi *Engimon Factory* diperbolehkan untuk menggunakan library / tools untuk membuat user interface (misal : Swing, JavaFX, dll). *Game engine* juga diperbolehkan seperti LWJGL. Desain user interface dibebaskan kepada setiap kelompok, namun buatlah user interface semenarik mungkin. Anda harus tetap memahami bagian-bagian user interface tersebut meskipun menggunakan tools.
- b. Anda wajib memberikan *graphical representation* untuk:
 - i. Entitas player di map
 - ii. Setiap spesies engimon
 - iii. Engimon player diberi indikasi grafik spesial yang membedakan mereka dari engimon liar (bebas, bisa sprite yang berbeda atau efek grafik lainnya)
 - iv. Engimon yang memiliki power level yang lebih tinggi dan lebih rendah dari engimon aktif akan diberi indikasi grafik yang berbeda (contoh: size lebih besar/icon spesial)

- v. Engimon akan memiliki indikasi grafik khusus menurut elemennya (contoh: aura/glow yang warnanya sesuai atau icon spesial)
- vi. Icon skill. Untuk setiap mastery level diberikan representasi grafik yang berbeda
- vii. Setiap jenis petak di map
- viii. *Layout inventory* player yang mengandung engimon dan skill item

2. Save & Load Functionality

- a. Sebelum memulai permainan, player diberikan opsi untuk *load save files* yang dimilikinya
- b. Player dapat melakukan save di tengah permainan
- c. Anda harus dapat menyimpan *state* dari *game* secara lengkap dengan memanfaatkan file eksternal
- d. Format penyimpanan dibebaskan

III. BONUS

1. Multi-threading (real-time gameplay) [BONUS 30 POIN]

- a. Implementasikan fitur *multi-threading* dengan menggunakan Java API untuk membuat *game* anda menjadi *real-time*!
- b. Dengan demikian tidak ada sistem turn lagi. Wild engimon dapat bergerak dan *spawn* tanpa harus menunggu gerakan dari player.
- c. Harus tetap menjaga aturan main map yaitu tidak boleh ada dua objek yang menempati posisi yang sama di map!

2. Unit Testing [BONUS 10 POIN]

- a. Unit Testing mencakup tes yang terisolasi untuk salah satu modul/kelas tertentu untuk menjamin fungsionalitas dasar sudah berjalan benar. Manfaat terbesar adalah dengan adanya *unit testing* dengan *coverage* yang cukup, anda dapat melakukan modifikasi kode/*refactoring* secara lebih praktis (tidak perlu uji coba fungsionalitas manual) dan aman (sudah terjamin benar).

- b. Anda dapat mengimplementasikan *unit testing* untuk setiap kelas yang anda buat. Minimal 50% dari kelas yang anda buat terjangkau oleh *unit testing* untuk dihitung sebagai bonus.
- c. Anda boleh menggunakan *library* apapun untuk implementasinya

Panduan Pengerjaan

Program yang Anda buat harus menggunakan pemrograman berorientasi objek dan menggunakan bahasa Java. Anda wajib membuat **Graphical User Interface (GUI)** dan harus mencakup 8 konsep pemrograman orientasi objek seperti:

- Polymorphism
- Inheritance/Composition/Aggregation
- Abstract Class
- Interface
- Generic Type & Wildcards
- Exception Handling
- Java Collection
 - Anda wajib memanfaatkan Java Collection pada tugas besar ini. Gunakanlah minimal tiga kelas collections yang berbeda dalam melakukan implementasi. **Anda boleh menambahkan fitur baru untuk memenuhi kuota 3 kelas collections.**
 - Misal: player command history with stack
- Java API
 - Anda wajib Melakukan extend/implement terhadap kelas/interface yang terdapat pada Java API (misal: Comparable, Iterable, Threading, dsb) pada rancangan anda. Lakukan *extend/implement* dari minimal dua kelas pada Java API.

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

- **Hindari menggunakan 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](#)

Selain dari 3 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 8 konsep OOP (selain yang diwajibkan) beserta alasannya
3. Bonus yang dikerjakan
4. *Library external* yang anda gunakan beserta alasannya
5. Pembagian tugas