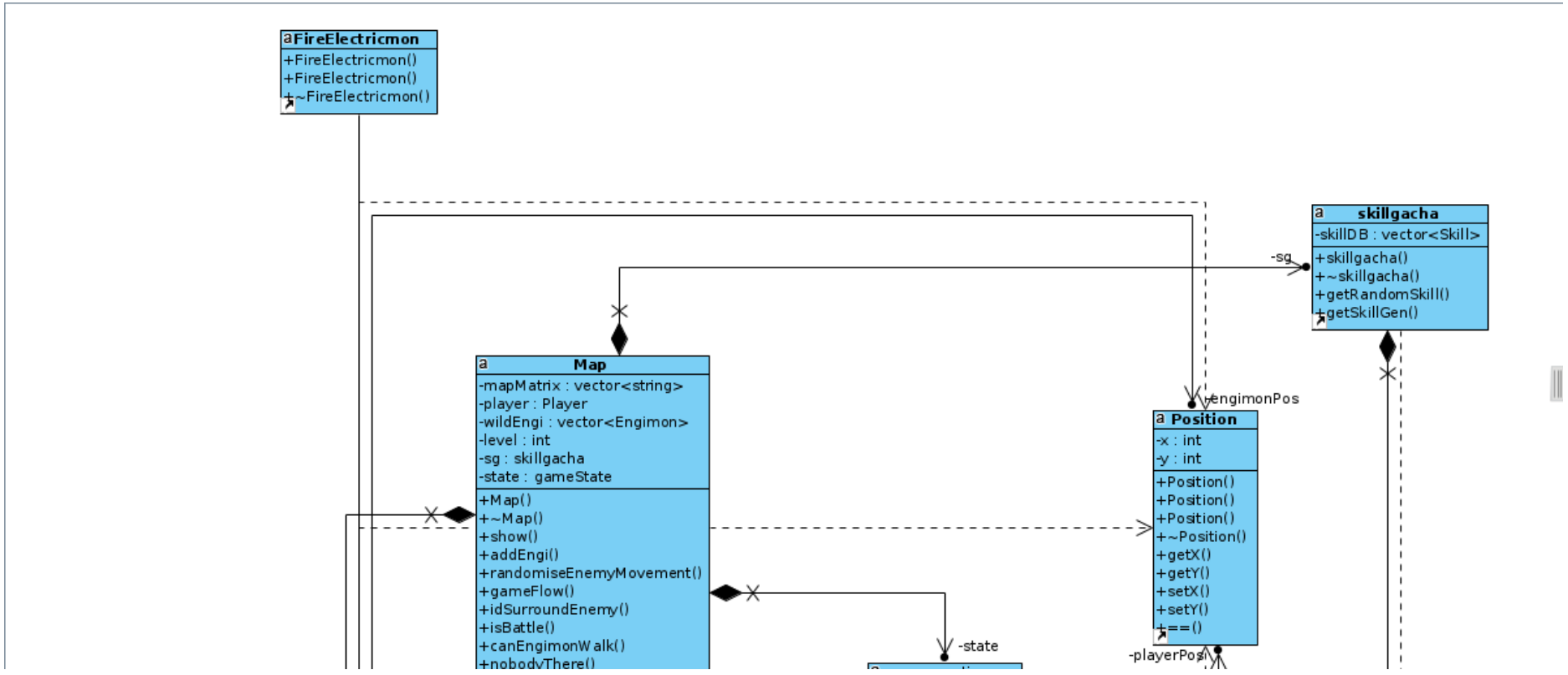
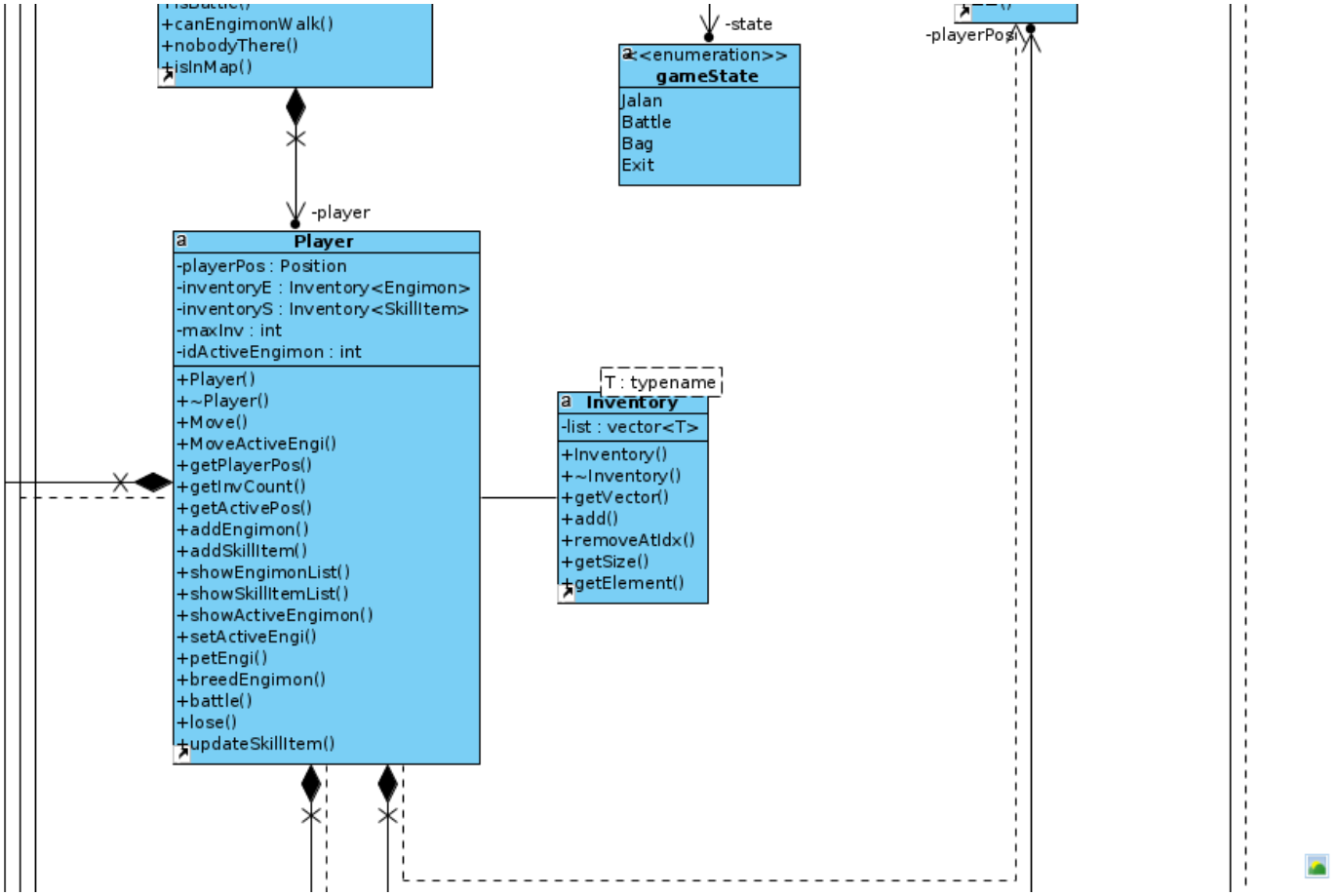
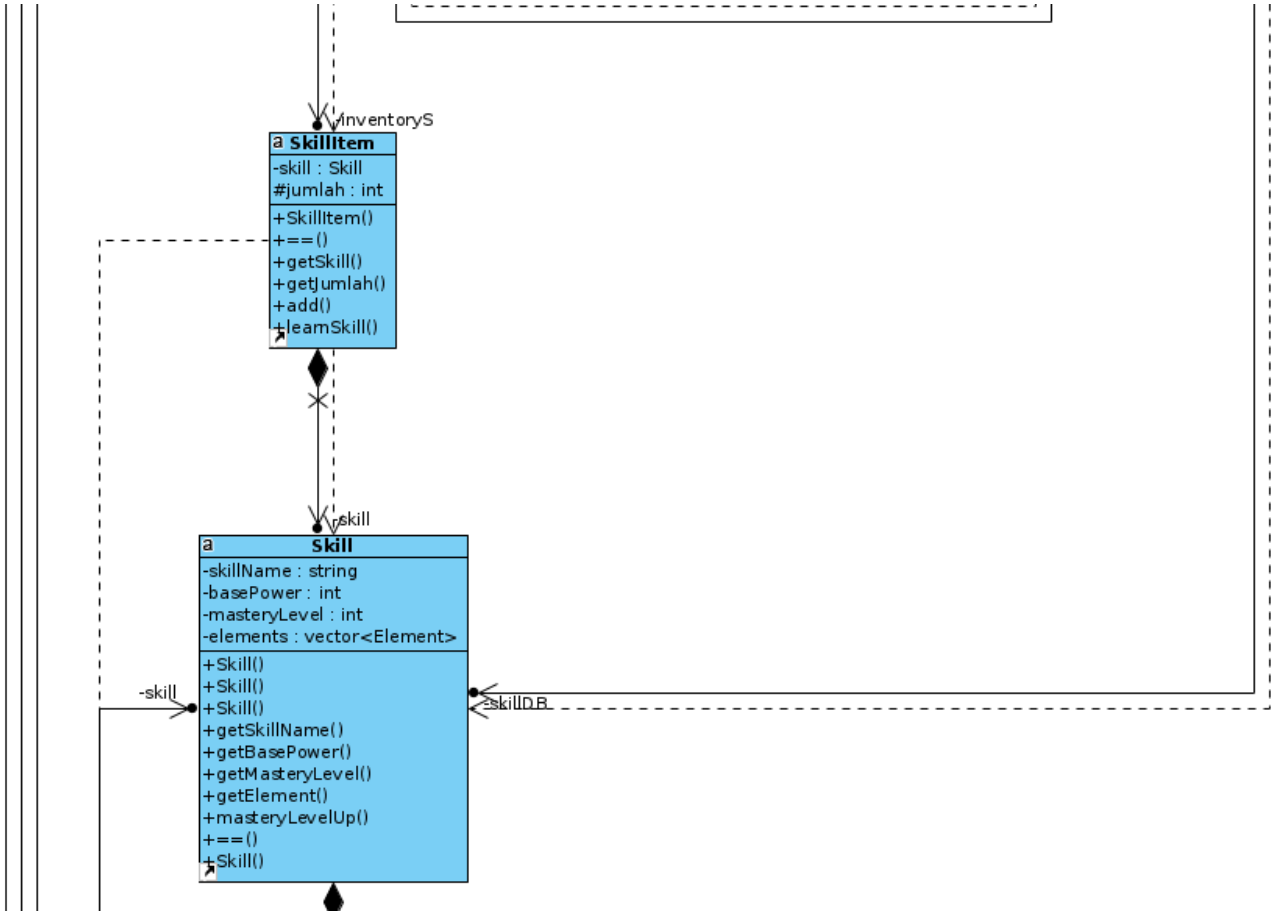
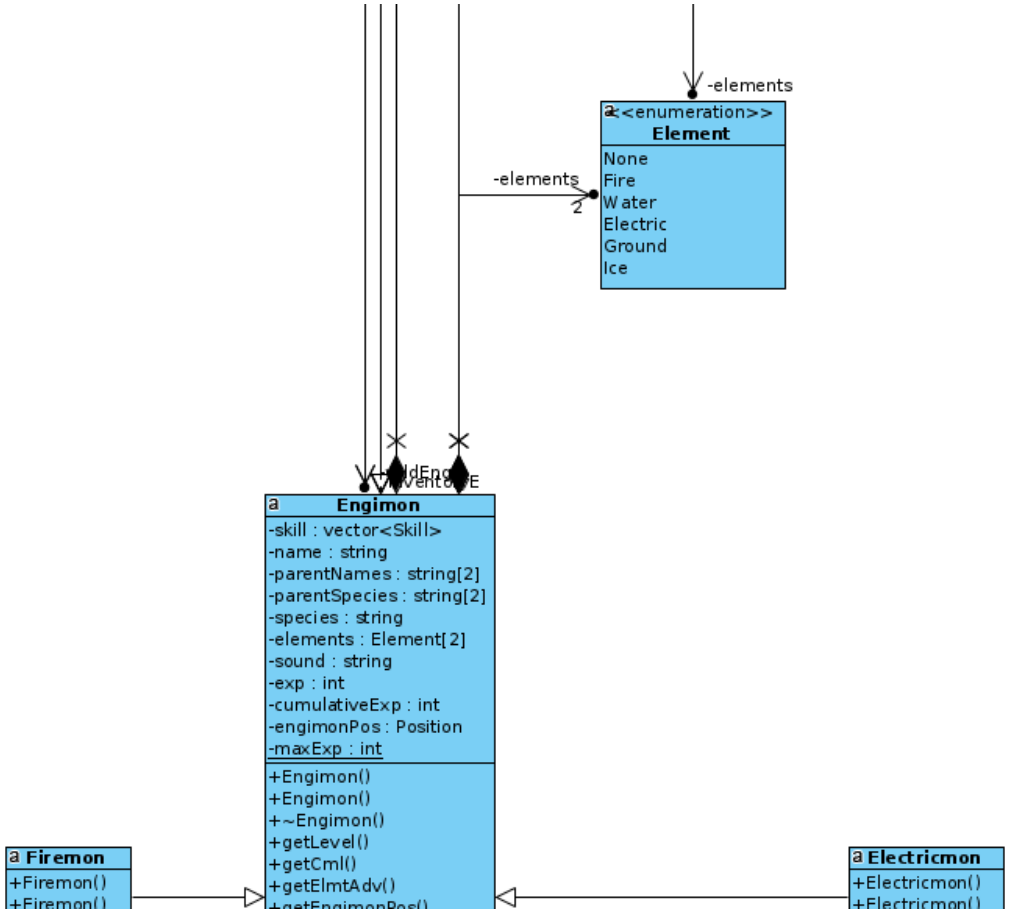
Kelas : 03
Nama Kelompok : JamberNih
1. 13519142 / Muhammad Akram Al Bari
2. 13519149 / Syihabuddin Yahya Muhammad
3. 13519150 / Imam Nurul Hukmi
4. 13519151 / Azmi Muhammad Syazwana
5. 13519158 / Muhammad Fikri Naufal
Asisten Pembimbing : Naufal Aditya Dirgandhavi

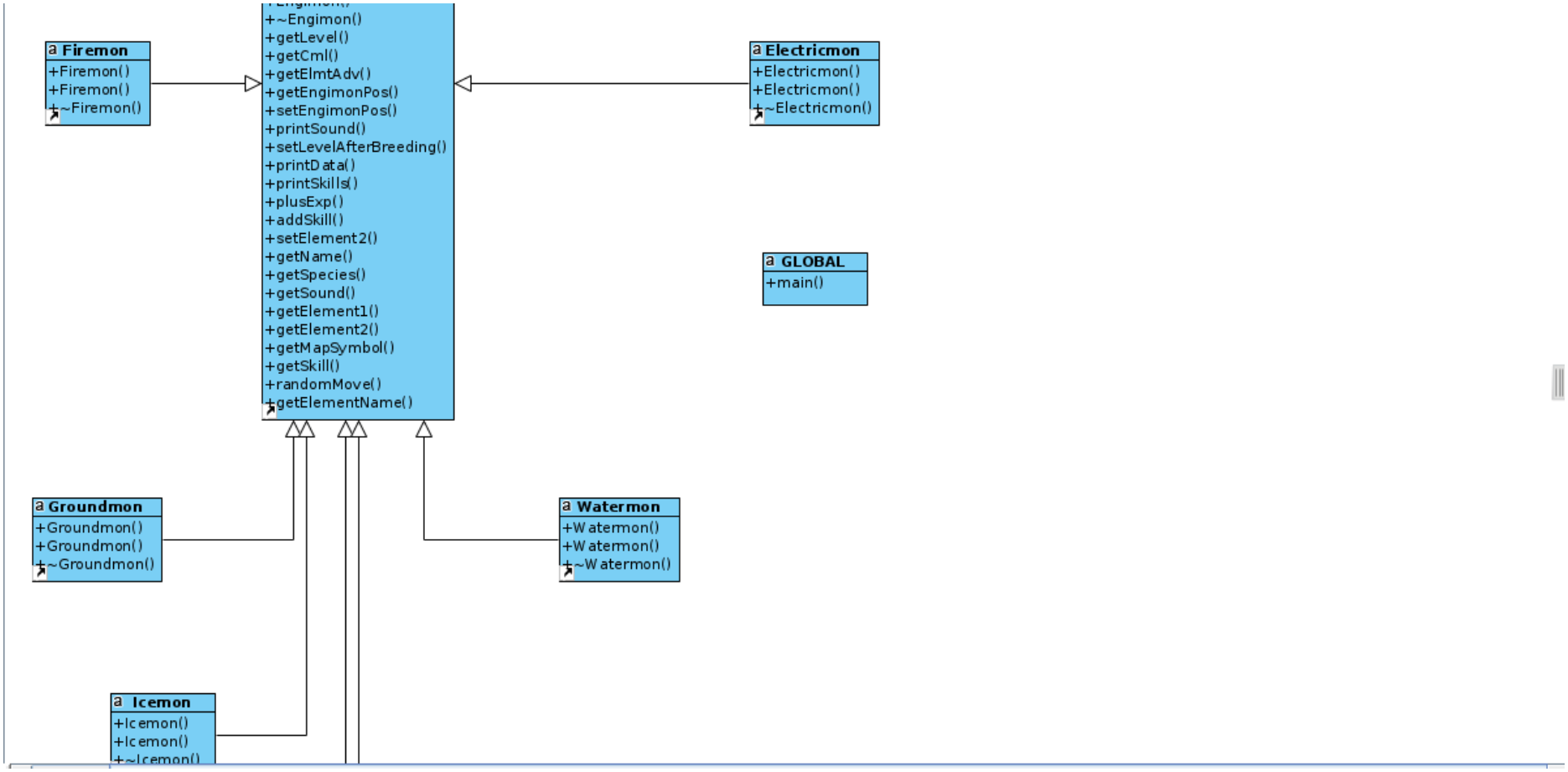
1. Diagram Kelas

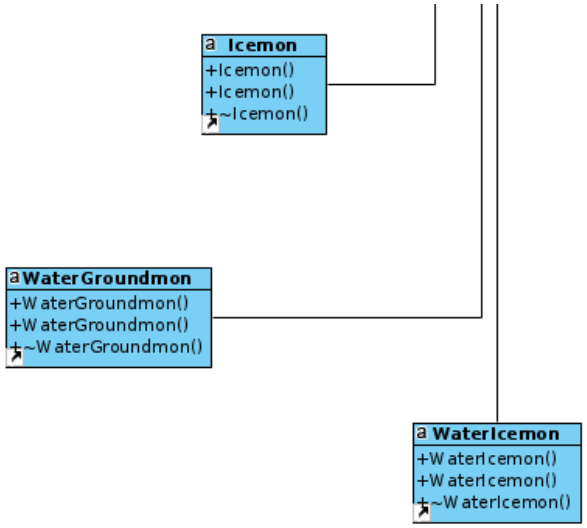



+canEngimonWalk()
+nobodyThere()
+isInMap()
-state
<<enumeration>>
gameState
Jalan
Battle
Bag
Exit
-playerPos
-player
Player
-playerPos : Position
-inventoryE : Inventory<Engimon>
-inventoryS : Inventory<SkillItem>
-maxInv : int
-idActiveEngimon : int
+Player()
+~Player()
+Move()
+MoveActiveEngi()
+getPlayerPos()
+getInvCount()
+getActivePos()
+addEngimon()
+addSkillItem()
+showEngimonList()
+showSkillItemList()
+showActiveEngimon()
+setActiveEngi()
+petEngi()
+breedEngimon()
+battle()
+lose()
+updateSkillItem()
T : typename
Inventory
-list : vector<T>
+Inventory()
+~Inventory()
+getVector()
+add()
+removeAtIdx()
+getSize()
+getElement()


inventoryS
SkillItem
-skill : Skill
#jumlah : int
+SkillItem()
+==()
+getSkill()
+getJumlah()
+add()
+learnSkill()
-skill
Skill
-skillName : string
-basePower : int
-masteryLevel : int
-elements : vector<Element>
+Skill()
+Skill()
+Skill()
+getSkillName()
+getBasePower()
+getMasteryLevel()
+getElement()
+masteryLevelUp()
+==()
+Skill()
-skill
-skillDB


-elements
<<enumeration>>
Element
None
Fire
Water
Electric
Ground
Ice
-elements
2
Engimon
-skill : vector<Skill>
-name : string
-parentNames : string[2]
-parentSpecies : string[2]
-species : string
-elements : Element[2]
-sound : string
-exp : int
-cumulativeExp : int
-engimonPos : Position
-maxExp : int
+Engimon()
+Engimon()
+~Engimon()
+getLevel()
+getCml()
+getElmtAdv()
Firemon
+Firemon()
Electricmon
+Electricmon()


Firemon
+Firemon()
+Firemon()
+~Firemon()
+~Engimon()
+getLevel()
+getCml()
+getElmtAdv()
+getEngimonPos()
+setEngimonPos()
+printSound()
+setLevelAfterBreeding()
+printData()
+printSkills()
+plusExp()
+addSkill()
+setElement2()
+getName()
+getSpecies()
+getSound()
+getElement1()
+getElement2()
+getMapSymbol()
+getSkill()
+randomMove()
+getElementName()
Electricmon
+Electricmon()
+Electricmon()
+~Electricmon()
GLOBAL
+main()
Groundmon
+Groundmon()
+Groundmon()
+~Groundmon()
Watermon
+Watermon()
+Watermon()
+~Watermon()
Icemon
+Icemon()
+Icemon()



Secara garis besar, kelas yang terdapat pada program ini adalah:

- Engimon
- Skill
- Player
- Inventory
- Map
- Position

Kemudian, beberapa kelas tersebut memiliki beberapa subclass. Engimon misalnya, ia memiliki subclass diantaranya Firemon, Watermon, dan Icemon. Yang mana, setiap subclass dari Engimon mewakili spesies dari tiap-tiap Engimon. Setiap class Engimon memiliki properti yang berkaitan dengan informasi Engimon tersebut seperti name, parentNames, parentSpecies, elements, skill dan exp. Engimon sengaja didesain memiliki subclass yang merepresentasikan spesies tiap Engimon dikarenakan hal ini menyebabkan deklarasi Engimon menjadi lebih eksplisit, serta mengurangi kebutuhan parameter ketika menginisialisasi object Engimon.

Skill merupakan class tersendiri yang menjadi salah satu bagian dari property Engimon. Kemudian, terdapat pula class yang memiliki relasi erat dengan Skill, yakni class SkillItem. Pada dasarnya SkillItem hanya menambahkan property berupa jumlah dan juga tetap menyimpan Skill sebagai salah satu propertynya. Desain class seperti ini memudahkan dalam mengimplementasikan bagian-bagian program yang lain, dikarenakan cukup seringnya kebutuhan untuk menggunakan class Skill dan SkillItem secara terpisah

Player merupakan class yang mendefinisikan pemain pada program ini. Di dalam class Player, terdapat property untuk menandai inventory, engimon yang dimiliki player, serta posisi player saat ini. Sebagian besar perintah yang dapat dijalankan oleh pengguna program ini terdefinisi sebagai method yang ada di class Player.

Inventory merupakan class yang hanya memiliki satu property bertipe Vector. Vector digunakan sebagai representasi dari property Inventory dikarenakan pada realisasi program ini, nilai maksimum dari suatu inventory disimpan dalam class Player. Penggunaan Vector memudahkan dalam mengintegrasikan kedua property ini dikarenakan sifat Vector yang merupakan array dinamis.

Map adalah class yang dapat dianggap sebagai class yang menjadi *container* dari class-class inti yang ada pada program ini. Di dalam Map, terdefinisi property dengan type Player, Engimon, dan Vector yang merupakan type-type inti pada program ini. Meskipun Player memiliki class tersendiri, namun object Player yang digunakan sepanjang program ini disimpan di dalam Map. Desain yang seperti ini membuat proses enkapsulasi dan juga keberjalanan program menjadi lebih rapi. Terakhir, terdapat pula class Position yang merupakan representasi dari posisi Player di Map, yang direpresentasikan dengan integer yang bertindak sebagai koordinat X dan Y.

2. Penerapan Konsep OOP

2.1. Inheritance & Polymorphism

Pada program yang dibuat, konsep Inheritance & polymorphism diterapkan pada objek/kelas engimon dan skill. Inheritance diterapkan pada kelas engimon untuk membuat setiap spesies unik. Setiap spesies kami buat menjadi kelas baru dengan konstruktor sesuai template spesies tersebut. Kami menggunakan hal ini untuk mempermudah proses generating engimon baru. Selain itu jika di kemudian hari suatu engimon dapat memiliki kemampuan khusus yang bergantung pada spesiesnya, dapat dijadikan atribut dari kelas spesies tersebut. Hal yang sama terjadi pada kelas skill, setiap skill unik kami simpan sebagai kelasnya sendiri karena alasan yang sama juga.

Cuplikan kode:

Atribut kelas parent Engimon:

```
class Engimon
{
    friend bool battle(Engimon e1, Engimon e2);
    friend Engimon breeding (Engimon& A, Engimon& B);
protected:
    vector<Skill> skill;
private:
    string name;
    string parentNames[2];
    string parentSpecies[2];
    string species;
    Element elements[2];
    string sound;
    static int maxExp;
    int exp;
    int cumulativeExp;
    Position engimonPos;
```

Contoh kelas child:

```
class Firemon : public Engimon{
public:
    Firemon(string name, string plname, string p2name, string plspc, string p2spc, int exp, int px, int py);
    Firemon(int, int, int);
    ~Firemon();
};

class Watermon : public Engimon{
public:
    Watermon(string name, string plname, string p2name, string plspc, string p2spc, int exp, int px, int py);
    Watermon(int, int, int);
    ~Watermon();
};

class Electricmon : public Engimon{
public:
    Electricmon(string name, string plname, string p2name, string plspc, string p2spc, int exp, int px, int py);
    Electricmon(int, int, int);
    ~Electricmon();
};
```

2.2. Method/Operator Overloading

Terdapat beberapa operator Overloading yang diimplementasikan pada program ini. Operator yang paling sering di overload adalah operator ==(equals). Hal ini disebabkan karena penilaian sama tidaknya suatu objek seringkali perlu memperhatikan atribut suatu objek sehingga dilakukan pada operator overloading pada operator ini. Operator overloading yang diimplementasikan adalah:

1. Operator == pada kelas skill

```
bool Skill::operator==(const Skill s) const{
    return skillName == s.skillName;
}
```

2. Operator == pada kelas skillItem

```
bool SkillItem::operator==(const SkillItem s){
    return skill==s.skill;
}
```

3. Operator == pada kelas position

```
bool Position::operator==(const Position &p){
    return (this->x == p.x && this->y == p.y);
}
```

4.

2.3. Template & Generic Classes

Konsep Template & generic Class digunakan pada implementasi inventory player. Hal ini terjadi karena inventory player dapat berisi 2 objek yang berbeda yaitu engimon dan skill item. Meskipun terdapat objek yang berbeda, penanganannya sebagai isi suatu inventory tidak lah berbeda sehingga dapat digunakan kelas generik Inventory untuk mewakili inventory kedua objek tersebut.

Cuplikan kode:

```
template <typename T>
class Inventory
{
private:
    vector<T> list;
public:
    Inventory();
    ~Inventory();
    vector<T>& getVector();
    void add(T);
    void removeAtIdx(int n);
    int getSize();
};
```

2.4. Exception

Konsep Exception digunakan pada implementasi dari metode yang mengatur letak active engimon pada map. Umumnya active engimon akan berada di bawah(selatan) player, namun ketika pada posisi tersebut terdapat sesuatu yang menghalangi, seperti engimon liar atau patas map, maka akan dilempar suatu exception untuk menandakan bahwa active engimon tidak akan berada pada tempat yang seharusnya, dan dilakukan reposisi sementara engimon tersebut. Exception tersebut dapat ditangkap oleh program utama ataupun metode lain, sehingga dapat diambil langkah yang tepat untuk menangani kasus tersebut.

Cuplikan kode:

```
void Player::MoveActiveEngi(){ // cek obstacle belom jadi
    int x = playerPos.getX(), y = playerPos.getY(), xl = x, yl = y-1;
    bool outidx = false, obstacle = false;
    if(xl < 0 || xl > 14 || yl < 0 || yl > 14/* || obstacle()*/){ // bawah gabisa
        outidx = true;
        yl = y+1;
    }

 if(outidx){
     throw "bambang mau kemana sih\n";
 }
```

2.5. C++ Standard Template Library

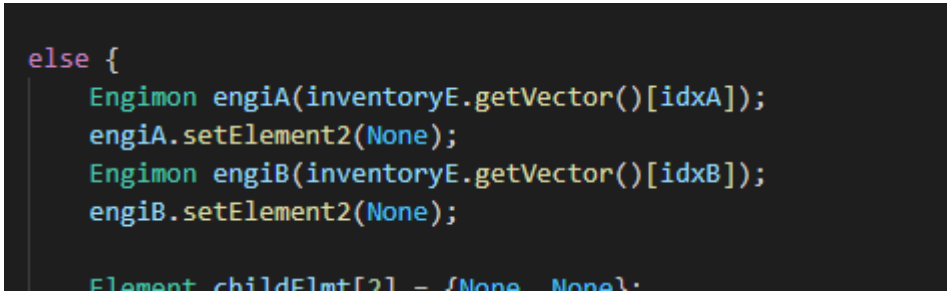
Mayoritas standard template yang digunakan pada program ini adalah container vector. Vector merupakan array dinamis sehingga proses menambahkan dan menghapus elemen pada vektor dapat dilakukan dengan mudah. Pada implementasinya vektor digunakan sebagai inventory player, penyimpanan engimon liar, dan penyimpanan map. Selain vektor digunakan juga STL yang berkaitan dengan container, seperti insert dan find.

3. Bonus Yang dikerjakan

3.1. Bonus yang diusulkan oleh spek

3.1.1. Dual Element Breeding

Dalam breeding, elemen anak dari dua engimon yang akan dipasangkan ditentukan oleh engimon yang memiliki keuntungan elemen. Hal itu tidak dapat dilakukan oleh engimon dengan dua elemen karena pembandingan keuntungan antar dua engimon dengan dua elemen tidak mengikuti aturan perbandingan kekuatan engimon yang hanya memiliki satu elemen (A.calculateAdvantage(B) + B.calculateAdvantage(A) ≠ 2). Masalah ini dipecahkan dengan menciptakan engimon baru hanya dengan elemen pertamanya saja.



3.1.2. Purely Random Wild Engimon Generation

Pada program ini, telah digunakan fitur untuk meng-generate wild engimon secara acak. Proses penciptaan wild engimon ini dimulai dengan memilih lokasi random untuk mem-*spawn* engimon baru. Setelah terpilih suatu lokasi, elemen engimon yang terbentuk akan dirandom kembali dari poolnya masing-masing(darat/air). Setelah ditentukan elemennya, kemudian akan di random level dan spesies dari engimon tersebut.

Cuplikan kode memilih lokasi dan level random:

```
void Map::addEngi(){
    if(wildEngi.size() < 10){
        int r = (rand()%30 + 1)*100;
        int x = rand()%mapMatrix[0].length(), y = rand()%mapMatrix.size();
        Engimon* w;
        Position p(x, y);

        while(!nobodyThere(p)){
            x = rand()%mapMatrix[0].length();
            y = rand()%mapMatrix.size();
            p.setX(x);
            p.setY(y);
        }
```

Cuplikan kode meng-spawn engimon random berdasarkan lokasi yang terpilih:

```
if (mapMatrix[y][x] == '+'){
    int el = rand()%4;
    switch(el){
        case 0:
            w = new Watermon(r, x, y);
            break;
        case 1:
            w = new Icemon(r, x, y);
            break;
        case 2:
            w = new WaterIcemon(r, x, y);
            break;
        case 3:
            w = new WaterGroundmon(r, x, y);
            break;
    }
    wildEngi.push_back(*w);
} else if(mapMatrix[y][x] == '-'){
    int e2 = rand()%4;
    switch(e2){
        case 0:
            w = new Firemon(r, x, y);
            break;
        case 1:
            w = new Groundmon(r, x, y);
            break;
        case 2:
            w = new Electricmon(r, x, y);
            break;
        ca
            w = new FireElectricmon(r, x, y);
            break;
    }
    wildEngi.push_back(*w);
}
```

Electricmon::Electricmon(int, int, in
Electricmon::Electricmon(string name,

3.2. Bonus Kreasi Mandiri

Pada program ini, dibuat sebuah logo yang merupakan ASCII ART yang akan muncul di awal dan akhir permainan.

4. Pembagian Tugas

Modul (dalam poin spek)	Designer	Implementer
1. Engimon	13519149, 13519150	13519158, 13519149, 13519150

2. Skill	13519142	13519158
3.a. Player Command	13519149, 13519158	13519158, 13519149
3.b. Inventory	13519149	13519158, 13519149
3.c. Active Engimon	13519149	13519158, 13519149
4. Battle	13519150	13519158, 13519150
5. Breeding	13519150, 13519142	13519158, 13519150
6.a-c. Tampilan Peta	13519149, 13519158	13519158, 13519149
6.f-i. Spawn dan Move Wild Engimon	13519149	13519158, 13519149
6.k. Membaca peta dari file eksternal	13519158	13519158
6.d, e, k. Penempatan Player dan Engimon	13519149, 13519158	13519158, 13519149
Bonus 1: Dual Element Breeding	13519150, 13519149	13519158, 13519150
Bonus 2: Purely Random Wild Engimon Generator	13519149	13519158, 13519149

Kreasi Mandiri: ASCII ART	13519158	13519158