Nama          : Lalu Rizki Aldi Saputra

Kelas           : A

Mata kuliah  : Praktikum PBO

**Penjelasan Code Modul 1**

Source Code

1. Main

|  |
| --- |
| package modul1\_pbo;  public class Modul1\_pbo {      public static void main(String[] args) {          //membuat player          Player p1 = new Player("Rizki");          Player p2 = new Player("zero");            //membuat senjata          Weapon w1 = new Weapon("pedang");          Weapon w2 = new Weapon("tombak");          Weapon w3 = new Weapon("Pisau");            //membuat pet          Pet pet1 = new Pet("Pikachu");          Pet pet2 = new Pet("raiju");          Pet pet3 = new Pet("togepi");            //membuat monster          Monster m1 = new Monster("Kaijuu");          Monster m2 = new Monster("Kaijuu no kodomo");          Monster m3 = new Monster("Ghoul");          Monster m4 = new Monster("Titan");          Monster m5 = new Monster("Hollow");            p1.getplayer();            //menyerang tanpa senjata          p1.menyerangMonster(m2);            //mengambil weapon          p1.menggunakanWeapon(w1);            //menyerang tanpa senjata          p1.menyerangMonster(m2);            //menangkap pet          p1.menangkapPet(pet2);            //menyerang menggunakan pet dan weapon          p1.menyerangMonster(m2);            //status player            //status senjata  //        p1.getSenjata();            //serang sampai level 2 dan pet mati          //status pemain          p1.getplayer();          p1.getSenjata();          p1.getPet();            p1.menyerangMonster(m1);          p1.menyerangMonster(m1);            p1.getplayer();          p1.getSenjata();          p1.getPet();            p1.menyerangMonster(m3);          p1.menyerangMonster(m3);            p1.getplayer();          p1.getSenjata();          p1.getPet();            m3.detailmonster();            //detail monster          //m2.detailmonster();        }    } |

class main merupakan class utama dimana terjadi membuatan objeck objeck dari class lain seperti player,pet,monster dan weapon. pada class ini lah nantinya method method pada class-class tadi akan digunakan dan di eksekusi sesuai dengan kemauan developer.

1. Player

|  |
| --- |
| package modul1\_pbo;  public class Player extends Entity{      Weapon weapon;      private Pet pet;        Player(String nama\_player){          super.name = nama\_player;          weapon = null;          pet = null;      }        public void menyerangMonster(Monster nama\_mosnter){          if(weapon == null && pet==null){              nama\_mosnter.hp -= super.base\_attack;              super.exp += nama\_mosnter.drop\_xp();            }else if(pet==null){              nama\_mosnter.hp -= super.base\_attack+ weapon.base\_attack;              int exp = nama\_mosnter.drop\_xp()/2;              super.exp += exp;              weapon.exp += exp;            }else{              nama\_mosnter.hp -= super.base\_attack+ weapon.base\_attack +pet.base\_attack;              nama\_mosnter.serangPet(pet);              int exp = nama\_mosnter.drop\_xp()/3;              super.exp += exp;              weapon.exp += exp;              pet.exp += exp;          }          nama\_mosnter.serangPlayer(this);          if(pet != null){              pet.healing(this);              pet.naiklevel();          }if(weapon != null){              weapon.naiklevel();          }          super.naiklevel();            gameOver();      }        public void menangkapPet(Pet nama\_pet){          pet = nama\_pet;      }        public void menggunakanWeapon(Weapon nama\_senjata){          weapon = nama\_senjata;      }          public void gameOver(){          if(hp<=0){              System.out.println("Game Over");          }      }        public void getplayer(){          System.out.println("----------------------------------------------------");          System.out.println("Nama          : "+super.name+                             "\nBase attack : "+super.base\_attack+                             "\nlevel       : "+super.level+                             "\nHP          : "+super.hp+                             "\nEXP         : "+super.exp+"\n\n");          System.out.println("----------------------------------------------------");      }        public void getSenjata(){          System.out.println("----------------------------------------------------");          System.out.println("Nama          : "+weapon.name+                             "\nBase attack : "+weapon.base\_attack+                             "\nlevel       : "+weapon.level+                             "\nHP          : "+weapon.hp+                             "\nEXP         : "+weapon.exp+"\n\n");          System.out.println("----------------------------------------------------");      }        public void getPet(){          System.out.println("----------------------------------------------------");          System.out.println("Nama          : "+pet.name+                             "\nBase attack : "+pet.base\_attack+                             "\nlevel       : "+pet.level+                             "\nHP          : "+pet.hp+                              "\nEXP         : "+pet.exp+"\n\n");          System.out.println("----------------------------------------------------");      }  } |

Class Player digunakan untuk menyimpan semua data charakter atau pemain. pada class ini terdapat beberapa methode/ behavior, yaitu menyerangMonster, menangkapPet, menggunakanWeapon, getplayer, getsenjata, getPet dan GameOver.

method menyerangMonster merupakan methode yang digunakan untuk menyerang monster yang ada. method pada dasarnya hanya memanggil variabel variabel yang ada berupa weapon atau pet jika ada maka attack ke monsterakan ditambahkan dengan base attack dari kedua attribut tersebut. pada method ini terdapat 4 pengkondisian yaitu saat player tidak memiliki senjata dan pet, saat player hanya memiliki senjata, saat player hanya memiliki pet dan saat player memiliki pet dan weapon. perlakuan disetiap pengkondisian hampir sama yang mebedakan hanya pada pembagian exp dan damage yang diberikan. pada method ini juga nantinya setelah player menyerang maka player akan mendapat counter attack dari monster yang diserang sesuai dengan base\_attack dari monster itu. kemudian jika monsternya mati maka exp akan ditambahkan ke dalam exp weapon, pet (jika ada) dan player. jika player sama dengan atau melebihi max exp maka level dari player atau perlengkapan itu akan naik level.

Method menangkapPet merupakan method yang digunakan untuk menangkap pet sesuai dengan pet yang diinginkan. codenya sangat simple karena disini player dan pet memiliki hubungan 1 ke 1 maka yang diperlukan hanya mengubah pet dari objeck player itu ke pet yang diinginkan menggunakan script “pet = nama\_pet;”.

Method menggunakanWeapon merupakan method yang digunakan player agar dapat menggunakan senjata yang tersedia. sama halnya dengan yang ada dimethod menangkapPet. class player disini memiliki hubungan 1 ke 1 dengan class weapon jadi yang dilakukan hanya mengubah weapon dari objeck player itu ke weapon yang diinginkan menggunakan script “weapon = nama\_senjata;”.

method GameOver merupakan method yang digunakan untuk mengecek apakah darah atau hp dari player masih belum habis jika sudah habis maka game akan selesai atau game over.

method getplayer, getsenjata dan getpet sebenarnya merupakan method yang mirip karena hanya menampilkan detail dari setiap perlengkapan. getplayer akan menampilkan detail status dari player, getweapon akan menampilkan detail status dari senjata dan getpet akan menampilkan status dari pet yang didapatkan player.

1. Mob

|  |
| --- |
| package modul1\_pbo;  public class Mob extends Entity{      private boolean jiwa = true;      public int drop\_xp(){          if(super.hp <= 0 && jiwa == true){              System.out.println("Monster Mati");              jiwa = false;              return super.level\*3;          }          return 0;      }          public void detailmonster(){          System.out.println("Nama          : "+super.name+                             "\nBase attack : "+super.base\_attack+                             "\nlevel       : "+super.level+                             "\nHP          : "+super.hp+                             "\nEXP         : "+super.exp);      }  } |

class mob merupakan superclass dari class pet dan monster. class ini memiliki dua method yaitu drop\_exp dan detailmonster. drop\_exp merupakan method dimana saat player membunuh monster atau monster membunuh pet maka akan terjadi drop exp. exp yang didrop akan sesuai dengan level monster dikalikan 3.

method detailmonster sama halnya dengan method getplayer, getweapon dan getpet pada class player, method ini hanya akan menampilkan status dari monster itu.

1. Entity

|  |
| --- |
| package modul1\_pbo;  public class Entity {      String name;      int level = 1;      int base\_attack = level\*2;      int max\_hp = 10;      int hp = max\_hp;      int exp = 0;      int max\_exp = 2\*level;        public void naiklevel(){          if(exp >= max\_exp){              level++;              exp -= max\_exp;          }      }  } |

class entity digunakan sebagai superclass dimana akan menyimpan nilai name, level, base attack, max hp, hp, exp, ,ax exp dan method naik level.

methode naik level ini digunakan untuk mengecek apakah player,weapon atau mob sudah memenuhi syarat untuk naik level atau tidak.

1. Monster

|  |
| --- |
| package modul1\_pbo;  public class Monster extends Mob{            Monster(String nama\_monster){              super.name = nama\_monster;          }            public void serangPlayer(Player Nama\_player){              Nama\_player.hp -= super.base\_attack;          }            public void serangPet(Pet Nama\_pet){              Nama\_pet.hp -= super.base\_attack;              if(super.hp >0){                  super.exp +=Nama\_pet.drop\_xp();              }          }   } |

class monster digunakan untuk menyimpan method method atau hal hal yang dapat dilakukan monster seperti menyerang player dan menyerang pet.

method serangPlayer digunakan untuk menyerang player dimana damage yang dihasilkan didapatkan dari base attack monster tersebut.

method serangPet digunakan untuk menyerang pet. sama halnya dengan pada saat menyerang player monster menggunakan base attack sebagai damage, bedanya jika monster membunuh pet maka dia akan mendapatkan exp yang dapat digunakan untuk naik level.

1. Pet

|  |
| --- |
| package modul1\_pbo;  import java.util.Random;  public class Pet extends Mob{  Random random = new Random();  private Player player;    Pet(String name){  super.name = name;  player = null;  }    public void ditangkap(Player nama\_player){  player = nama\_player;  }    public void healing(Player p){  if(p.hp<p.max\_hp){  int num = random.nextInt(2);  if(num==1){  p.hp +=super.level\*2;  }  }  }  } |

Method untuk ditangkap player dan dapat melakukan healing. Pada method healing terdapat pengkondisian jika hp dari player kurang dari max hpnya maka baru pet akan mengheal player dengan pertentase 50%. Untuk menentukan peluangnya saya menggunakan variabel random, dimana variabel random diset agar hanya menampilkan antara 1 atau 0 , jadi jika variabel random itu menampilkan angka 1 maka pet akan mengheal player yang menjadi tuannya.

1. Weapon

|  |
| --- |
| package modul1\_pbo;  public class Weapon extends Entity{  private Player player;    Weapon(String nama\_senjata){  super.name = nama\_senjata;  player = null;  }    public void mengganti\_senjata(Weapon new\_weapon, Player p){  this.player = player;  player.weapon = new\_weapon;  }    public void membuang\_senjata(Player player){  this.player = player;  player.weapon = null;  }    public void mendapatkan\_senjata(Weapon weapon, Player player){  this.player = player;  player.weapon = weapon;  }    } |