

Dasar Dasar Pemrograman 1

Kelas Ekstensi

Lab 08 – Name space

Deadline : 29 April 2017 23:55 WIB

1. Inisiasi Monster

Pada bulan April 2017, Pak Chanek mendapatkan sebuah *project* untuk membuat permainan yang berjudul GoMon. Permainan ini dirancang khusus untuk memberikan hiburan pada anak-anak. Di dalam permainan GoMon ini, pengguna dapat menggabungkan dua jenis monster. Tetapi proses penggabungan kedua monster ini belum dikerjakan oleh Pak Chanek. Pak Chanek membutuhkan bantuan dari mahasiswa ekstensi DDP 1 untuk dapat menyelesaikan proses penggabungan kedua monster. Pada permainan GoMon, para monster hanya dapat memiliki salah satu dari tiga elemen. Ketiga elemen tersebut adalah air, api dan tumbuhan. Setiap kali proses penggabungan berlangsung akan tercipta monster baru. Pak Chanek juga telah membuat beberapa catatan untuk mempermudah mahasiswa ekstensi DDP 1 dapat lebih memahami proses penggabungan monster. Selain itu **para monster dalam permainan GoMon memiliki tiga atribut yaitu: nama, elemen dan kekuatan.**

1. Nama monster

Setiap monster yang hidup di dalam permainan GoMon memiliki nama. Setiap nama monster akan berakhiran “mon”. Contohnya: monster dengan nama **yoyo** didalam permainan GoMon akan disimpan dengan nama **yoyomon**.

2. Elemen Monster

Di dalam permainan GoMon, setiap monster memiliki satu elemen. Elemen yang terdapat di permainan hanya ada air, api dan tumbuhan. Dan setiap elemen memungkinkan untuk terjadinya kasus untuk saling melemahkan.

- Elemen air lemah terhadap elemen tumbuhan.
- Elemen api lemah terhadap elemen air.
- Elemen tumbuhan lemah terhadap elemen api.

3. Penggabungan Monster

Di permainan GoMon, kedua monster dapat digabungkan dan menjadi sebuah monster baru. Akan tetapi pada saat proses penggabungan kedua monster ada beberapa hal yang harus diperhatikan.

- Kedua monster yang bergabung akan menghasilkan nama monster baru. Nama monster baru didapat dari penggabungan dari nama monster yang digabungkan. Contohnya: **monster pertama bernama warmon** dan **monster kedua bernama greymon**. Jika monster pertama dan kedua digabungkan maka akan menghasilkan **monster baru yang bernama wargreymon**. (Catatan: nama monster pertama tanpa kata “mon” digabungkan dengan nama monster kedua)

- Kedua monster yang bergabung akan menghasilkan monster baru dengan kekuatan yang baru pula. Akan tetapi kekuatan monster baru tidak selalu menghasilkan kekuatan yang besar. Total kekuatan monster baru berdasarkan elemen yang dimiliki oleh monster pertama dan monster kedua.
 - **Jika monster pertama memiliki elemen yang sama dengan monster kedua**, maka kekuatan monster baru didapat dari hasil **penjumlahan kekuatan monster pertama dan monster kedua**, lalu **ditambah 50%** dari hasil penjumlahan kekuatan monster pertama dan monster kedua.

Kekuatan monster baru = (kekuatan monster pertama + kekuatan monster kedua) + [(kekuatan monster pertama + kekuatan monster kedua) x 50%]

- **Jika kedua monster** yang digabungkan memiliki **elemen yang berbeda** seperti penjelasan yang telah Pak Chanek berikan pada poin (2), maka kekuatan monster baru didapat dari **hasil penjumlahan dari kekuatan kedua monster dikalikan 50%..**

Kekuatan monster baru = (kekuatan monster pertama + kekuatan monster kedua) x 50%

- Kedua monster yang akan bergabung akan **menghasilkan monster baru dengan elemen yang sama dengan elemen monster dengan kekuatan lebih besar.**
- **Jika kedua monster memiliki elemen yang berbeda** namun **kekuatan yang sama besarnya**, elemen dari monster hasil penggabungan monster tersebut memiliki **elemen yang sama dengan elemen monster pertama.**

Pak Chanek ternyata sudah membuat blok program dari bahasa Python untuk melakukan tes pada permainan GoMon. Anda diminta untuk membuat class GoMon dengan melengkapi blok program tersebut. Berikut ini adalah blok program yang telah dibuat oleh Pak Chanek:

```
import Play

class GoMon:

    def __init__(self, name, element, power):
        # TODO: Implementasikan code anda disini
        ...

    def add(self, otherGoMon):
        # TODO: Implementasikan code anda disini
        return ...

    def getNama(self):
        return self.nama

    def getElement(self):
```

```

        return self.element

    def getPower(self):
        return self.power

    def __str__(self):
        # TODO: Implementasikan code anda disini
        return ...

war = GoMon("war", "api", 1000)
sup = GoMon("sup", "api", 500)
leo = GoMon("leo", "api", 700)
goma = GoMon("goma", "air", 6000)
grey = GoMon("grey", "air", 400)
lada = GoMon("lada", "tumbuhan", 8500)
ku = GoMon("ku", "tumbuhan", 10)
pal = GoMon("pal", "tumbuhan", 11500)
rai = GoMon("rai", "api", 500)
te = GoMon("te", "api", 700)
mo = GoMon("mo", "air", 500)
lost = GoMon("lost", "air", 1000)

print("GoMon Start")
print(war)
print(sup)
print(leo)
print(goma)
print(grey)
print(lada)
print(ku)
print(pal)
print(rai)
print(te)
print(mo)
print(lost)
print()

newmon1 = war.add(grey)
newmon2 = sup.add(mo)
newmon3 = te.add(mo)
newmon4 = sup.add(lost)
newmon5 = lada.add(ku)
newmon6 = sup.add(lada)
newmon7 = war.add(te)
newmon8 = lost.add(grey)
newmon9 = pal.add(sup)
newmon10 = grey.add(ku)

print()
print("GoMon join")
print(newmon1)
print(newmon2)
print(newmon3)
print(newmon4)
print(newmon5)
print(newmon6)
print(newmon7)

```

```
print(newmon8)
print(newmon9)
print(newmon10)
```

Output

```
GoMon Start
=====GoMon=====
nama      : Warmon
elemen    : api
kekuatan  : 1000
=====GoMon=====
nama      : Supmon
elemen    : api
kekuatan  : 500
=====GoMon=====
nama      : Leomon
elemen    : api
kekuatan  : 700
=====GoMon=====
nama      : Gomamon
elemen    : air
kekuatan  : 6000
=====GoMon=====
nama      : Greymon
elemen    : air
kekuatan  : 400
=====GoMon=====
nama      : Ladamon
elemen    : tumbuhan
kekuatan  : 8500
=====GoMon=====
nama      : Kumon
elemen    : tumbuhan
kekuatan  : 10
=====GoMon=====
nama      : Palmon
elemen    : tumbuhan
kekuatan  : 11500
=====GoMon=====
nama      : Raimon
elemen    : api
kekuatan  : 500
=====GoMon=====
nama      : Temon
elemen    : api
kekuatan  : 700
=====GoMon=====
nama      : Momon
elemen    : air
kekuatan  : 500
```

```

=====GoMon=====
nama      : Lostmon
elemen    : air
kekuatan  : 1000

GoMon join
=====GoMon=====
nama      : Wargreymon
elemen    : api
kekuatan  : 700
=====GoMon=====
nama      : Supmomon
elemen    : api
kekuatan  : 500
=====GoMon=====
nama      : Temomon
elemen    : api
kekuatan  : 600
=====GoMon=====
nama      : Suplostmon
elemen    : air
kekuatan  : 750
=====GoMon=====
nama      : Ladakumon
elemen    : tumbuhan
kekuatan  : 12765
=====GoMon=====
nama      : Supladamon
elemen    : tumbuhan
kekuatan  : 4500
=====GoMon=====
nama      : Wartemon
elemen    : api
kekuatan  : 2550
=====GoMon=====
nama      : Lostgreymon
elemen    : air
kekuatan  : 2100
=====GoMon=====
nama      : Palsupmon
elemen    : tumbuhan
kekuatan  : 6000
=====GoMon=====
nama      : Greykumon
elemen    : air
kekuatan  : 205

```

2. Perbandingan Monster

Pak Chanek ingin para pemain dapat membandingkan dua monster untuk mengetahui informasi monster mana yang kuat, lemah atau monster mana yang memiliki elemen dan kekuatan yang seimbang. Pada setiap proses perbandingan dua monster, akan dilihat terlebih dahulu elemen yang dimiliki oleh kedua monster.

- Elemen air lebih kuat dari elemen api.
- Elemen api lebih kuat dari elemen tumbuhan.
- Elemen tumbuhan lebih kuat dari elemen air.
- Jika monster memiliki elemen yang sama, maka yang dibandingkan adalah besar kekuatannya.

Implementasikan dan lengkapi file `Play.py` untuk menangani pemanggilan code dibawah ini dan menghasilkan output yang sesuai.

```
Play.greaterThan(war,lost)      # Outputnya : False
Play.greaterThan(sup,lada)      # Outputnya : True
Play.greaterThan(leo,te)        # Outputnya : False
Play.lowerThan(goma < grey)     # Outputnya : False
Play.lowerThan(grey < lost)     # Outputnya : True
Play.lowerThan(lada < war)      # Outputnya : True
Play.equals(ku == mo)           # Outputnya : False
Play.equals(pal == sup)         # Outputnya : False
Play.equals(rai == sup)         # Outputnya : True
Play.greaterThan(te > war)      # Outputnya : False
Play.lowerThan(mo < mo)         # Outputnya : False
Play.greaterThan(lost > leo)    # Outputnya : True
```