

PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN
LEMBAR KERJA MAHASISWA
TUGAS BESAR : GAME ULAR



oleh:

Dwi Bagus Dermawan (DS-02-02 1206220023)

Yusril Ihsanurrijal (DS-02-02 120622003)

Easy Jovanie (DS-02-02 1206220016)

PROGRAM STUDI SAINS DATA
FAKULTAS TEKNOLOGI INFORMASI DAN BISNIS
INSTITUT TEKNOLOGI TELKOM SURABAYA
2022

DAFTAR ISI

BAB I	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan	1
1.4 Manfaat Penelitian	1
1.5 Batasan Masalah	1
BAB II	3
TINJAUAN PUSTAKA	3
2.1 Function (def)	3
2.2 Library pada Python	5
2.3 Module Pygame	6
2.4 Module Random	7
BAB III	9
METODOLOGI PENELITIAN	9
3.1 Sumber Data	9
3.2 Langkah Analisis	9
3.3 Alur Penelitian	10
3.4 Program	10
BAB IV	14
ANALISIS DAN PEMBAHASAN	14
4.1 Langkah-Langkah Memulai Program	14
4.2 Jawaban	14
BAB V	20
PENUTUP	20
5.1 Kesimpulan	20
5.2 Saran	20
DAFTAR PUSTAKA	21

BAB I

PENDAHULUAN

1.1 Latar Belakang

Python adalah salah Bahasa Pemrograman yang saat ini banyak digunakan Developer untuk membuat berbagai macam teknologi. Python saat ini menduduki Bahasa pemrograman paling populer di dunia per Agustus 2021 dengan mencapai rasio penggunaan sebesar 31.47% jika dibandingkan dengan seluruh Bahasa pemrograman yang digunakan oleh developer di seluruh dunia.

Python dapat digunakan untuk pengembangan Web, Machine Learning, bahkan untuk pengembangan Game. Pengembangan Game dengan Bahasa Python saat ini sangatlah memungkinkan karena adanya sebuah modul yang memudahkan developer dalam membuat game. Modul tersebut bernama PyGame.

1.2 Rumusan Masalah

Rumusan masalah dari praktikum ini yaitu:

1. Apa itu library pada Python?
2. Bagaimana cara membuat game ular menggunakan bahasa pemrograman Python?

1.3 Tujuan

Tujuan dari praktikum ini adalah:

1. Memahami library pada bahasa pemrograman Python,
2. Memahami cara membuat game ular menggunakan bahasa pemrograman Python.

1.4 Manfaat Penelitian

Adapun manfaat dalam praktikum ini yaitu:

1. Menambah wawasan mengenai algoritma dan pemograman pada bahasa Python,
2. Dapat menjalankan atau menggunakan library pada Python,
3. Dapat menjalankan dan membuat game ular menggunakan bahasa Python.

1.5 Batasan Masalah

Batasan masalah dalam praktikum ini adalah :

1. Bahasa pemograman yang digunakan ialah bahasa Python,
2. Program praktikum ini dijalankan menggunakan Visual Studio Code,

3. Program yang dijalankan menggunakan if, for, while, function dan beberapa library pada Python.

BAB II

TINJAUAN PUSTAKA

2.1 Function (def)

Fungsi adalah blok kode terorganisir dan dapat digunakan kembali yang digunakan untuk melakukan sebuah tindakan/action. Fungsi memberikan modularitas yang lebih baik untuk aplikasi anda dan tingkat penggunaan kode tinggi.

Mendefinisikan Fungsi Python

Anda dapat menentukan fungsi untuk menyediakan fungsionalitas yang dibutuhkan. Berikut adalah aturan sederhana untuk mendefinisikan fungsi dengan Python:

- Fungsi blok dimulai dengan def kata kunci diikuti oleh nama fungsi dan tanda kurung ().
- Setiap parameter masukan atau argument harus ditempatkan di dalam tanda kurung ini. anda juga dapat menentukan parameter di dalam tanda kurung ini.
- Pernyataan pertama dari sebuah fungsi dapat berupa pernyataan opsional – string dokumentasi fungsi atau docstring.
- Blok kode dalam setiap fungsi dimulai dengan titik dua(:) dan identasi.
- Pernyataan Kembali [ekspresi] keluar dari sebuah fungsi, secara opsional menyampaikan Kembali ekspresi ke pemanggil. Pernyataan pengembalian tanpa argument sama dengan return None.

Berikut contoh penggunaan fungsi(def):

#contoh1

```
def printme(str):
```

```
    "This prints a passed string into this function"
```

```
    print (str)
```

```
    return
```

```
printme("hello world")
```

output: hello world

#contoh2 penjumlahan sebuah barisan angka

```
def jumlah(angka):
```

```
    total = 0
```

```
    for x in angka:
```

```
        total += x
```

```
    return total
```

```
print ("jumlah: ",jumlah((8,3,1,4,5)))
```

Output: jumlah: 21

#contoh3 pengecekan angka apakah ganjil atau genap?

```
def cek_ganjil_genap(angka):
```

```
    if angka%2==0:
```

```
        print("genap")
```

```
    else:
```

```
        print("ganjil")
```

```
    return
```

```
cek_ganjil_genap(5) #input angka, misalkan 5
```

Output: ganjil

#contoh4 perhitungan rata - rata

```
def rata_rata(a,b,c):
```

```
    return (a+b+c)/3
```

```
rata_rata(1,2,3)
```

Output: 2.0

#contoh5 perhitungan sederhana matematika

```
def kalkulator(angka1, angka2):
```

```
    print(angka1+angka2)
```

```
    print(angka1-angka2)
```

```
    print(angka1*angka2)
```

```
    print(angka1/angka2)
```

```
kalkulator(1,2) #inputan 2 angka, angka1:1, dan angka2:2
```

Output:

3

-1

2

0.5

2.2 Library pada Python

Di Python terdapat library. Library Python ini ada berbagai fungsi. Sehingga ketika kita ingin menggunakan sebuah fungsi, maka kita harus memanggil library yang membawahnya terlebih dahulu. Library ini dapat diibaratkan sebagai rumah, jika kita ingin bertemu dengan seseorang maka kita harus menuju rumahnya terlebih dahulu.

Berikut cara menjalankan library pada python :

1. Buka new terminal di Visual Studio Code,
2. Ketik pip install (pilih salah satu library yang ada di Python),
3. Ketika sudah berhasil terinstall, lalu ketik import (nama library) pada syntax.

Berikut contoh penggunaan library pada Python :

```
#contoh_module_math
```

```
import math
```

```
log = 1e-4
```

```
print(math.log(math.fabs(log),10))
```

```
output: -3.9999999999999999
```

2.3 Module Pygame

Pygame dimulai pada musim panas tahun 2000. Sebagai seorang programmer C selama bertahun-tahun, saya menemukan Python dan SDL pada waktu yang hampir bersamaan. Anda sudah terbiasa dengan Python, yang ada di versi 1.5.2. Anda mungkin perlu pengenalan terhadap SDL, yang merupakan Simple DirectMedia Layer. Dibuat oleh Sam Lantinga, SDL adalah pustaka C lintas platform untuk mengendalikan multimedia, sebanding dengan DirectX. SDL telah digunakan untuk ratusan game komersial dan open source. Saya terkesan dengan betapa bersih dan mudahnya kedua proyek tersebut dan tidak lama kemudian saya menyadari bahwa menggabungkan Python dan SDL adalah sebuah proposal yang menarik.

Saya menemukan sebuah proyek kecil yang sudah berjalan dengan ide yang sama persis, PySDL. Dibuat oleh Mark Baker, PySDL adalah implementasi langsung dari SDL sebagai ekstensi Python. Antarmukanya lebih bersih daripada pembungkus SWIG yang umum, tetapi saya merasa bahwa ia memaksakan kode "gaya C". Kematian PySDL yang tiba-tiba mendorong saya untuk membuat proyek baru sendiri.

Saya ingin membuat sebuah proyek yang benar-benar memanfaatkan Python. Tujuan saya adalah membuatnya mudah untuk melakukan hal-hal yang sederhana, dan langsung melakukan hal-hal yang sulit. Pygame dimulai pada bulan Oktober 2000. Enam bulan kemudian, Pygame versi 1.0 dirilis.

Contoh Umum Modul Pygame

<u>cdrom</u>	playback
<u>cursors</u>	Membuat gambar kursor
<u>display</u>	Mengontrol tampilan layar
<u>draw</u>	Menggambar bentuk-bentuk sederhana
<u>event</u>	Mengelola event
<u>font</u>	Mengatur font
<u>image</u>	Menambahkan gambar
<u>joystick</u>	Mengatur joystick
<u>key</u>	Mengatur keyboard
<u>mouse</u>	Mengatur mouse
<u>sndarray</u>	Membuat suara
<u>surfarray</u>	manipulate images with numpy

<u>time</u>	Mengatur waktu
<u>transform</u>	Mengatur gaya pada gambar

Berikut contoh penggunaan pygame :

```
1import sys, pygame
2pygame.init()
3
4size = width, height = 320, 240
5speed = [2, 2]
6black = 0, 0, 0
7
8screen = pygame.display.set_mode(size)
9
10ball = pygame.image.load("intro_ball.gif")
11ballrect = ball.get_rect()
12
13while True:
14    for event in pygame.event.get():
15        if event.type == pygame.QUIT: sys.exit()
16
17    ballrect = ballrect.move(speed)
18    if ballrect.left < 0 or ballrect.right > width:
19        speed[0] = -speed[0]
20    if ballrect.top < 0 or ballrect.bottom > height:
21        speed[1] = -speed[1]
22
23    screen.fill(black)
24    screen.blit(ball, ballrect)
25    pygame.display.flip()
```

2.4 Module Random

Modul ini mengimplementasikan generator bilangan acak semu untuk berbagai distribusi.

Untuk bilangan bulat, ada pemilihan seragam dari suatu rentang. Untuk urutan, ada pemilihan seragam dari elemen acak, fungsi untuk menghasilkan permutasi acak dari daftar di tempat, dan fungsi untuk pengambilan sampel acak tanpa penggantian.

Pada garis nyata, terdapat fungsi untuk menghitung distribusi seragam, normal (Gaussian), lognormal, eksponensial negatif, gamma, dan beta. Untuk menghasilkan distribusi sudut, tersedia distribusi von Mises.

Fungsi-fungsi yang disediakan oleh modul ini sebenarnya adalah metode terikat dari instance tersembunyi dari kelas `random.Random`. Anda bisa menginstansiasi

instance Anda sendiri dari Random untuk mendapatkan generator yang tidak berbagi state.

Kelas Random juga bisa disubkelaskan jika Anda ingin menggunakan generator dasar yang berbeda yang Anda rancang sendiri: dalam hal ini, timpa metode `random()`, `seed()`, `getstate()`, dan `setstate()`. Secara opsional, sebuah generator baru bisa menyediakan metode `getrandbits()` - ini memungkinkan `randrange()` untuk menghasilkan pilihan pada rentang yang sangat besar.

Berikut contoh penggunaan modul random :

```
>>> random()                                # Random float:
0.0 <= x < 1.0
0.37444887175646646

>>> uniform(2.5, 10.0)                      # Random float:
2.5 <= x <= 10.0
3.1800146073117523

>>> expovariate(1 / 5)                      # Interval
between arrivals averaging 5 seconds
5.148957571865031

>>> randrange(10)                           # Integer from 0
to 9 inclusive
7

>>> randrange(0, 101, 2)                    # Even integer
from 0 to 100 inclusive
26

>>> choice(['win', 'lose', 'draw'])          # Single random
element from a sequence
'draw'

>>> deck = 'ace two three four'.split()
>>> shuffle(deck)                            # Shuffle a list
>>> deck
['four', 'two', 'ace', 'three']

>>> sample([10, 20, 30, 40, 50], k=4)        # Four samples
without replacement
[40, 10, 50, 30]
```

BAB III

METODOLOGI PENELITIAN

3.1 Sumber Data

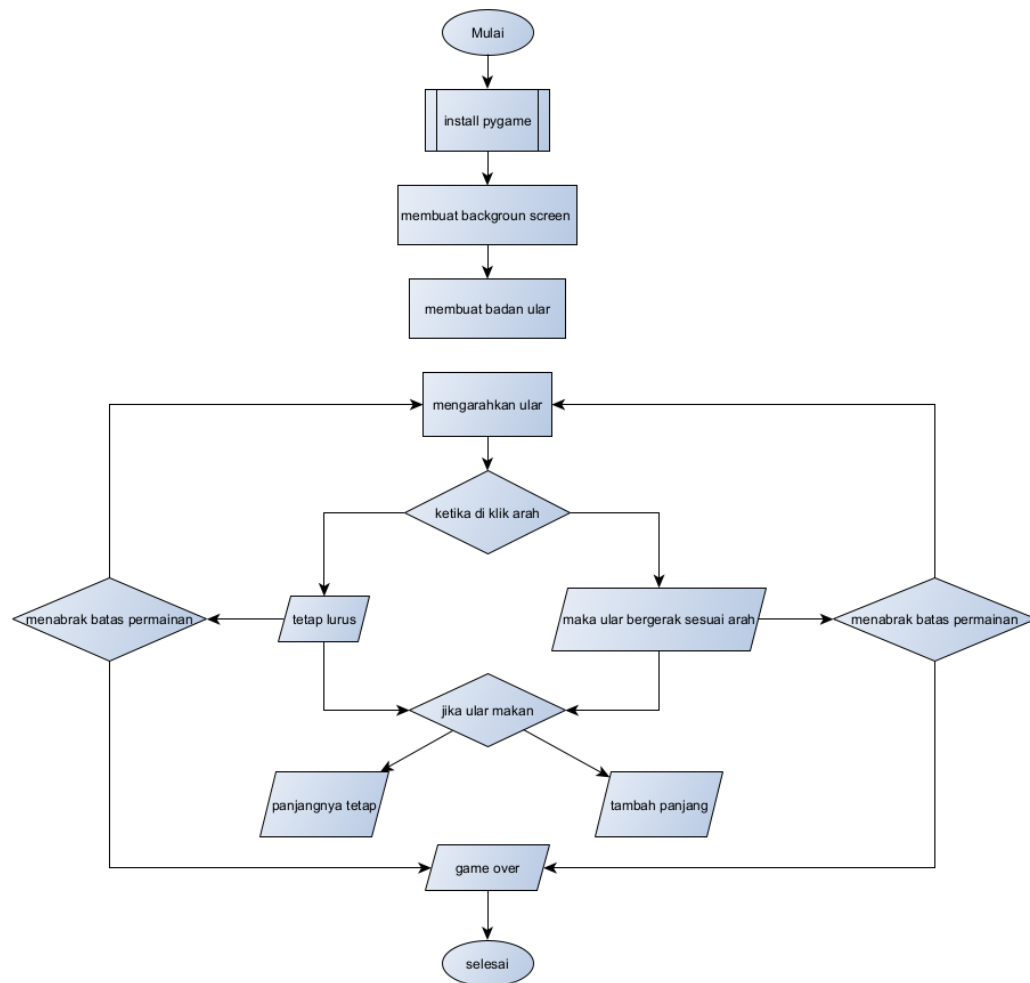
1. Data primer : source code Python
2. Data sekunder : Visual Studio Code

3.2 Langkah Analisis

Langkah analisis secara umum antara lain :

1. Menginstall library pada Python yaitu Pygame, dengan cara pip install pygame
2. Import pygame
3. Membuat background atau screen
4. Membuat badan ular
5. Membuat agar badan ular bisa bergerak
6. Membuat batasan permainan yang biasa disebut Game Over
7. Menambahkan makanan ular
8. Meningkatkan Panjang ular ketika memakan makanan ular tersebut
9. Menambahkan fitur skor.

3.3 Alur Penelitian



3.4 Program

Pseudocode

```
#PROGRAM_SNAKE_GAME
```

```
IMPORT pygame
```

```
IMPORT random
```

Deklarasi :

```
putih = (255,255,255)
```

```
hitam = (0,0,0)
```

```
merah = (213, 50, 0)
```

```
hijau = (0,255,0)
```

```
lebar = 1000
```

```
tinggi = 600
```

```
kepala_kotak = 10
```

```
speed = 15
```

```
layar = pygame.display.set_mode((lebar,tinggi))
```

```
clock = pygame.time.Clock
```

```
Algoritma :
```

```
DEF skormu(skor):
```

```
    hitung_skor = skor_font.render("Skormu: " + str(skor), True, putih)
```

```
    layar.blit(hitung_skor, [0, 0])
```

```
DEF our_snake(kepala_kotak, list_ular):
```

```
    FOR x IN list_ular:
```

```
        pygame.draw.rect(layar, hijau, [x[0], x[1], kepala_kotak, kepala_kotak])
```

```
DEF pesan(psn, warna):
```

```
    komen = font_style.render(psn, True, warna)
```

```
    layar.blit(komen, [lebar / 6, tinggi / 3])
```

```
DEF gameloop
```

```
    kalah = False
```

```
    keluar = False
```

```
    x1 = lebar/2
```

```
    y1 = tinggi/2
```

```
    arah_x = 0
```

```
    arah_y = 0
```

```
    badan = []
```

```
    Panjang_ular = 1
```

```
    makan_x = round(random.randrange(0, lebar – kepala_kotak)/10)*10
```

```
    makan_y = round(random.randrange(0, tinggi – kepala_kotak)/10)*10
```

```
    WHILE kalah=True
```

```
        layar.fill(hitam)
```

```
        pesan("Game Over!")
```

```
        skormu(panjang_ular-1)
```

```
        FOR event IN pygame.event.get()
```

```
            IF event.type == pygame.keydown
```

```
IF event.key == pygame.k_esc
    kalah = True
    keluar = False
IF event.key == pygame.k_space
    gameloop()
FOR event IN pygame.event.get()
    IF event.type == pygame.quit
        Kalah = true
    IF event.type == pygame.keydown
        IF event.key == pygame.k_left
            Arah_x = -kepala_kotak
            Arah_y = 0
        ELIF event.key == pygame.k_right
            Arah_x = kepala_kotak
            Arah_y = 0
        ELIF event.key == pygame.k_up
            Arah_y = -kepala_kotak
            Arah_x = 0
        ELIF event.key == pygame.k_down
            Arah_y = kepala_kotak
            Arah_x = 0
IF x1 >= lebar OR x1 < 0 OR y1 >= tinggi OR y1 < 0:
    keluar = True
x1 += arah_x
y1 += arah_y
layar.fill(hitam)
pygame.draw.rect(layar, merah, [makan_x, makan_y, kepala_kotak,
    kepala_kotak])
anggota_badan = []
anggota_badan.append(x1)
anggota_badan.append(y1)
```

```
badan.append(anggota_badan)
IF len(badan) > panjang_ular:
    DEL badan[0]
FOR x IN badan[: -1]:
    IF x == anggota_badan:
        keluar = True
our_snake(kepala_kotak, badan)
skormu(panjang_ular - 1)
pygame.display.update()
IF x1 == makan_x AND y1 == makan_y:
    makan_x = round(random.randrange(0, lebar - kepala_kotak) / 10.0) * 10.0
    makan_y = round(random.randrange(0, tinggi - kepala_kotak) / 10.0) * 10.0
    panjang_ular += 1
clock.tick(speed)
pygame.quit()
```

BAB IV

ANALISIS DAN PEMBAHASAN

4.1 Langkah-Langkah Memulai Program

Langkah-langkah memulai program secara umum:

1. Mengetahui persoalan yang akan diselesaikan
2. Membuat rumusan untuk pemecahan persoalan
3. Membuat suatu algoritma
4. Mengubah algoritma tersebut ke dalam bahasa pemrograman
5. Menguji coba dan mengevaluasi program yang telah dibuat
6. Pembuatan dokumentasi

4.2 Jawaban

```
#import library
```

```
import pygame, random, time
```

```
#init
```

```
pygame.init()
```

```
#membuat variable warna
```

```
putih = (255, 255, 255)
```

```
kuning = (255, 255, 102)
```

```
hitam = (0, 0, 0)
```

```
merah = (213, 50, 80)
```

```
hijau = (0, 255, 0)
```

```
#ukuran display
```

```
lebar = 1000
```

```
tinggi = 600
```

```
layar = pygame.display.set_mode((lebar, tinggi))
```



```
pygame.display.set_caption('Game Ular by Kelompok 01') #memberi caption
pada dsisplay / nama game

clock = pygame.time.Clock()

kepala_kotak = 10 #ukuran kepala ular
speed = 15 #kecepatan ular

font_style = pygame.font.SysFont("clanderon", 30)
skor_font = pygame.font.SysFont("clarendon", 40,bold=True) #font style untuk
papan skor

#membuat fungsi(function) perhitungan skor
def skormu(skor):
    hitung_skor = skor_font.render("Skormu: " + str(skor), True, putih)
    layar.blit(hitung_skor, [0, 0])

#fungsi(function) menambah panjang ular
def our_snake(kepala_kotak, list_ular):
    for x in list_ular:
        pygame.draw.rect(layar, hijau, [x[0], x[1], kepala_kotak, kepala_kotak])

#fungsi(function) memberi pesan ketika game over
def pesan(psn, warna):
    komen = font_style.render(psn, True, warna)
    layar.blit(komen, [lebar / 6, tinggi / 3])

def gameLoop(): #mengulangi permainan
    kalah = False
    keluar = False
```

```
x1 = lebar / 2
y1 = tinggi / 2
arah_x = 0
arah_y = 0
badan = []
panjang_ular = 1
makan_x = round(random.randrange(0, lebar - kepala_kotak) / 10.0) * 10.0
makan_y = round(random.randrange(0, tinggi - kepala_kotak) / 10.0) * 10.0
```

```
while not kalah:
```

```
    while keluar == True:
```

```
        layar.fill(hitam)
```

```
        pesan("Game Over! Tekan 'spasi'-lanjut! Tekan 'Esc'-keluar", putih)
```

```
        skormu(panjang_ular - 1)
```

```
        pygame.display.update()
```

```
    for event in pygame.event.get():
```

```
        if event.type == pygame.KEYDOWN:
```

```
            if event.key == pygame.K_ESCAPE:
```

```
                kalah = True
```

```
                keluar = False
```

```
            if event.key == pygame.K_SPACE:
```

```
                gameLoop()
```

```
    for event in pygame.event.get():
```

```
        if event.type == pygame.QUIT:
```

```
            kalah = True
```

```
        if event.type == pygame.KEYDOWN: #arah keyboard
```

```
if event.key == pygame.K_LEFT: #arah kiri
    arah_x = -kepala_kotak
    arah_y = 0
elif event.key == pygame.K_RIGHT: #arah kanan
    arah_x = kepala_kotak
    arah_y = 0
elif event.key == pygame.K_UP: #arah atas
    arah_y = -kepala_kotak
    arah_x = 0
elif event.key == pygame.K_DOWN: #arah bawah
    arah_y = kepala_kotak
    arah_x = 0

if x1 >= lebar or x1 < 0 or y1 >= tinggi or y1 < 0:
    keluar = True
x1 += arah_x
y1 += arah_y
layar.fill(hitam)
pygame.draw.rect(layar, merah, [makan_x, makan_y, kepala_kotak,
kepala_kotak])
anggota_badan = []
anggota_badan.append(x1)
anggota_badan.append(y1)
badan.append(anggota_badan)
if len(badan) > panjang_ular:
    del badan[0]

for x in badan[:-1]:
    if x == anggota_badan:
        keluar = True
```

```

our_snake(kepala_kotak, badan)

skormu(panjang_ular - 1)

pygame.display.update()

if x1 == makan_x and y1 == makan_y:
    makan_x = round(random.randrange(0, lebar - kepala_kotak) / 10.0) *
10.0
    makan_y = round(random.randrange(0, tinggi - kepala_kotak) / 10.0) *
10.0
    panjang_ular += 1

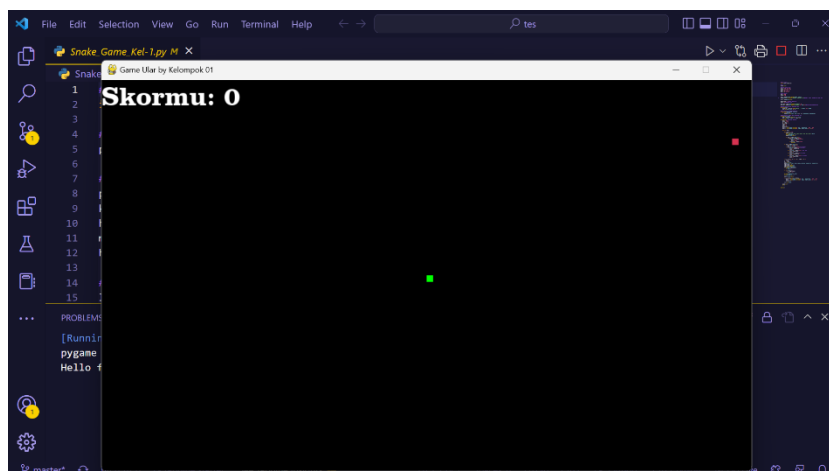
clock.tick(speed)

pygame.quit()
quit()

gameLoop()

```

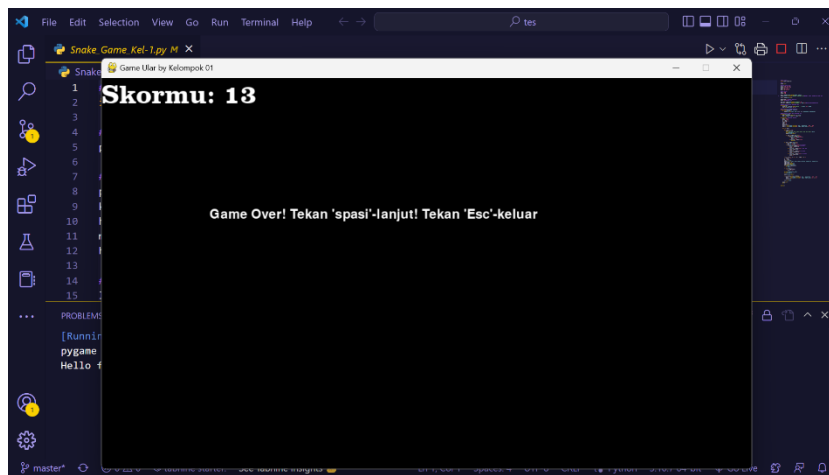
Hasilnya :



Gambar Hasil 1. 1



Gambar Hasil 1. 2



Gambar Hasil 1. 3

BAB V

PENUTUP

5.1 Kesimpulan

Pada praktikum kali ini terdapat beberapa library yaitu pygame dan random. PyGame adalah kumpulan dari beberapa modul Python yang khusus dibuat untuk mengembangkan Game. Di dalam PyGame sudah dilengkapi dengan library fungsional SDL yang mendukung untuk mengembangkan sebuah game. Library SDL (Simple DirectMedia Layer) adalah library yang memungkinkan developer untuk dapat mengakses audio, keyboard, mouse, joystick, perangkat grafis melalui OpenGL dan Direct3d. Hal ini sangatlah penting untuk menunjang pengembangan sebuah Game. Sedangkan library Random adalah inputan dimana nilai yang keluar itu secara acak.

Pada pemrograman python terdapat Function(def), fungsi ini digunakan untuk melakukan spesifikasi proses menurut inputannya dan mengeluarkan hasil. Di function(def) ini ada 2 macam fungsi yaitu tidak menggunakan parameter dan menggunakan parameter. Fungsi dari parameter yaitu dapat menerima satu, banyak atau tidak sama sekali parameter(masukkan) tetapi harus mengeluarkan hasil(keluaran). Tipe data dari parameter berupa; string, integer, float, decimal, array, atau yang lainnya.

Selain itu, ada juga function rekursif yang digunakan untuk memanggil dirinya sendiri yang menimbulkan efek perulangan.

5.2 Saran

Setelah melakukan praktikum ini, maka kami sebagai penyusun menyarankan beberapa hal seperti untuk praktikum ini yakni algoritma yang ditampilkan masih perlu penyederhanaan agar lebih bisa dikembangkan ke program yang lebih baik. Serta hasil membuat program membutuhkan waktu yang lama untuk menghasilkan output yang diinginkan. Dan bagi pemula kami menyarankan agar rajin berlatih coding python agar terbiasa dengan syntax di dalam program tersebut.



IT Telkom
Surabaya



Yayasan
Pendidikan
Telkom



Telkom
Indonesia

DAFTAR PUSTAKA

1. <https://www.pygame.org/docs/> diakses pada tanggal 29 Januari 2023
2. <https://www.dqlab.id/library-python-kenali-perbedaan-module-package-dan-library-pada-python> diakses pada tanggal 29 Januari 2023
3. https://www.w3schools.com/python/python_modules.asp diakses pada tanggal 28 Januari
4. https://www.w3schools.com/python/python_pip.asp diakses pada tanggal 28 Januari 2023
5. <https://docs.python.org/3/library/random.html> diakses pada tanggal 29 Januari 2023