**PEMROGRAMAN BERBASIS OBJEK**

**“KIVY DAN PYGAME”**

****

|  |  |
| --- | --- |
|  | **Disusun Oleh:** |
|  |  |
|  |  |
| NAMA | : - Teguh Subajra   * Holaw Savayona |
| NRP | : - 170411100020   * 170411100028 |
| KELAS/KELOMPOK | : B / PEMROGRAMAN BERBASIS OBJEK |
| DOSEN | : Arik Kurniawati, S.Kom., M.T |
|  |  |

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA**

**2018**

**KATA PENGANTAR**

Puji syukur kehadirat Allah SWT, atas segala limpahan rahmat, nikmat serta karunia-Nya yang tak ternilai dan tak dapat dihitung sehingga kami bisa menyusun dan menyelesaikan makalah ini. Makalah yang berjudul “Hasil Dokumentasi Kivy” ini disusun untuk memenuhi projek UAS Pemrograman Berbasis Objek

Makalah ini berisikan mengenai lika-liku persoalan sosial yang terjadi di sekitar objek wisata yang ada di Kota Tua, Jakarta. Masalah yang diangkat mulai dari masalah kemiskinan hingga kesenjangan sosial yang telah menjadi pemandangan sehari-hari di lokasi objek wisata tersebut.

Adapun, penyusunan makalah ini kiranya masih jauh dari kata sempurna. Untuk itu, kami menghaturkan permohonan maaf apabila terdapat kesalahan dalam makalah ini. Kamu pun berharap pembaca makalah ini dapat memberikan kritik dan sarannya kepada kami agar di kemudian hari kami bisa membuat makalah yang lebih sempurna lagi.

Akhir kata, kami ucapkan terima kasih kepada segala pihak yang tidak bisa disebutkan satu-persatu atas bantuannya dalam penyusunan makalah ini.

Kamal, 21 Desember 2018

**DAFTAR ISI**

KATA PENGANTAR ……………………………………………………………………………………………..2

Daftar Isi …………………………………………………………………………………………….3

Bab 1 Pendahuluan ……………………………………………………………………………..…………….4

1. **Kivy** ……………………………………………………………………………………………4
2. Pengertian …………………………………………………………………………4
3. Instalasi …………………………………………………………………………4
4. **Pygame** …………………………………………………………………………………….12
5. Pengertian ……………………………………………………………………….12
6. Instalasi ……………………………………………………………………….12

Bab 2 ……………………………………………………………………………..………….13

1. **Kivy** …………………………………………………………………………………………..13

a. Diagram Class …….……………………………………….13

b. Penjelasan Class dan Method .…………………………………………….16

c. Implementasi OOP pada kivy .…………………………………………….16

d. Aset Gambar …………………..…………………………………………….16

e. Screen Tampilan …………………..…………………………………………….16

f. Daftar Pusaka …………………..…………………………………………….17

1. **Pygame** ………………………………………..……………………………………………18

a. Diagram Class ….. …….………………..………………..18

b. Penjelasan Class dan Method .…………………………………………….22

c. Implementasi OOP pada Pygame .………………………………………23

d. Aset Gambar …………………..…………………………………………….23

e. Screen Tampilan …………………..…………………………………………….23

f. Daftar Pusaka …………………..…………………………………………….24

BAB 1. PENDAHULUAN

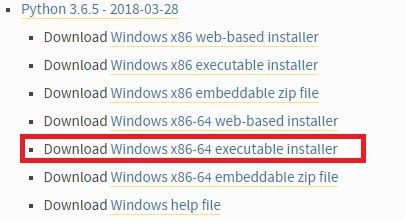
**1.a Pengertian kivy**

Kivy merupakan library Python open source yang digunakan untuk mengembangkan aplikasi mobile dengan user inetrface alami (Natural User Interface). Kivy bisa berjalan pada platform Android, iOS, Linux, OS X dan Windows. Kivy berlisensi MIT.

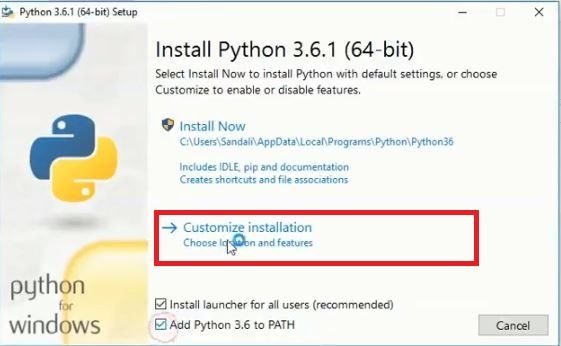
**1.b Instalasi kivy**

**Installasi Kivy**

### 1. Download Python versi 3.6 [disini](https://www.python.org/downloads/windows/)



### 2. Buka file yang sudahdiinstalltadi

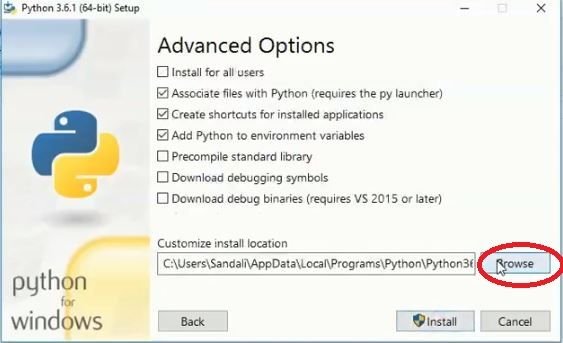


Sepertigambar yang diataskemudianklik Customize installation

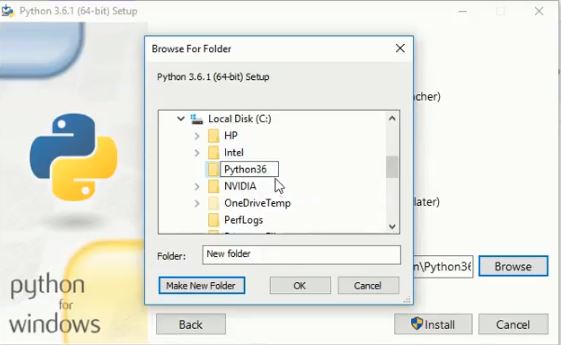
### 3. Pasitkansemuadiceklisdanklik next.



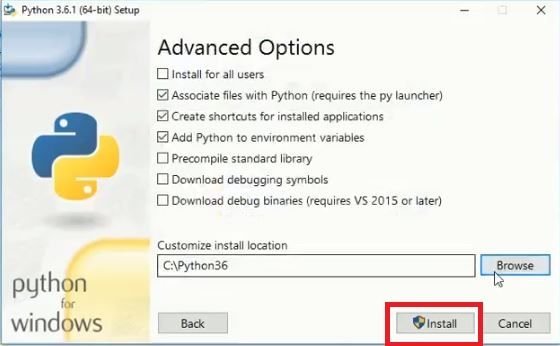
### 4. Kliktombol browse



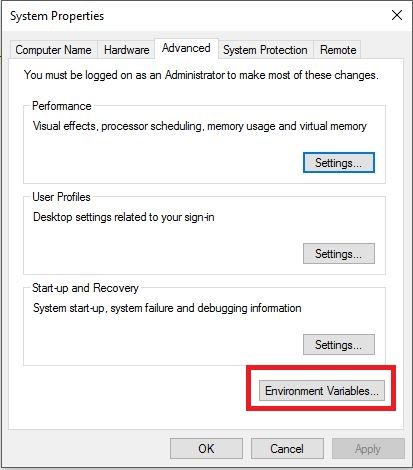
### 5. Lalupilih Local Disk (C:), danbuat folder barudengannama “Python36”(tanpatandapetik), klik ok.



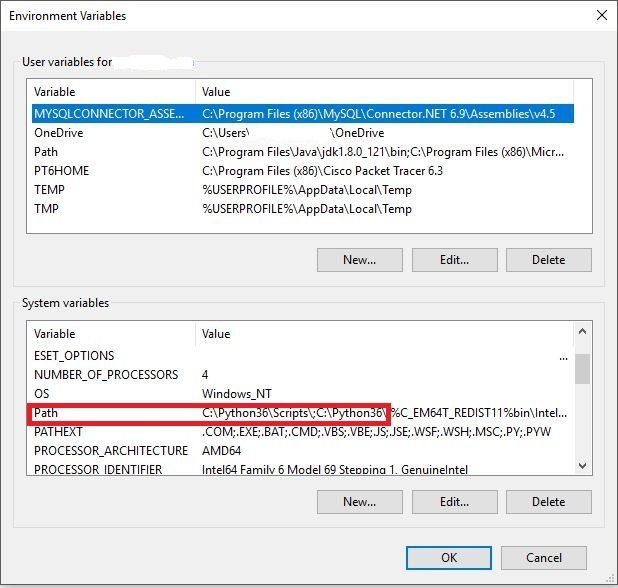
### 6. Pastikangambarnyaseperti yang dibawah, laluklik Install



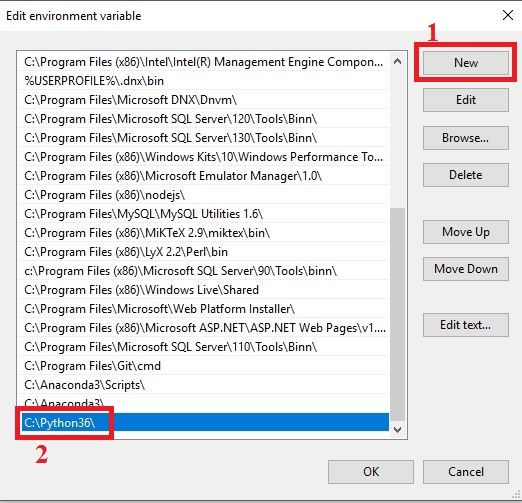
### 7. Bukacontol panel, System and Security, System, Advanced system settings, laluklik environment variables



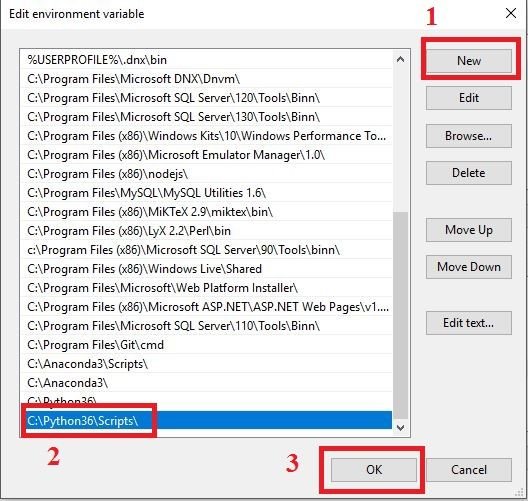
### 8. Pastikan file Python sudahditambahkan di path sepertigambar yang dibawah



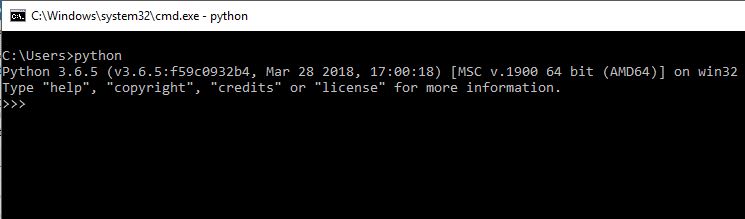
Jikabelumsilahkantambahkandengancaraklik Path, lalukliktombol edit, klik new, danisisepertigambardibawahini



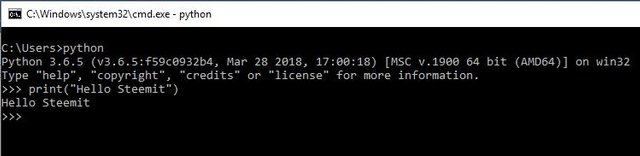
Klik new lagidanisisepertigambardibawahini



### 9. Bukacmduntukmemastikanbahwa Python sudahterinstalldenganbenar, tekantombol “windows + r” laluketik “cmd” dantekan enter. Setelahcmdterbukalaluketik python, dantekan enter, makatampilannyaakanseperti yang dibawah



### 10. ketik print(“Hello Steemit”)



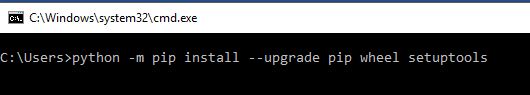
Makaoutputnyaadalah Hello Steemit.

## Cara menginstallKivy

Bukacmddengancaratekantombol “windows + r” ketik “cmd” lalutekan enter.

### 1. Padatampilancmdsilahkanandaketikperintahdibawahinilalutekan enter

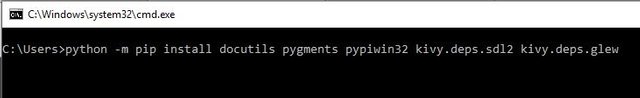
python -m pip install --upgrade pip wheel setuptools



Pastikanbahwaperangkatandaterhubungke internet, lalutungguprosesnyahinggaselesai, mungkinakanmemakanwaktusekitar 5 menittergantungpadakecepatan internet anda.

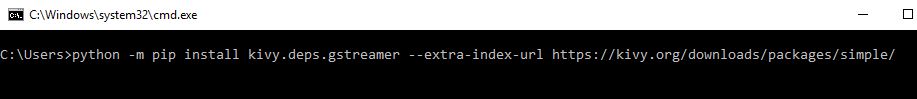
### 2. Jikalangkahpertamasudahselesaiketikperintahberikutpadacmd pula, tungguhingga proses selesai

python -m pip install docutilspygments pypiwin32 kivy.deps.sdl2 kivy.deps.glew



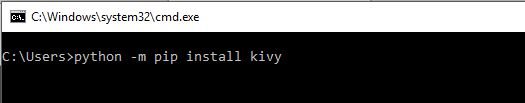
### 3. Samasepertisebelumya

python -m pip install kivy.deps.gstreamer --extra-index-url https://kivy.org/downloads/packages/simple/

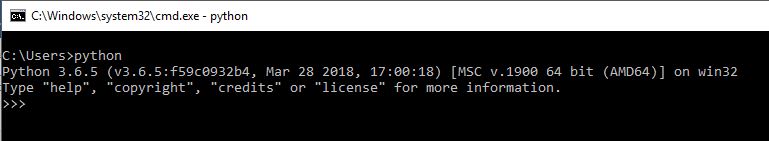


### 4. Langkahterakhir

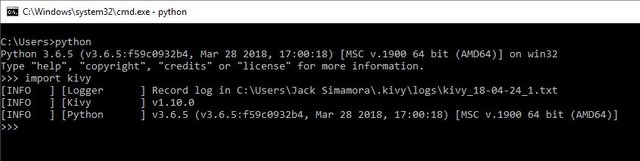
python -m pip install kivy



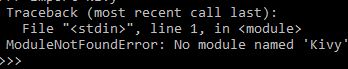
UntukmemastikanbahwaKivysudahterinstalldenganbenarsilahkanbuka “cmd” laluketik python dan enter.



Ketik import kivy lalu enter



Jikakivysudahterinstalldenganbenarmakatampilannyaakansedikitsepertigambar yang diatas, jikabelummakaakanseperti yang dibawah

  
Modultidakditemukan.

**2.a Pengertian Pygame**

**Pygame**

Pygame adalah seperangkat modul Python yang dirancang untuk membuat permainan. Pygame menambahkan fungsi di atas dengan sangat baik di SDL perpustakaan. Hal ini memungkinkan Anda untuk membuat sebuah game dengan fitur yang lengkap dan sebuah program multimedia dalam bahasa python. Pygame sangat portabel dan dapat berjalan pada hampir semua platform dan sistem operasi. Pygame sendiri telah didownload jutaan kali, dan telah memiliki jutaan kunjungan ke situsnya.  
  
Pygame ini gratis. Dan dirilis di bawah Lisensi GPL , Anda dapat membuat sebuah aplikasi yang open source, gratis, freeware, shareware, dan game komersial dengan pygame ini.

**2.b Instalasi Pygame**

**Instalasi Pygame**

Agar modul pygame dapat digunakan untuk pembuatan game, maka terlebih dahulu dilakukan instalasi pygame. Tahapan-tahapan yang harus dilakukan untuk instalasi pygame adalah:

1. **Siapkan file pygame.whl**

Download library pygame di <https://www.lfd.uci.edu/~gohlke/pythonlibs/> . Sesuaikan versi python yang terinstall dengan versi librari pygame yang akan di download. Contoh python v3.5 telah terinstall, maka download library pygame dengan cp3.5m-win32.

1. **Rubah file pygame.whl menjadi .zip**

Rubah extension file library pygame lalu extract isi library tersebut.

1. **Copy file pygame yang dibutuhkan**
2. Masuk ke dalam directory python (C:\User\#username\AppData\Local\Programs\Python\Python35-32)
3. Masuk kedalam folder “**include”** dan buat folder baru bernama **“pygame”**
4. Didalam folder hasil extract file library pygame yang sudah di download, masuk ke **“pygmae-1.9.4.data\header”**, copy semua file didalam folder tersebut dan masukkan ke dalam folder: **C:\User\#username\AppData\Local\Program\Python\Python35-32\include\pygame**
5. Kembali ke folder hasil extract file library pygame tadi, copy folder **“pygame”**  dan **“pygame-1.9.4.dist-info”** ke dalam:**C:\User\#username\AppData\Local\Program\Python\Python35-32\Lib\site-packages”**
6. **Cek hasil installasi**

Buka IDLE Python lalu lakukan perintah “import pygame”, jika tidak ada tulisan error maka pygame sudah berhasil terinstall.

BAB 2

**2.1.a Diagram Class Kivy**

|  |
| --- |
| Calculator(BoxLayout) |
| - |
| def backward(self, express)  def calculate(self, express) |

|  |
| --- |
| CalculatorApp(App) |
| - |
| def build(self)  def on\_pause (self) |

Code program kivy

**Main.py**

import kivy

kivy.require('1.0.6')

from kivy.app import App

from kivy.lang import Builder

from kivy.uix.widget import Widget

from kivy.uix.boxlayout import BoxLayout

#untuk close calculator

Builder.load\_string("""

<MyWidget>:

BoxLayout:

Button:

text: "Close"

on\_release: app.stop()

""")

class Calculator(BoxLayout): #tata letak kotak

def backward(self, express): # berfungsi untuk menghapus

if express:

self.display.text = express[:-1]

def calculate(self, express):# berfungsi untuk melakukan semua operasi yang ada di kalkulator

if not express: return

try:

self.display.text = str( eval(express) )

except Exception:# apabila operator di klik dengan sembarangan maka akan muncul tulisan eror

self.display.text = 'error'

class CalculatorApp(App):

title = 'calculator' #judul app

icon = 'icon.png' #gambar / icon app

def build(self):#untuk menampilkan boxlayout

return Calculator()

def on\_pause(self):

return True

if \_\_name\_\_ in ('\_\_main\_\_', '\_\_android\_\_'):# berfungsi untuk menjalankan class calculatorApp

CalculatorApp().run()

**Calculator.kv**

#:kivy 1.0

<Calculator>:

id: calculator

display: input

BoxLayout:

orientation: 'vertical'

padding: 20

spacing: 10

TextInput:

id: input

size\_hint: 1, None

readonly: True

multiline: False

font\_size: 50

GridLayout:

rows: 5

cols: 5

Button:

text: '1'

on\_press: input.text += self.text

Button:

text: '2'

on\_press: input.text += self.text

Button:

text: '3'

on\_press: input.text += self.text

Button:

text: '4'

on\_press: input.text += self.text

Button:

text: '5'

on\_press: input.text += self.text

Button:

text: '6'

on\_press: input.text += self.text

Button:

text: '7'

on\_press: input.text += self.text

Button:

text: '8'

on\_press: input.text += self.text

Button:

text: '9'

on\_press: input.text += self.text

Button:

text: '0'

on\_press: input.text += self.text

BubbleButton:

text: '+'

on\_press: input.text += self.text

BubbleButton:

text: '-'

on\_press: input.text += self.text

BubbleButton:

text: '\*'

on\_press: input.text += self.text

BubbleButton:

text: '/'

on\_press: input.text += self.text

BubbleButton:

text: 'C'

on\_press: calculator.backward(input.text)

BubbleButton:

text: '='

on\_press: calculator.calculate(input.text)

BubbleButton:

text: '^'

on\_press: input.text += self.text

Button:

text: "Close"

on\_press:app.stop()

**2.1.b Penjelasan Class dan Method**

**Builder.load\_string** = untuk close calculator

**class Calculator(BoxLayout):** tata letak kotak

**def backward(self, express):** berfungsi untuk menghapus

**def calculate(self, express):** berfungsi untuk melakukan semua operasi yang ada di kalkulator

**class CalculatorApp(App):** Class untuk menampilkan kalkulator

**def build(self):** untuk menampilkan boxlayout

**def on\_pause(self):** menampilkan , dengan return.

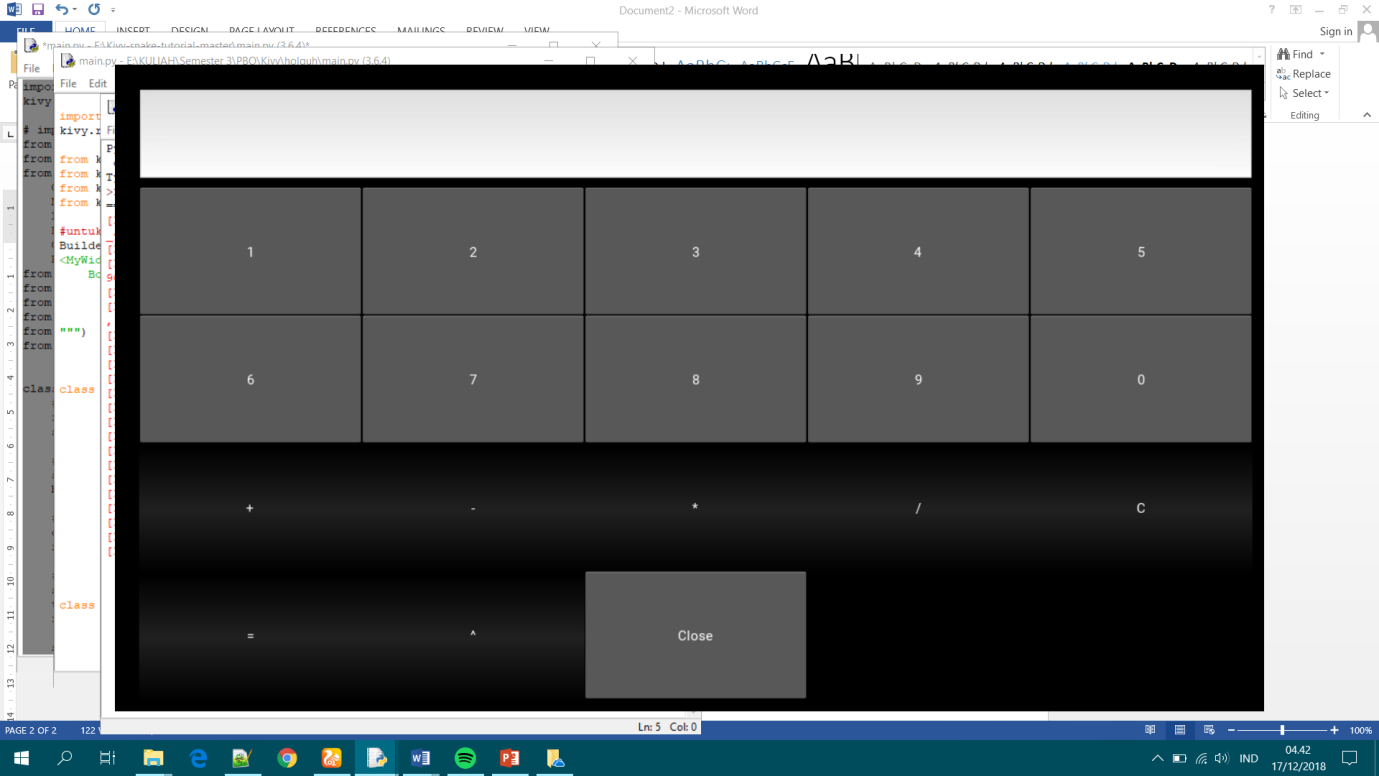
**2.1.c Implementasi OOP pada kivy**

|  |  |  |  |
| --- | --- | --- | --- |
| **CLASS** | **INHERITENCE** | **PHOLYMORPHISM** | **ENCAPSULATION** |
| Calculator(BoxLayout) | **-** | **-** | **-** |
| CalculatorApp(App) | **-** | **-** | **-** |

**2.1.d Aset Gambar**

**-**

**2.1.e Screen Tampilan**



Untuk angka 1 – 0 dan close menggunakan tombol yang timbul dan untuk operator tombol biasa .

Untuk pengoprasiannya :

Penjumlahan

Pengurangan

Perkalian

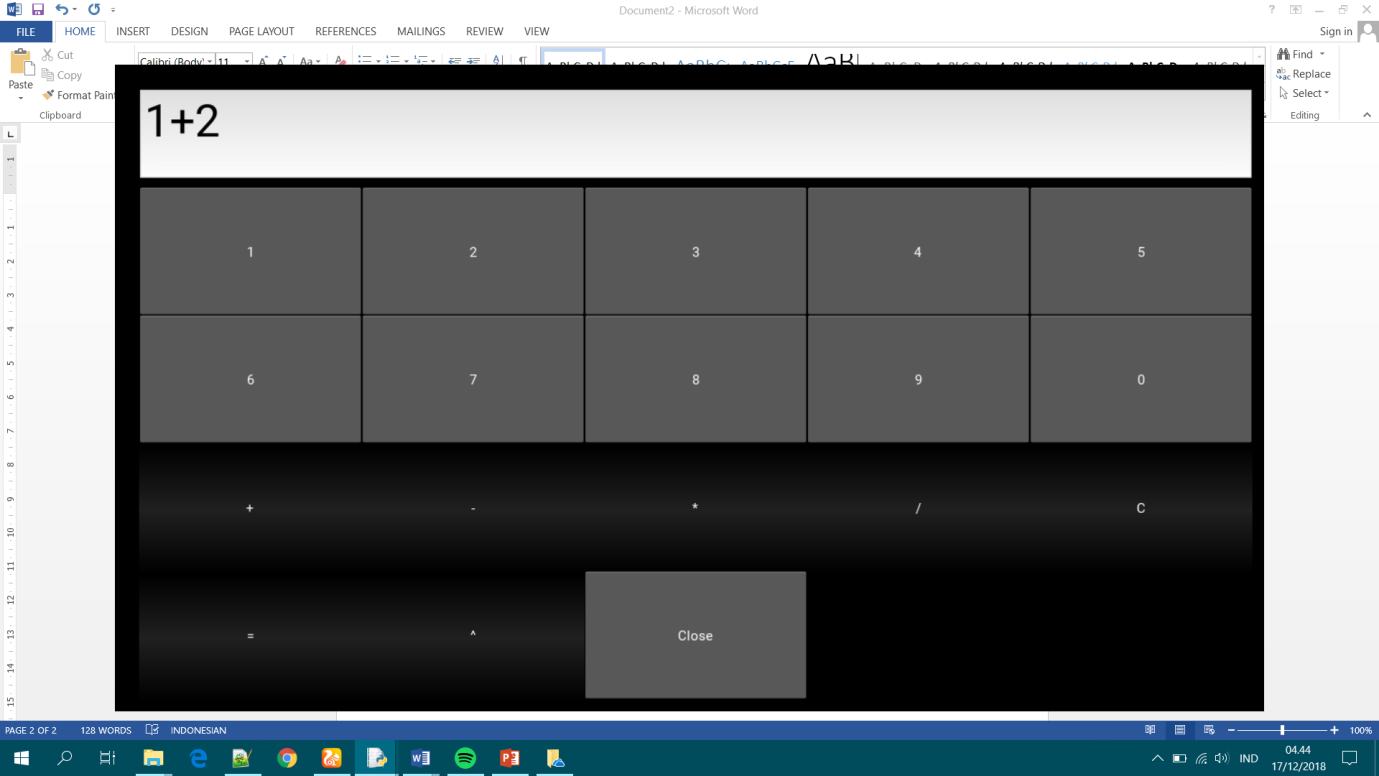
Pembagian

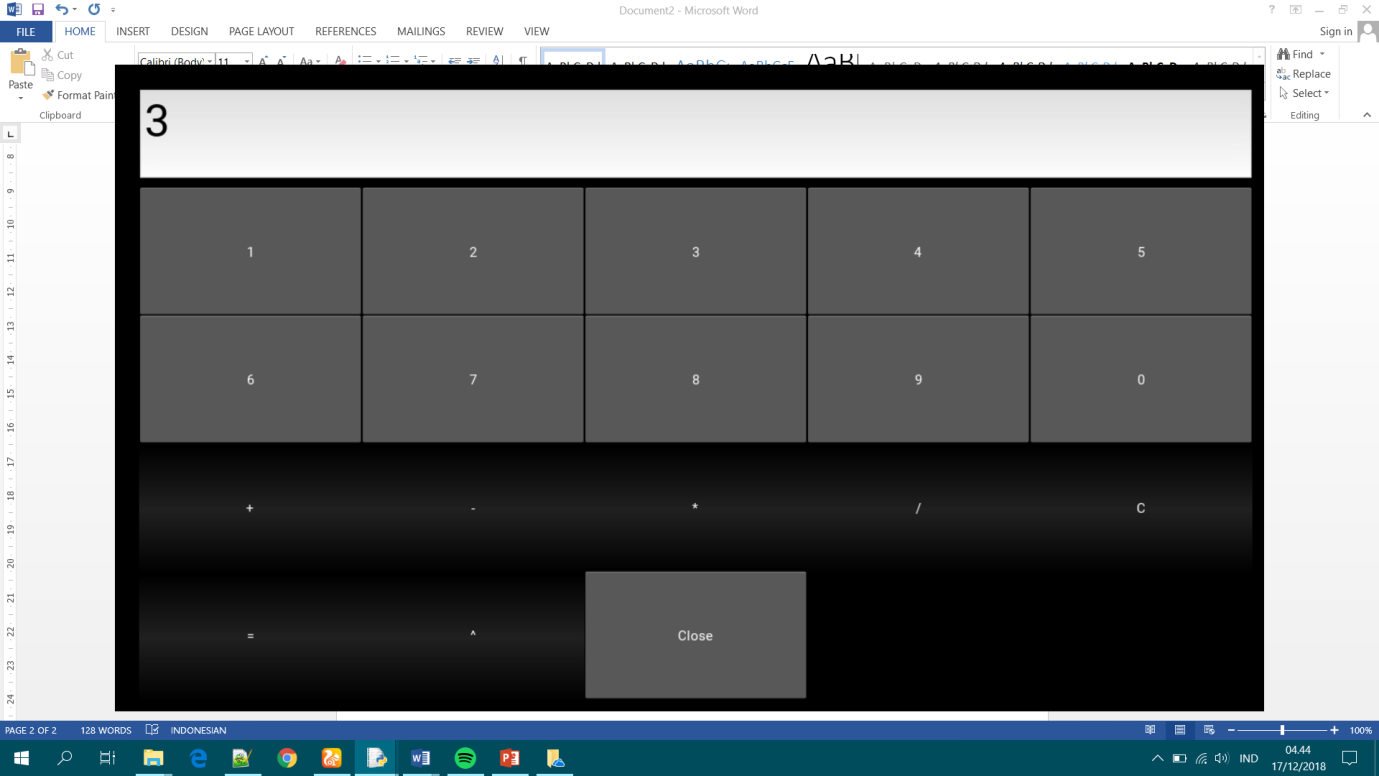
Hapus (menghapus satu persatu)

Sama dengan

Pangkat dan

Close (untuk menutup kalkulator)





**2.1.f Daftar pusaka**

[*http://programarcadegames.com/python\_examples/f.php?file=snake.py*](http://programarcadegames.com/python_examples/f.php?file=snake.py)

[*https://pythonspot.com/snake-with-pygame/*](https://pythonspot.com/snake-with-pygame/)

*youtube*

**2.2.a Diagram Class Pygame**

|  |
| --- |
| **class cube(object)** |
| **-** |
| **def \_\_init\_\_(self,start,dirnx=1,**  **dirny=0,color=(255,0,0))**  **def move(self,dirnx,dirny)**  **def draw(self,surface, eyes=False)** |

|  |
| --- |
| **class snake(object)** |
| **-** |
| **def \_\_init\_\_(self,color,pos)**  **def move(self)**  **def reset(self,pos)**  **def addcube(self)**  **def draw(self,surface)** |

**Method global**

|  |
| --- |
|  |
| **-** |
| **def redrawWindow(surface)**  **def randomsnack(rows,item)**  **def message\_box(subject,content)**  **def main()** |

**Code pygame**

import math

import random

import pygame

import sys

import tkinter as tk

from tkinter import messagebox

class cube(object):

rows = 20 #baris

w = 500 #lebar

def \_\_init\_\_(self,start,dirnx=1,dirny=0,color=(255,0,0)):

self.pos = start

self.dirnx = 1

self.dirny = 0

self.color = color

def move(self,dirnx,dirny): #perpindahan

self.dirnx = dirnx

self.dirny = dirny

self.pos = (self.pos[0] + self.dirnx, self.pos[1] + self.dirny)

def draw(self,surface, eyes=False): #menggambar ular ( permukaan ular,mata ular)

dis = self.w // self.rows

i = self.pos[0]

j = self.pos[1]

pygame.draw.rect(surface, self.color, (i\*dis+1,j\*dis+1,dis-2,dis-2))

if eyes:

center = dis//2

radius = 3

circleMiddle = (i\*dis+center-radius,j\*dis+8)

circleMiddle2 = (i\*dis+dis-radius\*2,j\*dis+8)

pygame.draw.circle(surface,(0,0,0), circleMiddle, radius)

pygame.draw.circle(surface,(0,0,0), circleMiddle2, radius)

class snake(object): #class untuk membuat objek ular

body = [] #menginisialisasikan badan ular

turns = {} #Untuk gerak belok dari si ular

def \_\_init\_\_(self,color,pos):

self.color = color

self.head = cube(pos)

self.body.append(self.head)

self.dirnx = 0

self.dirny = 1

def move(self): #untuk perpindahan

for event in pygame.event.get():

if event.type == pygame.QUIT:

pygame.quit()

keys = pygame.key.get\_pressed()

for key in keys: #pergerakan ular

if keys[pygame.K\_LEFT]: #gerak kekanan

self.dirnx = -1

self.dirny = 0

self.turns[self.head.pos[:]] = [self.dirnx, self.dirny]

elif keys [pygame.K\_RIGHT]: #gerak kekiri

self.dirnx = 1

self.dirny = 0

self.turns[self.head.pos[:]] = [self.dirnx, self.dirny]

elif keys [pygame.K\_UP]: #gerak keatas

self.dirnx = 0

self.dirny = -1

self.turns[self.head.pos[:]] = [self.dirnx, self.dirny]

elif keys [pygame.K\_DOWN]: #gerak ke bawah

self.dirnx = 0

self.dirny = 1

self.turns[self.head.pos[:]] = [self.dirnx, self.dirny]

for i,c in enumerate(self.body): #menambahkan count ke iterasi dan mengembalikanya (penghitung otomatis)

p = c.pos[:]

if p in self.turns:

turn = self.turns[p]

c.move(turn[0],turn[1])

if i == len(self.body)-1:

self.turns.pop(p)

else:

if c.dirnx == -1 and c.pos[0] <= 0: c.pos = (c.rows-1, c.pos[1])

elif c.dirnx == 1 and c.pos[0] >= c.rows -1: c.pos = (0,c.pos[1])

elif c.dirnx == 1 and c.pos[1] >= c.rows -1: c.pos = (c.pos[0],0)

elif c.dirnx == -1 and c.pos[0] <= 0: c.pos = (c.pos[0], c.rows-1)

else: c.move(c.dirnx,c.dirny)

def reset(self,pos):

self.head = cube(pos)

self.body = []

self.body.append(self.head)

self.turns = {}

self.dirnx = 0

self.dirny = 1

def addcube(self): #penambahan badan ular ketika ia makan

tail = self.body[-1]

dx,dy = tail.dirnx, tail.dirny

if dx == 1 and dy == 0:

self.body.append(cube((tail.pos[0]-1,tail.pos[1])))

elif dx == -1 and dy == 0:

self.body.append(cube((tail.pos[0]+1,tail.pos[1])))

elif dx == 0 and dy == 1:

self.body.append(cube((tail.pos[0],tail.pos[1]-1)))

elif dx == 0 and dy == -1:

self.body.append(cube((tail.pos[0],tail.pos[1]+1)))

self.body[-1].dirnx = dx

self.body[-1].dirny = dy

def draw(self,surface):

for i,c in enumerate(self.body):

if i ==0:

c.draw(surface,True)

else:

c.draw(surface)

def drawGrid(w,rows,surface): #ukuran grid pada layouyt permainan

sizebtwn = w // rows

x = 0

y = 0

for i in range(rows):

x = x + sizebtwn

y = y + sizebtwn

pygame.draw.line(surface,(255,255,255),(x,0),(x,w))

pygame.draw.line(surface,(255,255,255),(0,y),(w,y))

def redrawWindow(surface):

global rows , width,s,snack

surface.fill((0,0,0))

s.draw(surface)

snack.draw(surface)

drawGrid(width,rows,surface)

pygame.display.update()

def randomsnack(rows,item): #mengatur makanan secara random

positions = item.body

while True :

x = random.randrange(rows)

y = random.randrange(rows)

if len(list(filter(lambda z:z.pos == (x,y),positions)))>0:

continue

else:

break

return (x,y)

def message\_box(subject,content): #aturan untuk memunculkan suatu pesan masuk (massage box)

root = tk.Tk()

root.attributes("-topmost",True)

root.withdraw()

messagebox.showinfo(subject,content)

try:

root.destroy()

except:

pass

def main():

global width,rows,s,menu,snack

width = 500

rows = 20

win = pygame.display.set\_mode((width,width))

s = snake((255,0,0),(10,10))#posisi ular di awal

snack = cube(randomsnack(rows,s),color=(0,255,0))#makanan random

flag = True

clock = pygame.time.Clock()

while flag:

pygame.time.delay(50)

clock.tick(10)

s.move()

if s.body[0].pos == snack.pos: #saat ular makan

s.addcube()#tubuh ular akan bertambah

snack = cube(randomsnack(rows,s),color=(0,255,0))#makanan random

for x in range(len(s.body)):

if s.body[x].pos in list(map(lambda z:z.pos,s.body[x+1:])):

print("score:", len(s.body)) #menentukan score sesuai ular makan berapa kali

message\_box("kamu kalah",("SILAHKAN MAIN LAGI","\n score anda", len(s.body)))

#pesan masuk saat ular matimati

s.reset((10,10)) #jika main lagi ular akan mengulang di tempat yang awal

break

redrawWindow(win)

pass

main()

**2.2.b Penjelasan Class dan Method**

**class cube(object):** class untuk bentuk / menggambar tata letak kotak

**def move(self,dirnx,dirny):** untuk perpindahan

**def draw(self,surface, eyes=False):** menggambar ular ( permukaan ular, mata ular)

**class snake(object):** class untuk membuat objek ular

**def move(self):** untuk perpindahan

**def reset(self,pos):** mereset ulang permainan , dan kembali ke tempat semula

**def addcube(self):** penambahan badan ular ketika ia makan

**def draw(self,surface): berfungsi** sebagai gambar permukaan ular

**Method Global**

**def drawGrid(w,rows,surface):** menggambar tabel (ukuran grid pada layout permainan)

**def redrawWindow(surface):** mengupdate

**def randomsnack(rows,item):** mengatur makanan secara random

**def message\_box(subject,content):** aturan untuk memunculkan suatu pesan masuk (massage box)

**def main():** method utk menjalakan/memanggil seluruh class dan method

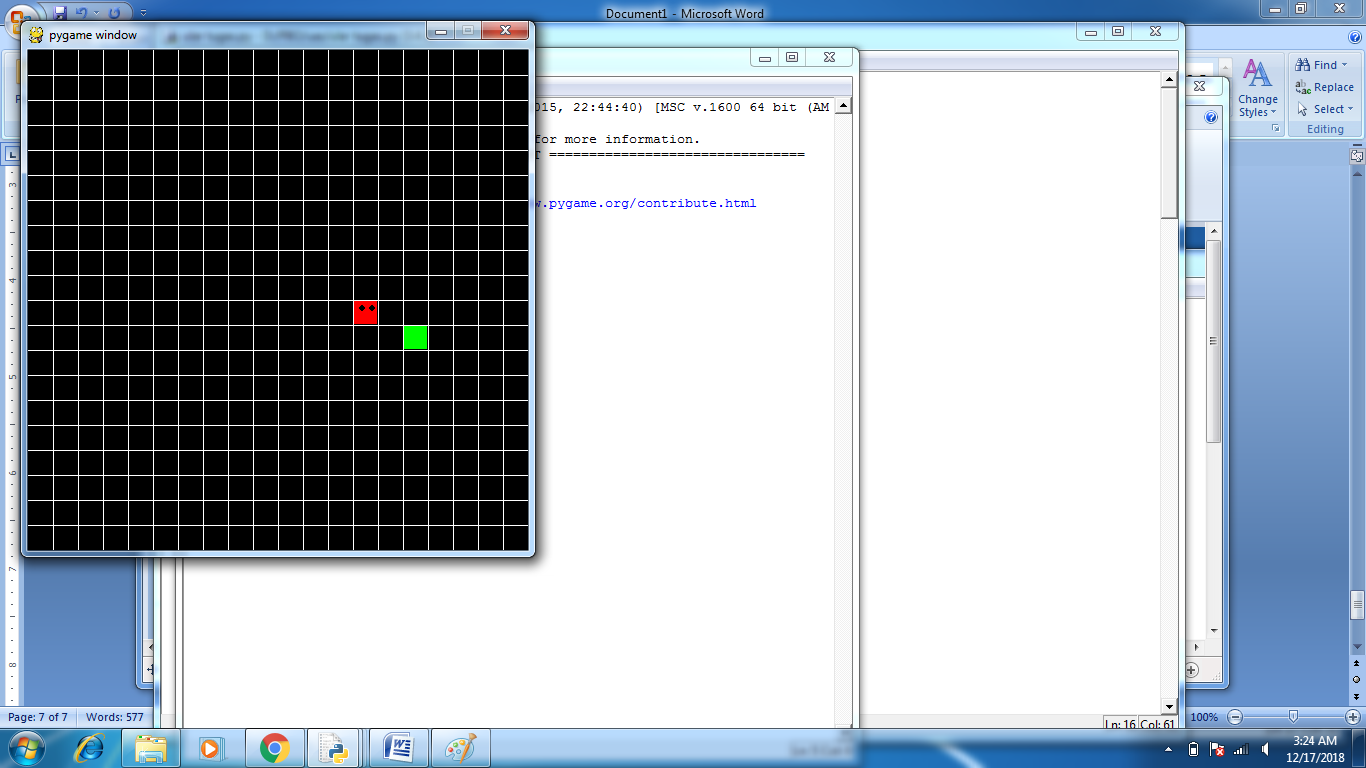
**2.2.c Implementasi OOP pada Pygame**

|  |  |  |  |
| --- | --- | --- | --- |
| **CLASS** | **INHERITENCE** | **PHOLYMORPHISM** | **ENCAPSULATION** |
| **CUBE** | **-** | **-** | **-** |
| **SNAKE** | **-** | **-** | **-** |

**2.2.d Aset Gambar**

**-**

**2.2.e Screen Tampilan**

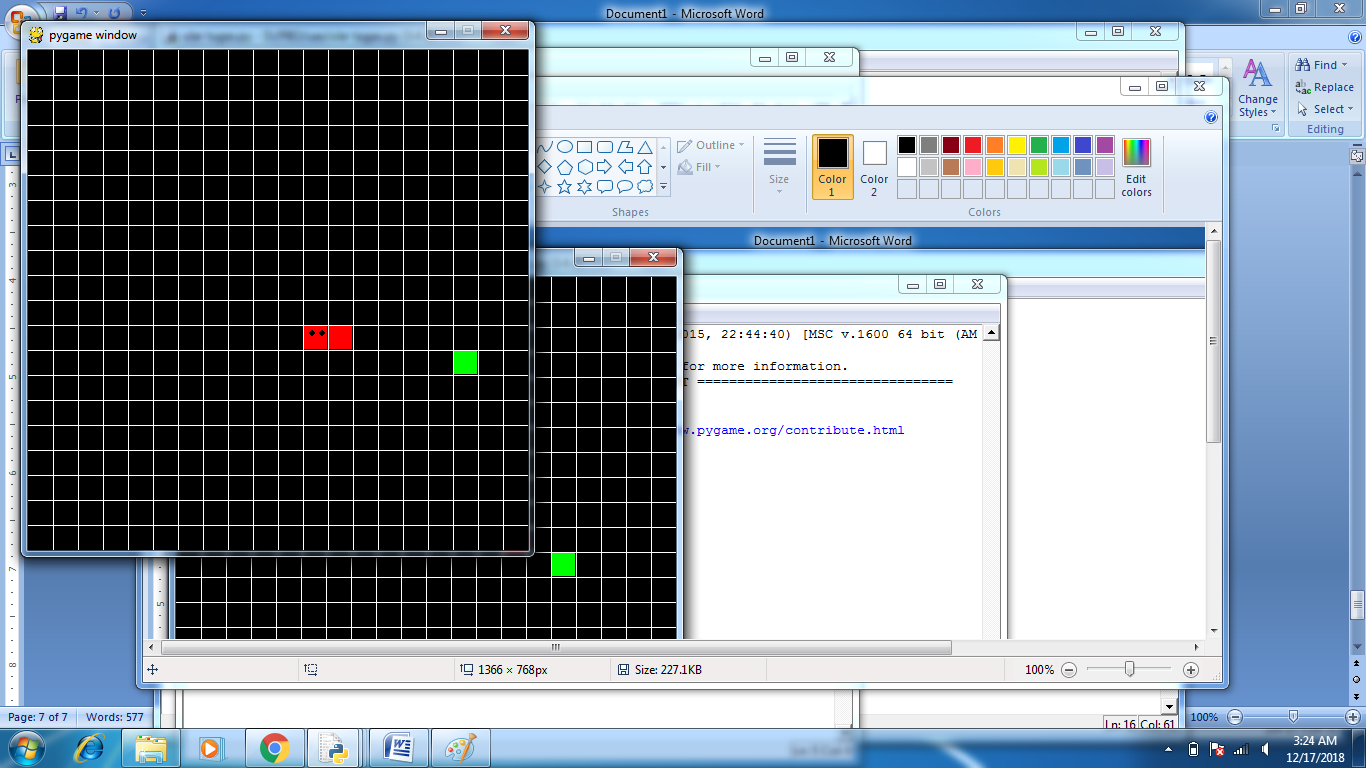


Ukuran grid pada layout backgroun ular 20 baris , dan untuk lebar nya 500

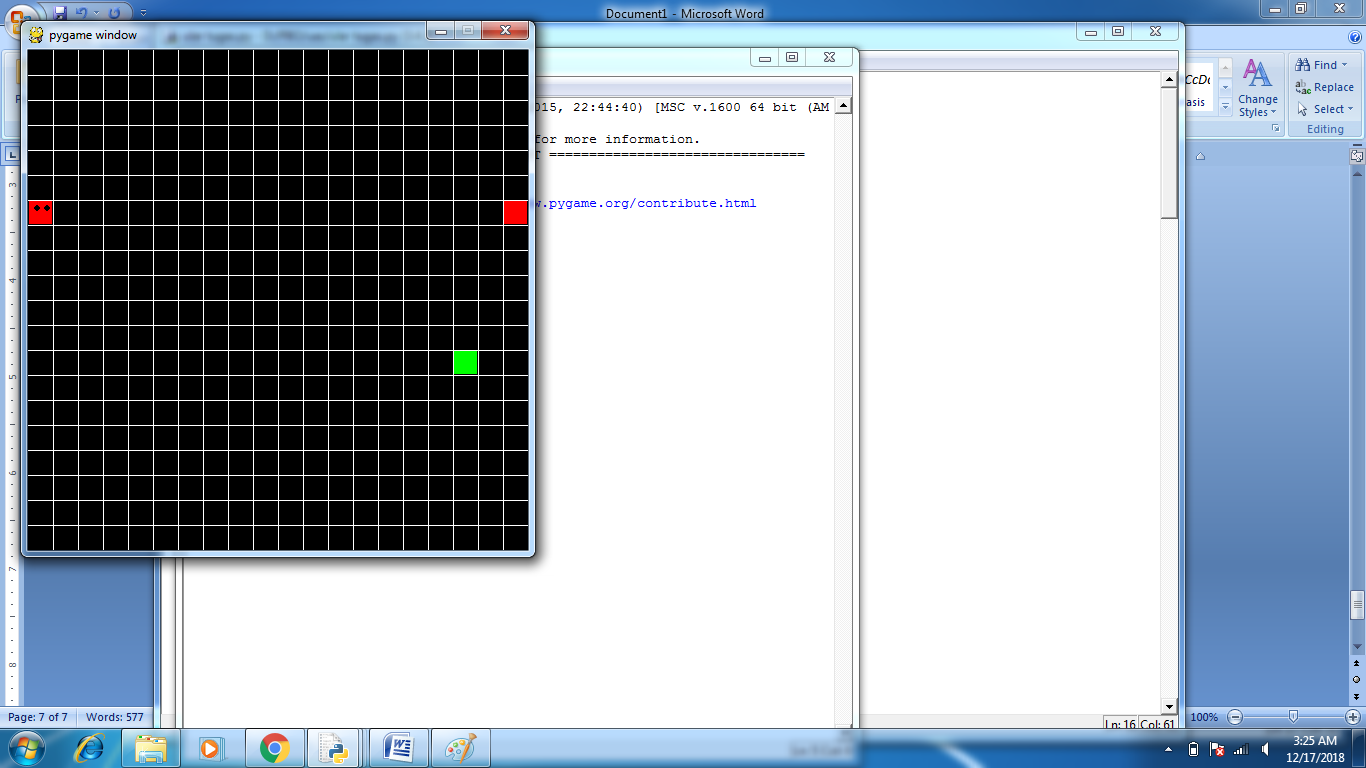
Untuk ular warna merah , dan makanan warna hijau.

Ular di awal di posisi (10,10) lalu berjalan ke kanan . ular dapat bergerak ke atas kebawah ka kanan ke kiri.

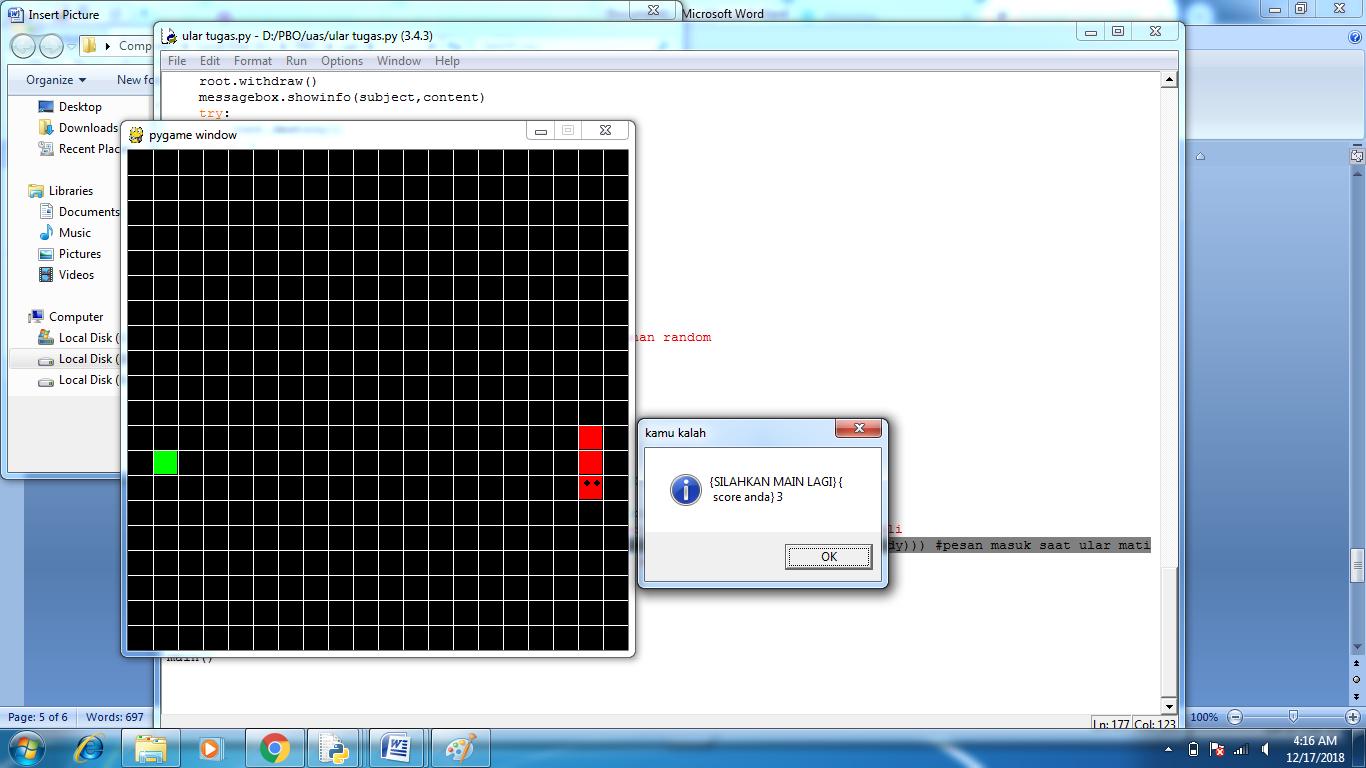
Untuk letak makanan random.



Untuk ular saat makan , tubuhnya akan bertambah panjang sesuai seberapa banyak yang telah iya makan



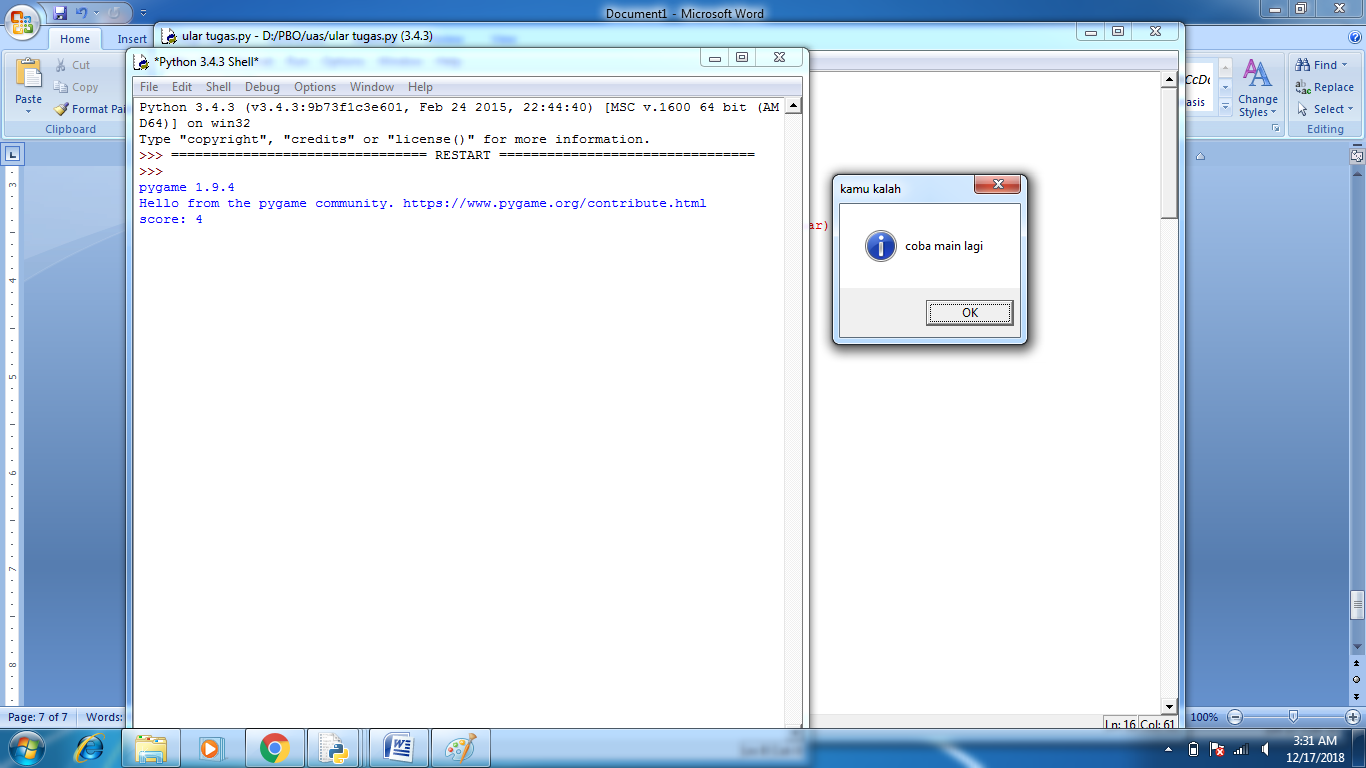
Ular bergerak dan tembus dari kanan ke kiri , atas ke bawah dan sebaliknya



Ular akan mati jika memakan dirinya sendiri , dan muncul pesan (massage box) “silahkan main lagi”

Dan menampilkan score yang telah di dapat.

Jika klik “ok” maka permainan akan me reset dari awal , ular akan di posisi semula dengan bentuk semula.



menghitung banyak skor makanan yang telah di makan oleh ular.

**2.2.f Daftar Pusaka**

[*http://programarcadegames.com/python\_examples/f.php?file=snake.py*](http://programarcadegames.com/python_examples/f.php?file=snake.py)

[*https://pythonspot.com/snake-with-pygame/*](https://pythonspot.com/snake-with-pygame/)

*youtube*