DOKUMEN SPESIFIKASI PROGRAM GAME ULAR BlackAs
PEMROGRAMAN BERORIENTASI OBJECT

Disusun untuk memenuhi tugas besar mata kuliah PBO

Semester 4 Tahun Ajaran 2020/2021

Disusun oleh :

Akmal Fauzan Suranta	119140203
Fahriza Yusefa	119140005
Hendamia Yohana Sembiring	119140178
Ikhsanudin Raka Siwi	119140058

Kelompok :

BlackAs

Dosen Pembimbing :

Muhammad Habib Algifari, S.Kom., M.T.I.

Mata Kuliah :

IF2222 Pemrograman Berorientasi Objek

Program Studi :

Teknik Informatika

INSTITUT TEKNOLOGI SUMATERA

JURUSAN TEKNIK ELEKTRO, INFORMATIKA DAN SISTEM FISIKA

LAMPUNG SELATAN

2021

A. Latar Belakang

Era globalisasi dan teknologi saat ini, penggunaan komputer sebagai alat teknologi informasi, keberadaannya sangat diperlukan pada semua aspek kehidupan. Biasanya, penggunaan komputer sebagai alat bantu untuk pengelolaan dan pengolahan data. sehingga penggunaan perangkat komputer dalam setiap informasi sangat mendukung sistem dalam pengambilan keputusan. Akan tetapi, pada era globalisasi sekarang ini komputer bukan digunakan hanya untuk itu saja, tetapi juga digunakan untuk hiburan seperti contohnya bermain game.

Game merupakan salah satu produk teknologi informasi yang sangat digemari pada saat ini, khususnya pada kalangan remaja. Game juga merupakan sebuah aplikasi yang edukatif, yang artinya game tersebut bisa dijadikan sebagai media pembelajaran. Game biasanya memiliki tingkat kesulitan yang dimulai dari tingkat kesulitan yang berawal dari tingkat kesulitan yang mudah, sedang, hingga sulit. Derajat kesulitan yang muncul menunjukkan adanya perubahan level pada game. Semakin tinggi level, maka semakin tinggi derajat kesulitannya.

Game snake adalah sebuah permainan yang tidak terlalu sulit dan tidak harus memerlukan pemikiran yang keras, akan tetapi hanya memerlukan sedikit ketelitian dalam memainkan game ini. Pada game ini prosesnya dengan menggerakkan snake untuk dapat memakan makanan dengan tujuan akhir yaitu untuk mendapatkan point. Pada game ini derajat kesulitannya yang kami buat terletak pada saat ular memakan sebanyak 10 makanan sehingga kecepatan ular tersebut bertambah.

B. Tujuan Pembuatan Game

Adapun pembuatan game yang berbasis python dengan konsep OOP tersebut dibuat dengan tujuan sebagai berikut :

a. Memahami pembelajaran mengenai *object oriented programming*.
b. Meningkatkan skill pemrograman dalam membuat game.
c. Pemenuhan tugas besar yang diberikan oleh dosen.

C. Deskripsi Game

Game ular yang dibuat oleh kelompok kami menggunakan bahasa pemrograman python dengan konsep *object oriented programming*. Berbekalkan materi pemrograman python yang telah diberikan, kelompok kami membuat program ular yang memiliki beberapa fitur, diantaranya terdapat pilihan menu di awal game (permainan baru dan keluar), menampilkan objek ular yang warna badannya berbeda dengan warna kepalanya, game selesai dan muncul tampilan “game over” saat dinding ditabrak ataupun saat ular menabrak dirinya sendiri, kecepatan ular meningkat di setiap kali memakan 10 makanan dan terdapat fitur bonus makanan di setiap 5 kali makan dengan rentang waktu 5 detik.

LINK Repository : https://github.com/ouuzannn/BlackAs---Game-Ular.git

D. Spesifikasi Program

1. Bisa menampilkan objek papan dari kelas papan
2. Bisa menampilkan objek menu dari kelas menu
3. Menu game berisi “permainan baru” dan “keluar”, jika memilih permainan baru maka game akan di reset ke kondisi awal dan jika memilih menu “keluar” program akan terhenti
4. Bisa menampilkan objek ular dari kelas ular
    a. Kepala ular ditampilkan berbeda dengan badan
    b. Ular bertambah panjang ketika ular berhasil memakan makanan
    c. Kecepatan ular bertambah setiap memakan 10 makanan
5. Bisa menampilkan objek makanan dari kelas makanan
    a. Posisi makanan harus didalam papan permainan
    b. Posisi makanan diacak
    c. Menampilkan makanan bonus setelah ular memakan makanan sebanyak 5 Kali
    d. Makanan bonus muncul selama 5 detik
6. Bisa menampilkan objek nilai dari kelas nilai
    a. Skor bertambah ketika ular berhasil memakan makanan
    b. Mendapat nilai +1 untuk setiap makanan
    c. Mendapat nilai +5 untuk setiap makanan bonus
7. Bisa menghentikan game saat memilih menu permainan baru atau keluar
8. Game berakhir saat ular menabrak batas luar papan
9. Game berakhir saat ular menabrak badan ular itu sendiri
10. Menampilkan pesan tambahan “permainan berakhir” ketika game berakhir

E. Penjelasan Program Game

Pada program ini kami menggunakan konsep *object oriented programming* (OOP) dimana dalam program ini kita perlu mengimpor pustaka ke dalam kode yang akan digunakan untuk membuat game ular ini kami mengimport library pygame dan pada main program (file main.py) kami menggabungkan tiga file yaitu MenuGame.py (untuk tampilan menu), TombolMenu.py (sebagai respon GUI pada tampilan menu), konstanta.py (untuk pemilihan warna pada GUI dan tampilan menu). Berikut penjelasan setiap file secara terurut :

- **Konstanta.py**

Pada file konstanta disini berisi beberapa konstanta yang mana digunakan untuk ketetapan warna yang akan digunakan pada game ini. Berikut nilai konstanta sebagai ketetapan warna.

LINK Repository : https://github.com/ouuzannn/BlackAs---Game-Ular.git

```
# set up the colors
BLACK = (0, 0, 0)
BLUE = (0, 0, 255)
GRAY = (100, 100, 100)
WHITE = (255, 255, 255)
RED = (255, 0, 0)
GREEN = (0, 255, 0)
```

Tidak hanya warna, di konstanta ini juga menetapkan ukuran dari layar jendela gamenya, dimana ditetapkan tingginya 400 dengan lebar 500. Berikut nilai konstantanya.

```
# set up the window
WINDOWWIDTH = 500
WINDOWHEIGHT = 400
```

❖ **TombolMenu.py**

Pada file TombolMenu ini berisi tentang bagaimana cara pembuatan tombol menu New Game dan quit. Pada pembuatan tombol ini kita perlu mengimport pustaka ke dalam kode yang mana kita perlu mengimport pada file konstanta yang dapat dilihat pada gambar di bawah ini :

```
import pygame
from konstanta import *
```

- **Class PyButton**

Pada TombolMenu terdapat sebuah kelas PyButton yang dapat membuat dan menggambarkan tombol ke layar serta dapat mengubah warna latar belakang. Pada kelas ini terdapat beberapa method yang ada di dalamnya.

Berikut penjelasan dari method yang ada pada kelas PyButton :

1. **Methods __init__**

Methods ini berfungsi untuk mendeklarasikan untuk membuat objek tombol dengan atributnya yang dapat dilihat pada gambar berikut :

```
class PyButton(object):
  def __init__(self, x, y, text, font="Krungthep", height=20, textcolor=WHITE, bgcolor=BLACK, hovercolor=GRAY):
    self.x, self.y = x, y
    self.FontHuruf = font
    self.height = height
    self.text = text
    self.textcolor = textcolor
    self.color = bgcolor
    self.bgcolor = bgcolor
    self.hovercolor = hovercolor
    self.surf = self._renderteks()
    self.rect = self._MembuatPersegiTeks()
    self.selected = True
    self.active = True
```

2. **Methods _renderteks**

Methods ini berfungsi untuk merender tombol teks.

LINK Repository : https://github.com/ouuzannn/BlackAs---Game-Ular.git

```
def _renderteks(self):
    FontHuruf = pygame.font.SysFont(self.FontHuruf, self.height)
    surface = FontHuruf.render(self.text, True, self.textcolor, self.bgcolor)
    return surface
```

3. **Methods _MembuatPersegiTeks**

   Methods ini berfungsi untuk membuat persegi panjang untuk permukaan teks.

```
def _MembuatPersegiTeks(self):
    button_rect = self.surf.get_rect()
    button_rect.center = (self.x, self.y)
    return button_rect
```

4. **Methods draw**

   Methods ini berfungsi untuk membuat tombol di permukaan layar.

```
def draw(self, surface):
    surface.blit(self._renderteks(), self._MembuatPersegiTeks())
```

5. **Methods setActive**

   Methods ini berfungsi untuk mengatur tombol menjadi aktif.

```
def setActive(self):
    self.active = True
```

6. **Methods setInactive**

   Methods ini berfungsi untuk mengatur tombol menjadi tidak aktif.

```
def setInactive(self):
    self.active = False
```

7. **Methods setHighlighted**

   Methods ini berfungsi untuk mengatur kursor mouse jika di arahkan pada tombol new game dan quit maka hal interaktif yang terjadi terdapat transisi warna yang berubah.

```
def setHighlighted(self):
    self.bgcolor = self.hovercolor
```

8. **Methods setNormal**

   Methods ini berfungsi untuk mengatur warna layar kembali ke warna normal.

```
def setNormal(self):
    self.bgcolor = self.color
```

9. **Methods isHovered**

   Methods ini berfungsi untuk membuat GUI pada setiap tombol seperti new game dan quit menjadi interaktif (hovered).

LINK Repository : https://github.com/ouuzannn/BlackAs---Game-Ular.git

```
def isHovered(self):
    return self.active and self.rect.collidepoint(pygame.mouse.get_pos())
```

❖ **MenuGame.py**

Pada file MenuGame ini berisi tentang pembuatan tampilan dari menu game. Dimana file ini perlu mengimport file pygame, TombolMenu, sys, pygame.locals, dan konstanta. Hal ini dapat terlihat pada gambar dibawah ini.

```
import pygame, TombolMenu, sys
from pygame.locals import *
from konstanta import *
```

- **Class PyMenu**

Pada file MenuGame terdapat kelas PyMenu yang dapat membuat tampilan dari game terlihat menarik, baik dari pewarnaan tombol dan layar, serta di kelas ini terdapat beberapa methods yang berfungsi untuk mengecek inputan dari mouse (klik pada tombol game), keyboard, dan masukan akan pilihan menu dari pengguna. Berikut penjelasan dari setiap methods yang ada.

1. **Methods __init__ :**

Pada methods ini, digunakan dalam menginisialisasi menu judul, tombol, perintah, dan warna dengan parameter x=0, y=0, judul=" " yang berarti default.

```
import pygame, TombolMenu, sys
from pygame.locals import *
from konstanta import *

class PyMenu(object):
  def __init__(self, menuwarna, x=0, y=0, judul="", font="Krungthep"):
    # menu judul, Tombol and perintah
    self.namaTombol = TombolMenu.PyButton(x, y, judul, font)
    self.Tombol = []
    self.perintah = []

    # menu indeks and color
    self._indeks = 1
    self._indeksMaks = len(self.Tombol)-1
    self.menuwarna = menuwarna
```

2. **Methods tampilan :**

Methods ini berfungsi untuk mengatur tampilan pada menu game diantaranya mengecek pilihan menu user, mengatur warna background, tampilan judul, tampilan tombol, tampilan warna sorotan tombol (dimana warnanya akan muncul saat kursor diarahkan pada tombol menu yang ada) serta terdapat fungsi pygame.display.update() untuk mengupdate atau mengaktifkan semua pengaturan yang telah disebutkan tadi.

```
def tampilan(self, latar):
    while True:
        # check for events
        self._checkEvents()
        # tampilan background
        latar.fill(self.menuwarna)
        # tampilan judul
        if self.namaTombol.text != "":  #teks
            self.namaTombol.draw(latar) #tampilan

        # tampilan Tombol
        for tombol in self.Tombol:
            tombol.draw(latar)

        # highlight Tombol
        for tombol in self.Tombol:
            if tombol.selected: #pilih
                tombol.setHighlighted()
            else:
                tombol.setNormal()

        # update menu
        pygame.display.update() #layar
```

3. **Methods tambahkanTombol :**

Pada methods ini, berfungsi untuk menambahkan tombol yang terletak pada menu game.

```
def tambahkanTombol(self, tombol, command):
    self.Tombol.append(tombol)
    self.perintah.append(command)
    self._indeksMaks = len(self.Tombol)-1
```

4. **Methods checkEvents :**

Pada methods ini, berfungsi untuk mengecek event yang dipilih oleh user pada game. Event pada game disini ada tiga yaitu saat memulai permainan baru (tombol menu), saat memainkan game dengan keyboard dan saat ingin mengakhiri permainan.

LINK Repository : https://github.com/ouuzannn/BlackAs---Game-Ular.git

```
#Cek perintah pada menu
  def _checkEvents(self):
    for event in pygame.event.get():
      # check if player quits
      self._cekQuit(event)
      # check for keyboard events
      self._cekKeyboard(event)
      # check for tombol events
      self._cekMouse(event)
```

5. Methods checkKeyboard :

Methods ini berfungsi untuk mengecek inputan yang didapat dari keyboard oleh pengguna saat memainkan permainan khususnya pada tampilan menu. Untuk mengecek masukkan keyboard terdapat percabangan if dengan kondisi yaitu event.type == KEYDOWN yang berarti apabila tombol ditekan, maka akan masuk ke dalam percabangan selanjutnya dengan kondisi event.key == K_UP or event.key == ord('w') apabila tombol yang ditekan itu tombol atas atau keyword ‘w’ maka self.indeks akan berkurang 1, tetapi jika self.indeks sebelumnya bernilai kurang dari 0 maka self.indeks dijumlahkan dengan self.indeksMaks+1. Jika tidak sesuai dengan kondisi pada percabangan sebelumnya, maka masuk ke dalam percabangan selanjutnya dengan kondisi event.key == K_DOWN or event.key == ord('s') yang berarti apabila tombol yang ditekan itu tombol bawah atau keyword ‘s’ maka self.indeks akan bertambah 1, tetapi jika self.indeks sebelumnya bernilai lebih dari self.indeksMaks, maka self.indeks dikurangi dengan self.indeksMaks+1.

```
def _cekKeyboard(self, event):
  """ Check perintah keyboard """
  # tombolBawah events
  if event.type == KEYDOWN:
    if event.key == K_UP or event.key == ord('w'):
      self._indeks -= 1
      if self._indeks < 0:
        self._indeks += (self._indeksMaks+1)
    if event.key == K_DOWN or event.key == ord('s'):
      self._indeks += 1
      if self._indeks > (self._indeksMaks):
        self._indeks -= (self._indeksMaks+1)
    if event.key == K_RETURN:
      self._perintahJalan(self._indeks) # call tombol action
    # tombol selection
    for tombol in self.Tombol:
      if tombol == self.Tombol[self._indeks] and tombol.active:
        tombol.selected = True
      else:
        tombol.selected = False
```

LINK Repository : https://github.com/ouuzannn/BlackAs---Game-Ular.git

6. **Methods _cekMouse :**

Methods ini berfungsi untuk mengecek inputan didapat melalui mouse (*mode click*). Inputan disini berasal dari user pada tampilan menu awal yang mana terdapat dua pilihan (tombol menu) yaitu '*new game*' dan '*quit*'. Di methods ini nantinya akan mengecek keadaan mouse, apabila mouse digerakkan (if event.type == MOUSEMOTION:) saja maka tampilan dari tombol menu yang dikenai oleh kursor akan tersorot warna abu-abu (menghighlight nama tombol) yang didapat dari fungsi tombol.isHovered() dan jika tidak terkena oleh kursor maka tidak terjadi apapun (self.Tombol[i] = False). Sedangkan apabila mouse digerakkan ke arah tombol menu dan diklik (if event.type == MOUSEBUTTONUP:) maka tombol kursor akan tersorot warna abu-abu dan langsung menjalankan aksi self._perintahJalan(self._indeks).

```
def _cekMouse(self, event):
    """ Check perintah pada mouse """
    # mouse motion events
    if event.type == MOUSEMOTION:
        for i in range(0, (self._indeksMaks+1)):
            if self.Tombol[i].isHovered():
                self.Tombol[i].selected = True
                self._indeks = i
            else:
                self.Tombol[i].selected = False
    # mouse click events
    if event.type == MOUSEBUTTONUP:
        #0 = new game
        #1 = exit
        for tombol in self.Tombol:
            if tombol.isHovered():
                self._perintahJalan(self._indeks) # call tombol action
```

7. **Methods cekQuit :**

Methods ini berfungsi untuk mengecek apakah pemain ingin keluar dari game. Pada proses ini ada dua cara untuk keluar dari game, dimana yang pertama yaitu jika pemain tersebut ingin keluar dari permainan maka dapat dengan cara mengklik tombol quit atau dengan menekan tombol escape. Apabila pemain mengklik atau menekan tombol tersebut, otomatis permainan akan langsung menghentikan program karena perintah dari module pygame dan sys yaitu pygame.quit dan sys.exit.

```
def _cekQuit(self, event):
  """ Checks kalau user keluar """
  if event.type == QUIT:
    pygame.quit()
    sys.exit()
  if event.type == KEYDOWN:
    if event.key == K_ESCAPE:
      pygame.quit()
      sys.exit()
```

8. **Methods _perintahjalan :**

Methode ini berfungsi untuk memanggil fungsi yang sesuai dengan indeks.

```
def _perintahJalan(self, indeks):
  """ Calls the function corresponding to the indeks. """
  return self.perintah[indeks](indeks)
```

❖ **Main.py (Program Utama Aplikasi)**

Import pada main program dapat dilihat pada gambar di bawah ini :

```
import pygame
import sys
import random
import TombolMenu,MenuGame
from konstanta import *
```

Pada main program terdapat 4 class yaitu class Ular, Makanan, MakananBonus, kotak.

- **Class Ular :**

1. **Methods __init__ :**

Berfungsi untuk mendeklarasi atribut untuk objek ular yang dapat dilihat pada gambar di bawah ini:

```
class Ular():
    def __init__(self):
        self.PanjangUlar = 1
        self.letakUlar = [((lebar_layar/2), (tinggi_layar/2))]
        self.arahUlar = random.choice([atas, bawah, kiri, kanan])
        self.nilai = 0
        self.kecepatan = 5
        self.hitungmakanan = 0
```

2. **Methods PosisiKepalaUlar :**

Berfungsi untuk mengetahui posisi letak kepala ular dimana self.letakUlar[0] itu menandakan bahwa kepala ular berada pada posisi ke 0 pada badan ular dari keseluruhan gambar ular .

```
def PosisiKepalaUlar(self) :
    return self.letakUlar[0]
```

3. **Methods turn :**

Selanjutnya, pada method def turn untuk proses pergerakan/perpindahan ular.

LINK Repository : https://github.com/ouuzannn/BlackAs---Game-Ular.git

```
def turn(self, point):
    if self.PanjangUlar > 1 and (point[0]*-1, point[1]*-1) == self.arahUlar:
        return
    else :
        self.arahUlar = point
```

4. **Methods PergerakanUlar :**

Pada method PergerakanUlar ini, dimana variabel cur diinisiasi oleh methods PosisiKepalaUlar dan new = (((cur[0]+(x*gridsize))%lebar_layar), (cur[1]+(y*gridsize))%tinggi_layar) untuk menghitung lokasi baru kepala ular. Pada bagian percabangan, jika kepala ular lebih besar dari 2 dimana kepala ular menabrak badan ular maka dalam hal ini berarti permainan berakhir. Dan jika kepala ular menabrak pembatas papan, sehingga dalam hal ini permainan juga berakhir. Karena permainan berakhir, maka dalam hal ini permainan akan di atur ulang dengan memulihkan ke nilai awal yang terdapat pada metode reset. Selain dari kedua kondisi itu maka ular pun akan terus bergerak.

```
def PergerakanUlar(self) :
    cur = self.PosisiKepalaUlar()
    x,y = self.arahUlar
    new = (((cur[0]+(x*gridsize))%lebar_layar), (cur[1]+(y*gridsize))%tinggi_layar)
    if (len(self.letakUlar) > 2 and new in self.letakUlar[2:]) :
        #akhir()
        cek=0
        self.reset()
        Menu(cek)
    elif cur[0] >= (lebar_layar-20) or cur[0] <= 0 or cur[1] >= (tinggi_layar-20) or cur[1] <= 0:
        #akhir()
        cek=0
        self.reset()
        Menu(cek)
    else :
        self.letakUlar.insert(0,new)
        if len(self.letakUlar) > self.PanjangUlar :
            self.letakUlar.pop()
```

5. **Methods reset :**

Selanjutnya metode def reset ini untuk mengatur ulang dengan memulihkan ke nilai default/awal.

```
def reset(self) :
    self.PanjangUlar = 1
    self.letakUlar = [((lebar_layar/2), (tinggi_layar/2))]
    self.arahUlar = random.choice([atas, bawah, kiri, kanan])
    self.nilai = 0
    self.kecepatan = 5
    self.hitungmakanan = 0
```

6. **Methods GambarUlar :**

Pada def draw ini untuk menggambarkan blok/kepala, dimana kepala ular tersebut terletak pada titik 0, r = pygame.rect berfungsi untuk membentuk ular dengan menyesuaikan grid. Jika PenandaKepala == 0, maka warna kepalanya hitam dengan garis tepi biru. Sedangkan kalau PenandaKepala !=0

(menunjukkan penggambaran badan) maka badan ular tersebut akan digambarkan dengan menyesuaikan grid dengan warna badannya adalah merah dengan garis tepi biru.

```
def GambarUlar(self, surface):
    PenandaKepala = 0
    for p in self.letakUlar:
        bagianular = pygame.Rect((p[0], p[1]), (gridsize,gridsize))
        if PenandaKepala == 0 :#penanda kepala
            self.warna = (25, 24, 47)
            pygame.draw.rect(surface, self.warna, bagianular)
        else :
            self.warna = (255, 24, 47)
            pygame.draw.rect(surface, self.warna, bagianular)
        pygame.draw.rect(surface, (25,24,228), bagianular, 1)
        PenandaKepala +=1
```

7. **Methods KontrolUlar**

Pada def KontrolUlar ini sebagai tombol penggerak, jika pemain ingin keluar dari permainan maka permainan akan langsung menghentikan program karena perintah dari module pygame dan sys yaitu pygame.quit dan sys.exit. Selain itu pergerakan/perpindahan ular dapat menggunakan tombol atas, bawah, kiri, kanan dimana disinilah terlihat methods turn berfungsi untuk mengembalikan hasil dari inputan keyboard(atas,bawah,kanan,kiri).

```
def KontrolUlar(self):
    for event in pygame.event.get():
        if event.type == pygame.QUIT :
            pygame.quit()
            sys.exit()
        elif event.type == pygame.KEYDOWN :
            if event.key == pygame.K_UP:
                self.turn(atas)
            elif event.key == pygame.K_DOWN :
                self.turn(bawah)
            elif event.key == pygame.K_LEFT :
                self.turn(kiri)
            elif event.key == pygame.K_RIGHT :
                self.turn(kanan)
```

- **Class Makanan**

1. **Methods __init__ :**

Selanjutnya terdapat class makanan, dimana def __init__ merupakan deklarasi variabel, dengan posisi ular di titik 0, warna makanan oren, dan posisi makanan acak

LINK Repository : https://github.com/ouuzannn/BlackAs---Game-Ular.git

```
def __init__(self):
    self.letakUlar = (0,0)
    self.warna = (223, 163, 49)
    self.PosisiAcakMakanan()
```

2. **Methods PosisiAcakMakanan**

Selanjutnya pada def PosisiAcakMakanan ini posisi makanan acak dengan bentuk makanan menyesuaikan grid/kotak dengan menggunakan function randint yang merupakan bawaan dari python/pygame. .

```
def PosisiAcakMakanan(self):
    self.letakUlar = (random.randint(1, grid_width-2)*gridsize, random.randint(1, grid_height-2)*gridsize)
```

3. **Methods gambarObjek**

Selanjutnya, pada bagian ini kita perlu menggambarkan objek makanan di permukaan layar, proses ini hampir sama dengan metode draw yang digunakan pada class ular. Akan tetapi kita hanya perlu menggambar satu blok untuk posisi makanan tersebut dengan mengikuti pola grid.

```
def gambarObjek(self, surface) :
    gambarmakanan = pygame.Rect((self.letakmakanan[0], self.letakmakanan[1]), (gridsize, gridsize))
    pygame.draw.rect(surface, self.warna, gambarmakanan)
    pygame.draw.rect(surface, (93, 216, 228), gambarmakanan, 1)
```

- **Class MakananBonus**

Selanjutnya terdapat class makanan bonus, dimana pada makanan bonus ini posisi ular pada titik 0, letak makanan bonus acak, dan makanan bonus tersebut berwarna hijau dengan gambarObjek makanannya lebih besar daripada grid.

```
class MakananBonus() :
    def __init__(self):
        self.LetakMakananBonus = (0,0)
        #self.warna = [(25, 163, 49), (0,0,0)]
        self.PosisiAcakMakanan()
        self.timer = 0
        self.waktubonus = 25
        self.tanda = 0

    def PosisiAcakMakanan(self):
            self.LetakMakananBonus = (random.randint(1, grid_width-2)*gridsize, random.randint(1, grid_height-2)*gridsize)

    def gambarObjek(self, surface, pilihwarna) :
        gambarmakananbonus = pygame.Rect(((self.LetakMakananBonus[0]-5), (self.LetakMakananBonus[1]-5)), (30, 30))
        pygame.draw.rect(surface, pilihwarna, gambarmakananbonus)
        pygame.draw.rect(surface, (93, 216, 228), gambarmakananbonus, 1)
```

- **Class kotak**

Hanya terdapat methods GambarKotak dengan parameter nya berasal dari variabel surface pada main program.

1. **Methods GambarKotak**

LINK Repository : https://github.com/ouuzannn/BlackAs---Game-Ular.git

Pada nested loop yang pertama berguna untuk mengoutputkan grid ke program. Pengoutputan grid dibantu dengan function dari pygame untuk menggambarkan gridnya dengan pygame.draw.rect. Pada isi dari nested loop (loop dengan variabel y) maka jika elemen x+y bernilai genap maka grid berwarna biru muda, jika ganjil maka grid berwarna biru hijau ketua-tuaan.

Sedangkan pada nested loop kedua berfungsi untuk penggambaran border berwarna hitam pada program. Yang membedakan proses dengan nested loop yang pertama adalah yang diproses hanya lah setiap sisi-sisi pada window dari game saja dimana sisi-sisi tersebut diberi warna hitam.

```
class kotak :
    def GambarKotak(self,surface) :
        for y in range(0, int(grid_height)) :
            for x in range(0, int(grid_width)) :
                if (x+y) %2 == 0:
                    gridgenap = pygame.Rect((x*gridsize, y*gridsize), (gridsize,gridsize))
                    pygame.draw.rect(surface,(93,216,228), gridgenap)
                else :
                    gridganjil = pygame.Rect((x*gridsize, y*gridsize), (gridsize, gridsize))
                    pygame.draw.rect(surface, (84,194,205), gridganjil)
        #gambarborder
        for y in range(0, int(grid_height)) :
            for x in range(0, int(grid_width)) :
                if (x<1 or y<1 or x>22 or y>22):
                    xxx = pygame.Rect((x*gridsize, y*gridsize), (gridsize,gridsize))
                    pygame.draw.rect(surface,(0,0,0), xxx)
```

- **Main Program**

  1. **Global Variable**

```
#GLOBAL VARIABEL
clock = pygame.time.Clock()
lebar_layar = 480
tinggi_layar = 480
latar = (0, 0, 0)
win = pygame.display.set_mode((lebar_layar, tinggi_layar))
pygame.init()
bonus=bool(0)
waktubatas = 5

gridsize = 20
grid_width = lebar_layar/gridsize
grid_height = tinggi_layar/gridsize

atas = (0, -1)
bawah = (0, 1)
kiri = (-1, 0)
kanan = (1,0)
hitung = 0
```

Pada global variable ini dimana lebar dan tinggi layar yaitu 480 dengan grid pada layar berukuran 20 serta pada bagian atas, bawah, kiri, kanan ini diinisialisasi untuk pergerakan ular dengan parameter tersebut dapat berfungsi dengan baik. Ke arah atas akan selalu berkurang -1 karena pada perulangan di methods GambarKotak semakin ke bawah nilainya semakin bertambah dan oleh sabab itu, semakin ke atas maka semakin berkurang nilai nya, begitu pula saat isi dari nested loop(secara

horizontal) karena semakin ke kanan maka nilai semakin bertambah maka dari itu semakin ke kiri maka semakin kecil nilainya.

2. Function mainin :

Pada function inilah game ular berjalan dimana semua objek diinisiasi oleh setiap class dalam main/py maupun luar file ini seperti menuGame. Setiap variabel diinisiasikan seperti clock yang diinisiasi oleh library dari pygame yaitu time.clock untuk membuat ular bergerak dengan kecepatan nya berdasarkan frame per seconds (FPS), screen berguna untuk menyiapkan layar pada game yaitu ukuran dasarnya dan surface pun juga dengan cara menggunakan function dari pygame untuk menampilkan layar dari variabel screen. Lalu ada inisiasi objek oleh setiap class seperti pada gambar dan deklarasi warna putih dan hijau serta variabel hitung untuk membuat bentuk makanan bonus menjadi interaktif dan myfont berfungsi untuk inisiasi bentuk font untuk tampilan score saat game berjalan.

```
def mainin(indeks) :

    clock = pygame.time.Clock()
    screen = pygame.display.set_mode((lebar_layar, tinggi_layar), 0, 32)

    surface = pygame.Surface(screen.get_size())
    surface = surface.convert()

    #deklarasi objek
    papan = kotak()
    ular = Ular()
    makanan = Makanan()
    makananbonus = MakananBonus()

    putih=(255, 255, 255)
    hijau=(25, 163, 49)
    warnaa=[putih,hijau]
    hitungwaktu=0
    hitung=0
    #Setting ukuran dan jenis font dari tulisan score
    myfont = pygame.font.SysFont("monospace",18)
```

Lalu masuk ke dalam perulangan while dimana parameter nya selalu TRUE (tandanya game berjalan) dimana setiap game berjalan akan memiliki kecepatan sesuai dengan variabel kecepatan di kelas ular, kecepatan game dijalankan menggunakan clock.tick. Kemudian terdapat juga fungsi untuk mengatur kontrol ular, menampilkan fungsi gambar kotak papan, serta fungsi pergerakan ular yang digunakan untuk mengatur jika ular menabrak dirinya sendiri atau jika ular menabrak dinding pembatas. Lalu dideklarasikan juga variabel hitung yang beriterasi bertambah sebanyak 1 untuk membuat warna makanan bonus berkedip.

Saat game dijalankan terdapat percabangan if yang digunakan jika ular memakan makanan yang berarti letak posisi kepala ular sama dengan letak makanan maka perintah yang dijalankan adalah panjang ular bertambah 1, nilai ular bertambah 1, hitung makanan bertambah 1, mengeluarkan makanan

kembali di posisi yang random dengan memanggil fungsi randomize position, dan men-set timer dari makanan bonus agar menjadi 0, serta terdapat lagi percabangan di dalam if ini yaitu untuk mengecek jika ular telah memakan makanan sebanyak 10 makanan, maka kecepatan ular bertambah, dan kecepatan makanan bonus bertambah juga untuk menyesuaikan waktu.

Dibuat lagi percabangan untuk membuat variabel tanda dari makanan bonus bernilai 1 jika variabel hitungmakanan di modulo 5 adalah 0, yaitu untuk memunculkan makanan bonus setiap nilai bertambah 5.

```
while True :
    if indeks == 1:
        pygame.quit()
        sys.exit()

    #print (indeks)
    clock.tick(ular.kecepatan) #kecepatan ular
    ular.KontrolUlar()
    #kotak.GambarKotak(surface)
    papan.GambarKotak(surface)
    ular.PergerakanUlar()
    hitung += 1

    if ular.get_head_position() == makanan.letakUlar :
        ular.PanjangUlar += 1
        ular.nilai += 1
        ular.hitungmakanan += 1
        makananbonus.timer=0
        print("k", ular.kecepatan)
        makanan.randomize_position()
        if ular.nilai%10==0:
            ular.kecepatan = ular.kecepatan * 1.5
            makananbonus.waktubonus = makananbonus.waktubonus * 1.5

    if(ular.hitungmakanan !=0 and ular.hitungmakanan % 5 == 0):
        makananbonus.tanda=1
```

Masuk ke percabangan untuk menampilkan makanan bonus dengan syarat variabel tanda bernilai 1 dan nilai timer kurang dari nilai waktu bonus. Waktu bonus defaultnya adalah 25, dapat bertambah jika kecepatan ular juga bertambah. Fungsi dari variabel timer dan bonus ini adalah untuk menampilkan makanan bonus hanya selama 5 detik saja. Kemudian jika syarat telah memenuhi, maka dibuat percabangan untuk warna dari makanan bonus berkedip setiap detik. Makanan bonus ditampilkan dengan fungsi gambar objek. Dan setiap perulangan, nilai dari variabel timer bertambah satu. Di percabangan makanan bonus terdapat percabangan kembali untuk mengecek apakah makanan bonus termakan oleh ular. Jika termakan, maka panjang ular bertambah 1 dan score atau nilai bertambah 5, mereset variabel hitung makanan, tanda, dan timer menjadi 0, serta terdapat lagi percabangan di dalam percabangan ini yaitu untuk mengecek jika nilai ular berkelipatan 10, maka kecepatan ular bertambah, dan kecepatan makanan bonus bertambah juga untuk menyesuaikan waktu. Dari if yang

LINK Repository : https://github.com/ouuzannn/BlackAs---Game-Ular.git

pertama terdapat else if untuk mereset variabel tanda dari makanan bonus menjadi 0 setiap timer melebihi waktu bonus.

```
if ( makananbonus.tanda==1 and makananbonus.timer<makananbonus.waktubonus) :
    if hitung%2==0:
        pilihwarna=warnaa[0]
    else:
        pilihwarna=warnaa[1]
    makananbonus.gambarObjek(surface, pilihwarna)
    makananbonus.timer+=1
    if ular.get_head_position() == makananbonus.letakUlar :
        ular.PanjangUlar += 1
        ular.nilai += 5
        ular.hitungmakanan = 0
        makananbonus.tanda=0
        makananbonus.timer=0
        if ular.nilai%10==0:
            ular.kecepatan = ular.kecepatan * 1.5
            makananbonus.waktubonus = makananbonus.waktubonus * 1.5
        makananbonus.randomize_position()
    print("t", makananbonus.timer)
    print("w", makananbonus.waktubonus)
elif makananbonus.timer>=makananbonus.waktubonus:
    makananbonus.tanda=0
```

Pada perulangan while juga di panggil fungsi draw dari ular untuk menampilkan ular serta gambar objek dari makanan. Screen blit digunakan untuk menampilkan gampar pada window game, screen blit dari variabel text untuk menampilkan score atau nilai dari game di pojok kanan atas. Dan pygame.display.update digunakan untuk meng-update semua pengaturan display.

```
ular.draw(surface)
makanan.gambarObjek(surface)

screen.blit(surface, (0,0)) #menampilkan gambar pada window game
text = myfont.render("Score : {0}".format(ular.nilai), 1, (putih))
screen.blit(text, (20,0))
pygame.display.update()
```

3. Function menu

Pada function ini dengan menggunakan parameter variabel cek(boolean), objek menu diinisiasi oleh class Pymenu (yang merupakan class dari file MenuGame.py) serta diinisiasikan pula objek TombolNewGame untuk tombol GUI “New Game” dan TombolQuit untuk GUI “Quit”. Lalu objek menu menjalankan methods pada Pymenu yaitu tambahkanTombol dengan menggunakan parameter TombolNewGame dan function mainin untuk mengoutputkan tombol GUI “New Game” pada game dan TombolQuit serta function mainin untuk mengoutputkan tombol GUI “Quit” pada game. Serta methods tampilan pada objek menu dijalankan untuk mengoutputkan tampilan frame pada menu yang sudah di setting pada variabel window dan methods ini berfungsi untuk me-render nya.

LINK Repository : https://github.com/ouuzannn/BlackAs---Game-Ular.git

```
cek=bool(1)

def Menu(cek):


    window = pygame.display.set_mode((lebar_layar, tinggi_layar), 0, 32)

    if cek == 1 :
        menu = MenuGame.PyMenu(BLACK, lebar_layar/2, 45, "GAME ULAR BLACK AS")
    else :
        menu = MenuGame.PyMenu(RED, lebar_layar/2, 45, "PERMAINAN BERAKHIR")

  # create the buttons
    TombolNewGame = TombolMenu.PyButton(lebar_layar/2, 130, "New Game")
    TombolQuit = TombolMenu.PyButton(tinggi_layar/2, 220, "Quit")

    menu.tambahkanTombol(TombolNewGame, mainin)
    menu.tambahkanTombol(TombolQuit, mainin)

  # draw the menu
    menu.tampilan(window)
```

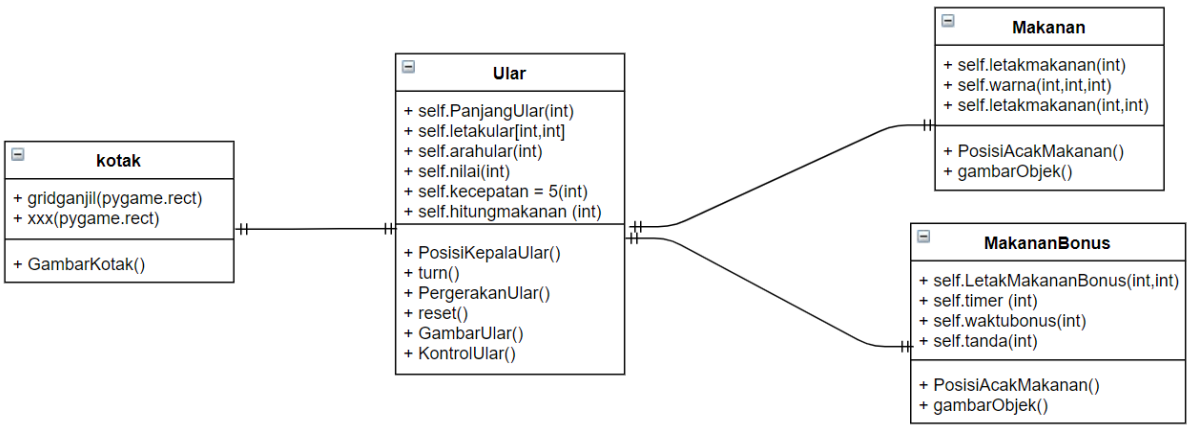
- **Eksekusi Program Pada Main Program**

Pada portal main program, function yang dieksekusi adalah Menu() sehingga terjadi efek berantai pada code yang satu padu menjalankan function dan methods nya sehingga program tersebut menjadi sebuah game ular. Berikut gambarnya pada eksekusi di main program :

```
if __name__ == '__main__':
    Menu(cek)

Menu(cek)
```

F. Diagram Class



LINK Repository : https://github.com/ouuzannn/BlackAs---Game-Ular.git

G. Kesimpulan

Berdasarkan game yang kami buat, dapat disimpulkan bahwa :

1. Game ular ini dibuat menggunakan bahasa python dengan konsep *object oriented programming* (OOP).
2. Pembuatan game memerlukan modul pygame yang berisi beberapa method dan fungsi yang digunakan dalam pembuatannya.
3. Pada game ini terdapat 3 class yaitu Ular,Makanan,MakananBonus, dan kotak. Class Ular untuk merender objek ular ke program game, class Makanan untuk merender makanan pada ular ke game, class MakananBonus untuk merender makanan bonus pada ular ke program game, dan class kotak untuk merender papan untuk ular merayap.

H. Daftar Pustaka

https://github.com/MysteryCoder456/Python.io

https://github.com/pbschmid/pymenu

https://github.com/InfernoKraftz/SnakeGamePython

Channel Youtube : “Kite” pada video “”How To Build Snake In Python”

LINK Repository : https://github.com/ouuzannn/BlackAs---Game-Ular.git