**PROPOSAL LAPORAN PROJECT AKHIR GRAFIKA KOMPUTER**



**Disusun oleh :**

**Kelompok I – Racing Car**

**Anggota Kelompok :**

**Joel Binsar Jupiter (202410103042)   
Bayu Dwi Satrio Utomo (202410103070)**

**Kelas: Grafika Komputer A Prodi: Informatika**

**FAKULTAS ILMU KOMPUTER UNIVERSITAS JEMBER**

**2021/2022**

**DAFTAR ISI**

[**BAB I PENDAHULUAN** 3](#_Toc91020912)

[**1.1 Latar Belakang** 3](#_Toc91020913)

[**1.2 Rumusan Masalah** 3](#_Toc91020914)

[**1.3 Tujuan** 3](#_Toc91020915)

[**1.4 Manfaat** 3](#_Toc91020916)

[**1.5 Batasan Masalah** 3](#_Toc91020917)

[**BAB II SKEMA PERMAINAN** 4](#_Toc91020918)

[**2.1 Game Actors** 4](#_Toc91020919)

[**2.2 FSM (Finite State Machine)** 4](#_Toc91020920)

[**2.3 Scenes** 4](#_Toc91020921)

[**2.4 Game Level** 4](#_Toc91020922)

[**BAB III PEMBAHASAN** 5](#_Toc91020923)

[**3.1 Import Library** 5](#_Toc91020924)

[**3.2 Latar Game** 5](#_Toc91020925)

[**3.3 Kendaraan** 5](#_Toc91020926)

[**3.4 Iterate, ShowScreen, dan main** 6](#_Toc91020927)

[**LAMPIRAN** 8](#_Toc91020928)

[**1.** **Tampilan Game** 8](#_Toc91020929)

[**2.** **Full Code** 9](#_Toc91020930)

# **BAB I PENDAHULUAN**

## **1.1 Latar Belakang**

Di era serba modern saat ini semua game yang kita kenal sudah menjadi lebih up dengan nuansa yang semakin mengikuti era modern nya. Lalu banyak nya game bermunculan dengan genre yang baru dan juga dengan metode yang di dalam in game nya yang serba baru beda dari game dulu yang kita kenal semasa kecil.

Oleh karena itu lah kami ingin membantu mengembalikan nuansa game lama yang telah mengisi waktu masa kecil kita agar bisa dirasakan oleh para player/pengguna lama dan juga oleh anak-anak kecil sekarang. Agar mereka bisa merasakan betapa serunya game pada era dulu yang tidak kalah asik dan menarik dengan game sekarang. Jadi kami disini lebih mengenalkan kembali game yang telah lama hilang, kami akan mengemas game tersebut dengan sebaik mungkin agar bisa menarik perhatian para player lama dan juga player baru di era sekarang.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah diuraikan di atas, terdapat beberapa pokok permasalahan yang akan dibahas, yaitu:

1. Bagaimana mekanisme yang dibuat di dalam *Racing Car*?
2. Bagaimana alur permainan pada *Racing Car?*
3. Ada berapa tingkat kesulitan yang ada dalam game tersebut ?

## **1.3 Tujuan**

Tujuan dari pembuatan game ini adalah:

1. Untuk mengajak para pemain agar bisa merasakan nuansa game lama,
2. Untuk membantu para player lama agar bisa bernostalgia dengan game ini,
3. Untuk mengisi waktu luang di masa pandemi.

## **1.4 Manfaat**

1. Dapat membuat mereka lebih nyaman dalam bermain game,
2. Dapat merasakan nostalgia akan game lama,
3. Mengisi waktu luang agar bisa bersantai dengan game.

## **1.5 Batasan Masalah**

Batasan masalah diperlukan untuk mencapai tujuan dari proposal ini. Berikut batasan masalah pada game *“ Racing Car“* ini :

1. Permainan ini hanya terdapat satu sirkuit saja,
2. Hanya bisa memakai mobil dari in game,
3. Tidak memiliki tingkatan kesulitan, tetapi nantinya setiap menit akan bertambah tingkat kesulitannya sendiri sampai akhir game.

# **BAB II SKEMA PERMAINAN**

## **2.1 Game Actors**

Pada game “Racing Car” ini kami memberikan kebebasan untuk pengguna khususnya kepada pengguna seperti kepada remaja ataupun anak-anak SD. Untuk pemeran utama yang ada pada game itu sendiri yaitu seorang *user* yang akan meng klik tampilan play game dan akan langsung masuk ke dalam in game nya.

## **2.2 FSM (Finite State Machine)**

Pada tampilan awal game pengguna akan langsung di beri tampilan Play Game. Ketika telah menakan tombol itu pengguna langsung di alihkan langsung ke dalam lintasan yang otomatis sudah disediakan:

Di dalam permainan ini nantinya si pengguna akan melakukan permainan dengan cara mendahului dari lawan dengan diberi rintangan dalam setiap sirkuitnya, jadi di dalam setiap sirkuit akan terdapat rintangan secara bersamaan maupun acak agar bisa memberi tantangan dalam in game nya.

## **2.3 Scenes**

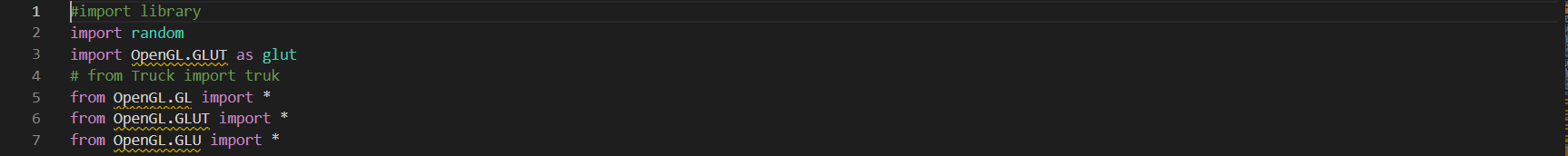
Pada tampilan awal pengguna akan diberikan tampilan seperti dalam game umunya, Yaitu terdapat mobil yang berada dalam lintasan. Kemudian ketika pengguna mengklik Play Game akan langsung masuk ke dalam in game nya.

## **2.4 Game Level**

Pada pilihan level yang ada pada permainan ini yaitu terdapat dalam menit nya. Jadi ketika kita bisa bertahan dalam hitungan menit yang lama, maka sistem dari game akan langsung menaikkan kecepatan mobil dan juga memperbanyak rintangan untuk bisa mencelakai mobil agar lebih susah untuk mencapai akhir game.

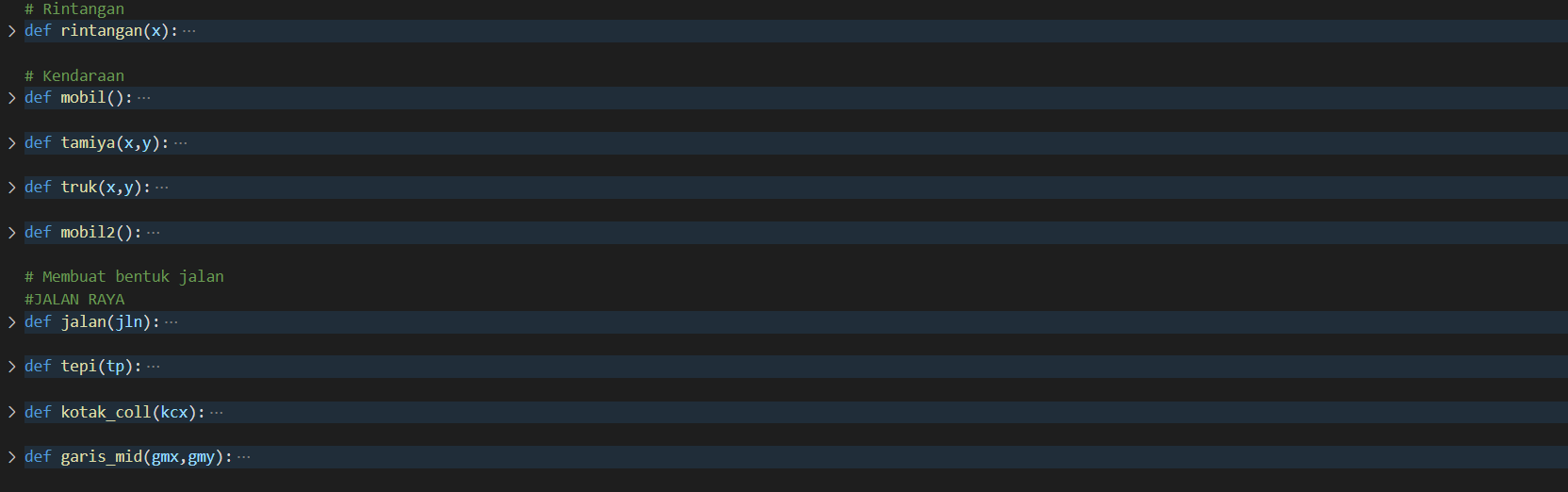
# **BAB III PEMBAHASAN**

## **3.1 Import Library**



Mengimport library yang digunakan untuk membuat game

## **3.2 Latar Game**

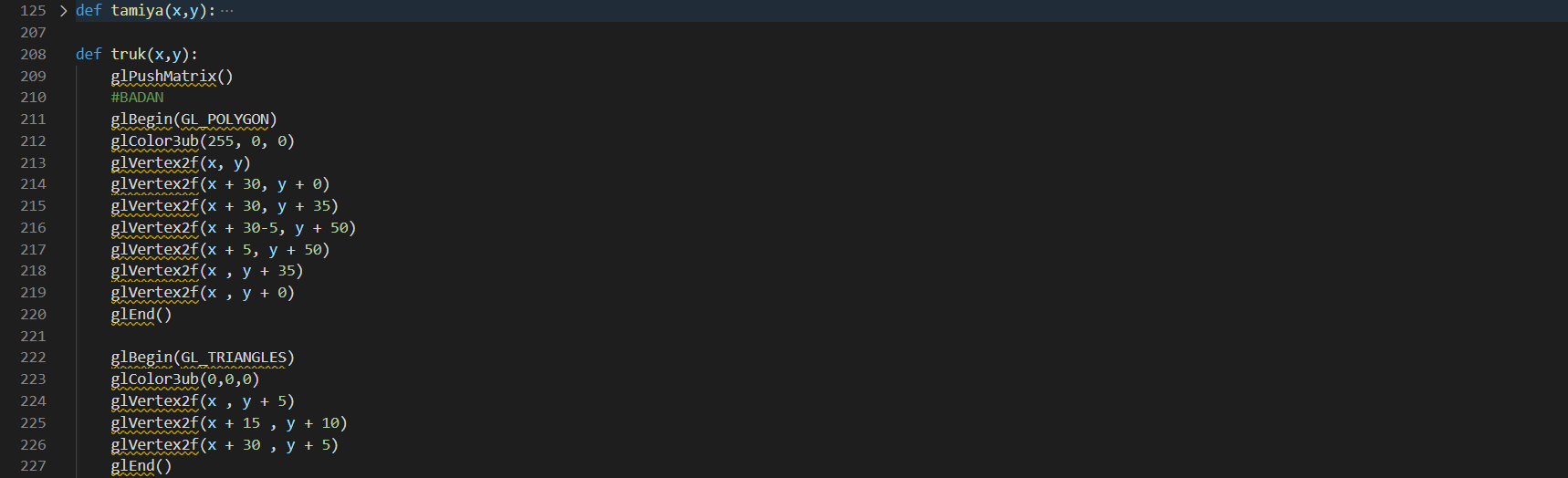


Untuk latar game yang kami gunakan diantaranya berupa Rintangan, kendaran, dan juga jalan raya. Pada latar game ini kami menggunakan fungsi :

* Rintangan : menggunakan Fungsi glPushMatrix dan glPopMatrix
* Jalan Raya : menggunakan Fungsi GL\_QUADS

Pada jalan raya juga diberikan fungsi Seperti glPushMatrix dan glPopMatrix dan juga glTranslated() digunakan untuk memindahkan objek, lalu ada juga glVertex2f() digunakan untuk membuat titik x,y

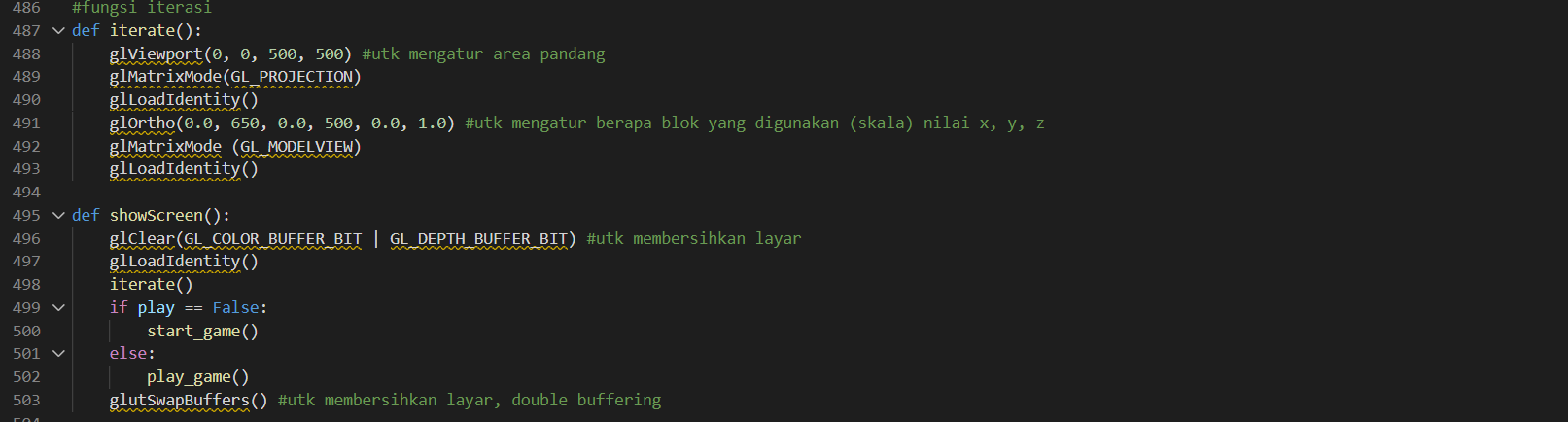
## **3.3 Kendaraan**

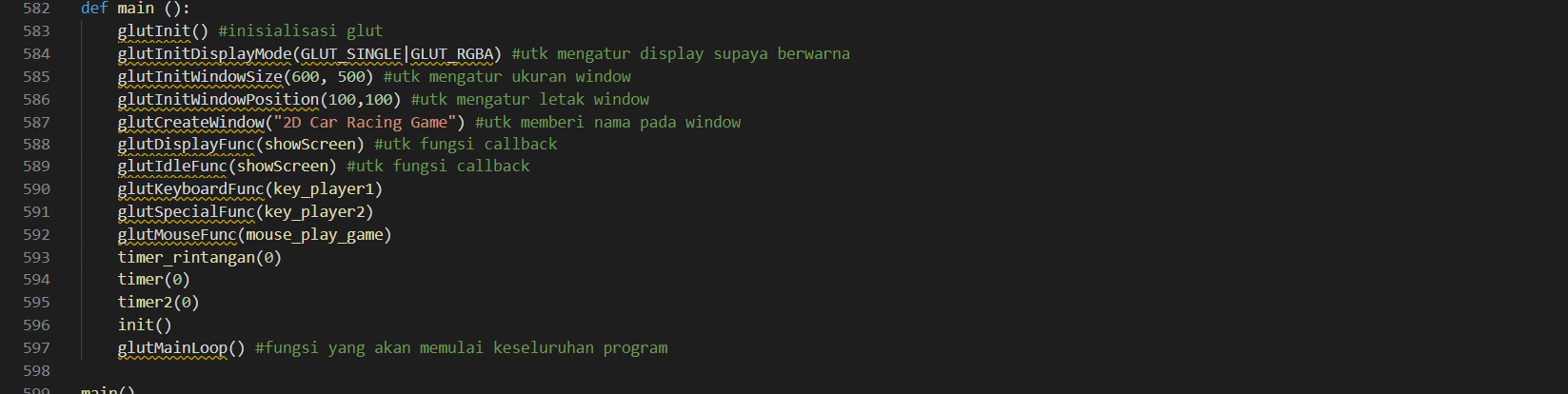


Pada deff kendaraan disini terdapat 2 yaitu Truck dan juga Tamiya yang masing-masing nya diberikan fungsi sebagai berikut :

* glTranslated() digunakan untuk memindahkan objek
* glPushMatix() digunakan untuk mendorong matrix yang sedang aktif, ke dalam tumpukan matrix(matrix stack
* glPopMatix() digunakan untuk mengeluarkan matrix dari matrix stack dan sistem koordinat akan kembali keadaan semula.
* glvertex2f() digunakan untuk membuat titik x,y
* glBegin(GL\_QUADS) digunakan untuk membuat objek primitif segi empat
* glBegin(GL\_POLYGON) digunakan untuk membuat objek primitif lebih dari empat titik.
* glBegin(GL\_TRIANGLES) digunakan untuk membuat objek primitif dari tiga titik.

## **3.4 Iterate, ShowScreen, dan main**





Def Iterate :

Pada deff iterate terdapat beberapa fungsi yang diantaranya :

* glViewport digunakan untuk menentukan titik awal dan titik akhir (pixel tampilan)
* glMatrixMode digunakan untuk memberi nilai masukan pada matrix dengan model projection
* glLoadIdentity digunakan untuk memastikan bahwa setiap kali kita memasuki mode proyeksi, matriks akan diatur ulang ke matrix identitas
* glOrtho digunakan untuk mengatur berapa blok yang digunakan (skala) nilai x, y, z
* glMatrixMode digunakan untuk mengubah objek ke ruang koordinat tampilan (atau ruang kamera)
* glLoadIdentity digunakan untuk memastikan bahwa setiap kali kita memasuki mode proyeksi, matriks akan diatur ulang ke matriks identitas

Deff showScreen

Pada deff showScreent digunakan untuk menampilkan tampilan game, dimana tampilan game yang pertama itu merupakan tulisan *“PLAY GAME”* yang nantinya ketika oleh user atau pengguna klik akan langsung masuk ke dalam in game.

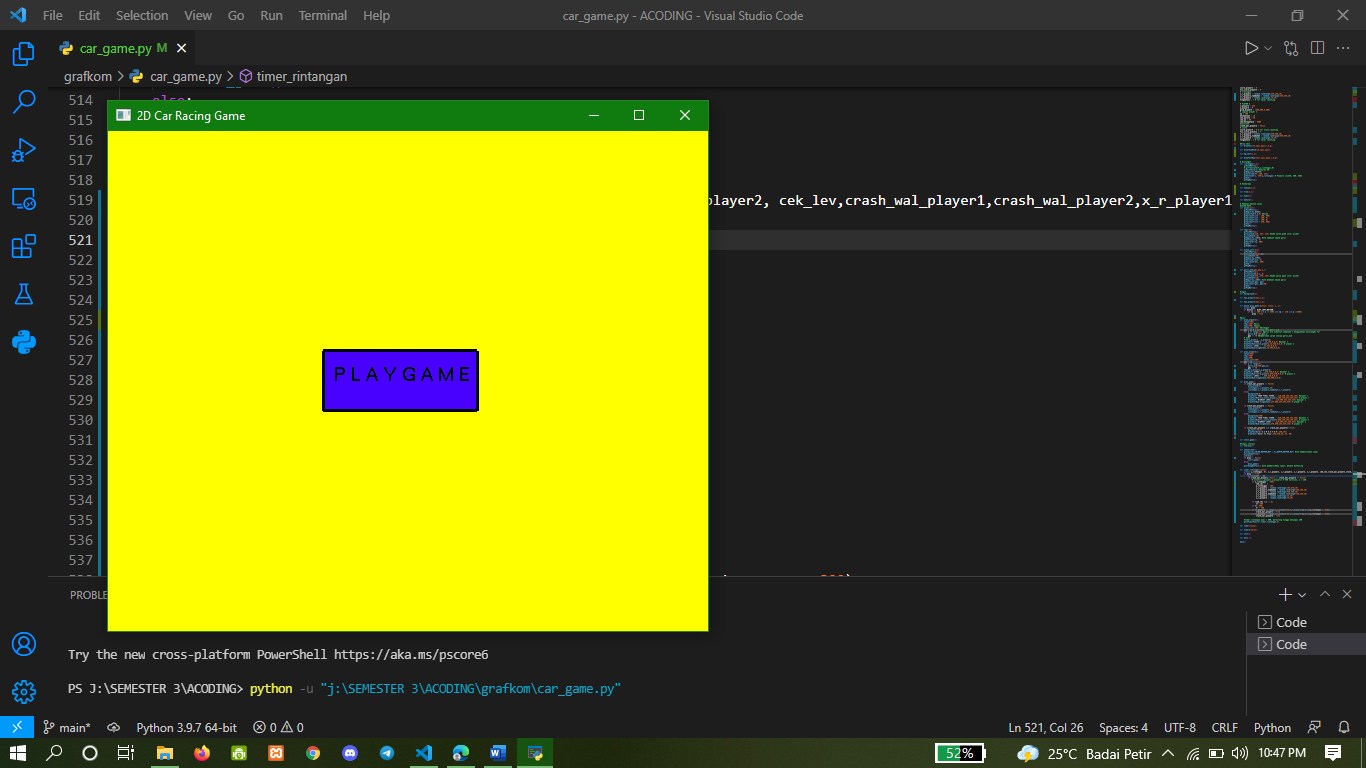
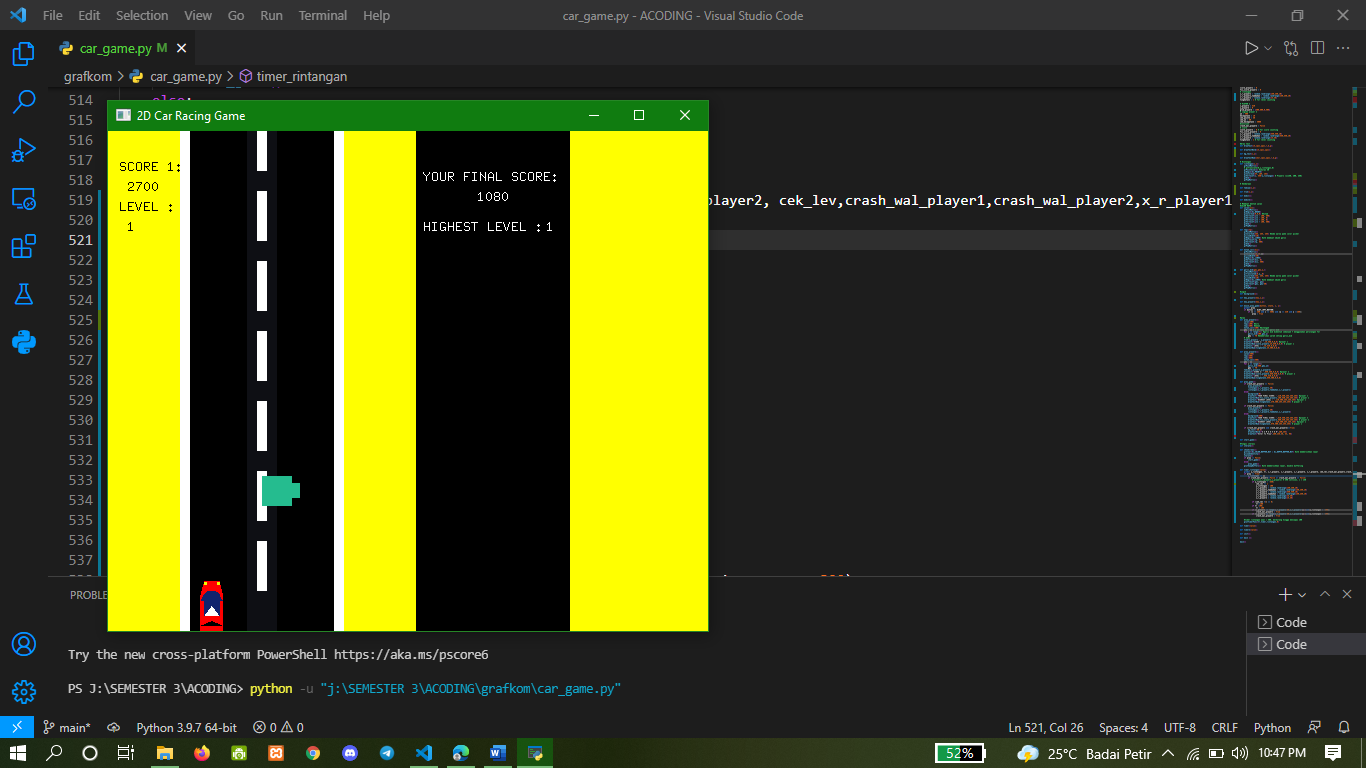
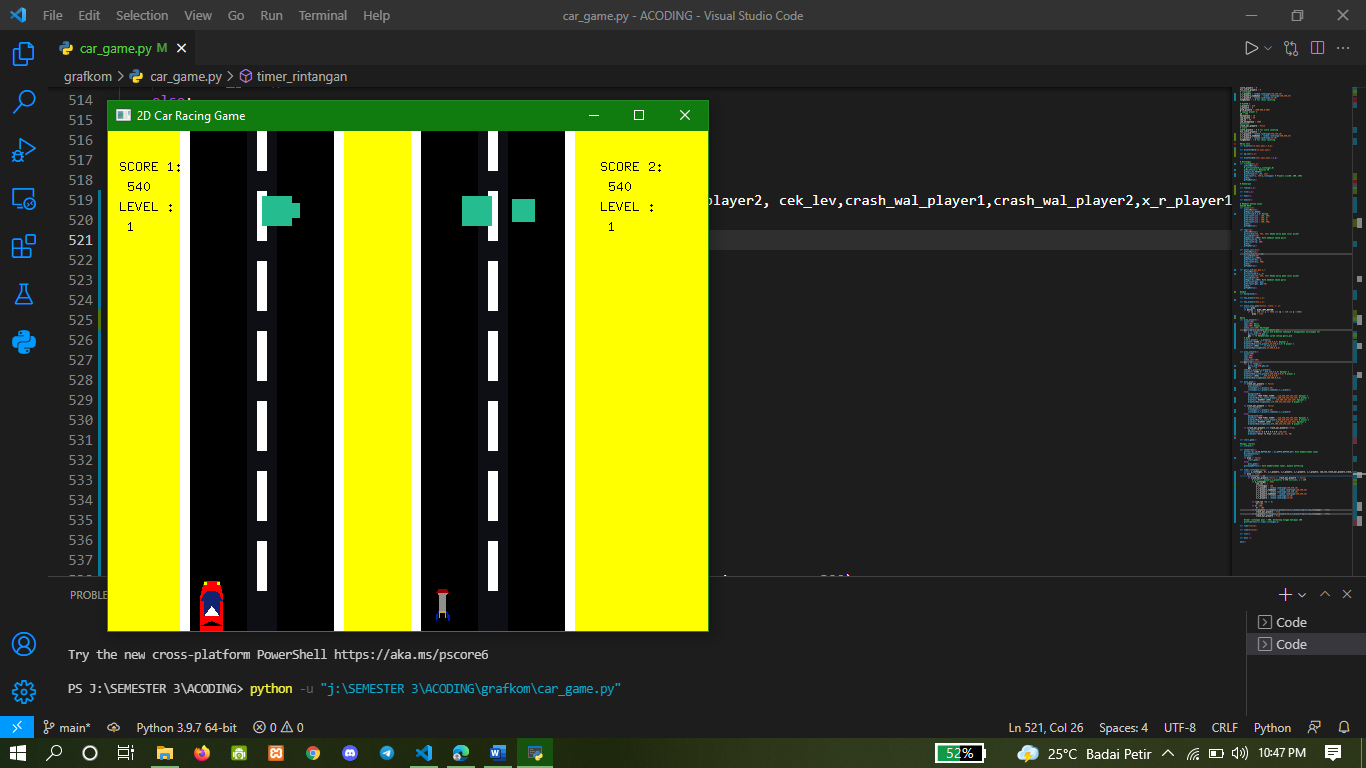
Deff main

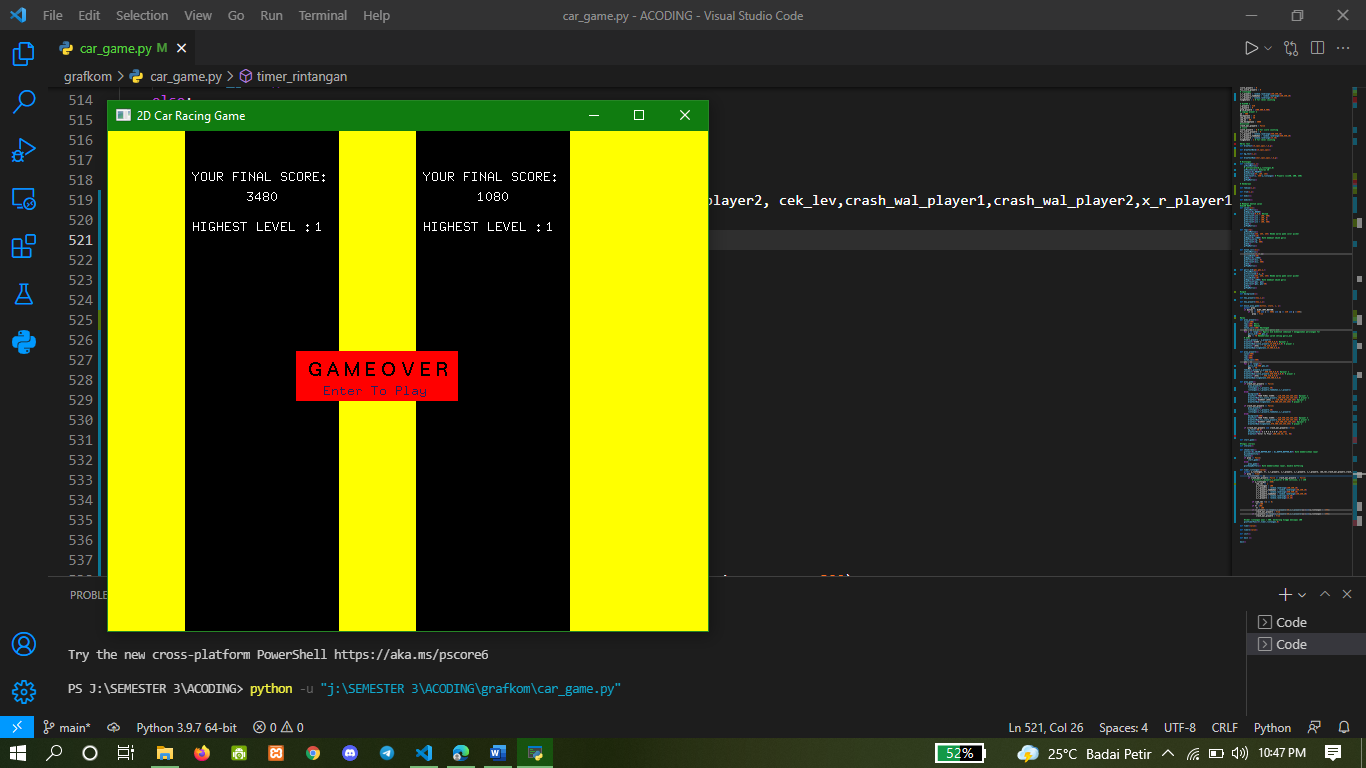
Pada deff main terdapat beberapa fungsi yang diantaranya :

* glutInit() digunakan untuk inisialisasi glut
* glutInitDisplayMode() digunakan untuk mengatur layer menjadi berwarna
* glutInitWindowSize() digunakan untuk mengatur ukuran layar atau window
* glutInitWindowPosition() digunakan untuk mengatur posisi window
* glutCreateWindow(“2D CAR RACING GAME”) digunakan untuk memberi nama pada windows
* glutDisplayFunc() digunakan untuk menampilkan objek pada layar, fungsi callback
* glutKeyboardFunc(Key\_Player1)digunakan untuk interaksi keyboard player 1
* glutSpesialFunc(Key\_Player2) digunakan untuk interaksi keyboard player 2
* glutMouseFunc(input\_mouse) digunakan untuk iteraksi mouse dalam game
* glutTimerFunc() digunakan untuk membuat sebuah objek yang dikontrol oleh waktu
* timer() digunakan untuk memanggil def timer
* iterate digunakan untuk memanggil def iterate
* glutMainLoop() digunakan untuk me-looping atau mengulang fungsi main

# **LAMPIRAN**

## **Tampilan Game**

1. Tampilan Awal 2. Tampilan didalam permainan

3. Tampilan ketika salah satu tabrakan 4. Tampilan Ketika permainan selesai

## **Full Code**

*#import library*

import random

import OpenGL.GLUT as glut

*# from Truck import truk*

from OpenGL.GL import \*

from OpenGL.GLUT import \*

from OpenGL.GLU import \*

play = False

y\_rintangan = 50 *# digunakan untuk pergerakan rintangan ke bawah*

tr = 500

cek\_lev = 1

*# PLAYER 1*

x\_player1 = 120

y\_player1 = 0

grid\_player1 = [100,300,0,500]

*# Logika player 1*

y = 500

kecepatan = 10

cek\_point = 30

cek\_y = 50

cek\_kecepatan = 5000

*#collison*

crash\_wal\_player1 = False *# True ketika menabrak*

*# score*

score\_player1 = 0

fix\_score\_player1 = 0

*# rintangan*

x\_r\_player1 = random.randrange(150,250,10)

x\_r\_player1\_tambahan = random.randrange(150,250,10)

s\_r\_player1 = random.randrange(15,30)

tingkatan1 = 1 *# for level counting*

*# PLAYER 2*

x\_player2 = 420

y\_player2 = 0

grid\_player2 = [400,600,0,500]

*# Logika player 2*

y2 = 500

kecepatan2 = 10

cek\_point2 = 30

cek\_y2 = 50

cek\_kecepatan2 = 5000

*#collison*

crash\_wal\_player2 = False

*# score*

score\_player2 = 0 *# for score counting*

fix\_score\_player2 = 0

x\_r\_player2 = random.randrange(450,550,10)

x\_r\_player2\_tambahan = random.randrange(450,550,10)

s\_r\_player2 = random.randrange(15,30)

tingkatan2 = 1 *# for level counting*

*#draw text*

def drawText(ch,xpos,ypos,r,b,g):

    color = (r, b, g)

    font\_style = glut.GLUT\_BITMAP\_8\_BY\_13

    glColor3ub(color[0],color[1],color[2])

    line=0

    glRasterPos2f (xpos, ypos)

    for i in ch:

       if  i=='\n':

          line=line+1

          glRasterPos2f (xpos, ypos\*line)

       else:

          glutBitmapCharacter(font\_style, ord(i))

def drawTextBold(ch,xpos,ypos):

    glPushMatrix()

    color = (0,0,0)

    font\_style = glut.GLUT\_BITMAP\_HELVETICA\_18

    glColor3ub(color[0],color[1],color[2])

    line=0

    glRasterPos2f (xpos, ypos)

    for i in ch:

       if  i=='\n':

          line=line+1

          glRasterPos2f (xpos, ypos\*line)

       else:

          glutBitmapCharacter(font\_style, ord(i))

    glPopMatrix()

def bg\_text(x,y):

    glColor3ub(255, 0, 0)

    glBegin(GL\_QUADS)

    glVertex2f(285+x,230+y)

    glVertex2f(495+x,230+y)

    glVertex2f(495+x,280+y)

    glVertex2f(285+x,280+y)

    glEnd()

def drawTextNum(skor,xpos,ypos,r,b,g):

    color = (r, b, g)

    font\_style = glut.GLUT\_BITMAP\_8\_BY\_13

    glColor3ub(color[0],color[1],color[2])

    line=0

    glRasterPos2f (xpos, ypos)

    for i in str(skor):

       if  i=='\n':

          line=line+1

          glRasterPos2f (xpos, ypos\*line)

       else:

          glutBitmapCharacter(font\_style, ord(i))

*# Rintangan*

def rintangan(x,s):

    glPushMatrix()

*# glTranslate(0,y\_rintangan,0)*

    glPointSize(s) *#ukuran 30*

    glBegin(GL\_POINTS)

    glColor3ub(37, 188, 143)

    glVertex2f(x, 450+y\_rintangan) *# Plyaer1 =x[150, 200, 250]*

    glEnd()

    glPopMatrix()

*# Kendaraan*

def tamiya(x,y):

    glColor3ub(150, 0, 0)

    glBegin(GL\_POLYGON)

    glVertex2f(8+x, 38+y)

    glVertex2f(6+x, 38+y)

    glVertex2f(8+x, 40+y)

    glVertex2f(10+x, 42+y)

    glVertex2f(20+x, 42+y)

    glVertex2f(22+x, 40+y)

    glVertex2f(24+x, 38+y)

    glVertex2f(22+x, 38+y)

    glEnd()

    glColor3ub(145, 142, 142)

    glBegin(GL\_POLYGON)

    glVertex2f(8+x, 38+y)

    glVertex2f(22+x, 38+y)

    glVertex2f(20+x, 36+y)

    glVertex2f(10+x, 36+y)

    glEnd()

    glColor3ub(145, 142, 142)

    glBegin(GL\_POLYGON)

    glVertex2f(10+x, 36+y)

    glVertex2f(20+x, 36+y)

    glVertex2f(20+x, 30+y)

    glVertex2f(10+x, 30+y)

    glEnd()

    glColor3ub(145, 142, 142)

    glBegin(GL\_POLYGON)

    glVertex2f(20+x, 30+y)

    glVertex2f(10+x, 30+y)

    glVertex2f(10+x, 20+y)

    glVertex2f(20+x, 20+y)

    glEnd()

    glColor3ub(0, 30, 180)

    glBegin(GL\_POLYGON)

    glVertex2f(20+x, 20+y)

    glVertex2f(24+x, 16+y)

    glVertex2f(24+x, 10+y)

    glVertex2f(22+x, 10+y)

    glVertex2f(22+x, 16+y)

    glVertex2f(20+x, 18+y)

    glEnd()

    glColor3ub(0, 30, 180)

    glBegin(GL\_POLYGON)

    glVertex2f(10+x, 20+y)

    glVertex2f(6+x, 16+y)

    glVertex2f(6+x, 10+y)

    glVertex2f(8+x, 10+y)

    glVertex2f(8+x, 16+y)

    glVertex2f(10+x, 18+y)

    glEnd()

    glColor3ub(218,165,32)

    glBegin(GL\_POLYGON)

    glVertex2f(10+x, 20+y)

    glVertex2f(10+x, 18+y)

    glVertex2f(10+x, 12+y)

    glVertex2f(12+x, 14+y)

    glVertex2f(12+x, 18+y)

    glEnd()

    glColor3ub(218,165,32)

    glBegin(GL\_POLYGON)

    glVertex2f(10+x, 20+y)

    glVertex2f(12+x, 18+y)

    glVertex2f(18+x, 18+y)

    glVertex2f(20+x, 20+y)

    glEnd()

    glColor3ub(218,165,32)

    glBegin(GL\_POLYGON)

    glVertex2f(20+x, 20+y)

    glVertex2f(20+x, 18+y)

    glVertex2f(20+x, 12+y)

    glVertex2f(18+x, 14+y)

    glVertex2f(18+x, 18+y)

    glEnd()

def truk(x,y):

    glPushMatrix()

*#BADAN*

    glBegin(GL\_POLYGON)

    glColor3ub(255, 0, 0)

    glVertex2f(x, y)

    glVertex2f(x + 30, y + 0)

    glVertex2f(x + 30, y + 35)

    glVertex2f(x + 30-5, y + 50)

    glVertex2f(x + 5, y + 50)

    glVertex2f(x , y + 35)

    glVertex2f(x , y + 0)

    glEnd()

    glBegin(GL\_TRIANGLES)

    glColor3ub(0,0,0)

    glVertex2f(x , y + 5)

    glVertex2f(x + 15 , y + 10)

    glVertex2f(x + 30 , y + 5)

    glEnd()

*#belakang kaca*

    glBegin(GL\_QUADS)

    glColor3ub(16, 35, 110)

    glVertex2f(x + 5 , y + 15)

    glVertex2f(x + 25 , y + 15)

    glVertex2f(x + 25 , y + 30)

    glVertex2f(x + 5 , y + 30)

    glEnd()

    glBegin(GL\_TRIANGLES)

    glColor3ub(255,255,255)

    glVertex2f(x + 5 , y + 15)

    glVertex2f(x + 15 , y + 25)

    glVertex2f(x + 25 , y + 15)

    glEnd()

*# KACA*

    glBegin(GL\_POLYGON)

    glColor3ub(16, 35, 110)

    glVertex2f(x , y + 30)

    glVertex2f(x + 30 , y + 30)

    glVertex2f(x + 25 , y + 35)

    glVertex2f(x + 25 , y + 37)

    glVertex2f(x + 20 , y + 40)

    glVertex2f(x + 10 , y + 40)

    glVertex2f(x + 5 , y + 37)

    glVertex2f(x + 5, y + 35)

    glEnd()

*#lampu*

    glPointSize(3)

    glBegin(GL\_POINTS)

    glColor3ub(238, 255, 0)

    glVertex2f(x + 7, y + 47)

    glEnd()

    glPointSize(3)

    glBegin(GL\_POINTS)

    glColor3ub(238, 255, 0)

    glVertex2f(x + 23 , y + 47)

    glEnd()

    glPopMatrix()

def mobil():

    glPushMatrix()

    glBegin(GL\_QUADS)

    glColor3ub(37, 188, 143)

    glVertex2f(x\_player1 , y\_player1      + 0)

    glVertex2f(x\_player1 + 30 , y\_player1 + 0)

    glVertex2f(x\_player1 + 30 , y\_player1 + 50)

    glVertex2f(x\_player1 , y\_player1      + 50)

    glEnd()

    glPopMatrix()

def mobil2():

    glPushMatrix()

    glBegin(GL\_QUADS)

    glColor3ub(37, 188, 143)

    glVertex2f(x\_player2 , y\_player2 + 0)

    glVertex2f(x\_player2 + 30 , y\_player2 + 0)

    glVertex2f(x\_player2 + 30 , y\_player2 + 50)

    glVertex2f(x\_player2 , y\_player2 + 50)

    glEnd()

    glPopMatrix()

*# Membuat bentuk jalan*

*#JALAN RAYA*

def jalan(jln):

    glPushMatrix()

    glBegin(GL\_QUADS)

    glColor3ub(0,0,0) *#hitam*

    glVertex2f(jln - 100, 500)

    glVertex2f(jln - 100, 0)

    glVertex2f(jln + 100, 0)

    glVertex2f(jln + 100, 500)

    glEnd()

    glPopMatrix()

def tepi(tp):

    glPushMatrix()

    glColor3ub(255, 255, 255) *#kode warna pake color picker*

    glLineWidth(10)

    glBegin(GL\_LINES) *#utk membuat objek garis*

    glVertex2f(tp, 0)

    glVertex2f(tp, 500)

    glEnd()

    glPopMatrix()

def kotak\_coll(kcx):

    glPushMatrix()

    glColor3ub(14,15,20)

    glLineWidth(30)

    glBegin(GL\_LINES)

    glVertex2f(kcx, 0)

    glVertex2f(kcx, 500)

    glEnd()

    glPopMatrix()

def garis\_mid(gmx,gmy,y\_):

    glPushMatrix()

    glTranslated(0,y\_,0)

    glColor3ub(255, 255, 255) *#kode warna pake color picker*

    glLineWidth(10)

    glBegin(GL\_LINES) *#utk membuat objek garis*

    glVertex2f(gmx, gmy)

    glVertex2f(gmx, gmy+50)

    glEnd()

    glPopMatrix()

*#Logik*

def background(x):

    glPushMatrix()

    glColor3ub(0,0,0)

    glBegin(GL\_QUADS)

    glVertex2f(100+x, 0)

    glVertex2f(300+x, 0)

    glVertex2f(300+x, 500)

    glVertex2f(100+x, 500)

    glEnd()

    glPopMatrix()

def key\_player1(key,x,y):

    global x\_player1, y\_player1, crash\_wal\_player1, play, crash\_wal\_player2

*# Untuk mengubah posisi kotak*

    if ord(key) == ord('w'):

        if y\_player1+50 > 500:

            y\_player1 += 0

        else:

            y\_player1 += 0

    elif ord(key) == ord('s'):

        if y\_player1-5 < 0:

            y\_player1 -= 0

        else:

            y\_player1 -= 0

    elif ord(key) == ord('d'):

        if crash\_wal\_player1 == False:

            if x\_player1+50 > grid\_player1[1]:

                x\_player1 +=0

                crash\_wal\_player1 = True

            else:

                x\_player1 += 10

        else:

            x\_player1 +=0

    elif ord(key) == ord('a'):

        if crash\_wal\_player1 == False:

            if x\_player1-20 < grid\_player1[0]:

                x\_player1 +=0

                crash\_wal\_player1 = True

            else:

                x\_player1 -= 10

        else:

            x\_player1 -= 0

    elif ord(key) == ord(b'\r'):

        play = False

        crash\_wal\_player1 = False

        crash\_wal\_player2 = False

def key\_player2(key,x,y):

    global x\_player2, y\_player2, crash\_wal\_player2

*# Untuk mengubah posisi karakter*

    if key == GLUT\_KEY\_UP:

        if y\_player2+50 > 500:

            y\_player2 += 0

        else:

            y\_player2 += 0

    elif key == GLUT\_KEY\_DOWN:

        if y\_player2-5 < 0:

            y\_player2 -= 0

        else:

            y\_player2 -= 0

    elif key == GLUT\_KEY\_RIGHT:

        if crash\_wal\_player2 == False:

            if x\_player2+50 > grid\_player2[1]:

                x\_player2 +=0

                crash\_wal\_player2 = True

            else:

                x\_player2 += 10

        else:

            x\_player2 += 0

    elif key == GLUT\_KEY\_LEFT:

        if crash\_wal\_player2 == False:

            if x\_player2-20 < grid\_player2[0]:

                x\_player2 +=0

                crash\_wal\_player2 = True

            else:

                x\_player2 -= 10

        else:

            x\_player2-=0

def mouse\_play\_game(button, state, x, y):

    global play

    if button == GLUT\_LEFT\_BUTTON:

        if (x >= 280 and x <= 480) and (y >= 220 and y <=280):

            play = True

*#play*

def play\_player1():

    jalan(200)

    tepi(100) *#kiri*

    tepi(300) *#kanan*

    kotak\_coll(200) *#ditengah*

    gmy = 20 *# y pertama dari garis mid*

    for i in range(7): *#garis mid dibentuk sebanyak 7 menggunakan perulangan for*

        garis\_mid(200,gmy,y)

        gmy += 70 *#memberikan jarak setiap garis\_mid*

*# mobi*

    truk(x\_player1 , y\_player1)

    drawText('SCORE 1: ',15,460,0,0,0) *#player 1*

    drawTextNum(score\_player1,25,440,0,0,0) *# player 1*

    drawText('LEVEL : ',15,420,0,0,0)

    drawTextNum(tingkatan1,25,400,0,0,0)

def play\_player2():

    jalan(500)

    tepi(400)

    tepi(600)

    kotak\_coll(500)

    gmy = 20

    for i in range(7):

        garis\_mid(500,gmy,y2)

        gmy += 70

    tamiya(x\_player2,y\_player2)

    drawText('SCORE 2: ',640,460,0,0,0) *#player 2*

    drawTextNum(score\_player2,650,440,0,0,0) *# player 2*

    drawText('LEVEL : ',640,420,0,0,0)

    drawTextNum(tingkatan2,650,400,0,0,0)

def play\_game():

    if crash\_wal\_player1 == False:

        play\_player1()

        rintangan(x\_r\_player1,30)

        rintangan(x\_r\_player1\_tambahan,s\_r\_player1)

    else:

        background(0)

        drawText('YOUR FINAL SCORE: ',110,450,255,255,255) *#player 1*

        drawTextNum(fix\_score\_player1,180,430,255,255,255) *# player 1*

        drawText('HIGHEST LEVEL : ',110,400,255,255,255) *#player 1*

        drawTextNum(tingkatan1,270,400,255,255,255) *# player 1*

    if crash\_wal\_player2 == False:

        play\_player2()

        rintangan(x\_r\_player2,30)

        rintangan(x\_r\_player2\_tambahan,s\_r\_player2)

    else:

        background(300)

        drawText('YOUR FINAL SCORE: ',410,450,255,255,255) *#player 2*

        drawTextNum(fix\_score\_player2,480,430,255,255,255) *# player 2*

        drawText('HIGHEST LEVEL : ',410,400,255,255,255) *#player 1*

        drawTextNum(tingkatan2,570,400,255,255,255) *# player 1*

    if (crash\_wal\_player1 and crash\_wal\_player2)==True:

        bg\_text(-40,0)

        drawTextBold("G A M E O V E R",260,255)

        drawText("Enter To Play",280,236,38, 33, 98)

def start\_game():

    glPushMatrix()

    glColor3b(36, 150, 127)

    glBegin(GL\_QUADS)

    glVertex2f(280, 220)

    glVertex2f(480, 220)

    glVertex2f(480, 280)

    glVertex2f(280, 280)

    glEnd()

    glColor3ub(0,0,0)

    glLineWidth(3)

    glBegin(GL\_LINE\_LOOP)

    glVertex2f(280, 220)

    glVertex2f(480, 220)

    glVertex2f(480, 280)

    glVertex2f(280, 280)

    glEnd()

    glPopMatrix()

    drawTextBold("P L A Y G A M E",295,250)

*#fungsi iterasi*

def iterate():

    glViewport(0, 0, 500, 500) *#utk mengatur area pandang*

    glMatrixMode(GL\_PROJECTION)

    glLoadIdentity()

    glOrtho(0.0, 650, 0.0, 500, 0.0, 1.0) *#utk mengatur berapa blok yang digunakan (skala) nilai x, y, z*

    glMatrixMode (GL\_MODELVIEW)

    glLoadIdentity()

def showScreen():

    glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT) *#utk membersihkan layar*

    glLoadIdentity()

    iterate()

    if play == False:

        start\_game()

    else:

        play\_game()

    glutSwapBuffers() *#utk membersihkan layar, double buffering*

def timer\_rintangan(value):

    global y\_rintangan, tr, x\_r\_player1, s\_r\_player1, x\_r\_player2, s\_r\_player2, cek\_lev,crash\_wal\_player1,crash\_wal\_player2,x\_r\_player1\_tambahan,x\_r\_player2\_tambahan

    if play == True:

        y\_rintangan -= 20

        if crash\_wal\_player1==False or crash\_wal\_player2 == False:

*# print(x\_r\_player1,x\_player1) # 390 collison, x = 140*

            if y\_rintangan < -450:

                cek\_lev += 1

                y\_rintangan = 100

                x\_r\_player1 = random.randrange(150,250,10)

                x\_r\_player1\_tambahan = random.randrange(150,250,10)

                x\_r\_player2 = random.randrange(450,550,10)

                x\_r\_player2\_tambahan = random.randrange(150,250,10)

                s\_r\_player1 = random.randrange(15,30)

                s\_r\_player2 = random.randrange(15,30)

            if (cek\_lev %2) == 0:

                tr-=100

            if tr <100:

                tr = 100

            if (x\_player1 in range(x\_r\_player1-50,x\_r\_player1+20))and(y\_rintangan < -390):

                crash\_wal\_player1 = True

            if (x\_player2 in range(x\_r\_player2-50,x\_r\_player2+20))and(y\_rintangan < -390):

                crash\_wal\_player2 = True

*#timer rintangan awal = 500, berkurang hingga mencapai 100*

    glutTimerFunc(tr,timer\_rintangan,0)

def timer(value):

    global y, kecepatan, score\_player1, cek\_point, cek\_y, cek\_kecepatan, tingkatan1, fix\_score\_player1

    if play==True:

        if crash\_wal\_player1 == False:

            y -= kecepatan

            if y < value :

*# 50 adalah tingkatan awal.... berkurang 5 hingga tingkatan akhir menjadi 20*

                y = cek\_y

            score\_player1 += kecepatan

            if score\_player1 % cek\_kecepatan == 0 :

                tingkatan1 += 1

                cek\_y -= 5

                cek\_point -= 5

                cek\_kecepatan += 10000

            if cek\_y < 20:

                cek\_y = 20

            if cek\_point < 10:

                cek\_point = 10

        else:

            fix\_score\_player1 = score\_player1

*#timer awal = 30, berkurang sebanyak 5 hingga mencapai 10*

    glutTimerFunc(cek\_point,timer,0)

def timer2(value):

    global y2, kecepatan2, score\_player2, cek\_point2, cek\_y2, cek\_kecepatan2, tingkatan2, fix\_score\_player2

    if play==True:

        if crash\_wal\_player2 == False:

            y2 -= kecepatan2

            if y2 < value :

*# 50 adalah tingkatan awal.... berkurang 5 hingga tingkatan akhir menjadi 20*

                y2 = cek\_y2

            score\_player2 += kecepatan2

            if score\_player2 % cek\_kecepatan2 == 0 :

                tingkatan2 += 1

                cek\_y2 -= 5

                cek\_point2 -= 5

                cek\_kecepatan2 += 10000

            if cek\_y2 < 20:

                cek\_y2 = 20

            if cek\_point2 < 10:

                cek\_point2 = 10

        else:

            fix\_score\_player2 = score\_player2

*#timer awal = 30, berkurang sebanyak 5 hingga mencapai 10*

    glutTimerFunc(cek\_point2,timer2,0)

def init():

    glClearColor(2,1,0, 2.0)

    gluOrtho2D(-500.0, 500.0, -500.0, 500.0)

def main ():

    glutInit() *#inisialisasi glut*

    glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGBA) *#utk mengatur display supaya berwarna*

    glutInitWindowSize(600, 500) *#utk mengatur ukuran window*

    glutInitWindowPosition(100,100) *#utk mengatur letak window*

    glutCreateWindow("2D Car Racing Game") *#utk memberi nama pada window*

    glutDisplayFunc(showScreen) *#utk fungsi callback*

    glutIdleFunc(showScreen) *#utk fungsi callback*

    glutKeyboardFunc(key\_player1)

    glutSpecialFunc(key\_player2)

    glutMouseFunc(mouse\_play\_game)

    timer\_rintangan(0)

    timer(0)

    timer2(0)

    init()

    glutMainLoop() *#fungsi yang akan memulai keseluruhan program*

main()