**Anterin Game : Distribusi Barang Sama Si Kurir Aja**

Untuk Memenuhi Mata Kuliah Pemrograman Berorientasi Objek

yang Diampu Oleh

Bapak Muhammad Jauharul Fuady, S.T., M.T.



OLEH

**Abdullah Sholum (220535608686)**

**Ahmad Ammar Musyaffa (220535601431)**

**Dafa Fadhilah Hilmi (2205356103090)**

**UNIVERSITAS NEGERI MALANG**

**FAKULTAS TEKNIK**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**APRIL 2023**

Kata Pengantar

Puji syukur Kami panjatkan kehadirat Tuhan Yang Maha Esa yang telah melimpahkan segala ilmu yang dapat kita manfaatkan, serta Ucapan terimakasih Kami sampaikan kepada semua pihak yang telah membantu dalam menyiapkan, menyusun, serta memberi masukan dan saran ketika penyusunan laporan “Anterin Game : Distribusi Barang Sama Si Kurir Aja”. Seperti yang kita tahu perkembangan teknologi selalu menarik untuk dibahas, termasuk dalam bidang permainan.

Dalam makalah yang berisi mengenai proses berkembangnya kecepatan pengiriman data yang terjadi disistem komputer, serta permasalahan yang dihadapi ketika proses perkembangan teknologi pengiriman data yang telah di analisis dan di rangkum dalam satu makalah. Kami berharap setelah membaca makalah ini pembaca dapat lebih memahami tentang prinsip dan cara kerja transfer data melalui PCIe.

Kami menyadari bahwa masih banyak kekurangan, kesalahan tatabahasa dan lain-lain dalam penyusunan makalah ini. Oleh sebab itu Kami sangat terbuka akan kritik dan saran yang bersifat membangun agar di masa yang akan datang Kami bisa membuat makalah yang lebih baik, agar pembaca bisa lebih tertarik dan memahami makalah yang kami buat.

**BAB I**

**PENDAHULUAN**

Seiring dengan berkembangnya teknologi yang semakin canggih, kini banyak orang yang memanfaatkan hal tersebut untuk semua bidang, bahkan ke permainan. Untuk kali ini kami akan mengkreasikan sebuah game sederhana yang juga menerapkan Pemrograman Berorientasi Objek, berbasis aplikasi yang berjudul “Anterin”. “Anterin” adalah sebuah game sederhana yang mengimpelentasikan seorang kurir yang sedang mendistribusikan barang-barang untuk ditujukan ke sebuah toko penjualan.

Barang-barang yang disediakan yaitu berupa barang yang ingin dijual ke sebuah toko dan didistribusikan melalui si kurir tersebut. Game sederhana kali ini kami menggunakan bahasa pemrograman Python untuk menjalankan logikanya. Kemudian kami juga mengimpor beberapa gambar yang ingin digunakan sebagai isi dari game tersebut. Pada game “Anterin” ini juga kami menggunakan pendefinisian class dari setiap object yang akan kami buat. Setiap object yang dibuat dalam game "Anterin" telah didefinisikan menjadi sebuah class dengan atribut dan methodnya masing-masing. Terdapat class untuk kurir, customer, toko, dan barang yang akan diantar. Setiap class memiliki atribut seperti koordinat posisi, kecepatan, jumlah barang, dan lain sebagainya. Selain itu, terdapat pula method untuk menggerakkan object, menghitung jarak, memuat atau menurunkan barang, dan lain sebagainya. Pemrograman Berorientasi Objek yang diterapkan dalam game "Anterin" memungkinkan setiap object untuk saling berinteraksi satu sama lain dan memberikan pengalaman bermain game yang lebih menyenangkan dan menarik. Dengan mengimpor beberapa gambar sebagai isi game, kami juga memberikan sentuhan visual yang menarik dan membuat game "Anterin" semakin menarik untuk dimainkan.

Kami memilih membuat game ini karena ditinjau dari kebanyakan game yang bertajuk kurir saat ini, karakter kurirnya masih menggunakan metode yang lama. Nah untuk game kami kali ini menggunakan metode yang berbeda yaitu si kurir ini mendistribusikan barang dari customer yang ingin menjual barangnya ke toko. Dengan begitu game sederhana ini mungkin akan membuat khalayak akan menjadi tertarik untuk memainkannya.

Manfaat dari game "Anterin" ini adalah memberikan pengalaman bermain game yang menyenangkan dan menarik, serta memperkenalkan konsep Pemrograman Berorientasi Objek bagi pemain yang tertarik dengan bidang pemrograman. Selain itu, game ini juga dapat meningkatkan keterampilan pemain dalam mengambil keputusan yang tepat dan mengelola waktu dengan efektif, karena pemain harus memilih rute yang tepat dan mengatur waktu pengiriman barang dengan baik. Dalam jangka panjang, game ini juga dapat membantu meningkatkan kemampuan kognitif dan logika pemain, serta mengajarkan nilai-nilai penting seperti tanggung jawab, kerja sama tim, dan ketepatan waktu.

**BAB II**

**RANCANGAN PENGEMBANGAN**

**2.1 KONSEP DASAR**

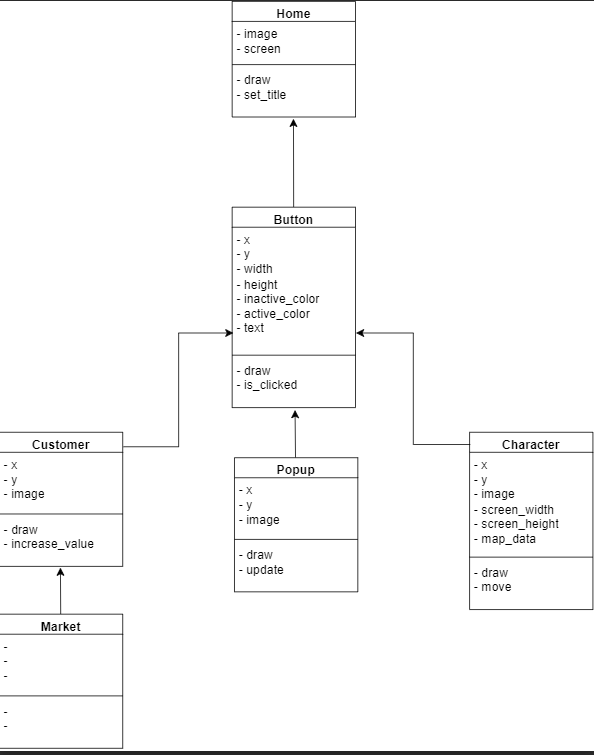
Anterin adalah game sederhana yang mengimplementasikan seorang kurir yang sedang mendistribusikan barang- barang customernya yang akan dijual ke suatu toko dengan nama “Si Kurir”. Beberapa sistem yang mungkin ada dalam Anterin, game sederhana yang mengimplementasikan seorang kurir yang sedang mendistribusikan barang-barang customernya yang akan dijual ke suatu toko dengan nama "Si Kurir" antara lain:

* Sistem Pengiriman : Sistem ini akan menampilkan rute terbaik yang dapat ditempuh oleh kurir untuk mengirimkan barang ke alamat tujuan pelanggan. Sistem ini juga dilengkapi dengan beberapa rintangan seperti adanya tembok sebagai penghalang dan rerumputan untuk memperlambat perjalanan.
* Sistem Score : Sistem ini akan menampilkan capaian si kurir ketika mengambil barang-barang dari customer. Sistem ini akan mencakup berapa banyak barang yang diambil oleh si kurir dan dikonversi menjadi skor.
* Sistem Timer : Sistem ini menampilkan batas waktu yang ada di dalam game tersebut yaitu 1 menit. Apabila 1 menit tersebut habis maka game akan selesai maka pemain akan dapat melihat skor akhir di dalam game yang dimainkannya.

**2.2 DIAGRAM CLASS**

Diagram class adalah salah satu jenis diagram UML (Unified Modeling Language) yang digunakan untuk memodelkan struktur objek dari suatu sistem atau program. Diagram class digunakan untuk menggambarkan kelas-kelas atau objek-objek yang terdapat dalam sistem, serta hubungan antara kelas-kelas atau objek-objek tersebut.

Diagram class menyajikan informasi tentang atribut (variabel) dan metode (fungsi) yang dimiliki oleh setiap kelas, serta hubungan antara kelas-kelas tersebut. Dalam pengimplementasian game sederhana “Anterin” kami juga menggunakan diagram kelas sebagai referensi yang dapat dilihat sebagai berikut :



Seperti yang kita lihat diatas, bahwa semua kelas dalam game “Anterin” memiliki keterhubungan.

* Class Home memiliki keterhubungan dengan semua kelas lainnya karena jika ingin membuka game nya kita harus menuju ke home terlebih dahulu.
* Class Market merupakan kelas turunan dari kelas customer.
* Semua class memiliki method draw karena kita harus menggunakannya untuk menampilkan di screen.

**BAB III**

**IMPLEMENTASI**

**3.1 LINGKUNGAN PENGEMBANGAN**

Visual Studio Code (VS Code) adalah lingkungan pengembangan terintegrasi (Integrated Development Environment/IDE) yang terkenal dan populer digunakan oleh para pengembang software. Lingkungan pengembangan ini menawarkan berbagai fitur dan dukungan yang lengkap untuk memudahkan proses pengembangan aplikasi atau software.

Mengapa kami menggunakan Visual Studio Code dalam lingkungan pengembangan aplikasi kami ? Karena VS Code memiliki antarmuka yang intuitif dan mudah digunakan, dengan banyak dukungan untuk bahasa pemrograman populer seperti JavaScript, Python, dan C#. Selain itu, VS Code juga dilengkapi dengan fitur debugging yang kuat, fitur refactor dan code completion yang otomatis, dan integrasi dengan sistem kontrol versi seperti Git. Kebetulan yang kami pakai dalam pengembangan aplikasi kami adalah Bahasa Pemrograman Python.

Selain fitur-fitur utama tersebut, VS Code juga menyediakan ekosistem yang luas melalui marketplace, di mana pengguna dapat menemukan dan menginstal banyak ekstensi dan plugin yang dapat memperluas kemampuan VS Code sesuai kebutuhan. Sehingga dengan menggunakan Visual Studio Code dapat mempermudah kerja kami.

**3.2 PACKAGE / MODUL**

Pada pengembangan aplikasi yang berupa game sederhana kali ini kami menggunakan beberapa library Python. Library pertama adalah pygame, yang digunakan untuk membuat game dan aplikasi multimedia interaktif. Dengan pygame, pengguna dapat membuat tampilan grafis, mengontrol pergerakan objek, dan menangani masukan dari pengguna seperti tombol keyboard dan mouse.

Library kedua yang digunakan adalah random, yang digunakan untuk menghasilkan bilangan acak. Library ini berguna dalam membuat game yang dinamis dan tidak terduga, karena dapat menghasilkan pergerakan dan kejadian yang berbeda setiap kali game dimainkan.

Selain itu, package ini juga menggunakan library tkinter, yang digunakan untuk membuat antarmuka pengguna grafis (GUI). Dengan tkinter, pengguna dapat membuat jendela, tombol, kotak teks, dan elemen lainnya yang digunakan dalam antarmuka pengguna.

Terakhir, package ini menggunakan library sys, yang digunakan untuk mengakses variabel dan fungsi yang terkait dengan sistem. Library ini berguna dalam mengelola input dan output, serta dalam mengakses informasi tentang sistem operasi yang digunakan.

Dengan menggabungkan semua library ini, pengguna dapat membuat aplikasi yang interaktif, dinamis, dan mudah digunakan. Package ini merupakan pilihan tepat bagi pengembang yang ingin menghasilkan aplikasi Python yang berkualitas tinggi.

**3.3 CARA KOMPILASI SOURCE CODE**

Untuk kompilasi source code aplikasi kami menjadikan file python menjadi file yang executable dengan melalui cara sebagai berikut :

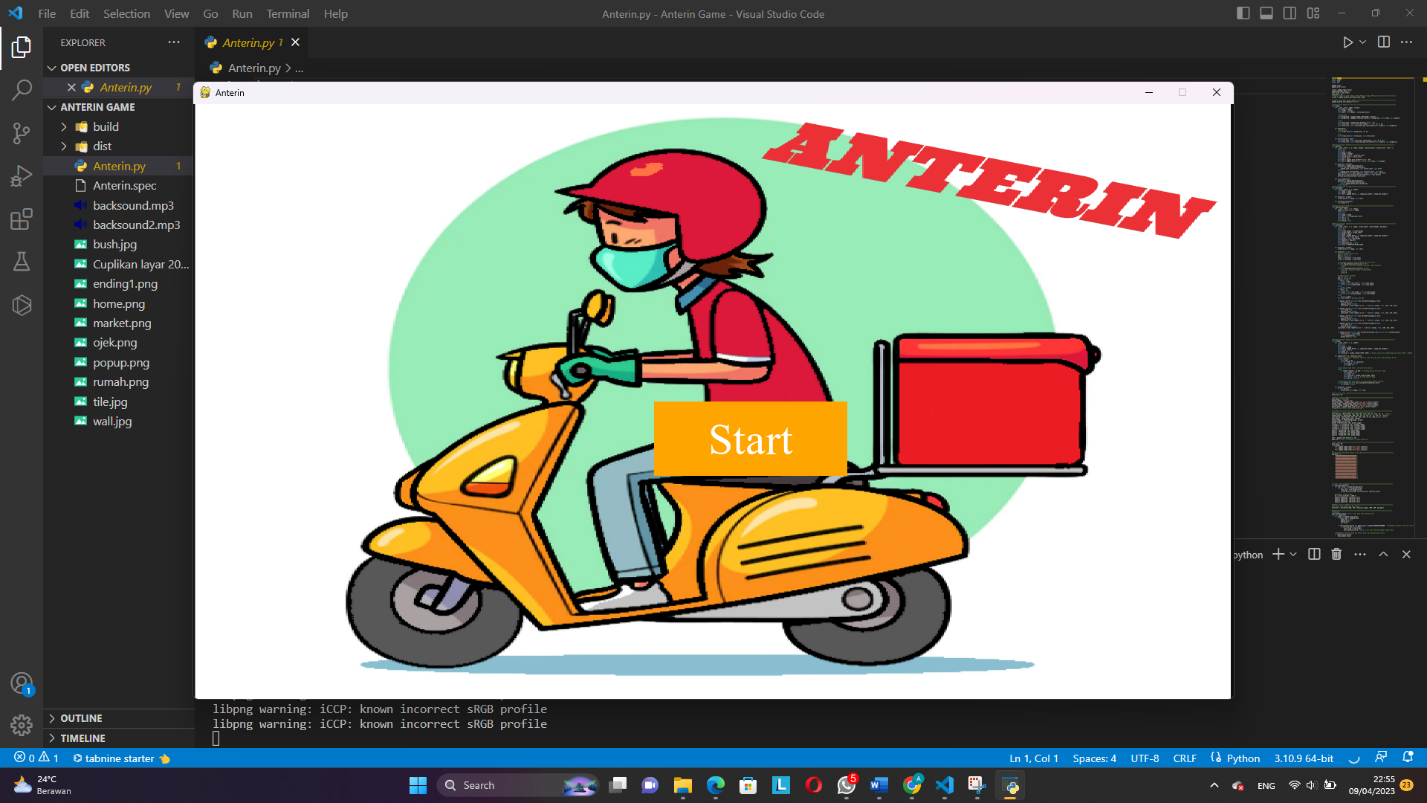
* Menginstall “PyInstaller” dengan menuliskan kode program sebagai berikut di terminal VS Code:



* Membuat file Python dengan kode yang akan kami ubah menjadi executable.
* Membuka terminal di VS Code dan kemudian mengarahkan ke direktori tempat file Python yang kami simpan.
* Menjalankan PyInstaller dengan menuliskan kode di terminal VS Code dengan sebagai berikut : “pyinstaller --onefile Anterin.py”.
* Kemudian menjalankan file executable di folder “dist” di dalam folder game kami.

**BAB IV**

**HASIL DAN PEMBAHASAN**

****

Gambaran awal dari proyek kami adalah tampilan home yang menampilkan judul game kami, yaitu Anterin. Halaman ini dirancang dengan sederhana, hanya menampilkan tombol start sebagai satu-satunya pilihan untuk memulai permainan Anterin. Tujuan dari tombol start adalah untuk memudahkan pengguna dalam memulai permainan, karena hanya dengan satu kali klik, pengguna akanlangsung memulai permainan tanpa perlu melalui proses lainnya. Dengan demikian, kami merancang tampilan home untuk memberikan pengalaman yang mudah dan intuitif bagi pengguna dalam memulai permainan Anterin.



Setelah pengguna mengklik tombol start pada halaman utama, pengguna akan diarahkan ke tampilan game Anterin. Di dalam tampilan ini, terdapat sebuah toko yang menjadi titik awal tukang ojek untuk mengambil barang dari pelanggan dan mengirimkannya ke toko untuk dijual. Selain itu, terdapat juga tampilan score yang akan terus bertambah setiap kali tukang ojek berhasil mengambil barang dari pelanggan dan mengirimkannya ke toko. Score ini menunjukkan seberapa banyak barang yang berhasil dikumpulkan oleh pengguna selama bermain.

Dalam game Anterin, pengguna diberikan waktu selama 1 menit untuk mengumpulkan sebanyak mungkin score. Waktu ini ditampilkan secara jelas pada tampilan game. Jika waktu habis sebelum pengguna berhasil mengumpulkan score yang cukup tinggi, maka game akan otomatis berhenti dan pengguna harus memulai dari awal jika ingin bermain lagi. Dengan tampilan game Anterin ini, kami berharap dapat memberikan pengalaman bermain yang menarik dan seru bagi pengguna, dengan menantang mereka untuk mengumpulkan score sebanyak-banyaknya dalam waktu yang terbatas.



Setelah waktu habis, pengguna akan diarahkan ke tampilan akhir game yang menampilkan pesan "Game Over". Selanjutnya, pada tampilan akhir ini akan ditampilkan score akhir yang telah diperoleh dari pengguna selama bermain. Score ini akan memberikan gambaran kepada pengguna tentang seberapa sukses pengguna dalam mengumpulkan barang dari pelanggan dan mengirimkannya ke toko. Setelah melihat score akhir, pengguna akan diberikan pilihan untuk keluar dari game melalui tombol "Exit". Tombol ini akan membawa pengguna kembali ke halaman utama atau keluar dari game sesuai dengan pilihan pengguna.

Dengan tampilan akhir ini, kami berharap pengguna dapat merasakan kepuasan dari permainan yang telah mereka mainkan dan mengetahui seberapa sukses mereka dalam mencapai target yang ditetapkan dalam game Anterin.

**BAB V**

**KESIMPULAN DAN SARAN**

Dalam kesimpulan, proyek membuat game sederhana tentang kurir yang menerapkan OOP dalam bahasa Python dan telah diexecutable merupakan sebuah proyek yang memperkuat pemahaman dan pengetahuan kita dalam pemrograman. Dalam proyek ini, kita telah berhasil membuat sebuah game yang interaktif dan menghibur, di mana pengguna dapat berinteraksi dengan objek-objek yang telah diimplementasikan dalam game.

Dalam proses membuat game ini, kita telah menggunakan konsep OOP dalam bahasa Python, di mana setiap objek di dalam game direpresentasikan sebagai sebuah kelas. Implementasi OOP dalam proyek ini memudahkan kita dalam mengorganisasi dan mengelola data dan fungsi dalam game.

Kita juga berhasil mengubah game Python yang telah kita buat menjadi executable dalam bentuk file yang dapat dijalankan pada berbagai platform. Hal ini memudahkan pengguna dalam mengakses dan memainkan game tersebut.

Namun, terdapat beberapa saran yang dapat kita lakukan untuk memperbaiki proyek ini. Pertama, kita dapat menambahkan lebih banyak fitur dalam game untuk membuatnya lebih menarik dan menantang bagi pengguna. Kedua, kita dapat meningkatkan kualitas grafis dan animasi dalam game agar lebih menarik dan menghibur. Ketiga, kita juga dapat menambahkan beberapa modul untuk mengelola kegagalan dalam game dan memberikan pengguna kesempatan untuk mencoba lagi.

Dalam keseluruhan, proyek ini merupakan proyek yang baik untuk memperdalam pemahaman kita dalam OOP dan bahasa Python. Dalam pengembangan game selanjutnya, kita dapat meningkatkan kualitas dan kuantitas fitur dalam game sehingga dapat memenuhi kebutuhan pengguna dan memberikan pengalaman bermain yang lebih baik lagi.

**BAB VI**

**LAMPIRAN**

**SOURCE CODE**

import pygame

import random

import time

import sys

pygame.init()

pygame.mixer.init()

clock = pygame.time.Clock()

CHARACTER\_START\_VALUE = 0

start\_time = time.time()

time\_left = 60

# Membuat tampilan game dengan lebar 1400 dan tinggi 800 pixel========================

screen = pygame.display.set\_mode((1400, 800))

# Memberi nama pada window game=======================================================

pygame.display.set\_caption("Anterin")

# menginisiasi home===================================================================

class Home:

    def \_\_init\_\_(self, image, screen):

        self.image = image

        self.screen = screen

        self.width, self.height = screen.get\_size()

        # Background

        self.background = pygame.image.load(image).convert()

        self.background = pygame.transform.scale(self.background, (self.width, self.height))

        # Text

        self.title\_font = pygame.font.SysFont('Arial', 32)

        self.title\_text = self.title\_font.render('', True, (0, 0, 0))

        self.title\_rect = self.title\_text.get\_rect(center=(self.width/2, self.height/3))

    def draw(self):

        # Background

        self.screen.blit(self.background, (0, 0))

        # Text

        self.screen.blit(self.title\_text, self.title\_rect)

    def set\_title(self, text):

        self.title\_text = self.title\_font.render(text, True, (0, 0, 0))

        self.title\_rect = self.title\_text.get\_rect(center=(self.width/2, self.height/3))

# Menginisialisasi button===========================================================

class Button:

    def \_\_init\_\_(self, x, y, width, height, inactive\_color, active\_color, text=''):

        self.x = x

        self.y = y

        self.width = width

        self.height = height

        self.inactive\_color = inactive\_color

        self.active\_color = active\_color

        self.text = text

        self.font = pygame.font.SysFont("Times", 60)

        self.rect = pygame.Rect(self.x, self.y, self.width, self.height)

    def draw(self, surface):

        mouse\_pos = pygame.mouse.get\_pos()

        if self.rect.collidepoint(mouse\_pos):

            pygame.draw.rect(surface, self.active\_color, self.rect)

        else:

            pygame.draw.rect(surface, self.inactive\_color, self.rect)

        text\_surface = self.font.render(self.text, True, (255, 255, 255))

        text\_rect = text\_surface.get\_rect(center=self.rect.center)

        surface.blit(text\_surface, text\_rect)

    def is\_clicked(self):

        mouse\_pos = pygame.mouse.get\_pos()

        if self.rect.collidepoint(mouse\_pos):

            return pygame.mouse.get\_pressed()[0]

        return False

# Menginisialisasi customer===========================================================

class Customer:

    def \_\_init\_\_(self, x, y, image):

        self.image = image

        self.image = image

        self.rect = pygame.Rect(x, y, image.get\_width(), image.get\_height())

    def draw(self, screen):

        screen.blit(self.image, self.rect)

    def increase\_value(self):

        self.value += 1

# Menginisialisasi Market==========================================================

class Market(Customer):

    def \_\_init\_\_(self, x, y, image):

        super().\_\_init\_\_(x, y, image)

        self.x = x

        self.y = y

        self.image = image

        self.rect = self.image.get\_rect()

        self.rect.x = x

        self.rect.y = y

        self.active = True

# Menginisialisasi Character==========================================================

class Character:

    def \_\_init\_\_(self, x, y, image, screen\_width, screen\_height, map\_data):

        self.x = x

        self.y = y

        self.screen\_height = screen\_height

        self.screen\_width = screen\_width

        self.image = image

        self.rect = pygame.Rect(x, y, image.get\_width(), image.get\_height())

        self.width = self.rect.width

        self.height = self.rect.height

        self.map\_data = map\_data

        self.score = 0

        self.touched\_market = False

        self.value = CHARACTER\_START\_VALUE

    def draw(self, screen):

        screen.blit(self.image, self.rect)

    def move(self, x, y):

        # kode validasi posisi baru

        new\_x = self.x + x

        new\_y = self.y + y

        tile\_x = int(new\_x / tile\_size)

        tile\_y = int(new\_y / tile\_size)

        # cek apakah karakter bisa bergerak ke posisi baru

        if self.map\_data[tile\_y][tile\_x] == "W":

            # karakter tidak bisa menembus dinding, tidak bergerak

            return

        elif self.map\_data[tile\_y][tile\_x] == "B":

            # karakter bergerak lambat di semak-semak

            x \*= 0.5

            y \*= 0.5

        # update posisi karakter

        new\_x = self.x + x

        new\_y = self.y + y

        if new\_x < 160:

            self.x = 160

        elif new\_x + self.rect.width > self.screen\_width:

            self.x = self.screen\_width - self.rect.width

        else:

            self.x = new\_x

        if new\_y < 0:

            self.y = 0

        elif new\_y + self.rect.height > self.screen\_height:

            self.y = self.screen\_height - self.rect.height

        else:

            self.y = new\_y

        self.rect.center = (self.x, self.y)

        if popup\_1.active and self.rect.colliderect(popup\_1.rect):

            self.value += 1

            popup\_1.active = False

            score\_text = font.render("Score: " + str(self.value), True, (255, 255, 255))

        if popup\_2.active and self.rect.colliderect(popup\_2.rect):

            self.value += 1

            popup\_2.active = False

            score\_text = font.render("Score: " + str(self.value), True, (255, 255, 255))

        if popup\_3.active and self.rect.colliderect(popup\_3.rect):

            self.value += 1

            popup\_3.active = False

            score\_text = font.render("Score: " + str(self.value), True, (255, 255, 255))

        if popup\_4.active and self.rect.colliderect(popup\_4.rect):

            self.value += 1

            popup\_4.active = False

        score\_text = font.render("Score: " + str(self.value), True, (255, 255, 255))

        if market.active and self.rect.colliderect(market.rect) and not self.touched\_market:

            print("nilai", self.value)

            self.touched\_market = True

            market.active = False

# Menginisialisasi Popup=============================================================

class Popup:

    def \_\_init\_\_(self, x, y, image):

        self.x = x

        self.y = y

        self.image = image

        self.rect = pygame.Rect(x, y, image.get\_width(), image.get\_height())

        self.active = False

        self.timer = 0

        self.duration = random.randint(3000, 8000)  # Durasi muncul dan menghilang acak antara 1000 - 5000ms

    def update(self, dt, character\_rect):

        # Jika popup aktif, tambahkan waktu ke timer dan cek jika sudah mencapai durasi

        if self.active:

            self.timer += dt

            if self.timer >= self.duration:

                self.active = False

                self.timer = 0

        # Jika popup tidak aktif, cek jika harus muncul

        else:

            if random.random() < 0.005:  # Peluang muncul 0.5% per frame

                self.active = True

                self.timer = 0

                self.duration = random.randint(3000, 8000)

                self.rect.x = self.x  # Letak muncul tetap

                self.rect.y = self.y

        # Cek apakah popup masih aktif dan bersentuhan dengan karakter

        if self.active and self.rect.colliderect(character\_rect):

            self.active = False

    def draw(self, screen):

        if self.active:

            screen.blit(self.image, self.rect)

#ukuran grid untuk tile===========================================================

tile\_size = 50

# Memuat resource gambar==========================================================

home\_image = "home.png"

ending\_image = "ending1.png"

character\_image = pygame.image.load("ojek.png").convert\_alpha()

market\_image = pygame.image.load("market.png").convert\_alpha()

customer\_image = pygame.image.load("rumah.png").convert\_alpha()

popup\_image = pygame.image.load("popup.png").convert\_alpha()

# pygame.mixer.music.load("backsound2.mp3")

# Inisiasi=======================================================================

#Button(x, y, width, height, inactive\_color, active\_color)

start\_button = Button(620, 400, 260, 100, (255, 165, 0), (204, 85, 0), 'Start')

repeat\_button = Button(540, 400, 160, 80, (0, 128, 0), (0, 255, 0), 'Repeat')

exit\_button = Button(580, 400, 260, 100, (255, 0, 0), (255, 0, 0), 'Exit')

home\_tampil = Home(home\_image, screen)

ending\_tampil = Home(ending\_image, screen)

ending\_tampil.set\_title('')

market = Market(1180, 300, market\_image)

customer\_1 = Customer(0, 50, customer\_image)

customer\_2 = Customer(0, 250, customer\_image)

customer\_3 = Customer(0, 450, customer\_image)

customer\_4 = Customer(0, 650, customer\_image)

popup\_1 = Popup(100, 50, popup\_image)

popup\_2 = Popup(100, 250, popup\_image)

popup\_3 = Popup(100, 450, popup\_image)

popup\_4 = Popup(100, 650, popup\_image)

#

font = pygame.font.Font(None, 36)

game\_over = False  # tambahkan variabel game\_over

# tile setup=======================================================================

tile\_size = 50

tile\_images = {

    "W": pygame.image.load("wall.jpg").convert(),

    "R": pygame.image.load("tile.jpg").convert(),

    "B": pygame.image.load("bush.jpg").convert()

}

# definisikan map dalam bentuk array 2D============================================

map\_data = [

    "WWWWWWWWWWWWWWWWWWWWWWWWWWWW",

    "WBBRRRRRRRBBBBBBBBRRRRBBBRWW",

    "WBBBBRRRRRWWWWWWWWRRWRWRRRRW",

    "WBRWWRRBBBBRRRBBRRRRWRRRRRRW",

    "WBRRRRWWRRRWWRRRBBRRRWRRRRRW",

    "WBRRRRRBBBRRRRRWWWRBRRRRRRRW",

    "WBRWRRRWWWWWWRRRRRRRBRRRWWWW",

    "WBRWRBBRRRRRRRWRRRRRRBRRRRRW",

    "WBRWRWWRBBBBBRWRWWWWWRRRWWWW",

    "WBRRRWWRRRRRRRWRRRBBBBRRWWWW",

    "WBRRRBBRRWWWWRRRWWRRRRRRRRRW",

    "WBRRRRRRRRRRRRBBBRRWWRRRRRRW",

    "WBRWWWRRRBBBRRRRRRRRRRWWRRRW",

    "WBBRRRRWWRWWWWWWWWRRRRWWWRRW",

    "WBBRRWRRRBBBBBBBBRRRRBBBBRWW",

    "WWWWWWWWWWWWWWWWWWWWWWWWWWWW"]

# fungsi untuk menggambar map=====================================================

def draw\_map(map\_data):

    for row, tiles in enumerate(map\_data):

        for col, tile in enumerate(tiles):

            tile\_img = tile\_images[tile]

            screen.blit(tile\_img, (col\*tile\_size, row\*tile\_size))

    #memanggil fungsi popup

    dt = clock.tick(60) / 1000.0

    popup\_1.update(dt, character.rect)

    popup\_2.update(dt, character.rect)

    popup\_3.update(dt, character.rect)

    popup\_4.update(dt, character.rect)

# Memuat resource gambar==========================================================

#tak taruh sini karena map ada diatas

character = Character(1300, 550, character\_image, 1400, 800, map\_data)

#gameloop========================================================================

# Gameloop

start\_button\_active = True  # set awal start button aktif

while not game\_over:

    for event in pygame.event.get():

        if event.type == pygame.QUIT:

            game\_over = True

            pygame.quit()

            sys.exit()

        if start\_button\_active and event.type == pygame.MOUSEBUTTONDOWN:  # tambahkan kondisi jika start button masih aktif

            if start\_button.is\_clicked():

                start\_time = time.time()

                start\_button\_active = False  # set start button menjadi tidak aktif

    # Menggambar tampilan Home dan button jika start button masih aktif

    if start\_button\_active:

        home\_tampil.draw()

        start\_button.draw(screen)

    else:  # jika start button tidak aktif

        # Mengatur waktu dan skor

        current\_time = time.time()

        elapsed\_time = current\_time - start\_time

        time\_left = max(5 -int(elapsed\_time),0)

        score\_text = font.render("Score: " + str(character.value), True, (255, 255, 255))

        time\_text = font.render("Time Left: " + str(time\_left), True, (255, 255, 255))

        if time\_left == 0:

            game\_over = True

            screen.fill((0, 0, 0))

            ending\_tampil.draw()

            font = pygame.font.SysFont("Times", 50)

            score\_text = font.render("Score: " + str(character.value), True, (255, 255, 255))

            screen.blit(score\_text, (643, 300))

            font = pygame.font.SysFont("Arial-Bold", 150)

            gameover\_text = font.render("Game Over ", True, (255, 255, 255))

            screen.blit(gameover\_text, (440, 150))

            # Menambahkan tombol keluar

            exit\_button.draw(screen)

            pygame.display.update()

            exit\_button\_active = True

            # Menunggu tombol keluar ditekan

            while True:

                for event in pygame.event.get():

                    if exit\_button\_active and event.type == pygame.MOUSEBUTTONDOWN:

                        if exit\_button.is\_clicked():

                            game\_over = True

                            pygame.quit()

                            sys.exit()

            time.sleep(5)

            pygame.quit()

            sys.exit()

        # Keluar dari game jika player menekan tombol close

        for event in pygame.event.get():

            if event.type == pygame.QUIT:

                game\_over = True

                pygame.quit()

                sys.exit()

        # set background color

        screen.fill((0, 0, 0))

        #fungsi menggerakkan character lewat masukan keyboard panah

        keys = pygame.key.get\_pressed()

        if keys[pygame.K\_LEFT]:

            character.move(-5, 0)

        if keys[pygame.K\_RIGHT]:

            character.move(5, 0)

        if keys[pygame.K\_UP]:

            character.move(0, -5)

        if keys[pygame.K\_DOWN]:

            character.move(0, 5)

        # Menggambar/memanggai Map, Class dan fungsi

        draw\_map(map\_data)

        character.draw(screen)

        market.draw(screen)

        customer\_1.draw(screen)

        customer\_2.draw(screen)

        customer\_3.draw(screen)

        customer\_4.draw(screen)

        popup\_1.draw(screen)

        popup\_2.draw(screen)

        popup\_3.draw(screen)

        popup\_4.draw(screen)

        screen.blit(score\_text, (10, 10))

        screen.blit(time\_text, (600, 10))

    pygame.display.update()

while True:

    for event in pygame.event.get():

        if event.type == pygame.QUIT:

            pygame.quit()

            exit(0)

    pygame.display.flip()