



Manual Pygame

Pygame merupakan modul **python** yang dapat digunakan untuk pembuatan Game. Pada dokumen ini akan dijelaskan cara instalasi **pygame**, penggunaan atau import **pygame** dalam file code **python**, property-property dasar termasuk pengenalan event di dalam **pygame**. Dokumen ini diakhiri dengan contoh pembuatan game sederhana.

Instalasi Modul Pygame

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

a. Siapkan file pygame.whl

Download library **pygame** di <https://www.lfd.uci.edu/~gohlke/pythonlibs/> .Sesuaikan versi **python** yang terinstall dengan versi library **pygame** yang akan di download. Contoh **python v3.5** telah terinstall, maka download library **pygame** dengan **cp35m-win32**.

b. Rubah file pygame.whl menjadi .zip

Rubah extension file library **pygame** lalu extract isi library tersebut.

c. Copy file pygame yang dibutuhkan

i. masuk ke dalam directory **python**

(C:\Users\#username\AppData\Local\Programs\Python\Python35-32)

ii. masuk kedalam folder “**include**” dan buat folder baru bernama “**pygame**”.

iii. Di dalam folder hasil extract file library **pygame** yang sudah didownload, masuk ke “**pygame-1.9.4.data\header**”, copy semua file di dalam folder tersebut dan masukkan ke dalam folder:

C:\Users\#username\AppData\Local\Programs\Python\Python35-32\include\pygame

iv. kembali ke folder hasil extract file library **pygame** tadi, copy folder “**pygame**” dan “**pygame-1.9.4.dist-info**” kedalam:

C:\Users\#username\AppData\Local\Programs\Python\Python35-32\Lib\site-packages

d. Cek hasil instalasi

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

Instalasi Subclass

Untuk dapat menggunakan method-method ataupun event yang terdapat pada modul pygame, maka terlebih dahulu dilakukan perintah import pygame, seperti contoh berikut ini :

```
import pygame
```

Contoh penggunaan pygame pertama kali, dapat dilihat sebagai berikut :

```
import pygame
pygame.init()
gameDisplay = pygame.display.set_mode((800,600))
gameDisplay.fill((255,255,255))
pygame.display.set_caption("Pygame")
pygame.display.update()
```

Pygame sendiri memerlukan syntax pembangun (**pygame.init**) untuk membuat sebuah window screen dasar dan menghidupkan setiap modul milik pygame, modul tersebut nantinya akan mengacu pada variable gameDisplay (seperti contoh diatas) atau disebut juga dengan **Surface**.

Property Pygame

Berikut ini adalah property-property dasar yang digunakan untuk pembuatan game dengan modul pygame.

a. Pewarnaan

Warna dalam Pygame tersusun dari kumpulan angka (0-255) yang menggambarkan komposisi warna merah, hijau, biru (RGB).

```
gameDisplay.fill((255,255,255))
```

b. Event

```
for event in pygame.event.get():
    if event.type == pygame.QUIT:
        print ("Close")
    if event.type == pygame.KEYDOWN:
        if event.key == pygame.K_SPACE:
            print ("GO!")
```

event.type berfungsi sebagai kondisi, syntax-syntax dibawah event tersebut akan dieksekusi jika *branchcondition* pada event.type bernilai True.

Jika Window Pygame di tutup (quit) maka akan menjalankan perintah print (“Close”), dan jika tombol space ditekan (keydown) maka akan menjalankan perintah print (“GO!”).

c. Object

Pygame telah menyediakan beberapa objek yang dapat digunakan. Berikut objek-objek pada pygame tersebut:

Object	Syntax	Contoh
Rectangle	<code>pygame.draw.rect (Surface, warna, [x, y, lebar, panjang])</code>	<code>pygame.draw.rect (gameDisplay, (255,255,0), [400,300,100,100])</code>
Circle	<code>pygame.draw.circle (Surface, warna, (x, y), radius)</code>	<code>pygame.draw.circle (gameDisplay, (255,255,0), (400,300), 30)</code>
Polygon	<code>pygame.draw.polygon (Surface, warna, [[x1, y1], [x2, y2], [x3, y3]])</code>	<code>pygame.draw.polygon (gameDisplay, (255,255,0), [[100, 100], [100, 400], [400, 300]])</code>
Line	<code>pygame.draw.line (Surface, warna, [x1, y1], [x2, y2], ketebalan)</code>	<code>pygame.draw.line (gameDisplay, (255,255,0), [100, 100], [200, 200], 10)</code>
Ellipse	<code>pygame.draw.ellipse (Surface, warna, [x1, y1, lebar, panjang])</code>	<code>pygame.draw.ellipse (gameDisplay, (255,255,0), [100, 100, 200, 50])</code>

d. Menambahkan gambar

Untuk menambahkan suatu gambar kedalam Pygame memerlukan 2 perintah dasar, yaitu ***pygame.image.load()*** dan ***blit()***.

Syntax	Contoh
<pre>variabel = pygame.image.load('nama file img.png') Surface.blit(variabel, (x,y))</pre>	<pre>img1 = pygame.image.load('a.png') gameDisplay.blit(img1, (0,0))</pre>

e. FPS (Frame Per Second)

Pemberian FPS dalam pygame dapat menggunakan ***time.clock()*** milik pygame. FPS diperlukan untuk memberikan efek animasi per detik, jika tidak memberikan efek FPS maka animasi akan berjalan langsung menuju hasil akhir (animasi akhir).

```

import pygame

pygame.init()
gameDisplay = pygame.display.set_mode((800,600))
gameDisplay.fill((255,255,255))
pygame.display.set_caption("Pygame")
clock = pygame.time.Clock()

while play:
    .
    .
    .
    .
    pygame.display.update()
    clock.tick(30)

pygame.quit()

```

f. Pergerakan Objek

Pergerakan disini bisa dibagi akan 2 jenis, pergerakan dengan Statis dan Dinamis (dengan bantuan inputan event). Setiap pergerakan dalam Pygame harus lah berada dalam looping agar setiap pergerakan dapat dimuat ulang untuk ditampilkan.

1. Pergerakan Statis

Pergerakan ini sangat tidak bergantung akan inputan suatu event dan terus dieksekusi saat pertama kali dijalankan, contoh objek akan bergerak dari titik A ke B sesaat setelah program berjalan.

```

import pygame

pygame.init()
gameDisplay = pygame.display.set_mode((800,600))
gameDisplay.fill((255,255,255))
pygame.display.set_caption("Pygame")
clock = pygame.time.Clock()

def bendaA(kordinat_y):
    pygame.draw.rect(gameDisplay, (255,255,0),
    [400,kordinat_y,100,100])
play = True
titik_awal_y = 100
while play:
    gameDisplay.fill((255,255,255))
    bendaA(titik_awal_y)
    if titik_awal_y < 300:
        ytitik_awal_y += 5

    pygame.display.update()
    clock.tick(30)

pygame.quit()

```

Jika program diatas dijalankan maka objek (rect/persegi) akan berjalan dari titik awal y=100 sampai y=300. Karena objek akan berpindah dari titik A ke B secara terus menerus maka didalam looping tersebut, maka screen harus dibersihkan, dan objek dipanggil kembali untuk membuat efek animasi dapat berjalan sempurna.

2. Pergerakan Dinamis

Berbeda dengan pergerakan Statis, pergerakan ini **sangat bergantung akan inputan suatu event terpenuhi**. Biasanya pergerakan ini digunakan untuk membuat suatu objek bisa bergerak sesuai keinginan user untuk berjalan dari titik A ke B atau C.

```
import pygame
pygame.init()
gameDisplay = pygame.display.set_mode((800,600))
gameDisplay.fill((255,255,255))
pygame.display.set_caption("Pygame")
clock = pygame.time.Clock()

def bendaA(kordinat_y):
    pygame.draw.rect(gameDisplay, (255,255,0),
        [400,kordinat_y,100,100])

play = True
titik_awal_y = 100
temp_y = 0

while play:
    gameDisplay.fill((255,255,255))

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            play = False
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_UP:
                temp_y = -5
            elif event.key == pygame.K_DOWN:
                temp_y = 5
        if event.type == pygame.KEYUP:
            if event.key == pygame.K_UP or event.key ==
pygame.K_DOWN:
                temp_y = 0

    titik_awal_y += temp_y
    bendaA(ynya)

    pygame.display.update()
    clock.tick(30)

pygame.quit()
```

Program diatas merupakan perkembangan dari contoh code pergerakan statis, penambahan event.get KEYDOWN dan KEYUP merupakan kunci dari pergerakan dinamisnya. Jika tombol panah atas ditekan maka objek (rect) akan berjalan ke atas, jika tombol panah bawah ditekan maka sebaliknya, dan KEYDOWN sendiri bertugas untuk memberi kondisi dimana jika tombol sudah tidak ditekan maka objek akan berhenti di posisi saat ini (tidak naik dan turun).

Contoh Game

Berikut adalah contoh game yang dibuat dengan menggunakan modul pygame. Game yang dibuat adalah game untuk pergerakan pesawat yang menghindari meteor. Meteor-meteor ini akan bergerak jatuh secara acak.

```
import pygame
import random

pygame.init()
gameDisplay = pygame.display.set_mode((800,600))
gameDisplay.fill((255,255,255))
pygame.display.set_caption("Pygame")
clock = pygame.time.Clock()

play = True

kordinat_x = 350
temp_x = 0

rocket = pygame.image.load('coba.png')
rocket = pygame.transform.scale(rocket, (100, 200))
gameDisplay.blit(rocket, (kordinat_x,380))

kordinat_x_meteor = random.randrange(0,600)
kordinat_y_meteor = -500

meteor = pygame.image.load('coba2.png')
meteor = pygame.transform.scale(meteor, (150, 150))
gameDisplay.blit(meteor, (kordinat_x_meteor,kordinat_y_meteor))

while play:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            play = False
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_LEFT:
                temp_x = -5
            elif event.key == pygame.K_RIGHT:
                temp_x = 5
        if event.type == pygame.KEYUP:
            if event.key == pygame.K_LEFT or event.key ==
pygame.K_RIGHT:
                temp_x = 0
```

```

gameDisplay.fill((255,255,255))

gameDisplay.blit(meteor, (kordinat_x_meteor,kordinat_y_meteor))
gameDisplay.blit(rocket, (kordinat_x,380))

kordinat_y_meteor += 5
kordinat_x += temp_x

if kordinat_y_meteor > 600:
    kordinat_y_meteor = -600
    kordinat_x_meteor = random.randrange(0,600)

pygame.display.update()
clock.tick(60)

pygame.quit()

```

Sebagai contoh, program diatas menampilkan permainan sederhana yang memanfaatkan fungsi `image.load`, `event.get`, perpindahan dinamis dan statis.

```

kordinat_x = 350
temp_x = 0

rocket = pygame.image.load('coba.png')
rocket = pygame.transform.scale(rocket, (100, 200))
gameDisplay.blit(rocket, (kordinat_x,380))

kordinat_x_meteor = random.randrange(0,600)
kordinat_y_meteor = -500

meteor = pygame.image.load('coba2.png')
meteor = pygame.transform.scale(meteor, (150, 150))
gameDisplay.blit(meteor, (kordinat_x_meteor,kordinat_y_meteor))

```

Pergerakan dinamis untuk bagian roket, dan statis untuk meteor. Roket dapat digerakan dengan menekan *arrow left/right* untuk mengerakannya, sedangkan meteor akan turun dari atas ke bawah dengan posisi x secara acak (dengan memanfaatkan library random).

```

while play:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            play = False
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_LEFT:
                temp_x = -5
            elif event.key == pygame.K_RIGHT:
                temp_x = 5
        if event.type == pygame.KEYUP:
            if event.key == pygame.K_LEFT or event.key ==
pygame.K_RIGHT:
                temp_x = 0

```

```
gameDisplay.fill((255,255,255))

gameDisplay.blit(meteor, (kordinat_x_meteor,kordinat_y_meteor))
gameDisplay.blit(rocket, (kordinat_x,380))

kordinat_y_meteor += 5
kordinat_x += temp_x

if kordinat_y_meteor > 600:
    kordinat_y_meteor = -600
    kordinat_x_meteor = random.randrange(0,600)
```

Pergerakan tadi berada dalam loop awal, setelah fungsi event akan memulai memuat ulang screen pygame untuk menampilkan perubahan yang terjadi selama loop, sedangkan meteor akan terus berjalan kewabah hingga batas yang ditentukan lalu kembali ke titik y awal namun x yang berbeda secara terus menerus.

