

**LAPORAN TUGAS BESAR
PEMROGRAMAN BERORIENTASI OBJEK**

KOCENG LONCAT



KELOMPOK 10 :

**121140108 – RADOT YOHANES NABABAN
121140071 – MAHARANI TRIZA PUTRI
121140142 – FADHIL FIROOS**

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI PRODUKSI DAN INDUSTRI
INSTITUT TEKNOLOGI SUMATERA
2023**

DAFTAR ISI

A. DESKRIPSI PENUGASAN.....	3
B. NAMA GAME	3
C. KATEGORI GAME.....	3
D. ENTITAS	3
E. DESKRIPSI GAME	3
F. TAMPILAN GAME	4
G. KELAS	8
H. OBJEK	10
I. ENKAPSULASI.....	10
J. PEWARISAN	10
K. POLIMORFISME.....	10
L. ABSTRAKSI.....	11
M. PENANGANAN KESALAHAN	13
N. GRAPHICAL USER INTERFACE	14
O. KODE.....	14
P. LAMPIRAN	17

A. DESKRIPSI PENUGASAN

Penugasan dalam tugas besar kelompok kami terbagi dalam 3 penugasan yaitu, membuat codingan, mendesain game, dan membuat laporan. Untuk penugasan membuat codingan adalah Fadhil Firoos dan Radot Yohanes Nababan. Mereka berbagi tugas dalam proses pembuatan codingan, tugasnya meliputi menulis kode untuk mengendalikan karakter kucing, memperhitungkan hal – hal dalam permainan, dan menambahkan audio serta visual. Untuk penugasan mendesain game ialah Fadhil Firoos, Radot Yohanes Nababan, dan Maharani Triza Putri. Mereka berbagi tugas dalam proses mendesain game, tugasnya meliputi membuat game tersebut menjadi lebih menarik, membuat karakter kucing dan latar belakangnya, membuat animasi dalam karakter kucing, serta memilih efek suara yang sesuai untuk karakter kucing. Untuk penugasan pembuatan laporan ialah Maharani Triza Putri. Tugasnya adalah membuat laporan, menyusun rencana, dan menyajikan hasil akhir dari game secara menyeluruh dalam sebuah laporan.

B. NAMA GAME

Game yang kami buat ini bernama “Koceng Loncat”. Alasan kami memilih nama tersebut karena karakter utama pada game ini ialah seekor kucing yang sedang berlari – lari dan ketika melihat sebuah rintangan maka ia akan meloncat. Nama dari “Koceng Loncat” sendiri mudah diingat dan unik sehingga bisa mudah untuk menarik perhatian pemain.

C. KATEGORI GAME

Kategori dari game “Koceng Loncat” ialah runner. Runner sendiri cocok dengan konsep game yaitu melompat dan menghindari sebuah rintangan, serta kategori ini mudah dipahami dan populer di kalangan pemain.

D. ENTITAS

Berikut ialah entitas yang terdapat pada game “Koceng Loncat” :

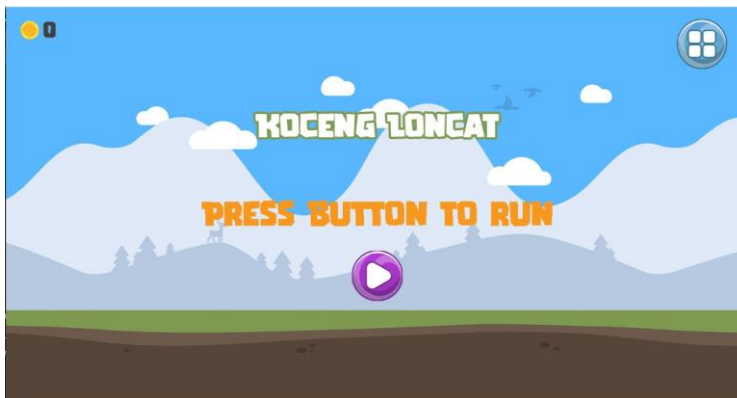
1. Koceng
Disini Koceng menjadi objek utama pada permainan yang bisa dikontrol oleh pengguna.
2. Obstacle
Disini Obstacle menjadi penghalang koceng sebagai objek utama, munculnya obstacle ini secara acak.
3. Tombol Play, Menu Tema, Setting, Resume, Buy dan Back
Tombol ini berfungsi masing - masing sebagai memulai permainan, menghentikan game dan melanjutkan game, membeli tema, dan kembali ke dalam game setelah di pause.

E. DESKRIPSI GAME

“Koceng Loncat” adalah sebuah game sederhana yang menghadirkan petualangan seru dari sudut pandang seekor kucing yang bisa melompat melewati rintangan. Game ini dikembangkan menggunakan library Pygame, yang merupakan library pemrograman dalam bahasa Python untuk membuat game 2D. Konsep permainan dari “Koceng Loncat” adalah pemain mengendalikan seekor kucing yang berlari secara otomatis di layar, dan tugas pemain adalah mengendalikan kucing tersebut untuk melompati rintangan dengan menggunakan tombol “space” yang muncul di jalannya. Pemain juga harus mendapatkan koin yang mencakupi untuk memainkan tema selanjutnya.

F. TAMPILAN GAME

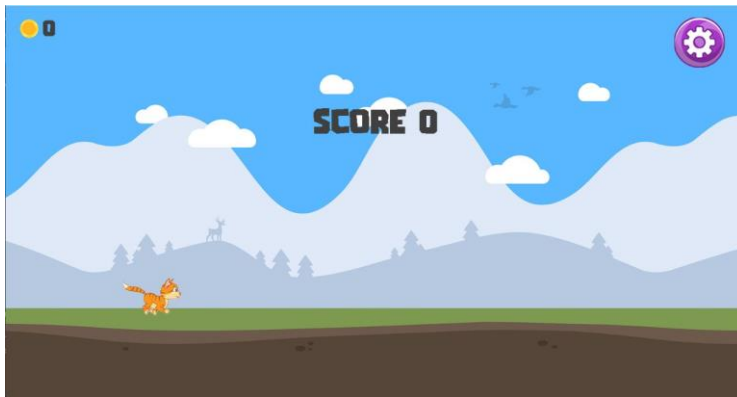
Berikut merupakan tampilan - tampilan pada game “Koceng Loncat” :



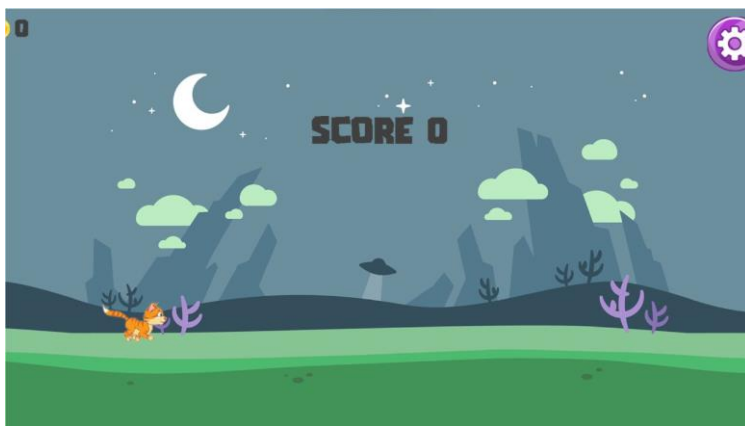
Tampilan diatas ialah tampilan halaman utama game pada tema 1 atau pada tema siang hari.



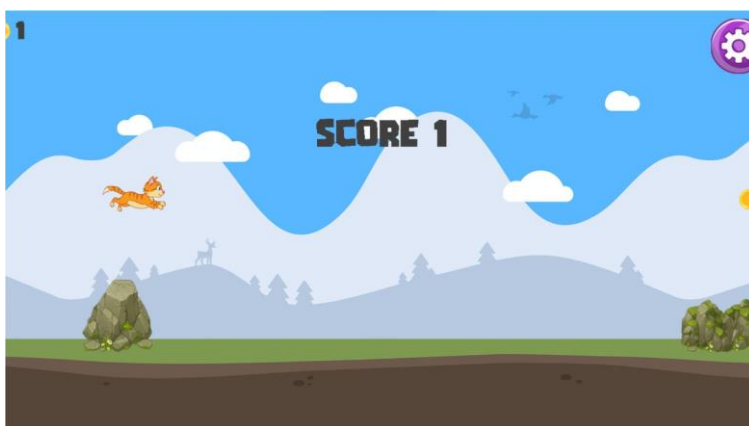
Tampilan diatas ialah tampilan halaman utama game pada tema 2 atau pada tema malam hari.



Tampilan diatas ialah tampilan ketika game sudah dimainkan dengan menggunakan tema 1 atau tema siang hari.



Tampilan diatas ialah tampilan ketika game sudah dimainkan dengan menggunakan tema 2 atau tema malam hari.



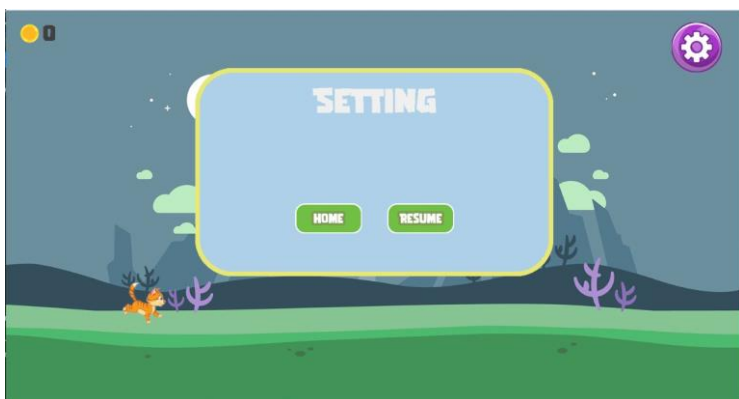
Tampilan diatas ialah tampilan ketika menghadapi obstacle dengan menggunakan tema 1 atau tema siang hari.



Tampilan diatas ialah tampilan ketika menghadapi obstacle dengan menggunakan tema 2 atau tema malam hari.



Tampilan diatas ialah tampilan ketika game di pause, ketika game di pause maka keluarlah pengaturan atau setting yang mempunyai opsi melanjutkan game atau kembali ke halaman utama. Pada tampilan diatas menggunakan tema 1 atau tema siang hari.



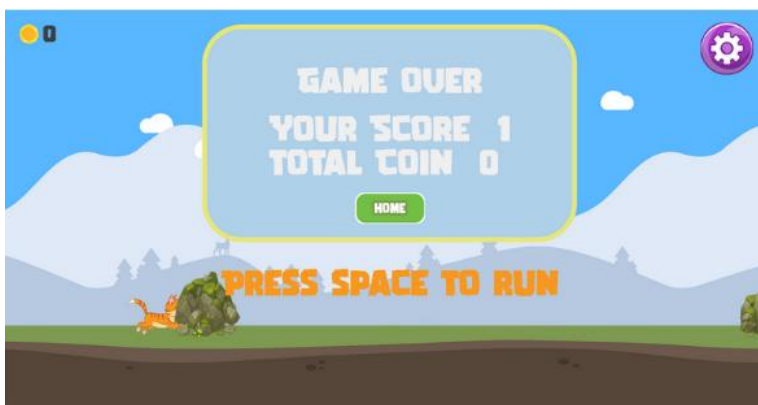
Tampilan diatas ialah tampilan ketika game di pause, ketika game di pause maka keluarlah pengaturan atau setting yang mempunyai opsi melanjutkan game atau kembali ke halaman utama. Pada tampilan diatas menggunakan tema 2 atau tema malam hari.



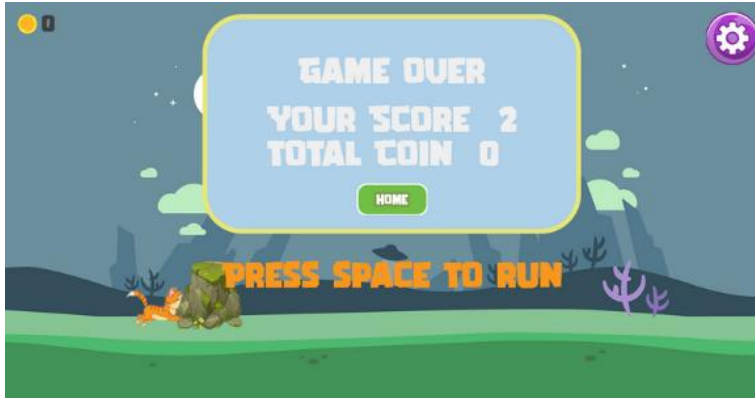
Tampilan diatas adalah tampilan shop ketika tema sebelum membeli tema baru, tema tersebut dibeli menggunakan koin.



Tampilan diatas ialah tampilan shop setelah membeli tema baru.



Tampilan diatas ialah tampilan game over atau akhir dari game, pada tampilan game over terdapat hasil akhir score dan koin. Pada tampilan diatas menggunakan tema 1 atau tema siang hari.



Tampilan diatas ialah tampilan game over atau akhir dari game, pada tampilan game over terdapat hasil akhir score dan koin. Pada tampilan diatas menggunakan tema 1 atau tema siang hari.

G. KELAS

Berikut ialah kelas - kelas yang terdapat pada game “Koceng Loncat” :

1. Bg()

Class Bg berisikan mengenai peraturan seluruh penggunaan tema atau latar belakang, seperti alur munculnya setiap tema.

2. Bg_1()

Class Bg_1() ialah turunan dari kelas Bg untuk menampilkan tema 1.

3. Bg_2()

Class Bg_2() ialah turunan dari kelas Bg untuk menampilkan tema 2.

4. Button()

Class Button() berisikan mengenai pengaturan seluruh tombol aktif yang ada di dalam game.

5. Button_play()

Class Button_play() berfungsi untuk memulai permainan.

6. Button_setting()

Class Button_setting() berfungsi untuk menghentikan permainan sementara.

7. Button_shop()

Class Button_shop() berfungsi untuk menampilkan menu shop atau pilih tema pada halaman utama.

8. Button_resume()

Class Button_resume() berfungsi untuk melanjutkan permainan pada saat permainan di pause.

9. Button_home()

Class Button_home() berfungsi untuk kembali ke halaman utama.

10. Button_buy()

Class Button_buy() berfungsi untuk membeli tema yang belum terbuka jika tema yang terkunci belum dibuka.

11. Skor()

Class Skor() berisikan mengenai logika perjalanan skor tokoh utama, pada kelas ini mengatur perolehan skor, reset skor, dan tampilan skor selama game berlangsung.

12. Player()

Class Player() berisikan mengenai logika tokoh utama, pada kelas ini mengatur tentang gambar tokoh utama yang di load ke dalam game, urutan keluarnya gambar, kemampuan tokoh utama untuk melompat, pengaturan tentang gravity setelah melompat dan kondisi suara jika tokoh utama mengakhiri game atau mendapatkan koin.

13. Kucing_1()

Class Kucing_1() berisikan logika animasi atau gerakan yang dilakukan player.

14. Koin()

Class Koin() berisikan mengenai logika koin yang digunakan di file main sebagai mata uang untuk mengakses tema yang terkunci, pada kelas ini mengatur tentang koin yang di load ke dalam game dan menampung jumlah koin yang didapatkan pengguna.

15. Obstacle()

Class Obstacle() berisikan mengenai logika pemrograman tentang penghalang tokoh utama yang terdapat di dalam game, pada kelas ini mengatur tentang apa saja batu yang di load ke dalam game, urutan kemunculan, jarak objek utama

terhadap penghalang, dan indikator penghentian game ketika objek utama menyentuh obstacle.

H. OBJEK

Berikut ialah objek - objek pada game “Koceng Loncat” :

1. Bg1
2. Bg2
3. Button_play
4. Button_shop
5. Button_setting
6. Button_resume
7. Button_home
8. Button_buy
9. obstacle_group
10. koin_group
11. player

I. ENKAPSULASI

Pada konsep OOP Enkapsulasi yang dipakai di game “Koceng Loncat” ini adalah pada perhitungan koin dengan skor agar tidak bisa di manipulasi. Implementasi enkapsulasi terdapat di kelas skor dan kelas koin yang masing - masing terdapat atribut `self.__koin__index` untuk kelas koin, dan `self.__current_skor`, `self.__score__surf` dan `self.__score__rect`.

J. PEWARISAN

Berikut ialah pewarisan yang terdapat pada game “Koceng Loncat” :

1. Kelas Bg1 dan Bg2 adalah warisan dari kelas Bg.
2. Kelas Kucing1 adalah warisan dari kelas Player.
3. Kelas Button_play, Button_setting, Button_shop, Button_resume, Button_home, Button_buy adalah warisan dari kelas Button.

K. POLIMORFISME

Polimorfisme yang dibuat dalam game ini dapat berfungsi sebagai pemanggil metode yang sama namun dengan objek yang berbeda.

Berikut ialah polimorfisme yang terdapat pada game “Koceng Loncat” :

1. Method `displaybg()` yang terdapat pada kelas Bg dan turunan kelas Bg

2. Method `action()` dan `button_display()` yang terdapat pada kelas `Button` dan turunan kelas `Button`
3. Method `masking()` dan `animation_state()` yang terdapat pada kelas `Player` dan turunan kelas `player`.

L. ABSTRAKSI

Implementasi Abstraksi terdapat pada file “**button.py dan player .py**” yaitu pada file “**button.py**” terletak pada method “**action()**” sebagai berikut :

Parent class

Class `Button()`

```
# parent class
You, 2 minutes ago | 1 author (You)
class Button(ABC):
    def __init__(self):
        self.button_click = pygame.mixer.Sound('asset/audio/button.mp3')

    @abstractmethod
    def button_display(self):
        pass

    @abstractmethod
    def action(self):
        pass
```

Child class

Class `Button_play()`, `Button_setting()`, `Button_shop()`, `Button_resume()`, `Button_home()`, `Button_buy()`.

```
def button_display(self):
    layar.blit(self.button, self.rect)
```

Class `Button_setting()`, `Button_shop()`, `Button_buy()`

```
def action(self, cond):
    return cond
```

Class `Button_play()`, `Button_resume()`, `Button_home()`

```
def action(self):  
    pass
```

Abstraksi pada file “**player.py**” :

Parent class

Class Player()

```
# Parrent class  
You, 3 weeks ago | 1 author (You)  
class Player(pygame.sprite.Sprite, ABC):  
    def __init__(self):  
        super().__init__()  
  
        self.gravity = 0  
        self.jump_sound = pygame.mixer.Sound('asset/audio/jump.mp3')  
        self.koin_get = pygame.mixer.Sound('asset/audio/get_coin.mp3')  
        self.game_over_sound = pygame.mixer.Sound('asset/audio/game_over.mp3')  
  
    @abstractmethod  
    def masking(self):  
        pass  
  
    def player_input(self):  
        keys = pygame.key.get_pressed()  
        if keys[pygame.K_SPACE] and self.rect.bottom >= 680:  
            self.gravity = -24  
            self.player_index = 0  
            self.jump_sound.play()  
            self.jump_sound.set_volume(0.15)  
  
    def apply_gravity(self):  
        self.gravity += 1  
        self.rect.y += self.gravity  
        if self.rect.bottom >= 680:  
            self.rect.bottom = 680  
  
    @abstractmethod  
    def animation_state(self):  
        pass
```

Child class

Class Kucing_1()

```
def masking(self):
    for masking1 in self.player_walk:
        pygame.mask.from_surface(masking1)
    for masking2 in self.player_jump:
        pygame.mask.from_surface(masking2)

def animation_state(self):
    if self.rect.bottom < 680:
        # jump
        self.player_index += 0.223
        if self.player_index >= len(self.player_jump):
            self.player_index = 0
        self.image = self.player_jump[int(self.player_index)]
    else:
        # walk
        self.player_index += 0.31
        if self.player_index >= len(self.player_walk):
            self.player_index = 0
        self.image = self.player_walk[int(self.player_index)]
```

M. PENANGANAN KESALAHAN

Implementasi Penanganan Kesalahan pada game “Koceng Loncat” terdapat pada file main, yang berisikan mengenai percobaan pembelian tema namun mata uang belum tercukupi.

```
if button_buy.rect.collidepoint(pygame.mouse.get_pos()):
    try :
        print(cn)
    except:
        print("coin tidak terdefinisi")
    else:
        if cn < button_buy.harga:
            buy = button_shop.action(False)
        else:
            buy = button_shop.action(True)
            kn = cn - button_buy.harga
            jsonedit(kn, sk, buy, thema)
    finally:
        print("succes")
```

N. GRAPHICAL USER INTERFACE

Graphical User Interface yang dipakai pada game “Koceng Loncat” ialah dari Pygame, karena pada Pygame ini memiliki fitur yang lebih lengkap dari Graphical User Interface untuk game lainnya.

O. KODE

Kode untuk melakukan import modul

```
# import modul
import pygame
import time
import json
from sys import exit
from random import randint
from obstacle import *
from player import *
from skor import Skor
from config import *
from koin import *
from button import *
from bg import *
```

Method ini berfungsi jika player bertabrakan dengan obstacle maka permainan akan selesai atau game over.

```
def collision_sprite(a):
    # jika player bertabrakan akan dijalankan
    if pygame.sprite.spritecollide(player.sprite, obstacle_group, False):
        player.sprite.game_over_sound_play() # play suara game_over_sound_play()
        obstacle_group.empty() # mengosongkan grup obstacle
        koin_group.empty() #mengosongkan grup koin
        bg_music.stop() # backsound akan dimatikan
        return False, 0
    elif a == 3 :
        return False, a
    else :
        return True, a
```

Method ini berfungsi jika player mengenai koin maka coin pada display permainan akan bertambah tetapi belum di record total koinnya.

```
# method player mengenai koin
def coin():
    if pygame.sprite.spritecollide(player.sprite, koin_group, True):
        player.sprite.get_koin_sound_play() # play get_koin_sound_play()
        return True
    else:
        return False
```

Method ini berfungsi untuk membaca file json yang berisi data utama game dan mengembalikan nilai dari data json yang ada.

```
# method untuk membaca data pada file json
def jsonread():
    with open(filename, "r") as f:
        temp = json.load(f)
        for data in temp:
            cn = data["coin"]
            sk = data["skor"]
            by = data["buy"]
            thm = data["thema"]
        return cn, sk, by, thm
```

Method ini berfungsi untuk merubah data json sesuai kondisi yang ada pada permainan.

```
# method untuk mengedit data file json
def jsonedit(cn, sk, by, thm):
    temp1 = {}
    with open(filename, "r") as f:
        temp = json.load(f)
        del temp[0]
    with open(filename, "w") as f:
        json.dump(temp, f, indent = 4)
    with open(filename, "r") as f:
        temp = json.load(f)

    temp1["coin"] = cn
    temp1["skor"] = sk
    temp1["buy"] = buy
    temp1["thema"] = thm
    temp.append(temp1)
    with open(filename, "w") as f:
        json.dump(temp, f, indent = 4)

    return cn, sk, by, thm
```

Deklarasi objek

```
# deklarasi objek
bg1 = Bg_1()
bg2 = Bg_2()

button_play = Button_play()
button_shop = Button_shop()
button_setting = Button_setting()
button_resume = Button_resume()
button_home = Button_home()
button_buy = Button_buy()

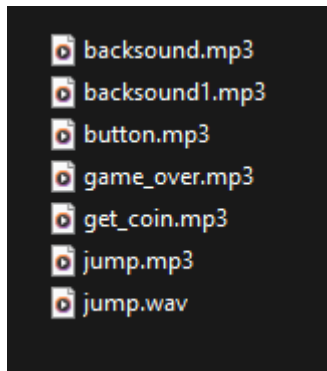
obstacle_group = pygame.sprite.Group()
koin_group = pygame.sprite.Group()

player = pygame.sprite.GroupSingle()
player.add(Kucing_1())
```


P. LAMPIRAN

Full Source Code : <https://github.com/Fadhil-Firoos/tubes>

Asset Audio :



Asset Font :

FunBlob.ttf	22/09/2021 13:12	TrueType font file	20 KB
GAMERIA.ttf	18/06/2020 03:18	TrueType font file	19 KB

Asset Gambar :

