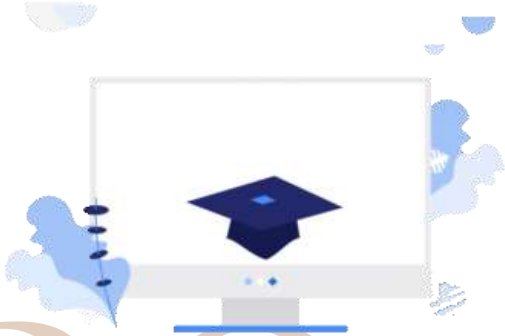
TUBES PBO



ITERA

rebah.an tech (RA-10)

KELOMPOK 10 PBO RA

ITERA

Anggota Kelompok

LEADER:

Ahmad Fadillah

120140092

Anisa Prasetya

120140087

Rahman Pajri

120140118

William Rusli

120140090

M. Fikri Damar

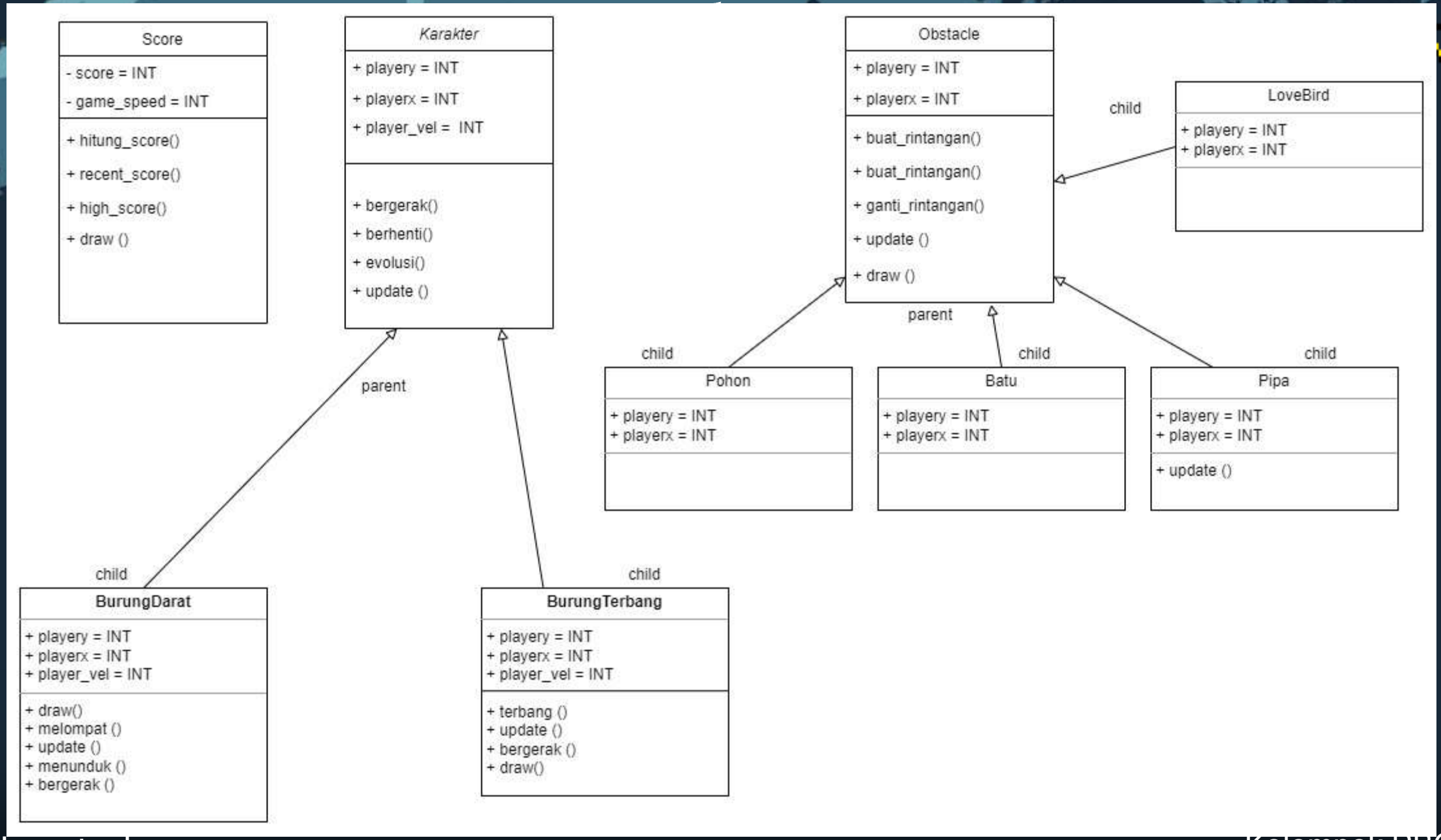
120140077

Alfian Kafilah

120140109

rebah.an tech

Kelompok PBO10-RA


Score
- score = INT
- game_speed = INT
+ hitung_score()
+ recent_score()
+ high_score()
+ draw ()
Karakter
+ playery = INT
+ playerx = INT
+ player_vel = INT
+ bergerak()
+ berhenti()
+ evolusi()
+ update ()
Obstacle
+ playery = INT
+ playerx = INT
+ buat_rintangan()
+ buat_rintangan()
+ ganti_rintangan()
+ update ()
+ draw ()
LoveBird
+ playery = INT
+ playerx = INT
child
parent
child
Pohon
+ playery = INT
+ playerx = INT
child
Batu
+ playery = INT
+ playerx = INT
child
Pipa
+ playery = INT
+ playerx = INT
+ update ()
parent
child
BurungDarat
+ playery = INT
+ playerx = INT
+ player_vel = INT
+ draw()
+ melompat ()
+ update ()
+ menunduk ()
+ bergerak ()
child
BurungTerbang
+ playery = INT
+ playerx = INT
+ player_vel = INT
+ terbang ()
+ update ()
+ bergerak ()
+ draw()

Implementasi Kelas

```
from abc import abstractmethod
import pygame
import random

pygame.init()

#Membuat screen, title, dan icon
width = 950
height = 836
screen = pygame.display.set_mode((width,height))
font = pygame.font.Font('freesansbold.ttf',20)
pygame.display.set_caption ("Morphling")
icon = pygame.image.load ('Gambarrr/pterodactyl.png')
pygame.display.set_icon(icon)
background = pygame.image.load('Gambarrr/background.png')
```

Implementasi Kelas

```python
class Karakter ():
    def __init__(self):
        self.playerx = 75
        self.playery = 570
        self.player_vel = 8
    def berhenti (self):
        pass
    def evolusi (self):
        pass
    @abstractmethod
    def update(self):
        pass
    @abstractmethod
    def bergerak (self):
        pass
```

```python
class BurungDarat (Karakter):
    def __init__(self):
        Karakter().__init__()
        self.gojox = self.playerx
        self.gojoy = self.playery
        self.gojo_vel = self.player_vel
        self.lompat = False
        self.nunduk = False
    def draw (self,screen):
        screen.blit(Gojo, (self.gojox, self.gojoy))
    def melompat (self,user_input):
        if (self.lompat is False and user_input[pygame.K_UP] ) or (self.lompat is False and user_input[pygame.K_SPACE]) :
            self.lompat = True
        if self.lompat is True:
            self.gojoy -= self.gojo_vel
            self.gojo_vel -= 0.1
            if self.gojo_vel < -self.player_vel:
                self.lompat = False
                self.gojo_vel = self.player_vel
    def menunduk (self,user_input):
        if (self.nunduk is False and user_input[pygame.K_DOWN] ) :
            self.nunduk = True
    def bergerak(self):
        pass
    def update (self, user_input):
        pass
```

Implementasi Kelas

```
class BurungTerbang(Karakter):
    def terbang (self):
        pass
    def update (self):
        pass
    def bergerak (self):
        pass
    def draw(self):
        pass
```

```
class Obstacle:
    def buat_rintangan(self, screen):
        pass
    def ganti_rintangan():
        pass
    def update():
        pass
    def draw():
        pass
```

```
class Pohon(Obstacle):
    pass
class Batu(Obstacle):
    pass
class LoveBird(Obstacle):
    pass
class Pipa(Obstacle):
    def update():
        pass
```

Implementasi Kelas

```python
class Score:
    def hitung_score():
        score = 0
        game_speed = 20
        score+=1
        if score % 100 == 0:
            game_speed +=1

        text = font.render("Score: " + str(score), True, (0, 0, 0))
        textRect = text.get_rect()
        textRect.center = (800, 100)
        screen.blit(text, textRect)

    def recent_score():
        pass
    def high_score():
        pass
```

```python
player1 = BurungDarat()
running = True
i = 0
obstacles = []
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

    screen.fill((255,255,255))
    screen.blit(background, (i,0))
    screen.blit(background, (width+i,0))
    if i == -width:
        screen.blit(background, (width+i,0))
        i = 0
    i -= 0.1

    player1.draw(screen)
    user_input = pygame.key.get_pressed()
    player1.melompat(user_input)

    Score.hitung_score()
    pygame.display.update()
```

Untuk lebih jelas dalam memahami source code tersebut, berikut kami cantumkan link github :
https://github.com/AhmadFadillah12/Rebah.an-tech.git

Kesimpulan

Dari design uml class dan potongan source code yang telah dipresentasikan mengenai game MORPHLING, kami memahami serta menerapkan 4 pilar utama dalam Pemrograman Berioerintasi Objek yaitu inheritance, abstraksi, enkapsulasi dan polymorphisme. Kelompok kami berusaha menerapkan konsep tersebut dengan harapan akan membuat programmer lain memahami 4 konsep utama Pemrograman Berorientasi Objek ketika memainkan game MORPHLING yang akan dibuat.

rebah.an tech

Kelompok PBO 10-RA


Terima
Kasih
Morphling