

**PERANCANGAN DAN IMPLEMENTASI GAME
TETRIS MENGGUNAKAN BAHASA PEMROGAMAN
PYTHON DAN PUSTAKA PYGAME**

Disusun untuk memenuhi tugas akhir mata kuliah Pemrograman Dasar

Dosen Pengampu: Dr. Atik Wintari, M.Kom.



Oleh:

Rendi Herlambang (25031554077)

Klemens Metami (25031554187)

Fakhri (25031554202)

**KELAS 2025D
PROGRAM STUDI PEMROGAMAN DASAR
FAKULTAS MATEMATIKA DAN ILMU
PENGETAHUAN ALAM
UNIVERSITAS NEGERI SURABAYA
2025**

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan proyek Pemrograman Dasar ini dengan baik dan tepat waktu. Laporan ini disusun sebagai bentuk pertanggungjawaban atas pelaksanaan proyek yang telah dilakukan selama proses pembelajaran pada mata kuliah Pemrograman Dasar.

Adapun proyek ini bertujuan untuk melatih kemampuan berpikir logis, sistematis, dan terstruktur dalam menyelesaikan permasalahan menggunakan konsep-konsep dasar pemrograman, seperti variabel, percabangan, perulangan, fungsi, serta penerapan algoritma sederhana. Melalui proyek ini, penulis diharapkan dapat memahami dan mengimplementasikan dasar-dasar pemrograman dengan lebih baik.

Penulis menyadari bahwa dalam penyusunan laporan ini masih terdapat berbagai kekurangan, baik dari segi isi maupun penyajiannya. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun demi penyempurnaan laporan ini di masa yang akan datang.

Akhir kata, penulis mengucapkan terima kasih kepada dosen pengampu mata kuliah Pemrograman Dasar yaitu Dr. Atik Wintari, M.Kom. serta semua pihak yang telah membantu dan memberikan dukungan dalam penyusunan laporan proyek ini. Semoga laporan ini dapat memberikan manfaat bagi pembaca.

Surabaya, 25 Desember 2025

Penyusun

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI	ii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah	1
1.3. Tujuan.....	2
BAB III PEMBAHASAN	3
2.1. Analisis Kebutuhan Aplikasi	3
2.2.1. Visual Studio Code (Python).....	3
2.1.2. Canva.....	5
2.1.3. Mixkit.....	6
2.2. Diagram Alur	6
2.3. Implementasi	6
2.3.1. Implementasi Antar Muka.....	6
2.3.2. Implementasi Logika Game.....	7
2.3.3. Implementasi Kontrol User.....	7
2.3.4. Implementasi Game Over.....	7
BAB III PENUTUP	9
3.1. Penjelasan Kode	9
3.2. Manual Penggunaan	10
3.3. SS APK.....	11

BAB I

PENDAHULUAN

1.1. Latar Belakang

Permainan Tetris diciptakan oleh Alexey Pajitnov pada tahun 1984 di Uni Soviet. Ia adalah seorang insinyur perangkat lunak Rusia yang bekerja di Pusat Komputasi Dorodnitsyn dari Akademi Ilmu Pengetahuan Soviet di Moskow. Nama tetris sendiri merupakan gabungan dari kata “Tetra” yang berarti empat dan “Tennis” olahraga favorit Pajitnov. Daya tarik global game Tetris berasal dari desainnya yang sederhana namun cerdas dan latihan mental yang diberikannya. Aturan permainannya yang mendasar mudah dipahami oleh siapa saja, terlepas dari bahasa atau latar belakang budaya, menjadikannya dapat dinikmati oleh semua kalangan usia. Di dalam tetris ini melibatkan beberapa logika dasar pemrograman seperti algoritma pergerakan, rotasi objek, pendeteksian tabrakan, dan lain-lain yang membuat game ini merupakan studi kasus yang sangat baik untuk memahami logika dasar pemrograman. Pada pengembangan tetris ini diimplementasikan menggunakan Bahasa python dan library pygame. Pygame adalah Pustaka python populer yang digunakan untuk mengembangkan video game. Pustaka ini menyediakan seperti fungsionalitas SDL (Simple DirectMedia Library), matematika vector, deteksi tabrakan, dukungan kamera dan MIDI, dll. Pustaka ini gratis dan open source. Untuk menjamin fungsionalitas permainan, proyek ini juga mengintegrasikan library pendukung seperti random untuk pengacakan balok, sys untuk control system, dan math untuk perhitungan geometris yang diperlukan. Melalui proyek ini, kelompok kami bertujuan untuk mengaplikasikan dan menguji pemahaman kami mengenai konsep dasar pemrograman, serta membangun aplikasi interaktif yang kompleks, dari perancangan logika hingga implementasi antarmuka pengguna berbasis grafis.

1.2. Rumusan Masalah

1. Bagaimana cara mengimplementasikan game loop Tetris menggunakan library Pygame untuk mengelola frame rate permainan secara stabil, menangani input pengguna, serta mengintegrasikan fungsi rendering dan rotasi yang berlaku untuk bentuk Polyomino (standar) maupun bentuk Polygon kustom (non-standar)?
2. Bagaimana cara merepresentasikan papan permainan Tetris sebagai struktur data grid dan mengembangkan algoritma yang tepat untuk mengatur pergerakan balok, mendeteksi tabrakan, serta mengimplementasikan logika penghapusan baris dan penentuan balok acak menggunakan library random?

3. Bagaimana cara mengimplementasikan game loop Tetris menggunakan library Pygame untuk mengelola frame rate permainan secara stabil, menangani input pengguna, serta mengintegrasikan fungsi rendering dan rotasi yang berlaku untuk bentuk Polyomino (standar) maupun bentuk Polygon kustom (non-standar)?

1.3. Tujuan

1. Mengimplementasikan struktur data grid (matriks) yang efisien untuk merepresentasikan papan permainan Tetris, serta mekanisme penyimpanan balok (freeze) pada grid tersebut.
2. Mengembangkan fungsi pergerakan dasar balok (kiri, kanan, dan jatuh) serta mekanisme rotasi, dengan mengintegrasikan pendeteksian tabrakan (collision detection) yang akurat terhadap batas papan dan balok lain yang sudah beku.
3. Menerapkan logika rendering dan pergerakan untuk bentuk polygon non-standar (seperti Jajargenjang, SegiLima, Trapesium) menggunakan perhitungan koordinat relatif dan absolut, memanfaatkan modul math untuk perhitungan geometris yang presisi (misalnya, dalam `get_absolute_vertices`).
4. Mendemonstrasikan penggunaan efektif library random untuk menciptakan variabilitas permainan, baik dalam penentuan jenis balok Tetromino yang muncul berikutnya (`_get_random_shape`) maupun dalam implementasi fitur tingkat kesulitan yang tidak terduga (misalnya, menentukan interval dan arah geseran pada fitur Earthquake).
5. Mengembangkan mekanisme yang menggeser balok yang sudah beku secara acak pada grid untuk meningkatkan tantangan.
6. Mengembangkan efek visual Glitch dan kontrol terbalik yang bersifat random (`draw_glitch_effect` dan `play`).
7. Merancang dan menguji fungsionalitas Math Challenge sebagai mekanisme game over atau penalti, yang terintegrasi dengan sistem Jumpscare untuk menunjukkan pemanfaatan kondisi program (`if game.state == "gameover"`) dan modul sys untuk mengontrol alur permainan (`sys.exit()`).

BAB III PEMBAHASAN

2.1. Analisis Kebutuhan Aplikasi

2.1.1. Visual Studio Code (Python)

VS Code dalam pemrograman ini berfungsi untuk mengedit, menulis program, menjalankan semua kode python. Program ini menggunakan beberapa modul pygame yaitu:

1) `pygame.init()`

Menginisialisasi seluruh modul Pygame (display, mixer, event, font). Fungsi dalam kodingan adalah sebelum layar victory ditampilkan, fungsi ini memastikan semua fitur Pygame aktif seperti:

- tampilan layar (`pygame.display`)
- event input mouse/keyboard (`pygame.event`)
- pemrosesan suara (`pygame.mixer`)
- rendering font (`pygame.font`)

Tanpa `pygame.init()`, hampir semua fungsi Pygame tidak bisa digunakan, dan program akan error.

2) `Pygame.mixer`

Mengatur audio, baik musik maupun efek suara. Contoh dalam kodingan

```
pygame.mixer.music.stop()  
play_sfx('clear')
```

Fungsi `pygame.mixer` dalam `victory_screen` kamu:

- Menghentikan background music ketika masuk layar kemenangan.
- Menjalankan efek suara kemenangan.

3) `Pygame.event`

Mengambil semua event (kejadian) yang terjadi di game:

- klik mouse
- gerakan mouse
- menekan tombol keyboard
- menutup jendela game

```

for event in pygame.event.get():
    if event.type == pygame.QUIT:
        pygame.quit()
        sys.exit()

    if event.type == pygame.MOUSEBUTTONDOWN:
        if btn.checkForInput(mouse_pos):
            main_menu()

```

Fungsi `pygame.event` dalam victory screen kamu:

- Menangani tombol X (close window).
- Mengecek apakah mouse diklik pada tombol MENU.

Tanpa event:

- Tombol tidak bisa diklik.
- Game tidak bisa ditutup secara normal.
- Layar victory “membeku”.

4) `pygame.display`

Mengatur jendela tampilan game dan melakukan update ke layar. Fungsi `pygame.display`:

- Menampilkan teks "VICTORY!"
- Menampilkan skor
- Menampilkan tombol MENU
- Refresh layar setiap frame

Tanpa `pygame.display`:

- Tidak ada perubahan tampilan yang muncul.
- Layar tidak akan memperbarui warna, teks, atau tombol.

5) `pygame.mouse`

Mengambil posisi kursor dan status tombol mouse. Penggunaan `pygame.mouse` dalam kodingan

```

mouse_pos = pygame.mouse.get_pos()
btn.changeColor(mouse_pos)

```

Fungsi `pygame.mouse`:

- Mengetahui posisi mouse di layar.
- Digunakan untuk efek hover (perubahan warna tombol).
- Digunakan untuk mendeteksi klik tombol MENU.

Tanpa `pygame.mouse`:

- Tombol tidak bisa tahu bahwa mouse berada di atasnya.
- Hover tidak berfungsi.
- Tidak bisa klik tombol.

6) `pygame.image`

- Mengambil file gambar dari folder aset
- Mengubahnya menjadi Surface yang bisa ditampilkan ke layar

7) `pygame.time`

Fungsinya untuk menghitung waktu sejak game dijalankan (dalam ms).

Dipakai untuk:

- interval gempa
- interval glitch
- timer soal matematika
- countdown untuk jumpscare

2.1.2. Canva

1) Membuat Background

- Background menu utama
- Background gameplay
- Background options
- Background difficulty selection

2) Membuat Tombol Menu (Button UI)

- Tombol PLAY
- Tombol OPTIONS
- Tombol QUIT
- Tombol DIFFICULTY

3) Membuat Logo

Contoh: “TETRIS GILA”

Dengan Canva:

- Gunakan teks 3D atau neon
- Tambahkan efek glow

4) Membuat Asset Informasi

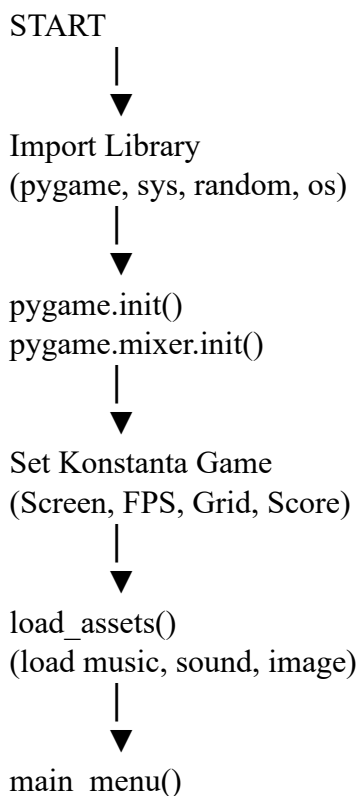
- Panel “Next Shape”
- Panel score
- Banner “Select Difficulty”
- Background untuk tulisan matematika challenge

2.1.3. Mixkit

Dalam implementasinya, file audio hasil unduhan dari Mixkit disimpan pada folder aset dan kemudian dimuat ke dalam program menggunakan modul `pygame.mixer`. Modul tersebut berfungsi untuk mengelola pemutaran audio selama permainan berlangsung. Pemanggilan suara dilakukan melalui fungsi khusus yang memastikan suara hanya diputar ketika pengaturan suara (*sound effect*) diaktifkan oleh pengguna.

Penggunaan Mixkit sebagai penyedia aset audio memberikan beberapa keuntungan, antara lain kemudahan akses, kualitas suara yang baik, serta kompatibilitas tinggi dengan library Pygame. Dengan adanya aset audio dari Mixkit, game *Tetris Gila* menjadi lebih interaktif, responsif, dan mampu memberikan suasana permainan yang lebih hidup dibandingkan tanpa penggunaan suara.

2.2. Diagram Alur



2.3. Implementasi

2.3.1. Implementasi Antar Muka

- Menu utama (Play, Options, Quit)
- Menu pemilihan tingkat kesulitan (Easy, Medium, Hard)
- Tampilan papan permainan (grid)

- Tampilan skor dan level
- Layar Game Over dan tombol Play Again

Diimplementasikan menggunakan:

- `pygame.display`
- Class `Button`
- Rendering font dan gambar

2.3.2. Implementasi Logika Game

- Pembuatan papan permainan 20×10
- Generasi balok Tetris secara acak
- Pergerakan balok ke bawah, kiri, kanan
- Rotasi balok
- Deteksi tabrakan antar balok dan dinding
- Penyimpanan balok ke dalam grid (freeze)
- Penghapusan baris penuh (clear line)
- Perhitungan skor

Diimplementasikan menggunakan:

- Class `Tetris`
- Class `Figure`
- Metode `intersects()`, `freeze()`, `break_lines()`

2.3.3. Implementasi Kontrol User

- Keyboard input untuk mengendalikan balok:
 - `↑` : rotasi
 - `← →` : geser
 - `↓` : turun cepat
 - `SPACE` : jatuh instan
- Mouse input untuk navigasi menu

Diimplementasikan menggunakan:

- `pygame.event`
- Event `KEYDOWN`, `KEYUP`, `MOUSEBUTTONDOWN`

2.3.4. Implementasi Game Over

- Game berakhir saat balok baru langsung bertabrakan
- Menampilkan teks Game Over

- Memberikan opsi restart atau kembali ke menu

Diimplementasikan dengan:

- Variabel `state`
- Tombol `Play Again`

BAB III PENUTUP

3.1. Penjelasan Kode

Pada bagian awal dilakukan inisialisasi untuk seluruh fitur, grafis, dan input yang ada di dalam program agar bisa diaktifkan dengan Pygame.

```
pygame.init()
SCREEN = pygame.display.set_mode((1280, 720))
pygame.display.set_caption("Menu & Tetris")
```

Pada bagian class button digunakan untuk mengatur tombol interaktif yang ada di game dan di main menu.

```
class Button():
    def update(self, screen):
        screen.blit(self.image, self.rect)
        screen.blit(self.text, self.text_rect)
```

Bagian class figure digunakan untuk mempresentasikan bentuk blok tetris

```
class Figure:
    figures = [
        [[1, 2, 3, 4]],
        [[4, 5, 5, 4]],
        ...
    ]
```

Pada bagian class Tetris digunakan sebagai inti logika dari game ini.

```
self.field = []
for i in range(height):
    self.field.append([0] * width)
```

Membuat grid tetris

```
def new_figure(self):
    self.figure = Figure(3, 0)
```

Generate blok baru

```
def intersects(self):
    if self.field[i + self.figure.y][j + self.figure.x] > 0:
        return True
```

Mendeteksi blok yang bertabrakan

```
def go_down(self):
    self.figure.y += 1
```

Pergerakan blok

```
def freeze(self):
    self.field[i + self.figure.y][j + self.figure.x] = self.figure.color
```

Freeze balok

```
def break_lines(self):
    if zeros == 0:
        lines += 1
```

Menghapus blok yang sudah penuh

3.2. Manual Penggunaan

Saat permainan dimulai, balok akan jatuh secara otomatis dari atas layar dan dapat dikendalikan menggunakan keyboard. Tombol panah kiri dan kanan digunakan untuk menggeser balok, panah atas untuk memutar balok, panah

bawah untuk mempercepat jatuh, dan tombol spasi untuk menjatuhkan balok secara langsung. Tujuan permainan adalah menyusun balok agar membentuk satu baris penuh secara horizontal sehingga baris tersebut akan menghilang dan skor bertambah.

Permainan akan berakhir (Game Over) apabila balok baru tidak dapat muncul karena papan permainan sudah penuh. Pada kondisi ini, pemain dapat memilih untuk mengulangi permainan dengan tombol Play Again atau kembali ke menu utama. Pemain juga dapat kembali ke menu utama kapan saja dengan menekan tombol Menu atau keluar dari game dengan memilih Quit.

3.3. SS APK

