

Penerapan Operasi Penjumlahan Vektor dalam Sistem Pergerakan Objek pada Game Snake

Fauzan Mohamad Abdul Ghani 13524113^{1,2}

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

13524113@std.stei.itb.ac.id, fauzan210mohamad@gmail.com

Abstract—Perkembangan pembuatan game sangat bergantung pada logika matematika, khususnya Aljabar Linear sebagai pondasi dalam memanipulasi objek grafis. Salah satu implementasinya adalah mekanisme pergerakan objek dalam ruang dua dimensi (2D). Makalah ini membahas penerapan operasi vektor sebagai algoritma dalam pergerakan karakter dari game klasik *Snake*. Metode yang digunakan adalah merepresentasikan area permainan sebagai sistem koordinat Kartesius, di mana posisi kepala dan setiap segmen tubuh ular didefinisikan sebagai vektor posisi (x, y) . Pergerakan karakter disimulasikan melalui operasi translasi, yaitu penjumlahan antara vektor posisi saat ini dengan vektor arah yang ditentukan oleh input pengguna. Selain itu, makalah ini juga membahas logika pergeseran vektor untuk menangani pergerakan segmen tubuh ular yang mengikuti jejak kepala.

Keywords—Aljabar Linear, Penjumlahan Vektor, Snake Game.

I. PENDAHULUAN

Industri perkembangan video game telah mengalami evolusi yang pesat, dari sekadar hiburan menjadi sistem perangkat lunak yang kompleks. Namun, terlepas dari kemajuan grafis dan teknologi *engine* yang canggih, fondasi utama dari setiap pergerakan karakter atau simulasi visual adalah matematika, khususnya di bidang Aljabar Linear. Konsep-konsep seperti vektor matriks bukan hanya sebagai teori abstrak, tetapi juga sebagai alat yang digunakan untuk pergerakan objek dalam ruang dua dimensi atau tiga dimensi.

Untuk mengkaji penerapan konsep ini secara mendalam, permainan *Snake* dipilih sebagai objek studi kasus. *Snake* merupakan permainan di mana pemain harus membuat objek ular memakan suatu *item* yang direpresentasikan sebagai buah dan akan membuatnya bertambah besar sehingga memenuhi *map* atau area permainan. *Snake* merupakan representasi ideal dari sistem pergerakan berbasis *grid*. Mekanisme permainan ini sangat bergantung kepada logika geometri, di mana area permainan dipetakan sebagai sistem koordinat

kartesius, dengan posisi kepala ular dan setiap bagian tubuhnya direpresentasikan sebagai vektor posisi (x, y) . Tantangan teknis dari permainan ini adalah pada bagaimana mengelola vektor posisi agar bergerak secara sinkron sesuai dengan input dari pemain.

Pergerakan objek adalah mekanisme dasar yang harus dipahami oleh setiap pengembang game. Pergerakan karakter terlihat sebagai aktivitas yang sederhana bagi pengguna. Namun, pergerakan tersebut merupakan proses manipulasi data koordinat yang terjadi secara berulang dengan sangat cepat (*frame per second*). Tanpa pemahaman matematika yang tepat, pengelolaan koordinat ini sering kali dilakukan dengan percabangan yang tidak efisien yang dapat menyebabkan kode menjadi rumit dan sulit dikembangkan. Oleh karena itu, penerapan operasi vektor menjadi solusi standar untuk menangani perubahan posisi dengan cara yang efisien secara komputasi.

Permasalahan utama yang dibahas dalam makalah ini adalah bagaimana menerjemahkan input arah dari pemain menjadi perubahan posisi karakter menggunakan operasi penjumlahan vektor. Pergerakan lurus pada bidang datar adalah translasi, yang secara aljabar dapat dinyatakan sebagai penjumlahan antara vektor posisi awal dengan vektor arah. Pendekatan ini mengandalkan pembaruan koordinat sumbu x dan y secara terpisah, menjadi satu operasi vektor tunggal yang lebih ringkas. Selain itu, pergerakan tubuh ular yang harus mengikuti jejak posisi kepala memerlukan algoritma pergeseran nilai vektor dalam struktur data.

II. LANDASAN TEORI

A. Definisi Vektor dan Skalar

Dalam memodelkan objek fisik di dunia nyata maupun simulasi computer, kuantitas fisik dibedakan menjadi dua jenis, yaitu vektor dan skalar. Skalar didefinisikan sebagai nilai numerik yang menyatakan besaran kuantitas fisik saja, seperti panjang atau temperature. Sementara vektor adalah kuantitas fisik yang memiliki dua atribut, yaitu besaran dan arah.

Sebuah vektor v dapat direpresentasikan sebagai pasangan terurut bilangan real

$$v = (v1, v2)$$

Di mana $v1$ dan $v2$ adalah komponen vektor yang menentukan proyeksi garis tersebut terhadap sumbu koordinat.

B. Ruang Vektor dan Basis Standar

Permainan *Snake* beroperasi dalam ruang dua dimensi yang secara matematis merupakan himpunan vektor-vektor dengan dua komponen. Agar vektor dapat didefinisikan secara konsisten, diperlukan sebuah sistem koordinat yang dibangun oleh basis.

Sebuah himpunan vektor $S = \{v1, v2, \dots, vn\}$ dinamakan basis jika S bebas linear dan membangun ruang vektor tersebut. Dalam ruang vektor R^2 (dua dimensi), digunakan basis standar yang terdiri atas vektor satuan:

$$i = (1,0) \text{ dan } j = (0,1)$$

Dalam implementasi game, basis ini merepresentasikan arah pergerakan dasar :

- Vektor $i = (1,0)$ merepresentasikan pergerakan satu satuan ke arah sumbu x positif
- Vektor $j = (0,1)$ merepresentasikan pergerakan satu satuan ke arah sumbu y positif

Setiap posisi objek dalam game pada dasarnya adalah kombinasi linear dari basis basis $v = x(i) + y(j)$.

C. Operasi Penjumlahan Vektor

Mekanisme inti dari pergerakan objek dalam makalah ini didasarkan pada operasi penjumlahan vektor. Jika terdapat dua vektor u dan v di ruang R^2 , maka penjumlahan keduanya dilakukan dengan menjumlahkan komponen-komponen yang bersesuaian :

$$u + v = (u1 + v1, u2 + v2)$$

Jika titik awal vektor v diletakkan pada titik ujung vektor u , maka hasil $u + v$ adalah vektor yang menghubungkan titik awal u ke titik ujung v . Prinsip ini yang akan digunakan untuk memindahkan posisi kepala ular dari titik lama ke titik baru berdasarkan vektor arahnya.

D. Translasi

Perpindahan posisi objek tanpa rotasi dikenal sebagai translasi. Translasi adalah pemetaan yang menggeser setiap titik (x, y) sejauh vektor translasi (tx, ty) , di mana rumus translasi dari titik (x, y) menjadi (x', y') adalah:

$$x' = x + tx$$

$$y' = y + ty$$

Dalam konteks game *Snake*, (x, y) adalah koordinat kepala ular saat ini, sedangkan (tx, ty) adalah vektor kecepatan yang bernilai 1 atau -1 yang bergantung dari input pemain. Operasi ini dilakukan secara berulang pada setiap *frame* permainan untuk menciptakan ilusi pergerakan yang mulus.

III. ANALISIS DAN PERANCANGAN SISTEM

A. Metodologi Pengembangan Sistem

Dalam membangun simulasi permainan *Snake* berbasis vektor, diterapkan metodologi pengembangan perangkat lunak yang disesuaikan dengan pendekatan studi eksperimental matematika yang dibagi menjadi empat fase utama, yaitu :

1. Fase Analisis Teoretis:

- Mengkaji literatur mengenai operasi Aljabar Linear pada ruang Euclidean R^2 .
- Memetakan rumus transformasi geometri, yaitu translasi ke dalam logika pemrograman.

2. Fase Pemodelan Matematis:

- Merumuskan objek permainan, yaitu ular dan buah(*item*) sebagai entitas vektor.
- Menetapkan aturan sistem koordinat layer yang berbeda dengan koordinat kartesius standar.

3. Fase Perancangan Algoritma:

- Menyusun diagram alir (*flowchart*) untuk siklus permainan.
- Merancang algoritma penanganan batas layer menggunakan algoritma modular.

4. Fase Implementasi dan Pengujian:

- Penerjemahan desain ke dalam kode program Python.
- Verifikasi kesesuaian antara hasil perhitungan manual vektor dengan pergerakan objek di layar.

B. Analisis Kebutuhan Sistem

Untuk memastikan sistem dapat berjalan dengan lancar dan operasi vektor dapat divisualisasikan secara *real-time*, didefinisikan kebutuhan sistem sebagai berikut:

1. Kebutuhan Fungsional

- Representasi Vektor: Sistem harus menyimpan posisi setiap segmen tubuh ular sebagai objek vektor (x, y) .

- Kontrol Translasi: Sistem harus mengubah input arah dari pengguna (atas, bawah, kiri, kanan) menjadi vektor arah satuan.
- Visualisasi Grid: Area permainan harus dibagi menjadi petak petak diskrit berukuran $N * M$.
- Objek yang keluar dari satu sisi layar harus muncul kembali di sisi yang berlawanan.

2. Kebutuhan Non-Fungsional

- Responsivitas: Jeda waktu antara penekanan tombol dan perubahan arah vektor tidak boleh terlalu lama.
- Simulasi berjalan konstan pada kecepatan yang ditentukan untuk menjaga konsistensi perhitungan.

C. Analisis Model Matematis

1. Sistem Koordinat Layar

Berbeda dengan kuadran I pada sistem Kartesius konvensional di mana Y positif mengarah ke atas, sistem grafika computer menempatkan titik asal (0, 0) pada pojok kiri atas layar.

- Sumbu X: Positif ke arah kanan.
- Sumbu Y: Positif ke arah bawah.
- Vektor arah atas bernilai (0, -1), bukan (0, 1)

2. Definisi Objek Vektor

Setiap entitas dinamis dalam permainan didefinisikan sebagai himpunan vektor:

- Vektor Kepala (H): Sebuah vektor posisi tunggal $H \in \mathbb{R}^2$ yang merepresentasikan lokasi kepala ular.
- Vektor Tubuh (B): Himpunan terurut (Ordered Set) dari vektor-vektor posisi segmen tubuh. $B = \{b_0, b_1, \dots, b_n\}$ di mana $b_0 = H$ (Kepala), dan b_n adalah ekor.
- Vektor arah: Vektor satuan yang menentukan arah translasi pada *frame* berikutnya. Terdapat 4 basis arah: Kanan = (1, 0), kiri = (-1, 0), bawah = (0, 1), atas = (0, -1)

3. Operasi Translasi dan Aritmatika Modular

Pergerakan ular bukan merupakan animasi *keyframe*, melainkan hasil operasi matematika

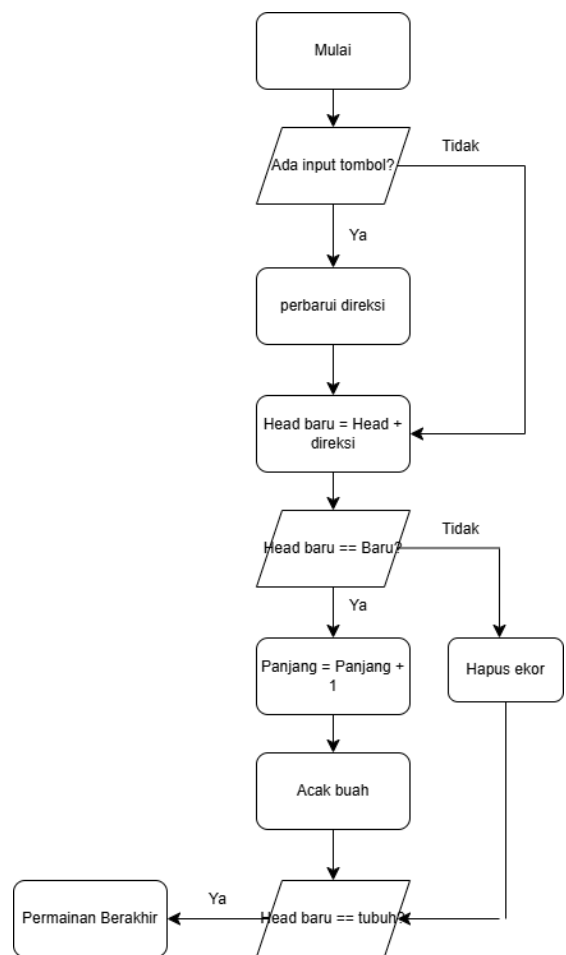
yang dihitung setiap siklus waktu t .

Rumus Translasi: Posisi baru kepala adalah hasil penjumlahan vektor posisi lama dengan vektor arah.

Rumus Topologi Torus (Tembus Dinding): Untuk menangani batas layar dengan lebar W dan tinggi H , diterapkan operasi modulo (%). Operasi ini memastikan koordinat selalu berada dalam rentang $[0, W - 1]$ dan $[0, H - 1]$.

D. Perancangan Algoritma dan Alur Kerja

Logika permainan dirancang menggunakan diagram alir sistem untuk menggambarkan bagaimana data diproses dari input hingga berakhir.



E. Perancangan Struktur Data

Efisiensi memori sangat bergantung pada pemilihan struktur data. Dalam perancangan ini, digunakan struktur data dinamis.

1. Tipe Data Vektor (Vector2): Menggunakan objek yang memiliki atribut x (float) dan y (float)
2. Tubuh ular disimpan dalam struktur data List. operasi penambahan elemen di depan dan penghapusan di belakang harus memiliki kompleksitas waktu yang rendah agar game tidak lambat saat ular menjadi sangat panjang.

F. Perancangan Arsitektur Perangkat Lunak

Sistem ini dibangun dengan paradigma Pemrograman Berorientasi Objek. Berikut adalah desain kelas yang direncanakan:

Rincian Kelas:

1. Kelas Snake:
 - Atribut: body (List[Vector2]), direction (Vector2), add_segment (Boolean).
 - Metode: move(), reset(), draw().
2. Kelas Fruit:
 - Atribut: pos (Vector2), color (Tuple).
 - Metode: randomize().
3. Kelas Main:
 - Mengelola game loop, timer, dan pemanggilan metode update serta draw.

G. Perancangan Antarmuka (User Interface)

Antarmuka permainan dirancang minimalis untuk memfokuskan pengamatan pada pergerakan vektor objek.

- Area Permainan: Grid berwarna kontras untuk memudahkan melihat posisi koordinat.
- Objek Ular: Dirender sebagai kotak kotak hijau yang tersusun mengikuti alur vektor.
- Objek Buah: Dirender dengan warna merah.

IV. IMPLEMENTASI DAN PEMBAHASAN

Sistem permainan *Snake* berbasis vektor ini diimplementasikan menggunakan Bahasa pemrograman Python dengan bantuan Pustaka Pygame. Pustaka ini dipilih karena menyediakan modul `pygame.math.Vector2` yang mendukung operasi aljabar linear secara natif, sehingga mempermudah penerapan rumus translasi yang telah dirancang.

A. Implementasi Kode Program

Pada tahap ini, rancangan algoritma dan diagram alir yang dibuat pada Bab 3 diterjemahkan ke dalam kode program.

1. Inisialisasi Objek Vektor

Posisi kepala dan tubuh ular tidak disimpan sebagai variabel integer terpisah, melainkan sebagai objek vektor dua dimensi yang memungkinkan manipulasi posisi dilakukan sebagai satu kesatuan entitas matematika.

```
class Snake:
    def __init__(self):
        #posisi awal ular
        self.body = [Vector2(5, 10), Vector2(4, 10), Vector2(3, 10)]

        #Arah pergerakan awal ular ke arah kanan
        self.direction = Vector2(1, 0)

        #Menentukan status perut ular
        self.new_block = False
```

Gambar 1. Inisialisasi Ular
(Sumber : dokumentasi pribadi)

2. Implementasi Operasi Translasi

Fungsi utama pergerakan menerapkan rumus $\text{Posisi baru} = \text{Posisi lama} + \text{vektor arah}$. Dalam Python, operasi ini memanfaatkan operator `+` pada objek `Vector2`

```
def move_snake(self):
    if self.new_block == True:
        #jika makan buah salin seluruh elemen
        body_copy = self.body[:]
    else:
        #jika tidak, salin seluruh tubuh kecuali elemen terakhir
        body_copy = self.body[:-1]

    # Hitung posisi baru (Translasi Biasa)
    new_head = body_copy[0] + self.direction

    # IMPLEMENTASI MODULO (Agar Tembus Dinding)
    new_head.x = new_head.x % CELL_NUMBER
    new_head.y = new_head.y % CELL_NUMBER

    #Menyisipkan vektor kepala baru ke depan list
    body_copy.insert(0, new_head)

    #Memperbarui atribut utama self.body dengan list baru
    self.body = body_copy[:]

    #reset flag newblock
    if self.new_block:
        self.new_block = False
```

Gambar 2. Pergerakan Ular
(Sumber : dokumentasi pribadi)

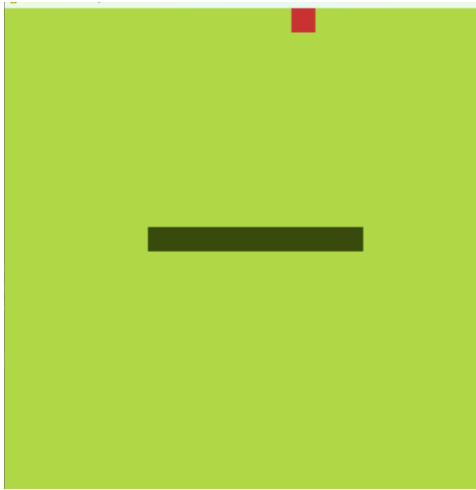
B. Hasil Pengujian Sistem

Pengujian dilakukan untuk memverifikasi apakah perilaku objek dalam permainan sesuai dengan logika vektor yang diharapkan. Pengujian dibagi menjadi tiga skenario

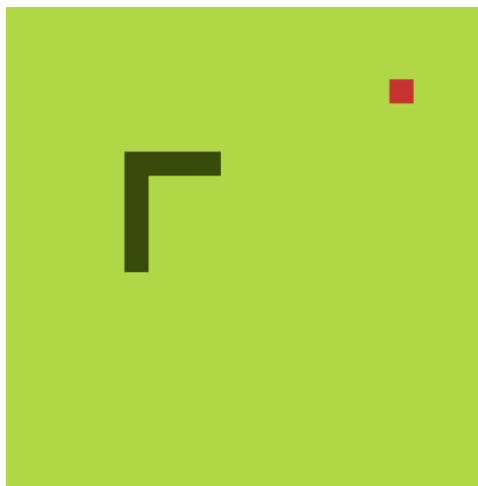
1. Pengujian Translasi Dasar

Skenario ini menguji respons pergerakan ular terhadap input tombol arah.

- Input : Tombol Kanan (->).
- Vektor Arah: (1,0).
- Hasil: Kepala ular berpindah dari sel (x, y) ke (x+1, y) pada setiap *frame*.



Gambar 3. Tampilan Permainan saat Ular Bergerak
Lurus
(Sumber : dokumentasi pribadi)

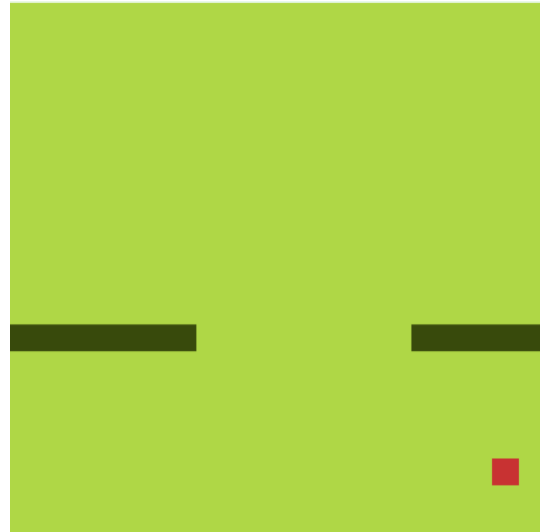


Gambar 4. Tampilan Permainan saat Ular Berbelok
(Sumber : dokumentasi pribadi)

2. Pengujian Mekanisme Wrapping

Skenario ini menguji apakah ular dapat menembus batas layar dan muncul di sisi sebaliknya.

- Kondisi: Kepala ular bergerak ke kiri melewati koordinat $x = 0$.
- Hasil: Kepala ular muncul seketika di sisi kanan layar.



Gambar 5. Visualisasi Ular menembus dinding kiri dan muncul di dinding kanan
(Sumber : dokumentasi pribadi)

3. Pengujian Interaksi Objek (Tabrakan)

Skenario ini menguji deteksi kesamaan posisi vektor antara kepala ular dengan tubuhnya sendiri.

- Hasil: Ketika vektor kepala memiliki nilai (x, y) yang sama persis dengan salah satu vektor tubuh, permainan berhenti dan mereset posisi ular ke awal.

C. Pembahasan dan Analisis Validitas Vektor

Hasil pengujian memperlihatkan bahwa luaran program identik dengan perhitungan manual. Hal ini membuktikan bahwa implementasi kelas `Vector2` dan operasi penjumlahan pada kode program telah berjalan akurat. Selain itu, penggunaan vektor terbukti menyederhanakan kode karena tidak memerlukan struktur percabangan yang rumit untuk setiap sumbu.

V. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem permainan *Snake* yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

1. Efektivitas Operasi Vektor dalam Simulasi Pergerakan Penerapan konsep Aljabar Linear, khususnya pada operasi penjumlahan vektor di ruang Euclidean R^2 , terbukti berhasil menggantikan logika pergerakan konvensional. Penggunaan objek vektor `Vector2` menyederhanakan manipulasi koordinat (x, y) menjadi satu entitas tunggal, sehingga kode program menjadi lebih ringkas dan terstruktur.
2. Validitas Transformasi Geometri Logika pergerakan kepala ular telah dibuktikan secara matematis mengikuti prinsip Translasi Geometri.

Rumus Posisi baru = Posisi lama + vektor arah menghasilkan perpindahan posisi yang akurat dan konsisten pada setiap *frame* permainan, tanpa memerlukan perhitungan manual pada masing-masing sumbu secara terpisah.

3. Efisiensi Penanganan Topologi Ruang dengan Aritmatika Modular Penggunaan operasi Modulo terbukti efektif untuk menciptakan topologi ruang permainan yang bersifat tembus dinding.
4. Optimasi Struktur Data Tubuh Ular Penerapan struktur data antrean dengan logika “tambah kepala, hapus ekor” terbukti efisien dalam mensimulasikan pergerakan benda panjang.

Fauzan Mohamad Abdul Ghani 13524113

VI. UCAPAN TERIMA KASIH

Puji Syukur serta rasa terima kasih selaku penulis kepada Allah SWT atas segala berkat, rahmat, dan karunia-Nya sehingga penulis dapat menyelesaikan makalah ini dengan lancar. Selain itu penulis juga berterima kasih kepada :

1. Orang tua penulis, yang selalu mendukung, dan mendoakan penulis.
2. Bapak Arrival Dwi Sentosa, S.Kom.,MT., atas bimbingannya mengajar mata kuliah Aljabar Linear Geometri di kelas K03
3. Bapak Dr. Ir. Rinaldi Munir, MT., yang telah memberikan sumber materi pembelajaran.

DAFTAR PUSTAKA

- [1] R. Munir, “Vektor di Ruang Euclidean (Bagian 1),” diakses pada 24 Desember 2025
<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-11-Vektor-di-Ruang-Euclidean-Bag1-2025.pdf>
- [2] R. Munir, “Ruang Vektor Umum (Bagian 2),” diakses pada 24 Desember 2025
<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-16-Ruang-vektor-umum-Bagian2-2025.pdf>
- [3] R. Munir, “Ruang Vektor Umum (Bagian 4),” diakses pada 24 Desember 2025
<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-18-Ruang-vektor-umum-Bagian4-2025.pdf>
- [4] J. F. Hughes, “Computer Graphics: Principles and Practice (3rd Edition),” 2014, diakses pada 24 Desember 2025
<https://cgpp.net/>
- [5] R. Nystrom, “Game Programming Patterns,” 2014, diakses pada 24 Desember 2025
<https://gameprogrammingpatterns.com/game-loop.html>
- [6] A. Sweigart, “Making Games with Python & Pygame,” 2012, diakses pada 24 Desember 2025
<https://inventwithpython.com/pygame/v>

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 24 Desember 2025