

ANALISIS IMPLEMENTASI GAME MOBILE 2D BERBASIS JAVA MENGGUNAKAN EVENT DRIVEN PROGRAMMING STUDI KASUS FLAPPY BIRD

Agil Prianda¹, Ahmad Farhan Zakia², Ilham Hasby³, Muhamad Alda⁴

^{1,2,3,4} Prodi Sistem Informasi, Universitas Islam Negeri Sumatera Utara, Kampus IV Jalan Lapangan Golf, Desa Durian Jangak, Kecamatan Pancur Batu, Kabupaten Deli Serdang, Sumatera Utara, 20353.

Keywords:

Game 2D, Java, Mobile
Game Simulation

Correspondent Email:

agilprianda31@gmail.com

Abstrak. Perkembangan industri game mobile menuntut efisiensi dalam pemrosesan logika permainan dan responsivitas interaksi pengguna. Penelitian ini bertujuan untuk menganalisis implementasi sistem event-driven programming pada pengembangan game 2D Flappy Bird menggunakan bahasa pemrograman Java. Pentingnya topik ini terletak pada bagaimana paradigma pemrograman berbasis kejadian dapat mengoptimalkan penggunaan resource melalui pengelolaan input real-time dan mekanisme game loop yang stabil. Metode penelitian dilakukan melalui pengembangan sistem menggunakan pustaka Java Swing dan AWT, dengan menerapkan ActionListener untuk pengaturan timer serta KeyListener untuk menangani input melompat dari pengguna. Hasil penelitian menunjukkan bahwa penggunaan event-driven programming memungkinkan sinkronisasi yang presisi antara pembaruan logika gravitasi bird dan pergerakan pipa secara konstan pada frame rate 60 FPS. Pengujian sistem membuktikan bahwa mekanisme deteksi tabrakan (collision detection) dan pembaruan skor berjalan akurat tanpa adanya lag yang signifikan. Kesimpulannya, implementasi paradigma event-driven dalam Java sangat efektif untuk membangun game 2D sederhana yang membutuhkan responsivitas tinggi, memberikan dasar yang kuat untuk pengembangan aplikasi mobile berbasis desktop atau sistem embedded yang lebih kompleks.



Copyright © [JITET](http://www.jitet.org) (Jurnal Informatika dan Teknik Elektro Terapan). This article is an open access article distributed under terms and conditions of the Creative Commons Attribution (CC BY NC)

Abstract. The continuous growth of the mobile gaming industry demands efficiency in game logic processing and user interaction responsiveness. This study aims to analyze the implementation of an event-driven programming system in the development of a 2D Flappy Bird game using the Java programming language. The significance of this topic lies in how the event-based programming paradigm can optimize resource utilization through real-time input management and a stable game loop mechanism. The research method involves system development using the Java Swing and AWT libraries, implementing ActionListener for timer management and KeyListener to handle user jump inputs. The results demonstrate that the use of event-driven programming enables precise synchronization between the bird's gravity logic updates and constant pipe movement (velocity) at a frame rate of 60 FPS. System testing proves that the collision detection mechanism and score updates function accurately without significant lag. In conclusion, the implementation of the event-driven paradigm in Java is highly effective for building simple 2D games that require high responsiveness, providing a solid foundation for developing more complex desktop-based or mobile-simulated applications.

1. PENDAHULUAN

Perkembangan game mobile terus mengalami peningkatan seiring dengan kemajuan perangkat seluler dan meningkatnya kebutuhan hiburan digital. Game dua dimensi (2D) masih menjadi pilihan populer karena memiliki kebutuhan sumber daya yang relatif rendah serta kemudahan dalam pengembangan dibandingkan game tiga dimensi [1], [2].

Salah satu aspek penting dari dalam pengembangan game 2D adalah penerapan teknik pemrograman yang mampu menangani berbagai peristiwa secara *real-time*. *Event Driven Programming* menjadi pendekatan yang umum digunakan dalam game mobile karena kemampuannya merespons input pengguna, perubahan status permainan, serta interaksi objek secara dinamis [3], [4]. Pendekatan ini terbukti sangat efektif dalam mengelola alur permainan, meningkatkan responsivitas sistem, dan menjaga performa aplikasi pada perangkat dengan keterbatasan sumber daya [2], [5].

Dalam implementasi game 2D, deteksi tabrakan (*collision detection*) merupakan komponen krusial yang dapat memengaruhi pengalaman bermain yang secara langsung. Kesalahan dalam perhitungan tabrakan dapat menyebabkan inkonsistensi logika permainan dan menurunkan kualitas game secara keseluruhan [1]. Oleh karena itu, pemilihan metode collision detection yang tepat menjadi faktor penting dalam pengembangan game casual seperti Flappy Bird dan variasinya [6], [7].

Selain aspek teknis, desain game juga memegang peranan penting dalam menentukan kualitas dan daya tarik permainan. Penggunaan *Game Design Document* (GDD) terbukti membantu pengembang dalam merancang fitur, mekanisme, serta alur permainan secara sistematis [3], [6]. Model desain yang baik akan mendukung implementasi logika permainan dan mempermudah proses evaluasi serta pengembangan lanjutan [2].

Evaluasi kualitas game mobile juga menjadi pusat perhatian dalam penelitian-penelitian sebelumnya. Beberapa pendekatan evaluasi berbasis data mining dan pemodelan fitur game telah digunakan untuk menilai pengalaman pengguna dan efektivitas desain permainan [4]. Evaluasi ini penting untuk memastikan bahwa game tidak hanya berjalan dengan baik secara teknis, tetapi juga dapat mampu memberikan

pengalaman bermain yang optimal dan menarik bagi pengguna [8], [9].

2. TINJAUAN PUSTAKA

Pada bagian ini akan menjelaskan beberapa teori yang menjadikan pondasi dan landasan pada jurnal ini, yang meliputi *Game Mobile 2D*, *Event Driven Programming*, *Collision detection*, *Evaluasi dan Kualitas Game Mobile*.

2.1 Game Mobile 2D

Game mobile 2D merupakan permainan digital yang menyajikan elemen visual dalam dua dimensi dan banyak digunakan pada platform dengan keterbatasan perangkat keras [1]. Pendekatan ini memungkinkan efisiensi pemrosesan sekaligus mempertahankan kualitas interaksi pengguna [10]. Struktur game 2D juga relatif lebih sederhana, sehingga cocok digunakan sebagai media pembelajaran maupun hiburan kasual [11], [12], [13].

2.2 Event Driven Programming

Event Driven Programming adalah paradigma pemrograman yang berfokus pada penanganan peristiwa seperti input pengguna, perubahan status objek, dan interaksi lingkungan. Dalam game mobile, pendekatan ini memungkinkan sistem merespons aksi pemain secara *real-time* dan menjaga kelancaran alur permainan [3], [4], [5], [8].

2.3 Collision Detection

[14] merupakan teknik untuk mendeteksi interaksi fisik antar objek dalam game. Penerapan *collision detection* yang efisien sangat penting untuk menjaga konsistensi logika pada permainan dan meningkatkan pengalaman bermain, terutama pada game dengan mekanisme refleks cepat seperti Flappy Bird [1], [6], [15].

2.4 Evaluasi dan Kualitas Game Mobile

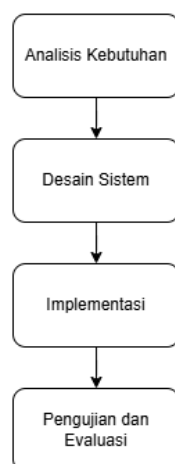
Evaluasi kualitas game mobile dapat dilakukan untuk menilai kualitas permainan dari sisi teknis dan pengalaman pengguna. Pendekatan evaluasi berbasis pemodelan fitur dan analisis data digunakan untuk mengukur tingkat kenyamanan, keterlibatan, dan kepuasan pemain [4], [2], [9].

1. METODE PENELITIAN

Penelitian ini menggunakan pendekatan **rekayasa perangkat lunak eksperimental** (*software-based experimental research*) dengan studi kasus pengembangan game mobile 2D *Flappy Bird*. Pendekatan ini dipilih karena memungkinkan para peneliti untuk mengimplementasikan, mengamati, dan mengevaluasi secara langsung kinerja sistem yang dibangun. Fokus utama penelitian adalah menganalisis penerapan paradigma *event-driven programming* dalam mengelola logika permainan, mekanisme *game loop*, serta interaksi pengguna secara *real-time* menggunakan bahasa pemrograman Java. Evaluasi dilakukan dengan mengamati responsivitas input, stabilitas frame rate, dan akurasi logika permainan pada sistem yang dikembangkan.

1. Rancangan Penelitian

Rancangan penelitian pada studi ini menggunakan metode **Research and Development (R&D)** dengan pendekatan eksperimental. Tahapan penelitian disusun secara berurutan untuk memastikan proses pengembangan dan evaluasi sistem dapat dilakukan secara sistematis dan terukur. Tahapan penelitian dalam pengembangan game 2D *Flappy Bird* disusun secara sistematis seperti ditunjukkan pada Gambar 1.

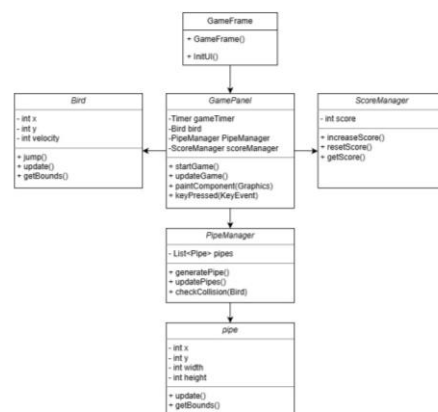


1. **Analisis kebutuhan sistem**, yaitu mengidentifikasi kebutuhan fungsional dan nonfungsional game, termasuk mekanisme interaksi pengguna, logika permainan, serta target performa sistem.
2. **Perancangan sistem**, yang mencakup perancangan arsitektur game, alur event, serta pemodelan hubungan antar objek dalam permainan.
3. **Implementasi sistem**, dengan menerapkan paradigma *event-driven programming* menggunakan bahasa pemrograman Java serta pustaka Java Swing dan AWT.
4. **Pengujian dan evaluasi sistem**, untuk menilai kinerja permainan berdasarkan parameter responsivitas input, stabilitas frame rate, dan keakuratan logika permainan.

2. Arsitektur Sistem Game

Arsitektur sistem pada penelitian ini dirancang menggunakan pendekatan **Object-Oriented Programming (OOP)** dengan memanfaatkan paradigma *event-driven programming*. Struktur sistem dibagi ke dalam beberapa komponen utama yang saling berinteraksi melalui mekanisme event, sehingga proses eksekusi logika permainan tidak berjalan secara linier, melainkan berdasarkan kejadian yang terjadi.

Secara umum, arsitektur sistem terdiri dari dua lapisan utama, yaitu **lapisan antarmuka (presentation layer)** dan **lapisan logika permainan (game logic layer)**.



Gambar 1. Class Diagram

Class diagram pada Gambar 2 menunjukkan struktur sistem game Flappy Bird yang dibangun menggunakan pendekatan Object-Oriented Programming (OOP). Kelas *GameFrame* berfungsi sebagai antarmuka utama, sedangkan *GamePanel* menangani logika permainan, rendering, dan event input. Objek *Bird* merepresentasikan karakter utama dengan mekanisme gravitasi dan lompatan. Rintangan dikelola oleh *PipeManager* yang bertanggung jawab atas pembuatan, pergerakan, dan deteksi tabrakan objek *Pipe*. Penghitungan skor ditangani oleh *ScoreManager* secara terpisah untuk menjaga modularitas sistem.

1. Lapisan Antarmuka (*Presentation Layer*)

Lapisan ini bertanggung jawab dalam membentuk tampilan permainan dan mengelola jendela aplikasi. Implementasi dilakukan menggunakan *JFrame* dengan ukuran layar tetap 360×640 piksel. Pemilihan resolusi ini dimaksudkan untuk merepresentasikan rasio layar perangkat mobile, sehingga perilaku visual game dapat menyerupai lingkungan permainan pada perangkat seluler.

2. Lapisan Logika Permainan (*Game Logic Layer*)

Lapisan ini mengelola seluruh proses inti permainan, meliputi *game loop*, pembaruan posisi objek, deteksi tabrakan, serta penghitungan skor. Komponen ini diimplementasikan dalam kelas yang mewarisi *JPanel*, sehingga mendukung proses *rendering* grafis dan pembaruan tampilan secara berulang melalui mekanisme *repaint()*.

Interaksi antara kedua lapisan tersebut dikendalikan oleh mekanisme event, seperti event waktu (*timer event*) dan event input pengguna. Pendekatan ini memungkinkan sistem memproses logika permainan hanya ketika terjadi suatu kejadian, sehingga penggunaan sumber daya sistem menjadi lebih efisien dan responsif.

3. Implementasi Event-Driven Programming

Implementasi paradigma *event-driven programming* pada penelitian ini bertujuan untuk memastikan bahwa logika permainan dan interaksi pengguna dapat diproses secara responsif dan efisien. Dalam pendekatan ini, alur eksekusi program tidak berjalan secara berurutan, melainkan dipicu oleh kejadian (*event*) tertentu yang terjadi selama permainan berlangsung. Pada sistem yang dikembangkan, terdapat tiga jenis event utama yang mengendalikan jalannya permainan, yaitu event waktu, event input pengguna, dan event pembuatan rintangan.

1. Event Waktu (*Timer Event*)

Event waktu diimplementasikan menggunakan objek *javax.swing.Timer* dengan interval 16,6 milidetik. Interval ini dipilih untuk mempertahankan laju pembaruan tampilan sebesar 60 frame per second (FPS). Setiap event waktu memicu pembaruan logika permainan yang meliputi perhitungan posisi karakter, pergerakan rintangan, pemeriksaan kondisi tabrakan, serta pemanggilan proses *rendering* ulang tampilan permainan.

2. Event Input Pengguna (*Input Event*)

Interaksi pengguna ditangani melalui event input yang diimplementasikan menggunakan antarmuka *KeyListener*. Ketika pengguna menekan tombol spasi, sistem memicu event lompatan yang secara langsung mengubah kecepatan vertikal karakter burung. Pendekatan ini memungkinkan respons input terjadi secara instan tanpa harus menunggu siklus pembaruan logika lainnya, sehingga meningkatkan kenyamanan dan responsivitas permainan.

3. Event Pembuatan Rintangan (*Pipe Generation Event*)

Event ini diatur menggunakan *timer* terpisah yang dipicu secara periodik untuk menghasilkan rintangan pipa baru. Setiap event pembuatan rintangan menentukan posisi vertikal pipa secara acak dalam batas tertentu agar tingkat kesulitan permainan tetap seimbang dan tidak monoton.

Penerapan pada paradigma *event-driven programming* memungkinkan setiap komponen sistem bekerja secara sinkron berdasarkan kejadian yang relevan, sehingga menjaga kestabilan *game loop* dan menghindari pemborosan sumber daya akibat proses yang tidak diperlukan.

4. Logika Pergerakan dan Model Fisika Objek

Logika pergerakan objek dalam permainan dirancang menggunakan model pada fisika sederhana dengan pendekatan diskrit. Model ini digunakan untuk mensimulasikan efek gravitasi terhadap karakter burung serta memastikan pergerakan objek berjalan secara konsisten pada setiap siklus pembaruan permainan.

Pada setiap event waktu, kecepatan vertikal karakter pada burung diperbarui dengan menambahkan konstanta gravitasi. Hubungan antara kecepatan dan posisi vertikal karakter dapat dirumuskan sebagai berikut:

$$v_t = v_t - 1 + g$$

$$y_t = y_t - 1 + v_t$$

Keterangan :

v_t menyatakan kecepatan vertikal karakter pada waktu ke- t

g merupakan konstanta gravitasi yang ditetapkan sebesar 1

y_t menyatakan posisi vertikal karakter burung.

Ketika event input lompatan terjadi, nilai kecepatan vertikal karakter diatur ulang ke nilai negatif tertentu untuk melawan gaya gravitasi, sehingga karakter bergerak ke atas. Nilai ini dipertahankan secara konstan untuk menjaga konsistensi tingkat kesulitan permainan.

Sementara itu, pergerakan rintangan pipa dilakukan secara horizontal dengan kecepatan konstan. Pendekatan ini digunakan untuk menciptakan ilusi pergerakan maju pada permainan tanpa memerlukan perhitungan fisika yang kompleks. Kombinasi antara model

gravitasi sederhana dan kecepatan horizontal tetap memungkinkan permainan berjalan stabil pada laju frame yang konstan.

5. Mekanisme Deteksi Tabrakan

Mekanisme deteksi tabrakan pada penelitian ini dirancang untuk memastikan interaksi antar objek permainan dapat diproses secara akurat dan efisien. Metode yang digunakan adalah **Axis-Aligned Bounding Box (AABB)**, yaitu teknik deteksi tabrakan berbasis persegi pembatas yang sejajar dengan sumbu koordinat.

Setiap objek dalam permainan, baik karakter burung maupun pada rintangan pipa, direpresentasikan sebagai sebuah persegi pembatas yang ditentukan oleh posisi koordinat dan dimensinya. Tabrakan dinyatakan terjadi apabila terdapat irisan antara persegi pembatas objek burung dan pipa.

Pendekatan AABB dipilih karena memiliki kompleksitas komputasi yang rendah dan sesuai untuk game 2D dengan kebutuhan performa tinggi. Deteksi tabrakan dilakukan pada setiap event waktu, sehingga sistem dapat segera merespons kondisi pada tabrakan tanpa menimbulkan jeda (*lag*) yang signifikan.

Selain tabrakan dengan rintangan, kondisi *game over* juga dipicu apabila posisi vertikal karakter burung melewati batas bawah area permainan. Ketika kondisi tersebut terjadi, sistem menghentikan seluruh proses *timer* dan permainan dinyatakan berakhir. Pendekatan ini memastikan konsistensi logika permainan dan mencegah terjadinya kesalahan pembaruan status setelah permainan selesai.

6. Teknik Pengumpulan dan Analisis Data

Teknik pengumpulan data pada penelitian ini dilakukan melalui pendekatan **eksperimental dan observasional** terhadap sistem game yang dikembangkan. Data dikumpulkan selama proses pengujian permainan dengan menjalankan game- game pada lingkungan pengembangan yang sama untuk menjaga konsistensi hasil.

Pengujian dilakukan dengan mengamati perilaku sistem pada beberapa aspek utama, yaitu responsivitas input pengguna, stabilitas frame rate, serta keakuratan logika permainan. Responsivitas input diukur dengan mengamati perubahan posisi karakter burung segera setelah event input lompatan diberikan. Stabilitas frame rate dievaluasi berdasarkan kemampuan sistem mempertahankan laju pembaruan tampilan sebesar 60 frame per second (FPS) selama permainan berlangsung.

Selain itu, keakuratan logika permainan dianalisis melalui pengujian deteksi tabrakan dan mekanisme pembaruan skor. Sistem dianggap berjalan dengan baik apabila tabrakan terdeteksi secara tepat dan skor bertambah setiap kali karakter burung berhasil melewati rintangan tanpa mengalami keterlambatan atau kesalahan perhitungan.

Data yang diperoleh dianalisis secara **deskriptif kualitatif** dengan membandingkan hasil pengujian terhadap tujuan implementasi sistem. Analisis difokuskan pada sejauh mana penerapan paradigma *event-driven programming* mampu meningkatkan responsivitas dan kestabilan logika permainan pada game mobile 2D yang dikembangkan.

1.1.1. Gambar dan tabel. (Figures and tables)

Semua **Tabel** dan **Gambar** yang anda masukkan dalam dokumen harus jelas terbaca.

1.1.2. Rumus Matematika. (Math formula.)

Jika anda menggunakan *Word*, gunakan persamaan Microsoft Equation Editor atau *MathType*, ditulis ditengah, dan diberi nomor persamaan mulai dari (1), (2) dst. (If you use *Word*, use the Microsoft Equation Editor or *MathType* equation, centered, and numbered equations starting from (1), (2) etc.)

$$p(x, y); (0 \leq x \leq M - 1, 0 \leq y \leq N - 1) \quad (1)$$

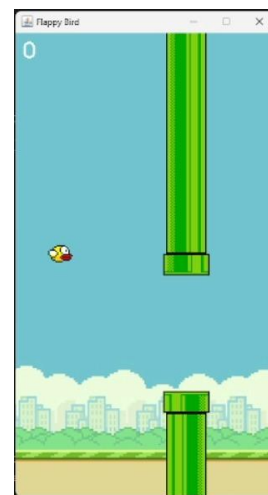
3. HASIL DAN PEMBAHASAN

3.1 Hasil Implementasi Game Flappy Bird

Game Flappy Bird ini memiliki gameplay yang sangat sederhana. Flappy Bird adalah permainan yang mengharuskan pemainnya selalu mengontrol dengan melihat

ketinggian pesawat dengan mengetuk layar agar dapat melewati celah diantara dua pipa sebanyak mungkin yang selalu berdatangan. Permainan ini memiliki tujuan untuk melewati celah antar pipa yang memiliki ketinggian beraneka ragam.

Struktur pada sistem yang memisahkan antarmuka *user interface* dan logika permainan memungkinkan proses eksekusi berjalan lebih terkontrol. Tampilan permainan ditampilkan pada resolusi 360×640 piksel untuk menyerupai rasio layar perangkat mobile, sehingga interaksi visual tetap konsisten selama permainan berlangsung. Pendekatan ini sejalan dengan karakteristik game mobile 2D yang menekankan efisiensi pemrosesan dan kesederhanaan arsitektur sistem [1], [7].



Gambar 2. Tampilan Game Start

3.2 Responsivitas Input Pengguna

Respons sistem terhadap input pengguna diuji melalui mekanisme lompatan karakter yang dipicu oleh penekanan tombol spasi. Hasil pengamatan menunjukkan bahwa setiap input yang diberikan langsung direspons oleh sistem melalui perubahan kecepatan vertikal karakter tanpa jeda yang terlihat.

Penerapan event-driven programming pada mekanisme input memungkinkan proses interaksi pengguna berjalan secara terpisah dari pembaruan logika permainan, sehingga responsivitas tetap terjaga meskipun permainan berjalan secara real-time. Temuan ini mendukung penelitian sebelumnya yang menyatakan bahwa pemrograman berbasis

event efektif dalam meningkatkan kecepatan respons dan kenyamanan bermain pada game kasual[4],[10].

3.3 Stabilitas Frame Rate Permainan

Stabilitas frame rate menjadi salah satu indikator performa utama pada game real-time. Selama pengujian, permainan mampu mempertahankan laju pembaruan tampilan sebesar 60 frame per second (FPS) secara konsisten.

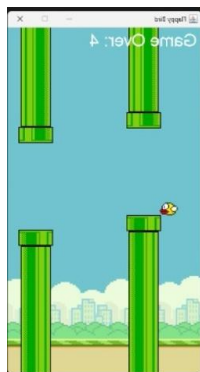
Penggunaan mekanisme timer berbasis event memastikan pembaruan logika permainan dan rendering grafis berjalan pada interval waktu yang tetap. Pendekatan ini terbukti mampu mencegah terjadinya penurunan performa meskipun jumlah objek permainan meningkat, sebagaimana dibahas pada penelitian terkait optimasi performa dan arsitektur game mobile [8], [12], [13].

3.4 Evaluasi Logika Pergerakan Objek

Pergerakan karakter dan rintangan dievaluasi berdasarkan konsistensi dan kestabilannya selama permainan berlangsung. Model fisika sederhana berbasis gravitasi menghasilkan pergerakan vertikal karakter yang halus dan mudah dikontrol oleh pemain.

Penggunaan model fisika sederhana ini sesuai dengan prinsip pengembangan game 2D yang menekankan keseimbangan antara realisme dan performa sistem. Pendekatan serupa juga banyak digunakan pada game kasual untuk menjaga kestabilan permainan tanpa membebani komputasi sistem [7], [11].

3.5 Deteksi Tabrakan dan Kondisi Game Over



Gambar 3. Tampilan Game Over

Deteksi tabrakan antara karakter dan rintangan dilakukan menggunakan metode Axis-Aligned

Bounding Box (AABB). Selama pengujian, sistem mampu mendeteksi tabrakan secara akurat dan langsung menghentikan permainan ketika kondisi tabrakan atau batas area permainan terlampaui.

Metode AABB dipilih karena memiliki kompleksitas komputasi yang rendah dan sesuai untuk game 2D dengan kebutuhan performa tinggi. Hasil ini sejalan dengan penelitian yang menyebutkan bahwa AABB merupakan metode deteksi tabrakan yang efisien dan banyak diterapkan pada pengembangan game dua dimensi [3], [11].

3.6 Pembaruan Skor

Pembaruan skor dilakukan setiap kali karakter berhasil melewati rintangan tanpa mengalami tabrakan. Skor bertambah secara konsisten sesuai dengan kondisi permainan dan ditampilkan secara real-time pada layar.

Konsistensi pembaruan skor berkontribusi terhadap kejelasan progres permainan dan meningkatkan keterlibatan pemain. Hal ini sesuai dengan kajian evaluasi kualitas game yang menekankan pentingnya umpan balik sistem yang jelas terhadap aksi pengguna [5], [9].

3.7 Pembahasan Penerapan Event-Driven Programming

Penerapan event-driven programming memungkinkan setiap komponen sistem merespons kejadian yang relevan secara langsung. Pendekatan ini menghasilkan alur eksekusi program yang lebih efisien serta meningkatkan responsivitas permainan secara keseluruhan.

Dibandingkan dengan pendekatan linier konvensional, pemrograman berbasis event memberikan fleksibilitas yang lebih baik dalam pengelolaan input, pembaruan logika, dan rendering tampilan secara sinkron. Temuan ini konsisten dengan penelitian terkait penerapan event-driven programming pada sistem real-time [4], [10].

3.8 Implikasi Hasil

Hasil implementasi menunjukkan bahwa paradigma event-driven programming sesuai diterapkan pada pengembangan game mobile 2D dengan mekanisme sederhana. Pendekatan ini mampu menjaga performa sistem sekaligus meningkatkan kualitas interaksi pengguna.

Implikasi praktis dari penelitian ini adalah tersedianya model pengembangan game 2D yang responsif dan mudah dikembangkan lebih lanjut. Secara teoretis, hasil ini memperkuat konsep bahwa pemrograman berbasis kejadian memiliki peran penting dalam meningkatkan kualitas sistem game real-time [12], [15].

4. KESIMPULAN

Penelitian ini menunjukkan bahwa penerapan bahwa paradigma *event-driven programming* pada pengembangan game mobile 2D Flappy Bird berbasis Java mampu meningkatkan responsivitas sistem dan menjaga kestabilan logika permainan secara real-time. Penggunaan mekanisme timer event berhasil mempertahankan frame rate yang konsisten, sementara pengelolaan input berbasis event memungkinkan respons yang cepat terhadap aksi pengguna. Selain itu, penerapan metode deteksi tabrakan Axis-Aligned Bounding Box (AABB) terbukti efektif dan efisien dalam menjaga konsistensi permainan tanpa menurunkan performa sistem. Secara keseluruhan, hasil penelitian ini membuktikan bahwa pendekatan event-driven programming layak dapat digunakan sebagai dasar pengembangan game mobile 2D sederhana dan dapat dikembangkan lebih lanjut untuk sistem permainan yang lebih kompleks.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada seluruh pihak yang telah memberikan dukungan dan kontribusi dalam pelaksanaan penelitian ini, khususnya kepada Program Studi Sistem Informasi Universitas Islam Negeri Sumatera Utara atas fasilitas dan bimbingan yang diberikan selama proses penelitian berlangsung.

DAFTAR PUSTAKA

- [1] P. Mileff, "Collision detection in 2D games," *Production Systems and Information Engineering*, vol. 11, p. 10, Sep. 2023, doi: 10.32968/psaie.2023.3.2.
- [2] J. K. Ochab, P. Węgrzyn, P. Witaszczyk, D. Dążyk, and G. J. Nalepa, "Mobile Game Evaluation Method Based on Data Mining of Affective Time Series," *Sensors*, vol. 25, no. 9, p. 2756, 2025, doi: 10.3390/s25092756.
- [3] S. Retno, "Implementation of Game Design Document (GDD) in the Development of the 2D Android-Based Game Komodo Isle," *Gameology and Multimedia Expert*, vol. 2, pp. 45–54, Apr. 2025, doi: 10.29103/game.v2i2.21228.
- [4] A. Solihin, E. Hidayat, and A. Putra Aldya, "Application of the Finite State Machine Algorithm on 2D Platformer Rabbit Games vs Zombies," *Jurnal Online Informatika*, vol. 4, p. 33, Sep. 2019, doi: 10.15575/join.v4i1.293.
- [5] I. Sarifah, A. Muhajir, A. Marini, G. Yarmi, and ..., "Mobile games and learning interest: for fifth graders in mathematics," ... of Education and ..., no. Query date: 2025-12-22 14:29:47, 2025, [Online]. Available: <http://edulearn.intelektual.org/index.php/EduLearn/article/view/21118>
- [6] C. Ma and J. Shao, "Modeling Mobile Game Design Features Through Grounded Theory: Key Factors Influencing User Behavior," *Journal of Theoretical and Applied Electronic Commerce Research*, vol. 20, no. 2, p. 132, 2025, doi: 10.3390/jtaer20020132.
- [7] R. Khamtue, Y. Tongpaeng, and K. Kemavuthanon, "Learning 2D Mobile Casual Musical Pitch Training Game Application (Pitchero)," in *2023 7th International Conference on Information Technology (InCIT)*, Nov. 2023, pp. 315–318. doi: 10.1109/InCIT60207.2023.10413013.
- [8] J. Horton, *Learning Java by Building Android Games: Learn Java and Android from scratch by building five exciting games*. Packt Publishing, 2021. [Online]. Available: <https://books.google.co.id/books?id=gowmEAAAQBAJ>
- [9] H. Teng, Z. Li, K. Cao, S. Long, S. Guo, and A. Liu, "Game Theoretical Task Offloading for Profit Maximization in Mobile Edge Computing," *IEEE Transactions on Mobile Computing*, vol. 22, no. 9, pp. 5313–5329, Sep. 2023, doi: 10.1109/TMC.2022.3175218.
- [10] J. A. Zhang et al., "Enabling Joint Communication and Radar Sensing in Mobile Networks—A Survey," *IEEE Communications Surveys & Tutorials*, vol. 24,

- no. 1, pp. 306–345, Firstquarter 2022, doi: 10.1109/COMST.2021.3122519.
- [11] A. Alcantara, J. Galasan, R. Remojo, and ..., “PLAYJAVA: An Android-based Tutorial Game for Learning Basic Java Programming,” *Information ...*, no. Query date: 2025-12-22 14:25:20, 2025, [Online].
- [12] B. Xie, “High Altitude Railway Construction Risk Warning Based on Improved Game Theory-2d Cloud Model,” *Railway Standard Design*, vol. 69, no. 2, pp. 144–152, 2025, doi: 10.13238/j.issn.1004-2954.202306120001.
- [13] H. T. T. Saurik, “Evaluating Player Experience for Fear Modeling of 2D East Java Horror Game Alas Tilas,” *Jurnal Resti*, vol. 7, no. 4, pp. 858–864, 2023, doi: 10.29207/resti.v7i4.5043.
- [14] R. Fan, “Learning Collision-Free Space Detection from Stereo Images: Homography Matrix Brings Better Data Augmentation,” *IEEE ASME Transactions on Mechatronics*, vol. 27, no. 1, pp. 225–233, 2022, doi: 10.1109/TMECH.2021.3061077.
- [15] J. D. M. Esteban et al., “OCLEAN: An Endless 2D Mobile Game Focused on The Awareness of Cleaning Marine Plastic Waste,” in *2022 2nd International Conference in Information and Computing Research (iCORE)*, Dec. 2022, pp. 139–143. doi: 10.1109/iCORE58172.2022.00045.