

PERANCANGAN GAME ANDROID *ENDLESS-RUNNER* DENGAN EFEK VISUAL DINAMIS MENGGUNAKAN METODE FOUNTAIN

Dicky Alviriandy

Fakultas Teknik dan Komputer, Prodi Teknik Informatika, Universitas Harapan Medan

Medan, Indonesia

Email: dickyalviriandy45@gmail.com

Keywords

Android Game, Endless-Runner, Dynamic Visual Effects, Fountain Method, Unity

Game Android, Endless-Runner, Efek Visual Dinamis, Metode Fountain, Unity

Abstract

The rapid growth of the Android-based mobile gaming industry has driven innovation in both visual and gameplay aspects. However, many endless-runner games still present monotonous visuals, making them less engaging for players. This study aims to design and develop an endless-runner game with dynamic visual effects using the Fountain method, which allows the design, implementation, and testing stages to run concurrently, improving development efficiency. The game was developed with Unity using the C# programming language and designed with Figma. Key features include automatic background changes based on score, particle effects during jumps and collisions, and a simple yet functional interface. Black box testing was applied to verify that all features worked as expected. The results showed that menu navigation, character control, collision detection, score calculation, background transitions, and visual effects ran smoothly without errors on Android devices running version 10 or above. This game successfully meets the research objective of producing an endless-runner game that is visually appealing and responsive to player actions.

Perkembangan industri game mobile berbasis Android mendorong inovasi dalam aspek visual dan gameplay. Banyak game endless-runner masih memiliki tampilan monoton sehingga kurang menarik. Penelitian ini merancang dan mengembangkan game endless-runner dengan efek visual dinamis menggunakan metode Fountain, yang memungkinkan tahapan desain, implementasi, dan pengujian berjalan secara paralel untuk meningkatkan efisiensi. Game dikembangkan menggunakan Unity dengan bahasa C# dan dirancang melalui Figma. Fitur utama meliputi pergantian latar otomatis berdasarkan skor, efek partikel saat karakter melompat dan bertabrakan, serta antarmuka sederhana namun fungsional. Pengujian dilakukan menggunakan black box testing untuk memastikan seluruh fitur berjalan sesuai spesifikasi. Hasil pengujian menunjukkan bahwa navigasi menu, kontrol karakter, deteksi tabrakan, perhitungan skor, pergantian latar, dan efek visual berfungsi baik tanpa error pada perangkat Android versi 10 ke atas. Game ini berhasil memenuhi tujuan penelitian, yaitu menghasilkan endless-runner yang menarik secara visual dan responsif terhadap aksi pemain.

1. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi saat ini tumbuh sangat cepat di berbagai bidang, salah satunya perkembangan *game*. *Game* merupakan suatu media hiburan yang digunakan oleh berbagai kalangan untuk menghilangkan rasa jenuh. *Game* tidak hanya digunakan sebagai hiburan, akan tetapi *game* juga bisa sebagai media belajar, lahan bisnis, dan sebagai cabang olahraga [1].

Dalam dunia rekayasa perangkat lunak, terdapat berbagai metode kerja *Software Development Life Cycle* (SDLC) yang umum digunakan oleh para pengembang. Beberapa metode yang paling dikenal antara lain adalah metode waterfall. 'Metode waterfall setiap langkahnya memiliki proses dan fungsinya memiliki masing-masing tersendiri dalam membantu keberhasilan pembuatannya tersendiri' [2]. Menurut [3] 'metode ini juga memiliki kekurangan jika terjadi kesalahan kecil akan menjadi kesalahan besar dan merugikan'. Kerugian tersebut dapat berupa revisi mendasar pada fase akhir menghambat alur kerja dan *deadline* proyek dan akumulasi *bug* yang tidak tertangani mengurangi stabilitas sistem.

Kekurangan pada metode konvensional seperti waterfall, yang bersifat linier dan kaku sehingga sulit beradaptasi dengan perubahan kebutuhan, menunjukkan adanya sebuah masalah untuk proyek pengembangan *game*.

Dibutuhkan sebuah metode kerja yang dapat menyeimbangkan antara kebutuhan akan struktur yang jelas dan fleksibilitas untuk beradaptasi. Selain itu beberapa *game endless-runner* memiliki visual yang monoton seperti *game google dino chrome* yang hanya menampilkan visual warna hitam putih dan tidak ada pergantian latar tempat.

Metode fountain diajukan sebagai sebuah solusi alternatif yang berpotensi untuk menjawab masalah tersebut. Metode fountain adalah perbaikan logis dari metode waterfall. Dengan menggunakan model fountain, memungkinkan terjadinya tumpang tindih antar tiap proses perancangan aplikasi dan tidak berfokus pada satu langkah saja [4].

Hasil penelitian dari [5] terlihat bahwa metode fountain cocok untuk perancangan aplikasi dikarenakan metode fountain memungkinkan untuk melewati tahapan-tahapan yang lain kecuali tahapan design dibandingkan dengan metode waterfall yang harus melewati tahapan sesuai urutan.

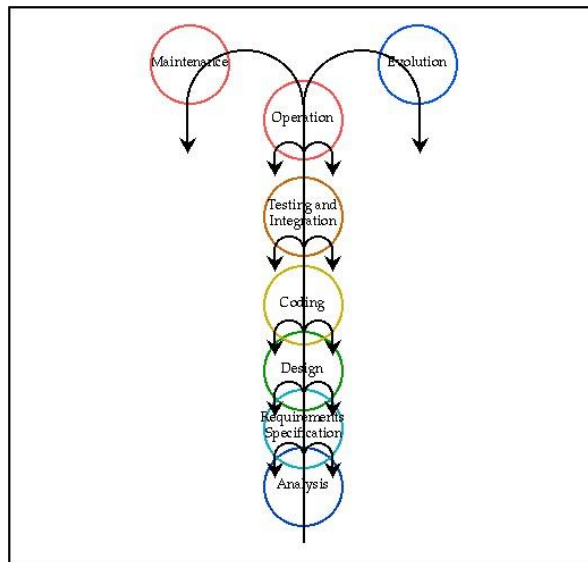
Pada penelitian ini, metode fountain akan diterapkan dan dianalisis dalam sebuah studi kasus nyata, yaitu proyek perancangan *game* android. *Game* tersebut, termasuk di

dalamnya proses pembuatan fitur "efek visual dinamis", akan menjadi tolak ukur untuk mengevaluasi efektivitas metode kerja ini.

TINJAUAN PUSTAKA

METODE FOUNTAIN

Metode ini dikenal dengan model air mancur dimana arus air akan naik keatas melalui tengah dan kemudian turun kembali. Metode Fountain merupakan perbaikan logis dari model waterfall [6]. Model ini dipilih karena sesuai dengan kebutuhan pengembangan game yang memerlukan fleksibilitas dalam perancangan dan implementasi. Tahapan dalam metode fountain terbagi dalam 6 proses utama, yaitu *analysis*, *requirement specifcaitons*, *design*, *coding*, *testing and integration*, *operation*, dan 2 proses pembaruan aplikasi, yaitu *maintenance* dan *evolution*. Setiap proses bisa kembali ke tahapan sebelumnya jika terjadi kesalahan atau ada kekurangan kebutuhan untuk melanjutkan ke tahapan selanjutnya sesuai dengan gambar berikut.



Gambar 1, Tahapan Metode Fountain

Adapun tahapan-tahapan tersebut dijelaskan sebagai berikut.

Analysis

Ini adalah tahap paling awal dari siklus pengembangan. Pada tahap ini, pengembang atau analis sistem mengidentifikasi masalah, mempelajari kelayakan proyek, dan mengumpulkan kebutuhan awal dari pengguna atau pemangku kepentingan.

Requirements Specification

Pada tahap ini, semua kebutuhan yang telah diidentifikasi pada fase analisis didokumentasikan secara formal dan rinci. Hasil dari tahap ini biasanya adalah

dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) yang mencakup kebutuhan fungsional dan non-fungsional. Panah iteratif dengan tahap *analysis* dan *design* menunjukkan bahwa spesifikasi ini dapat direvisi jika ada temuan baru saat analisis lebih dalam atau saat menghadapi kendala teknis pada tahap perancangan.

Design

Tahap ini berfokus pada pembuatan "cetak biru" atau arsitektur dari perangkat lunak yang akan dibangun. Ini meliputi perancangan alur sistem, desain antarmuka pengguna (UI/UX), desain basis data, dan arsitektur teknis lainnya. Perancangan dilakukan berdasarkan dokumen spesifikasi kebutuhan.

Coding

Ini adalah tahap di mana para programmer menerjemahkan hasil dari tahap perancangan menjadi kode program yang fungsional menggunakan bahasa pemrograman yang telah ditentukan. Dalam metode fountain, tahap ini sangat erat kaitannya dengan tahap *design* dan *testing*. Programmer bisa meminta klarifikasi atau perubahan desain, dan setiap modul kode yang selesai akan langsung diteruskan ke tahap pengujian.

Testing and Integration

Pada tahap ini, modul-modul kode yang telah dibuat diuji untuk menemukan dan memperbaiki *bug* atau kesalahan (pengujian unit). Setelah itu, modul-modul tersebut digabungkan (integrasi) dan diuji kembali secara keseluruhan untuk memastikan semua bagian sistem dapat bekerja sama dengan baik (pengujian integrasi).

Operation

Ini adalah tahap di mana perangkat lunak yang telah lulus pengujian akhirnya diluncurkan dan digunakan oleh pengguna akhir dalam lingkungan yang sesungguhnya. Dalam diagram, tahap ini merupakan puncak dari pengembangan awal, dari mana siklus hidup perangkat lunak selanjutnya akan bercabang.

Maintenance

Setelah produk beroperasi, tahap pemeliharaan dimulai. Aktivitas pada tahap ini meliputi perbaikan *bug* yang baru ditemukan oleh pengguna, pembaruan kecil, dan dukungan teknis untuk memastikan sistem tetap berjalan dengan stabil dan andal. Ini adalah proses untuk menjaga versi produk yang sudah ada.

Evolution

Berjalan paralel dengan *maintenance*, tahap evolusi berfokus pada pengembangan produk ke tingkat selanjutnya. Ini bukan sekadar perbaikan, melainkan penambahan fitur-fitur baru yang signifikan, perombakan teknologi, atau adaptasi produk untuk memenuhi kebutuhan pasar yang berubah. Tahap ini pada dasarnya memulai siklus pengembangan baru untuk versi produk berikutnya.

GAME

Game merupakan suatu media hiburan yang digunakan oleh berbagai kalangan untuk menghilangkan rasa jenuh. *Game* tidak hanya digunakan sebagai hiburan, akan tetapi *game* juga bisa sebagai media belajar, lahan bisnis, dan sebagai cabang olahraga [1]. Video *game* berbeda dari jenis media yang lainnya dikarenakan pemain diberikan kontrol untuk memilih pilihan dan aksi yang bermoral, daripada hanya membiarkan penonton untuk menilai terhadap aksi dan pilihan yang mereka tidak memiliki kontrol untuk memilih. Video *game* juga membiarkan pemain untuk menentukan pilihan yang biasanya berdasarkan dari moralitas pemain tersebut sendiri [7].

UNITY

Unity adalah sebuah *game engine* yang memberikan keuntungan yang besar dibandingkan *game engine* lainnya yang terdaftar di pasaran saat ini. Unity memberikan kapabilitas drag and drop pada alur kerja visualnya serta mendukung pemrograman pada bahasa c#, yang mana bahasa tersebut sangat terkenal. Unity sudah mendukung pengembangan grafis 3d dan 2d, juga menyediakan seperangkat peralatan untuk dua jenis grafis ini yang selalu berkembang, menjadi semakin mudah digunakan pada setiap pembaruan [8].

ENDLESS-RUNNER

Endless runner game adalah jenis permainan yang tidak ada akhir, dan tujuan pemain adalah harus mendapatkan skor tertinggi. Jenis permainan ini ialah untuk berlari sejauh mungkin dan melewati beberapa rintangan untuk menghindari kekalahan dan mendapatkan score tertinggi. *Endless runner game* dapat di mainkan pada semua usia, karena dengan konsep permainan yang mudah di mengerti dan di pahami. Jenis *game* ini bisa di mainkan di *smartphone* android dengan menggunakan model karakter 3D [8].

ANDROID

Android adalah sebuah sistem operasi (*operating system*) yang diterapkan pada perangkat mobile dan banyak digunakan pada *smartphone* saat ini. Sistem ini

menggunakan basis linux yang dijadikan sebagai pondasi dasar menjalankan sistem operasi android. Linux sendiri adalah salah satu sistem operasi yang digunakan oleh perangkat komputer. Android dirancang khusus untuk perangkat yang memiliki kemampuan layar sentuh. Saat ini hampir semua perangkat smartphone menggunakan *touch screen* [9].

VISUAL DINAMIS

Menurut [9] Visual dinamis adalah jiwa dari responsivitas sebuah *game*. Mereka memberi tahu pemain. Misalnya, ketika karakter melompat, animasi '*squash-and-stretch*' yang halus atau efek partikel yang mengikuti gerakan bisa membuat pergerakan terasa hidup. Ini bukan sekadar hiasan melainkan mekanisme umpan balik yang mengkomunikasikan aturan *game* dan emosi. Tombol yang tertekan secara visual saat diklik, musuh yang berkedip merah saat terkena, atau koin yang berkilau saat dikumpulkan interaksi mikro ini menciptakan kesenangan taktil. Semakin dinamis dan berlebihan respons ini, semakin 'hidup' rasa *game* tersebut [10].

BAHASA PEMROGRAMAN C#

C # (C Sharp) adalah sebuah bahasa pemrograman berbasis objek yang didukung dapat berkomunikasi dengan sistem operasi yang digunakan oleh komputer . Selain itu , microsoft .Net Framework memungkinkan C# untuk berkomunikasi dengan bahasa pemrograman lainnya yang juga didukung oleh .Net Framework seperti VB . NET , F # , atau C ++. Dengan kata lain , aplikasi yang kita buat dapat menggunakan komponen - komponen yang lain yang dibuat dengan menggunakan VB . NET , F # , atau C ++ [11].

2. METODE PENELITIAN

ANALISIS KEBUTUHAN

Analisis kebutuhan adalah tahap awal dalam pengembangan sistem yang bertujuan untuk mengidentifikasi dan mendokumentasikan persyaratan yang diperlukan untuk membangun sebuah *game*. Analisa ini mencakup aspek *hardware* dan *software* yang relevan.

Analisis Hardware

Spesifikasi *hardware* laptop yang digunakan untuk membuat game meliputi :

- a. *Windows 10 Home.*
- b. *Processor AMD Ryzen 3 5300u.*

- c. *Integrated Graphics Processing Unit (IGPU) AMD Vega 6.*
- d. *Ram 8 gb dan Rom 256gb.*
- e. *Vram 2gb.*
- f. *DirectX version 12.*

Sedangkan *hardware* android untuk menguji game sebagai berikut :

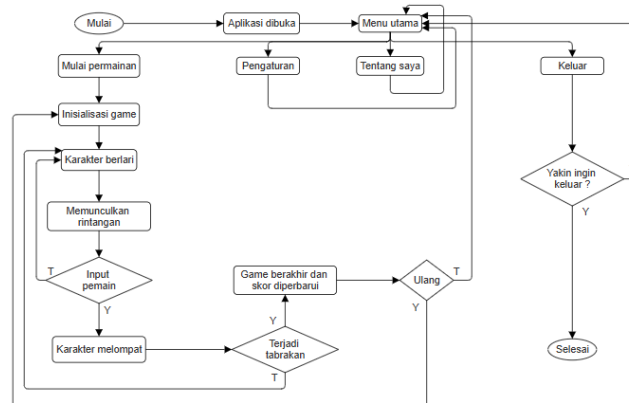
- a. *Device Realme 9 Pro 5G.*
- b. *Processor Snapdragon 695.*
- c. *GPU Adreno 619.*
- d. *Ram 8gb dan Rom 128gb.*
- e. *Android version 14.*

Analisis Software

Software utama yang digunakan untuk membuat game yaitu Unity 2D dan Figma. Unity 2D sebagai *engine* utama game sedangkan Figma sebagai alat untuk desain.

ALUR GAMEPLAY

Perancangan alur *gameplay* menggambarkan urutan aktivitas atau aliran kerja dalam *game*. Ini menunjukkan bagaimana pemain berinteraksi dengan sistem dan bagaimana sistem meresponsnya. Perancangan alur *game* ini dapat divisualisasikan menggunakan flowchart berikut.



Gambar 2. Flowchart Gameplay

Pada gambar 2 tersebut menggambarkan alur permainan dari game *endless runner* yang dirancang. Proses dimulai saat aplikasi dibuka dan pengguna diarahkan ke menu utama yang berisi pilihan mulai permainan, pengaturan, tentang saya, serta keluar. Jika pemain memilih mulai permainan, maka sistem ingin melakukan inisialisasi game, kemudian karakter akan otomatis berlari dan rintangan mulai dimunculkan. Pemain dapat memberikan input berupa lompatan untuk menghindari rintangan, jika

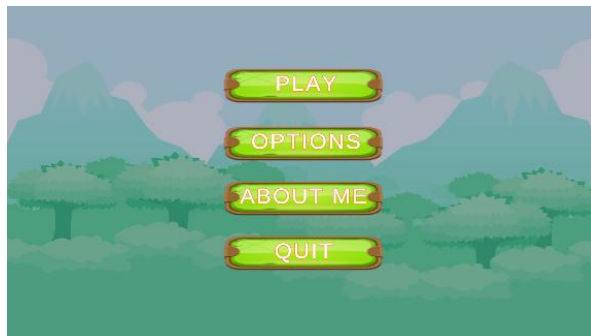
berhasil maka permainan terus berlanjut, namun jika terjadi tabrakan maka game berakhir dan skor diperbarui. Setelah itu pemain dapat memilih untuk mengulang permainan atau kembali ke menu utama. Selain itu, jika pengguna memilih opsi keluar, sistem akan menampilkan konfirmasi sebelum benar-benar mengakhiri aplikasi. Flowchart ini menjelaskan secara terstruktur interaksi pengguna dan alur logika game dari awal hingga selesai.

3. HASIL DAN PEMBAHASAN

TAMPILAN GAME

Tampilan Menu Utama

Tampilan ini menunjukkan halaman menu utama saat pertama kali aplikasi *game* dibuka.



Gambar 3. Tampilan Menu Utama Game

Tampilan *Gameplay*

Tampilan ini menunjukkan dari *gameplay* permainan *endless-runner*. Karakter akan terus berlari ke kanan dan pemain harus membuatnya menghindari rintangan yang ada dengan cara menyentuh layar agar karakter melompat.



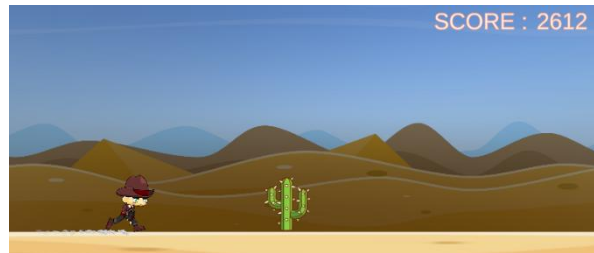
Gambar 4. Tampilan *Gameplay*

Tampilan Perubahan Latar

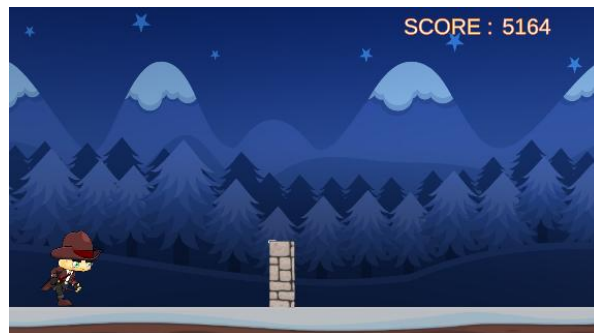
Tampilan ini menunjukkan akan adanya perubahan latar tempat pada *game* disaat skor permainan bertambah 2000. Perubahan latar tempat memiliki 3 macam latar, yaitu kota, mesir dan salju.



Gambar 5. Tampilan Latar Kota Game



Gambar 6. Tampilan Latar Mesir



Gambar 7. Tampilan Latar Salju

Tampilan Efek Partikel

Tampilan ini menunjukkan efek partikel ketika karakter berlari dan melompat. Efek partikel hanya berupa seperti asap.



Gambar 8. Tampilan Efek Partikel

Tampilan *Game Over*

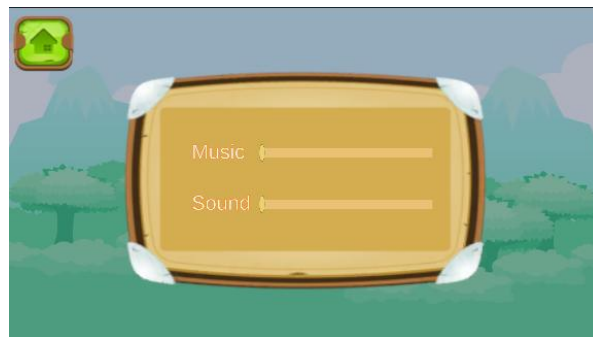
Tampilan ini menunjukkan peralihan *scene game* saat karakter menabrak rintangan. Pada *scene game over*, akan menampilkan total perolehan jarak tempuh skor selama karakter berlari dan terdapat pilihan untuk mengulang permainan atau kembali ke halaman menu utama *game*.



Gambar 9. Tampilan Game Over

Tampilan Menu *Options*

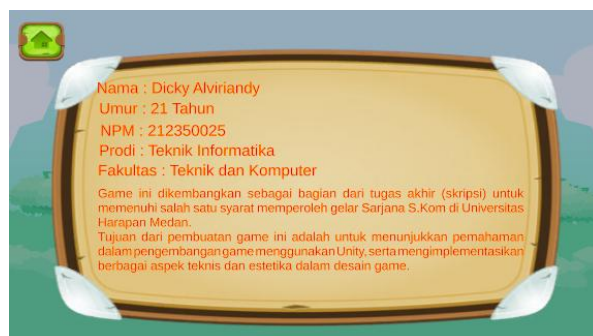
Tampilan ini dapat mengatur *volume* dan *sound effect* dari musik permainan selama *game* dijalankan. Pemain dapat mengaturnya sesuai keinginan.



Gambar 10. Tampilan Menu *Options*

Tampilan Menu *About Me*

Tampilan ini hanya menampilkan seputar informasi terkait penulis.



Gambar 11. Tampilan Menu *About Me*

Tampilan Menu *Quit*

Tampilan ini menunjukkan jika pemain ingin keluar dan menutup penuh aplikasi, *game* akan menampilkan pemberitahuan informasi terlebih dahulu. Jika pemain

memilih tombol simbol “ya/benar” maka *game* akan ditutup penuh, jika pemain memilih tombol simbol “tidak” maka tampilan informasi akan dihilangkan.



Gambar 12. Tampilan Menu Quit

PENGUJIAN

Pengujian dilakukan menggunakan *Black Box Testing* untuk memastikan setiap fungsi utama dalam *game* berjalan sesuai dengan yang diharapkan tanpa memperhatikan kode program secara internal. Pengujian difokuskan pada fungsionalitas yang dapat diamati oleh pengguna, mencakup seluruh menu, kontrol karakter, efek visual dinamis, serta transisi antar *scene*. Tabel berikut menunjukkan skenario pengujian, langkah uji, hasil yang diharapkan, hasil aktual, dan status pengujian :

Tabel 1. Pengujian Fungsional

No	Skenario	Hasil Yang Diharapkan	Hasil
1	Menjalankan Aplikasi	Aplikasi terbuka	Berhasil
2	Menampilkan Menu Utama	Halaman menu utama berfungsi dengan baik	Berhasil
3	Fungsi tombol Play	Memulai permainan	Berhasil
4	Fungsi tombol Options	Membuka menu options	Berhasil
No	Skenario	Hasil Yang Diharapkan	Hasil
5	Fungsi tombol About Me	Membuka menu about me	Berhasil
6	Fungsi tombol Quit	Menampilkan informasi untuk menutup aplikasi	Berhasil

7	Input layar untuk melompat	Karakter akan melompat	Berhasil
8	Muncul rintangan	Memunculkan rintangan secara acak	Berhasil
9	Deteksi tabrakan	Game akan dialihkan ke layar game over	Berhasil
10	Tombol ulang di Game Over	Memulai ulang permainan	Berhasil
11	Tombol home di Game Over	Kembali ke halaman menu utama	Berhasil
12	Pergantian latar	Mengubah latar tempat setiap skor mencapai 2000	Berhasil
13	Efek partikel melompat dan berlari	Memunculkan efek partikel asap saat karakter melompat dan berlari	Berhasil

INTEGRASI

Setelah seluruh modul dan fitur pada *game* diuji secara individual menggunakan metode *Black Box Testing*, tahap selanjutnya adalah integrasi. Integrasi dilakukan dengan menggabungkan seluruh modul menjadi satu sistem utuh, kemudian memastikan bahwa setiap modul dapat berinteraksi dengan modul lain tanpa menimbulkan konflik atau error.

Adapun proses integrasi pada *game endless-runner* dengan meliputi:

1. Integrasi menu utama dengan *scene gameplay*
 - a. Menghubungkan tombol *play* di menu utama agar dapat memanggil *scene gameplay*.
 - b. Menghubungkan tombol *options*, *about me*, dan *quit* agar mengarah ke scene atau fungsi yang sesuai.
2. Integrasi kontrol karakter dengan sistem rintangan
 - a. Memastikan karakter yang berjalan otomatis dapat merespons input pemain untuk melompat.
 - b. Menghubungkan deteksi tabrakan rintangan dengan fungsi *game over*.
3. Integrasi sistem skor dengan pergantian latar
 - a. Menghubungkan sistem penghitungan skor dengan logika pergantian latar otomatis setiap skor mencapai kelipatan 2000.
4. Integrasi efek visual dengan *event* permainan

- a. Menghubungkan efek partikel berlari saat karakter terus berlari.
- b. Menghubungkan efek partikel lompat dengan event input pemain saat melompat.
5. Integrasi *scene game over* dengan *scene* lain
 - a. Memastikan tombol ulang mengarahkan pemain untuk memulai permainan dari awal dengan skor direset.
 - b. Memastikan tombol *home* mengembalikan pemain ke menu utama.

Berdasarkan hasil pengujian setelah proses integrasi, seluruh modul dapat berfungsi dengan baik dan saling berinteraksi sesuai dengan spesifikasi. Tidak ditemukan *error* atau *bug* yang mengganggu selama proses transisi antar *scene*, penghitungan skor, pergantian latar, maupun pemanggilan efek visual.

OPERASI (OPERATION)

Tahap operasi merupakan proses menjalankan *game endless-runner* pada perangkat target, yaitu *smartphone* android versi 10 ke atas. Pada tahap ini, aplikasi dijalankan dalam kondisi normal sesuai tujuan pengembangan, sehingga pengguna dapat memanfaatkan seluruh fitur yang telah disediakan.

Berikut adalah langkah-langkah operasi *game*:

1. Persiapan perangkat
 - a. Pastikan perangkat Android memiliki sistem operasi minimal versi 10.
 - b. Pastikan terdapat ruang penyimpanan yang cukup untuk menginstal *game*.
2. Instalasi *game*
 - a. Salin file APK hasil *build* dari unity ke perangkat android.
 - b. Lakukan proses instalasi dengan memberikan izin pemasangan aplikasi dari sumber tidak dikenal (jika diperlukan).
3. Menjalankan *game*
 - a. Buka ikon *game* pada layar utama perangkat android.
 - b. Tunggu hingga muncul menu utama.
4. Navigasi menu utama
 - a. *Play* akan memulai permainan.
 - b. *Options* akan mengatur volume musik dan efek suara (SFX).
 - c. *About Me* akan menampilkan informasi pembuat *game*.
 - d. *Quit* akan keluar dari aplikasi *game*.
5. Proses *gameplay*
 - a. Karakter akan bergerak otomatis ke arah kanan.

- b. Pemain dapat mengetuk layar untuk membuat karakter melompat. Pemain juga dapat mengetuk layar 2 kali untuk melompat lebih tinggi jika terdapat rintangan udara dan darat yang muncul secara bersamaan.
 - c. Hindari rintangan yang muncul secara acak untuk mempertahankan permainan.
 - d. Skor akan bertambah seiring jarak atau waktu permainan.
 - e. Latar akan berganti secara otomatis setiap skor bertambah 2000.
6. *Game over*
- a. Terjadi saat karakter menabrak rintangan.
 - b. Layar *game over* akan menampilkan skor akhir dan memberikan opsi tombol ulang dan tombol home.
7. Keluar dari *Game*
- a. Pilih tombol *quit* di menu utama, lalu konfirmasi pilihan keluar.

4. KESIMPULAN

Hasil pengujian menunjukkan bahwa navigasi menu, kontrol karakter, deteksi tabrakan, perhitungan skor, pergantian latar, dan efek visual berfungsi baik tanpa error pada perangkat Android versi 10 ke atas. *Game* ini berhasil memenuhi tujuan penelitian, yaitu menghasilkan *endless-runner* yang menarik secara visual dan responsif terhadap aksi pemain.

5. DAFTAR PUSTAKA

- [1] Hakim, K. F., Pasha, D., & Adrian, Q. J. (2022). Rancang bangun game platform 2D petualangan Si Gajah berbasis Android. Format: Jurnal Ilmiah Teknik Informatika, 11(2), 153–159. <https://doi.org/10.22441/format.2022.v11.i2.007>
- [2] Irpan, K., Amaranullah, P. S., Muhammad, D. O., Muhammad, R. A. A., Muhammad, S., & Yulianti. (2022). Penerapan metode waterfall dalam perencanaan sistem informasi penjualan buku berbasis aplikasi website (Studi kasus: Penjual Buku Toko 21 Jombang). Jurnal Teknologi Sistem Informasi dan Aplikasi, 5(4), 268–277. <https://openjournal.unpam.ac.id/index.php/JTSI/article/download/21378/pdf/66476>
- [3] Sayyid, M. U. F., Siva, F., Pahlevi, S. A., & Yaqin, M. A. (2023). Survei metode-metode software development life cycle dengan metode systematic literature review. ILKOMNIKA: Journal of Computer Science and Applied Informatics, 5(2), 123–135.

- <https://journal.unublitar.ac.id/ilkomnika/index.php/ilkomnika/article/download/447/132>
- [4] Ramadhan, H. A., Sari, R. P., & Prawira, D. (2023). Rancang bangun aplikasi sampah market menggunakan model Fountain. *Jurnal Informasi dan Teknologi*, 5(1), 1–10. <https://jidt.org/jidt/article/download/258/153/>
- [5] Ramadhan, H. A., Sari, R. P., & Prawira, D. (2023). Rancang bangun aplikasi sampah market menggunakan model Fountain. *Jurnal Informasi dan Teknologi*, 5(1), 1–10. <https://jidt.org/jidt/article/download/258/153/>
- [6] Siddik, S. M., & Wijoyo, A. (2023). Pembuatan aplikasi sarana prasarana berbasis web pada Yayasan Al-Hasanah menggunakan metode Fountain. *Jurnal Ilmu Komputer dan Pendidikan*, 2(1), 109–117. <https://journal.mediapublikasi.id/index.php/logic>
- [7] Wibowo, T., Cahyani, S. D., Hendar, I., & Rasyidhy, F. A. (2024). GAMED dan GDLC: Pembuatan video game untuk edukasi moral [GAMED and GDLC: The creation of video games for moral education]. *IJISWIRATAMA: Jurnal Ilmiah Teknologi dan Informasi*, 9(1), 1–12. <https://doi.org/10.36549/ijis.v9i1.299>
- [8] Ghani, M. A., Pohan, A. B., Gunawan, D., & Saputra, Y. (2024). Penerapan game edukasi 3D endless runner berbasis Android sebagai media belajar matematika anak [Application of 3D endless runner educational game based on Android as a children's mathematics learning media]. *Jurnal Ilmu Teknologi*, 7(1), 288–298. <https://doi.org/10.29408/jit.v7i1.24194>
- [9] Rahmayanti, N. (2021). Jurnal aplikasi game berbasis Android sebagai media pembelajaran untuk anak usia dini. *OSF Preprints*. <https://osf.io/preprints/b4hsz/>
- [10] Schell, J. (2019). *The art of game design: A book of lenses* (3rd ed.). CRC Press.
- [11] Habdi, & Supardi, R. (2021). Pembuatan game balap kelinci dengan Unity berbasis Android. *Jurnal Ilmiah Rekayasa dan Manajemen Sistem Informasi*, 7(1), 19–26. <https://doi.org/10.24014/rmsi.v7i1.10531>