

Perancangan Space Rock Pixel Game Antariksa Menggunakan *Game Maker Language*

Stevanus Oktoberrikho¹, Yansensius Reins Dima², Rendy³, Candra Gudiato⁴,
Christian Cahyaningtyas⁵

Program Studi Teknologi Informasi, Institut Shanti Bhuana, Bengkayang, Jl. Bukit Karmel, No. 1,
Bengkayang, Indonesia^{1,2,3,4,5}

stevanus@shantibhuana.ac.id¹, yansensius@shantibhuana.ac.id², rendy@shantibhuana.ac.id³,
candra.gudiato@shantibhuana.ac.id⁴, christi@shantibhuana.ac.id⁵

Abstrak – Perkembangan industri game digital telah mendorong munculnya berbagai genre permainan, termasuk game *Pixel art* yang menawarkan estetika retro dengan pengalaman bermain yang menarik. Penelitian ini membahas pengembangan *Space Rock*, sebuah game *Pixel art* bertema antariksa yang dibangun menggunakan *Game Maker Language* (GML). Game ini dirancang untuk memberikan pengalaman bermain yang menantang dengan berbagai *Level* dan musuh yang semakin sulit di setiap tahapannya. Metode pengembangan yang digunakan adalah metode *waterfall*, yang mencakup analisis kebutuhan, perancangan sistem, implementasi, pengujian, serta pemeliharaan dan pengembangan lanjutan. Hasil penelitian menunjukkan bahwa penggunaan GML memungkinkan pengembangan game 2D yang efisien dengan kontrol responsif serta mekanisme *Gameplay* yang adiktif. Pengujian menggunakan metode *Black-box testing* memastikan bahwa game berjalan sesuai dengan spesifikasi yang dirancang. Dengan demikian, *Space Rock* diharapkan dapat menjadi referensi bagi pengembang game *Pixel art* yang ingin menggunakan GML dalam pengembangan game mereka.

Kata Kunci – Game *Pixel art*, *Game Maker Language*, Pengembangan Game, Game Antariksa, *Waterfall Model*.

I. PENDAHULUAN

Perkembangan teknologi game di era digital saat ini semakin pesat, mendorong munculnya berbagai jenis game dengan beragam genre dan platform. Permainan video merupakan hal yang sering dimainkan untuk mengisi waktu luang dan dijadikan hiburan bagi masyarakat saat ini[1]. Pemanfaatan komputer di zaman modern ini semakin canggih, selain digunakan sebagai alat pengolah informasi belakangan komputer juga digunakan sebagai media hiburan yaitu game[2]. Salah satu genre yang menarik perhatian adalah game *Pixel art* yang menawarkan estetika retro dan sederhana namun tetap mampu menghadirkan pengalaman bermain yang seru dan menantang. Zaman sekarang genre yang menguasai pasaran adalah *Battle Royale* dan *MOBA*, hal ini menyebabkan pemain game bosan dengan game yang ada dan merasa tidak semangat dengan game-game yang akan datang karena

para pemain game sudah mengira-ngira genre game-nya pasti *Battle Royale* atau *MOBA*[3].

Dalam game development terdapat komponen yang dapat membangun kualitas dan media penyampaian gagasan game yang dikonsep, yaitu *Game Asset*. Sebuah game memiliki asset yang berhubungan dengan karakter, objek, *environment*, dan musik.[4] Perkembangan dunia game sekarang ini, lebih cenderung pada permainan bersifat game mobile[5]. Jurnal ini membahas pengembangan game *Pixel art* bertema antariksa berjudul "*Space Rock*", yang dibangun menggunakan *Game Maker Language*. Untuk membangun game yang telah dirancang dibutuhkan suatu game engine yang sesuai dengan genre dari game yang akan dikembangkan. [6] Pilihan tema antariksa dipilih karena popularitasnya yang tinggi dan potensi eksplorasi *Gameplay* yang luas, mulai dari eksplorasi planet hingga pertempuran antarbintang. Penggunaan *Game Maker Language* didasarkan pada kemudahan penggunaan dan kemampuannya dalam

menghasilkan game 2D yang efisien, cocok untuk pengembangan game *Pixel art* yang relatif sederhana namun tetap berkualitas.

"Space Rock" merupakan permainan yang menawarkan berbagai *Level* menantang dan beragam musuh yang harus dihadapi oleh pemain. Setiap *Level* diisi dengan rintangan dan tantangan yang semakin sulit, sehingga pemain harus menggunakan strategi dan keahlian untuk bisa melanjutkan ke *Level* berikutnya. Usia anak banyak menyukai game petualangan dan strategi. realita bahwa anak – anak lebih menyukai video game daripada belajar membuat situasi ini menjadi lebih kompleks[7]. Di mana tiap permainan memiliki target atau misi yang harus diselesaikan untuk bisa lanjut ke tahap berikutnya. Misi yang diberikan juga tak jarang diberi batasan waktu, sehingga ia perlu menggunakan strategi tertentu agar bisa menyelesaiannya dengan baik[8]. Dengan kontrol yang responsif dan *Gameplay* yang adiktif, Selain itu dapat juga digunakan media bermain seperti video game untuk melatih perilaku hidup bersih dan sehat pada anak-anak.[9] "Space Rock" berhasil menciptakan pengalaman bermain yang menarik dan menghibur bagi para penggemar game retro. *Framework* ini dapat digunakan disemua jenis video gim, sehingga terkadang banyak aspek yang terlewat dalam perancangannya[10]. Para pemain akan merasakan sensasi petualangan antariksa yang seru dan mendebarkan, sambil menikmati desain visual yang memukau dan *soundtrack* yang mengasyikkan. Dengan setiap *Level* yang berhasil dilewati, pemain merasa semakin tertantang dan terpacu untuk terus maju.

Tujuan dari pengembangan game "Space Rock" ini adalah untuk menghasilkan sebuah game *Pixel art* antariksa yang fun, menantang, dan mudah dimainkan. Game ini dirancang untuk memberikan pengalaman bermain yang menghibur bagi pemain dengan mekanisme *Gameplay* yang sederhana namun tetap adiktif. Namun, beberapa pemain mungkin merasa bosan dengan *Gameplay* yang monoton dan kurangnya variasi dalam tantangan yang diberikan setiap *Level*. Oleh karena itu, penulis berharap dapat memberikan solusi atas kekurangan ini dengan menghadirkan pembaruan dalam hal variasi *Gameplay* agar para pemain tetap tertarik dan tidak merasa bosan.

Harapan dari penulisan jurnal ini adalah untuk memberikan kontribusi dalam dokumentasi pengembangan game *Pixel art* menggunakan *Game Maker Language*. Jurnal ini diharapkan dapat menjadi referensi bagi para pemula yang ingin belajar mengembangkan game 2D dengan *Game Maker Language*, khususnya dalam genre *Pixel art*. Selain itu, pengembang juga perlu memperhatikan umpan balik dari para pemain untuk terus meningkatkan kualitas permainan. Meskipun desain visual dan *soundtrack* yang memukau tidak selalu cukup untuk mengimbangi kekurangan *Gameplay*, pengembang dapat menambahkan mode permainan baru yang lebih menantang, seperti mode survival dengan *Level-Level* yang semakin sulit.

Meskipun telah diupayakan semaksimal mungkin, pengembangan game "Space Rock" dan penulisan jurnal ini memiliki beberapa kekurangan. Salah satu kekurangannya adalah ruang lingkup game yang masih terbatas, dengan fitur-fitur yang belum terlalu kompleks. Hal ini disebabkan oleh keterbatasan waktu dan sumber daya yang tersedia. Namun, bagi sebagian pemain, tantangan yang semakin sulit di setiap *Level* justru menjadi daya tarik tersendiri. Mereka merasa bahwa perjuangan untuk mengalahkan setiap *Level* memberikan kepuasan yang luar biasa.

Dengan adanya variasi *Gameplay* yang menarik, diharapkan para pemain akan terus merasa tertantang dan terhibur. Pengembang perlu terus menghadirkan tantangan baru dan inovasi dalam permainan agar para pemain tidak merasa monoton. Namun, terlalu banyak penambahan fitur dan mode permainan baru juga dapat membuat permainan terlalu kompleks dan sulit dipahami oleh pemain baru. Oleh karena itu, pengembang perlu melakukan uji coba dan penyesuaian secara terus-menerus untuk memastikan bahwa permainan tetap menarik tanpa kehilangan daya tariknya. Dengan perhatian yang terus-menerus terhadap kualitas permainan dan respons terhadap umpan balik pemain, pengembang dapat menciptakan pengalaman bermain yang menyenangkan dan memuaskan bagi para pemain. Dengan demikian, diharapkan permainan ini dapat terus berkembang dan menarik minat pemain dari berbagai kalangan.

II. TINJAUAN PUSTAKA

A. Game Pixel Art dan Estetika Retro

Genre Pixel art telah mengalami kebangkitan popularitas baru-baru ini, tidak hanya karena estetika retro-nya yang unik, tetapi juga karena kemampuannya untuk membangkitkan nostalgia dan daya tarik emosional bagi pemain. Keterbatasan teknis Pixel art justru menjadi kekuatannya, memaksa para desainer untuk lebih kreatif dan efisien dalam menyampaikan informasi visual.

Penggunaan Pixel art dalam game indie tidak hanya membangkitkan nostalgia tetapi juga memungkinkan pengembangan yang efisien dengan sumber daya terbatas. Seperti pada penelitian [11] Dianta (2016) dalam Game Kluyuran di Kediri, desain Pixel art 2D dipilih untuk menciptakan visual yang menarik bagi anak-anak, dengan latar belakang menampilkan lingkungan Kota Kediri. Pendekatan ini relevan dengan Space Rock yang mengadopsi estetika retro untuk tema antariksa, memastikan daya tarik visual tanpa memerlukan grafis kompleks.

B. Game Maker Language untuk Pengembangan Game 2D

Game Maker Language (GML) dipilih sebagai engine pengembangan "Space Rock" karena kemudahan penggunaannya dan kemampuannya untuk menghasilkan game 2D yang efisien. GML memungkinkan pengembang untuk fokus pada aspek desain dan Gameplay tanpa terbebani oleh kompleksitas pemrograman tingkat tinggi[12]. Fleksibilitas GML juga memungkinkan implementasi berbagai fitur dan mekanisme Gameplay yang kompleks, sesuai dengan kebutuhan "Space Rock".

Kemampuan GML dalam menangani berbagai aspek pengembangan game, seperti grafis, audio, dan fisika, menjadikannya pilihan yang tepat untuk proyek ini. Menurut dokumentasi resmi YoYo Games[12], GML menyediakan fungsi-fungsi built-in untuk sprite management, collision detection, dan audio processing yang mempercepat proses development.

C. Metode Waterfall dalam Pengembangan Game

Metode waterfall merupakan model pengembangan perangkat lunak yang sistematis dan sequential[13]. Dalam konteks pengembangan game, metode ini cocok untuk proyek dengan ruang lingkup yang jelas dan kebutuhan yang stabil. Tahapan dalam metode waterfall meliputi analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan [14].

Penelitian oleh Pawar (2022) [13] menunjukkan bahwa metode waterfall efektif untuk pengembangan game indie dengan tim kecil dan *deadline* yang ketat, karena memberikan struktur yang jelas dan memudahkan *monitoring progress development*.

III. METODOLOGI PENELITIAN

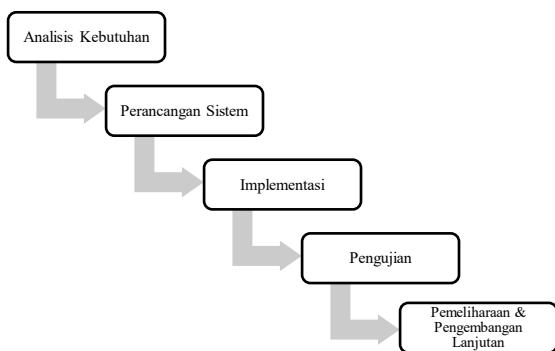
Penelitian ini menguraikan metode analisa yang digunakan dalam pengembangan *Space Rock*, menjelaskan variabel-variabel penelitian, serta memberikan gambaran sistem secara umum.

A. Model Pengembangan Waterfall

Model Waterfall merupakan pendekatan pengembangan perangkat lunak yang sistematis dan berurutan, dimana setiap tahapan harus diselesaikan sepenuhnya sebelum melanjutkan ke tahapan berikutnya. Metode ini disebut "*waterfall*" (air terjun) karena alur kerjanya yang mengalir satu arah ke bawah seperti air terjun, tanpa adanya proses iterasi atau pengulangan ke tahap sebelumnya.

Dalam pengembangan game Space Rock, metode *Waterfall* dipilih karena memberikan struktur yang jelas dan cocok untuk proyek dengan ruang lingkup yang telah terdefinisi dengan baik sejak awal. Metode ini memungkinkan tim development untuk fokus menyelesaikan satu tahapan secara tuntas sebelum beralih ke tahapan berikutnya, sehingga mengurangi risiko kesalahan dan memudahkan proses pengawasan.

Tahapan metode *Waterfall* yang diterapkan dalam pengembangan game Space Rock bisa dilihat pada Gambar 1.



Gambar 1. Metode *Waterfall* dalam Pengembangan Space Rock

Gambar 1 menunjukkan tahapan metode *Waterfall* yang digunakan dalam pengembangan *Space Rock*, meliputi analisis kebutuhan, perancangan sistem, implementasi, pengujian (*Black-box testing*), serta pemeliharaan dan pengembangan lanjutan.

1. Analisis Kebutuhan (*Requirements Analysis*)

Tahap ini merupakan fondasi dari seluruh proses pengembangan. Tim melakukan identifikasi dan dokumentasi kebutuhan fungsional dan non-fungsional game. Kebutuhan fungsional mencakup fitur-fitur yang harus ada dalam game, seperti mekanisme pergerakan pesawat, sistem penembakan, dan perhitungan skor. Sedangkan kebutuhan non-fungsional meliputi aspek performa, kompatibilitas, dan pengalaman pengguna.

2. Perancangan Sistem (*System Design*)

Berdasarkan analisis kebutuhan, tahap ini memfokuskan pada perancangan arsitektur game secara keseluruhan. Termasuk dalam tahap ini adalah pembuatan *flowchart* alur game, desain sprite karakter dan *environment*, perencanaan mekanisme *gameplay*, serta perancangan *user interface*. Semua desain didokumentasikan secara detail sebagai panduan untuk tahap implementasi.

3. Implementasi (*Implementation*)

Tahap implementasi merupakan proses penerjemahan desain menjadi kode program menggunakan Game Maker Language (GML). Pada tahap ini, semua komponen game seperti objek pesawat, asteroid, bullet, sistem *collision*, dan mekanisme scoring

diimplementasikan sesuai dengan spesifikasi yang telah dirancang.

4. Pengujian (*Testing*)

Setelah implementasi selesai, dilakukan pengujian menyeluruh untuk memastikan semua fitur berfungsi sesuai dengan spesifikasi. Metode *black-box testing* digunakan dengan fokus pada fungsionalitas game tanpa memperhatikan struktur kode internal. Pengujian mencakup aspek gameplay, kontrol, performa, dan stabilitas sistem.

5. Pemeliharaan dan Pengembangan Lanjutan

Tahap final dimana game siap dirilis dan dipublikasikan. Setelah deployment, dilakukan pemantauan dan pemeliharaan untuk menangani bug yang mungkin terlewat serta pengembangan fitur tambahan berdasarkan feedback pengguna.

B. Analisis Kebutuhan Sistem

a. Kebutuhan Fungsional

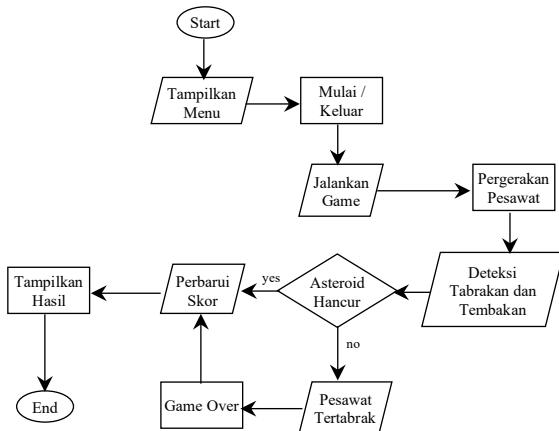
- Pemain dapat mengontrol pergerakan pesawat antariksa secara responsif
- Sistem penembakan otomatis terhadap asteroid dengan *cooldown mechanism*
- Mekanisme spawn asteroid yang progresif berdasarkan level
- Perhitungan skor *real-time* dengan sistem multiplier
- Deteksi tumbukan (collision detection) yang akurat

b. Kebutuhan Non-Fungsional

- Responsivitas kontrol $\leq 100\text{ms}$
- Frame rate konsisten 60 FPS pada hardware standar
- Kompatibilitas dengan Windows 10/11
- Ukuran file executable $< 100\text{MB}$
- Loading time < 5 detik

C. Perancangan Sistem

Flowchart digunakan untuk menggambarkan alur permainan secara sistematis agar memudahkan pemahaman tentang bagaimana game berjalan dari awal hingga akhir.



Gambar 2. Flowchart Diagram Game Space Rock

Gambar 2 menampilkan Flowchart yang menggambarkan alur permainan *Space Rock* secara sistematis, mulai dari inisialisasi game, menu utama, proses *Gameplay* (pergerakan pemain, penembakan, penghindaran asteroid), hingga kondisi kemenangan atau kekalahan. Diagram ini membantu pemahaman tentang struktur logika permainan.

D. Metode Pengambilan Data

Pengumpulan data dilakukan untuk memperoleh informasi yang relevan dalam pengembangan game *Space Rock*. Metode yang digunakan meliputi:

1. Studi Pustaka

Studi pustaka dilakukan dengan menelusuri berbagai literatur terkait pengembangan game 2D, mekanisme permainan, teori desain *Pixel art*, serta teknik pemrograman menggunakan *Game Maker Language* (GML).

2. Observasi

Observasi dilakukan terhadap berbagai game dengan genre serupa untuk memahami pola permainan, desain *Level*, tantangan yang diberikan, serta mekanisme kontrol yang digunakan. Observasi ini membantu dalam merancang *Gameplay* yang menarik dan adiktif bagi pemain.

IV. IMPLEMENTASI DAN HASIL

A. Persiapan Project dan Asset

- Membuat project baru di Game Maker Studio 2
- Membuat *folder structure* yang terorganisir (Sprites, Objects, Rooms, Scripts)
- Import sprite assets (pesawat, asteroid, bullet, background)
- *Setup room* dengan ukuran 800x600 pixels

B. Implementasi Objek Pesawat Antariksa

Game *Space Rock* adalah permainan bertema luar angkasa dengan konsep *Pixel art*, di mana pemain mengendalikan pesawat luar angkasa untuk menghindari rintangan dan menyerang musuh. Berikut adalah beberapa komponen utama sistem:

1. Gameplay:

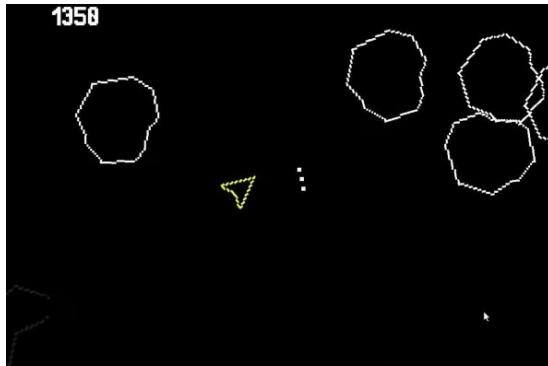
- Pemain mengontrol pesawat luar angkasa untuk bertahan dari gelombang besar Asteroid.
- Asteroid memiliki pola Kemunculan yang beragam dengan tingkat kesulitan yang meningkat.
- Pemain mengumpulkan poin berdasarkan jumlah asteroid yang dihancurkan dan mendapatkan skor tertinggi.

GML:

```

// Shooting System -
obj_spaceship
shoot_cooldown = max(0,
shoot_cooldown - 1);

if (keyboard_check(vk_space) &&
shoot_cooldown == 0) {
    var bullet =
instance_create_layer(x, y,
"Bullets", obj_bullet);
    bullet.direction = 0; // Shoot upward
    bullet.speed = 8;
    audio_play_sound(snd_laser,
1, false);
    shoot_cooldown = 15; // Cooldown frames
}
  
```



Gambar 3 Gameplay

Pada Gambar 3 memperlihatkan *Gameplay* pemain yang mengendalikan pesawat antariksa untuk menghindari atau menembak asteroid. Terlihat elemen visual *Pixel art*, antarmuka skor, dan pola pergerakan meteor yang semakin menantang.

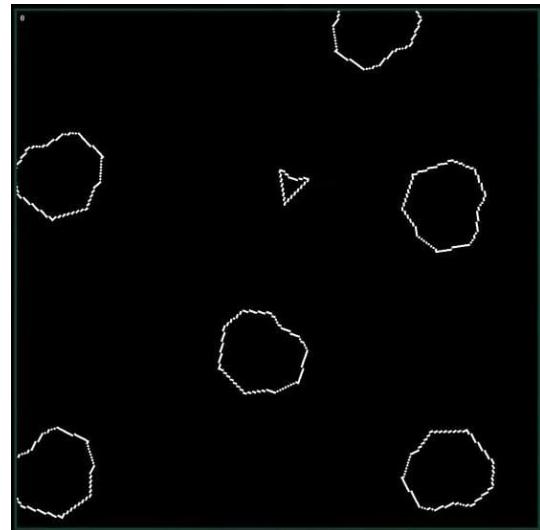
2. Kontrol Pemain:

- Pemain dapat menggerakkan pesawat ke kiri atau kanan serta menembak untuk menyerang musuh.
- Kontrol yang responsif menjadi faktor utama untuk pengalaman bermain yang optimal.

GML :

```
// Player Movement Script -  
obj_spaceship  
key_left =  
keyboard_check(vk_left);  
key_right =  
keyboard_check(vk_right);  
key_up = keyboard_check(vk_up);  
key_down =  
keyboard_check(vk_down);  
  
// Horizontal movement with  
boundary check  
if (key_left && x > 50) {  
    x -= move_speed;  
    sprite_index =  
spr_ship_left;  
} else if (key_right && x <  
room_width - 50) {  
    x += move_speed;  
    sprite_index =  
spr_ship_right;  
} else {  
    sprite_index =  
spr_ship_idle;  
}
```

```
// Vertical movement with  
boundary check  
if (key_up && y > 50) {  
    y -= move_speed;  
} else if (key_down && y <  
room_height - 50) {  
    y += move_speed;  
}
```



Gambar 4 Kontrol Pergerakan

Gambar 4 memperlihatkan ilustrasi kontrol pemain menggunakan tombol keyboard (kiri/kanan untuk bergerak, spasi untuk menembak). Desain ini memastikan responsivitas dan kemudahan penggunaan.

3. Sistem Skor dan Tantangan:

- Pemain mendapatkan skor berdasarkan performa mereka dalam permainan.
- Sistem tantangan diterapkan untuk meningkatkan daya tarik permainan.

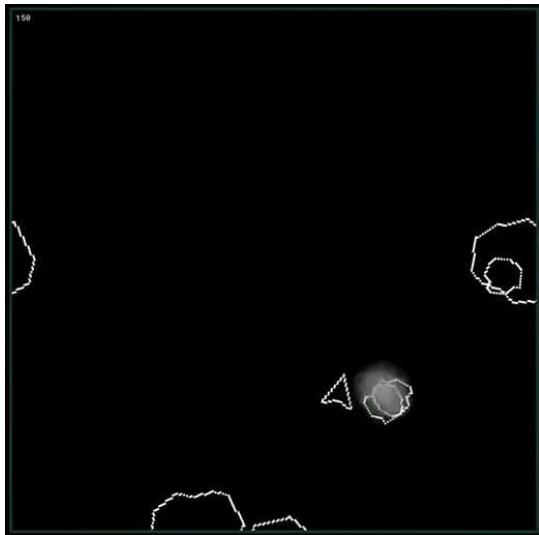
GML:

```
// Collision Detection -  
obj_bullet with obj_asteroid  
if (place_meeting(x, y,  
obj_asteroid)) {  
    with (other) {  
        hp -= 1;  
        if (hp <= 0) {  
            // Score calculation  
            global.score += 100  
* global.level;  
            instance_destroy();  
            // Explosion effect
```

```

        instance_create_layer(x, y,
        "Effects", obj_exlosion);
    }
    instance_destroy();
}

```



Gambar 5 Tampilan Skor

Tampilan skor yang dinamis berdasarkan jumlah asteroid yang dihancurkan. Sistem ini dilengkapi dengan peningkatan kesulitan di setiap *Level* untuk menjaga keterlibatan pemain seperti pada Gambar 5. Dengan perancangan sistem yang matang, *Space Rock* diharapkan mampu memberikan pengalaman bermain yang seru dan menantang bagi para pemain, serta menjadi referensi bagi pengembangan game serupa menggunakan *Game Maker Language*.

C. Hasil Pengujian Black-box

Hasil pengembangan *Space Rock* menunjukkan bahwa penggunaan *Game Maker Language* (GML) efektif untuk menciptakan game 2D dengan mekanisme *Gameplay* yang adiktif dan kontrol responsif.

Kelebihan:

- Pergerakan pesawat dan sistem penembakan.
- Pola spawn asteroid yang progresif.
- Perhitungan skor yang akurat.

Keterbatasan:

- Variasi musuh dan tantangan masih terbatas.

- Fitur tambahan seperti *power-up* belum diimplementasikan.

D. Hasil Implementasi

Setelah menyelesaikan semua step implementasi, game *Space Rock* berhasil dibangun dengan fitur-fitur berikut:

1. Kontrol yang Responsif: Pergerakan pesawat yang smooth dengan batasan area layar
2. Sistem Penembakan: Mekanisme shoot dengan cooldown yang seimbang
3. *Spawn Asteroid Dinamis*: Pola spawn yang semakin challenging seiring naik level
4. *Collision System*: Deteksi tumbukan yang akurat antara bullet-asteroid dan pesawat-asteroid
5. *Progression System*: Peningkatan kesulitan yang gradual berdasarkan level
6. *Scoring System*: Perhitungan score real-time dengan bonus level

Dengan perancangan sistem yang matang, *Space Rock* diharapkan mampu memberikan pengalaman bermain yang seru dan menantang bagi para pemain, serta menjadi referensi bagi pengembangan game serupa menggunakan *Game Maker Language*.

V. KESIMPULAN

Penelitian ini telah membahas perancangan dan pengembangan game *Space Rock*, sebuah game *Pixel art* bertema antariksa yang dibangun menggunakan *Game Maker Language* (GML). Proses pengembangan dilakukan dengan metode *waterfall*, yang mencakup tahapan analisis kebutuhan, perancangan, implementasi, pengujian, serta pemeliharaan dan pengembangan lanjutan. Hasil penelitian menunjukkan bahwa penggunaan GML memungkinkan pengembangan game 2D yang efisien dengan kontrol yang responsif serta mekanisme *Gameplay* yang adiktif. Game *Space Rock* menawarkan pengalaman bermain yang menantang dengan tingkat kesulitan yang meningkat di setiap *Levelnya*. Pengujian menggunakan metode *Black-box testing* memastikan bahwa game berjalan sesuai dengan spesifikasi yang dirancang, memberikan pengalaman bermain yang optimal bagi pengguna. Meskipun demikian,

terdapat beberapa keterbatasan dalam pengembangan game ini, seperti variasi tantangan yang masih dapat ditingkatkan dan fitur tambahan yang dapat memperkaya *gameplay*. Ke depannya, pengembangan lebih lanjut dapat mencakup penambahan mode permainan baru, peningkatan kecerdasan buatan musuh, serta optimalisasi performa untuk meningkatkan pengalaman bermain. Dengan adanya inovasi yang berkelanjutan dan respons terhadap umpan balik pemain, diharapkan *Space Rock* dapat terus berkembang dan menjadi referensi bagi pengembang game *Pixel art* lainnya yang ingin menggunakan *Game Maker Language*.

REFERENSI

- [1] H. Bima Dirgantara and P. Windriyani, “Pembangunan Permainan Video Action-Shooter 2 Dimensi Berbasis Desktop Menggunakan Game Maker Studio 1.4,” *KALBISCIENTIA J. Sains dan Teknol.*, vol. 7, no. 2, p. 116, 2020, doi: 10.53008/kalbiscientia.v7i2.95.
- [2] J. Wahyudi, “Rancang Bangun Game Edukasi 3 Dimensi Perawatan Orang Utan Berbasis Game Maker: Studio,” *Bitnet J. Pendidik. Teknol. Inf.*, vol. 6, no. 1, pp. 24–32, 2021, doi: 10.33084/bitnet.v6i1.1788.
- [3] M. R. Ikhsani, H. A. Chandra, and D. Agustini, “Game Teka-Teki Harta Firaun Menggunakan Role Playing Game Maker Mv,” *J. Tek. Inform. Teknol. Inf.*, vol. 1, no. 1, p. 2, 2021.
- [4] U. Nugroho Riyanto and F. Persada, “Perancangan Asset Game Carborundum,” *Visualideas*, vol. 3, no. 1, pp. 1–8, 2023, doi: 10.33197/visualideas.vol3.iss1.2023.1104.
- [5] A. Agustin, A. Evel, S. Susanti, and R. Rahmaddeni, “Implementasi Metode Finite State Machine pada Permainan Tradisional Setatak Berbasis Android,” *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 8, no. 2, pp. 738–751, 2021, doi: 10.35957/jatisi.v8i2.580.
- [6] I. A. Fadila, I. A. Putra, and D. A. Prambudi, “Perancangan Game Kenangan Balikpapan Dengan RPG Maker Menggunakan Metode Game [7] Development Life Cycle,” *Poros Tek.*, vol. 15, no. 1, pp. 22–29, 2023, doi: 10.31961/porosteknik.v15i1.1953.
- [8] I. F. Anshori, S. Ayuni Kaffah, N. Supa, and R. Rizal Setiawan Mail, “Perancangan Game Edukasi Pengenalan Bahasa Pemrograman Menggunakan Construct 2,” *Infotek J. Inform. dan Teknol.*, vol. 5, no. 2, pp. 275–286, 2022, doi: 10.29408/jit.v5i2.5803.
- [9] M. B. Hartanto and Y. Yuniarthe, “Membangun Aplikasi Game Petualangan 2D Pesawat Ufo Berbasis Android,” *J. Teknol. dan Inform.*, vol. 1, no. 1, pp. 1–11, 2020, doi: 10.57084/jeda.v1i1.963.
- [10] M. Lubis, K. Khairunnisa, and Y. F. A. Lubis, “Perancangan Game Ayo Bersih Untuk Anak Menggunakan Metode System Development Life Cycle Waterfall,” *J. Tek. Inf. dan Komput.*, vol. 5, no. 2, p. 393, 2022, doi: 10.37600/tekinkom.v5i2.643.
- [11] D. Nadhifah, E. Muhammad, A. Jonemaro, and W. S. Wardhono, “Penerapan Design, Dynamics, Experience (DDE) Framework Pada Perancangan Gim Bermain Peran Karma,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 3, pp. 776–781, 2020.
- [12] Ashafidz Fauzan Dianta, “Pembuatan Game Kluyuran di Kediri Berbasis Android Dan C#,” *J. Inform. dan Multimed.*, vol. 8, no. 1, pp. 35–39, 2016, doi: 10.33795/jim.v8i1.1150.
- [13] YoYo Games, “Game Maker Studio 2 Documentation: Game Maker Language Reference.” [Online]. Available: <https://manual.gamemaker.io/monthly/en/#t=Content.htm>
- R. P. Pawar, “A Comparative study of Agile Software Development Methodology and traditional waterfall model A Comparative study of Agile Software Development Methodology and traditional waterfall model,” no. December 2015, 2022.