

# Pengembangan Game Platformer Slime Jump Menggunakan Metode Sekuensial Linier

Aneu Yulianeu<sup>[1]\*</sup>, Adam Abdul Aziz Kurniawan Rosadi<sup>[2]</sup>

Program Studi: Manajemen Informatika<sup>[1]</sup>, Teknik Informatika<sup>[2]</sup>

STMIK DCI

Jln. Sutisna Senjaya No. 158A, Kota Tasikmalaya, 46112, Indonesia

aneu@stmik-dci.ac.id<sup>[1]</sup>.

**Abstract**— *The purpose of this research is to create a Slime Jump platformer game using the linear sequential method that is robust and informative without sacrificing the main function of the game, which is to provide entertainment and fun. The method used in this research is the linear sequential or waterfall method which has main stages that reflect basic development activities, namely requirement analysis and definition, system and software design, implementation and unit testing, integration and system testing, and operation and maintenance. Based on the results of functionality testing using black box testing, the Slime Jump platformer game has run well. Based on the results of gameplay testing, it can be concluded that the Slime Jump platformer game that has been developed using the linear sequential method is quite entertaining, interactive, and informative.*

**Keywords**— *Game Platformer, Slime Jump, Linear Sequential.*

**Abstrak**— Tujuan penelitian ini adalah untuk membuat game platformer Slime Jump dengan menggunakan metode linear sequential yang tangguh dan informatif tanpa mengorbankan fungsi utama game tersebut, yaitu untuk memberikan hiburan dan kesenangan. Metode yang digunakan pada penelitian ini adalah metode sekuensial linier atau waterfall yang memiliki tahapan utama yang mencerminkan aktifitas pengembangan dasar, yaitu requirement analysis and definition, system and software design, implementation and unit testing, integration and system testing, dan operation and maintenance. Berdasarkan hasil uji fungsionalitas menggunakan black box testing, game platform Slime Jump bekerja dengan baik. Berdasarkan hasil gameplay testing dapat disimpulkan bahwa game platformer Slime Jump yang dikembangkan dengan metode sekuensial linier cukup menghibur, interaktif, dan informatif.

**Kata Kunci**— *Game Platformer, Slime Jump, Sekuensial Linier.*

## I. PENDAHULUAN

Potensi game komputer sebagai alat untuk penelitian *artificial intelligence* terus berkembang [1]. Game biasanya menyenangkan dan hiburan ringan, tetapi kadang-kadang menghadirkan tugas menantang yang memungkinkan pemain mempelajari metode kontrol interaktif melalui simulasi pertempuran melawan musuh. [2]. Game menghibur dan menghabiskan waktu bagi orang-orang dari segala usia. [3]. Game meningkatkan pertumbuhan dan konsentrasi otak [4]. Pemain game membutuhkan pemikiran cepat dan kemampuan

memecahkan masalah agar berhasil dan menyelesaikan tantangan tepat waktu [5]–[8].

Para pemain game akan terdorong untuk berpikir kritis ketika dihadapkan pada permasalahan yang menuntut mereka untuk tampil dengan solusi kreatif [9]. Dalam sebuah permainan tentunya pemain akan menghadapi berbagai macam rintangan yang harus diselesaikan agar pemain dapat memenangkan permainan tersebut. Bermain game juga dapat membantu manusia meningkatkan keterampilannya, yang merupakan cara lain dimana game dapat bermanfaat [10].

Game yang dibuat *slime jump*, game yang berjenis *platformer* dengan grafik 2 dimensi (2D). *Game platformer* merupakan sebuah *genre game* dan *sub-genre* dari game aksi [11]–[13]. Pemain *game platformer* mengendalikan karakter dan membimbing melalui permainan dengan meminta untuk menghindari bahaya dan melompati *platform* yang menggantung [13]. Super Mario Bros merupakan contoh game yang masuk dalam kategori *game platformer* [14].

Dalam *game platformer*, pemain mengendalikan protagonis, yang harus menggunakan *platform game* untuk menavigasi lingkungan berbahaya dengan aman termasuk tebing, jebakan, dan objek penghalang [15]. *Game platform* biasanya menampilkan karakter utama, musuh, dan elemen game lainnya dari sisi layar. Melompat adalah mekanisme umum dalam *game platform*, dan sering digunakan untuk mengalahkan musuh atau memajukan cerita karakter utama [16].

Secara umum dan matematika, *game 2D* hanya memiliki 2 sumbu, yaitu sumbu x dan y. Sehingga kamera dapat fokus langsung ke game dan pemain dapat terlihat [17]. *Game slime jump* menerapkan metode sekuensial linier (*waterfall method*). Pengembangan sistem dalam metode sekuensial linier melibatkan analisis perangkat lunak, desain, pengkodean, dan pengujian dalam urutan tersebut [18]. *Game slime jump* adalah hasil sintesis elemen kreatif dan pengalaman, serta elemen teknis, seperti penggunaan beberapa perpustakaan terbuka sebagai landasan pengetahuan dan desain antarmuka pengguna grafis 2D dengan antarmuka yang menyenangkan. *Slime jump* merupakan *game platform 2D* orisinal yang menekankan tantangan di setiap level (*stage*). Nilai jual khas game ini adalah mode cerita, yang dapat diakses setelah sejumlah *stage*

diselesaikan. Sebenarnya, ada dua kemungkinan hasil dari cerita *game* tersebut. Meskipun antarmuka dan kontrol langsung dari *game slime jump* membuatnya mudah untuk dimainkan, hanya sedikit pemain yang berhasil menyelesaikannya.

Mengingat konteks ini, penelitian ini bertujuan untuk mengembangkan *game platformer slime jump* yang menantang dan mendidik menggunakan pendekatan sekuensial linier, tanpa mengorbankan tujuan utama *game* untuk menyediakan hiburan yang menyenangkan.

## II. STUDI LITERATUR

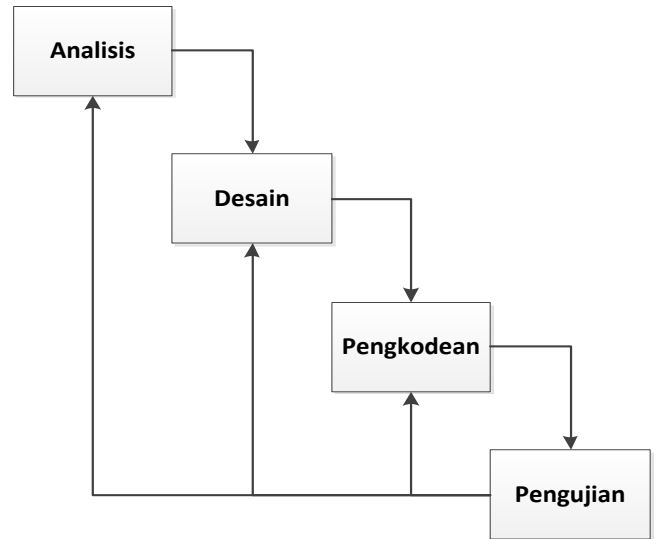
Penelitian ini merujuk beberapa penelitian, Penelitian Putra (2017), *game* balap kucing berbasis *android*, penelitian tersebut menciptakan sebuah *game* balapan 3D yang dapat dijalankan pada *platform android* [19]. *Game* ini memiliki karakter unik, dimana karakter yang digunakan pemain memiliki karakteristik manusia kerdil yang memiliki telinga dan ekor seperti kucing, dalam *game* ini juga pemain memiliki keterampilan yang dapat digunakan untuk meraih juara. *Game* ini menggunakan *Blender* sebagai aplikasi untuk desain karakter 3D dan *Unity* sebagai *engine game*. Penelitian yang dilakukan menunjukkan bahwa *game* ini merupakan cara unik untuk bersenang-senang karena tidak seperti kebanyakan *game* balapan yang biasanya berbentuk kendaraan. Memilih *platform android* juga memudahkan untuk memainkan *game* ini dimana saja dan kapan saja.

Penelitian kedua dilakukan Septiko (2018), *platformer game* edukasi cerita gajah mada menggunakan metode iteratif. Tujuan permainan ini adalah untuk mengajari anak-anak tentang cerita Gajah Mada. [20]. *Game* ini adalah *platformer* 2D, yang berarti pemain berpindah dari satu *platform* ke *platform* berikutnya untuk bermain. Cara memainkan *game* ini adalah melewati rintangan dan melawan musuh dalam waktu yang ditentukan. Selain itu, pemain juga memiliki hal-hal seperti nyawa, poin kesehatan, dan item.

Penelitian ketiga dilakukan oleh Yedithia (2022), *game platformer* ayo lawan virus. Penelitian tersebut bertujuan untuk membuat permainan yang dapat mengedukasi pemain untuk mengenal virus dan memperkenalkan pola hidup sehat [21]. Penelitian ini menggunakan metode sekuensial linier, yang memiliki beberapa langkah, yaitu analisis, desain, pengkodean, pengujian, implementasi, dan pemeliharaan. Langkah-langkah tersebut harus dilakukan secara berurutan dan tidak dapat dialihkan dengan langkah lain.

## III. METODE PENELITIAN

Metode yang digunakan pada penelitian ini adalah metode sekuensial linier yang memiliki tahapan utama yang mencerminkan aktifitas pengembangan dasar [22]. Sekuensial linier merupakan metode pengembangan sistem untuk membuat perangkat lunak yang dimulai dengan proses analisis perangkat lunak, dilanjutkan dengan desain, kemudian pengkodean dan diakhiri dengan pengujian perangkat lunak.

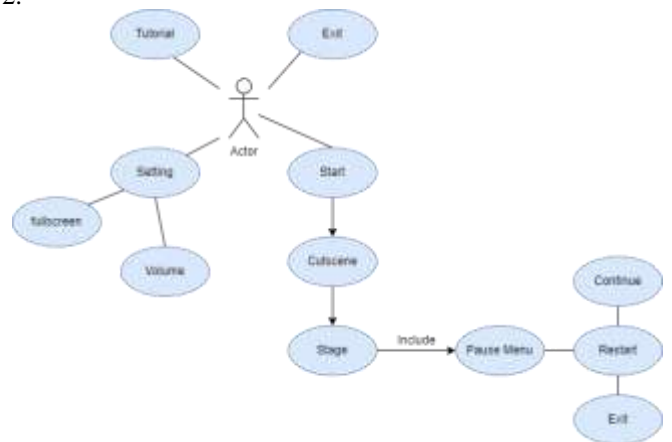


Gambar 1. Tahapan metode sekuensial linier

Berdasarkan Gambar 1 dapat diuraikan bahwa pada tahapan analisis meliputi analisis kebutuhan fungsional, analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras, dan analisis pengguna. Pada tahap desain dihasilkan desain *game*, UML, *flowchart*, dan *storyboard* berdasarkan hasil analisis pada tahap sebelumnya. Tahap pengkodean merupakan proses pembuatan program *game platformer Slime Jump* berdasarkan desain yang sudah dibuat. Setelah proses pengkodean selesai, maka dilakukan pengujian pada *game platformer Slime Jump* untuk mengetahui kesesuaian hasil analisis, desain, dan pengkodean.

### A. Use case diagram

Pengembangan akan lebih mudah dalam pembuatan *game slime jump*. Desain *use case diagram* dapat dilihat pada Gambar 2.



Gambar 2. Use case diagram game slime jump

Penjelasan Gambar 2 dapat dilihat pada Tabel 1.

Tabel 1. Penjelasan use case diagram

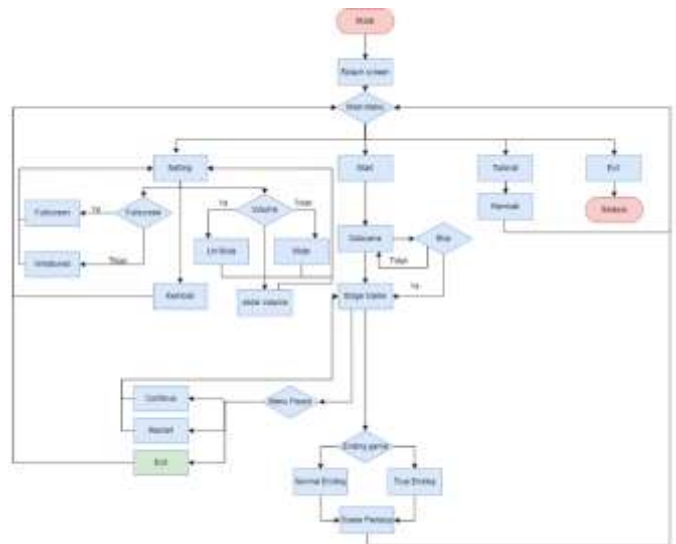
No	Use Case	Deskripsi
1	Start	Fitur memulai permainan.
2	Tutorial	Berguna untuk memberi petunjuk

		dasar dalam permainan.
3	Exit (Main Menu)	Berguna untuk keluar dari game.
4	Setting	Fitur yang berisi pengaturan game.
5	Fullscreen	Jika dihidupkan maka game akan menggunakan mode fullscreen, jika tidak maka game akan memasuki mode windowed.
6	Cutscene	Berisi story dari game.
7	Stage	Stage merupakan tempat dimana player memainkan game agar bisa melewati semua rintangan untuk membuka chapter berikutnya.
8	Pause Menu	Berfungsi untuk menghentikan game sementara dan disini player bisa memilih beberapa pilihan.
9	Continue	Digunakan untuk melanjutkan permainan.
10	Restart	Digunakan untuk memulai stage dari awal.
11	Exit (Pause Menu)	Digunakan untuk kembali ke menu utama.

#### IV. HASIL DAN PEMBAHASAN

##### A. Flowchart Sistem

Setelah game dijalankan, sistem akan memutar scene berupa splash screen sekaligus memutar file audio, sistem akan menajalakan object parent berupa audio manager dimana object ini akan dimainkan setelah scene splashscreen berputar dan object ini juga yang digunakan sistem untuk memutar audio yang dibutuhkan dalam game. Object audio manager bisa tetap diputar pada scene manapun karena sistem mendeteksi sebagai object yang tidak bisa dihilangkan kecuali game ditutup, hal ini bisa terjadi karena object audio managaer menggunakan perintah DontDestroyOnLoad. Setelah itu sistem akan masuk ke bagian menu utama. Pada menu utama terdapat beberapa pilihan diantaranya start, settings, tutorial, dan exit. Jika pemain memilih settings, pemain akan memasuki scene halaman settings yang akan membuat sistem berpindah dari scene menu utama ke scene setting, scene ini memiliki beberapa pilihan antara lain setting screen yang berguna untuk mengatur mode layar permainan apakah fullscreen atau windowed untuk tampilan bawaan game adalah fullscreen ketika player memilih option fullscreen, maka sistem akan mengubah tampilan menjadi mode windowed dan ketika di klik kembali, maka game akan kembali fullscreen. Pilihan kedua yaitu setting volume yang berfungsi untuk mengatur audio yang ada dalam game, selain pada volume setting juga memiliki pilihan berupa slider untuk mengatur volume suara yang dikeluarkan.



Gambar 3. Flowchart perancangan sistem

Jika player memilih tutorial, maka sistem akan memasuki scene tutorial. Jika player memilih exit, maka sistem akan menutup program, jika memilih start, maka sistem akan memasuki scene berupa video untuk cutscene story stage, jika pemain melakukan klik pada layar permainan, maka cutscene akan berhenti dan sistem akan memulai stage, jika tidak melakukan klik, maka cutscene akan terus berputar hingga habis dan akan memulai stage baru.

Ketika pemain menekan tombol pause, game akan berhenti dan memasuki mode pause, mode ini merupakan mode dimana sistem merubah timescale dari 1 yang berarti waktu berjalan menjadi 0 yang berarti waktu dalam game tersebut berhenti. Dalam menu pause terdapat beberapa pilihan untuk pemain. Ketika pemain memilih untuk pilihan continue permainan akan keluar dari mode pause, sistem merubah timescale dari 0 menjadi 1. Lalu pemain akan melanjutkan permainan dengan posisi dan kondisi permainan sama seperti sebelum melakukan pause.

Ketika pemain memilih restart, permainan akan keluar dari mode pause. Sistem merubah timescale dari 0 menjadi 1. Lalu pemain akan melanjutkan permainan dengan posisi dan kondisi permainan kembali ke posisi awal stage dimulai. Ketika pemain memilih exit, maka sistem merubah timescale dari 0 menjadi 1 pemain akan keluar dari mode pause dan sistem akan memindahkan tampilan ke menu utama. Pemain akan dinyatakan telah menyelesaikan suatu stage apabila object yang berupa player menyentuh object berupa finish line. Kemudian sistem akan berpindah ke scene berikutnya dengan cara +1 pada scene active.

Pada scene stage terdapat beberapa object lain selain ground biasa sebagai obstacle game, diantaranya adalah stone teleportation, object tersebut bisa digunakan ketika object player menyentuh object teleportation. Ketika pemain menggunakan tombol E, sistem akan memindahkan object player dari posisi player ke posisi tujuan stone teleportation tujuan yang sebelumnya object player sentuh,

Glue ground adalah object dimana object player tidak bisa menggunakan kemampuan melompat. Hal ini terjadi karena object player hanya bisa melompat ketika player menyentuh

object lain berupa *object ground*. Ketika *player* tidak menyentuh *object ground* mana kemampuan melompat akan hilang.

*Falling ground* adalah *object* berupa *ground* yang akan terjatuh ketika *object player* menyentuh *object* tersebut. *Time stage* adalah *object* yang berfungsi untuk melakukan *limit time* pada suatu *stage*, ketika memasuki *scene stage* sistem akan memulai hitung mundur dengan waktu yang telah ditentukan, ketika mencapai 0 maka sistem akan segera mengganti *scene stage* yang aktif ke *scene stage* lainnya, pergantian ke *scene stage* lain dilakukan dengan cara *random* sesuai dengan beberapa *stage* yang telah ditentukan. Ketika pemain telah menyelesaikan semua *stage*, pemain akan mendapatkan salah satu diantara 2 *ending* dalam *game*. Ketika *cutscene ending* selesai, sistem akan memasuki *scene* penutup pertanda akhir dari permainan. Ketika pemain menekan tombol kembali maka sistem akan memindahkan tampilan ke menu utama.

**B. Desain Aset Game**

Setelah diselesaikannya konsep *gameplay* dan fitur yang akan dihadirkan dalam *game slime jump* ini, maka dilanjutkan dengan pembuatan desain aset *game*. Penulis menggunakan Clip Studio untuk pembuatan karakter *game*, *background game*, dan *cutscene game*. Karakter *game* merupakan tokoh yang ada dalam *game slime jump*. Dalam *game slime jump* terdapat 2 karakter yaitu Neru sebagai karakter utama dan Mofu sebagai karakter pendukung.

Untuk *background* dan objek lainnya penulis menggunakan Corel Draw. Gambar yang digunakan pada *background* digunakan sebagai pelengkap sesuai dengan konsep yang telah dibuat.

**C. Desain Graphical User Interface (GUI)**

Tampilan grafis yang digunakan sebagai perantara antara pemain dan sistem. GUI yang digunakan pada *game slime jump*, yaitu tombol *start*, tombol menu pengaturan (*setting*), tombol halaman *tutorial*, tombol keluar (*exit*) dari *game*, tombol *mode screen*, tombol kontrol suara (*audio*), tombol *pause*, tombol ulang *game* (*restart*), dan tombol menu utama.

Selain aset *game* utama, perlu dibuat aset *game* pendukung yang dapat digunakan pada *game slime jump*. Aset pendukung tersebut diantaranya *flatformer stage* aset yang digunakan pada setiap *stage*, *finish* aset yang menjadi tujuan pemain (*player*) dalam menyelesaikan setiap *stage*, *timer stage* aset yang menunjukkan waktu tersisa untuk menyelesaikan *stage*, *background music* aset *audio* yang digunakan dalam *game slime jump*.

**D. Story Board Game Slime Jump**

Penjelasan dari alur *story board game slime jump* dapat dilihat pada Tabel 2.

Tabel 2. Story board game slime jump

Scene	Nama	Isi
1	Splash Screen	<ul style="list-style-type: none"> <li>• Logo Slime Jump</li> <li>• Logo Unity</li> <li>• Logo Team Project</li> </ul>
2	Menu Utama	<ul style="list-style-type: none"> <li>• Logo "Slime Jump"</li> </ul>

		<ul style="list-style-type: none"> <li>• Tombol <i>Start</i> yang berguna untuk memasuki <i>cutscene</i></li> <li>• Tombol <i>Setting</i> berguna untuk memasuki <i>setting</i>.</li> <li>• Tombol <i>Tutorial</i> untuk memasuki tampilan tutorial <i>game slime jump</i>.</li> <li>• Tombol <i>Exit</i> untuk keluar permainan</li> </ul>
3	Setting	<ul style="list-style-type: none"> <li>• Tombol <i>fullscreen</i> untuk mengatur <i>mode</i> layar</li> <li>• Tombol <i>volume</i> untuk mengatur <i>volume</i> suara.</li> </ul>
4	Tutorial	<ul style="list-style-type: none"> <li>• Gambar <i>background</i> tutorial.</li> <li>• Isi tutorial berisi cara bermain <i>game slime jump</i></li> <li>• Tombol kembali untuk kembali ke menu utama</li> </ul>
5	Cutscene	<ul style="list-style-type: none"> <li>• Berisi video <i>story game slime jump</i>.</li> <li>• Fitur skip yang dilakukan dengan cara klik layar didalam <i>game</i>.</li> </ul>
6	Stage	<ul style="list-style-type: none"> <li>• Objek <i>Slime</i> yang bernama Neru sebagai karakter utama <i>game</i></li> <li>• Objek <i>obstacle</i> adalah objek yang harus dilalui pemain untuk bisa mencapai gerbang.</li> <li>• Gerbang untuk menuju <i>stage</i> selanjutnya.</li> <li>• Tombol <i>pause</i> untuk mengakses <i>pause</i> menu dan menghentikan <i>game</i> sementara.</li> </ul>
7	Pause menu	<ul style="list-style-type: none"> <li>• Tombol <i>continue</i> berguna untuk melanjutkan permainan.</li> <li>• Tombol <i>restart</i> berguna untuk mengulangi <i>stage</i>.</li> <li>• Tombol <i>Exit</i> berguna untuk kembali ke menu utama.</li> </ul>
8	Penentu Ending Story	<ul style="list-style-type: none"> <li>• Video <i>ending story game</i>.</li> <li>• <i>Hidden Button</i>.</li> </ul>
9	Ending Story	<ul style="list-style-type: none"> <li>• Berisi Video <i>ending</i> dari <i>story game slime jump</i>.</li> </ul>
10	Scene penutup	<ul style="list-style-type: none"> <li>• Berisi peran untuk <i>player</i>.</li> </ul>

**E. Implementasi**

Pada tahap implementasi pembuatan *game slime jump* dibagi menjadi 3 bagian, yaitu implementasi kode program, implementasi tampilan dan desain *game*.

1) Implementasi Kode Program

Pada bagian ini akan dijelaskan mengenai implementasi *coding* dari *game slime jump*. *Code* yang dicantumkan merupakan kode-kode penting dalam pembuatan *game slime jump* ini.



Gambar 4. Source code game manager

Penjelasan dari *source code game manager* pada Gambar 4 adalah:

- a) 6-9: Untuk mengganti musik yang di *scene* yang sedang berjalan sesuai dengan isi dari *audio manager*.
- b) 10-13: Memuat ulang *scene* yang sedang berjalan.
- c) 15-19: Memuat *scene* berdasarkan nama *scene* dalam bentuk *string*.
- d) 20-23: Memuat *scene* berdasarkan urutan angka dari *scene* yang di masukan di *build settings* dalam bentuk *integer*.
- e) 25-28: Untuk menghentikan *game*.



Gambar 5. Source code player movement

Penjelasan dari *source code player movement* pada Gambar 5 adalah:

- a) 14-17: Untuk mendapatkan komponen dari objek yang diberi *rigidbody 2D*.
- b) 22: Menentukan *input player* secara horizontal. Jika menekan a atau panah kiri, maka akan menghasilkan *input* angka -1, dan jika menekan d atau panah kanan, maka akan menghasilkan *input* angka 1.

- c) 23: Menentukan kecepatan *player* secara horizontal sesuai dengan nilai *movement speed* yang dimasukan.
- d) 25: Mengecek status *game* pada *game pause manager*. Ketika *gameispause = false*, maka *playermovement* bisa dijalankan, ketika *gameispause = true*, maka *playermovement* tidak bisa digunakan, karena *game* dalam posisi *pause*.
- e) 27-30: Untuk mengubah *sprite player* sesuai dengan arah *input*. Jika nilai horizontal 1 maka posisi *sprite* akan sama dengan *scale* yang telah ditentukan, jika nilai -1 maka posisi *sprite player* akan di ubah menjadi vektor 2 baru.
- f) 32-42: Jika *input space* dan kondisi *gameispause = false*, nilai *vector2.up* akan dikali dengan nilai *jumpfore* sehingga *object player* akan melambung keatas dengan kondisi dimana *object player* berada pada posisi *IsGrounded*.



Gambar 6. Source code finish

Penjelasan dari *source code finish* pada Gambar 6 adalah:

- a) 8-11: Untuk mengganti musik pada *scene* yang sedang berjalan sesuai dengan isi dari *audio manager*.
- b) 12-18: Ketika objek yang memiliki tag "Player" menyentuh objek *finish*, maka akan dilakukan *loadscene* berdasarkan *scene* yang aktif lalu ditambahkan nilainya 1 sehingga menuju *scene* berikutnya sesuai dengan urutan yang ada pada *build settings*.



Gambar 7. Source code game manager pause

Penjelasan dari *source code game manager pause* pada Gambar 7 adalah:

- a) 8: Untuk memberi status awal *GameIsPause* adalah *false* yang artinya *game* tidak dalam keadaan *pause*.

- b) 11-15: Membuat fungsi untuk tombol *continue* dengan cara merubah status *GameIsPause* menjadi *false* dan mengubah *timescale* menjadi 1 yang berarti waktu berjalan kembali.
- c) 18-23: Membuat fungsi untuk tombol *pause* dengan cara merubah status *GameIsPause* menjadi *true* dan mengubah *timescale* menjadi 0 yang berarti waktu berhenti.
- d) 25-31: Membuat fungsi untuk tombol *restart* dengan cara merubah status *GameIsPause* menjadi *false* dan mengubah *timescale* menjadi 1 yang berarti waktu berjalan kembali dan memulai dari awal *scene active* dengan memuat *scene* yang sedang berjalan.
- e) 33-38: Membuat fungsi untuk tombol *exit* dengan cara merubah status *GameIsPause* menjadi *false* dan mengubah *timescale* menjadi 1 yang berarti waktu berjalan kembali dan memuat *scene* menu utama.

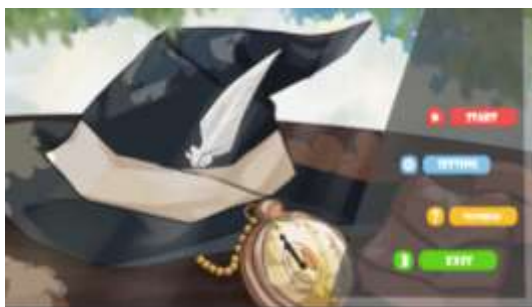
2) Implementasi Tampilan dan Desain Game

Implementasi tampilan *splash screen*. Halaman yang terbuka pertama kali sejak *game* dijalankan. Hasil implementasi tampilan *splash screen* dapat dilihat pada Gambar 8.



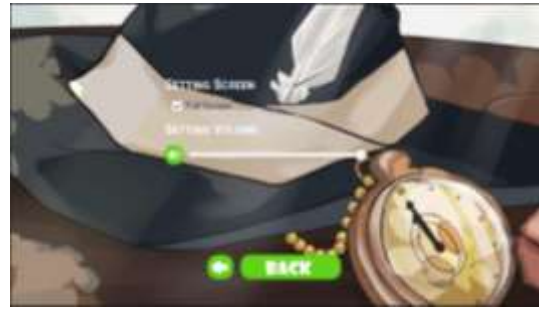
Gambar 8. Implementasi tampilan *splash screen*

Implementasi tampilan menu utama. Halaman yang muncul setelah *splash screen*. Menu utama menampilkan pilihan tombol *start*, *setting*, *tutorial*, dan *exit*. Hasil implementasi tampilan menu utama dapat dilihat pada Gambar 9.



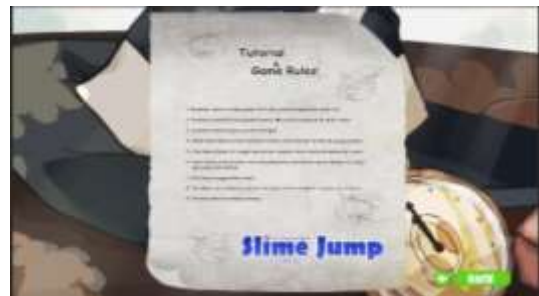
Gambar 9. Implementasi tampilan *splash screen*

Implementasi tampilan *settings*. Halaman yang menampilkan beberapa pilihan menu untuk mengatur *game*. *Setting screen option* berfungsi mengatur tampilan *game fullscreen* atau *windowed*, *setting volume option* yang berfungsi mengatur besar kecil suara, *volume slider*, *mute*, dan *unmute*, dan tombol *back* untuk kembali ke menu utama. Hasil implementasi tampilan *settings* tersebut dapat dilihat pada Gambar 10.



Gambar 10. Implementasi tampilan *settings*

Implementasi tampilan tutorial. Halaman yang berisikan tutorial dasar dan peraturan yang digunakan dalam bermain *game slime jump*. Hasil implementasi tampilan tutorial tersebut dapat dilihat pada Gambar 11.



Gambar 11. Implementasi tampilan tutorial

Implementasi tampilan *cutscene*. Halaman yang berisi potongan *scene* dari progres cerita yang ada dalam *game*. Disini pemain dapat melakukan *skip* cerita dengan cara melakukan klik dimanapun pada bagian layar *game*. Hasil implementasi tampilan *cutscene* tersebut dapat dilihat pada Gambar 12.



Gambar 12. Implementasi tampilan *cutscene*

Implementasi tampilan menu *pause* yang dapat diakses ketika telah memasuki *game*. Fungsi dari menu *pause* selain untuk menghentikan permainan sementara, sebetulnya memiliki beberapa fungsi lain seperti tombol *continue* untuk melanjutkan permainan, tombol *restart* untuk mengulang permainan dari awal *stage* yang dimainkan, tombol *exit* untuk kembali ke menu utama. Hasil implementasi tampilan menu

pause tersebut dapat dilihat pada Gambar 13.



Gambar 13. Implementasi tampilan menu *pause*

Implementasi tampilan *stage* yang terdiri dari 4 *stage*. *Stage* 1 merupakan *stage* pertama yang akan muncul ketika game dimulai. Disini ada beberapa objek game seperti *player*, rintangan, *finish*, dan tombol *pause in game*.

Implementasi tampilan *stage* 2 memiliki objek yang hampir sama seperti *stage* 1, *stage* 2 memiliki beberapa tambahan objek tersendiri seperti menu *tutorial*, *stone teleportation*, dan *glue ground* yang dimana pemain tidak dapat melompat diatas objek tersebut.

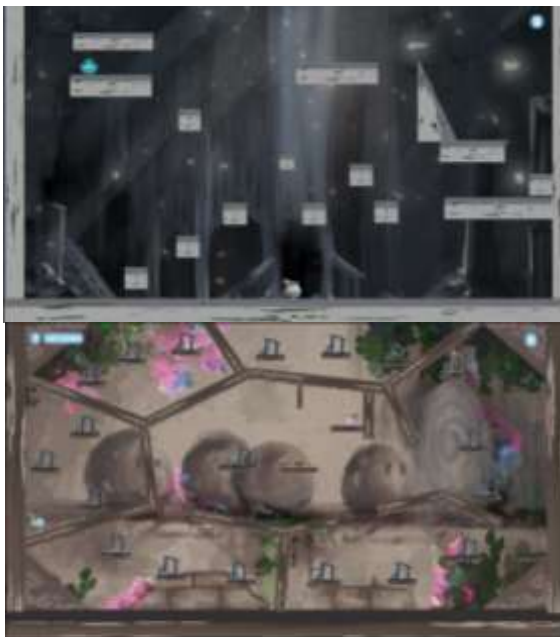
Implementasi tampilan *stage* 3, pada *stage* tersebut terdapat objek baru, yaitu *timer*. Setiap *timer* menyentuh angka 0 dan pemain tidak menyentuh *finish*, pemain akan masuk salah satu dari 3 *stage* yang telah tersedia secara *random*. Selain itu tersedia *falling ground* yang dimana ketika *player* menyentuh *falling ground*, maka *ground* tersebut akan terjatuh.

Implementasi tampilan *stage* 4 merupakan *stage* terakhir dari *game slime jump*. Memiliki objek yang hampir sama seperti *stage* 3, akan tetapi garis *finish* dihalangi dengan *invisible wall* yang dimana nantinya pemain akan terjatuh ke bagian bawah *stage*. Hasil implementasi tampilan *stage* tersebut dapat dilihat pada Gambar 14.



Gambar 14. Implementasi tampilan *stage*

Implementasi tampilan *cutscene ending*. Sedikit berbeda dengan *cutscene stage*. *Cutscene ending* merupakan penentuan cerita yang akan didapatkan oleh pemain, pada game ini memiliki 2 *ending* cerita yang berbeda tergantung apakah pemain melakukan klik terhadap objek bola energi yang merupakan *hidden button* atau bukan *hidden button*. Hasil implementasi tampilan *cutscene ending* tersebut dapat dilihat pada Gambar 15.



Gambar 15. Implementasi tampilan *cutscene ending*

Implementasi tampilan *scene* penutup. Halaman yang berisi ucapan terimakasih dan pesan dari pembuat *game*. Bagian ini merupakan bagian paling akhir dari permainan dan tersedia tombol *exit* untuk kembali ke menu utama. Hasil implementasi

tampilan *scene* penutup dapat dilihat pada Gambar 16.



Gambar 16. Implementasi tampilan *scene* penutup

F. Pengujian Game Platformer Slime Jump

Pada tahap pengujian fungsionalitas terhadap program menggunakan *blackbox testing* yang digunakan untuk mengamati proses dari *input* dan *output* kesesuaian suatu program yang diinginkan atau tidak diinginkan. Pengujian fungsionalitas sistem dilakukan oleh peneliti dan beberapa pengguna dengan menggunakan teknik *black box testing* dan *gameplay testing*. Hasil pengujian fungsionalitas sistem dapat dilihat pada Tabel 3.

Tabel 3. Hasil pengujian game slime jump

No .	Modul	Prasyarat	Hasil yang diharapkan	Hasil Pengujian
1	<i>Splash Screen</i>	Sebelum menampilkan <i>splashscreen</i> , pemain harus membuka <i>game</i> terlebih dahulu dengan cara <i>double click</i> di <i>shortcut game</i>	<i>Game</i> menampilkan <i>splashscreen</i>	Valid
2	Menu Utama	Pemain yang telah memasuki <i>game</i> setelah <i>splashscreen</i> .	Dapat memasuki Tampilan menu main dan memilih beberapa <i>option</i> diantaranya <i>start, setting, tutorial, exit.</i>	Valid
3	Menu <i>Setting</i>	Pemain menekan tombol <i>setting</i> pada menu utama	Dapat masuk ke tampilan menu <i>setting</i> untuk mengatur <i>screen</i> dan <i>audio</i> didalam permainan.	Valid
4	<i>Cutscene</i>	Pemain menekan tombol <i>start</i> atau telah menyelesaikan <i>stage</i> permainan.	Memutar video <i>cutscene</i> dan berhenti ketika video selesai atau melakukan <i>click</i> dibagian layar <i>game</i> .	Valid
5	<i>Stage</i> Permainan	Ditampilkan	Pemain dapat	Valid

		setelah pemain melihat <i>cutscene story.</i>	menyelesaikan permainan dengan cara menyentuh <i>finish.</i> Tersedia tombol <i>pause.</i>	
6	<i>Pause Menu</i>	Pemain menekan tombol <i>pause</i> di <i>stage</i> permainan.	<i>Game</i> akan berhenti. Berisi beberapa <i>option</i> seperti <i>continue, restart, exit.</i>	Valid
7	Gerakan <i>player</i>	Pemain menekan tombol A, Tombol D, panah kiri, panah kanan, spasi diluar kondisi <i>game pause.</i>	Ketika pemain menekan tombol A atau panah kiri <i>player</i> akan bergerak kearah kirim, ketika pemain menekan tombol D atau panah kanan <i>player</i> akan bergerak kearah kanan, dan ketika menekan spasi <i>player</i> akan melompat sebanyak satu kali.	Valid
8	Batu <i>teleport</i>	Karakter <i>player</i> menyentuh batu <i>teleport</i> lalu menekan tombol E	Setelah pemain menyentuh batu <i>teleport</i> , pemain dapat melakukan <i>teleportasi</i> dimanapun dengan tujuan <i>teleportasi</i> yang telah ditentukan tergantung batu terakhir mana yang disentuh.	Valid
9	<i>Trap Ground</i>	Karakter <i>player</i> berada diatas <i>trap ground.</i>	Karakter tidak dapat melompat	Valid
10	<i>Falling Ground</i>	Karakter <i>player</i> menyentuh <i>falling Ground</i>	<i>Ground</i> akan terjatuh.	Valid
11	<i>Cutscene ending</i>	Pemain menekan tombol rahasia.	Menampilkan video <i>cutscene ending</i> , dimana Ketika pemain menekan tombol rahasia maka	Valid



			pemain akan masuk ke <i>cutscene ending</i> . Ketika tidak ditekan pemain akan memasuki <i>cutscene ending</i> .	
12	Scene Penutup	Pemain telah selesai melihat <i>ending cutscene</i> yang dipilih.	Menampilkan ucapan terimakasih dan pesan untuk pemain yang telah memainkan <i>game</i> . Tersedia tombol <i>exit</i> untuk kembali ke menu utama.	Valid

Berdasarkan hasil pengujian fungsionalitas menggunakan *black box testing*, *game platformer Slime Jump* bekerja dengan baik dan memiliki kesesuaian dengan hasil analisis kebutuhan dan desain. Berdasarkan hasil *gameplay testing* bahwa *game platformer Slime Jump* cukup menghibur, interaktif, dan informatif.

#### V. KESIMPULAN

*Slime jump* merupakan *game platformer* 2D yang memiliki kontrol *game* yang sangat mudah untuk dimainkan. Meskipun begitu, *slime jump* juga dapat melatih para pemainnya dalam hal respon dan strategi pemain dalam menyelesaikan *game* tersebut. *Game* ini ditunjukkan untuk *player* yang menyukai tantangan akan tetapi sudah bosan memainkan *game* yang terlalu sulit dibalut dengan *story* yang ada dalam *game*. Selain itu, *game* ini juga memiliki beberapa keunikan dan kesulitan tersendiri pada setiap *stage*.

Pada *stage* pertama pemain diharuskan mencari cara untuk sampai ke tempat tujuannya dengan menggunakan mekanisme *game* yang ada. Pada *stage* 2 hampir sama seperti *stage* sebelumnya, hanya saja disini pemain mendapat fitur khusus, yaitu *teleportation*. Pemain bertujuan untuk mencari tempat teleportasi yang bisa digunakan untuk mencapai tempat tujuan dan bagaimana cara kesana tanpa melupakan mekanisme dasar *game slime jump*. Pada *stage* ini pemain bisa mengingat dan mencatat tujuan dari tiap alat teleportasi. Pada *stage* 3 pemain ditantang untuk berfikir cepat, karena *stage* 3 memiliki beberapa kejutan, selain itu pemain dihadapkan dengan suatu hal yang hanya ada di *stage* 3, yaitu *time limit*. Ketika *timer* menyentuh angka 0, *stage* akan berubah dan mulai dari awal. *Stage* 4 merupakan *stage* yang sangat sederhana namun penting, karena pada *stage* 4 penentuan dari *story ending* dari *game slime jump* yang ditentukan dengan cara yang simple, namun rahasia. Tujuan dibuatnya *game* ini untuk bersenang-senang dan mengajarkan kegagalan yang menyenangkan. Kesulitan dalam pembuatan *game* dari nol sampai akhir menjadi sebuah *game* yang dapat dimainkan, seperti

menentukan konsep awal *game* seperti apa, mencari ide apa saja yang bisa dimasukkan dalam *game*, bagaimana cara menerapkan semua ide dan konsep tersebut dalam *game* yang dibuat, dan memperhatikan keseruan dan keunikan dari *game* yang dibuat.

Berdasarkan hasil pengujian fungsionalitas menggunakan *black box testing*, *game platformer Slime Jump* bekerja dengan baik. Berdasarkan hasil *gameplay testing* dapat disimpulkan bahwa *game platformer Slime Jump* yang dikembangkan dengan metode sekuensial linier cukup menghibur, interaktif, dan informatif.

#### REFERENCES

- [1] C. Tang, Z. Wang, X. Sima, and L. Zhang, "Research on Artificial Intelligence Algorithm and Its Application in Games," in *2020 2nd International Conference on Artificial Intelligence and Advanced Manufacture (AIAM)*, 2020, pp. 386–389. doi: 10.1109/AIAM50918.2020.00085.
- [2] S. Murtiningsih, *Filsafat Pendidikan Video Games: Kajian tentang Struktur Realitas dan Hiperrealitas Permainan Digital*. Gajah Mada University Press, 2021. [Online]. Available: <https://books.google.co.id/books?id=cPwWEAAAQBAJ>
- [3] P. H. Miranda, H. D. Waluyanto, and A. Zacky, "Perancangan Buku Komik sebagai Media Informasi Mengenai Dampak Game Terhadap Perilaku Sosial Anak," *J. DKV Adiwarna*, vol. 1, no. 12, p. 9, 2018, [Online]. Available: <http://publication.petra.ac.id/index.php/dkv/article/view/7259>
- [4] T. S. A. Setyaningsih and H. Wahyuni, "Stimulasi Permainan Puzzle Berpengaruh Terhadap Perkembangan Sosial dan Kemandirian Anak Usia Prasekolah," *J. Keperawatan Silampari*, vol. 2, no. 1, pp. 1–13, 2018, doi: 10.31539/jks.v1i2.9 STIMULASI.
- [5] M. R. Hamadi, A. S. Lumenta, and M. D. Putro, "Rancang Bangun Aplikasi Game Edukasi Hafalan Doa Agama Islam," *J. Tek. Inform.*, vol. 12, no. 1, 2017, [Online]. Available: <https://ejournal.unsrat.ac.id/index.php/informatika/article/view/17791>
- [6] N. Ramsari and G. Ramadhan, "Pembuatan Game Side Scrolling 2D the Naila ' S Survival Berbasis Android," *J. Teknol. Inf. dan Komun.*, vol. VIII, no. 2, pp. 67–80, 2018, [Online]. Available: <file:///C:/Users/USER/PC/Downloads/310-Article Text-547-1-10-20200402.pdf>
- [7] W. C. Setyawan, Sulthoni, and S. Ulfa, "Pengembangan Multimedia Game Edukasi IPA Lapisan Bumi Untuk MTS," *JKTP J. Kaji. Teknol. Pendidik.*, vol. 2, no. 1, pp. 30–36, 2019, doi: 2615-8787.
- [8] I. Ahmad and W. Widodo, "Penerapan Algoritma A Star (A\*) pada Game Petualangan Labirin Berbasis Android," *Khazanah Inform. J. Ilmu Komput. dan Inform.*, vol. 3, no. 2, pp. 57–63, 2017.
- [9] M. Iqbal, S. Rahayu, and T. Herdiawan, "Rancang Bangun Sistem Pembelajaran Game Edukasi Berbasis Web Guna Meningkatkan Ranah Psikomotorik Pada Mata Pelajaran Matematika di Level SMP," *J. CoreIT J. Has. Penelit. Ilmu Komput. dan Teknol. Inf.*, vol. 6, no. 1, p. 8, 2020, doi: 10.24014/coreit.v6i1.9115.
- [10] G. P. Mahardhika, "Gim Proses Donor Darah Sebagai Alat Promosi Donor Darah," *JMAI (Jurnal Multimed. Artif. Intell.)*, vol. 3, no. 1, pp. 9–16, 2019, doi: 10.26486/jmai.v3i1.91.
- [11] A. Zulkarnais, P. Prasetyawan, and A. Sucipto, "Game Edukasi Pengenalan Cerita Rakyat Lampung Pada Platform Android," *J. Inform. J. Pengemb. IT*, vol. 3, no. 01, pp. 96–102, 2018.
- [12] U. Umair and H. A. Tawakal, "Pengembangan Aplikasi Permainan Pembelajaran Matematika menggunakan Model-Driven Game Development," *J. Teknol. Terpadu*, vol. 7, no. 1, pp. 39–44, 2021, doi: 10.54914/jtt.v7i1.346.
- [13] Herdiman, J. Pragantha, and D. A. Haris, "Pembuatan Game Platformers 'Robot Hero' Multi-Platform," *J. Ilmu Komput. dan Sist. Inf.*, pp. 191–195, 2017.
- [14] F. A. Jellyanto, I. Kuswardayan, and N. Suciati, "Rancang Bangun Pembangkit World Dinamis Menggunakan Algoritma Recursive Backtracking pada Game 2D Platformer 'Mine Meander,'" *J. Tek. ITS*,

- vol. 5, no. 2, 2016.
- [15] A. M. Pratama and A. P. Armin, "Perancangan Permainan Platformer Ninja Runner Menggunakan Metode Finite State Machine Berbasis Desktop Dengan Tools Unity 2D Game Engine," Universitas 17 Agustus 1945 Surabaya, 2021.
- [16] A. Summerville, J. Osborn, C. Holmgaard, and D. W. Zhang, "Mechanics Automatically Recognized via Interactive Observation: Jumping," 2017. doi: 10.1145/3102071.3102104.
- [17] Nanang Hoesen, "Rancang Bangun Game Berbasis Android Bertemakan Cerita Rakyat Betawi Si Pitung," *J. Esensi Infokom*, vol. 5, no. 2, pp. 32–37, 2021.
- [18] P. Sharma and N. Hasteeer, "Analysis of linear sequential and extreme programming development methodology for a gaming application," in *2016 International Conference on Communication and Signal Processing (ICCSP)*, 2016, pp. 1916–1920. doi: 10.1109/ICCSP.2016.7754505.
- [19] Y. G. Putra, "Pembuatan Game Balap Kucing Dengan Unity Berbasis Android," 2017.
- [20] W. A. Septiko, "Pengembangan Game Edukasi Platformer Kisah Gajah Mada Menyatukan Nusantara Menggunakan Metode Iterative With Rapid Prototyping," Universitas Brawijaya, 2018.
- [21] F. Yedithia, "Pengembangan Game Platformer 'Ayo Lawan Virus' Berbasis Android dengan Menggunakan Game Engine Unity," *J. Hum. Teknol.*, vol. 8, no. 1, pp. 1–9, 2022.
- [22] I. Sommerville, *Engineering Software Products: An Introduction to Modern Software Engineering*, First edit. Hoboken: Pearson Education, 2019. [Online]. Available: <https://lcn.loc.gov/2019004631>