

Rancang Bangun Game 2D Shooter Menggunakan Metode Finite State Machine

Agung Nopriansyah¹, Indra Kanedi², Prahasti³
^{1,2,3} Universitas Dehasen Bengkulu

Jl. Meranti No.32, Sawah Lebar, Kec. Ratu Agung, Kota Bengkulu, Bengkulu 38228
Email: meagung62@gmail.com¹, indrankanedi12@gmail.com², prahasti.mona82@gmail.com³

ABSTRAK

Game 2D Shooter ini merupakan permainan video petualangan dimana pemain memainkan karakter seorang *Survivor*. *Game* ini berfokus pada petualangan *Survivor* membunuh semua monster yang ada agar bisa melanjutkan ke *level* selanjutnya hingga tamat. *Game* ini menggunakan *Game Engine Unity*, dengan menerapkan *Finite State Machine*. *Finite State Machine* digunakan pada *NPC (Non-Playable Character)* yaitu karakter yang muncul pada permainan sebagai monster dalam *game* tersebut seperti karakter *Enemy* dan *Boss*. Dalam pengujian yang telah dilakukan hasil dari *FSM (Finite State Machine)* pada *game 2D Shooter* ini sebagian responden dengan rata-rata 32% menjawab Sangat Bagus, 41% menjawab Bagus, 24% menjawab Biasa Saja, 3% menjawab Buruk.

Kata Kunci: Game, Unity, Game Engine, 2D Shooter, Finite State Machine

ABSTRACT

This 2D Shooter game is an adventure video game where the player plays the character of a Survivor. This game focuses on the Survivor's adventure to kill all the monsters in order to continue to the next level until the end. This game uses the Unity game engine, by implementing the Finite State Machine. Finite State Machine is used on NPCs (Non-Playable Characters), namely characters that appear in the game as monsters in the game such as Enemy and Boss characters. In the tests that have been carried out the results of the FSM (Finite State Machine) in this 2D Shooter game, most of the respondents with an average of 32% answered Very Good, 41% answered Good, 24% answered Moderate, 3% answered Poor.

Keywords: Game, Unity, Game Engine, 2D Shooter, Finite State Machine

Pendahuluan

Permainan Video atau lebih dikenal dengan *Video Game* biasanya identik dengan permainan yang dimainkan khalayak ramai dikomputer atau di perangkat elektronik lainnya. Sejak dulu dunia *Game* salah satu yang paling diminati, di jaman sekarang *Game* sudah diminati dari banyak kalangan pria dan wanita dari berbagai umur sudah banyak bermunculan. *Game* yang dimainkan pun sudah banyak jenisnya mulai dari *Game Portable*, *Game PC* ataupun *Game Mobile*. Seiring dengan perkembangan waktu dan teknologi, *Industri Game* pun ikut berkembang dengan pesat, mulai dari berkembangnya *Hardware* yang cukup berat untuk mengangkat *game* yang berat pula, *Tampilan grafis* yang memukau yang hampir menyerupai dunia nyata. [1], [2]

Perkembangan yang pesat ini diikuti pula dengan banyaknya orang yang membuat *game* atau *Game Developer* secara independen bermunculan. Seiring dengan perkembangan teknologi dan zaman semakin banyak pula bermunculan insan kreatif. Di masa sekarang *Game* telah menjadi salah satu lahan

bisnis yang menjanjikan karena dizaman sekarang perkembangannya juga pesat dan banyak diminati. *Teknologi* di bidang *Game development* juga berkembang pesat dengan munculnya berbagai *engine* yang mempermudah para *developer* untuk membuat *game* yang sesuai keinginannya. Salah satunya *Unity*, *Unity* adalah *game editor* berbasis *C#* yang dikembangkan oleh *Unity Technologies* yang memberikan kebebasan kepada user untuk membuat *game* berbasis 2D atau 3D. Karena *Unity* berbasis bahasa pemrograman *C#* maka diperlukan pemahaman tentang *Coding*. [3], [4]

1. Game

Game merupakan kalimat bahasa inggris yang memiliki arti dasar "Permainan". Permainan dalam hal ini merujuk pada pengertian "Kelincahan Intelektual" (*intellectual playability*). *Game* bisa diartikan sebagai arena keputusan dan aksi pemainnya. Ada target-target dan misi untuk dicapai pemainnya. *Kelincahan intelektual* pada tingkat tertentu merupakan ukuran sejauh mana *game* itu menarik untuk dimainkan secara maksimal.

2. Game Engine

Game Engine adalah sebuah perangkat lunak (*software*) yang dirancang untuk pembuatan dan pengembangan suatu *video game*. *Game engine* memberikan kemudahan dalam menciptakan konsep sebuah *game* yang akan di buat. Mulai dari sistem *rendering*, *physics*, arsitektur suara *scripting*, *Artificial Intelligence*, dan bahkan *sistem networking*. *Game engine* bisa disebut sebagai jiwa aspek dari sebuah *game*

3. Unity

Unity Technologies berdiri tahun 2004 oleh David Helgason, Nicholas Francis dan Joachim Ante. *Game Engine* ini dibangun atas dasar kepedulian mereka terhadap *indie developer* yang tidak bisa membeli *game engine* karena terlalu mahal. Perusahaan ini berfokus untuk membuat sebuah perangkat lunak yang bisa digunakan oleh semua orang, khususnya untuk membangun sebuah *game*. Di tahun 2009, *Unity* diluncurkan secara gratis dan pada bulan Agustus 2012, *Unity* mencapai popularitas tertinggi dengan lebih dari 1 juta *developer* terdaftar di seluruh dunia.

Menurut [5] *Unity* memungkinkan seseorang untuk membuat sebuah *game 3D* dengan mudah dan cepat. Secara *default*, *Unity* telah diatur untuk pembuatan *game bergenre First Person Shooter (FPS)*, namun *Unity* juga bisa digunakan untuk membuat *game bergenre Role Playing Game (RPG)*, dan *Real-Time Strategy (RTS)*. Selain itu, *Unity* merupakan sebuah *engine multiplatform* yang memungkinkan *game* yang anda bangun dipublish untuk berbagai platform seperti *Windows*, *Mac*, *Android*, *iOS*, *PS3* dan juga *Wii*.

Unity merupakan suatu aplikasi yang digunakan untuk mengembangkan *game multi-platform* yang didesain untuk mudah digunakan. *Unity* itu bagus dan penuh perpaduan dengan aplikasi yang profesional. Editor pada *Unity* dibuat dengan user interface yang sederhana.[4]

4. Finite State Machine

Menurut [6], [7] *Finite State Machine* adalah sebuah perangkat atau model yang memiliki beberapa jumlah kondisi yang terbatas pada waktu tertentu dan dapat beroperasi pada masukan baik melalui transisi dari salah satu kondisi ke kondisi yang lain sehingga menyebabkan output atau tindakan untuk memutuskan sesuatu. *Finite State Machine* hanya dapat digunakan untuk satu kondisi pada setiap peristiwa.

5. C# (C-Sharp)

Menurut Nugroho *C#* atau *C-Sharp* adalah bahasa pemrograman yang diciptakan oleh *Microsoft* (dikembangkan oleh Anders Hjelberg yang telah menciptakan berbagai macam bahasa

pemrograman termasuk *Borland Turbo C++* dan *Borland Delphi*). Bahasa *C#* juga telah distandarisi secara internasional oleh ECMA. [8]–[10][11], [12]

C# merupakan salah satu aplikasi yang memiliki kemampuan dalam penguatan *Framework.NET*. *C#* dibuat sejalan dengan perkembangan *Framework.NET*, *C#* sendiri dikembangkan oleh *Microsoft*. Dalam penerapannya *C-Sharp (C#)* menjanjikan produktifitas, fleksibilitas serta kemudahan yang ada dari aplikasi sebelumnya yaitu *Visual Basic*, *Java* dan *C++*. *C#* mengadopsi kemampuan dari penggabungan aplikasi sebelumnya (mempelajari-*c*) *Microsoft* membuat *C#* seiring dengan pembuatan *Framework.NET*. *Chief Architect* dalam pembuatan *C#* adalah Anders Hejlsberg yang sebelumnya berperan dalam pembuatan *Borland Delphi* dan *Turbo Pascal*. *C#* menjanjikan produktifitas dan kemudahan yang ada di *Visual Basic* dengan kemampuan dan fleksibilitas yang ada di *C/C++*. Menurut spesifikasi bahasanya, "*C# (pronounced "C Sharp") is a simple, modern, object oriented, and type-safe programming language. It will immediately be familiar to C and C++ programmers. C# combines the high productivity of Rapid Application Development (RAD) languages and the raw power of C++.*". Untuk mencapai produktifitas tinggi ini konsep-konsep sulit *C++* disederhanakan dan fitur-fitur baru ditambahkan. Hal ini mungkin terasa mirip dengan *Java*, karena itulah *C#* bisa dianggap sebagai sepupu *Java* [13]–[15]

Metode Penelitian

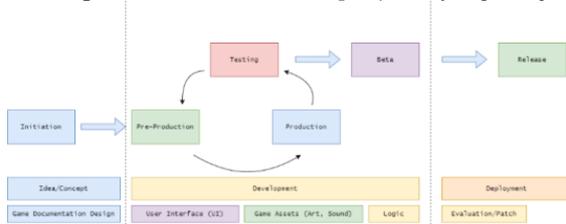
Pada penelitian ini metode yg digunakan adalah metode *Game Development Life Cycle (GDLC)*. Menurut [5], [16], [16], [17] *Game Development Life Cycle (GDLC)* merupakan tahapan yang dilaksanakan secara sistematis untuk membangun *game (video game)* yang terdiri dari 6 fase yaitu *Initiation*, *Pre-Production*, *Production*, *Testing* (pengujian), *Beta Release* dan *Full Release*. Rangkaian *GDLC* biasanya disederhanakan menjadi 3 fase yaitu *Idea/Concept*, *Development* dan *Deployment*.

Idea/Concept, Pengembang *game* akan membuat ide dasar terkait latar belakang pembuatan sebuah *game*, *main story*, spesifikasi karakter, genre *game*, jenis *art* yang digunakan, *platform* yang akan dituju, dimensi *game* yang akan dibuat sampai monetisasi *game* jika sasarannya untuk diperjual belikan. Biasanya, *Output* dari fase ini adalah *Game Document Design (GDD)* yang akan menjadi acuan utama dalam proses pengembangan selanjutnya.

Development, Fase ini berfokus pada penetapan jadwal pengembangan, penentuan tim yang terlibat serta tenggat waktu yang akan

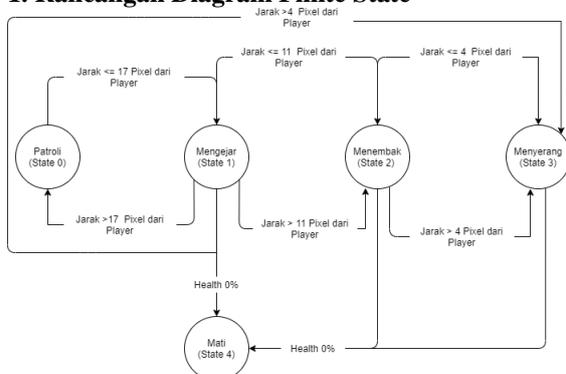
digunakan dalam fase pembangunan yang sudah memasukkan unsur pengujian dan penetapan tanggal rilis beta.

Deployment lebih diarahkan pada teknis publikasi *game* pada *platform* yang dituju, termasuk skema persiapan patch jika *game* yang dihasilkan memiliki bug atau mengalami gangguan teknis terkait perbedaan *environment platform* yang dituju.



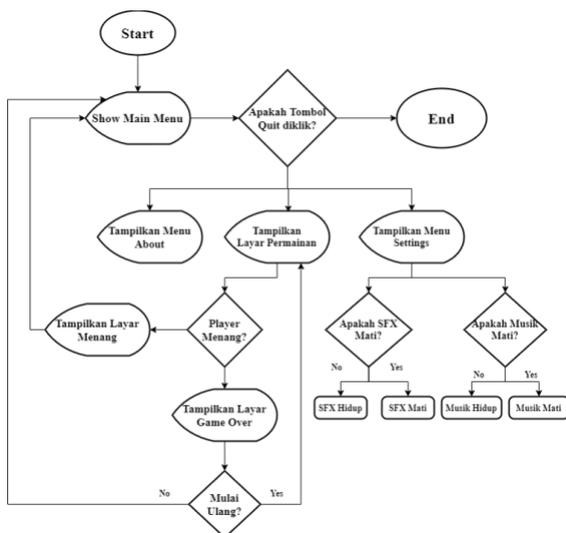
Gambar 1. Game Development Life Cycle (GDLC)

1. Rancangan Diagram Finite State



Gambar 2. Rancangan Finite State Machine

2. Flowchart Game



Gambar 3. Tampilan Flowchart

Hasil dan Pembahasan

2D Shooter Platformer adalah *game* yang bergenre 2D Action Shooter yang dirancang menggunakan aplikasi Unity serta menggunakan metodologi Finite State Machine (FSM).[16], [18]–[20]

1. Tampilan Menu Utama

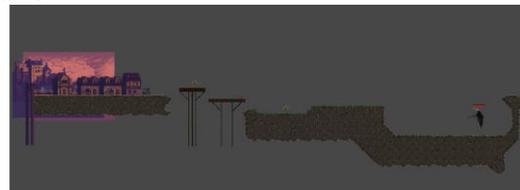
Bagian ini merupakan bagian pertama yang muncul saat *game* baru dimulai. Pada tampilan ini terdapat tombol seperti Mulai, Pengaturan, Tentang dan Keluar.



Gambar 4. Tampilan Menu Utama

2 Tampilan Level 1

Tampilan *level 1* ini terdapat musuh tengkorak yang mana jika kita berada didekat nya maka ia akan langsung mengejar player dan jika berada terlalu dekat monster tengkorak akan menyerang player, gambar berikut adalah tampilan keseluruhan map pada *game engine Unity* bukan tampilan dalam *game*.



Gambar 5. Tampilan map Level 1

3. Tampilan Boss Level 1

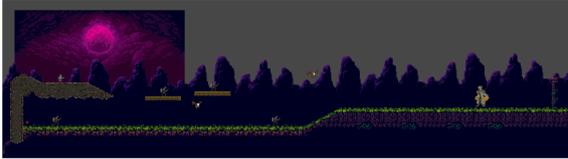
Tampilan *boss level 1* ini *boss* bisa mengejar *player* pada jangkauan tertentu, *boss* ini hanya bisa menembak ke arah *player*.



Gambar 6. Tampilan Boss Level 1

4. Tampilan Level 2

Di level 2 ini ada monster yang berbeda yaitu monster yang bisa terbang mengikuti *player*, jika berada dekat dengan *player* maka monster ini akan menembak ke arah *player*. Selain monster yang terbang ada monster tengkorak yang sama seperti di level 1.



Gambar 7. Tampilan Level 2

5. Tampilan Boss Level 2

Boss pada level 2 ini tidak mengejar *player* tapi jika berada jarak tertentu *boss* ini akan melempar batu ke arah *player* dengan kencang, jadi *player* harus menghindari setiap batu yg dilempar.



Gambar 8. Tampilan Boss level 2

6. Tampilan Level 3

Pada level 3 ini terdapat 2 monster baru pemanah dan *goblin*, *Goblin* prinsipnya sama seperti dengan tengkorak hanya gerakannya lebih cepat. Pemanah tidak bisa mengejar pemain tapi jika berada pada jarak tertentu pemanah akan menembak busur ke arah *player*.



Gambar 9. Tampilan Level 3

7. Tampilan Boss Level 3

Boss level terakhir ini prinsipnya sama dengan *boss* sebelumnya tapi *boss level* terakhir ini jika *player* berada pada jarak tertentu dengan *boss* maka *boss* ini akan menembakkan bola api ke arah *player*.



Gambar 10. Tampilan Boss Level 3

8. Tampilan Jeda

Tampilan ini muncul jika pemain menekan tombol *ESC* pada *keyboard*, ketika muncul permainan akan berhenti dan akan muncul pilihan untuk kembali ke permainan, pengaturan dan keluar ke menu. Jika kembali ke permainan maka game akan lanjut bergerak lagi seperti biasa, jika tombol pengaturan dipilih akan muncul pilihan mematikan suara, dan jika tombol keluar ke menu maka akan kembali ke tampilan menu awal *game*.



Gambar 11. Tampilan Jeda

9. Tampilan Jika Player Mati

Berikut adalah tampilan yang akan muncul jika *player* mati diserang oleh musuh atau jatuh ke lubang, ketika tampilan ini muncul pemain diberi dua pilihan yaitu memulai ulang permainan atau kembali ke menu.



Gambar 12. Tampilan Jika Player Mati

10. Tampilan Menu Pengaturan

Pada tampilan tombol pengaturan ini berisi tentang pengaturan suara pada *game*, menu ini mengatur ada atau tidaknya suara dalam *game* seperti musik dan efek suara tembakan.



Gambar 13. Tampilan menu Pengaturan

11. Tampilan Menu Tentang

Pada tampilan ini berisi tentang petunjuk kontrol permainan dan data diri pembuat *game*.



Gambar 14. Tampilan Menu Tentang

12 Pengujian Menggunakan Black Box Testing

Tabel 1. Pengujian Menggunakan Black Box Testing

No	Percobaan yang dilakukan	Hasil	Keterangan
1		Memulai permainan	Valid
2		Menampilkan keterangan tentang <i>game</i> dan biodata pembuat	Valid
3		Menampilkan pengaturan suara	Valid
4		Efek suara dan musik akan mati	Valid

5		Keluar dari permainan	Valid
6	Jika tombol spasi atau panah atas ditekan	Karakter akan melompat	Valid
7	Jika tombol A atau panah kiri ditekan	Karakter akan bergerak ke kiri	Valid
8	Jika tombol D atau panah kanan ditekan	Karakter akan bergerak ke kanan	Valid
9	Jika mengklik kanan pada mouse atau <i>ctrl</i> kiri	Karakter akan menembakkan peluru	Valid
10	Jika tombol <i>ESC</i> ditekan	Memunculkan menu berhenti	Valid
11	Jika musuh menyerang karakter	Darah player akan berkurang	Valid
12	Jika karakter menyerang musuh	Darah lawan berkurang atau mati	Valid
13	Jika player mati	Mengulang ke awal level	Valid
14	Jika player berhasil mengalahkan <i>boss</i> disetiap level	Menuju ke level berikutnya	Valid

13. Hasil Pengujian Game Kepada User

Pengujian ini dilaksanakan sebagai bentuk untuk mengetahui apakah *2D Game Shooter Platformer* sudah berjalan dengan baik. Pengujian ini dilakukan terhadap 15 orang responden dengan mengisi kuesioner pada *Google Form* setelah responden mencoba *game*.

Kuesioner yang sudah diisi dapat dilihat pada lampiran. Dari hasil kuesioner akan dilakukan perhitungan dengan interval nilai 0% - 24% (Tidak Layak), 25% - 49% (Cukup Layak), 50% - 74% (Layak), 75% - 100% (Sangat Layak).

Keterangan :
 1 = Sangat Bagus
 2 = Bagus
 3 = Biasa Saja
 4 = Buruk

Tabel 2. Hasil Kuesioner Pengujian Game

No	Pertanyaan	Penilaian
----	------------	-----------

	1	2	3	4
1	5	8	2	0
2	5	8	2	0
3	6	6	3	0
4	5	4	5	1
5	3	5	6	1
Rata-Rata	32%	41%	24%	3%

Berdasarkan tabel 2 dapat disimpulkan bahwa *game 2D Shooter Platform Shooter* ini layak dimainkan dinyatakan dengan nilai rata-rata 32% menjawab Sangat Bagus. 41% menjawab Bagus, 24% menjawab Biasa Saja, 3% menjawab Buruk, maka dari itu hasil pengujian terhadap 15 responden berada pada persenan tertinggi yaitu 41% yang artinya berada pada interval Layak.

Kesimpulan

Berdasarkan hasil perancangan serta pengujian *game 2D Shooter Platformer* dapat disimpulkan bahwa *Finite State Machine (FSM)* berhasil diterapkan pada *game* yang bergenre *Action Shooter* ini untuk kecerdasan musuh pada *game* ini dan berdasarkan pengujian *black box* berjalan dengan lancar sesuai dengan rancangan dan berdasarkan kuesioner kepada 15 orang bahwa *game* ini layak untuk dimainkan dan musuh pada *game* ini berfungsi dengan baik.

Daftar Pustaka

[1] A. Anastasya and F. Yuamita, "Pengendalian Kualitas Pada Produksi Air Minum Dalam Kemasan Botol 330 ml Menggunakan Metode Failure Mode Effect

Analysis (FMEA) di PDAM Tirta Sembada," *J. Teknol. dan Manaj. Ind. Terap.*, vol. 1, pp. 15–21, 2022, doi: <https://doi.org/10.55826/tmit.v1i1.4>.

[2] F. N. Rahman and A. Y. Pratama, "Analisis Beban Kerja Mental Pekerja Train Distribution PT. Solusi Bangun Indonesia," *J. Teknol. dan Manaj. Ind. Terap.*, vol. 1, pp. 7–14, 2022, doi: <https://doi.org/10.55826/tmit.v1i1.11>.

[3] "Apa Itu Game Engine «Ricky's Blog.» <https://rickykurn.wordpress.com/2012/03/08/apa-itu-game-engine/> (accessed Jun. 05, 2022).

[4] T. Ardyanto, A. P.-G. I. J. Ilmiah, and undefined 2017, "Pembuatan Game 2D Petualangan Hanoman Berbasis Android," *jurnal.stmik-aub.ac.id*, Accessed: Jun. 05, 2022. [Online]. Available: <http://jurnal.stmik-aub.ac.id/index.php/goinfotech/article/view/79>.

[5] R. Roedavan, "Unity tutorial game engine," 2018, Accessed: Jun. 05, 2022. [Online]. Available: http://repo.unikadelasalle.ac.id/index.php?p=show_detail&id=12642&keywords=.

[6] S. Asmiatun and A. Putri, *Belajar Membuat Game 2D dan 3D Menggunakan Unity*. 2017.

[7] C. wei Huang, "Assessing the performance of tourism supply chains by using the hybrid network data envelopment analysis model," *Tour. Manag.*, vol. 65, pp. 303–316, 2018, doi: 10.1016/j.tourman.2017.10.013.

[8] M. A. Jauhari, D. Hamidin, and M. Rahmatuloh, "Stabilitas Eksekusi Kode Bahasa Pemrograman. Net C# Versi 4.0. 3019 Dengan Google Golang Versi 1.4. 2 Menggunakan Algoritma Bubble Sort dan Insertion Sort," *ejurnal.poltekpos.ac.id*, vol. 9, no. 1, 2017, Accessed: Jun. 05, 2022. [Online]. Available: <https://ejurnal.poltekpos.ac.id/index.php/informatika/article/view/34>.

[9] J. Hasil Penelitian dan Karya Ilmiah, M. Rizki, D. Devrika, F. Surayya Lubis, and I. Hadiyul Umam, "Aplikasi Data Mining dalam penentuan layout swalayan dengan menggunakan metode MBA," *ejournal.uin-suska.ac.id*, vol. 5, no. 2, 2019, Accessed: Jun. 05, 2022. [Online]. Available: <http://ejournal.uin-suska.ac.id/index.php/jti/article/view/8958>.

[10] E. Permata, M. Rizki, ... P. P.-S. J. S., and undefined 2020, "Analisa Strategi Pemasaran Dengan Metode BCG (Boston

- Consulting Group) dan Swot,” *ejournal.uin-suska.ac.id*, vol. 17, no. 2, pp. 92–100, 2020, Accessed: Aug. 25, 2021. [Online]. Available: <http://ejournal.uin-suska.ac.id/index.php/sitekin/article/view/12329>.
- [11] R. Kuo, M. Rizki, F. Zulvia, A. K.-C. & Industrial, and undefined 2018, “Integration of growing self-organizing map and bee colony optimization algorithm for part clustering,” *Elsevier*, Accessed: Jun. 05, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0360835218301888>.
- [12] M. Rizki, A. Wenda, ... F. P.-2021 I., and undefined 2021, “Comparison of Four Time Series Forecasting Methods for Coal Material Supplies: Case Study of a Power Plant in Indonesia,” *ieeexplore.ieee.org*, Accessed: Jun. 05, 2022. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9493522/>.
- [13] T. Solusi, P. Non, L. Yahya, and A. M. Nur, “Pengaruh Aplikasi C# dalam Proses Perhitungan Numerik Terhadap Solusi Persamaan Non Linier,” *ejournal.hamzanwadi.ac.id*, vol. 1, no. 2, pp. 79–87, 2018, doi: 10.29408/jit.v1i2.901.
- [14] J. N. A. Aziza, “Perbandingan Metode Moving Average, Single Exponential Smoothing, dan Double Exponential Smoothing Pada Peramalan Permintaan Tabung Gas LPG PT Petrogas Prima Services,” *J. Teknol. dan Manaj. Ind. Terap.*, vol. 1, pp. 35–41, 2022, doi: <https://doi.org/10.55826/tmit.v1i1.8>.
- [15] A. Wicaksono and F. Yuamita, “Pengendalian Kualitas Produksi Sarden Menggunakan Metode Failure Mode and Effect Analysis (FMEA) Untuk Meminimumkan Cacat Kaleng Di PT. Maya Food Industries,” *J. Teknol. dan Manaj. Ind. Terap.*, vol. 1, pp. 1–6, 2022, doi: <https://doi.org/10.55826/tmit.v1i1.6>.
- [16] N. S.-S. J. Sains, T. dan Industri, and undefined 2014, “Rancang Bangun Aplikasi Pembelajaran Berbasis Teknologi Augmented Reality Pada Smartphone Android (Studi Kasus: Materi Sistem Tata Surya Kelas Ix),” *ejournal.uin-suska.ac.id*, vol. 12, no. 1, pp. 41–47, 2014, Accessed: Jun. 05, 2022. [Online]. Available: <http://ejournal.uin-suska.ac.id/index.php/sitekin/article/view/772>.
- [17] U. Berbasis, A. Pada, K. Perhubungan, K. P. Anofrizen, and M. L. Hamzah, “Rancang Bangun Sistem Informasi Geografis Rute Angkutan Umum Berbasis Android Pada Kementerian Perhubungan Kota Pekanbaru,” *ejournal.uin-suska.ac.id*, vol. 18, no. 2, pp. 283–291, 2021, Accessed: Jun. 05, 2022. [Online]. Available: <http://ejournal.uin-suska.ac.id/index.php/sitekin/article/view/13766>.
- [18] F. Jurusan, S. Informasi, F. Sains, T. Uin, and S. Riau, “Rancang Bangun Aplikasi Sistem Informasi Geografis (SIG) Pemilihan Sekolah Menengah Pertama (SMP) Menggunakan Analytical Hierarchy Process (AHP)(Studi,” *ejournal.uin-suska.ac.id*, vol. 9, no. 2, 2012, Accessed: Jun. 05, 2022. [Online]. Available: <http://ejournal.uin-suska.ac.id/index.php/sitekin/article/view/595>.
- [19] H. Erwanto, S. M.-J. sains, T. dan I. sains, and undefined 2016, “Rancang Bangun Sistem Informasi Fitness Berbasis Web dan SMS Gateway pada Rajawali Fitness Pekanbaru,” *ejournal.uin-suska.ac.id*, vol. 14, no. 1, pp. 72–78, 2016, Accessed: Jun. 05, 2022. [Online]. Available: <http://ejournal.uin-suska.ac.id/index.php/sitekin/article/view/2693>.
- [20] M. Zawaruddin Abdullah *et al.*, “Rancang Bangun Sistem Informasi Akuntansi Berbasis Website menggunakan Framework Laravel,” *ejournal.uin-suska.ac.id*, vol. 18, no. 1, pp. 49–56, 2020, Accessed: Jun. 05, 2022. [Online]. Available: <http://ejournal.uin-suska.ac.id/index.php/sitekin/article/view/11313>.