

Perancangan Aplikasi Game Maze Escape Dengan Menerapkan Algoritma Greedy Untuk Pencarian Jalur Terpendek

Nanda Sopianto, Mesran

Prodi Teknik Informatika, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Budi Darma, Medan, Indonesia

Email: ^{1*}nandasopianto@gmail.com, ²mesran.skom.mkom@gmail.com

Email Penulis Korespondensi: nandasopianto@gmail.com

Submitted: 03/12/2021; Accepted: 15/12/2021; Published: 31/12/2021

Abstrak—Pada jaman modern permainan yang membutuhkan ruang yang luas dan menghabiskan banyak sumber daya seperti permainan maze mulai banyak menghilang. Namun perkembangan teknologi membuat permainan-permainan seperti maze escape berkembang dalam bentuk simulasi komputer. Tujuan dari penelitian ini adalah merancang sebuah game yang dapat dimainkan untuk mengisi waktu luang dan menghilangkan kejenuhan dan sekaligus melatih otak pemain. Untuk menentukan rute yang paling optimal yaitu rute yang mempunyai jarak terpendek menggunakan pendekatan algoritma Greedy. Algoritma Greedy adalah algoritma yang memecahkan masalah langkah demi langkah dan merupakan salah satu metode dalam masalah optimasi. Pendekatan yang digunakan di dalam algoritma Greedy adalah membuat pilihan yang terlihat memberikan perolehan terbaik yaitu dengan membuat pilihan optimum local pada setiap langkah dan diharapkan akan mendapatkan solution optimum global.

Kata Kunci: Labirin; Game; Algoritma Greedy; Construct 2

Abstract—In modern times games that require extensive space and spend a lot of resources such as the maze game are starting to disappear. But the development of technology made games such as maze escape develop in the form of computer simulations. The purpose of this study is to design a game that can be played to fill free time and eliminate boredom and simultaneously train the player's brain. To determine the most optimal route, the route that has the shortest distance is using the Greedy algorithm approach. The Greedy algorithm is an algorithm that solves problems step by step and is one method in optimization problems. The approach used in the Greedy algorithm is to make a choice that looks to provide the best gain, namely by making a choice of local optimum at each step and is expected to get a global optimum solution.

Keywords: Maze; Game; Greedy Algorithm; Construct 2

1. PENDAHULUAN

Permainan (*Game*) merupakan kegiatan menyenangkan dengan tujuan bersenang-senang atau mengisi waktu luang disaat suntuk. *Game* biasanya dilakukan sendiri (*Single Player*) atau berkelompok (*Multi Player*). *Game* tidak mengenal umur dan status pemain itu sendiri. Semua orang boleh menikmati *game*, mau itu muda ataupun tua, kaya ataupun miskin, pria ataupun wanita. Semua orang boleh menikmati *game*. Biasanya *game* banyak disukai dikalangan anak-anak dan orang tua. Tujuan *game* sebenarnya penting dalam perkembangan otak manusia. Selain untuk meningkatkan konsentrasi, juga melatih dalam memecahkan masalah secara cepat dan tepat dikarenakan dalam sebuah *game* terdapat berbagai macam konflik ataupun masalah yang menuntut kita untuk menyelesaikan konflik dan masalah itu dengan cepat dan tepat.

Game itu sendiri meliputi proses mewujudkan sebuah ide menjadi sebuah produk yang dapat dimainkan semua orang. Perlu dipahami bahwa tidak ada aturan standar dalam merancang sebuah aplikasi *game*. Contohnya saat kita merancang sebuah aplikasi *game* seperti aplikasi *game* Tetris, akan sangat berbeda jika dibandingkan dengan merancang aplikasi *game platform* seperti aplikasi *game* Super Mario. Sama dengan penggunaan tampilan grafis, aplikasi *game* dengan grafis 2D akan sangat berbeda proses pembuatannya dengan aplikasi *game* 3D. Meskipun begitu, terdapat karakteristik yang sama dalam mengembangkan sebuah aplikasi *game*, dimana hal tersebut dapat diaplikasikan pada sebagian besar proses merancang sebuah *game*. Hal pertama yang perlu kita pahami dalam membuat sebuah aplikasi *game* adalah dengan mengetahui kemampuan, kelebihan, dan kelemahan dalam aplikasi *game* tersebut.

Setelah mengetahui kemampuan, kelebihan dan kelemahan dalam aplikasi *game* tersebut. Maka kita bisa memulai mencari ide *game* untuk diwujudkan kedalam aplikasi *game* tersebut. Ide *game* itu sendiri meliputi pertanyaan tentang bagaimana aplikasi *game* tersebut dimainkan, apakah tujuan dari memainkan *game* tersebut, apakah pemain akan memainkan aplikasi *game* secara tersebut berulang-ulang atau hanya sekali bermain, apakah *game* tersebut nantinya menyenangkan atau membosankan, dan berbagai gambaran lainnya[1].

Algoritma *Greedy* merupakan salah satu algoritma yang biasa digunakan untuk memecahkan masalah optimasi. Algoritma *Greedy* juga merupakan algoritma yang dapat menentukan sebuah jalur terpendek yang akan digunakan dengan mengambil secara terus menerus dan menambahkan ke dalam jalur yang akan dilewatinya. Tertuju pada konsep *greedy* yang menganggap bahwa pada setiap langkah akan dipilih tempat atau kota yang belum pernah dikunjungi, dimana tempat atau kota tersebut memiliki jarak terdekat dari tempat atau kota sebelumnya. Algoritma ini tidak mempertimbangkan nilai heuristic, yang dalam hal ini bisa berupa jarak langsung antar dua tempat. Algoritma *Greedy* ini metode yang paling populer dalam menemukan solusi langkah perlangkah (*step by step*). Pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Keputusan yang telah diambil pada suatu langkah tidak dapat diubah lagi pada langkah selanjutnya[2].

Kelebihan dari Algoritma *Greedy* adalah algoritma ini cepat dalam bertindak (*fast respon*). Dan Algoritma *Greedy* tidak membutuhkan waktu yang lama untuk menyelesaikan masalah. namun, kekurangan dari Algoritma *Greedy* adalah algoritma ini hasil akhirnya tidak sebaik algoritma lain seperti contoh nya algoritma *brute force*.

Masalah-masalah yang akan dihadapi dapat dipecahkan dengan melakukan pencarian dari pilihan-pilihan yang ada, didukung oleh cara pandangan penyelesaian yang biasa dilakukan. Proses pencarian merupakan proses yang amat sangat penting dalam mendapatkan solusi masalah yang dimana tidak ada lagi teknik langsung yang dimungkinkan untuk dipakai. Dalam kebanyakan masalah, ruang keadaan bertambah secara luar biasa seiring dengan penambahan jumlah keadaan yang dimungkinkan. Algoritma *Greedy* ini diharapkan dapat memecahkan masalah dalam pencarian jalur terpendek didalam *game maze escape* tersebut.

Pada tahun 2016, Apriyanto Apriyanto, Ishak Saputra Laso pernah meneliti tentang *game* labirin menggunakan aplikasi Construct 2 berbasis *online*, yang dimana penelitiannya menghasilkan *game* labirin yang dapat dimainkan secara *online* pada semua *web browser* yang menggunakan HTML 5[3].

Ada beberapa algoritma yang dapat digunakan untuk menemukan jalur terpendek dari *game maze escape* tersebut. Algoritma yang dapat digunakan diantaranya adalah Algoritma *Greedy*, Algoritma *Artificial Bee Colony*, Algoritma *Cheapest Insertion Heuristics*, Algoritma *Genetika* dan masih banyak yang lainnya[2].

2. METODOLOGI PENELITIAN

2.1 Game

Permainan (*Game*) merupakan salah satu bentuk hiburan yang dapat dijadikan sebagai penyemangat, dan menghilangkan rasa jenuh pikiran akibat dari padatnya aktivitas sehari-hari. Serta *game* juga merupakan salah satu aplikasi yang paling banyak dipakai dan dinikmati oleh pengguna media elektronik seperti (PC)[9].

2.2 Labirin

Labirin (*Maze*) merupakan permainan yang terbentuk dari suatu jalur-jalur yang saling berhubungan dan dibatasi oleh dinding-dinding yang memisahkan jalur yang satu dengan jalur yang lainnya. Jalur-jalur tersebut memiliki belokan dan terkadang merupakan jalur buntu. Labirin juga dapat diartikan sebagai permainan mencari jalur keluar dan kemudian pemain dilatih berkonsentrasi bagaimana cara menemukan jalan keluar didalam labirin tersebut. Pemain akan diletakkan disuatu tempat didalam labirin tersebut. Pemain akan mencapai finish ketika berhasil menemukan jalan keluar dari sebuah labirin tersebut. Pencarian jalur keluar akan dipersulit dengan adanya jalur belokan dan rintangan yang harus diselesaikan oleh pemain[11].

2.3 Algoritma Greedy

Algoritma *Greedy* merupakan metode yang populer untuk memecahkan persoalan optimasi. Hanya ada dua pilihan dalam persoalan optimasi, yaitu maksimasi dan minimasi. Algoritma *greedy* membentuk solusi langkah per langkah. Pada setiap langkah, terdapat banyak pilihan yang perlu di eksplorasi. Oleh sebab itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihannya. Pada setiap langkah merupakan pilihan, untuk membuat langkah optimum lokal dengan harapan bahwa langkah sisanya mengarah ke solusi optimum global[13]. Algoritma *greedy* merupakan algoritma yang memecahkan masalah langkah demi langkah[14], pada setiap langkah:

- a. Mengambil pilihan yang terbaik yang dapat diperoleh saat itu.
- b. Berharap bahwa dengan memilih optimum lokal pada setiap langkah akan mencapai optimum global. Algoritma *greedy* mengasumsikan bahwa optimum lokal merupakan bagian dari optimum global.





3. HASIL DAN PEMBAHASAN

3.1 Analisa Masalah

Misi pada *game* ini adalah seekor tikus harus mendapatkan keju seperti yang diminta oleh bos tikus. saat misi selesai, sitikus harus keluar dari labirin tersebut secepat mungkin karna dikejar oleh waktu. Selain itu sitikus tidak boleh tertangkap oleh sikucing. saat tertangkap, pemain akan langsung kalah dalam permainan.

Disetiap level, bos tikus akan lebih banyak meminta keju pada si tikus dan tantangan pun akan menjadi lebih sulit. Perancangan suatu *game* tidak bisa lepas dari yang namanya komponen pembentuk *game*, dalam permainan *maze escape* ini terdapat beberapa komponen yang akan dibuat didalam *game*. Berikut ini adalah komponen yang ada didalam pembentukan *game*.

Tabel 1. Komponen Dalam *Game*

No	Nama	Gambar	Keterangan
1	Tikus		Karakter utama sitikus yang mencari keju didalam labirin
2	Bos Tikus		Karakter pendukung sebagai bos yang memberikan misi pada si tikus
3	Kucing		Kucing yang menjaga keju di dalam labirin rumah.
4	Keju		Keju sebagai poin saat sitikus mengambilnya

Pada permainan maze escape ini terdapat dua level yang dibentuk berdasarkan kesulitan, waktu serta jumlah keju yang akan di ambil oleh si tikus untuk menyelesaikan permainan. Berikut ini tingkah kesulitan dalam permainan ini.

Tabel 2. Tingkat Kesulitan Setiap Level

Tingkat Level	Waktu	Jumlah Keju yang harus dikumpulkan	Jumlah Kucing
Level 1	1 menit	5 keju	1 kucing
Level 2	2 menit	7 keju	2 kucing

3.1.1. Penerapan Algoritma *Simplified Memory Bounded A**

Dalam menentukan jalur yang paling optimal dalam game ini, penulis melakukan langkah-langkah yaitu menggambarkan jalur-jalur yang akan dilalui oleh user dengan diagram jaringan dengan menggunakan teori graf, kemudian mencari jalur yang paling optimal dengan menggunakan algoritma greedy.

Permainan maze escape ini merupakan *game* meloloskan diri dari labirin, didalam *game* maze escape ini terdapat karakter yang terjebak di dalam labirin. Algoritma pencarian jalur terpendek ini dengan menggunakan algoritma *greedy*. Algoritma *greedy* merupakan salah satu algoritma yang biasa digunakan untuk memecahkan masalah optimasi. Algoritma *greedy* juga merupakan algoritma yang dapat menentukan sebuah jalur terpendek yang akan digunakan dengan mengambil secara terus menerus dan menambahkan ke dalam jalur yang akan dilewatinya.



Gambar 1. Tampilan Permainan Maze Escape

Pada gambar 1. dapat dilihat sebuah labirin, dari labirin tersebut terdapat jalan buntu dan juga kucing penjaga keju, dan tikus harus bisa melewati itu semua agar bisa mencapai ketujuan dan menyelesaikan misi dari bos tikus. Bos tikus meminta tikus agar mengambil 5 keju didalam labirin tersebut dalam waktu tertentu. Dan bos tikus memperingati adanya kucing penjaga keju agar jangan sampai tertangkap oleh si kucing seperti gambar berikut.



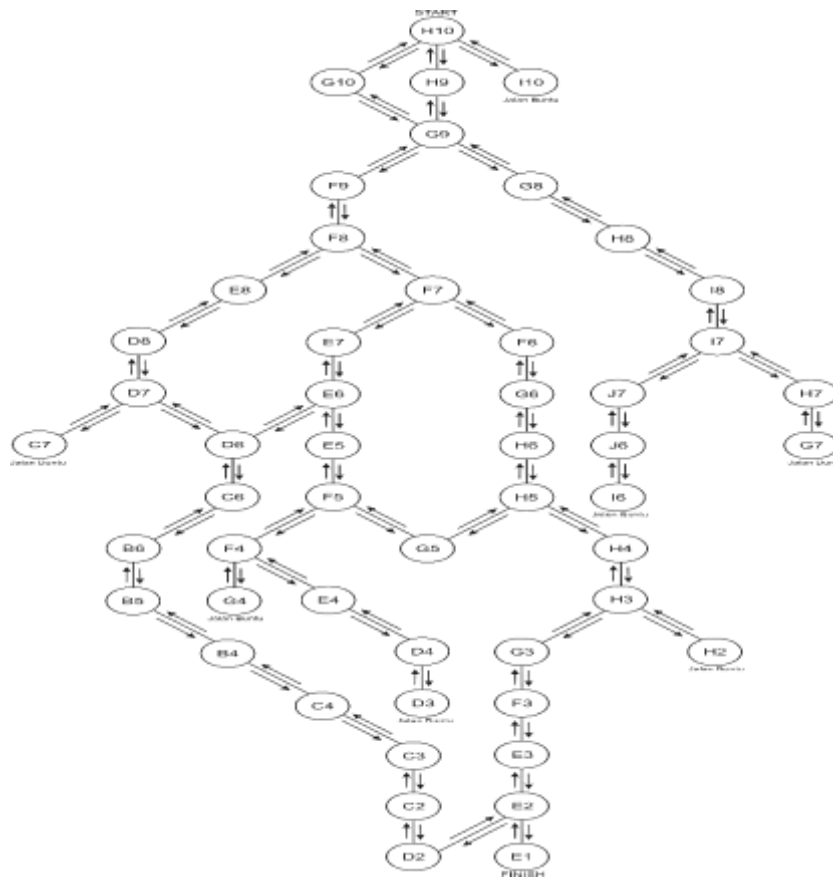
Gambar 2. Perintah Bos Tikus

Penyelesaian untuk menentukan jalur terpendek untuk sampai ke tujuan, langkah pertama yang dilakukan oleh penulis adalah membuat graf secara keseluruhan dari jalur game antara garis start sampai garis finish. Pada gambar 1, yang merupakan gambar peta dari jalur penelusuran game, maka penulis membuat graf jalur penelusuran untuk pencarian jalur terpendek dengan menggunakan algoritma greedy yang sesuai pada gambar tersebut, tujuan dari membuat sketsa jalur pada game adalah agar penulis menentukan dan memecahkan masalah dalam pencarian jalur terpendek, dan juga untuk memaparkan secara keseluruhan tahapan-tahapan dalam proses penyelesaian pencarian jalur terpendek pada game.



Gambar 3. Tampilan Game

Dari gambar diatas, terdiri dari 3 sampel jalur, 3 sampel ini dinyatakan sebagai jalur untuk sampai ke garis finish pada gambar 3. Tahap selanjutnya adalah membuat algoritma penelusuran struktur graf berdasarkan dari peta jalur penelusuran sebagai tahap proses pencarian jalur terpendek seperti gambar berikut.



Gambar 4. Graf Game

Lintasan pertama yang harus dipilih oleh block H10 yaitu block G10, H9, I10 yang ditunjukkan pada gambar 4.

Setelah block H10 memilih lintasan ke block H9, maka lintasan selanjutnya yang harus dipilih oleh block H9 yaitu block G9.

Lintasan selanjutnya yang harus dipilih oleh block G9 yaitu block F9, G8.

Setelah block G9 memilih lintasan ke block G8, maka lintasan selanjutnya yang harus dipilih oleh block G8 yaitu block H8.

Lintasan selanjutnya yang harus dipilih oleh block H8 yaitu block I8.

Setelah block H8 memilih lintasan ke block I8, maka lintasan selanjutnya yang harus dipilih oleh block I8 yaitu block I7.

Lintasan selanjutnya yang harus dipilih oleh block I7 yaitu block J7, H7 yang pertama.

Setelah block I7 memilih lintasan ke block H7, maka lintasan selanjutnya yang harus dipilih oleh block H7 yaitu block G7 agar user mendapatkan keju.

.Lintasan selanjutnya yang harus dipilih oleh block G7 yaitu block H7.

Setelah block G7 memilih lintasan ke block H7, maka lintasan selanjutnya yang harus dipilih oleh block H7 yaitu block I7.

Lintasan selanjutnya yang harus dipilih oleh block I7 yaitu block J7.

Setelah block I7 memilih lintasan ke block J7, maka lintasan selanjutnya yang harus dipilih oleh block J7 yaitu block J6.

Lintasan selanjutnya yang harus dipilih oleh block J6 yaitu block I6.

Setelah block J6 memilih lintasan ke block I6, maka lintasan selanjutnya yang harus dipilih oleh block I6 yaitu block H6.

Lintasan selanjutnya yang harus dipilih oleh block H6 yaitu block H5.

Setelah block H6 memilih lintasan ke block H5, maka lintasan selanjutnya yang harus dipilih oleh block H5 yaitu block G5.

Lintasan selanjutnya yang harus dipilih oleh block G5 yaitu block F5.

Setelah block G5 memilih lintasan ke block F5, maka lintasan selanjutnya yang harus dipilih oleh block H5 yaitu block F4.

Begitu seterusnya sampai karakter(tikus) sampai kegaris finish dan menyelesaikan permainan.

Dari hasil penelusuran diatas yang terdapat pada gambar 3.10 maka dapat dengan jelas di lihat bahwa proses penelusuran pencarian jalur terpendek dimulai dari grid H10, H9, G9, G8, H8, I8, I7, H7, G7, lalu kembali lagi ke H7, I7, J7, J6, I6, H6, H5, G5, F5, F4, G4, lalu kembali lagi ke F4, F5, G5, H5, H4, H3, H2, lalu kembali lagi ke H3, G3, F3, E3, E2, E1.

3.2. Implelentasi

Dalam sebuah sistem agar dapat berjalan seperti yang diinginkan karena system komputerisasi tidak dapat dipisahkan antara *hardware* dan *software*. Demikian juga sistem yang dirancang dengan spesifikasi hardware minimal sebagai berikut:

1. Processor Minimal Pentium Dual-Core
2. Harddisk 500 GB
3. RAM 2 GB

Sistem ini agar dapat berjalan tidak lepas dari software yang jelas mendukung hardware diatas perangkat lunak seperti Sistem Operasi dan program aplikasi seperti berikut:

1. Sistem Operasi Windows XP, Windows Vista, Windows 7.
2. Construct 2.

Desain pengujian sistem merupakan gambaran program ketika dirancang didalam bahasa pemrograman, disini aplikasi dirancang menggunakan Construct 2

Pada bab ini, penulis akan menampilkan tampilan hasil perancangan yang telah dijelaskan pada bab sebelumnya, yang terdiri dari lima tampilan seperti berikut:

Tampilan *opening* merupakan tampilan yang muncul setelah membuka aplikasi game pertama kali.



Gambar 5. Tampilan *Opening Game*

Tampilan menu utama merupakan tampilan yang muncul setelah menekan tombol opening game. Pada tampilan ini disediakan 4 tombol menu, yaitu menu *Start* untuk memulai game baru, menu *Help* untuk menampilkan petunjuk permainan, menu *About* untuk menampilkan tentang sipembuat program, menu *Exit* untuk mengakhiri game sebelum game ini dimulai.



Gambar 6. Tampilan Menu Utama

Tampilan menu level merupakan tampilan yang muncul setelah menekan tombol start. Pada tampilan ini disediakan 3 tombol menu, yaitu menu Level 1 untuk memulai game dengan tingkat level 1, menu Level 2 untuk memulai game dengan tingkat level 2, menu back untuk kembali ke tampilan menu utama.



Gambar 7. Tampilan Menu Level



Gambar 8. Tampilan *Game Level 1*



Gambar 9. Tampilan *Game Level*

4. KESIMPULAN

Setelah menyelesaikan perancangan perangkat lunak permainan game maze escape, penulis menarik kesimpulan Pencarian jalur terpendek sangat berguna untuk pencarian jalan tersingkat menuju garis finish. Sehingga, kita dapat sampai tepat waktu menuju tempat tujuan. Algoritma greedy dapat diimplementasikan pada kasus pencarian jalur terpendek yang memiliki jarak antara block satu dengan block yang lain dan algoritma greedy ini hanya diimplementasikan kedalam game maze escape saja

REFERENCES

- [1] W. Wibawanto, *Memprogram Game Flash 3D Itu Mudah*, Yogyakarta: C.V Andi Offset, 2013.
- [2] G. Aristi, P. Studi, and M. Informatika, "PERBANDINGAN ALGORITMA GREEDY , ALGORITMA CHEAPEST INSERTION HEURISTICS DAN DYNAMIC PROGRAMMING DALAM PENYELESAIAN TRAVELLING," vol. XVI, no. 2, pp. 52-58, 2014.
- [3] A. Apriyanto and I. S. Lasodi, "PEMBUATAN GAME LABIRIN MENGGUNAKAN APLIKASI CONSTRUCT 2 BERBASIS ONLINE," vol. 2, no. 2, 2016.
- [4] R. Ahmad Shukri Mohd Nain dan Amran Md, *Pengurusan Teknologi*, Johor Darul Ta'zim: Universiti Teknologi Malaysia, 2005.
- [5] A. a. S. M. Rosa, *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*, Bandung: Informatika Bandung, 2013.
- [6] P. P. a. H. Widodo, *Menggunakan UML*, 1 ed., Bandung: INFORMATIKA, 2011.c
- [7] Dedynggego, Mohammad, and Moh.Affan, "Perancangan Media Pembelajaran Interaktif 3D Tata Surya Menggunakan Teknologi Augmented Reality Untuk Siswa Kelas 6 Sekolah Dasar Sangira," *J. Elektron. Sist. Inf. Dan Komput.*, vol. 1, no. 2, pp. 45-60, 2015.
- [8] A. Juansyah, "PEMBANGUNAN APLIKASI CHILD TRACKER BERBASIS ASSISTED – GLOBAL POSITIONING SYSTEM (A-GPS) DENGAN PLATFORM ANDROID Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)," *J. Ilm. Komput. dan Inform.*, vol. 1, no. 1, pp. 1-8, 2015.
- [9] L. B. Masalah, "APLIKASI MOBILE INFORMASI KAFE 24 JAM DI YOGYAKARTA BERBASIS ANDROID Pendahuluan Landasan Teori Analisis dan Perancangan Sistem," vol. 14, no. 4, pp. 49-53, 2013.
- [10] M. A. Dewi, V. D. Anggraeni, S. A. Mudjadi, and A. Wicaksono, "Aplikasi Rekapitulasi Elektronik Absensi Guru & Pegawai (Area-Gp) Pada Sekolah Menengah Atas," *Skripsi, STMIK Raharja Tangerang*, vol. 2014, no. Sentika, 2014.
- [11] M. A. Murotul and H. Tahun, "Simki-Techsain Vol. 01 No. 01 Tahun 2017 ISSN : XXXX-XXXX," *Simki-Techsain*, vol. 1, no. 1, pp. 1-7, 2017.
- [12] K. T. Martono, "Pengembangan Game Dengan Menggunakan Game Engine Game Maker," *J. Sist. Komput.*, vol. 5, no. 1, pp. 23-30, 2015.
- [13] S. G. A. Bhaskara, "Permainan Edukasi Labirin Virtual Reality Dengan Metode Collision Detection Dan Stereoscopic," *Lontar Komput.*, vol. Volume 8, no. Nomor 2, p. Halaman 65-76, 2017.
- [14] G. G. Maulana, "Pembelajaran Dasar Algoritma Dan Pemrograman Menggunakan El-Goritma Berbasis Web," *J. Tek. Mesin*, vol. 6, no. Spesial, pp. 69-73, 2017.
- [15] A. E. Ghozali, B. D. Setiawan, and M. T. Furqon, "Aplikasi Perencanaan Wisata di Malang Raya dengan Algoritma Greedy," vol. 1, no. 12, pp. 1459-1467, 2017.
- [16] E. N. Hayati and A. Yohanes, "Pencarian Rute Terpendek Menggunakan Algoritma Greedy," *Semin. Nas. IENACO*, pp. 2337-4349, 2014.
- [17] A. Apriyanto and I. S. Lasodi, "PEMBUATAN GAME LABIRIN MENGGUNAKAN APLIKASI CONSTRUCT 2 BERBASIS ONLINE," vol. 2, no. 2, 2016.