

GIM PLATFORMER “PINA SMART ADVENTURE” DENGAN PEMBANGKITAN LEVEL PROCEDURAL DUA FASE

Mochamad Gian Razan¹⁾, Pratama Wirya Atmaja²⁾, Afina Lina Nurlaili³⁾
E-mail : ¹⁾mgianr.if@gmail.com , ²⁾pratama_wirya.fik@upnjatim.ac.id , ³⁾
afina.lina.if@upnjatim.ac.id

^{1,2,3} Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan
Nasional “Veteran” Jawa Timur

Abstrak

Sering berkembangnya teknologi, semakin bertambah peminat dalam bidang teknologi dan juga semakin tinggi minat kepuasan orang untuk mencukupi kebutuhan. Salah satu bidang teknologi yang diminati oleh banyak orang adalah game. *Game* merupakan media teknologi dalam bentuk 2D ataupun 3D untuk bersenang-senang. *Platform game* adalah yang dalam gameplay nya melibatkan perjalanan antar platform dengan cara melompat (biasanya juga berayun dan memantul). Genre ini biasanya dihubungkan dengan tokoh-tokoh kartun seperti Sonic The Hedgehog, Mario, dan Rayman, walaupun mungkin mempunyai tema yang lainnya. Elemen-elemen tradisional dari game ini termasuk berlari, lompat, dan memanjat tangga atau pijakan. Genre ini seringkali meminjam elemen dari genre lain seperti fighting atau perkelahian dan shooting atau tembak-menembak. *Procedural Content Generation* (PCG) merupakan algoritma untuk membuat konten secara otomatis, alih-alih yang mengandalkan konten yang dirancang oleh pengguna. Sementara pendekatan PCG secara tradisional berfokus pada pembuatan konten untuk permainan video, mereka diterapkan pada semua jenis lingkungan virtual, sehingga memungkinkan pelatihan sistem pembelajaran mesin yang secara signifikan lebih umum. Sebagai contoh, kemampuan PCG untuk menghasilkan aliran level baru yang tidak pernah berakhir telah membuat agen *Capture the Flag Deep Mind* dapat melampaui kinerja manusia. Selain itu, metode yang terinspirasi PCG seperti pengacakan domain memungkinkan lengan robot OpenAI untuk belajar memanipulasi objek dengan ketangkasan yang belum pernah terjadi sebelumnya. Maka dari itu pengujian akan membuat sebuah platform game dengan menggunakan algoritma *Procedural Content Generation* (PCG) yang berjudul Pina Smart Adventure.

Kata kunci: *Game, Platform game, Procedural Content Generation (PCG)*

1. PENDAHULUAN

Game saat ini sudah menjadi alternatif hiburan bagi kalangan anak-anak, remaja, orang tua, laki-laki maupun perempuan. Industri bisnis pengembangan *game* juga sudah menjadi suatu hal yang menjanjikan, terbukti dengan banyaknya perusahaan pengembangan *game* di Amerika, Eropa dan Asia. Khususnya *game mobile* baik *Android* dan *Ios* yang saat ini juga di perbincangkan. Negara Indonesia masih terhitung sebagai konsumen *game*, dilihat dari konsumsi *game* yang sangat tinggi [1].

Game Platformer 2D khususnya merupakan *genre game* klasik yang dinamakan dari *gameplay-nya* di mana karakter pemain melompat diatas di atas *platform* untuk melewati rintangan yang ada. *Genre* biasanya dipadukan dengan *genre* lainnya seperti *shooter, puzzle, maupun action*. *Genre* ini sudah mulai semakin populer di waktu

sekarang ini, di tahun 2010-an semenjak kepopuleran *game mobile* dan banyaknya *game platformer* yang muncul di *platform* tersebut [3]. Melihat keadaan industri *video game* saat ini, pengembangan *game platform* menjadi salah satu pilihan yang terbaik untuk membuat *game*. Pengembangan *game* seperti ini dapat meminimalisir waktu dan biaya daripada pembuatan *game* 3D maupun bentuk *game* lainnya [2].

Procedural Content Generation adalah sebuah metode dimana konten dibuat oleh komputer dengan suatu algoritma [4]. Pertama kali muncul pada tahun 1980 oleh Michael Toy dan Glenn Wichman's [4]. Walaupun dapat digunakan untuk berbagai hal, termasuk memberi variasi, mengurangi waktu pembuatan dan biaya pembuatan menghemat ruang memori di disk, menambah potensi kreativitas para *developer* dan memberi adaptasi ke *games* [4] tidak banyak riset dan perkembangan yang dilakukan terhadap *Procedural Content Generation* dimana *workshop* pertama mengenai topik ini diadakan pada tahun 2009 dan jurnal pertama mengenai *Procedural Content Generation* pada tahun 2011 [4]. Algoritma ini biasanya dipakai untuk membuat konten secara otomatis, alih-alih yang mengandalkan konten yang dirancang oleh pengguna. Sementara pendekatan PCG secara tradisional berfokus pada pembuatan konten untuk permainan video, mereka diterapkan pada semua jenis lingkungan virtual, sehingga memungkinkan pelatihan sistem pembelajaran mesin yang secara signifikan lebih umum. Algoritma *Procedural Content Generation* memiliki permasalahan dalam menghasilkan level yang tidak ada temanya. Contoh *game* yang memakai algoritma *Procedural Content Generation* yaitu *Temple Run*, *Cookie Run*, *Endless Runner Game*.

Namun pada metode *Procedural Content Generation* dalam menghasilkan level - level yang tidak bertema atau setiap potongan-potongan level tidak memiliki sebuah tema (Abstrak). Oleh karena itu pada penelitian ini penulis membuat *game platformer* 2D berjudul “Pina Smart Adventure” dengan menerapkan level yang acak dan bertema menggunakan metode *Procedural Content Generation* 2 fase yang terdiri dari fase pertama yaitu menciptakan potongan-potongan besar level yang memiliki tema yang ditentukan, lalu pada fase kedua menciptakan potongan – potongan kecil untuk mengisi dari potongan - potongan besar tersebut.

2. LANDASAN PERPUSTAKAAN

2.1. Game Platformer

Game platformer 2D khususnya merupakan *genre game* klasik yang dinamakan dari *gameplay-nya* di mana karakter pemain melompat di atas *platform* untuk melewati rintangan yang ada. *Genre* ini sering digabungkan dengan *genre* lainnya seperti *shooter*, *puzzle*, maupun *action*. *Genre* ini juga mulai semakin populer di waktu sekarang ini, di tahun 2010-an semenjak kepopuleran *mobile gaming* dan banyaknya *game platformer* yang muncul di *platform* tersebut [3].

Game Platformer adalah *genre game* yang dinamakan dari *gameplay-nya* di mana karakter pemain melompat di atas *platform*, melewati rintangan yang ada dan melewati atau membunuh musuh yang ada untuk mencapai garis akhir *game* tersebut.

2.2. Procedural Content Generation (PCG)

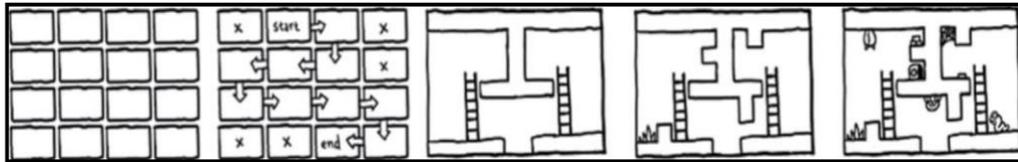
Procedural Content Generation adalah sebuah metode dimana konten dibuat oleh komputer dengan suatu algoritma [4]. Pertama kali muncul pada tahun 1980 oleh Michael Toy dan Glenn Wichman's [4]. Walaupun dapat digunakan untuk berbagai hal, termasuk memberi variasi, mengurangi waktu pembuatan dan biaya pembuatan menghemat ruang memori di disk, menambah potensi kreativitas para *developer* dan memberi adaptasi ke *games* [4] tidak banyak riset dan perkembangan yang dilakukan terhadap *Procedural Content Generation* dimana *workshop* pertama mengenai topik ini diadakan pada tahun 2009 dan jurnal pertama mengenai *Procedural Content Generation* pada tahun 2011 [4].

Procedural Content Generation adalah sebuah metode dimana konten dibuat oleh

komputer dengan suatu algoritma dengan secara otomatis.

2.3. Procedural Content Generation 1 Fase

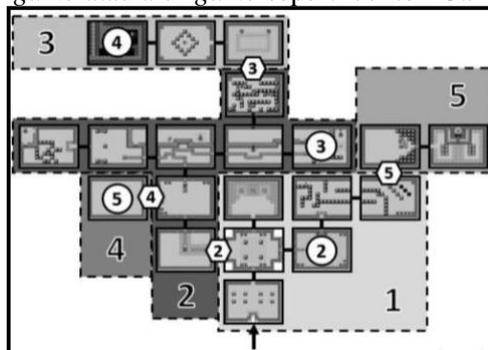
Procedural Content Generation 1 Fase adalah suatu fase dimana membuat potongan – potongan *template game* atau alur *game* dari mulainya *game* sampai selesainya *game* secara berhubungan seperti contoh Gambar 1 dibawah ini.



Gambar 1. Proses pembuatan level Spelunky, dari kiri ke kanan: kisi kamar 4x4, penentuan jalur solusi, pemilihan templat kamar, penambahan hambatan, dan penyisipan harta dan musuh [5].

2.4. Procedural Content Generation 2 Fase

Procedural Content Generation 2 Fase adalah suatu fase di mana membuat potongan – potongan besar *template game* atau alur *game* dari mulainya *game* sampai selesainya *game* secara berhubungan yang nantinya akan diisi dengan potongan – potongan kecil *template game* atau alur *game* seperti contoh Gambar 2 dibawah ini.



Gambar 2. Gnarled Root Dungeon, dari The Legend of Zelda: Oracle of Seasons [6].

3. METODOLOGI

Pada penelitian kali ini, proses pembuatan aplikasi dibuat ke dalam 6 proses utama, ke-enam proses utama itu adalah:

1. Menciptakan potongan - potongan kecil
2. Membuat larangan - larangan penempatan potongan
3. Menciptakan potongan – potongan besar
4. Pembacaan data potongan besar dan potongan kecil
5. Meletakkan potongan besar pada level
6. Meletakkan potongan kecil pada level

Setelah selesai membuat keenam proses utama yang sudah dijabarkan sebelumnya, maka lanjut dalam proses pengujian aplikasi.

3.1. Menciptakan Potongan – Potongan Kecil

Potongan – potongan kecil adalah potongan yang berisikan objek-objek dari elemen permainan. Potongan kecil merupakan daftar dari potongan-potongan yang akan muncul pada *level* tersebut. Potongan kecil memiliki parameter objek dan parameter larangan yang berfungsi untuk penempatan potongan tersebut pada *level*.

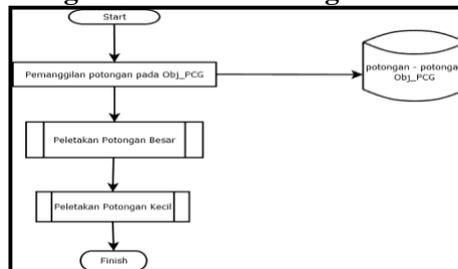
3.2. Membuat Larangan - Larangan Penempatan Potongan

Larangan potongan adalah larangan-larangan potongan yang muncul disebelah, diatas atau dibawah potongan yang di deklarasikan. Terdapat 8 larangan yang digunakan untuk penempatan potongan – potongan yang di deklarasikan sebagai berikut, larangan kiri, larangan bawah, larangan kanan , larangan atas yang gunanya untuk melarang potongan agar tidak muncul disebelah larangan potongan yang dideklarasikan supaya pemain tidak menemukan jalan tertutup. Adapun deklarasi larangan dengan parameter Boolean yang hanya dapat diisi dengan *true or false* yaitu darat saja, udara saja, paling kanan dan paling kiri.

3.3. Menciptakan Potongan - Potongan Besar

Potongan - potongan besar adalah potongan yang bariskan potongan - potongan kecil yang sudah diatur dalam deklarasi potongan besar. Potongan besar merupakan daftar dari potongan-potongan yang akan muncul pada *level* tersebut. Potongan besar memiliki parameter panjang, parameter tinggi, parameter potongan kecil dan parameter larangan yang berfungsi untuk penempatan potongan tersebut pada *level*.

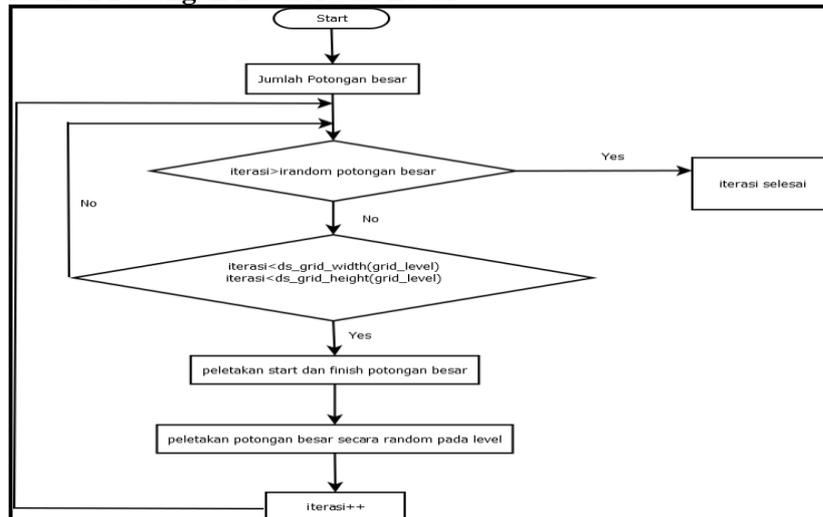
3.4. Pembacaan Data Potongan Besar dan Potongan Kecil



Gambar 3. Flowchart pembacaan potongan

Seperti Gambar 3 Flowchart pembacaan potongan diatas. Setelah membuat objek-objek pada potongan-potongan, dilakukan pemanggilan data potong-potongan pada Obj_PCG untuk meletakkan potongan-potongan yang sudah ada aturan melalui Obj_PCG.

3.5. Meletakkan Potongan Besar Pada Level



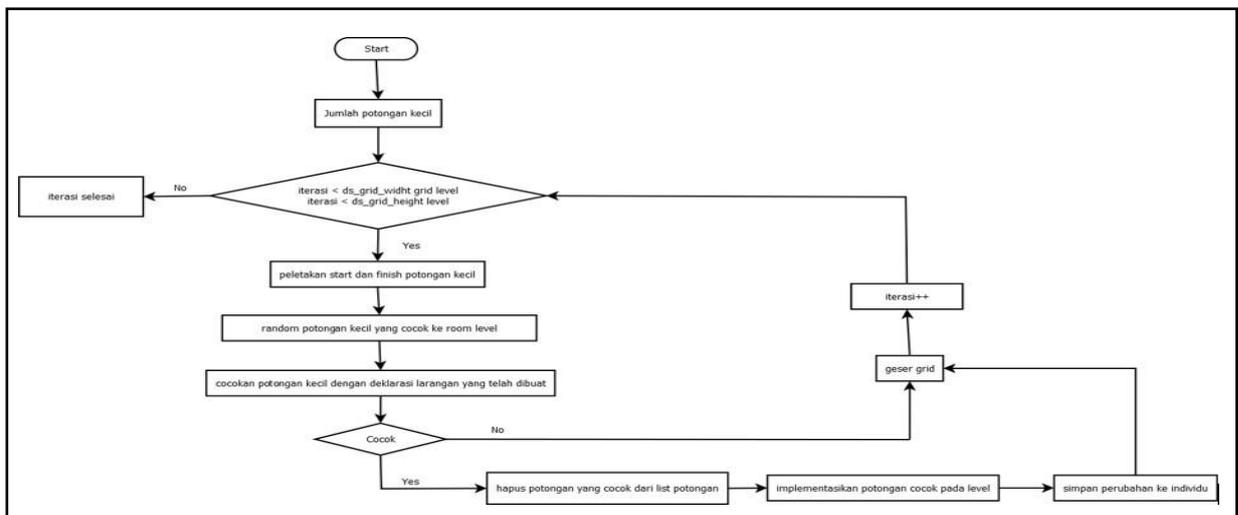
Gambar 4. Flowchart peletakan potongan besar pada level

Seperti Gambar 4 Flowchart di atas, setelah membuat deklarasi potongan-potongan dan aturan larangan, maka lanjut ke peletakan potongan besar. Pertama melihat jumlah potongan besar, lalu jika iterasi < irandom potongan besar (irandom potongan besar pada aturan ini adalah 3+irandom(2)), maka lanjut ke proses selanjutnya yaitu jika iterasi < dari lebar dan tinggi ds_grid_level, maka lanjut ke proses peletakan start dan finish

potongan besar dan jika lebih besar maka kembali ke proses iterasi < irandom potongan besar, setelah itu peletakan potongan besar secara random pada level, setelah iterasi berlooping hingga iterasi > irandom potongan besar maka proses akan selesai.

3.6. Meletakkan Potongan Kecil Pada Level

Setelah peletakan potongan besar, maka lanjut ke peletakan potongan kecil. Pertama melihat jumlah potongan kecil, lalu jika iterasi < panjang dan lebar ds_grid_level (bagian-bagian yang masih kosong), lanjut ke proses peletakan start dan finish, setelah itu proses peletakan potongan kecil secara random ke dalam room, lalu proses pencocokan larangan jika tidak cocok maka geser ke grid selanjutnya dan jika cocok potongan kecil tersebut maka lanjut ke proses penghapus potongan tersebut dari list potongan, setelah itu lanjut ke proses implementasi potongan tersebut pada level, lalu simpan perubahannya jika sudah geser gridnya lalu iterasi akan berlooping hingga iterasi > Panjang dan lebar ds_grid_level. Seperti Gambar 5 Flowchart berikut.



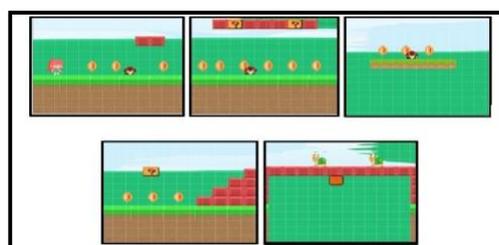
Gambar 5. Flowchart peletakan potongan kecil pada level

4. HASIL DAN PEMBAHASAN

Selain memaparkan hasil pengujian baik pengujian survey maupun pengujian fitness, demonstrasi hasil proses PCG dan alur dari program yang sudah jadi pun juga di tampilkan.

4.1. Potongan kecil

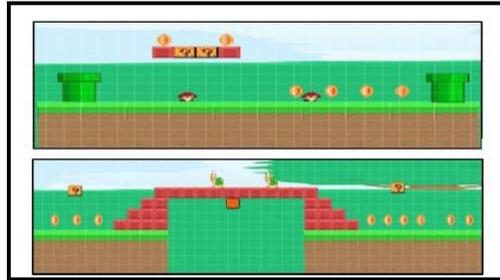
Setelah mendeklarasikan potongan – potongan kecil akan menghasilkan potongan - potongan kecil yang akan mengisi room level secara aturan larangan yang sudah dibuat. Berikut hasil dari deklarasi potongan kecil. Berikut Gambar 11 adalah contoh potongan – potongan kecil.



Gambar 11. Contoh potongan kecil

4.2. Potongan besar

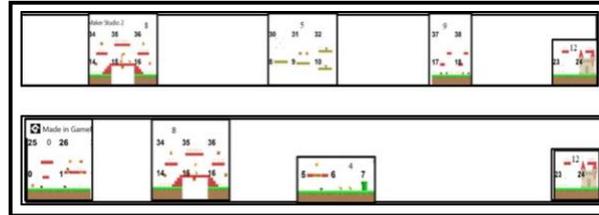
Setelah mendeklarasikan potongan – potongan kecil maka terbentuklah potongan – potongan besar, potongan besar akan menghasilkan potongan-potongan kecil yang akan mengisi room *level* yang sudah diatur dalam deklarasi potongan besar yang sudah ditentukan temanya. Berikut hasil dari deklarasi potongan besar. Berikut Gambar 12 adalah contoh potongan – potongan besar



Gambar 12. Contoh potongan besar

4.3. Peletakan potongan besar

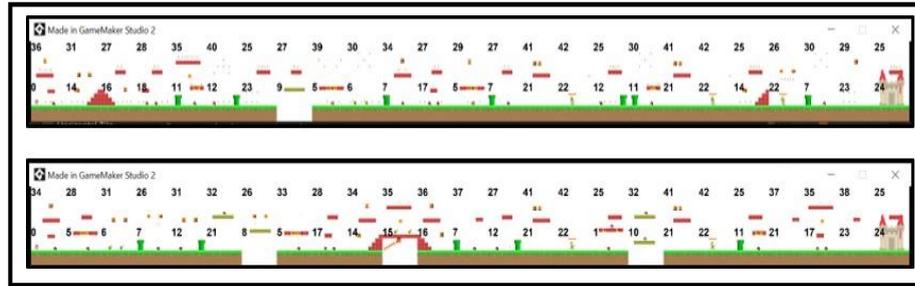
Setelah membuat potongan-potongan dan aturan larangan nya, maka di lanjut dengan peletakan potongan-potongan besar berdasarkan irandomnya ($3 + \text{irandom}(2)$). Pertama melihat jumlah potongan besar, lalu jika iterasi $< \text{irandom}$ potongan besar (irandom potongan besar pada aturan ini adalah $3 + \text{irandom}(2)$), jika iterasi $<$ dari lebar dan tinggi ds_grid_level maka lanjut ke proses peletakan *start* dan *finish* potongan besar, setelah itu peletakan potongan besar secara random pada *level*, setelah iterasi berlooping hingga iterasi $> \text{irandom}$ potongan besar maka proses akan selesai. Berikut merupakan contoh-contoh peletakan potongan besar. Berikut Gambar 13 merupakan contoh-contoh peletakan potongan besar.



Gambar 13. Peletakan potongan besar

4.4. Peletakan potongan kecil

Setelah peletakan potongan besar selesai, maka di lanjut dengan peletakan potongan-potongan kecil. Pertama melihat jumlah potongan kecil, lalu jika iterasi $<$ Panjang dan lebar ds_grid_level (bagian-bagian yang masih kosong), lanjut ke proses peletakan *start* dan *finish*, setelah itu proses peletakan secara random ke room, lalu proses pencocokan larangan jika tidak maka geser *grid* dan jika cocok potongan kecil tersebut maka lanjut ke proses penghapus potongan dari list potongan, setelah lanjut ke proses implementasi potongan tersebut pada *level* lalu simpan perubahannya jika sudah geser gridnya lalu iterasi akan berlooping hingga iterasi $>$ Panjang dan lebar ds_grid_level . Berikut contoh-contoh peletakan potongan kecil. Berikut Gambar 14 merupakan contoh-contoh peletakan potongan kecil.



Gambar 14. Peletakan potongan kecil

4.5. Pengujian Metode Procedural Content Generation menggunakan Analitik Heat Map (Peta Panas)

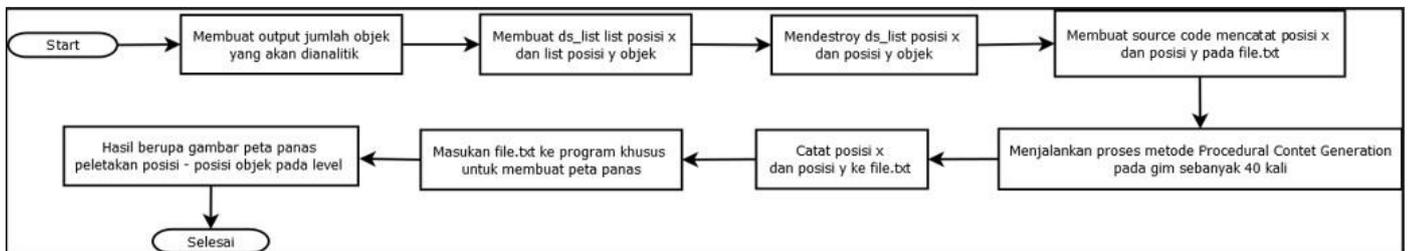
Pada pengujian metode Procedural Content Generation dilakukannya metode analitik gim dengan visualisasi dengan peta panas (Heat Map) dengan mencoba menjalankan metode Procedural Content Generation sebanyak 40 kali untuk menciptakan level. Yang akan dianalisis adalah berupa objek 3 jenis yaitu item, musuh dan platform. Adapun skema warna penunjukan peletakan objek seperti gambar 6 berikut:



Gambar 6. Skema warna hasil heat map

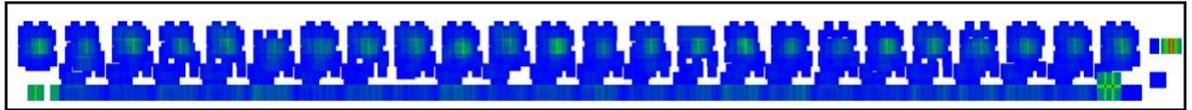
Pada Gambar 6 dapat disimpulkan sebagai berikut:

1. Indikator Biru: wilayah yang jarang ditempati oleh objek.
2. Indikator Hijau: wilayah yang lebih sering ditempati oleh objek.
3. Indikator Merah: wilayah yang paling sering ditempati oleh objek Berikut proses atau langkah – langkah analitik menggunakan heat map.



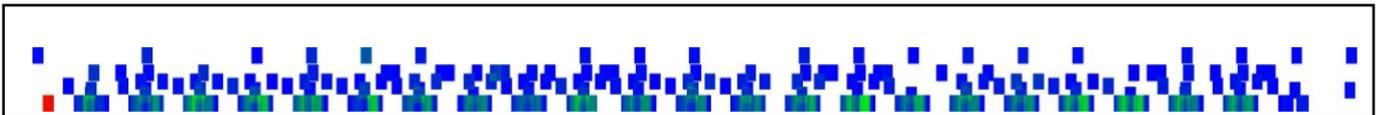
Gambar 7. Flowchart proses analitik

Dari Gambar 7 Flowchart proses analitik peletakan objek menggunakan peta panas diatas langkah pertama yang dilakukan adalah membuat output jumlah objek yang akan dianalisis menggunakan fungsi `instance_number`, setelah membuat output jumlah objek lanjut membuat `ds_list` posisi x dan posisi y objek, lanjut `mendestroy ds_list` posisi x dan posisi y objek, setelah itu membuat source code mencatat posisi x dan posisi y ke `file.txt` berikut source code untuk mencatat posisi x dan posisi y. Setelah membuat source code mencatat posisi x dan posisi y ke `file.txt`, setelah itu jalankan proses metode Procedural Content Generation sebanyak 40 kali dengan cara me-restart game, setelah itu lanjut ke proses mencatat posisi x dan posisi y ke `file.txt` dengan source code yang telah dibuat, setelah itu masukan `file.txt` ke dalam program khusus untuk membuat peta panas, setelah dijalankan program khusus tersebut maka mendapatkan hasil berupa gambar peta panas peletakan objek – objek pada level, berikut hasil analitik peta panas (Heat Map) objek 3 jenis.



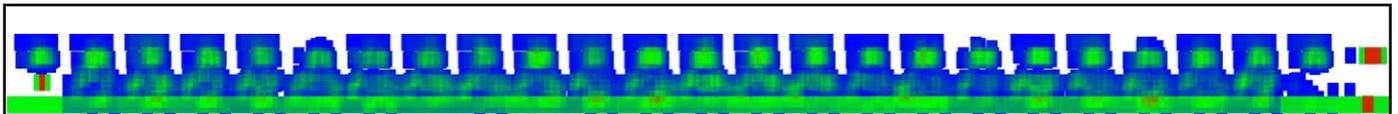
Gambar 8. Analitik peta panas objek item

Seperti Gambar 8 diatas terlihat terdapat warna merah di ujung kanan oleh karena itu maka objek item paling sering diletakkan oleh metode Procedural Content Generation pada ujung kanan level, hal ini disebabkan pada pembuatan deklarasi objek item pada level terdapat pada potongan finish, pada pembentukan level potongan start dan finish selalu didahulukan oleh karena itu objek item sering muncul pada bagian kanan level (bagian finish level) dan terdapat warna hijau di bagian bawah kiri, kanan dan merata di atas level menunjukkan bahwa objek item lebih sering diletakkan ditempat tersebut.



Gambar 9. Analitik peta panas objek musuh

Seperti Gambar 9 diatas terlihat terdapat warna merah di ujung kiri bawah dapat disimpulkan bawah objek musuh paling sering diletakkan oleh metode Procedural Content Generation pada ujung kiri bawah level, hal ini disebabkan oleh pembuatan deklarasi objek musuh pada level terdapat pada potongan start, pada pembentukan level potongan start dan finish selalu didahulukan oleh karena itu objek musuh sering muncul pada bagian kiri level (bagian start level) dan terdapat warna hijau merata di bagian bawah level hal tersebut dapat disimpulkan objek musuh juga lebih sering diletakkan pada tempat tersebut.



Gambar 10. Analitik peta panas objek platform

Seperti Gambar 10 diatas terlihat terdapat warna merah di ujung kiri dan kanan dapat disimpulkan bahwa objek platform paling sering diletakkan oleh metode Procedural Content Generation pada ujung kiri dan kanan level, hal ini disebabkan oleh pembuatan deklarasi objek platform terdapat pada potongan start dan finish, pada pembentukan level potongan start dan finish selalu didahulukan oleh karena itu objek platform sering muncul pada bagian kanan dan kiri level (bagian start dan finish) dan terdapat warna hijau merata di bagian atas dan bawah level hal tersebut dapat disimpulkan objek musuh juga lebih sering diletakkan pada tempat tersebut.

Tabel 1. Hasil Kuesioner

No	Pernyataan	Nilai rata-rata
1	Sewaktu bermain, saya selalu tahu cara mencapai tujuan-tujuan permainannya.	3,97
2	Antarmuka gimnya mudah digunakan.	4,22
3	Saya dapat langsung memainkan gimnya tanpa harus membaca manual mendetail.	4,08

4	Informasi-informasi di dalam jalan permainan gimnya dapat dipahami dengan jelas.	4,25
5	Gimnya asyik dimainkan hingga membuat saya melupakan hal-hal di sekitar saya.	3,68
6	Saya tidak merasa capek sewaktu memainkan gimnya.	4,11
7	Tanpa sadar, saya memainkan gimnya lebih lama dari rencana saya.	3,82
8	Setelah saya selesai memainkan gimnya, saya merasa ingin memainkannya lagi.	3,91
	Nilai rata-rata total	4.01

4.6. Hasil Survey Kuesioner

Melalui survei yang sudah dilakukan melalui kuesioner memakai *google form* dengan 35 responden dengan 8 pertanyaan dengan skala linkert 1 (sangat tidak setuju) – 5 (sangat setuju).

Berikut hasil nilai rata-rata dan simpangan baku dari jawaban 8 pertanyaan dari 35 responden. Berikut Tabel 1 merupakan hasil kuesioner dari 8 pertanyaan yang telah dibuat oleh peneliti

Dari hasil kuesioner 35 responden kebanyakan sudah tau dengan permainan menggunakan metode *Procedural Content Generation* (pengacakan level) yaitu seperti *Minecraft*, *No Man’s Sky*. Lalu pada penilaian dari 8 pertanyaan mempunyai nilai rata-rata dengan nilai 4.01 jika dikonversikan ke skala linkert 4 yang berarti setuju untuk *game* ini,

5. KESIMPULAN DAN SARAN

Melalui pemaparan dari penelitian yang telah dilakukan, hasil penelitian mampu menjawab rumusan masalah yang telah dibuat. Dapat disimpulkan bahwa algoritma *Procedural Content Generation* dapat menghasilkan level-level yang bertema dari setiap potongan-potongan kecil maupun besar. Algoritma *Procedural Content Generation* ini dapat diterapkan pada *game platform* “Pina Smart Adventure” yang dibuat dengan aturan-aturan deklarasi dan larangan potongan yang dibuat. Selain menjawab rumusan masalah, penelitian juga menghasilkan sebuah *game* yang telah teruji menggunakan metode *Kuesioner* dengan penilaian rata-rata dari 35 responden yaitu 4.01.

Saran untuk penelitian ke depan algoritma ini sangat cocok bila potongan besar dan potongan kecil sedikit dikarenakan larangan terlalu banyak makan merandomnya akan monoton. Selain itu disarankan juga untuk menambahkan penambahan level dan konten-konten permainan yang seru dan asyik. Kalau membuat larangan jangan terlalu banyak akan membuat *game* memulai secara lama dikarenakan *gameya* akan *looping* hingga selesai semua larangannya.

6. DAFTAR RUJUKAN

- [1] Eugen Triyudi Prabowo "Rancang Bangun Game “Jumping Noob” Berbasis Windows Phone," Institusi Teknologi Nasional Malang.
- [2] Darius Andana Harris, Jeanny Pragantha, Muliadi. (2016) “PEMBUATAN GAME PLATFORMER “BEYOUND” MENGGUNAKAN UNITY DENGAN XBOX 360 CONTROLLER

- [3] Eriq M. Adam Jonemaro, Monel Lindu Sagala and Wibisono Sukmo Wardhono. (2017) “Pengembangan Game Platformer 2D Menggunakan Teknik Projection Mapping”, 1(11), hlmn 1160-1168
- [4] Wijaya, I Gusti Ngurah Taksu (2018) *Rancang Bangun Game First Person Shooter Dengan Procedural Content Generation Berbasis Virtual Reality*. Bachelor Thesis thesis, Universitas Multimedia Nusantara
- [5] David Stammer, Mike Preuss, Tobias Guther. (2015) “Player-Adaptive Spelunky Level Generetaion” 2015 IEEE Conference on Computational Intelligence and Games (CIG)
- [6] Daniel Manrique, Jose M. Font, Julian Togelius and Roberto Izquierdo. (2016) “Constrained Level Generetaion Throuht Grammar-Based Evolutionary Algorithms”. 9th European Conference on the Applications Of Evolutionary Computation. EvoApplications 2016.
- [7] G. Wallner dan S. Kriglstein, “Visualization-based analysis of gameplay data – A review of literature,” Entertainment Computing, vol. 4, no. 3, hal. 143–155, Agustus 2013.