[[1]](#footnote-1)

Procedural Map Generation untuk game “Splatted”

Lukky Hariyanto, *Departemen Informatika Institut Sains dan Teknologi Terpadu Surabaya*, Hendrawan Armanto, S.Kom., M.Kom., *Departemen Informatika Institut Sains dan Teknologi Terpadu Surabaya*,

*Abstrak*—Game merupakan aspek hiburan yang sekarang ini merajalela di seluruh dunia, dimana video game telah menjadi salah satu industri hiburan terbesar melebihi buku, film dan musik. Tetapi itu disebabkan oleh jumlah talenta yang diperlukan untuk membuat sebuah game, diisi oleh beberapa desainer / direktur yang memimpin pekerjaan yang diperlukan seperti direktur seni, direktur lagu, direktur efek suara, deasainer mekanik game dan masih banyak lagi, dengan salah satu aspek yang tidak kalah penting tapi tidak terlalu terkenal adalah desain level. Dalam tugas akhir ini, akan dibuat sebuah game yang mencoba memakai Level Generation. Game ini adalah Splatted, sebuah game 5v5 dimana setiap orang akan bertarung dengan mengambil salju di tanah, membuat nya menjadi sebuah bola salju, lalu melempar bola ke tim musuh sekaligus menangkap atau menghindari bola – bola yang datang ke tim nya. Di dalam game ini juga terdapat beberapa bola spesial yang bisa membuat permainan lebih menarik dengan menambah jumlah bola yang bisa dilempar oleh setiap orang. Lalu untuk level generation yang dipakai akan menggunakan algoritma genetik, sebuah algoritma yang didasarkan pada teori evolusi Darwin, dimana hewan – hewan yang memiliki fitur tertentu akan memiliki keunggulan dalam bertahan hidup dan beranak. Menggunakan itu akan dibuat 2 jenis pembuatan level, yaitu Tile Generation yang membuat level berdasarkan isi dari setial tile di level atau Template Generation dimana setiap tile 5x5 akan direpresentasikan menggunakan sebuah template yang telah dibuat menggunakan tangan. Dengan ini akan diberi 3 opsi ukuran level kepada player, yaitu kecil (20 x 15 tile), sedang (30 x 20) tile dan besar (40 x 25 tile).Dari game yang telah dibuat ini disimpulkan bila Procedural Map Generation bisa dipakai di kasus – kasus tertentu dalam sebuah game dan bekerja dengan baik, tapi salah satu saran yang penting untuk diingat setiap pemakai dari Procedural Map Generation adalah untuk memasukkan unsur – unsur buatan tangan kedalam level – level yang dibuat karena bila tidak maka level akan bersifat membosankan, dan dikarenakan memasukkan unsur – unsur manusiawi akan membuat level jauh lebih unik dan menarik untuk dimainkan.

*Kata Kunci*—Algoritma Genetik, Game, Procedural Map Generation.

# Pendahuluan

D

esain level telah menjadi salah satu komponen utama dari membuat sebuah game, dimana meskipun komponen ini tidak terlalu tampak ketimbang peran – peran lain seperti direktur seni atau desain musuh, desain level merupakan sebuah peran yang tidak kalah pentingnya dalam membuat game. Tetapi seperti peran – peran lain, mendesain level memakan waktu dan tenaga yang signifikan. Sebagai contoh, game Live Service seperti Valorant, Apex Legends, Fortnite, Overwatch 2 dan banyak lagi memerlukan desainer map untuk selalu bekerja membuat map baru agar para pemain tidak bosan bermain di map yang sama terus. Alternatif lain dari membuat level secara manual adalah untuk membiarkan game tersebut membuat level sendiri dengan proses yang dinamakan Procedural Map Generation[1].

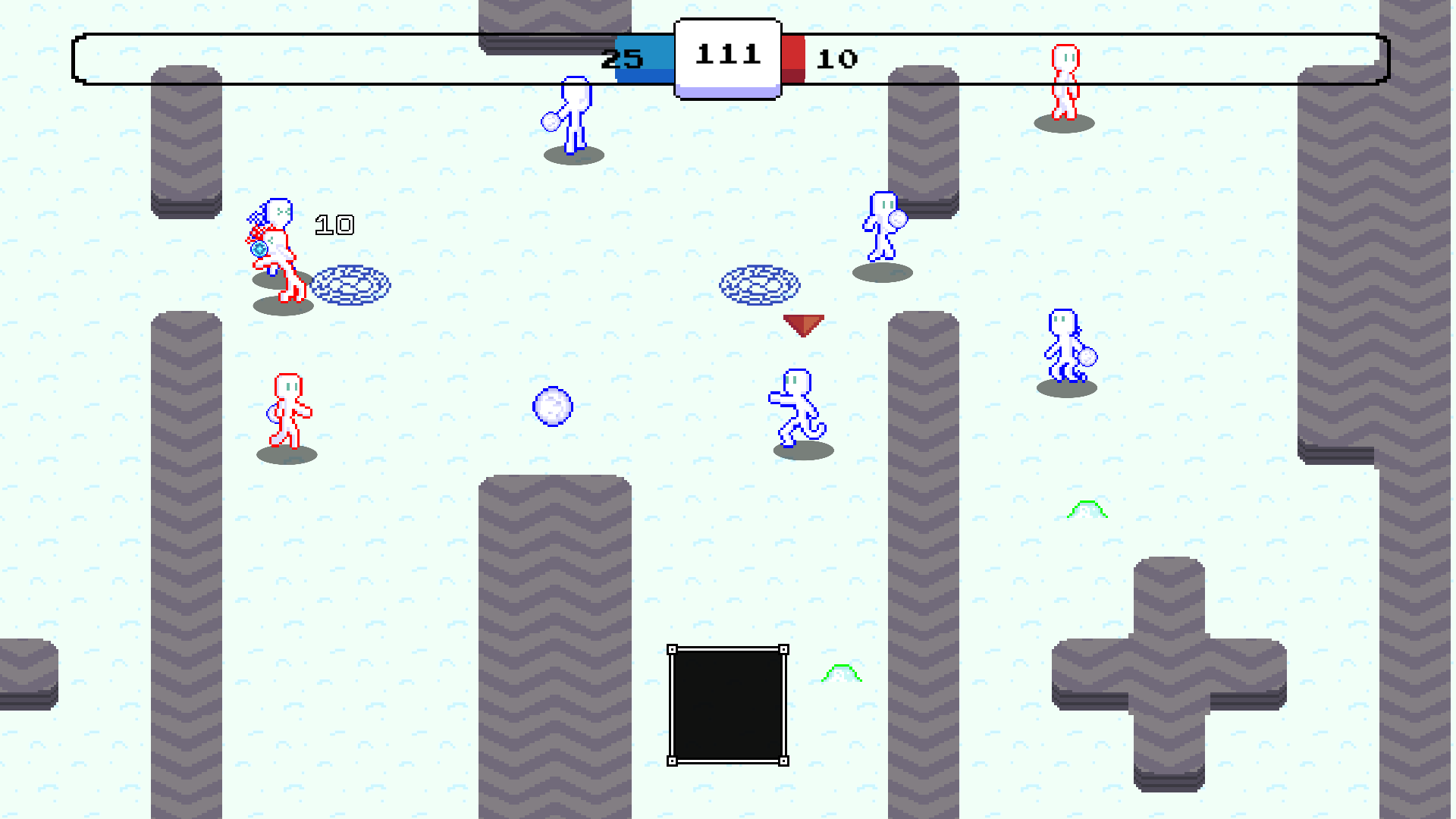
Procedural Map Generation adalah sebuah proses dimana sebuah game menggunakan sebuah algoritma untuk membuat level. Contoh populer dari game – game yang menggunakan Procedural Map Generation adalah Minecraft, No Man’s Sky, Terraria dan Deep Rock Galactic. Seluruh game tersebut membuat level mereka sendiri dengan algoritma seperti Perlin Noise untuk Minecraft atau Fractal Terrain Generation untuk Terraria. Untuk tugas akhir ini, kita akan mencoba menggunakan algoritma genetik[2] untuk membuat level dalam Game Splatted.

Algoritma genetik adalah sebuah algoritma yang didasarkan pada teori evolusi Darwin, dimana spesies yang berevolusi untuk beradaptasi mengikuti lingkungannya akan mendapat keuntungan dalam bertahan hidup dan melahirkan anak – anak. Dengan teori tersebut, algoritma genetik adalah sebuah algoritma dimana ada banyak solusi terhadap masalah optimisasi yang dihadapi, dimana solusi – solusi terbaik akan membuat solusi – solusi baru didasarkan dari solusi – solusi terbaik sebelumnya. Dan proses ini akan diulang terus hingga solusi tidak bisa berkembang lagi atau algoritma telah berjalan untuk waktu yang cukup lama.

Dengan itu, tujuan dari Tugas Akhir ini adalah untuk mencoba membuat sebuah game dengan Procedural Map Generation yang menggunakan algoritma genetik, untuk melihat apabila algoritma genetik dapat digunakan untuk membuat level – level game dan melihat apabila algoritma genetik merupakan sebuah alternatif yang sama bagusnya ketimbang algoritma – algoritma yang lain.

# Game “Splatted”

Sebelum memasuki algoritma genetik, perlu diketahui apa itu Splatted dan peran algoritma genetik dalam membuat game ini. Splatted adalah sebuah game 5 vs 5 dimana 2 kelompok bermain perang salju untuk mendapat poin tertinggi dari kedua kelompok tersebut. Player bisa melempar bola ke tim lawan untuk mendapatkan skor, dan player juga bisa menangkap atau menghindari bola yang dilempar oleh tim lawan untuk mencegah tim lawan untuk mendapat skor. Dalam game ini juga terdapat bola spesial yang bila diambil memiliki perilaku khusus dibandingkan bola biasa.



Gambar. 1. Cuplikan layar game Splatted

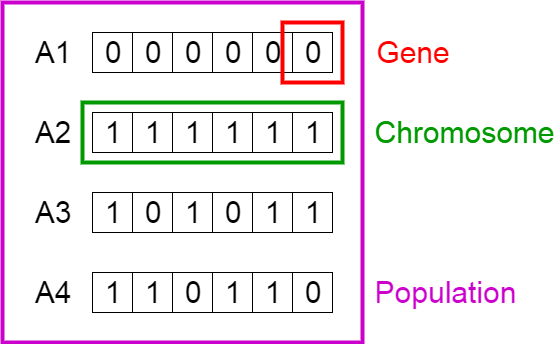
Peran algoritma genetik dalam membuat game ini adalah untuk membentuk level – level sesuai dengan ukuran dari level yang diminta oleh player. Terdapat 4 komponen yang bisa mengisi sebuah tile di dalam level Splatted, yaitu tile kosong, batu, bola spesial dan player. Dengan 4 komponen tersebut, dibuat 2 jenis Map Generation, yaitu Tile Generation dimana setiap tile memiliki isi tersendiri dan Template Generation dimana setiap 5X5 tile diisi menggunakan template yang telah dibuat oleh tangan.

# Algoritma Genetik

Dengan game telah dijelaskan, sekarang memasuki algoritma genetik, dimana terdapat beberapa tahap yang perlu dilakukan untuk menemukan solusi terbaik untuk optimisasi, dalam kasus ini, isi dari level yang akan dibuat oleh game Splatted. Tetapi sebelum memasuki tahap – tahap dari membuat algoritma genetik perlu diketahui beberapa terminologi dalam algoritma genetik:

## Terminologi Algoritma Genetik

Dalam algoritma genetik, seluruh konsep – konsep yang diterapkan mengambil inspirasi dari kromosom DNA kita, dimana sebuah representasi dari sebuah solusi dinamakan sebuah kromosom, lalu di dalam kromosom tersebut terdapat variabel – variabel yang membentuk solusi dinamakan sebuah Gene. Dan terakhir, sebuah kelompok yang berisi banyak kromosom dinamakan sebuah populasi dimana sebuah populasi ke – n juga bisa dipanggil generasi n.



Gambar. 2. Terminologi yang digunakan pada algoritma genetik

## Menentukan representasi dari solusi

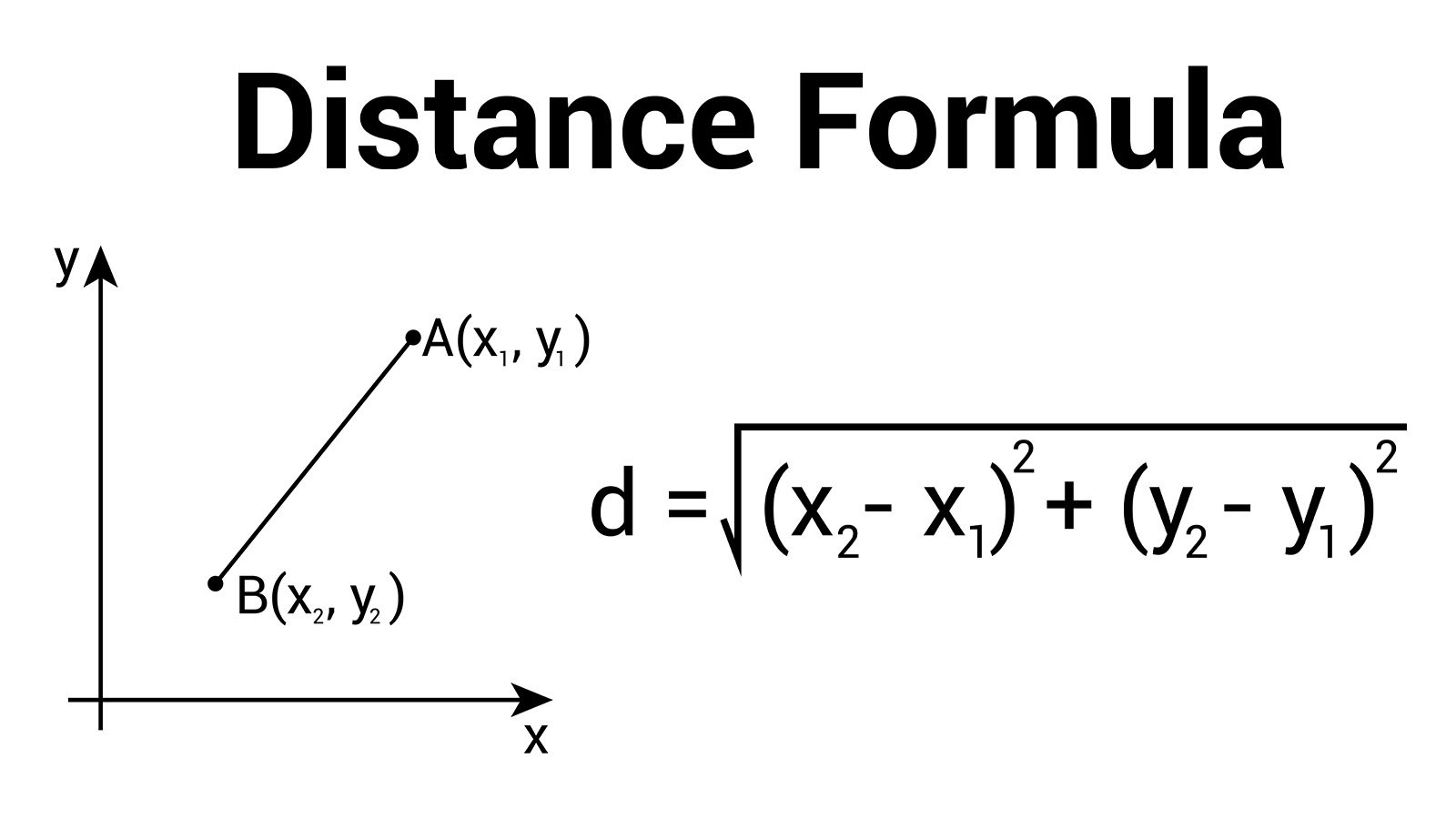
Tahap paling pertama dalam melakukan algoritma Genetik adalah untuk menentukan representasi dari solusi yang akan kita pakai untuk permasalah kita. Solusi dari masalah akan diubah bentuknya menjadi sebuah tipe data agar bisa dibaca oleh algoritma kita. Salah satu contoh permasalahan yang sederhana adalah mencari garis terpanjang yang bisa dibuat dalam sebuah persegi. Dalam kasus ini diperlukan 2 titik, dengan masing – masing titik memerlukan nilai x dan y. Sehingga dikarenakan ada 2 titik, solusi dari permasalahan ini akan direpresentasikan dengan sebuah kromosom berisi 4 gene, 2 gene untuk x dan y titik pertama dan 2 lagi untuk x dan y titik kedua.

## Pembuatan Populasi Pertama

Untuk memulai proses algoritma genetik, kita memerlukan populasi pertama untuk berperan sebagai kakek – kakek buyut dari seluruh kromosom yang akan mendatang. Di tahap ini, algoritma genetik akan membuat sebuah populasi berisi kromosom – kromosom dengan jumlah yang telah ditentukan, masing – masing dengan Gene yang benar – benar acak di dalam batasan yang telah ditetapkan. Setiap Gene akan memiliki nilai yang berbeda diantara satu sama lain dengan tujuan untuk memperbesar variasi dari anak – anak kromosom yang bisa lahir pada generasi selanjutnya.

## Perhitungan fitness

Dengan populasi pertama dibuat, tahap selanjutnya dalam proses algoritma genetik adalah untuk menghitung Fitness dari seluruh kromosom yang tersedia. Fitness adalah sebuah nilai yang menandakan apabila sebuah kromosom memiliki solusi yang bagus dengan permasalahan yang diberikan, dimana semakin tinggi fitness yang didapat menandakan semakin bagus kualitas kromosom dalam menangani masalah yang diberikan. Setiap kromosom akan dihitung fitnessnya menggunakan rumus – rumus yang ditetapkan sesuai kebutuhan yang diperlukan, untuk contohnya kita gunakan kembali contoh garis terpanjang dalam sebuah persegi diatas. Seperti yang kita tahu garis terpanjang dalam sebuah persegi adalah garis diagonal dimana titik – titik diletakkan di ujung persegi yang berlawanan. Tetapi algoritma tidak mengetahui ini dan inilah peran dari sebuah fitness.

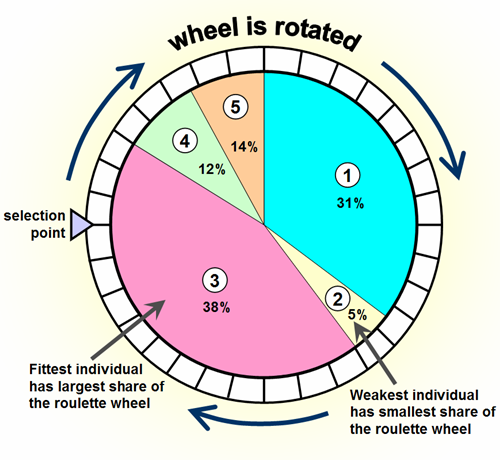


Gambar. 3. Rumus jarak antara 2 titik

Di gambar 3 terdapat rumus yang dapat digunakan untuk mencari jarak antara 2 titik. Variabel d diatas dapat digunakan sebagai nilai fitness kita, dimana kita mencari panjang garis terpanjang dalam sebuah persegi, dan bila garis semakin panjang, maka fitness akan semakin membesar, menandakan bahwa solusi ini adalah solusi yang diinginkan

## Seleksi & Crossover

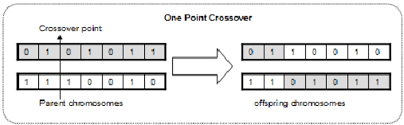
Setelah selesai menghitung nilai Fitness dari seluruh populasi, tahap selanjutnya adalah untuk melakukan tahap seleksi, dimana dalam tahap ini digunakan algoritma yang menentukan kromosom – kromosom yang akan dipilih untuk tahap Crossover berdasarkan fitness yang dimiliki kromosom – kromosom tersebut. Terdapat banyak algoritma yang bisa dipakai, seperti Tournament Selection, Ranking Selection, dan masih banyak lagi []–[], tetapi seluruh algoritma memiliki tujuan utama untuk memprioritaskan kromosom – kromosom dengan fitness tertinggi, dengan contoh yang paling populer adalah Roulette Wheel Selection.



Gambar. 4. Ilustrasi Roulette Wheel Selection

Dalam Roulette Wheel Selecrtion, setiap kromosom memiliki kesempatan untuk dipilih, tetapi semakin besar fitness sebuah kromosom dibandingkan dengan kromosom – kromosom yang lain, maka semakin besar bagian yang diperoleh dalam Roulette Wheel, sehingga semakin besar juga kesempatan Roulette Wheel memilih kromosom dengan Fitness yang besar. Seperti di gambar 4 diatas, kromosom 3 yang merupakan kromosom dengan Fitness tertinggi memiliki kesempatan 38% untuk dipilih, diikuti dengan kromosom 3, tetapi kromosom 2,4 dan 5 memiliki kesempatan yang kecil dikarenakan Fitness mereka yang kecil, membuat mereka kesulitan dipilih.

Menggunakan metode seleksi yang dipilih, diambil 2 kromosom yang akan menjadi orang tua dari sepasang kromosom baru. Tahap selanjutnya setelah memilih 2 kromosom menggunakan seleksi adalah tahap Crossover, dimana Gene – Gene di dalam kedua orang tua kromosom ditukar – menukar untuk membuat sepasang anak kromosom baru. Ini berfungsi untuk membuka posibilitas untuk mencari solusi atau kromosom yang lebih bagus daripada seluruh kromosom yang ada di generasi sekaran ini. Sama seperti Seleksi, terdapat banyak metode untuk melakukan Crossover, seperti N-Point, Cycle Crossover, ada juga Crossover yang menggunakan 3 parent ketimbang 2[]-[], tetapi untuk Crossover yang sederhana, One Point Crossover dapat melakukan tugas ini dengan baik.



Gambar. 5. Contoh One Point Crossover

Dalam One Point Crossover, dipilih sebuah posisi diantara 2

## Mutation

Tahap opsional yang sangat membantu setelah membuat anak kromosom dari tahap Crossover diatas adalah untuk melakukan Mutation untuk menambah variasi dari setiap kromosom yang dilahirkan. Mutation atau mutasi dalam algoritma genetik adalah sebuah tahap dimana Gene – Gene dalam sebuah kromosom dapat diubah – ubah, entah isinya yang diubah atau urutan Gene-nya Algoritma mutasi yang digunakan oleh Splatted adalah Partial Shuffle Mutation, dimana dipilih 2 titik diantara kedua Gene dalam kromosom kita, lalu seluruh Gene yang terdapat diantara kedua kromosom tersebut akan diacak posisinya.

## Elitism

Setelah dibentuk generasi terbaru, maka terdapat pilihan untuk melakukan Elitism, dimana Elitism mengambil beberapa dari kromosom yang memiliki fitness tertinggi dari generasi sebelumnya, dan meletakkan mereka ke dalam generasi terbaru, dimana Elitism dapat berfungsi sebagai jaring keamanan yang menjamin solusi terbaik tidak akan hilang dan bisa dipakai bila algoritma tidak menemukan solusi yang baik.

## Kondisi berhenti

Dan terakhir yang perlu ditentukan adalah kondisi berhenti dari algoritma genetik yang dibuat, atau algoritma akan berjalan terus untuk mencari jawaban terbaik tanpa selesai. Terdapat beberapa solusi untuk kasus ini, seperti berhenti setelah n generasi dimana fitness terbaik dari tiap generasi tidak berubah banyak, atau mungkin algoritma akan berhenti setelah n generasi sehingga waktu berjalan algoritma akan kurang lebih sama. Untuk Splatted digunakan opsi pertama, dikaranakan opsi tersebut dapat menghasilkan fitness terbaik yang bisa ditemukan.

# Implementasi Algoritma Genetik untuk Splatted

## Representasi

Dalam game Splatted, dikarenakan terdapat 2 jenis generasi level, maka terdapat 2 representasi yang berbeda untuk algoritma genetik kita, dimana representasi Tile Generation direpresentasikan oleh sebuah kromosom 1 dimensi sepanjang luas level dengan isi 0 hingga 3 yang merepresentasikan isi tile, sementara Template Generation direpresentasikan menggunakan sebuah array yang akan diisi dengan id – id dari template yang digunakan.

## Fitness – Fitness

dalam kasus Splatted, setiap kromosom memiliki beberapa fitness yang berbeda, salah satunya adalah Rock Amount Fitness, dimana sebuah level akan dihitung jumlah batu yang ada, dimana bila batu lebih banyak atau lebih sedikit dari batas yang ditentukan, maka fitness akan berkurang, dimana semakin jauh jumlah batu dari ekspektasi, maka semakin kecil fitness yang diberikan. Proses ini akan dilakukan untuk setiap kromosom yang tersedia untuk tahap selanjutnya, yaitu :

## Metode Seleksi

seperti seleksi yang dipakai oleh Splatted, yaitu Roulette Wheel Selection, dimana setiap kromosom bisa dipilih sejelek apapun kromosom tersebut, tetapi semakin tinggi fitness sebuah kromosom maka semakin tinggi kemungkinan kromosom tersebut dipilih.

## Metode Crossover

Untuk Splatted, algoritma Crossover yang dipakai adalah Uniform Crossover, dimana dimulai dari Gene pertama hingga terakhir, setiap Gene memiliki kesempatan 50% untuk ditukar, membuat Uniform Crossover sebagai Crossover yang bergantung pada keberuntungan. Setelah membuat 2 anak baru tersebut, kedua tahap ini dilakukan berkali – kali hingga terbentuk populasi baru dari anak – anak kromosom yang dibuat.

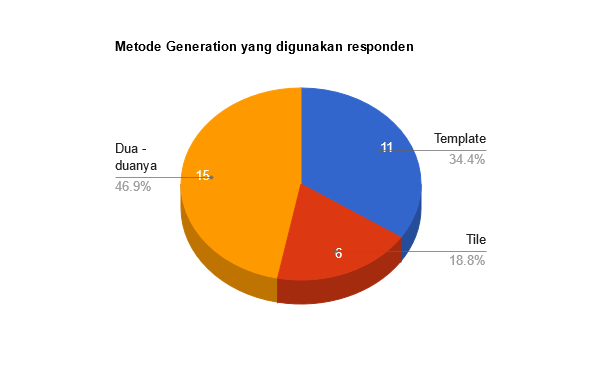
## Metode Mutation

## Kondisi Berhenti

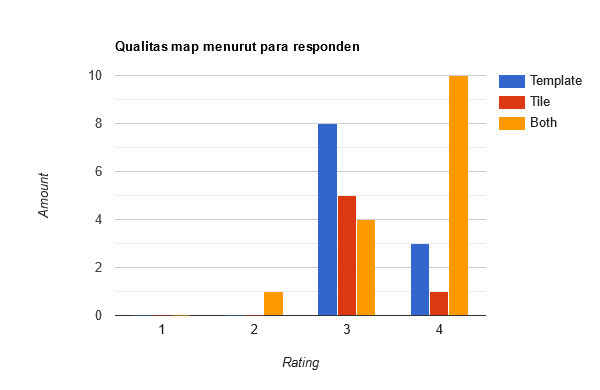
# Uji Coba

Untuk melihat kualitas dari level – level yang telah dibuat oleh game “Splatted”, akan diadakan 2 metode uji coba. Yang pertama adalah untuk membuat kuisioner yang berisi pendapat – pendapat dari para responden yang memainkan game ini, lalu kedua adalah untuk mengambil 2 level dari masing – masing metode Generation, dan mencoba untuk menganalisa tiap level yang ada tersebut.

## Tanggapan kuisioner



Gambar. 3. Metode Generation yang dipakai oleh para responden



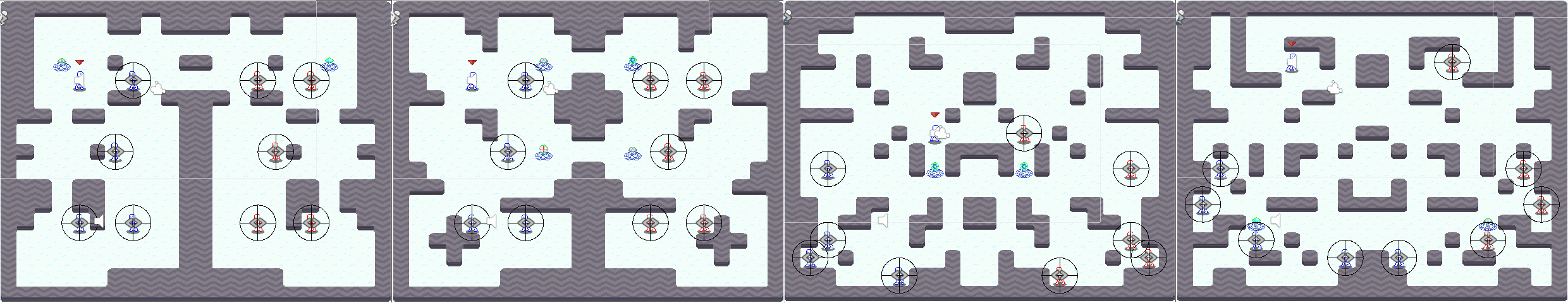
Gambar. 4. Skor yang diberi Responden

Dua gambar diatas merupakan responden – responden yang telah memainkan game Splatted dan memberikan pendapat mereka mengenai kualitas dari level – level yang telah dibuat. Terdapat 3 kelompok responden yang direpresentasikan menggunakan warna, yaitu responden yang mencoba Template Generation direpresentasikan warna biru, responden yang mencoba Tile Generation menggunakan warna merah dan terakhir adalah responden yang mencoba kedua – dua Generation ditampilkan dengan warna kuning.

Total responden saat menulis tugas akhir / karya ilmiah ini adalah 32 responden, dengan 11 mencoba template generation, 8 mencoba tile generation dan 15 sisanya mencoba kedua – dua generation berdasarkan gambar 3. Lalu dari seluruh responden tersebut, nilai nilai yang diberi dapat dilihar di Gambar 4 dimana rata rata nilai berada diantara 3 dan 4, dengan 1 responden tidak puas dengan level yang dibuat dan memberi nilai 2.

Dari gambar 4 diatas, diambil konklusi apabila menurut para responden, level – level yang dibuat oleh algoritma genetik bersifat memuaskan secara keseluruhan meskipun mungkin masih ada beberapa aspek yang dapat dikembangkan dari generasi level game “Splatted’.

## Analisa Level



Gambar. 5. Level - level yang dibuat oleh algoritma genetik

Di gambar 5 diatas terdapat 4 level berukuran kecil (15 X 20 Tile) yang dibuat oleh algoritma genetik, dimana terdapat 2 level di kiri yang menggunakan Template Generation dan 2 lagi di kanan menggunakan Tile Generation. Dilihat dari sini, keempat level memiliki beberapa tempat terbuka yang bisa digunakan para player untuk bermain, dan juga tidak ada tempat yang tidak bisa diakses berkat fitness yang dibuat. Lalu selanjutnya, seluruh level memiliki bola spesial tersebar dalam setiap level dengan jumlah yang tidak terlalu banyak. Untuk konklusi dari pengamatan level yang dibuat, level – level bisa dilihat tidak menghalang player dalam bermain dan terlihat tidak jelek secara estetika.

# Kesimpulan

Algoritma Genetik dapat digunakan untuk membuat sebuah Procedural Map Generation yang bagus, mungkin setara atau lebih bagus dengan Procedural Map Generation lainnya, selama terdapat cukup banyak unsur manusia yang dapat membuat sebuah level tidak membosankan untuk dimainkan.

# Ucapan Terima Kasih

Terima kasih diucapkan kepada keluarga saya yang mendorong saya untuk berkembang terus, dosen – dosen iSTTS yang bersabar dalam mengajari saya, staff iSTTS yang selalu siap membantu, teman – teman seperjuangan saya yang juga membantu saya secara langsung maupun tidak langsung, dan terakhir pak Hendrawan karena telah membimbing saya dari awal hingga akhir pembuatan tugas akhir ini.

Daftar Pustaka

1. Barriga, Nicholas A. *Short Introduction to Procedural Content Generation Algorithms for Videogames.* Chile : University of Talca. 2018.
2. Pereira, Leonardo et. All. *A multi-population genetic algorithm for procedural generation of levels for platform games*. Sao Carles : Conference Paper. 2014.
3. Whitley, Darrell. *A Genetic Algorithm Tutorial*. Technical Report CS-93-103. 1993.

**Lukky Hariyanto** lahir di Surabaya, Jawa Timur pada tahun 2001. Dia memulai studi S1 di program studi Informatika STTS pada tahun 2019. Minatnya pembelajarannya adalah bidang game development dan Computer Science.

1. Lukky Hariyanto, Departemen Informatika, Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail: lukky.h20@mhs.istts.ac.id)

   Hendrawan Armanto, Departemen Informatika, Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail: hendrawan@stts.edu) [↑](#footnote-ref-1)