Procedural Map Generation for 'Splatted': Enhancing Player Experience through Genetic Algorithms and AI State Machines in a Snowball Throwing Game

Lukky Hariyanto1 dan Hendrawan Armanto1

1Departemen Informatika, Fakultas Sains dan Teknologi, Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Indonesia

**Corresponding author:** Lukky Hariyanto (e-mail: lukky.h20@mhs.istts.ac.id).

ABSTRACT The world of gaming, now a global entertainment phenomenon, has grown to become one of the largest entertainment industries, even surpassing books, films, and music. However, game development is a complex endeavor requiring diverse talents, including directors, project leaders, designers, and programmers. This study aims to alleviate this complexity by creating a game named "Splatted" with key features centered around procedural map generation using genetic algorithms and artificial intelligence employing state machines. The genetic algorithm used is inspired by Darwin's theory of evolution, where each entity possesses unique traits, and superior ones can thrive and pass on their characteristics to their descendants. The research explores two map generation methods: tile-based and template-based, each with its own advantages and drawbacks. The results of the experiments in this study suggest that the genetic algorithm performs effectively and leads to a relatively high level of player satisfaction.

KEYWORDS Game, Genetic Algorithm, Map, Procedural Map Generation.

ABSTRAK Game merupakan aspek hiburan yang sekarang ini merajalela di seluruh dunia dan telah menjadi salah satu industri hiburan terbesar melebihi buku, film, bahkan musik. Tetapi dunia game bukanlah dunia yang mudah, sejumlah talenta diperlukan dalam pembuatannya mulai dari direktur, kepala proyek, desainer, hingga programmer. Untuk membantu beban pekerjaan tersebut maka penelitian ini akan membuat sebuah game bernama Splatted dengan fitur utama adalah procedural map generation berbasiskan algoritma genetik dan kecerdasan buatan berbasiskan state machine. Algoritma genetik itu sendiri adalah sebuah algoritma yang didasarkan pada teori evolusi Darwin, dimana setiap makhluk hidup memiliki fitur sendiri-sendiri dimana makhluk hidup yang unggul dapat bertahan hidup dan meneruskan fiturnya ke keturunannya. Dalam penelitian ini terdapat dua jenis map generation yaitu tile based generation dan template based generation dimana masing-masing jenis memiliki kelebihan dan kelemahannya sendiri-sendiri. Berdasarkan hasil ujicoba yang telah dilakukan pada penelitian ini dapat kami simpulkan bahwa algoritma genetik bekerja dengan baik dan memperoleh tingkat kepuasan pemain cukup tinggi.

KATA KUNCI Algoritma Genetik, Game, Map, Procedural Map Generation.

1. PENDAHULUAN

Desain map merupakan salah satu tahap utama dari membuat sebuah game, dimana meski tahap ini tidak terlalu tampak dibandingkan dengan tahap lainnya seperti desain UI atau desain musuh, desain map merupakan sebuah peran yang tidak kalah pentingnya dalam membuat sebuah game. Tetapi seperti tahap lainnya, mendesain map 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 map secara manual adalah untuk membuat map menggunakan algoritma dengan proses yang dinamakan Procedural Map Generation[1]-[3].

Procedural Map Generation adalah proses dimana sebuah game menggunakan algoritma untuk membuat tempat bermain para player dinamakan sebuah map. 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 map mereka sendiri dengan algoritma seperti Perlin Noise[4] untuk Minecraft atau Fractal Terrain Generation[5] untuk Terraria. Untuk tugas akhir ini, akan digunakan algoritma genetik[6]-[9] untuk membuat map dalam Game Splatted.

Algoritma genetik adalah sebuah algoritma yang didasarkan pada teori evolusi Darwin, dimana dikatakan bila spesies yang berevolusi untuk beradaptasi mengikuti lingkungannya akan mendapat keuntungan dalam bertahan hidup dan melahirkan keturunan. Mengaplikasikan teori tersebut kedalam sebuah algoritma, algoritma genetik adalah sebuah algoritma dimana dalam setiap iterasi akan dibuat solusi – solusi baru berdasarkan 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.

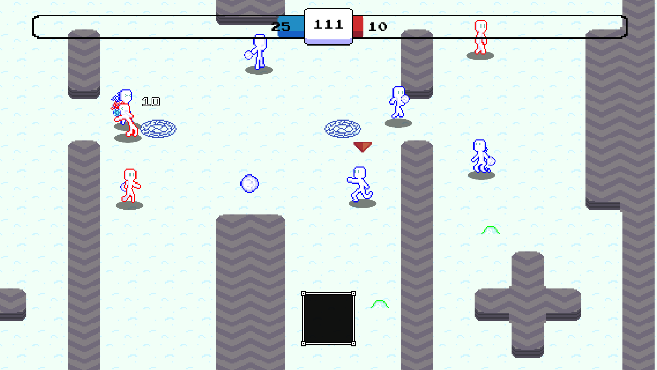
Dengan itu, tujuan dari Tugas Akhir ini adalah untuk membuat sebuah game dengan Procedural Map Generation yang menggunakan algoritma genetik, untuk melihat apabila algoritma genetik dapat digunakan untuk membuat map untuk game dan melihat apabila algoritma genetik merupakan sebuah alternatif yang sama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

1. GAME “SPLATTED”

Sebelum memasuki algoritma genetik, perlu diketahui apa itu Splatted dan peran algoritma genetik dalam membuat game ini. Dalam bab ini akan dijelaskan cara kerja dari permainan Splatted.

1. SPLATTED

Splatted adalah sebuah game dimana 2 kelompok terdiri atas 5 orang bermain perang salju untuk mendapat poin tertinggi dan menjadi pemenang dari perang salju tersebut. Bila sebuah tim memenuhi skor permainan tim mereka atau memiliki skor lebih banyak pada saat waktu permainan habis, maka permainan akan berakhir dan tim tersebut akan dinyatakan sebagai pemenang dari game ini. Dalam game ini terdapat beberapa aksi, diantaranya adalah mengambil bola, dimana player dapat mengambil sebuah gumpalan salju di tanah dan menggunakan salju tersebut sebagai amunisi, melempar bola dengan tujuan utama mengenai anggota tim lawan dan mendapat skor, menangkap bola dimana player dapat menangkap bola yang dilempar ke arah player tersebut, dan terakhir adalah aksi khusus milik player dimana player bisa melakukan fakeout untuk menipu bot untuk menangkap bola lebih cepat.



1. Cuplikan layar game Splatted
2. BOLA SPESIAL

Dalam game ini terdapat bola spesial yang bisa diambil dari spawner–spawner khusus, dimana bila diambil memiliki perilaku khusus dibandingkan bola biasa yang bisa membuat game menjadi lebih menarik. Berikut adalah bola – bola beserta efek – efeknya :

TABEL I

Bola-Bola Spesial Dalam Splatted

|  |  |  |
| --- | --- | --- |
| **Nama** | **Tampilan** | **Kegunaan** |
| Ice Piercer |  | Bisa Menembus Player – player yang dilewati |
| Snow-A-Rang |  | Bola bisa kembali ke pelempar |
| Explod-o-Ball |  | Bola bisa meledak, mengenai seluruh player di area ledakan |
| Freezing Winter |  | Bola Bisa memperlambat player yang kena |
| Stone Auger |  | Bisa menembus player dan dinding lalu pecah menjadi 3 bola normal |

1. STATE MACHINE BOT

Untuk mengatur behavior atau perilaku dari bot, digunakan sebuah Finite State Machine, sebuah sistem yang terdiri atas beberapa keadaan atau State dalam sebuah sistem. Dalam Finite State Machine, setiap state akan memberikan perilaku yang berbeda untuk bila digunakan oleh bot sesuai dengan situasi yang terjadi entah di bot atau di lingkungan sekitar bot, dimana bila kondisi-kondisi tertentu diraih, maka bot bisa berubah state, memberi bot perilaku baru, memberi ilusi seakan-akan bot sadar akan sekitarnya dan bisa beradaptasi. Berikut adalah state-state yang dimiliki oleh bot :

1. RaNDOM WALKING

Bot akan berjalan menuju lokasi acak yang telah dipilih dengan tujuan mencari salju di tanah bila tidak memiliki sebuah bola atau mencari lawan bila sudah memiliki bola, bila dicapai lokasi yang dituju tanpa menemukan objek yang dicari, maka akan dipilih lagi sebuah lokasi baru hingga ditemukan objek yang dicari.

1. AMBIL BOLA

Bila dalam state Random Walking bot tidak memiliki bola di tangan dan melihat sebuah gumpalan salju atau Spawner bola spesial, maka bot akan memasuki state ini. Dalam state ini bot akan berjalan menuju salju yang telah dilihat untuk mengambil salju tersebut. Setelah mendapat bola yang dilihat atau bila bola tersebut diambil orang lain, maka State bot akan kembali ke Random Walking.

1. AIM & THROW

Bila dalam State Random Walking bot memiliki bola di tangan dan melihat seorang anggota tim lawan, maka bot akan memasuki state Aim & Throw. Dalam state ini bot akan berhenti bergerak untuk membidik pergerakan lawan, dan setelah beberapa detik bot akan melempar bola ke arah prediksi yang ditentukan. Setelah berhasil melempar bola, maka State bot akan kembali ke Random Walking untuk mencari bola salju baru.

1. FOLLOW TARGET

Bila target yang dibidik dalam State Aim & Throw kabur dari pandangan bot, maka bot memasuki State Follow Target, dimana bot akan mencoba mengejar lawan hingga bot bisa melihat lawan tersebut lagi. Bila bot terlalu lama mengejar target, maka bot bisa capek dan menyerah dalam mengejar target dan kembali ke state Random Walking, atau bila berhasil mengejar target bot akan kembali memasuki state Aim & Throw.

1. CATCH BALL

State Catch Ball yang bisa diraih dari semua state dilakukan bila bot melihat sebuah bola terbang ke pandangannya dan bila bot tahu bola tersebut akan kena bot, maka bot akan mencoba untuk menangkap bola yang dilihat.

1. PERAN ALGORITMA GENETIK DALAM GAME INI

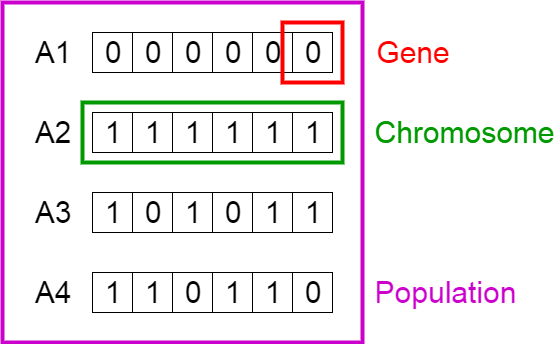
Peran algoritma genetik dalam membuat game ini adalah untuk membentuk map – map sesuai dengan ukuran dari map yang diminta oleh player. Terdapat 4 komponen yang bisa mengisi sebuah tile di dalam sebuah map Splatted, yaitu tile kosong, batu, bola spesial dan player. Dengan 4 komponen tersebut, dibuat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

1. ALGORITMA GENETIK

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

1. 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 Chromosome, yang mengandung variabel – variabel yang membentuk Chromosome dinamakan sebuah Gene. Terakhir, sebuah kelompok terdiri atas banyak kromosom dinamakan sebuah populasi, dimana sebuah populasi ke–n dari algoritma genetik juga bisa dipanggil generasi n.



1. Terminologi yang digunakan pada algoritma genetik
2. MENENTUKAN REPRESENTASI SOLUSI

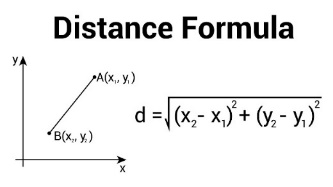
Tahap paling pertama dalam melakukan algoritma Genetik adalah untuk menentukan representasi dari solusi yang akan kita pakai untuk permasalah kita. Solusi dari masalah akan diubah bentuknya menjadi sebuah tipe data agar bisa dibaca oleh algoritma kita, cenderung dalam bentuk array 1 dimensi. Salah satu contoh permasalahan sederhana yang bisa dipakai 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 Chromosome berisi 4 Gene, 2 Gebe untuk nilai x dan y titik pertama dan 2 lagi untuk nilai x dan y titik kedua.

1. PEMBUATAN POPULASI PERTAMA

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

1. PERHITUNGAN FITNESS

Dengan populasi pertama dibuat, tahap selanjutnya dalam algoritma genetik adalah menghitung Fitness dari seluruh Chromosome yang tersedia. Fitness adalah sebuah nilai yang menandakan apabila sebuah kromosom memiliki solusi yang bagus dengan permasalahan yang diberikan, dimana semakin tinggi atau rendah fitness yang didapat (tinggi atau rendah dapat berbeda dalam tiap kasus) dapat menandakan kualitas sebuah kromosom. Setiap kromosom akan dihitung fitnessnya menggunakan rumus–rumus yang ditetapkan sesuai kebutuhan yang diperlukan, sebagai contoh kita gunakan kembali permasalahan garis terpanjang dalam sebuah persegi diatas. Seperti yang kita tahu garis terpanjang dalam sebuah persegi adalah garis diagonal diantara 2 titik berlawanan. Tetapi algoritma tidak mengetahui ini dan inilah peran dari sebuah fitness.

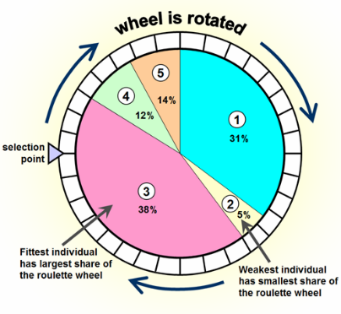


1. Rumus jarak antara 2 titik

Di gambar 3 terdapat rumus[10] yang dapat digunakan untuk mencari jarak antara 2 titik. Rumus ini dapat digunakan sebagai nilai fitness kita, karena kita mencari panjang garis terpanjang dalam sebuah persegi, dan bila garis semakin panjang, maka fitness akan semakin membesar, menandakan bahwa Chromosome yang dipilih adalah solusi yang diinginkan untuk permasalahan ini.

1. SELEKSI & CROSSOVER

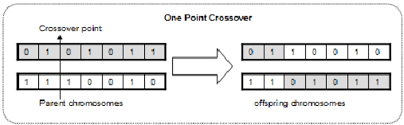
Setelah selesai menghitung nilai Fitness dari seluruh populasi, tahap selanjutnya adalah untuk melakukan tahap seleksi, dimana dalam tahap ini digunakan algoritma yang menentukan kromosom – kromosom yang akan dipilih untuk tahap Crossover berdasarkan fitness yang dimiliki kromosom – kromosom tersebut. Terdapat banyak algoritma yang bisa dipakai, seperti Tournament Selection[11], Ranking Selection, dan masih banyak lagi [12], tetapi seluruh algoritma memiliki tujuan utama untuk memprioritaskan kromosom – kromosom dengan fitness tertinggi, dengan contoh yang paling populer adalah Roulette Wheel Selection[13].



1. Ilustrasi Roulette Wheel Selection

Dalam Roulette Wheel Selection, setiap kromosom memiliki kesempatan untuk dipilih, tetapi semakin besar fitness sebuah kromosom, maka semakin besar bagian yang diperoleh dalam Roulette Wheel, alhasil semakin besar juga kesempatan Roulette Wheel memilih kromosom dengan Fitness yang besar tersebut. Seperti contoh gambar 4 diatas, Chromosome 3 yang merupakan kromosom dengan Fitness tertinggi memiliki kesempatan 38% untuk dipilih, diikuti dengan Chromosome 1 dengan kesempatan 31%, tetapi Chromosome 2,4 dan 5 memiliki kesempatan yang kecil dikarenakan Fitness mereka yang kecil, membuat mereka lebih tidak mungkin dipilih.

Setelah memilih dan menggunakan sebuah metode seleksi, diambil 2 Chromosome. Tahap selanjutnya setelah memilih 2 Chromosome tersebut adalah tahap Crossover [14] – [17] dimana Gene – Gene di dalam kedua orang tua kromosom ditukar untuk membuat sepasang anak kromosom baru. Ini berfungsi untuk membuka posibilitas mencari solusi atau kromosom yang lebih bagus dibandingkan dengan kromosom yang tersedia. Sama seperti Seleksi, terdapat banyak metode untuk melakukan Crossover, seperti N-Point, Cycle Crossover, ada juga Crossover yang menggunakan 3 parent, tetapi untuk Crossover yang sederhana, One Point Crossover dapat bekerja dengan baik.



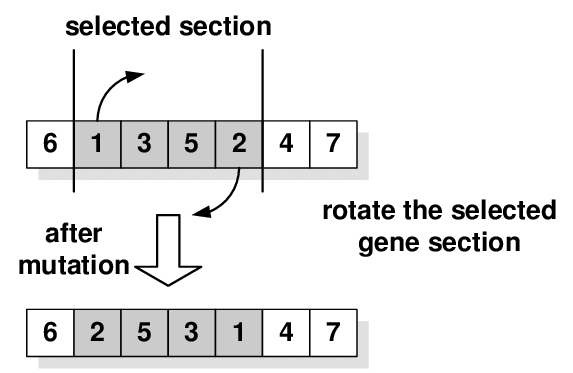
1. Contoh One Point Crossover

Dalam One Point Crossover, dipilih secara acak sebuah pembatas di antara 2 Gene. Tahap selanjutnya dari One Point Crossover adalah untuk menukar entah sisi kanan atau sisi kiri dari pembatas tersebut, menghasilkan 2 anak kromosom baru dengan isi Gene yang tercampur seperti pada gambar 5 diatas. Setelah tahap Crossover dilakukan, maka akan dilakukan lagi tahap Seleksi dan Crossover. Kedua tahap tersebut akan diulangi terus hingga jumlah anak Chromosome yang telah dibuat dapat memenuhi sebuah populasi.

1. MUTATION

Tahap opsional yang sangat membantu setelah membuat anak - anak kromosom dari tahap Crossover diatas adalah untuk melakukan Mutation dengan tujuan 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. Metode – metode Mutation tidak sedikit juga, terdapat banyak Mutation seperti

Polynomial Mutation[18], Gaussian Mutation, dan banyak lagi Mutation sejauh yang bisa dipikirkan [19]. Untuk Mutation yang sederhana, bisa dilihat Inversion Mutation :



1. Visualisasi Inversion Mutation

Dalam Inversion Mutation, diletakkan 2 pembatas di antara 2 Gene secara acak, lalu untuk seluruh Gene yang berada di antara 2 pembatas tersebut, balik urutan dari Gene yang terpengaruh, seperti Chromosome yang memiliki urutan Gene 1 3 5 2 dibalik menjadi 2 5 3 1 setelah Mutation dilaksanakan.

1. ELITISM

Setelah dibentuk generasi terbaru, maka terdapat pilihan untuk melakukan Elitism[20]-21, 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.

1. KONDISI BERHENTI

Dan terakhir yang perlu ditentukan untuk algoritma genetik adalah kondisi berhenti dari algoritma genetik yang dibuat, atau algoritma akan berjalan terus untuk mencari jawaban terbaik tanpa berhenti. 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, dimana kondisi berbeda dapat diuntungkan dalam kondisi – kondisi tertentu.

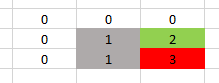
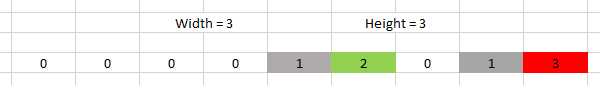
IMPLEMENTASI ALGORITMA KE SPLATTED

Dengan Metodologi Algoritma Genetik dijelaskan, sekarang akan dijelaskan Implementasi dari algoritma genetik untuk game Splatted ini, seperti metode representasi, Fitness – Fitness yang digunakan, dan metode – metode yang digunakan untuk operasi – operasi dalam algoritma genetik.

1. REPRESENTASI

Dalam game Splatted, dikarenakan terdapat 2 jenis generasi map, maka terdapat 2 representasi yang berbeda untuk algoritma genetik kita.

Pertama adalah Tile Generation. Map kita akan berbentuk sebuah array 2D, tetapi untuk algoritma genetik disarankan untuk memakai array 1 dimensi, sehingga untuk representasi Tile Generation, digunakan array 1 dimensi sepanjang luas dari map yang akan dibuat, dimana setiap Gene akan merepresentasikan sebuah tile yang ada di dalam map dengan isi Gene 0 hingga 3, merepresentasikan objek yang diletakkan di tile tersebut, dan setelah game mendapat array hasil algoritma genetik, maka array 1 dimensi tersebut dapat diubah menjadi array 2D, yang bisa diubah menjadi map yang siap dimainkan.

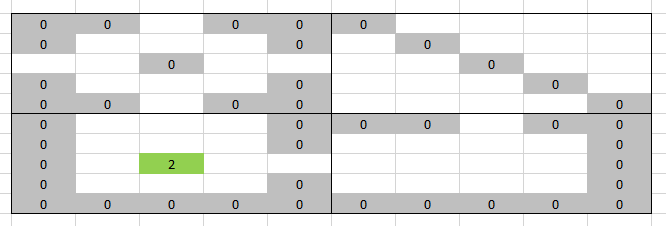
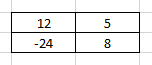
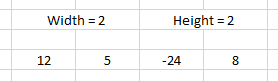


1. Contoh Representasi Tile Based menjadi sebuah map

Dengan contoh gambar 7 diatas, Representasi untuk Splatted ada di bagian atas gambar 7. Menggunakan variabel Width dan Height yang ditentukan oleh player, array 1 dimensi yang berisi angka angka berwarna (abu abu = batu, hijau = bola spesial dan 3 = player) diatas dapat dibentuk menjadi sebuah array 2D yang siap diolah menjadi map.

Lalu selanjutnya, mirip dengan Tile Generation diatas Template Generation direpresentasikan oleh sebuah kromosom 1 dimensi, tetapi tidak seperti Tile Generation Template Generation memiliki 2 perbedaan besar dalam representasi, yang pertama adalah setiap Gene di dalam Template Generation tidak diisi dengan angka 0 – 3 yang merepresentasikan obyek yang bisa diletakkan, tetapi -n hingga n dimana n adalah jumlah template yang siap dipakai. Setiap Gene akan diisi dengan id dari sebuah template, dimana bila id tersebut bernilai negatif maka template tersebut akan memiliki bola spesial di tengah template tersebut.

Perbedaan kedua yang berhubungan dengan perbedaan pertama diatas adalah panjang dari kromosom Template Generation tidak sepanjang luas dari map, tetapi luas dari map dibagi dengan 25. Ini dikarenakan setiap Template yang akan dipakai memiliki ukuran 5 x 5, membuat setiap Gene merepresentasikan sebuah tile berukuran 5 x 5 di dalam map. Untungnya map telah diatur sehingga pilihan – pilihan yang disediakan berkelipatan 5 untuk menghindari error – error yang mungkin terjadi.



1. Contoh Representasi Template Based menjadi sebuah map

Dalam gambar 8, terdapat sebuah array 1 dimensi yang berisi 4 id dari template yang disiapkan. Dilakukan perubahan menjadi 2D, lalu setelah itu template akan dibuka, memperlihatkan sebuah pola yang diibuat oleh tangan.Di bawah kiri ada bola spesial dikarenakan bila template bersifat negatif maka bila di map tersebut akan diletakkan bola spesial di tengah – tengah template, sebagai pengganti bola spesial yang tidak mumcul dari Template Management.

1. FITNESS - FITNESS

Dalam Splatted, terdapat 6 fitness yang akan bekerja sama untuk menentukan kualitas dari sebuah map. Seluruh fitness–fitness ini akan dijelaskan di bab selanjutnya, dimana kegunaan fitness beserta dengan rumus-rumus yang dipakai fitness tersebut akan dijelaskan.

1. METODE SELEKSI

Metode seleksi yang dipakai oleh Splatted adalah Roulette Wheel Selection seperti yang telah dijelaskan pada contoh diatas, dimana setiap kromosom bisa dipilih sejelek apapun Fitness dari kromosom tersebut, tetapi kromosom yang memiliki fitness lebih tinggi akan memiliki kemungkinan lebih besar untuk dipilih dibandingkan dengan kromosom yang memiliki fitness rendah.

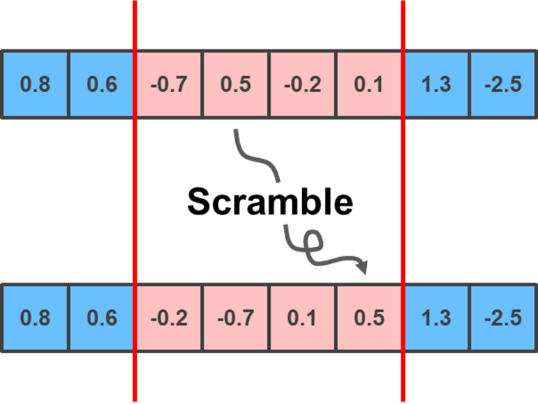
1. METODE CROSSOVER

Kebanyakan Untuk Splatted, algoritma Crossover yang dipakai adalah Uniform Crossover, dimana dimulai dari Gene pertama di paling kiri hingga Gene terakhir di paling kanan, setiap Gene memiliki kesempatan 50% untuk ditukar, membuat Uniform Crossover sebagai Crossover yang bisa mencampur aduk isi dari kromosom orang tua secara individual dimana kebanyakan Crossover lain memindahkan Gene secara kelompok, yang tidak cocok untuk game ini.



1. Visualisasi Uniform Crossover
2. METODE MUTATION

Algoritma mutasi yang digunakan oleh Splatted adalah Partial Shuffle Mutation atau juga bisa disebut Scramble Mutation, dimana diletakkan 2 pembatas di antara kedua Gene dalam kromosom kita, lalu seluruh Gene yang terdapat di antara kedua kromosom tersebut akan diacak posisinya. Mutation ini berguna untuk mengocok urutan dari bayak Gene secara sekaligus.



1. Visualisasi Scramble Mutation
2. KONDISI BERHENTI

Kondisi Berhenti dari Splatted adalah bila algoritma genetik tidak menemukan perkembangan yang signifikan setelah 100 generasi, maka algoritma akan dihentikan. Ini berguna agar algoritma berusaha mencari solusi terbaik yang bisa didapat, tetapi akan berhenti bila tidak membuahkan hasil untuk sementara. Ini bekerja di game Splatted dikarenakan kita perlu mencari map dengan kualitas bagus, tetapi tidak bisa mengorbankan terlalu banyak waktu dikarenakan ada player yang menunggu algoritma genetik tersebut.

*FITNESS – FITNESS SPLATTED*

Seperti yang telah ditetapkan di Bab IVB, fitness – fitness akan dipisah ke bab baru dikarenakan jumlah isinya yang banyak dibandingkan dengan aspek – aspek lain dari sebuah algoritma genetik. Berikut adalah fitness – fitness yang akan dijelaskan :

1. FITNESS JUMLAH BATU

Fitness Jumlah Batu digunakan untuk mengatur jumlah batu yang tersedia di dalam sebuah map agar map tidak koson atau memiliki terlalu banyak batu. Di sini terdapat beberapa variabel yang diperlukan oleh Fitness, yaitu MinR, yaitu jumlah minimal batu yang diperbolehkan di dalam map dan MaxR yang berisi jumlah maksimal batu yang diperbolehkan di dalam map.

(1)

Di rumus 1 diatas dicari m, sebuah nilai minus yang nanti akan digunakan untuk mengurangi nilai fitness yang berawal dari 1, dimana semakin jauh jumlah batu dari batas yang ditentukan maka semakin besar nilai minus. Sesuai dengan rumus 1, terdapat 3 posibilitas isi dari m. Pertama adalah batu minimal dikurang dengan batu di map, kedua adalah batu di map dikurangi batu maksimal bila batu di map melebihi batas batu maksimal yang ditentukan, dan terakhir 0 bila jumlah batu berada di antara kedua batas yang ditentukan.

(2)

Di rumus 2 diatas dicari nilai minus maksimal, yang nanti akan digunakan untuk menormalisasi nilai minus agar bila nilai minus digunakan untuk mengurangi Fitness, nilai Fitness tidak akan berada dibawah 0. Terdapat 2 posibilitas dari nilai minus maksimal, di antara luas map dikurangi jumlah batu maksimal atau jumlah batu minimal, entah yang mana yang lebih besar.

(3)

Tahap terakhir sesuai dengan rumus 3 adalah untuk mengurangi 1 dengan nilai minus yang dibagi nilai minus maksimal (untuk normalisasi). Dengan itu fitness sudah bisa didapatkan. Tetapi untuk membuat algoritma genetik lebih sensitif terhadap kesalahan maka nilai fitness akan dikuadratkan. Lalu W yang merupakan Weight akan dikalikan ke dalam nilai fitness agar bisa mengatur besar pengaruh fitness ini terhadap fitness lain. Rumus ini hanya digunakan pada Tile Generation dikarenakan bila ini diaplikasikan pada Template Generation maka akan sangat membatasi variasi dalam Template Generation.

1. FITNESS UKURAN KELOMPOK BATU

Selanjutnya adalah Fitness ukuran kelompok batu, dimana Fitness sebelumnya menghitung jumlah batu yang ada di dalam map, Fitness ini menghitung ukuran dari kelompok – kelompok batu yang ada di dalam map, sehingga batu – batu yang berkumpul bersama di dalam map tidak akan membentuk kelompok batu yang terlalu besar maupun kecil.

(4)

(5)

(6)

Fitness ini memiliki variabel – variabel yang sama dengan Fitness jumlah batu, seperti MaxR dan MinT, dan dikarenakan dengan konsepnya yang mirip dengan fitness jumlah batu, maka rumus yang dipakai untuk kedua fitness memiliki banyak kemiripan, kecuali pada rumus 6, dimana rumus 6 menghitung rata – rata dari seluruh fitness kelompok batu dibandingkan dengan rumus 3 yang hanya menghitung jumlah batu.

Rumus 4 dan 5 akan diulangi untuk setiap kelompok batu yang ditemui, dimana ditentukan nilai minus dan nilai minus maksimal dari kelompok batu tersebut. Setelah itu, dilakukan sebuah fungsi yang seperti rumus 3, lalu simpan nilai fitness itu untuk sementara. Setelah seluruh kelompok batu dihitung, maka dijalankan rumus 6, dimana seluruh fitness yang didapatkan dijumlah dan di rata – rata. Setelah selesai maka nilai Fitness dikuadratkan dan dikali W dan nilai Fitness siap dipakai. Fitness ini juga hanya dipakai pada Tile Generation karena Template Generation memiliki kelompok – kelompok yang cukup besar tapi hampir terjamin aman dan tidak akan mengganggu permainan

1. FITNESS LUAS AREA YANG BISA DILEWATI

Fitness ini meluhat seluruh tanah kosong yang ada di dalam map, lalu memeriksa berapa segmen – segmen di dalam map yang tidak bisa mengakses satu sama lain dan mengurangi Fitness sesuai dengan segmen – segman yang tidak bisa diakses. Fitness ini digunakan untuk mencegah dibuatnya area – area yang tidak bisa diakses oleh siapapun, dikarenakan bila dalam sebuah map hanya sedikit area yang bisa diakses oleh player, maka map akan sulit digunakan untuk bermain perang salju, dimana terdapat banyak bola yang dilempar untuk memenangkan permainan.

(7)

Di dalam Fitness ini hanya terdapat 1 rumus, yaitu area terbesar dibagi dengan area total sesuai dengan rumus 7. Game akan menyimpan ukuran – ukuran dari segmen – segmen dalam map, mengambil segmen terbesar dalam map dan membaginya dengan total ukuran dari semua segmen di dalam map. Dengan ini apabila sebuah map tidak memiliki sebuah segmen yang tidak dapat diakses, maka map tersebut akan mendapatkan nilai Fitness 1 dari Fitness luas area ini.

1. FITNESS BOLA SPESIAL YANG BISA DIRAIH

Fitness ini melihat setiap bola spesial yang ada di dalam map, dan mengecek bola – bola spesial tersebut terhadap player terdekat. Apabila player terdekat tersebut bisa mengakses bola spesial, maka Fitness akan bertambah. Tetapi bila Player tidak bisa mengakses bola spesial yang ada di dekatnya, maka Fitness ini akan berkurang. Fitness ini berguna untuk mencegah adanya bola spesial yang tidak bisa diraih oleh player, dimana bila player terdekat tidak bisa mengakses bola spesial, maka player lain akan dilewati untuk menghemat waktu dalam algoritma genetik.

(8)

Dalam fitness ini, diambil seluruh posisi dari bola spesial dan player yang ada di dalam map, lalu setiap bola spesial akan melihat player yang terdekat di antara seluruh player yang dicatat. Setelah menemukan player terdekat, dilakukan algoritma A\* dari player menuju bola. Bila A\* berhasil membuat sebuah jalan, maka pakses yang menandakan jumlah bola spesial yang bisa diakses akan bertambah sebanyak 1. Tahap ini akan diulangi terus hingga seluruh bola telah diperiksa, lalu pakses akan dibagi dengan ptotal yang merupakan jumlah bola spesial dalam sebuah map. Dengan itu nilai Fitness telah didapatkan.

1. FITNESS RASIO BOLA SPESIAL

Fitness berikutnya adalah Fitness Rasio Bola Spesial, dimana fitness ini digunakan untuk mengatur jumlah bola spesial yang ada di dalam map, sehingga bola spesial pasti hadir dalam sebuah map dan tidak memenuhi map.

(9)

(10)

(11)

Fitness ini bekerja dengan menghitung jumlah bola spesial yang tersedia, dan memasukkannya ke dalam rumus 9 hingga 11, yang sama seperti rumus 1 hingga 3 dikarenakan konsep nilai minusnya yang sama, dengan satu – satunya perbedaan adalah rumus ini menghitung jumlah bola spesial dibandingkan rumus 1 hingga 3 yang menghitung jumlah batu.

1. FITNESS VARIASI TEMPLATE

Terakhir adalah Fitness variasi template. Fitness ini melihat seluruh template yang ada di dalam map, dan mengurangi fitness apabila terdapat template sama yang muncul beberapa kali. Fitness ini berfungsi untuk menambah variasi dalam Template Generation agar tidak terjadi sebuah map yang mengandung 1 / 2 template yang sama diulang berkali – kali.

(12)

Untuk Fitness ini disediakan sebuah variabel x yang akan membantu untuk rumus 13 dibawah, dimana x adalah nilai negatif yang diberikan Fitness apabila ada sebuah template yang terlalu sering tampil dalam map, dimana bila frekuensi tampil melebihi toleransi (Tl), maka akan diberi fitness nilai frekuensi – toleransi yang akan bernilai negatif.

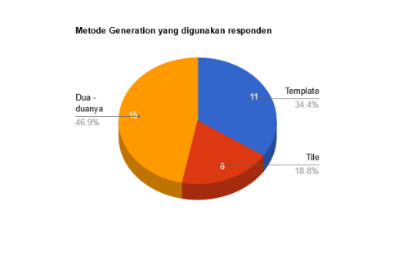
(13)

Setelah diperiksa seluruh template yang ada di dalam map, maka lakukan rata – rata dari seluruh fitness yang disimpan lalu dirata – rata dengan jumlah template yang unik di dalam map seperti rumus 6. Bila hasil dari rata – rata tersebut diatas 0 maka kuadratkan lalu kali W. Tetapi bila hasil rata – rata dibawah 0, maka nilai Fitness yang diberikan adalah 0, dikarenakan bila Fitness diberi nilai negatif bisa menyebabkan error yang tidak diinginkan.

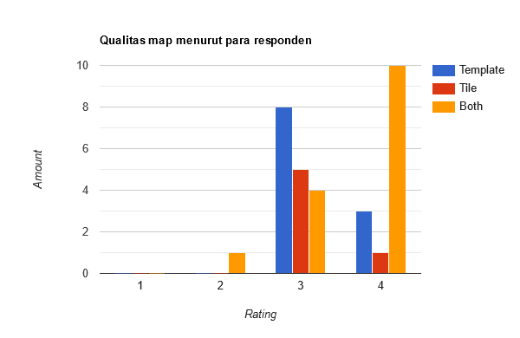
UJI COBA

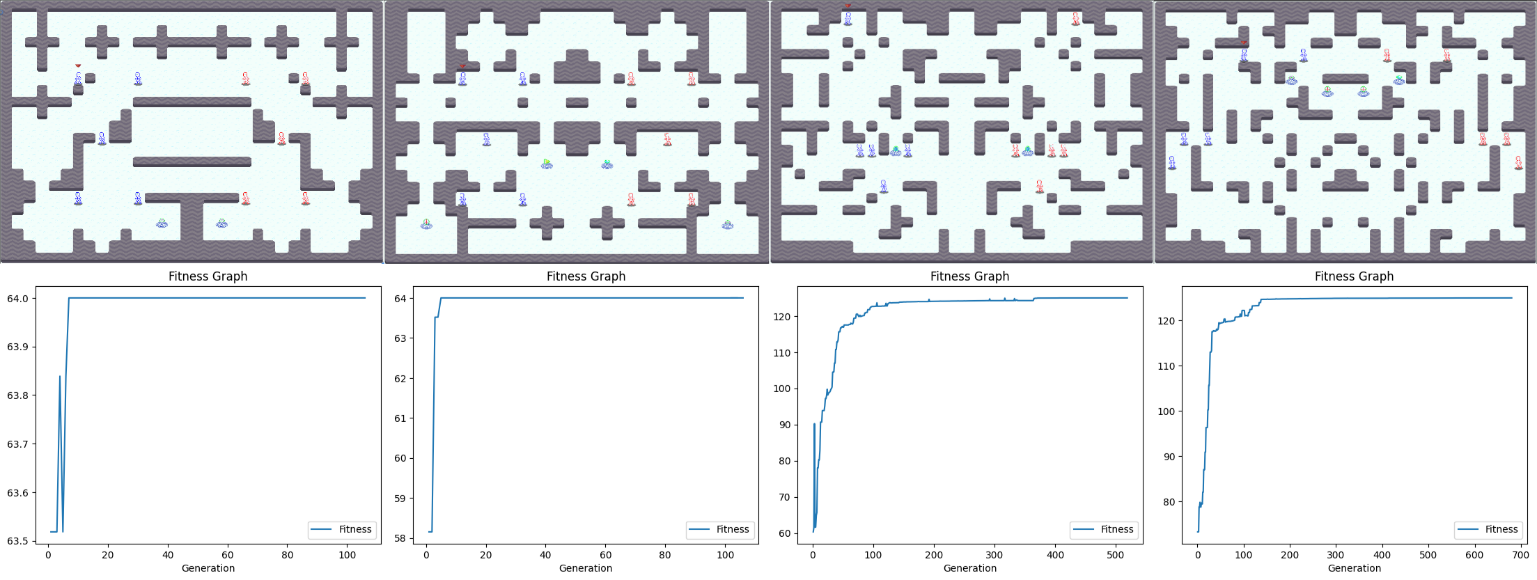
Untuk melihat kualitas dari map – map yang telah dibuat oleh game “Splatted”, akan diadakan 2 metode uji coba. Yang pertama adalah untuk membuat kuesioner yang berisi pendapat – pendapat dari para responden yang memainkan game ini, lalu kedua adalah untuk mengambil 2 map dari masing – masing metode Generation,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

1. TANGGAPAN KUESIONER



1. Metode Generation yang dipakai Responden



1. Skor yang diberi Responden
2.  Skor yang diberi Responden

Dua gambar 11 dan 12 diatas merupakan responden – responden mahasiswa iSTTS yang telah memainkan game Splatted dan memberikan pendapat mereka mengenai kualitas dari map – map 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 11. Lalu dari seluruh responden tersebut, nilai nilai yang diberi dapat dilihat di Gambar 12 dimana rata rata nilai berada di antara 3 dan 4, dengan 1 responden tidak puas dengan map yang dibuat dan memberi nilai 2.

Dari gambar 12 diatas, diambil konklusi apabila menurut para responden, map – map yang dibuat oleh algoritma genetik bersifat memuaskan secara keseluruhan meskipun mungkin masih ada beberapa aspek yang dapat dikembangkan dari generasi map game “Splatted’.

1. ANALISA MAP

Di gambar 13 diatas terdapat 4 map berukuran menengah (20 X 30 Tile) yang dibuat oleh algoritma genetik, dimana terdapat 2 map di kiri yang menggunakan Template Generation dan 2 lagi di kanan menggunakan Tile Generation, bersama dengan grafik fitness dari masing – masing map. Dilihat dari sini, keempat map 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 map memiliki bola spesial tersebar dalam setiap map dengan jumlah yang tidak terlalu banyak. Untuk konklusi dari pengamatan map yang dibuat, map – map bisa dilihat tidak menghalang player dalam bermain dan terlihat tidak jelek secara estetika.

Terakhir dari segi fitness, fitness dari kedua map Template Generation di kiri cepat meraih skor maksimal dari fitness, yaitu 64 dikarenakan ukuran kromosom yang kecil dan 4 fitness yang mudah dipenuhi syaratnya. Di fitness kedua map Tile Generation di kanan terdapat sedikit kesulitan untuk meraih map dengan nilai fitness 128 dikarenakan setiap tile diperhitungkan dalam membuat sebuah map, dikarenakan itu fitness graph di milik Tile Generation memerlukan lebih banyak generasi untuk meraih fitness maksimal.

KESIMPULAN

Bagian Algoritma Genetik dapat digunakan untuk membuat sebuah Procedural Map Generation yang bagus, mungkin setara atau lebih bagus dengan metode – metode Procedural Map Generation lainnya selama Fitnesss yang diterapkan pada algoritma genetik didesain dengan baik.

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.

COPYRIGHT

[Creative Commons License](https://creativecommons.org/licenses/by-nc-sa/4.0/)This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

DAFTAR PUSTAKA

1. Barriga, Nicholas A. *Short Introduction to Procedural Content Generation Algorithms for Videogames.* Chile : University of Talca. 2018.
2. Kraner, Vid et. All. *Procedural Content Generation of Custom Tower Defense Game using Genetic Algorithms.* University of Maribor, Slovenia. 2021
3. Putra, Samuel and Istiono, Wirawan. *Implementation Simple Additive Weighting in Procedural Content Generation Strategy Game.* Universitas Multimedia Nusantara, Banten. 2022
4. Olsson, Niclas and Frank, Elias. *Procedural city generation using Perlin noise.* Blekinge Institute of Technology. Sweden. 2017
5. Sainio, Niko. *Terrain Generation Algorithms*. Tampere University. Finland. 2023.
6. Katoch, Sourabh. *A review on genetic algorithm: past, present, and future*. Multimed Tools Appl. 2020.
7. Lambora, Annu. *Genetic Algorithm- A Literature Review.* 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon). Faridabad. 2019
8. Haldurai, L. *A Study on Genetic Algorithm and its Applications*. Kongunadu Arts and Science College. 2016.
9. Mirjalili, S.*Genetic Algorithm*. Evolutionary Algorithms and Neural Networks. 2018.
10. UNACADEMY. Analytical Geometry-Two Dimensions-Distance Formula. [Online]. Available: https://unacademy.com/content/nda/study-material/mathematics/analytical-geometry-two-dimensions-distance-formula/
11. S, Prayudani et al. *Analysis Effect of Tournament Selection on Genetic Algorithm Performance in Traveling Salesman Problem (TSP)*. Universitas Sumatera Utara, Medan. 2020.
12. Sohal, Asha and Yadav, Saneh Lata. *Comparative Study of Different Selection Techniques in Genetic Algorithm*. Institute of Engineering & Technology, Hyderabad. 2017.
13. Setiawan, Jessen Yaputra et al. *Algoritma Genetika dengan Roulette Wheel Selection dan Arithmetic Crossover untuk Pengelompokan*. Teknik Informatika Universitas Tarumanegara, Jakarta. 2019.
14. Pachuau, Joseph L. et All. *An Overview of Crossover Techniques in Genetic Algorithm* Systems and Technologies, vol 206, Singapore. 2021.
15. Kora, Padmavathi and Yadlapalli, Priyanka. *Crossover Operators in Genetic Algorithms: A Review*. Institute of Engineering & Technology, Hyderabad. 2017.
16. Malik. Ashima. *A Study of Genetic Algorithm and Crossover Techniques.* nternational Journal of Computer Science and Mobile Computing. 2019.
17. Manzoni, Luca. et All. *Balanced crossover operators in Genetic Algorithms.* Dipartimento di Matematica e Geoscienze, Italy. 2020.
18. Abed-Alguni, Bilal H. *Island based Cuckoo Search with Highly Disruptive Polynomial Mutation*. Yarmouk University, Jordan. 2019.
19. Hassanat, Ahmad et All. *Choosing Mutation and Crossover Ratios for Genetic Algorithms—A Review with a New Dynamic Approach*. University of Tabuk, Saudi Arabia. 2019.
20. Guariso, Giorgio and Sangiorgio, Matteo. *Improving the Performance of Multiobjective Genetic Algorithms: An Elitism-Based Approach*. Politecnico di Milano. Italy. 2020.
21. Du, Haiming et All. *Elitism and Distance Strategy for Selection of Evolutionary Algorithms*. School of Electrical and Information Engineering, Zhengzhou. 2018.