BAB II

GENETIC ALGORITHM

Dalam bab ini akan dijelaskan mengenai dasar-dasar mengenai Genetic Algorithm atau juga disebut algoritma Genetik. Algoritma Genetik adalah sebuah Algoritma yang terinspirasi oleh proses seleksi alam, diciptakan oleh John Holland bersama dengan tim-nya pada tahun 1960. Algoritma Genetik dapat dipakai untuk menemukan pilihan terbaik dari sebuah permasalahan yang memiliki beberapa solusi, seperti catur, optimisasi Decision Tree dan banyak lagi. [[1]](#footnote-1)

1. Terminologi

Di dalam bab algoritma Genetik, akan ada beberapa istilah yang berhubungan dengan biologi, seperti Chromosome dan Gene. Dikarenakan itu, akan ada beberapa kata yang jarang ditemukan di dalam kehidupan sehari–hari. Jadi sebelum mendalami Genetic Algorithm, dalam subbab ini akan diperdalam mengenai kata-kata asing yang akan muncul dalam algoritma Genetik, yaitu:

1. Gene & Chromosome (Kromosom)

Chromosome adalah sebuah representasi calon solusi yang dibuat oleh algoritma Genetik, umumnya direpresentasikan menggunakan sebuah array 1 dimensi. Sementara bagian – bagian yang membentuk Kromosom adalah sebuah Gene, yang merepresentasikan sebuah variabel dalam kromosom, direpresentasikan menggunakan angka, huruf ataupun angka biner. Gene dapat diisi dengan tipe data apapun sesuai dengan permasalahan yang ditemukan, seperti char,int atau binary.

1. Populasi / Generasi

Populasi adalah kumpulan dari kromosom – kromosom yang ada. Dalam sebuah populasi dapat berisi satuan hingga ratusan jumlah kromosom yang nantinya akan mempengaruhi kecepatan dari Algoritma tetapi akan meningkatkan kemungkinan mendapat solusi yang lebih baik. Generasi adalah nama lain dari sebuah populasi dimana Generasi menampilkan iterasi dari populasi tersebut. Seperti sebuah Generasi ke-7 menandakan bahwa Generasi tersebut adalah populasi ke-7 yang dibuat oleh Genetic Algorithm.



Gambar 2.1

Visualisasi Gene, Kromosom dan Populasi

1. Fitness

Fitness adalah sebuah angka yang menunjukkan kualitas jawaban dari sebuah kromosom, cenderung semakin tinggi Fitness-nya semakin baik kualitasnya. Semisal ada Kromosom pertama dengan Fitness 50 dan Kromosom ke-3 dengan Fitness 300, maka dilihat bila Kromosom ke-3 lebih baik daripada kromosom pertama, tetapi bisa juga ditentukan apabila bila Fitness semakin rendah menandakan solusi Chromosome tersebut semakin bagus, tergantung dari permasalahan yang dihadapi. Nilai Fitness dapat diperoleh menggunakan Fitness Function buatan kita sendiri yang akan dijelaskan di dalam subbab selanjutnya.

1. Parent & Child Chromosome

Parent Chromosome adalah Chromosome yang dipilih berdasarkan Fitness Chromosome itu sendiri. Parent tersebut akan dipasangkan dengan parent lain untuk membuat sepasang atau lebih anak Chromosome baru. Anak tersebut dipanggil Child Chromosome dan akan menjadi parent dalam Generasi selanjutnya.

1. Metodologi

Genetic Algorithm didasarkan pada seleksi alam, dimana didalam sebuah popuulasi spesies makhluk hidup, hanya makhluk hidup dengan fitur tertentu yang akan bertahan hidup untuk waktu yang lama. Dari konsep tersebut Genetic Algorithm akan menciptakan sejumlah kandidat solusi yang pada algoritma ini akan berevolusi dengan harapan setiap Generasi solusi yang dibuat akan selalu berkembang dan lebih bagus daripada solusi Generasi sebelumnya.[[2]](#footnote-2)



Gambar 2.2

Visualisasi Garis besar cara kerja Algoritma Genetik

Pada gambar 2.2 diatas, ditunjukkan secara garis besar apa yang terjadi di dalam sebuah Algoritma Genetik. Di saat program dimulai dalam “Initial Population” akan dibuat Generasi pertama, biasanya dengan Gene yang isinya acak. Setelah dibuat maka akan diambil nilai Fitness dari setiap kromosom di Generasi pertama tersebut yang diwakilkan dengan tag “Calculate Fitness”.

Setelah itu dilakukan “Selection” yang memilih parent – parent berdasarkan nilai Fitness setiap Chromosome dimana biasanya semakin tinggi Fitness, semakin mungkin dipilih. Lalu terdapat “Crossover”, dimana setiap pasang parent akan membuat anak kromosom baru berisi Gene parent-nya. Pemilihan parent dan pembuatan anak diulangi hingga dapat mengisi sebuah populasi penuh dengan Chromosome-Chromosome baru, setelah itu dilakukan “Mutation” kepada seluruh Chromosome-Chromosome yang dibuat untuk manambah variasi dalam Generasi baru. Tahap terakhir yang tidak terdapat di gambar adalah Elitism, dimana beberapa dari Chromosome-Chromosome dengan Fitness terbaik dari Generasi sebelumnya diletakkan pada Generasi baru untuk mempreservasi Fitness terbaik.

Setelah Generasi ke-2 baru telah dibuat, proses yang tadi dilakukan diulangi mulai dari menghitung Fitness setiap Chromosome. Pengulangan ini akan terjadi terus sampai sebuah syarat berhenti yang ditentukan tercapai, seperti jumlah Generasi maksimal, Fitness terbaik sudah mencapai nilai tertentu, Fitness tidak membaik setelah banyak Generasi, dan banyak lagi. Tapi sebelum syarat berhenti tercapai, algoritma Genetik akan berjalan terus.

1. Langkah-langkah Algoritma Genetik

Setelah dijelaskan Metodologi dan seluruh Terminologi yang diperlukan, maka akan dilanjutkan dalam subbab ini. Pada subbab ini akan dijelaskan tahap – tahap dalam Algoritma Genetik dan apa yang terjadi. Tahap – tahap yang terdapat dalam algoritma Genetik adalah sebagai berikut;

1. Inisialisasi Populasi

Pada tahap ini, dibuatlah populasi pertama atau juga bisa disebut Generasi pertama dengan Chromosome-Chromosome yang memiliki isi Gene acak. Di sini perlu ditetapkan beberapa hal penting, seperti : cara merepresentasi solusi, jumlah kromosom dalam 1 populasi dan jumlah iterasi maksimal agar Algoritma bisa berhenti. Dengan Generasi pertama dibuat, maka algoritma gentik dapat dilanjutkan ke tahap berikutnya.

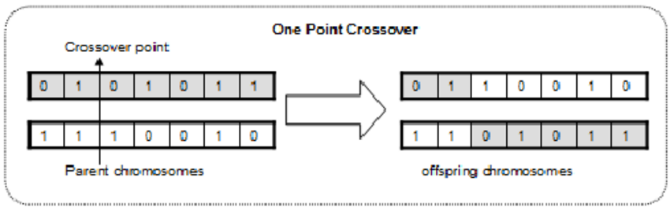
1. Seleksi

Setelah populasi pertama dibuat, maka akan dipilih Chromosome- Chromosome terbaik berdasarkan Fitness mereka. Fitness adalah sebuah nilai yang menunjukkan kualitas Chromosome terhadap permasalah yang dihadapi, cenderung bila nilai Fitness semakin tinggi, maka itu menandakan bila kromosom semakin bagus, tetapi bisa diatur sehingga Fitness lebih rendah menandakan Chromosome dengan solusi yang lebih bagus. Fitness dihitung menggunakan sebuah fungsi, dimana nilai seluruh Gene dalam sebuah Chromosome akan dimasukkan dalam sebuah rumus dengan hasil nilai Fitness dari Chromosome tersebut.

Setelah mendapatkan Fitness seluruh Chromosome dalam sebuah populasi, Chromosome-Chromosome dengan nilai Fitness tertinggi akan dipilih untuk membuat keturunan. Terdapat banyak metode untuk memilih Chromosome, tetapi terdapat 2 unsur utama yang harus ada dalam sebuah metode pemilihan tersebut; pemilihan harus berdasarkan Fitness dan memiliki unsur acak. Sebagai contoh pemilihan Chromosome yang sederhana adalah metode Roulette Wheel, dimana setiap Chromosome yang ada memiliki kesempatan untuk dipilih dimana semakin bagus Fitness-nya, semakin tinggi kemungkinan agar kromosom tersebut dipilih.

1. Reproduksi

Tahap ini akan dilakukan setelah dipilih 2 kromosom yang dirujuk sebagai parent atau orang tua. Menggunakan 2 parent yang telah dipilih, akan diciptakan sepasang anak Chromosome baru menggunakan metode yang dinamakan Crossover, dimana kedua Chromosome parent yang dipilih akan mencampur Gene untuk membuat sepasang anak Chromosome baru. Terdapat banyak metode untuk melakukan Crossover, salah satu contoh yang umum adalah One Point Crossover.



Gambar 2.3

Visualisasi One Point Crossover

Dalam One Point Crossover, dipilih sebuah titik diantara 2 Gene yang akan memisah sebuah Chromosome menjadi 2. Lalu setelah posisi tersebut dipilih, potong kedua Chromosome di posisi tersebut, tukar Chromosome yang telah terpotong, entah yang kiri atau yang kanan, dan sambung kembali kedua Chromosome tersebut. Dari situ telah diciptakan 2 anak Chromosome baru, dan algoritma Genetik kembali ke tahap seleksi untuk memilih parent baru. Proses ini akan diulang hingga jumlah Chromosome anak mencapai jumlah Chromosome yang telah ditetapkan di tahap inisialisasi.

1. Mutation (Mutasi)

Saat seluruh anak Chromosome telah dibuat dan sebelum memasuki iterasi selanjutnya, Mutasi dilakukan terhadap setiap anak Chromosome baru. Dalam mutasi, sebuah kromosom akan mengalami perubahan ke dalam struktur Gene-nya yang dapat terjadi dalam berbagai cara. Salah satunya adalah perubahan isi Gene secara acak yang dinamakan Random Mutation.



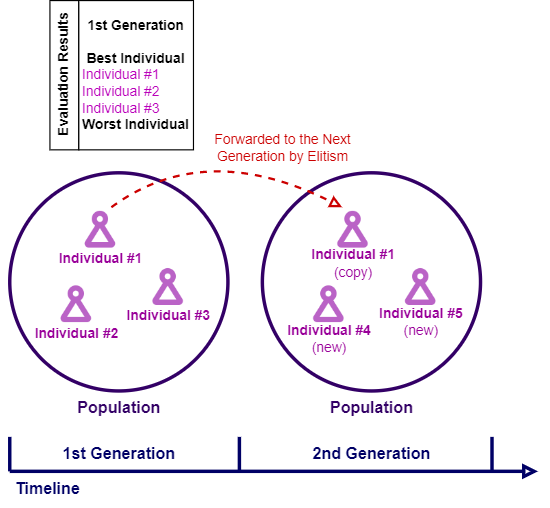
Gambar 2.4

Visualisasi Random Mutation

Ide dari Random Mutation itu sederhana, dimana setiap Gene yang ada di dalam kromosom memiliki kemungkinan untuk mengganti isi sebuah Gene menjadi isi yang baru. Terdapat banyak lagi metode untuk melakukan mutasi, seperti Bit-String mutation dimana setiap Gene dilihat sebagai sebuah angka biner dan setiap bit bisa diganti. Perbedaan utama antara mutasi dan Crossover adalah meskipun mereka memiliki metode yang mirip, yaitu mengubah sebuah kromosom, Crossover mengubah kromosom berdasarkan Gene 2 kromosom, sementara Mutation mengubah sebuah kromosom berdasarkan isi kromosom itu sendiri atau mengubah isi Gene menjadi Gene yang baru.

1. Elitism

Elitism adalah sebuah tahap yang bersifat opsional tapi dapat membantu dalam mencegah penurunan kualitas kromosom. Elitism adalah sebuah tahap dimana sejumlah Chromosome terbaik dalam sebuah populasi dimasukkan dalam populasi Generasi berikutnya, dengan bertujuan bila Chromosome-Chromosome yang terdapat di Generasi baru secara tidak beruntung memiliki tingkat Fitness yang secara keseluruhan lebih rendah daripada Generasi sekarang ini, Chromosome-Chromosome yang dipindah ke Generasi baru dapat menjadi jaminan yang menuntun Chromosome lain untuk menjadi lebih baik.



Gambar 2.5

Visualisasi konsep Elitism

1. Repetisi dan Pemberhentian

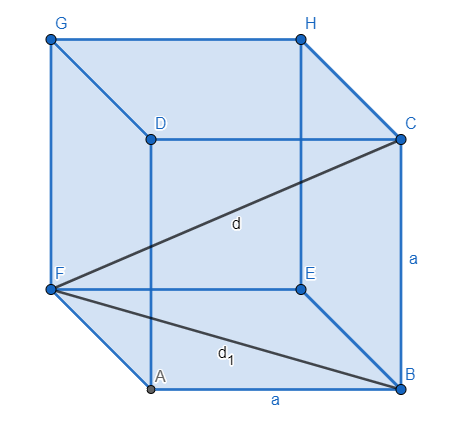
Setalah tahap-tahap di atas telah diselesaikan, algoritma Genetik telah membuat sebuah Generasi baru. Dimana Generasi baru ini diharapankan memiliki nilai Fitness yang lebih baik dibandingkan dengan Generasi sebelumnya. Setelah Generasi baru dibuat, maka algoritma Genetik akan kembali ke tahap ke-2 yaitu seleksi, maju ke tahap ke-3 reproduksi dan siklus ini akan terus berlanjut hingga kondisi berhenti yang telah ditentukan tercapai.

1. Library GeneticSharp

Dalam tugas akhir ini, akan dipakai library yang telah disediakan oleh Diego Giacomelli dengan username Github giacomelli bernama GeneticSharp. GeneticSharp sesuai namanya adalah sebuah library yang telah menyediakan Genetic Algorithm siap dipakai. Dalam subbab ini akan diberi contoh penggunaan GeneticSharp berdasarkan sebuah contoh permasalahan.

1. Permasalahan

Permasalahan yang akan kita pakai sebagai contoh adalah mencari garis terpanjang dalam sebuah kubus tiga dimensi dengan sisi 10 cm. Bagi kita yang sudah mendalami matematika mengetahui bila garis terpanjang dalam sebuah kubus 3D adalah sebuah garis diagonal yang melalui titik tengah kubus. Tetapi algoritma Genetik tidak mengetahui itu, algoritma hanya mengetahui panjang sisi kubus dan harus mencari nilai XYZ dari kedua titik yang akan membuat garis kita.

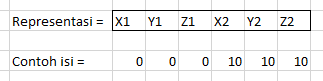


Gambar 2.6

Sebuah Kubus dan 2 garis diagonal

1. Representasi

Representasi dari permasalahan di atas akan menggunakan array integer 1 dimensi dengan panjang 6. Setiap isi array akan diisi dengan sebuah integer yang akan merepresentasikan entah X,Y atau Z dari satu titik. Berikut adalah representasi yang akan dipakai untuk permasalahan ini :



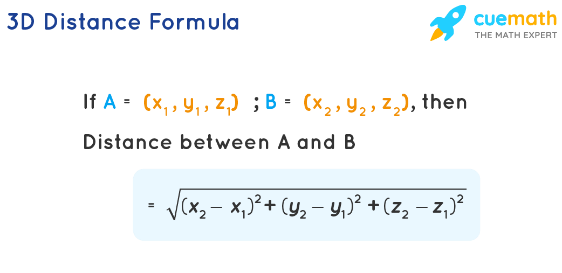
Gambar 2.7

Contoh Representasi permasalahan kubus

Seperti yang bisa dilihat dari gambar 2.7, Representasi yang akan kita pakai dalam permasalahan ini adalah XYZ titik pertama dalam 3 nilai array pertama, dan XYZ titik 2 di 3 nilai array terakhir. Berdasarkan contoh isi di atas tersebut, dari representasi dibawah akan didapat titik 1 dengan posisi 0,0,0 sementara titik ke-2 akan memiliki posisi 10,10,10. Perlu ditegaskan bila ini bukanlah cara yang “paling benar” untuk masalah ini, urutannya bisa diatur sebagai XXYYZZ, atau bisa juga XYZZYX, dan itu juga merupakan sebuah representasi yang valid.

1. Fitness Function

Fitness Function atau fungsi Fitness adalah sebuah fungsi yang dapat digunakan untuk mendapatkan nilai Fitness dari sebuah Chromosome. Menggunakan contoh permasalahan garis terpanjang dari 2 titik dalam sebuah kubus, titik kita bisa berada di mana mana dalam sebuah kubus, jadi tidak cocok untuk menggunakan rumus Pythagoras standar seperti sisi kubus sebagai fungsi Fitness kita. Sebagai pengganti yang lebih layak, kita akan menggunakan Euclidian distance dari 2 titik XYZ sebagai fungsi Fitness kita. Rumus tersebut dapat dilihat di dalam gambar 2.8.



Gambar 2.8

Rumus menghitung jarak antara 2 titik di dalam kubus

Dari rumus ini, dapat diambil panjang dari garis yang dibuat, dan panjang tersebut akan kita gunakan sebagai Fitness value kita. Genetic Algorithm kita akan mengambil Chromosome dengan Fitness value terbanyak pada saat mekalukan tahap Seleksu. Dikarenakan itu, semakin panjang garis yang dibuat dari kedua titik tersebut, maka akan semakin besar kemungkinan kromosom tersebut dipilih untuk beranak.

1. Metode-metode pemrosesan data

Bila dilihat dari penjelasan di bab 2.3, teradpat 3 proses yang memiliki lebih dari 1 pilihan metode, yaitu Seleksi,Crossover dan Mutasi. Untuk permasalahan ini, berikut adalah metode yang dipilih:

* 1. Seleksi

Roulette Wheel Selection adalah sebuah fungsi Seleksi. Dimana akan dilakukan pemilihan random seorang parent seperti memutar roda Roulette, tapi semakin bagus Fitness kromosom tersebut, maka akan semakin besar kemungkinan kromosom tersebut dipilih. Hal ini juga berlaku untuk Fitness yang jelek, dimana Fitness jelek akan memiliki kemungkinan yang lebih kecil untuk dipilih.

* 1. Crossover

Uniform Crossover adalah sebuah metode Crossover, dimana setiap Gene dalam sebuah kromosom dapat ditukar. Kemungkinan penggantian Gene terjadi dengan kemungkinan 50%. Tetapi nilai tersebut dapat diganti oleh user bila diinginkan.

* 1. Mutasi

Uniform Mutation, bersifat mirip dengan Uniform Crossover dimana setiap Gene memiliki kemungkinan untuk berubah. Bila ada Gene yang berubah, maka akan diambil nilai baru secara acak dari range yang telah ditentukan. Kemungkinan dari sebuah Gene melakukan mutasi di Uniform Mutation secara bawaan adalah 1/Jumlah Gene dalam kromosom, ini berguna agar hanya sedikit Gene yang berubah dalam setiap Mutasi.

1. Pseudocode dan Penjelasan

Sekarang dengan seluruh variabel dan metode dipersiapkan dalam subbab-subbab diatas, dalam subbab ini akan diberikan sebuah Pseudocode yang menggunakan GeneticSharp untuk menyelesaikan permasalahan garis terpanjang dalam sebuah kubus. Untuk mempersingkat penjelasan, akan digunakan Class-Class siap pakai oleh GeneticSharp. Berikut adalah Pseudocode dari permasalahan kubus diatas:

Algoritma 2.1 Pseudocode Library GeneticSharp

1. Menggunakan library GeneticSharp.Domain.\*
2. varKromosom = new FloatingPointChromosome(
3. Array batas minimum Gene => [0,0,0,0,0,0],
4. Array batas maximum Gene => [10,10,10,10,10,10],
5. Array jumlah bit Gene => [3,3,3,3,3,3],
6. Array jumlah bilangan desimal => [0,0,0,0,0,0]
7. )
8. varPopulasi = new Population(
9. minPopulasi,maxPopulasi,kromosom => varKromosom
10. )
11. varFungsiFitness = new FuncFitness((c)=>
12. kromosomSekarang = c as FloatingPointChromosome
13. arrKromosom = kromosomSekarang.ToFloatingPoints()
14. X1 = arrKromosom[0]
15. Y1 = arrKromosom[1]
16. Z1 = arrKromosom[2]
17. X2 = arrKromosom[3]
18. Y2 = arrKromosom[4]
19. Y2 = arrKromosom[5]
20. euclDist = Akar2(Pangkat2(X1-X2)+ Pangkat2(Y1-Y2) + Pangkat2(Z1-Z2))
21. return euclDist
22. )
23. varMetodeSeleksi = new RouletteWheelSelection()
24. varCrossover = new UniformCrossover(kemungkinan crossover terjadi {0-1})
25. varMutasi = new UniformMutation()
26. varStopCon = new GenerationNumberTermination(100);
27. varGA = new GeneticAlgorithm(
28. Populasi => varPopulasi,
29. Fitness => varFungsiFitness,
30. Metode Seleksi => varMetodeSeleksi,
31. Crossover => varCrossover,
32. Mutasi => varMutasi
33. )
34. varGA.Termination = varStopCon
35. varGA.Start();
36. hasilJadi = varGA.BestChromosome

Sekarang akan diperjelas kegunaan dari masing – masing Line of Code ;

1. Line 2-7

Dalam membuat sebuah kromosom, GeneticSharp memerlukan 4 parameter, yaitu : batas maksimal Gene, batas minimal Gene, jumlah bit Gene, dan jumlah bilangan desimal, dimana semua variabel tersebut disimapan dalam sebuah array. Batas maksimal dan minimal Gene berguna untuk mengatur batas dari isi Gene dalam membuat populasi awal, ini berguna dikarenakan setiap Gene memiliki batas maksimal yang berbeda, dalam kasus ini akan diisi array dengan panjang 6 berisi 0 dan 10 untuk batas min dan max secara berurutan dikarenakan panjang sisi dari setiap kubus adalah 10cm dan sebuah titik tidak bisa berada di luar kubus.

Parameter selanjutnya adalah jumlah bit dikarenakan GeneticSharp menggunakan mewarisi Class BinaryChromosomeBase yang hanya menggunakan angka biner, dipilih 10 karenakan 10 hanya memerlukan 3 bit maka parameter ini diisi 3. Parameter terakhir adalah jumlah angka desimal. Disini bisa diisi sesuka hati, tapi untuk kesederhanaan diisi 0 agar tidak ada angka desimal dalam Gene.

GeneticSharp juga menyediakan Class IntegerChromosome yang memiliki Chromosome numerik dan Gene angka biner yang mendirikan angka numerik tersebut, lalu ada BinaryChromosomebase yang merupakan basis dari 2 kelas Kromosom yang lainnya. Bila user tidak puas dengan pilihan ini maka user dapat membuat sebuah Class Chromosome baru yang wajib mewarisi class ChromosomeBase, dan melakukan Override Function tertentu.

1. Line 8-10 :

Di Line ini ditentukan jumlah minimum dan maximum jumlah Chromosome dari sebuah populasi. Minimum kromosom akan dipakai dalam inisialisasi pembuatan Generasi pertama, sementara maximum kromosom adalah batas maksimal kromosom dalam setiap Generasi. Lalu terakhir di parameter ketiga adalah kromosom yang telah dibuat di line 3.

1. Line 11–22

Di sini, dapat dibuat sebuah fungsi Fitness yang akan menghitung Fitness dari semua Chromosome yang ada. Gene-Gene dari Chromosome yang diperiksa dapat diambil menggunakan kode di line 12 dan 13, lalu dengan array Chromosome tersebut, kita bisa mengambil nilai XYZ dari kedua titik berdasarkan representasi yang telah ditentukan sebelumnya. Dengan XYZ dari kedua titik diambil, kita dapat mencari Euclidian Distance dari kedua titik tersebut sebagai nilai Fitness. dan setelah mendapat Euclidian distance, nilai Fitness tersebut akan di return untuk dipakai dalam selection.

1. Line 23

Line ini dipakai untuk menentukan metode Seleksi kromosom. Dalam kasus kita Roulette Wheel Selection. Metode yang disediakan oleh Diego Giacomelli diantaranya ada Elite, Stochastic Universal sampling, Uniform Crossover dan masih banyak lagi. Bila ingin membuat seleksi sendiri dapat membuat class baru yang mewarisi Class SelectionBase.

1. Line 24

Di sini dipilih Crossover yang akan dipakai oleh Algoritma Genetik. Dalam kasus kita Uniform Crossover,dimana Giacomelli sekali lagi menawari kita dari banyak pilihan seperti One-Point, Two-Point, Cut and spliced, dan banyak lagi. Crossover buatan sendiri bisa dibuat juga tetapi mewarisi Class CrossoverBase.

1. Line 25

Sekarang baris kode ini dipakai untuk memilih mutasi yang akan digunakan dalam algoritma Genetik. Dalam permasalahan ini akan dipakai Uniform Mutation. Beberapa pilihan yang lain diantaranya adalah; Flip-Bit, Reverse Sequence dan Twors. Bila ingin membuat mutasi khusus harus mewarisi Class MutationBase yang telah disediakan.

1. Line 26

Kita dipersilahkan memilih dari berbagai kondisi berhenti untuk algoritma Genetik kita, beberapa faktor yang tersedia adalah Jumlah Generasi, Lama Evolusi, Fitness yang tidak naik naik, dan Fitness yang telah mencapai nilai tertentu. Tapi dalam kasus ini kita akan memilih Generation Number Terminantion dimana setelah 100 iterasi Algoritma Genetik akan berhenti. Seperti biasa dapat menggunakan Class sendiri yang mewarisi Class TerminationBase.

1. Line 27 – 35

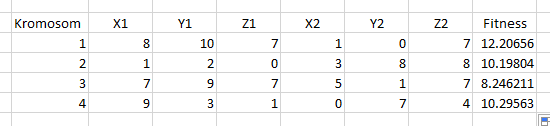
Di sini seluruh pilihan dan fungsi yang telah dibuat sebelumnya dimasukkan dalam Constructor Genetic Algorithm yang baru ini. Parameter yang dimasukkan merupakan seluruh variabel yang sebelumnya telah dipersiapkan mulai dari populasi hingga mutasi. Lalu di line 35 algoritma Genetik dijalankan dan mulai mencari solusi hingga nilai Fitness mengalami stagnasi.

1. Line 36

Setelah Genetic Algorithm selesai dijalankan, maka hasilnya akan disimpan dalam sebuah variabel yang ditentukan. Hasil ini dapat diakses untuk menunjukkan jawaban terbaik yang telah ditemukan oleh Algoritma Genetik dan dapat diaplikasikan ke masalah yang diperlukan.

1. Tracing

Dengan pseudocode yang sudah dijelaskan, kita akan melihat apa yang terjadi di dalam Algoritma Genetik dengan melakukan tracing (melaksanakan kode menggunakan secara manual) terhadap algoritma Genetik selama 1 Generasi dengan 5 kromosom dalam tiap populasi (karena kalau kita menggunakan 100 kromosom per populasi akan memakan waktu yang lama). Untuk inialisasi, akan diambil angka acak untuk setiap angka dalam Gene. Hasilnya adalah sebagai berikut:



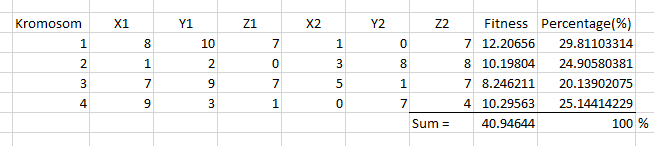
Gambar 2.9

Generasi Pertama

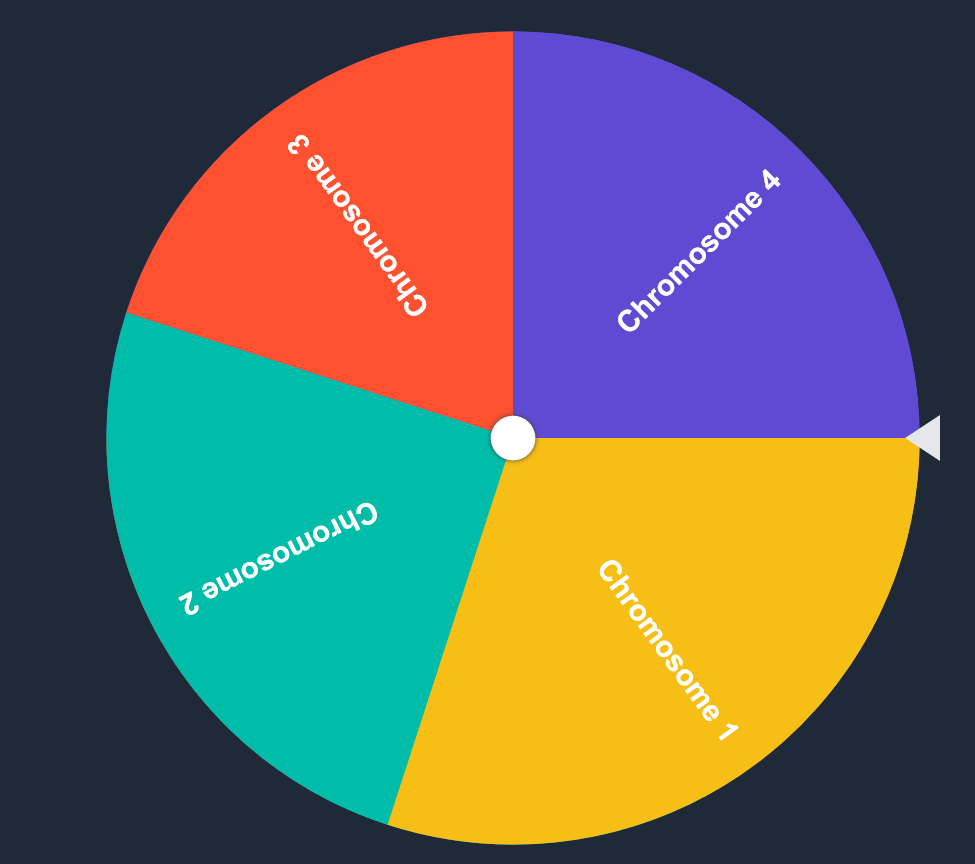
Dengan angka yang didapat secara acak, kita dapat menghitung Fitness dari ke-empat kromosom tersebut. Kita akan coba menghitung Kromosom yang pertama dengan metode Euclidian Distance untuk mencari Fitnessnya. Berikut adalah hasil perhitungannya:

(2.1)

Setelah menghitung Fitness dari Chromosome pertama, kita dapat melakukan hal yang sama terhadap 3 Chromosome lainnya. Dimana hasilnya telah tertera di gambar 2.10 sebelah kanan. Dengan nilai Fitness yang didapat, kita akan memasukkan ke-4 Fitness tersebut ke dalam sebuah roda Roulette. Dan dengan itu akan diputar Roulette Wheel-nya untuk mengambil Chromosome yang beruntung.

****

(a)

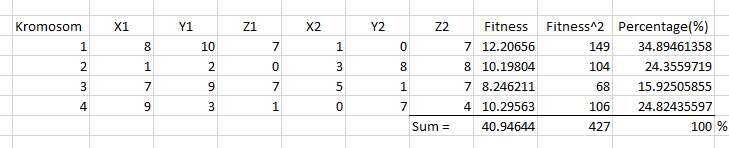
****

**(b)**

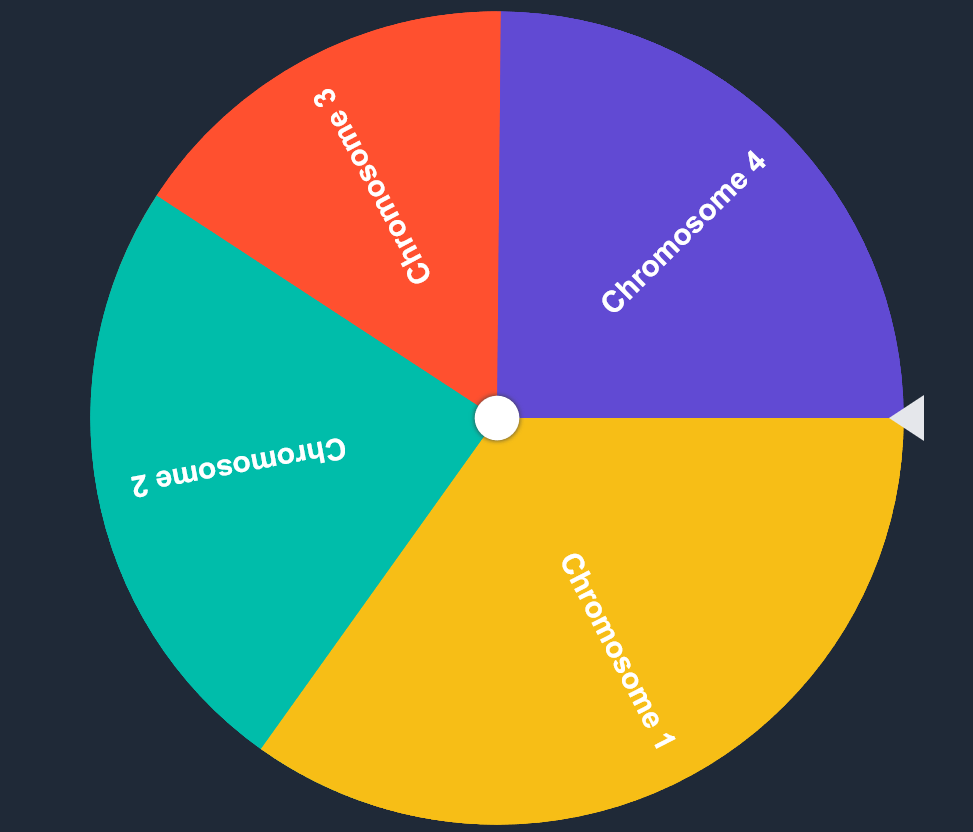
Gambar 2.10

**(a) Persentase seluruh Populasi dan (b) Visualisasi roda Roulette**

Persentase dari setiap kromosom dapat dilihat di gambar 2.10 beserta dengan visualisasi roda Roulette-nya. Bisa dilihat bila Kromosom 1 memikiki daerah yang lebih besar dibandingkan dengan kromosom yang lainnya, khususnya kromosom 3 yang hanya memiliki persentase 20%. Tapi meskipun begitu, bisa disadari perbedaan dari tiap kromosom tidak terlalu mencolok, sehingga kromosom 3 masih memiliki kemungkinan yang bagus untuk dipilih. Cara menangani permasalahan ini adalah dengan memberi pangkat terhadap nilai Fitness kita, sehingga perbedaan antar Fitness dapat lebih mencolok lagi. Seperti berikut;



(a)

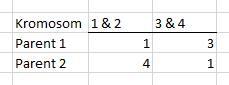


**(b)**

Gambar 2.11

**(a) Persentase dan (b) Visualisasi Fitness kuadrat**

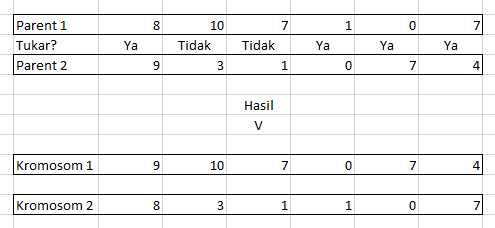
Dengan menghitung persentasi Fitness kuadrat, maka visualisasi yang didapat lebih menguntungkan Kromosom 1, dan Kromosom 3 lebih tidak mungkin untuk dipilih. Fitness bisa diberi pangkat sesuai keperluan user, tapi bagi kita pangkat 2 saja sudah cukup. Selanjutnya menggunakan roda roulette ini pilih parent Chromosome untuk Generasi selanjutnya. Pilih sepasang kromosom sebanyak 2 kali dikarenakan sepasang parent membuat 2 pasang anak. Ingat bila sepasang Chromosome tersebut tidak boleh Chromosome yang sama. Inilah hasil yang didapat menggunakan situs Random Number Generator :

****

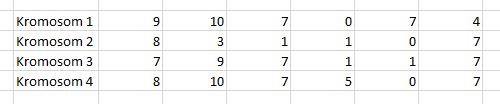
Gambar 2.12

**Hasil Memutar roda Roulette**

Dan dengan parent dipilih, kita dapat memasuki tahap selanjutnya yaitu Crossover, dimana sepasang parent akan menukar Gene satu sama lain. Kita menggunakan metode Uniform Crossover,yang telah dijelaskan pada subbab 2.4.4B. Dan dengan itu dilakukan tracing terhadap pasangan kromosom yang pertama. Inilah hasil yang didapat dari Crossover tersebut :



**(a)**

****

**(b)**

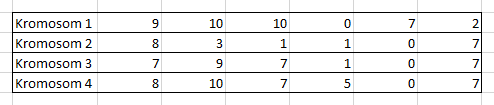
Gambar 2.13

**Crossover (a) pasangan kromosom pertama dan (b) hasil keseluruhan**

Tahap terakhir setelah membuat Generasi baru adalah untuk melakukan mutasi terhadap setiap Kromosom yang ada. Mutasi kita adalah Uniform Mutation dimana setiap Gene yang terpilih akan berubah menjadi Gene baru berdasarkan batas yang kita tentukan di awal. Kita akan melakukan Mutation ini terhadap Chromosome baru yang pertama. Perlu diingat karena satu kromosom memiliki 6 Gene, maka ditentukan bila setiap Gene memiliki kemungkinan untuk melakukan mutasi.



**(a)**



**(b)**

Gambar 2.14

**Mutasi (a) kromosom pertama dan (b) hasil keseluruhan**

Setelah melakukan mutasi terhadap kemungkinan pertama,dilakukan hal yang sama terhadap 3 kromosom lainnya. Dalam tracing ini, yang berubah hanya Gene ke 5 di Chromosome 3. Setelah mutasi dilakukan, maka Generasi ke – 2 ini siap dipakai untuk menemukan Generasi ke-3. Dengan itu dilanjutkan ke tahap Elitism yang tidak akan dijelaskan dalam subbab ini dikarenakan jumlah Chromosome yang kecil.

Dan dengan tahap mutasi selesai, Generasi ke – 2 siap memasuki tahap seleksi, dilanjutkan dengan Crossover dan diakhiri dengan Mutasi. Tahap ini akan berulang – ulang terus hingga mencapai 100 Generasi. Selama itu, Chromosome terbaik akan selalu diperbarui dan disimpan, jadi setelah algoritma genetik selesai, Chromosome terbaik dapat diambil dan dipakai untuk permasalahan yang dihadapi, dalam kasus ini menentukan garis terpanjang dalam sebuah kubus bersisi 10cm.

1. Xin-She Yang, Yang. *Genetic Algorithms. Nature-Inspired Optimization Algorithms.* X.-S. (2014) [↑](#footnote-ref-1)
2. Vijini Mallawaarachchi, *Introduction to Genetic Algorithms — Including Example Code* (https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3, diakses 27 Januari 2023) [↑](#footnote-ref-2)