

## Genetic Algorithm

- Algoritma ini bekerja dengan sebuah populasi yang terdiri atas individu-individu (kromosom). Individu dilambangkan dengan sebuah nilai kebugaran (*fitness*) yang akan digunakan untuk mencari solusi terbaik dari persoalan yang ada. Pertahanan yang tinggi dari individu memberikan kesempatan untuk melakukan reproduksi melalui perkawinan silang dengan individu yang lain dalam populasi tersebut. Individu baru yang dihasilkan dalam hal ini dinamakan keturunan (*offspring*), yang membawa beberapa sifat dari induknya. Sedangkan individu dalam populasi yang tidak terseleksi dalam reproduksi akan mati dengan sendirinya.

## Perbedaan GA

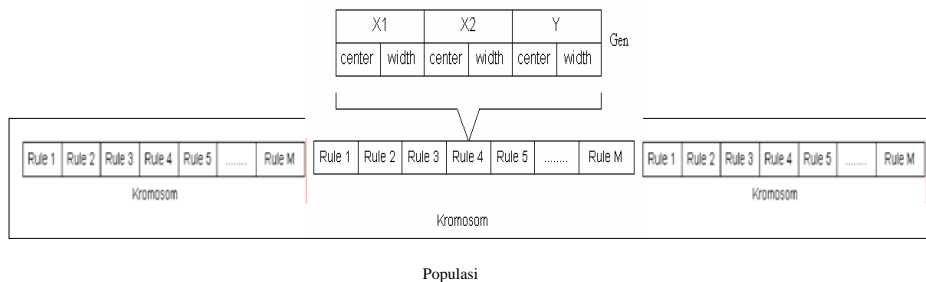
Perbedaan mendasar yang dimiliki GA's dibandingkan dengan metode dan algoritma pencarian lainnya adalah :

- GA memanipulasi kode-kode set parameter, bukan manipulasi nilai parameter tersebut.
- GA melakukan pencarian pada waktu tertentu di beberapa titik sekaligus (populasi), tidak pada satu titik
- GA menggunakan fungsi obyektif sebagai referensi pencarian, tidak berdasarkan suatu nilai turunan atau informasi lain
- GA menggunakan aturan transisi probabilistik, bukan aturan deterministik.

## Istilah GA

| Istilah   | Keterangan  |
|-----------|---|
| Kromosom  | Individu berupa segmen string yang sudah ditentukan             |
| Gen       | Bagian dari string  |
| Loci      | Posisi dari gen   |
| Allele    | Nilai yang dimasukkan dalam gen                                 |
| Phenotype | String yang merupakan solusi terakhir                           |
| Genotype  | Sejumlah string hasil perkawinan yang berpotensi sebagai solusi |

## Contoh Struktur GA untuk Identifikasi Parameter Fuzzy Logic



## Istilah GA

- Jumlah Generasi (MAXGEN) :

Merupakan jumlah perulangan (iterasi) dilakukannya rekombinasi dan seleksi. Jumlah generasi ini mempengaruhi kestabilan output dan lama iterasi (waktu proses Genetic Algorithms). Jumlah generasi yang besar dapat mengarahkan ke arah solusi yang optimal, namun akan membutuhkan waktu yang lama. Sedangkan jika jumlah generasinya terlalu sedikit maka solusi akan terjebak pada lokal optimum.

- Ukuran Populasi (POPSIZE) :

Ukuran populasi mempengaruhi kinerja dan efektifitas dari Genetic Algorithms. Ukuran populasi kecil maka populasi tidak menyediakan cukup materi untuk mencakup ruang permasalahan, sehingga pada umumnya kinerja GA's menjadi buruk. Selain itu penggunaan populasi yang besar dapat mencegah terjadinya konvergensi pada wilayah lokal, banyak aplikasi GA's menggunakan populasi pada range 50-100.

## Istilah GA

- Probabilitas Crossover ( $P_c$ ) :

Probabilitas crossover ini digunakan untuk mengendalikan frekuensi operator crossover. Dalam populasi terdapat  $P_c \times POPSIZE$  struktur (individu) akan melakukan pindah silang. Semakin besar nilai probabilitas crossover maka semakin cepat struktur baru diperkenalkan dalam populasi. Namun jika probabilitas crossover terlalu besar maka struktur dengan nilai fungsi obyektif yang baik dapat hilang dengan lebih cepat dari seleksi. Sebaliknya probabilitas crossover kecil akan menghalangi proses pencarian dalam GA's. banyak aplikasi GA's menggunakan angka probabilitas crossover pada range 0,65-1.

- Probabilitas Mutasi ( $P_m$ ) :

Mutasi digunakan untuk meningkatkan variasi populasi. Probabilitas mutasi ini digunakan untuk menentukan tingkat mutasi yang terjadi, karena frekuensi terjadinya mutasi tersebut menjadi  $P_m \times POPSIZE \times N$ , dimana  $N$  adalah panjang struktur/gen dalam satu individu. Probabilitas mutasi yang rendah akan menyebabkan gen-gen yang berpotensi tidak dicoba. Dan sebaliknya, tingkat mutasi yang tinggi akan menyebabkan keturunan kehilangan kemiripan dengan induknya sehingga akan menghancurkan daerah solusi.

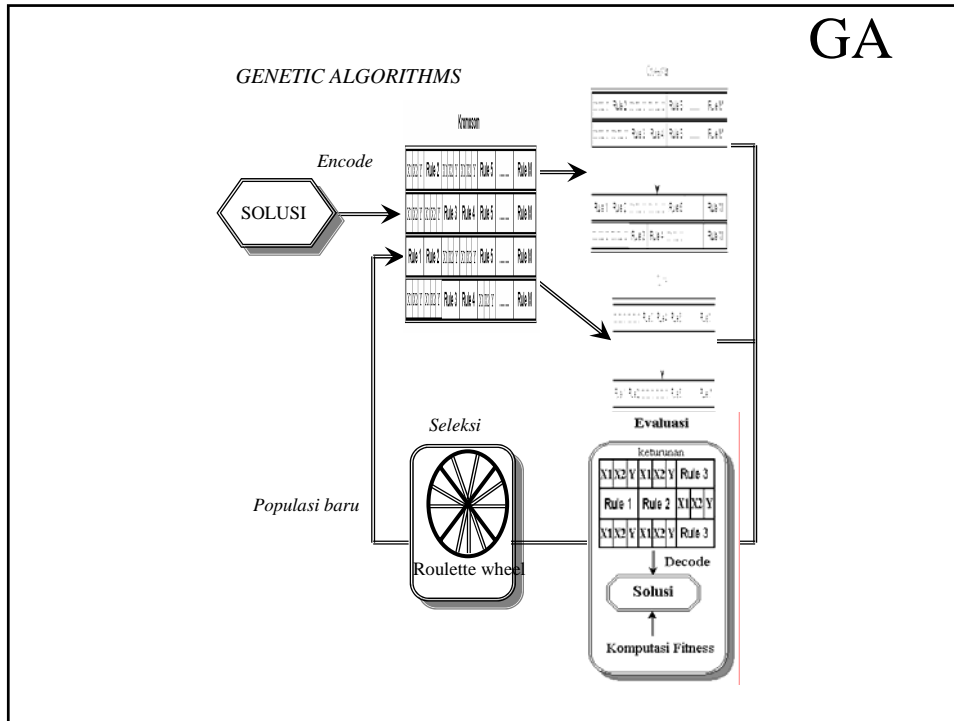
## Istilah GA

Dalam GA's, mutasi menjalankan peranan yang penting yaitu :

1. Penempatan gen-gen yang hilang dari populasi selama proses seleksi.
  2. Pemberian gen-gen yang tidak muncul pada saat inialisasi populasi awal. Banyak aplikasi GA's menggunakan angka probabilitas mutasi pada range 0,001-0,01.
- Panjang Kromosom (*NVAR*) :  
Panjang kromosom berbeda-beda sesuai dengan model permasalahan. Titik solusi dalam ruang permasalahan dikodekan dalam bentuk kromosom/string yang terdiri atas komponen genetik terkecil yaitu gen. Pengkodean dapat memakai bilangan seperti string biner, integer, floating point, dan abjad.

## Algorithma GA

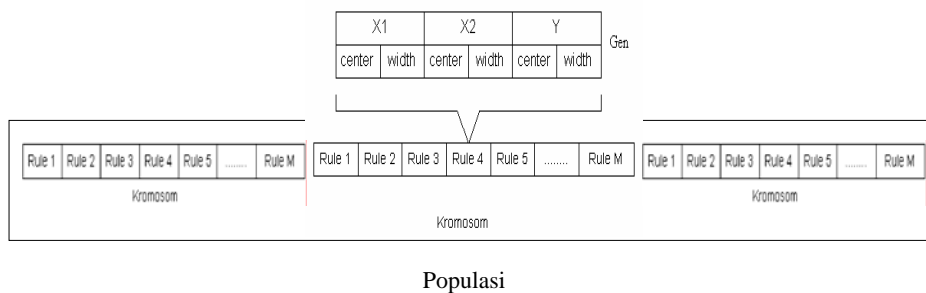
- Inialisasi populasi awal (kumpulan nilai random)
- Mengevaluasi setiap kromosom dalam populasi dengan menggunakan ukuran fitness melalui fungsi tujuan
- Mendapatkan kromosom baru melalui kromosom sebelumnya dengan menggunakan operator mutasi dan operator crossover
- Menggantikan beberapa kromosom dengan kromosom yang baru
- Mengevaluasi kromosom baru dan memasukkan ke dalam populasi



## Algorithma GA

### 1. Pengkodean

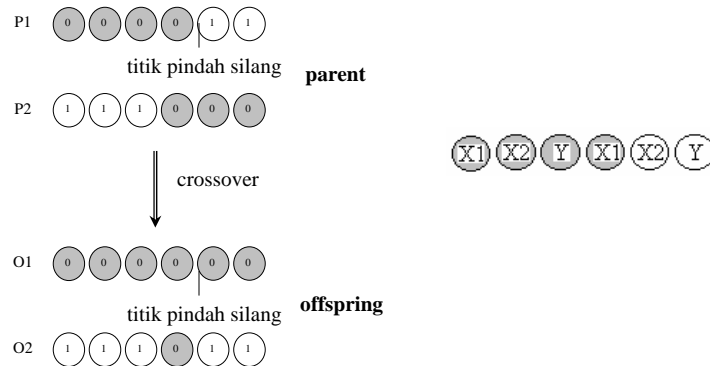
- Mengubah Parameter ke bentuk Gen
- Gen-gen tersebut membentuk kesatuan Kromosom
- Kromosom-kromosom tsb membentuk Populasi



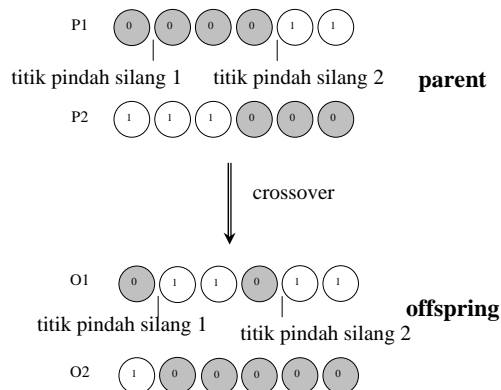
## Algorithma GA : Crossover

### 2. Crossover (Pindah Silang)

- menghasilkan kromosom anak dari kombinasi materi-materi gen dua kromosom induk
- Operator ini bekerja dengan membangkitkan sebuah nilai random  $r_k$  dimana  $k = 1, 2, \dots, \text{POPSIZE}$



- Probabilitas crossover ( $p_c$ ) ditentukan dan digunakan untuk mengendalikan frekuensi operator crossover. Apabila nilai  $r_k < p_c$  maka kromosom ke-k terpilih untuk mengalami crossover
- Posisi titik pindah silang (split point) ditentukan secara random pada range satu sampai panjang kromosom



## Algoritma GA : Crossover

### Crossover untuk variabel real

- **Crossover Arithmatika** : sebagai kombinasi linear dua vektor : bila  $s_v^t$  dan  $s_w^t$  akan disilangkan (crossed) maka keturunan yang akan dihasilkan adalah  $s_v^{t+1} = a.s_w^t + (1-a)s_v^t$  dan  $s_w^{t+1} = a.s_v^t + (1-a)s_w^t$  , dimana  $0 \leq a \leq 1$