

Genetic Programming (GP)

Dr. Suyanto, S.T., M.Sc.

HP/WA: 0812 845 12345

Intelligence Computing Multimedia (ICM)
Informatics faculty – Telkom University

GA

ES

EP

GP?

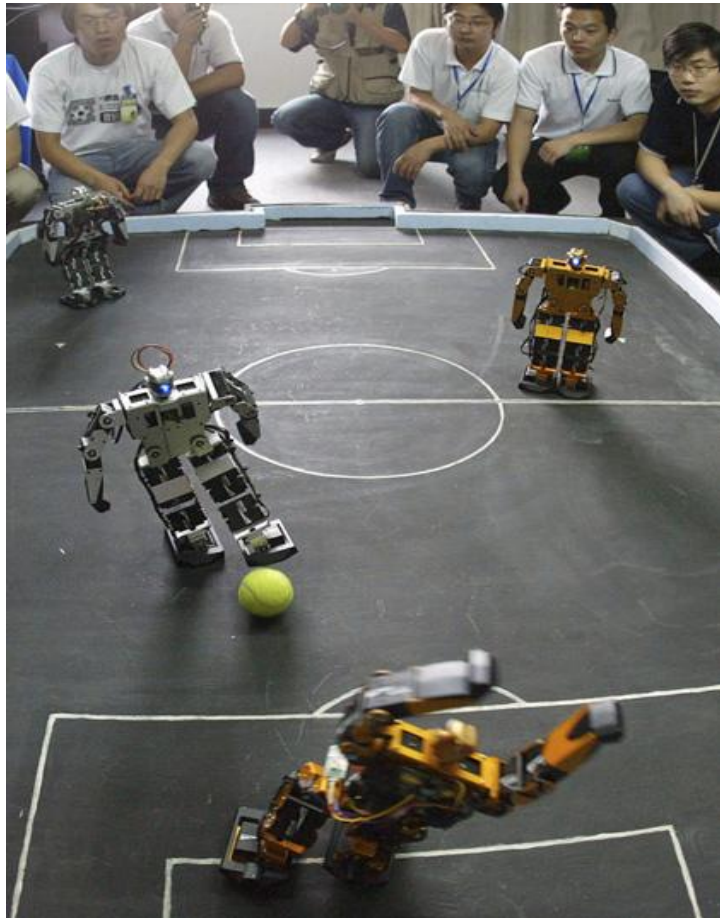


GP for Robot Soccer



WC-2050 is ours !!!





Intro

- Tujuan utama ilmu komputer: *automatic programming*
- Arthur Samuel (1959): “tujuan *automatic programming* adalah bagaimana komputer dapat dibuat mampu mengerjakan hal-hal yang perlu dikerjakan tanpa diberitahu secara pasti bagaimana cara mengerjakannya.
- Misalnya, kita memiliki sangat banyak pasangan data masukan dan keluaran. Bagaimana menemukan suatu program secara otomatis yang bisa memetakan masukan dan keluaran tersebut dengan benar?

Intro

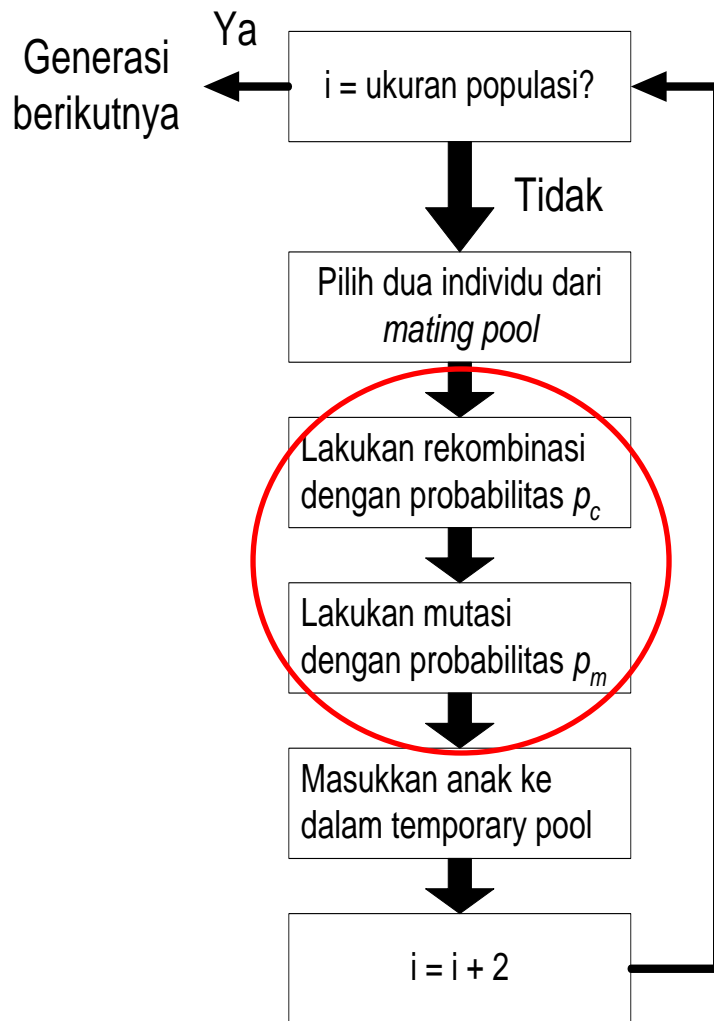
- GP menggunakan representasi *tree* atau *graph*
- Diperkenalkan J. Koza pada era 1990-an di USA
- Untuk *machine learning*: prediksi, klasifikasi, dsb
- Ukuran populasi besar: ribuan individu
- GP lebih lambat dibandingkan GA, ES, dan EP

Intro

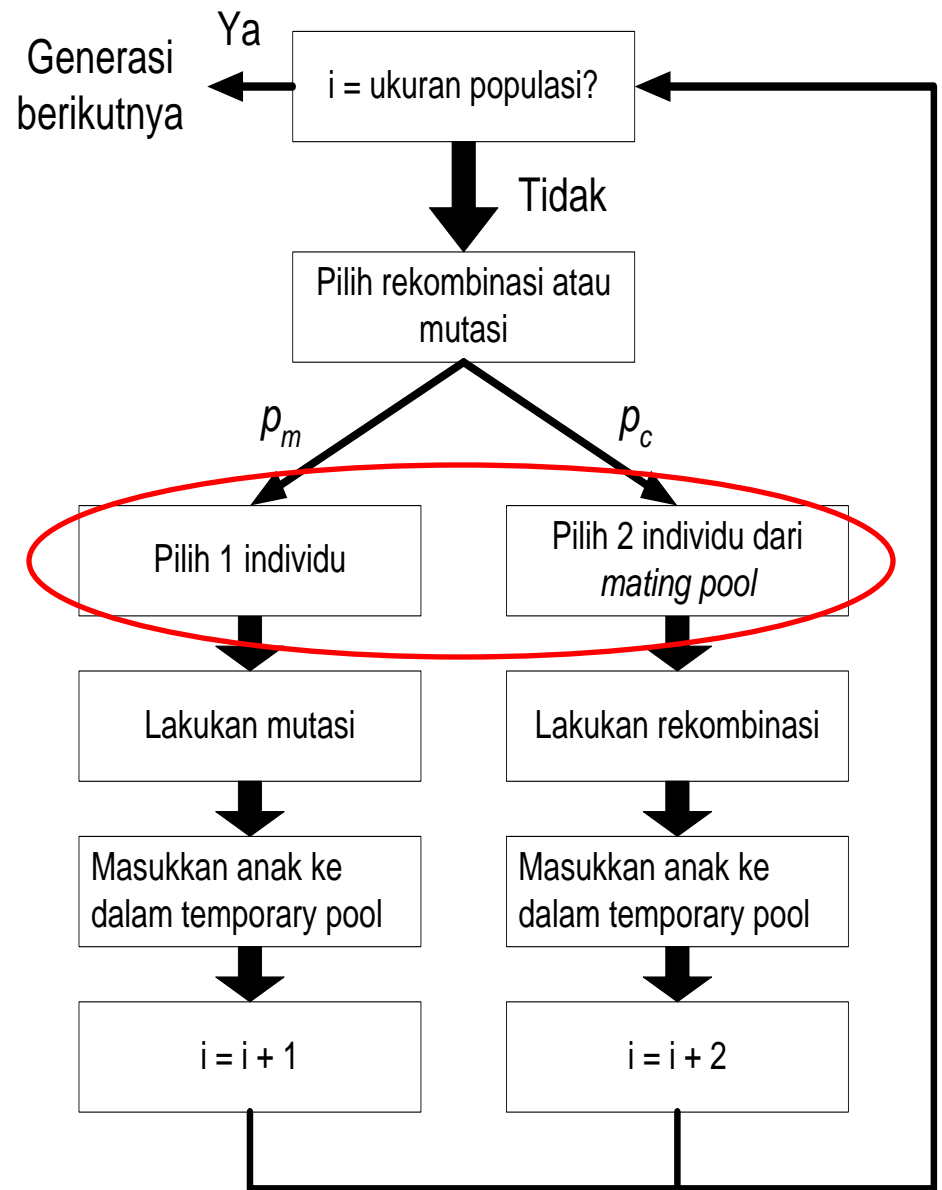
- Dalam satu populasi, kromosom-kromosom bisa memiliki panjang yang berbeda-beda
- GP bisa menggunakan proses rekombinasi dan mutasi dengan probabilitas tertentu
- Rekombinasi dilakukan dengan cara saling menukar ranting (*sub tree*)
- Sedangkan mutasi dilakukan dengan mungubah pohon secara acak

Spesifikasi teknis GP

Representasi	Struktur pohon (<i>tree</i>)
Seleksi orangtua	Proporsional terhadap <i>fitness</i>
Rekombinasi	Pertukaran ranting (<i>sub tree</i>)
Mutasi	Perubahan acak pada <i>tree</i>
Seleksi <i>survivor</i>	<i>Generational replacement</i>
Ciri khusus	Dalam satu populasi, kromosom-kromosom bisa memiliki panjang yang berbeda-beda



GA



GP

Representasi Individu

- Misalkan si A menghadapi masalah dalam proses seleksi pegawai tahun ini
- Dia bingung bagaimana menemukan model yang tepat untuk proses seleksi tahun ini
- Aturan seleksi yang pernah digunakan tahun lalu tidak dia temukan
- Satu-satunya dokumen proses seleksi pegawai tahun lalu adalah data sebagai berikut:

Data histori penerimaan pegawai

Kandidat	IPK	Psikologi	Diterima
1	2,98	73	Tidak
2	3,08	97	Ya
3	2,56	83	Tidak
4	3,01	84	Ya
...

Kromosom

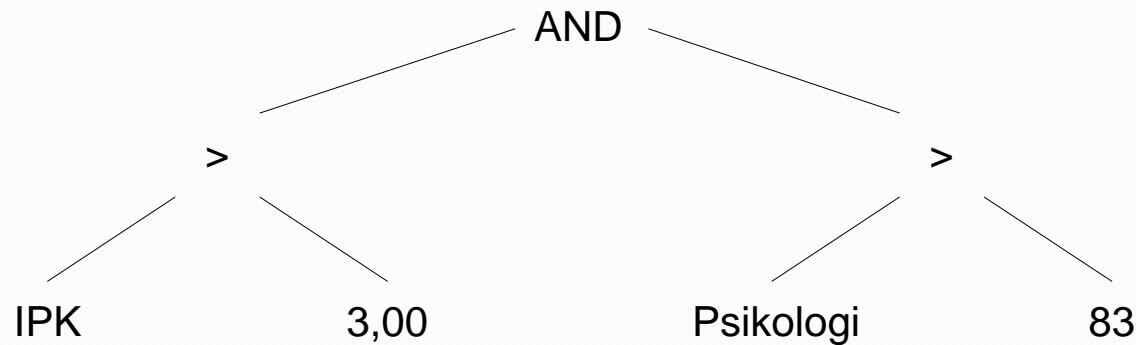
Misalkan, suatu model yang bisa digunakan untuk masalah tersebut adalah

IF (IPK > 3,00 AND Psikologi > 83)

THEN Diterima

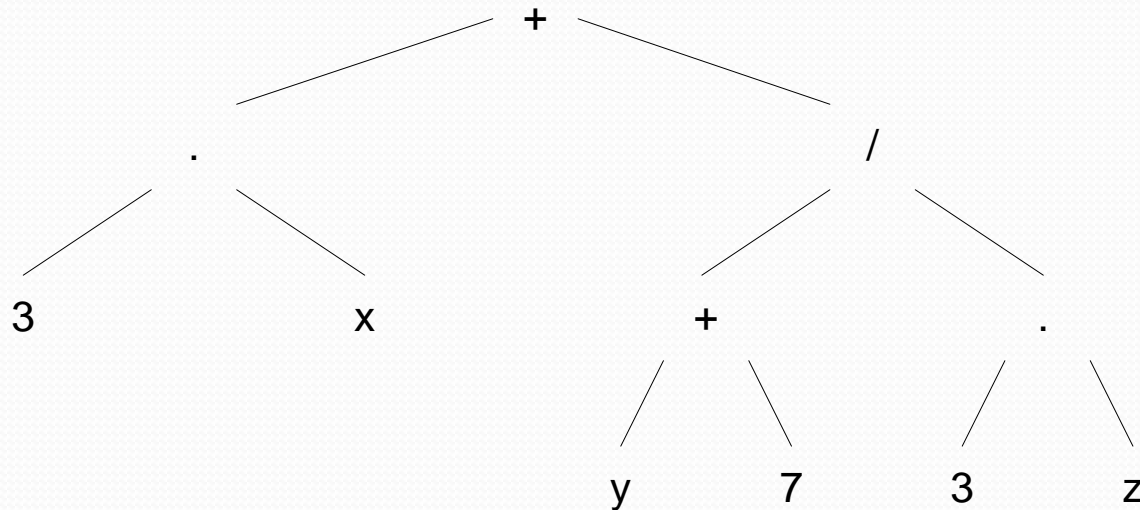
ELSE Tidak.

Kromosom



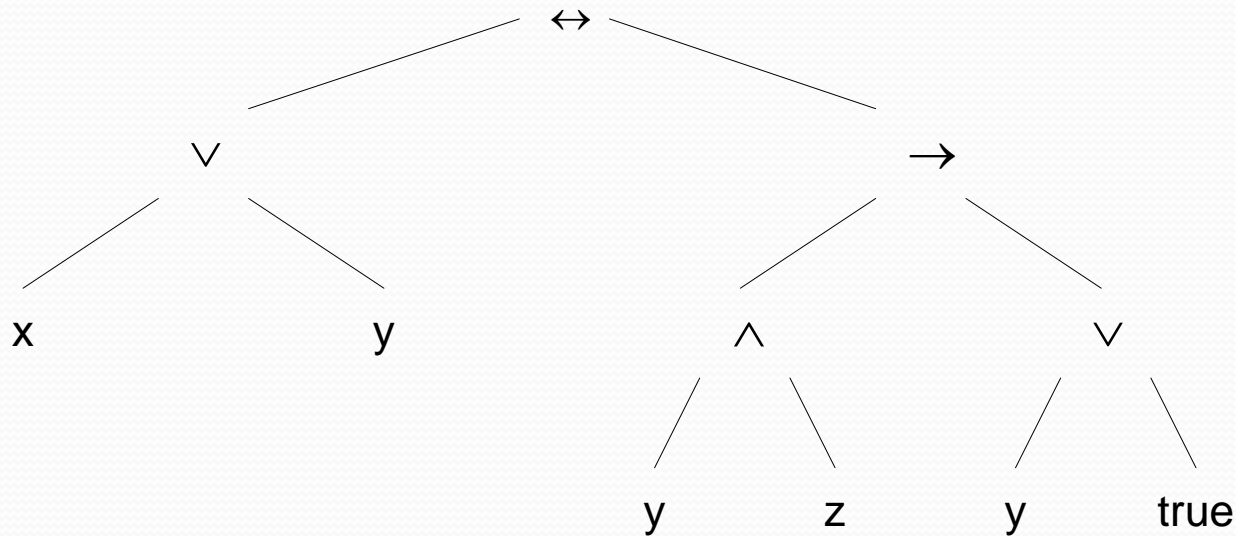
- Struktur pohon tersebut bisa dibayangkan sebagai suatu kromosom yang bisa ber-evolusi dengan proses rekombinasi dan mutasi.
- Untuk permasalahan yang lebih kompleks, misalnya jumlah atribut masukan lebih dari dua, tentu saja diperlukan waktu yang lama untuk menemukan model yang tepat.
- Proses evolusi pada GP sengaja dibuat untuk mengatasi masalah ini.

Kromosom: Formula Aritmetika



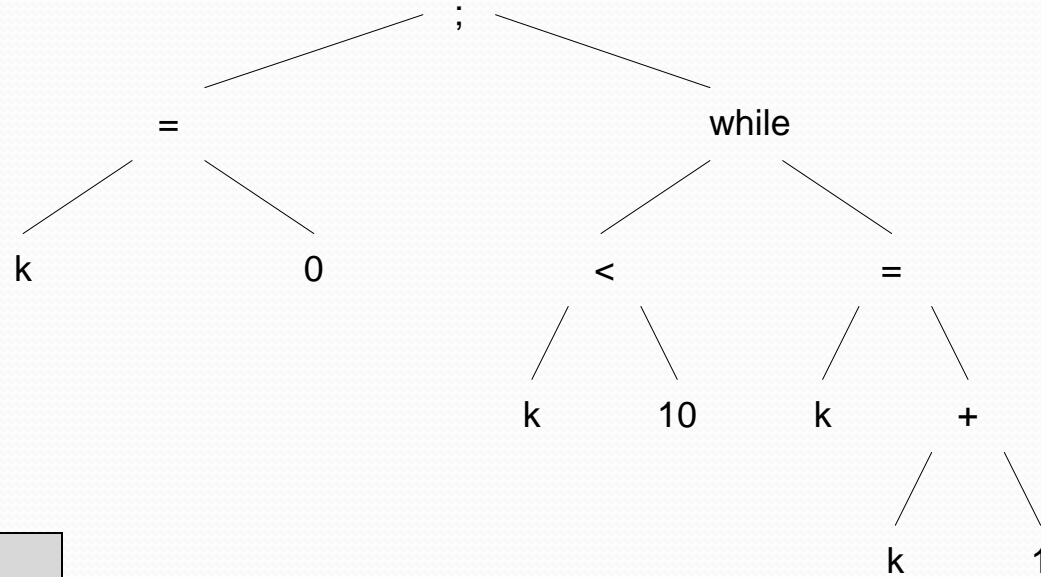
$$3x + \frac{(y + 7)}{3z}$$

Kromosom: Formula Logika



$$(x \vee y) \leftrightarrow ((y \wedge z) \rightarrow (y \vee \text{true}))$$

Kromosom: Program



```
k = 0;  
while k < 10  
{  
    k = k + 1  
}
```

Mapping

- Bagaimana merepresentasikan suatu program ke dalam kromosom dan sebaliknya?
- Michael Lynn Cramer (1985) mengusulkan cara pembangunan kromosom untuk meng-evolusi program dalam bahasa pemrograman PL
- Idenya: representasi integer seperti pada GA
- Terdapat dua pendekatan pemetaan (*mapping*) yang diusulkan:
 - JB *mapping*
 - TB *mapping*

JB Mapping

Operasi bahasa pemrograman PL dan kode *integer*-nya

Kode integer	Operasi
0	BLOCK
1	LOOP
2	SET
3	ZERO
4	INC

JB Mapping

Genotype: representasi integer

0	0	1	3	5	8	1	3	2	1	4	3	4	5	9
Triplet 1			Triplet 2			Triplet 3			Triplet 4			Triplet 5		



(0 0 1) ;;main statement ? (: BLOCK AS0 AS1)
(3 5 8) ;;auxiliary statement 0 ? (: ZERO V5)
(1 3 2) ;;auxiliary statement 1 ? (: LOOP V3 AS2)
(1 4 3) ;;auxiliary statement 2 ? (: LOOP V4 AS3)
(4 5 9) ;;auxiliary statement 3 ? (: INC V5)

Phenotype: program PL

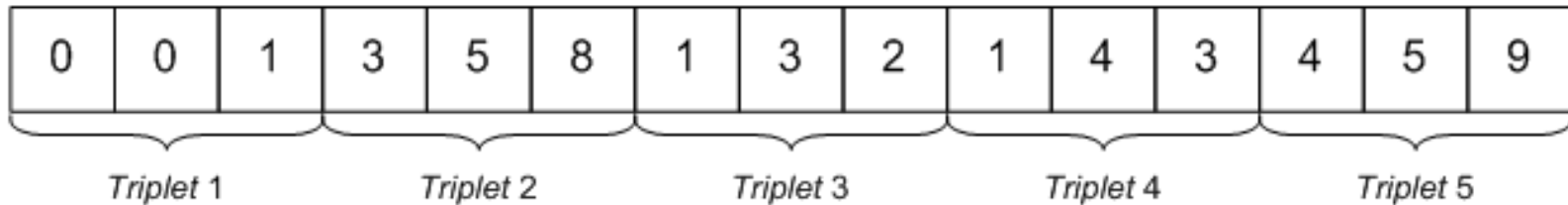
```
;; Program perkalian: Kalikan V3 dengan V4 dan simpan hasilnya di V5  
(: ZERO V5)  
(: LOOP V3 (: LOOP V4 (: INC V5)))
```

TB *mapping*

- Kalau JB *mapping* menggunakan *auxiliary statement* (AS), TB *mapping* bekerja secara rekursif untuk mendekodekan ekspresi
- Hal ini membuat kromosom yang berbasis TB *mapping* menjadi lebih singkat

JB Mapping

Genotype: representasi integer



TB Mapping

(0 (3 5) (1 3 (1 4 (4 5))))

```
;; Program perkalian: Kalikan V3 dengan V4 dan simpan hasilnya di V5  
(: ZERO V5)  
(: LOOP V3 (: LOOP V4 (: INC V5)))
```

GP untuk bahasa LISP

- Bagaimana penggunaan GP untuk bahasa pemrograman LISP (**LIS**t **P**rogramming), yaitu suatu bahasa pemrograman yang berbasis *list* (senarai)
- Pada masa-masa awal kemunculannya, GP banyak difokuskan pada bahasa LISP
- Pada LISP, suatu ekspresi program dituliskan secara *prefix*, yaitu menggunakan aturan **operasi operan operan**

GP untuk bahasa LISP

- Sebagai contoh, suatu ekspresi LISP berikut:

$$(+ (* 5 x) (* 7 y))$$

menyatakan ekspresi matematika

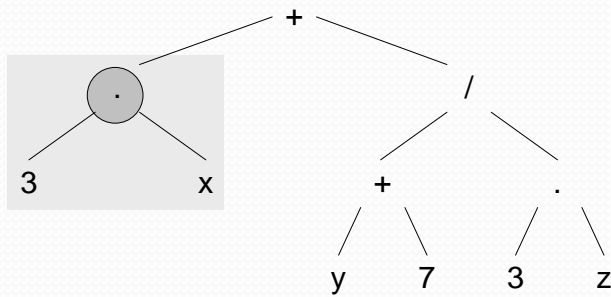
$$5x + 7y$$

- Berapa banyak kode operasi yang harus digunakan?
- Tentu saja bergantung pada masalah yang dihadapi.
- Semakin banyak operasi yang digunakan, maka GP bisa digunakan untuk menyelesaikan lebih banyak masalah tetapi tentu saja komputasinya semakin kompleks

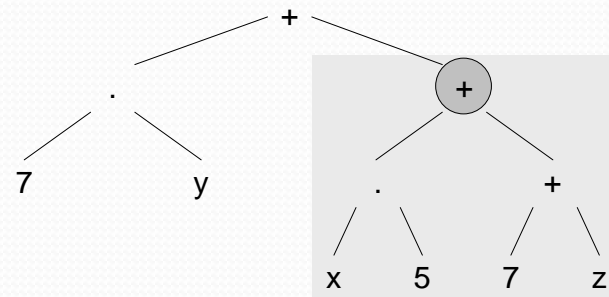
Seleksi Orangtua

- Pada GP, seleksi ortu biasanya dilakukan secara proporsional terhadap nilai *fitness*.
- GP biasanya menggunakan populasi yang berukuran sangat besar.
- Agar lebih efisien, seleksi ortu dilakukan dengan cara:
 - Buat perankingan berdasarkan *fitness* dan bagi individu ke dalam dua grup.
 - Grup 1 berisi $x\%$ individu terbaik dan Grup 2 berisi $(100-x)\%$ individu yang lain.
 - Lakukan 80% seleksi ortu pada Grup 1 dan 20% pada Grup 2.
 - Untuk ukuran populasi = 1000, 2000, 4000, dan 8000 gunakan secara berturut-turut $x = 32\%$, 16% , 8% dan 4% .

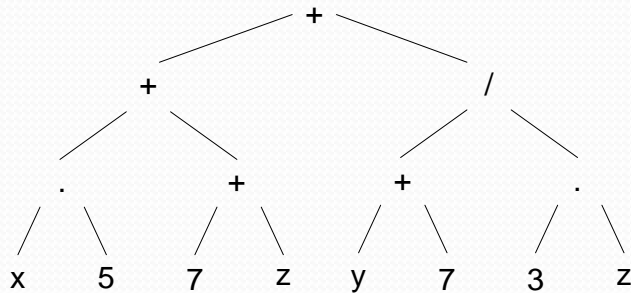
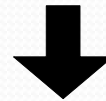
Rekombinasi



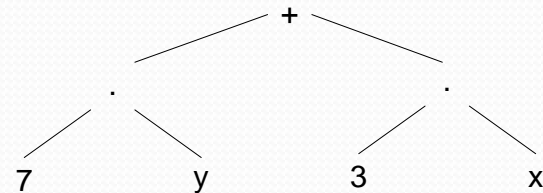
Orangtua 1



Orangtua 2

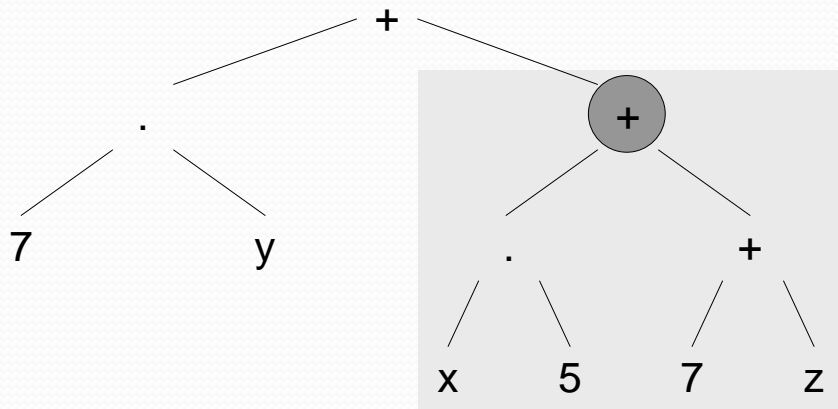


Anak 1

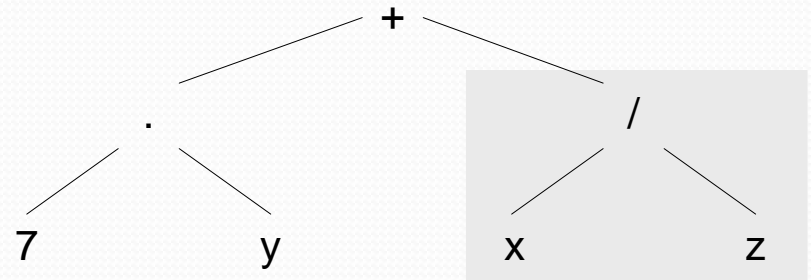


Anak 2

Mutasi



Kromosom awal



Kromosom hasil mutasi

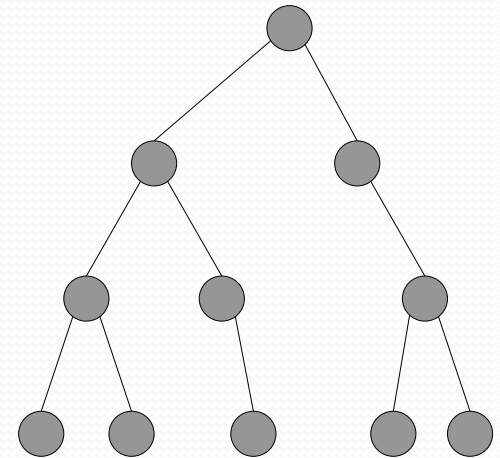
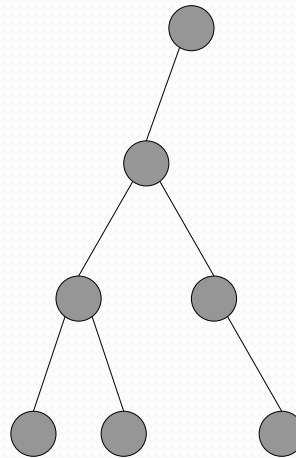
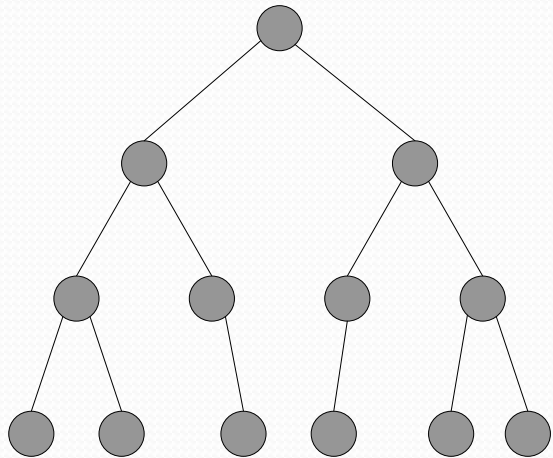
Seleksi *Survivor*

- Umumnya GP menggunakan model populasi *generational scheme* sehingga tidak ada seleksi survivor.
- Tetapi, akhir-akhir ini model populasi *steady-state* lebih populer dan banyak digunakan dibandingkan *generational scheme*.

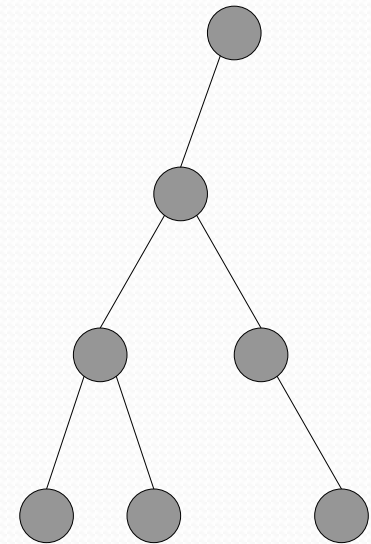
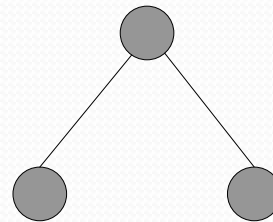
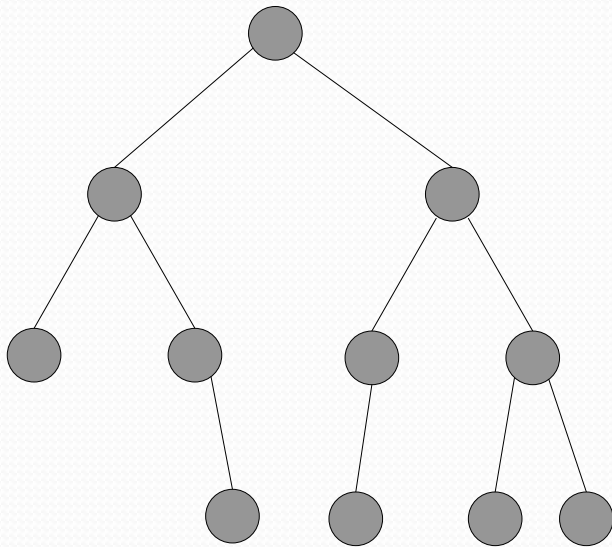
Inisialisasi

- Karena kromosom pada GP bisa memiliki ukuran yang berbeda-beda, maka pada generasi pertama kita bisa melakukan proses inisialisasi dengan membatasi kedalaman pohon.
- Misalkan, kedalaman pohon maksimum adalah D_{\max} . Selanjutnya, dengan menggunakan kedalaman maksimum ini kita bisa membangkitkan kromosom awal dengan cara:
 - *Full method*
 - *Grow method*
 - *Ramped half-and-half*

Inisialisasi: Full method



Inisialisasi: Grow method



Inisialisasi: *Ramped half-and-half*

- Metode ini adalah yang paling umum digunakan pada GP.
- Pada metode ini pembangkitan kromosom dilakukan menggunakan kombinasi *full method* dan *grow method*.
- Caranya bisa bervariasi. Misalnya, 50% kromosom dibangkitkan menggunakan *grow method* dan 50% kromosom lainnya dibangkitkan menggunakan *full method*.
- Cara lain yang bisa digunakan adalah dengan membangkitkan bilangan acak, misalnya r (dimana r dalam interval $[2, D_{\max}]$), sebagai batas kedalaman pada setiap pembangkitan pohon.
- Dengan demikian, cara ini juga merupakan gabungan dari *full method* dan *grow method*.
- Metode *Ramped half-and-half* memang lebih disukai karena proses inisialisasi bisa menghasilkan pohon dengan kedalaman yang bervariasi.

Aplikasi GP

- *symbolic regression*
- *grammar induction*
- *data mining and data analysis*
- *logic function synthesis*
- *circuit design and layout*
- *high-level circuit design*
- *medicine*
- *breeding financial and trading rules*
- *microwave antenna design*
- *finding cellular automata rules*
- *learning of rules for geometric structures*
- *automated programming*

Kesimpulan

- GP digunakan untuk *autamtic programming*
- GP biasanya menggunakan populasi berukuran besar
- Kromosom merepresentasikan tree atau graph
- GP lebih fokus pada bahasa LISP

Daftar Pustaka

- [SUYo8] Suyanto, 2008, Evolutionary Computation: Komputasi Berbasis “Evolusi” dan “Genetika”, penerbit Informatika Bandung.