


Algoritma Evolusi

Genetic Programming (GP) Dan Evolutionary Programming (EP)

Imam Cholissodin | imam.cholissodin@gmail.com



Pokok Bahasan

- 1. Genetic Programming (GP)**
 - 2. Siklus Genetic Programming**
 - 3. Evolutionary Programming (EP)**
 - 4. Studi Kasus 1: Pohon Keputusan**
 - 5. Tugas**
- 

Genetic Programming

- **GP** merupakan satu **metode untuk** secara sistematis **mengevolusi program komputer**.
- Ide dasarnya adalah **bagaimana komputer dapat secara otomatis menghasilkan program komputer berdasarkan** sebuah permasalahan yang diberikan.
- Perbedaan dengan **solusi GAs** yang berupa nilai numeris berdasarkan fungsi obyektif yang digunakan, **GP menghasilkan rangkaian program komputer yang bisa direpresentasikan dalam bentuk struktur pohon (tree)**.
- **GP telah diaplikasikan dalam berbagai bidang**, misalnya untuk mendesain *analog circuit* pada skala industri secara otomatis ([Koza et al. 2004](#)), mendesain *quantum computing circuits* ([Spector 2004](#)), dan membantu manajer untuk mendesain organisasi yang optimal pada sebuah proyek ([Khosraviani, Levitt & Koza 2004](#)).

Genetic Programming

- **GP** juga cukup **efektif dikombinasikan/dihibridisasi** dengan algoritma lain seperti ***ant colony optimization*** untuk memperbaiki desain geometris pada reflector *light-emitting diode* (LED). Perbaikan ini akan berpengaruh secara signifikan terhadap intensitas cahaya yang dipancarkan oleh LED ([Hsu 2011](#)).
- Pada bidang *data mining*, **hybrid GP** dan ***simulated annealing*** telah sukses diaplikasikan untuk membentuk pohon keputusan pada klasifikasi data ([Folino, Pizzuti & Spezzano 2000](#)).
- Contoh permasalahan sederhana berdasarkan tabel berikut :

x_1	x_2	y
1	2	1
2	3	12
3	2	5
4	5	46
5	3	18
6	5	56
7	4	35
8	6	84
9	5	56
10	8	148

Buatlah sebuah fungsi dengan variabel bebas x_1 dan x_2 yang menghasilkan variabel tak bebas y !

Note : Nilai-nilai pada tabel di atas sebenarnya dihasilkan dari sebuah fungsi non-linear dengan persamaan berikut:

$$y = f(x_1, x_2) = (3x_1x_2 - x_1^2) + (2x_2 - 8)$$

Genetic Programming

- Contoh permasalahan sederhana berdasarkan tabel berikut :

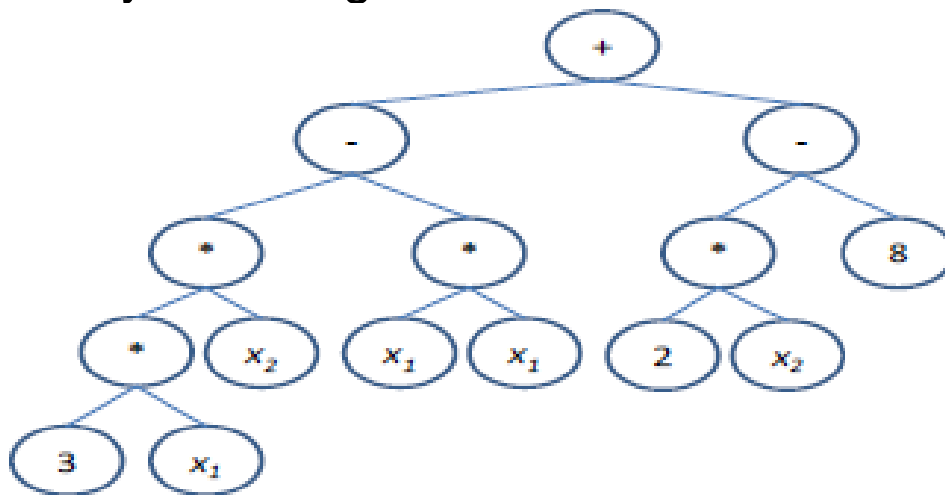
x_1	x_2	y
1	2	1
2	3	12
3	2	5
4	5	46
5	3	18
6	5	56
7	4	35
8	6	84
9	5	56
10	8	148

Buatlah sebuah fungsi dengan variabel bebas x_1 dan x_2 yang menghasilkan variabel tak bebas y !

Note : Nilai-nilai pada tabel di atas sebenarnya dihasilkan dari sebuah fungsi non-linear dengan persamaan berikut:

$$y = f(x_1, x_2) = (3x_1x_2 - x_1^2) + (2x_2 - 8)$$

- Binary tree fungsi non-linear di atas :

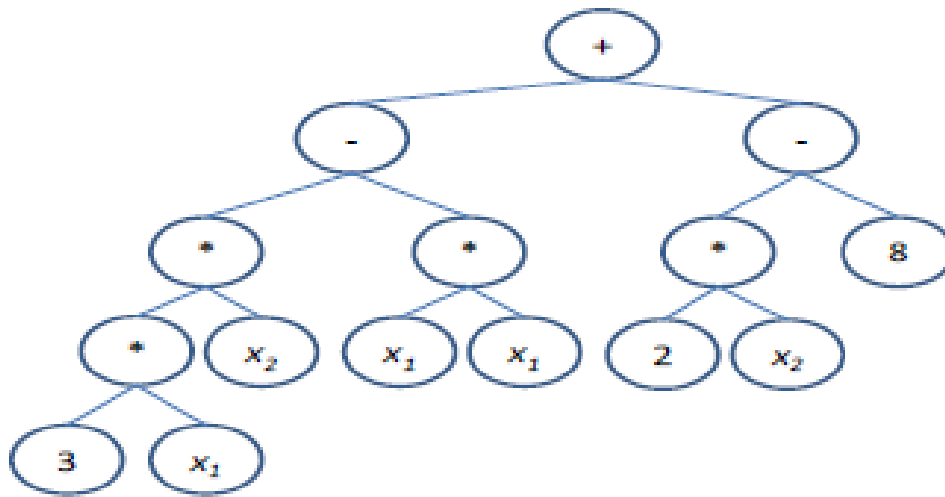


Node menunjukkan instruksi yang harus dilaksanakan (misal * dan +).

Link merujuk ke argumen dari tiap instruksi (misal x_1 dan x_2).

Genetic Programming

- *Binary tree* fungsi non-linear $y = f(x_1, x_2) = (3x_1x_2 - x_1^2) + (2x_2 - 8)$:



Node menunjukkan instruksi yang harus dilaksanakan (misal * dan +).

Link merujuk ke argumen dari tiap instruksi (misal x_1 dan x_2).

Arithmetic Exp. : $(3x_1x_2 - x_1^2) + (2x_2 - 8)$

Preorder : $+-**3x_1x_2*x_1x_1-2x_28$

Inorder : $3*x_1*x_2-x_1*x_1+2*x_2-8$

Postorder : $3x_1*x_2*x_1x_1*-2x_2*8-+$

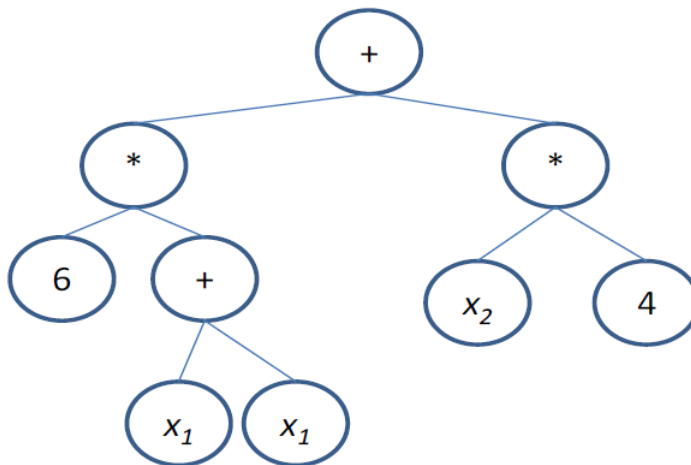
- **Masalahnya**, bagaimana **merancang GP** untuk menghasilkan *binary tree* di atas (yang fungsinya belum diketahui sebelumnya).
- Secara umum, siklus GP mirip dengan GAs. Dua hal yang harus ditambahkan pada **representasi solusi GP** adalah:
 - Penentuan himpunan terminal (*set of terminals*)
 - Penentuan himpunan fungsi (*set of functions*)

Siklus Genetic Programming

- Untuk kasus penentuan fungsi non-liner bisa didefinisikan himpunan terminal dan fungsi sebagai berikut:
 - Himpunan terminal : $T = \{ x_1, x_2, x \in R \}$, dimana R adalah real_number
 - Himpunan fungsi : $F = \{ +, -, *, / \}$

1. Representasi Chromosome

- Setiap chromosome berupa sebuah *binary tree*. Bagaimana sebuah *binary tree* disimpan/ditangani oleh program komputer, silahkan membuka ulang materi mata kuliah struktur data.



Arithmetic Exp. : $(6 * (x_1 + x_1)) + (x_2 * 4)$

Preorder : $+ * 6 + x_1 x_1 * x_2 4$

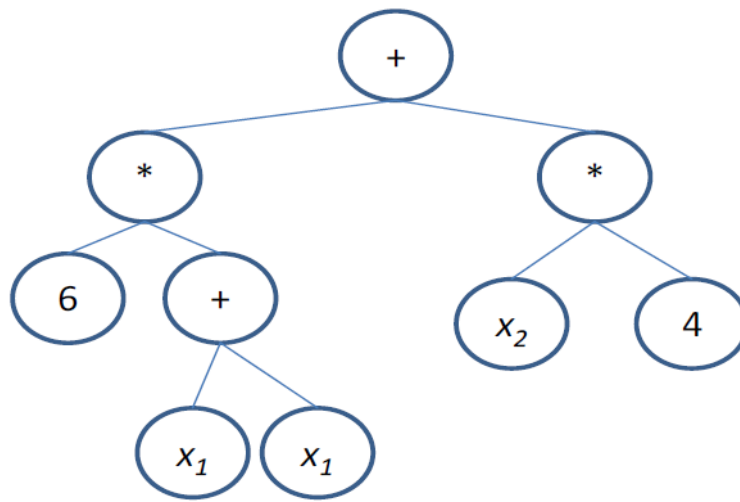
Inorder : $6 * x_1 + x_1 + x_2 * 4$

Postorder : $6 x_1 x_1 + * x_2 4 * +$

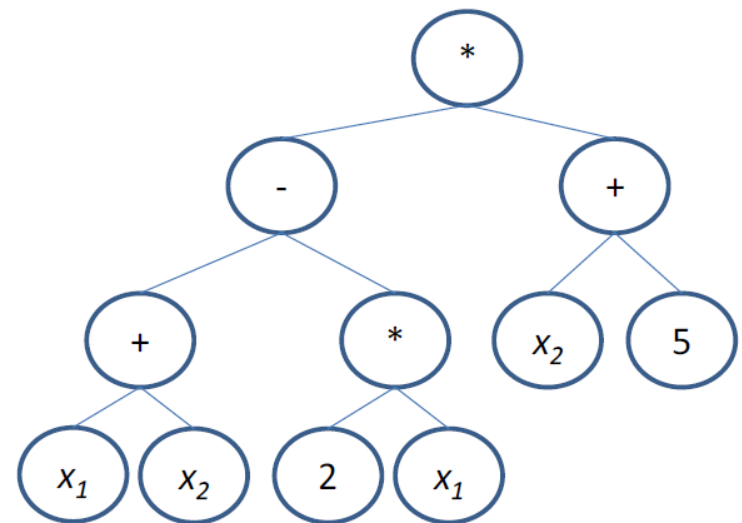
Siklus Genetic Programming

2. Inisialisasi dan Evaluasi

- Chromosome dari dua individu (P_1 dan P_2) yang dibangkitkan secara acak diberikan sebagai berikut:



individu 1 (P_1)



individu 2 (P_2)

Arithmetic Exp. : $(6 * (x_1 + x_1)) + (x_2 * 4)$
Preorder : $+ * 6 + x_1 x_1 * x_2 4$
Inorder : $6 * x_1 + x_1 + x_2 * 4$
Postorder : $6 x_1 x_1 + * x_2 4 * +$

Arithmetic Exp. : $((x_1 + x_2) - (2 * x_1)) * (x_2 + 5)$
Preorder : $* - + x_1 x_2 * 2 x_1 + x_2 5$
Inorder : $x_1 + x_2 - 2 x_1 * x_2 + 5$
Postorder : $x_1 x_2 + 2 x_1 * - x_2 5 + *$

Siklus Genetic Programming

2. Inisialisasi dan Evaluasi

- Menghitung nilai *fitness* dari individu (P_1 dan P_2) :

x_1	x_2	y	P_1 $yp_1 = 6(x_1+x_1)+4x_2$		P_2 $yp_2 = ((x_1+x_2)-2x_1)(x_2+5)$	
			yp_1	$ y-yp_1 $	yp_2	$ y-yp_2 $
1	2	1	20	19	7	6
2	3	12	36	24	8	4
3	2	5	44	39	-7	12
4	5	46	68	22	10	36
5	3	18	72	54	-16	34
6	5	56	92	36	-10	66
7	4	35	100	65	-27	62
8	6	84	120	36	-22	106
9	5	56	128	72	-40	96
10	8	148	152	4	-26	174
Total			371		596	
Fitness			$1/(1+371)=0,002688$		$1/(1+596)=0,001675$	

Arithmetic Exp. : $(6*(x_1+x_1))+(x_2*4)$

Preorder : $+*6+x_1x_1*x_24$

Inorder : $6*x_1+x_1+x_2*4$

Postorder : $6x_1x_1+*x_24*+$

Arithmetic Exp. : $((x_1+x_2)-(2*x_1))*(x_2+5)$

Preorder : $*-+x_1x_2*2x_1+x_25$

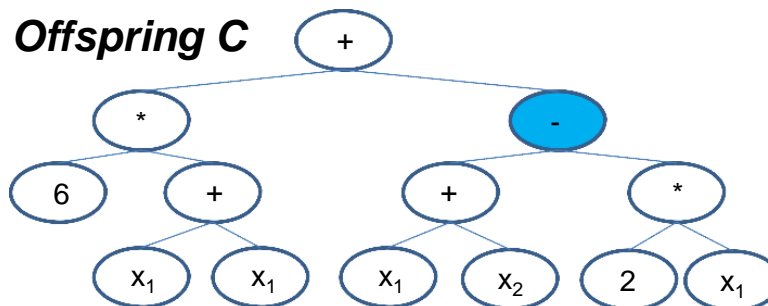
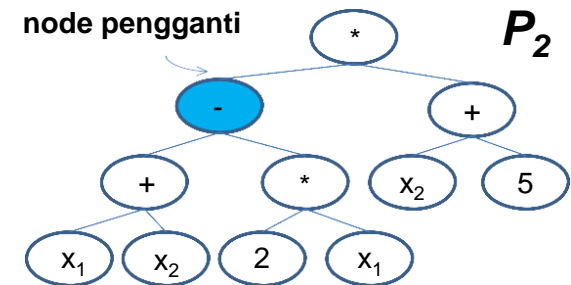
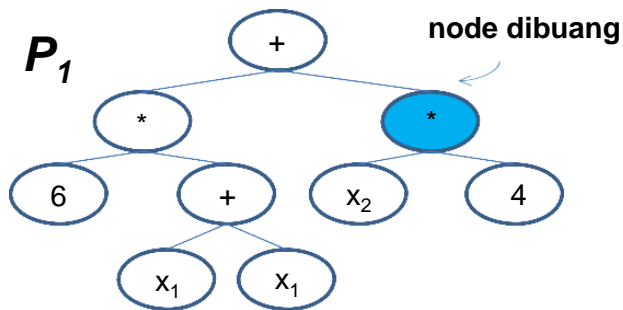
Inorder : $x_1+x_2-2x_1*x_2+5$

Postorder : $x_1x_2+2x_1*-x_25*+$

Siklus Genetic Programming

3. Crossover

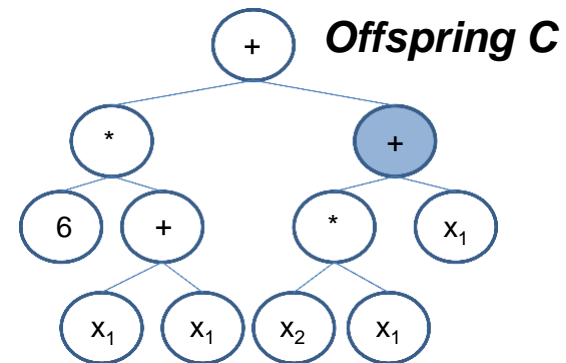
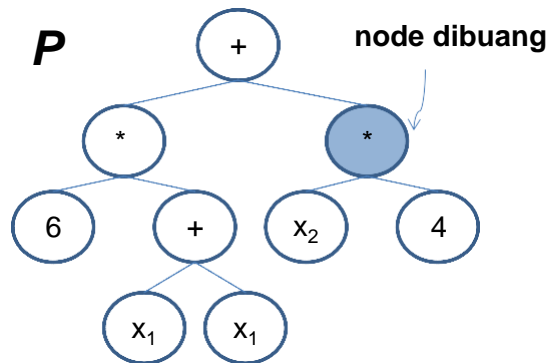
- Crossover antara dua induk P_1 dan P_2 dilakukan dengan memilih satu node secara acak dari P_1 .
- *Sub-tree* yang dimulai dari node ini dihapus dan digantikan dengan *sub-tree* yang dipilih secara acak dari P_2 .



Siklus Genetic Programming

4. Mutasi

- Mutasi terhadap induk P dilakukan dengan memilih satu node secara acak. *Sub-tree* yang dimulai dari node ini digantikan dengan *sub-tree* yang dibangkitkan secara acak seperti pada proses inisialisasi.



5. Seleksi

- Metode seleksi pada GP sama dengan yang digunakan pada GAs.

Evolutionary Programming (EP)

- **EP** juga bisa untuk menghasilkan rangkaian program komputer tapi **prinsip kerjanya seperti ES**. Finite State Machines (FSM) biasanya digunakan untuk merepresentasikan program komputer.
- **Contoh EP** untuk mencari **komposisi campuran minuman terbaik** berdasarkan penilaian *tester*. Bahan campuran : *sugar water, ginger flavoring, strawberry nectar, salt water, pineapple juice, tea, raspberry juice, peach nectar, milk, grapefruit juice, cranberry juice, coffee, apple juice, grape juice*, dan *chocolate*.
- **Banyak percobaan membuktikan kekuatan EP** dalam pencarian solusi (Bell & Alexande 207). Pada bidang kesehatan, EP digunakan untuk melatih ANN untuk mendeteksi penyakit kanker (Fogel et al. 1998).
- EP untuk penjadwalan, misal pada *short-term hydrothermal scheduling* (Hota, et. al. 1999) dan penjadwalan proyek (Sebt & Alipouri 2013), serta **efektif untuk dihibridisasi** dengan algoritma lain seperti PSO untuk penjadwalan perbaikan (*maintenance*) pada pembangkit tenaga listrik (Samuel & Rajan 2013).

Studi Kasus 1: Pohon Keputusan

- Perhatikan data historis keputusan kredit usaha yang diajukan oleh perusahaan pada sebuah bank sebagai berikut:

No	Aset (A)	Kinerja (Ki)	Jaminan (J)	Keputusan (Ke)
1	1.5	6	0	0
2	0.7	6	1	0
3	2	7	0	1
4	1.6	5	1	1
5	0.9	8	0	1
6	0.8	8	1	1

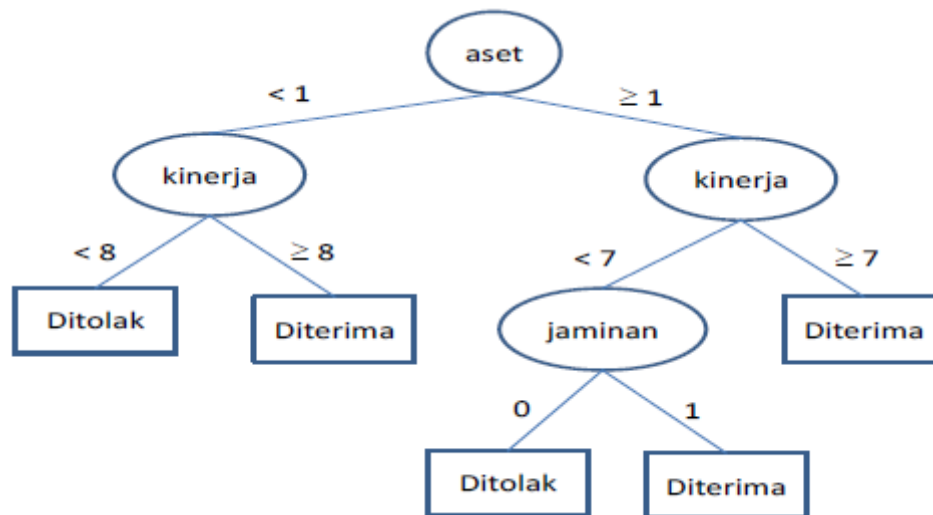
- Data historis ini berdasarkan penilaian ahli perkreditan bank tersebut. Keputusan untuk menerima (1) atau menolak (0) pengajuan kredit didasarkan atas tiga variabel yaitu: nilai aset (A) yang dimiliki perusahaan, nilai kinerja (Ki) perusahaan setahun terakhir, dan ketersediaan jaminan (J).
- Bank memutuskan untuk membangun sistem yang bisa secara otomatis memberikan keputusan penerimaan. Sistem ini bisa memberikan *output* Ke jika misalkan diberikan *input* A=1.2, Ki=5, dan J=6.

Studi Kasus 1: Pohon Keputusan

- Perhatikan data historis keputusan kredit usaha :

No	Aset (A)	Kinerja (K _i)	Jaminan (J)	Keputusan (K _e)
1	1.5	6	0	0
2	0.7	6	1	0
3	2	7	0	1
4	1.6	5	1	1
5	0.9	8	0	1
6	0.8	8	1	1

- Solusi GP berupa pohon keputusan (*decision tree*).



Gambar tersebut menunjukkan alur penerimaan/penolakan pengajuan kredit berdasarkan input A, K_i, dan J. Pengecekan dimulai dari node paling atas (*root*).

Masalahnya adalah bagaimana menghasilkan pohon keputusan seperti ini berdasarkan data historis keputusan kredit usaha.

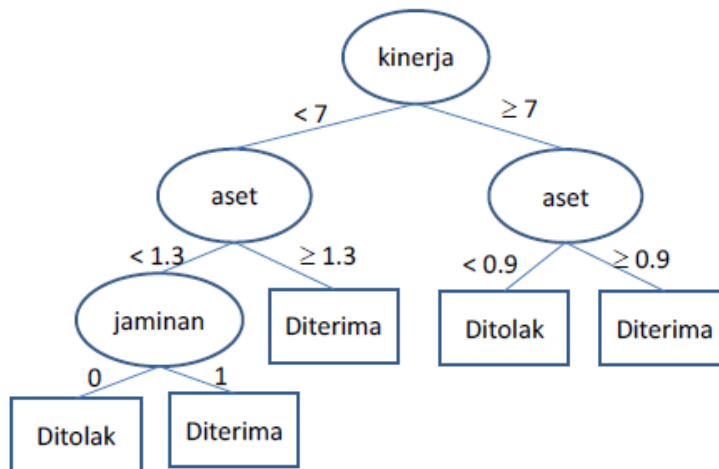
Studi Kasus 1: Pohon Keputusan

1. Representasi Chromosome

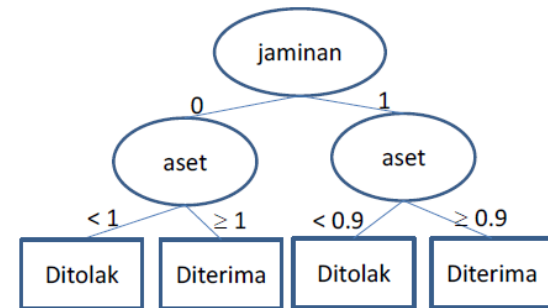
- Karena solusi yang diinginkan berupa pohon keputusan maka setiap chromosome berupa sebuah *binary tree*.

2. Inisialisasi dan Evaluasi

- Chromosome dari dua individu yang dibangkitkan secara acak diberikan sebagai berikut:



individu 1 (P₁)



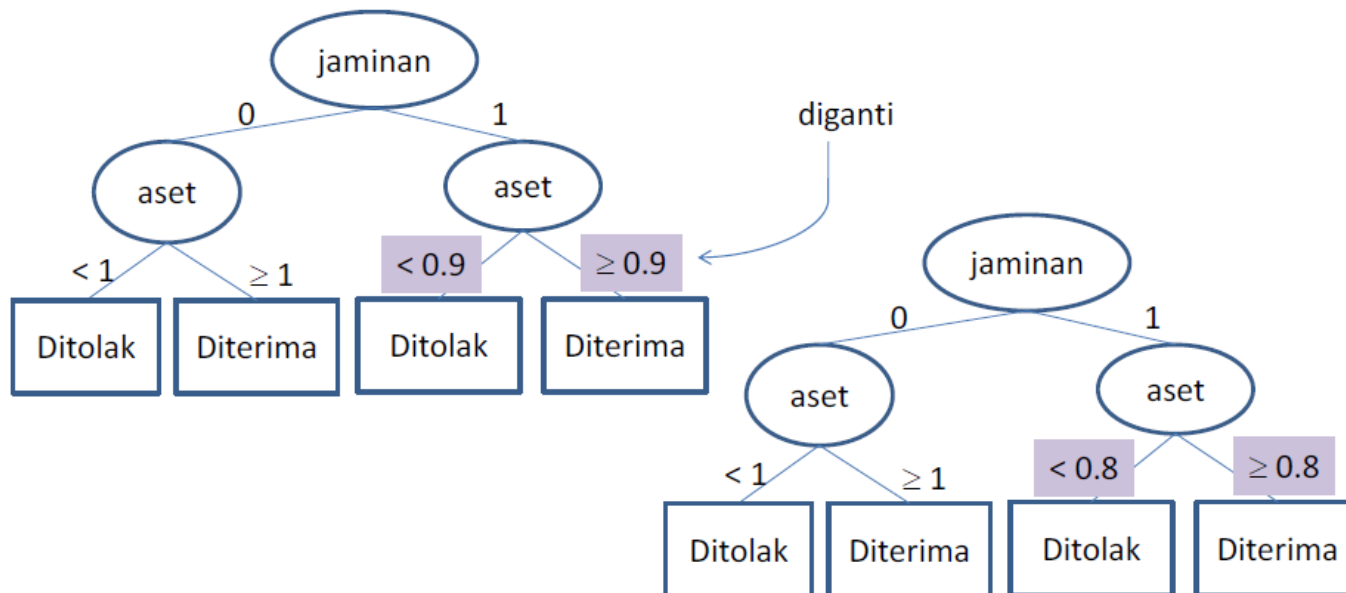
individu 2 (P₂)

No	Aset (A)	Kinerja (Ki)	Jaminan (J)	Keputusan (Ke)	P ₁		P ₂	
					Keputusan Prediksi	Skor	Keputusan Prediksi	Skor
1	1.5	6	0	0	1	0	1	0
2	0.7	6	1	0	1	0	0	1
3	2	7	0	1	1	1	1	1
4	1.6	5	1	1	1	1	1	1
5	0.9	8	0	1	1	1	0	0
6	0.8	8	1	1	0	0	0	0
<i>Fitness</i>						3		3

Studi Kasus 1: Pohon Keputusan

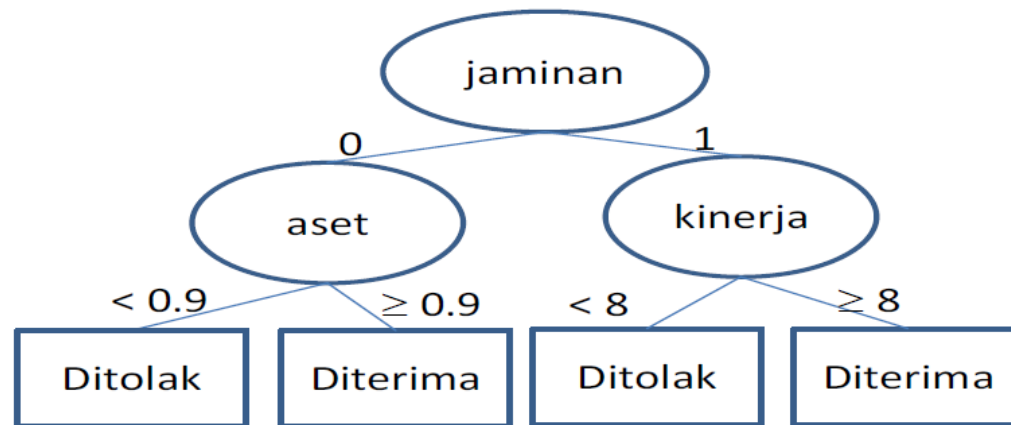
3. Crossover, Mutasi, dan Seleksi

- Operator reproduksi dan seleksi sama seperti yang digunakan pada slide-slide sebelumnya.
- Khusus untuk mutasi, karena pada *link* terdapat angka yang menunjukkan batasan sebuah variabel maka harus ditambahkan mekanisme mutasi yang digunakan hanya untuk mengubah angka ini seperti berikut:



Tugas Kelompok

1. Jelaskan apakah output dari GP?
2. Apakah komponen utama pada representasi solusi dari GP?
3. Untuk kasus pada tabel dari slide ke-14, hitunglah fitness untuk chromosome P berikut:

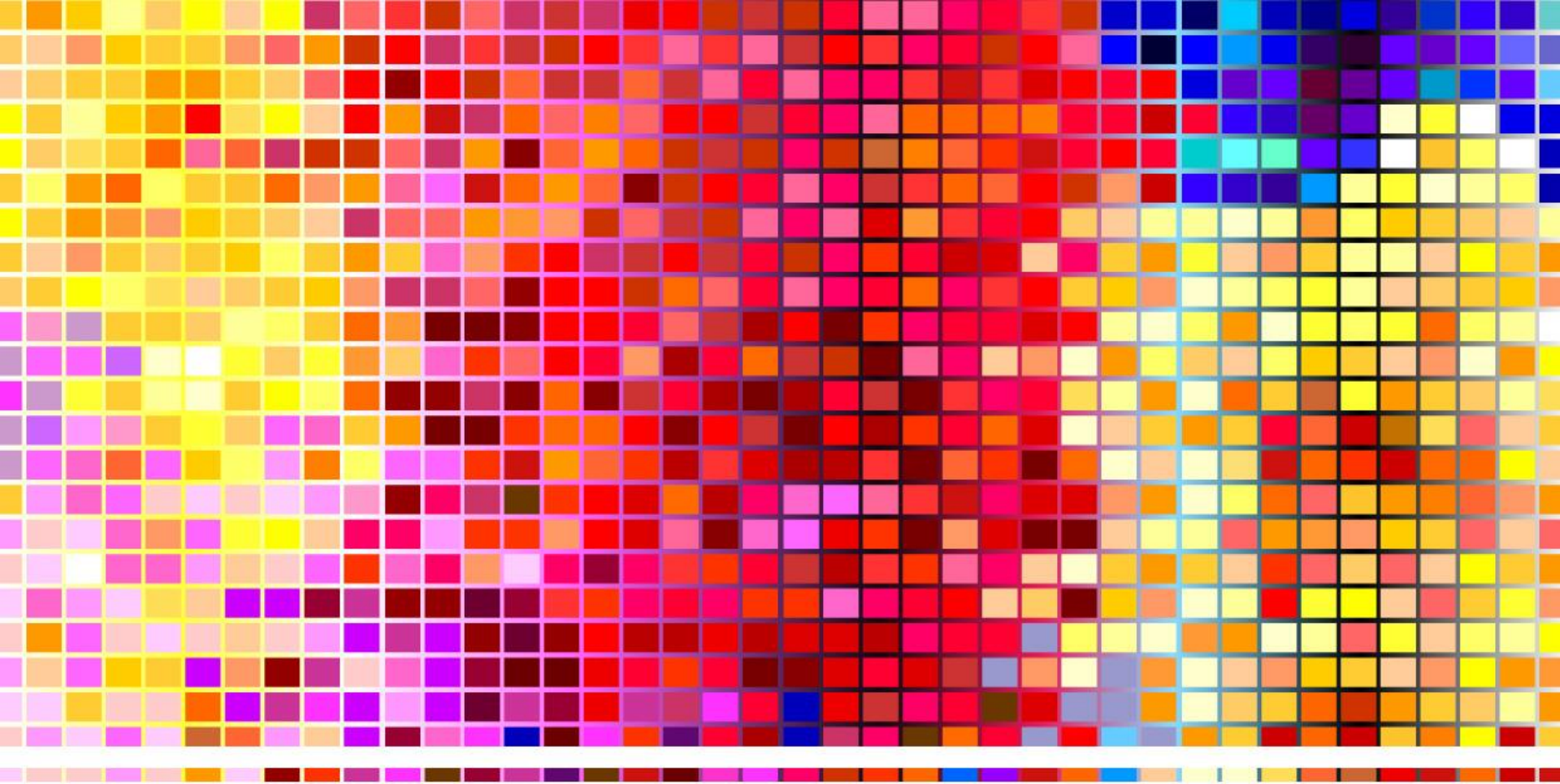


4. Selesaikan studi kasus pada Slide ke-7. Siklus Genetic Programming sampai 2 generasi dengan menggunakan GP!
5. Selesaikan studi kasus pada tabel dari slide ke-14. Studi Kasus: Pohon Keputusan sampai 2 generasi dengan menggunakan GP!

Tugas Kelompok

6. Dengan menggunakan EP, susun *binary tree* untuk data pada tabel berikut!

x_1	x_2	y
3	4	18
4	2	24
6	3	51
6	4	48
7	1	87
8	2	104
9	3	123
10	4	144
10	6	128
11	9	143



Terimakasih

Imam Cholissodin | imam.cholissodin@gmail.com

