

Cara untuk mendapatkan salah satu nilai desimal dari contoh kromosom-kromosom di atas adalah sebagai berikut:

$$\begin{aligned}\text{Nilai desimal (110011)} &= (1 \times 2^5) + (1 \times 2^4) + (0 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) \\ &= 51\end{aligned}$$

Untuk memulai proses algoritma genetika, sejumlah kromosom yang berperan sebagai populasi awal dibangkitkan **secara acak**. Misalkan populasi ini terdiri atas empat buah kromosom, yaitu sebagai berikut.

No.	Kromosom pada populasi awal	Nilai desimal
1.	110011	51
2.	010111	23
3.	101100	44
4.	011011	27

Nilai *fitness* keempat kromosom di atas dapat diketahui dengan mensubstitusikan nilai desimal masing-masing kromosom ke dalam fungsi *fitness*. Hasil penghitungan ini dapat dilihat pada Tabel 2.1.

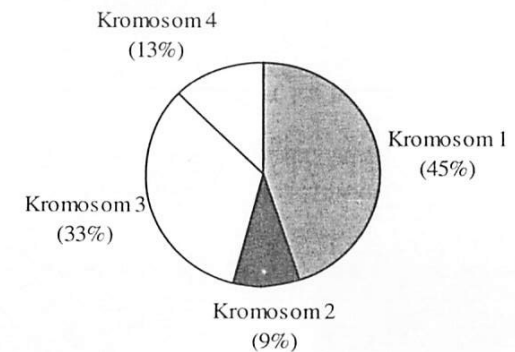
Tabel 2.1 Hasil penghitungan untuk contoh Algoritma Genetika Sederhana

No.	Kromosom pada populasi awal	Nilai desimal (x)	Nilai fitness $f(x)=x^2$
1.	110011	51	2601
2.	010111	23	529
3.	101100	44	1936
4.	011011	27	729
Total			5795
Rata-rata			1448,75
Maksimum			2601

Seleksi

Sebelum reproduksi dimulai, dilakukan seleksi terhadap kromosom yang ada untuk dijadikan sebagai induk. Secara alami, kromosom dengan nilai *fitness* yang lebih tinggi akan memiliki peluang lebih besar terpilih. Besarnya peluang masing-masing kromosom dalam contoh ini dapat dilihat pada Gambar 2.4.

Salah satu teknik seleksi adalah *Roulette Wheel Selection*, seperti yang telah dijelaskan pada subbab terdahulu. Jumlah populasi yang digunakan adalah 4 kromosom. Dengan demikian, agar jumlah populasinya konstan, roda rolet diputar sebanyak 4 kali. Setiap pemutaran roda rolet akan dihasilkan satu buah kromosom induk. Setelah roda rolet diputar empat kali, misalkan secara berurutan diperoleh kromosom 1, kromosom 3, kromosom 2, dan kromosom 4. Kromosom-kromosom inilah yang akan menjadi kromosom induk, untuk kemudian akan mengalami penyilangan dan atau juga mutasi.



Gambar 2.4 Proporsi peluang terpilihnya kromosom dalam tahap *Roulette Wheel Selection*

Penyilangan (*Crossover*)

Ilustrasi proses penyilangan kromosom 1 (*Parent 1*) dan kromosom 4 (*Parent 4*) dengan titik potong pada posisi 2 yang ditentukan secara *random* (acak) adalah sebagai berikut:

Parent 1	1 1 0 0 1 1	→ 51
Parent 4	0 1 1 0 1 1	→ 27
Child 1	0 1 0 0 1 1	→ 59
Child 2	1 1 1 0 1 1	→ 19

Ilustrasi proses penyilangan kromosom 2 (*Parent 2*) dan kromosom 3 (*Parent 3*) dengan titik potong pada posisi 3 yang ditentukan secara *random* (acak) adalah sebagai berikut:

Parent 2	0 1 0 1 1 1	→ 23
Parent 3	1 0 1 1 0 0	→ 44
Child 3	1 0 1 1 1 1	→ 47
Child 4	0 1 0 1 0 0	→ 20

Berdasarkan penjelasan di atas, dapat dilihat bahwa proses penyilangan akan menghasilkan keturunan yang lebih *fit* (yang memiliki nilai *fitness* lebih tinggi). Populasi yang dihasilkan disajikan pada Tabel 2.2.

Tabel 2.2 Populasi baru pada contoh Algoritma Genetika Sederhana

	Kromosom anak	Nilai desimal (x)	Nilai <i>fitness</i> $f(x)=x^2$
1	111011	59	3481
2	010011	19	361
3	101111	47	2209
4	010100	20	400
Total			6451
Rata-rata			1612,75
Maksimum			3481

Dari Tabel 2.2 dapat dilihat bahwa nilai total dan nilai *fitness* rata-rata dari populasi baru lebih tinggi dari populasi awal. Hal ini menandakan bahwa proses pencarian menggunakan algoritma genetika mampu mendapatkan nilai x yang lebih baik dari yang sebelumnya. Di mana kromosom dengan nilai x yang baik merupakan calon optimal permasalahan yang dihadapi. Untuk mempertahankan kromosom yang memiliki nilai yang optimal maka diperlukan proses elitisme serta penambahan generasi sehingga menghasilkan solusi optimal seperti pada Gambar 2.5.

Dari Gambar 2.5 di atas dapat kita lihat bahwa nilai *fitness* dari generasi ke generasi kian membaik (meningkat). Peningkatan ini dimulai pada generasi ke-30 hingga generasi ke-40. Setelah itu, nilai *fitness* akan konvergen hingga generasi ke-300. Saat konvergen inilah, dicapai nilai x optimal (x^*) yaitu 63. Struktur kromosom untuk x ini adalah 111111 dengan nilai *fitness* 3969. Nilai ini adalah nilai *fitness* terbaik (paling optimum) yang pernah ditemukan oleh algoritma genetika.