

## **BAB 2**

### **LANDASAN TEORI**

Telaah literatur yang akan dibahas dalam perancangan dan implementasi Penjadwalan Shift Kerja Karyawan Operasional Kompas Gramedia dengan Metode Algoritma Genetika adalah sebagai berikut:

#### **2.1 Modul HR Shift Scheduling**

Pada dasarnya shift kerja merupakan sebuah strategi yang dilakukan untuk meningkatkan jam operasional perusahaan secara maksimal dan efisien. Bagi seorang pegawai, *shift* kerja berarti berada pada lokasi kerja yang sama, baik teratur pada jam yang sama (*shift* kontinyu) atau jam kerja yang berlainan (*shift* kerja rotasi). *Shift* kerja itu sendiri berbeda dengan hari kerja biasa dimana hari kerja biasa berlangsung antara jam 8 pagi hingga 5 sore sedangkan shift kerja dapat dilakukan lebih dari satu kali untuk memenuhi jadwal operasional sebuah perusahaan [5].

Modul HR *Shift Scheduling* sendiri berhubungan dengan pergantian shift kerja. Karyawan yang umumnya membutuhkan jadwal shift antara lain seperti *Front Office*, *Security*, *Office Boy*, dan karyawan percetakan. Untuk mengganti shift kerja, *HR Admin* hanya perlu mengganti kode *shift*, secara otomatis *shift* kerja baru akan ditampilkan. Shift yang baru akan tersimpan di database dan ditampilkan di kalender pengingat.

#### **2.2 Penjadwalan**

Penjadwalan berfungsi sebagai pengambilan keputusan berkaitan dengan pendefinisian proses terjadwal, dan penjadwalan sebagai teori dengan prinsip yang dapat dibuktikan, model teknis dan logika kesimpulan, secara jelas menunjukkan kedalaman fungsionalitas penjadwalan, dan prosedur administrasi di kantor pusat. Jika kapasitas melebihi batas daya yang tersambung, atau ada banyak tugas yang tertunda, tugas yang harus diprioritaskan harus ditentukan. Perencanaan waktu kerja dilakukan secara sistematis agar rantai proses dapat berjalan lancar dengan mengoptimalkan fasilitas yang tersedia [6].

### 2.3 Algoritma Genetika

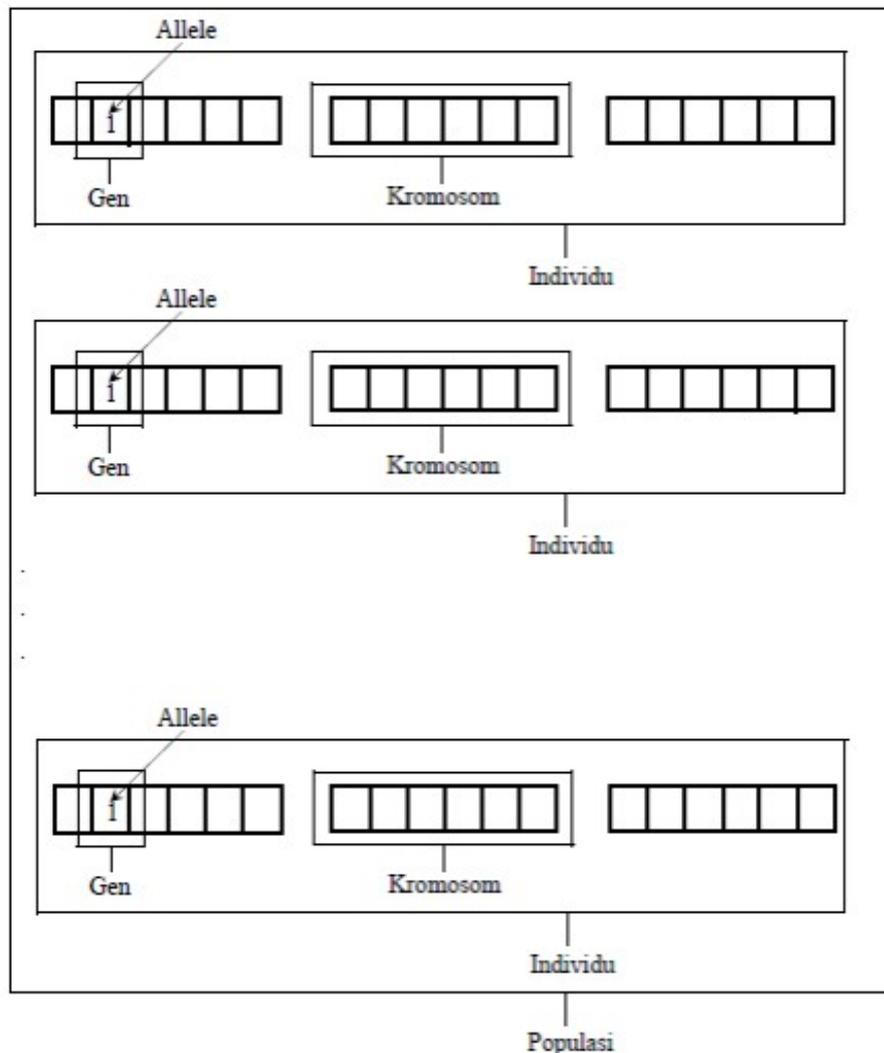
Algoritma Genetika adalah suatu algoritma pencarian yang meniru mekanisme dari genetika alam. Algoritma genetika pertama kali dikemukakan oleh John Holland awal tahun 1975. [7] Algoritma genetika banyak digunakan untuk pemecahan masalah dan pemodelan di bidang teknologi seperti pengoptimalan, pemrograman otomatis, dan pembelajaran mesin. Tidak seperti teknik pencarian tradisional, tahap awal pencarian algoritma genetik dimulai dengan serangkaian solusi acak yang disebut populasi (random).

Menurut Stuart (2010) terdapat beberapa hal penting yang digunakan untuk membangun penyelesaian permasalahan dengan algoritma Genetika, yaitu [8]:

1. Genotype (Gen), sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom. Dalam algoritma Genetika, gen ini bisa berupa nilai biner, float, integer maupun karakter atau kombinatorial.
2. Allele, nilai dari gen.
3. Kromosom, gabungan gen-gen yang membentuk nilai tertentu.
4. Individu, menyatakan satu nilai keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat
5. Populasi, merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi
6. Generasi, menyatakan satu siklus proses evolusi atau satu iterasi di dalam algoritma Genetika

Visualiasi dari Gen, Allele, Individu, dan Populasi ditunjukkan pada Gambar 2.1 [9].

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 2.1. Visualisasi Gen, Allele, Individu, dan Populasi

Berikut merupakan penjelasan mengenai struktur Dasar Algoritma Genetika:

- **Populasi Awal**  
Populasi awal ini dibangkitkan secara acak untuk mendapatkan solusi aslinya. Populasi itu sendiri terdiri dari beberapa kromosom yang mewakili solusi yang diinginkan [10].
- **Membentuk Generasi Baru**  
Untuk membentuk generasi baru digunakan operator reproduksi/seleksi, crossover dan mutasi. Proses ini dilakukan secara iteratif agar jumlah kromosom yang ada cukup untuk membentuk generasi baru sehingga

generasi baru ini merepresentasikan solusi baru. Generasi baru ini disebut *Offsprings* [10].

- Evaluasi Solusi

Dengan setiap generasi, kromosom melalui proses evaluasi menggunakan alat pengukuran yang disebut *fitness*. Nilai *fitness* suatu kromosom menggambarkan kualitas kromosom pada populasi tersebut. Proses ini akan mengevaluasi setiap populasi dengan menghitung nilai *fitness* dari setiap kromosom dan mengevaluasinya hingga kriteria terpenuhi [10].

Algoritma Genetika mempunyai beberapa proses yaitu Inisiasi Populasi, Fungsi Fitness, Selection, Crossover, dan Mutasi [11].

- Inisiasi Populasi

Algoritma Genetika akan menghasilkan populasi awal menggunakan teknik pengkodean kromosom. Kromosom dapat dianggap sebagai kumpulan individu yang ada di alam dan siap untuk seleksi. Panjang kromosom tergantung pada panjang urutan gen [11].

- Fungsi Fitness

Fungsi Fitness digunakan sebagai alat untuk melihat hasil dari suatu solusi baik atau buruknya suatu masalah. Dalam hal keteraturan, kualitas sebuah kromosom diukur dengan jumlah batasan yang dipatuhi atau jumlah batasan yang dilanggar. Validitas setiap kromosom akan diperiksa dengan menguji Hard dan Soft constraint. Pada hard Constraint semua kasus tidak boleh ada pelanggaran, dan pada Soft Constraint pelanggaran bisa saja terjadi tetapi pelanggarannya sekecil mungkin. Setiap soft constraint yang dilanggar akan dievaluasi sebagai penalti [11]. Rumus dari fungsi fitness adalah sebagai berikut berikut.

$$f(x) = \frac{1}{(1.0 * penalty + 1)}$$

- *Tournament Selection*

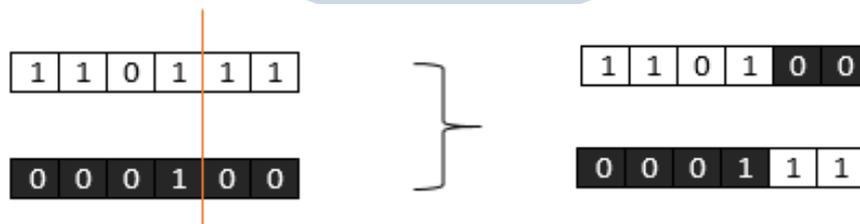
Seleksi turnamen adalah variasi antara seleksi rank dan seleksi roda *roulette* [12]. Pemilihan peringkat mengurutkan populasi berdasarkan nilai *fitness* dan memberinya skor. Nilai *fitness* tertinggi akan berada di urutan pertama dan nilai *fitness* terendah akan berada di urutan terakhir [11]. Gambar 2.2 menunjukkan visualisasi dari *Tournament Selection* [13].



Gambar 2.2. Visualisasi Tournament Selection

- *Crossover*

Crossover adalah perkawinan individu yang dipilih pada saat pemilihan peringkat. Ada beberapa Crossover yang populer yaitu One-Point Crossover, Multi-Point Crossover dan Order Crossover [11]. Menurut Chaturvedi, disarankan untuk memilih probabilitas *Crossover* sebesar 20 kali lebih besar dari probabilitas mutasi, yaitu berkisar antara 0,25 hingga 0,95 [14]. Sslah satu visualisasi dari Crossover yaitu One-Point Crossover ditunjukkan pada Gambar 2.3 [13].



Gambar 2.3. Visualisasi One-Point Crossover

- *Mutation*

Mutasi dilakukan untuk mengatasi bila hasil Crossover belum memberikan hasil yang maksimal. Mutasi berguna untuk memodifikasi sebagian kecil keturunan untuk memenuhi persyaratan yang diinginkan. Semakin besar probabilitas mutasi, semakin besar fluktuasi nilai *fitness*. Nilai probabilitas mutasi biasanya bervariasi antara 0,001 sampai 0,5 [14]. Visualisasi Mutasi ditunjukkan pada Gambar 2.4 [13]



Gambar 2.4. Visualisasi *Mutation*