**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

# **IDENTITAS PENGUSUL**

Nama : **Teguh Prasetyo Utomo**

NRP : **5107 100 158**

Dosen Wali : **Daniel O. Siahaan, S.Kom, M.Sc, P.D.Eng**

1. **JUDUL TUGAS AKHIR**

***“Implementasi Algoritma Genetik Hibrida pada Job Shop Scheduling Problem“***

1. **LATAR BELAKANG**

Dalam sebuah industri, baik itu industri manufaktur maupun jasa, adanya suatu proses penyusunan penjadwalan yang baik adalah sebuah hal yang penting. Hal ini dikarenakan dengan adanya penjadwalan yang baik akan dapat meningkatkan efektivitas serta efisiensi sistem produksi industri tersebut yang pada akhirnya akan mengurangi biaya produksi. Scheduling dapat diartikan sebagai pengalokasian sejumlah sumber daya untuk melakukan sejumlah operasi dalam jangka waktu tertentu. Baik secara teori maupun prakteknya di lapangan, untuk dapat melakukan suatu proses penjadwalan yang baik sangat sulit untuk dibuat. Hal ini berdasar pada kenyataan bahwa begitu banyak parameter yang harus diperhatikan[1].

Job shop scheduling problem merupakan salah satu masalah penjadwalan yang memiliki kendala urutan pemrosesan tugas, dan setiap tugas harus melalui setiap mesin tepat satu kali. Terdapat dua jenis metode yang biasa digunakan untuk menyelesaikan masalah job shop scheduling problem. Metode eksak, seperti pemrograman linier dan pemrograman non-linier, dapat digunakan untuk ukuran job shop scheduling problem yang kecil. Sedangkan untuk ukuran masalah yang besar, digunakan suatu pendekatan secara aproksimasi, seperti local search, simulated annealing, genetic algorithm, tabu search, dan ant colony optimization.

Algoritma genetik ditemukan oleh John Holland pada tahun 1960. Algoritma ini menerapkan suatu proses evolusi biologi. Pendekatan yang diambil oleh algoritma ini adalah dengan menggabungkan secara acak berbagai pilihan solusi terbaik di dalam suatu kumpulan untuk mendapatkan generasi solusi terbaik berikutnya yaitu pada suatu kondisi yang memaksimalkan kecocokannya atau lazim disebut fitness. Generasi ini akan merepresentasikan perbaikan-perbaikan pada populasi awalnya.

Banyak percobaan dalam menyelesaikan job shop scheduling problem dengan menggunakan metode algoritma genetik, tetapi masih terdapat beberapa percobaan yang menghasilkan solusi yang tidak layak. Pada penelitian ini, ketidaklayakan dari solusi dapat dihindari dengan menggunakan dua metode yaitu : operator crossover baru dan operator mutasi baru. Untuk meningkatkan kecepatan algoritma, pendekatan pencarian efisien lokal diusulkan dan diintegrasikan ke dalam algoritma genetika yang diusulkan. Kualitas dari solusi akan diuji dengan menggunakan beberapa masalah uji yang biasa dipakai untuk menyelesaikan job shop scheduling problem[2].

1. **TUJUAN TUGAS AKHIR**

Mengimplementasikan metode algoritma genetik hibrida yang baru yaitu operator crossover baru dan operator mutasi baru untuk menangani job shop scheduling problem.

1. **PERMASALAHAN**

Masalahnya adalah menemukan jadwal untuk meminimalkan waktu proses total (makespan), yaitu untuk meminimalkan waktu yang dibutuhkan untuk menyelesaikan semua pekerjaan.

1. **BATASAN MASALAH**

Batasan parameter yang digunakan pada simulasi ini yaitu ukuran populasi N=100, probabilitas crossover pc=0.7, probabilitas mutasi pm=0.1, dan probabilitas pencarian lokal pl=0.5. Pada operator seleksi, =0.8, dan =0.7. Pada operator pencarian lokal, =10. Algoritma dijalankan 10 kali pada tiap uji coba. Waktu pengerjaannya sekitar 4 bulan. Dataset yang digunakan berasal dari dua kelas standar permasalahan uji coba dari JSSP : Fisher and Thompson [3] instant FT06,FT10,FT20, dan Lawrence [4] instant LA01-LA40. Software yang digunakan dalam implementasi ini adalah Matlab 7.6.

1. **RINGKASAN TUGAS AKHIR**

Pada tugas akhir ini, akan diimplementasikan metode algoritma genetik hibrida yang terdiri dari operator crossover baru dan operator mutasi baru. Ide dasar dari algoritma ini adalah membentuk suatu populasi awal secara acak, yang akan diproses melalui tiga operator dasar dari algoritma genetik, sehingga menghasilkan populasi baru untuk generasi berikutnya. Tiga operator dasar tersebut adalah: reproduksi, crossover, dan mutasi. Reproduksi digunakan untuk menyeleksi kromosom-kromosom yang akan diproses, crossover digunakan untuk mengawinkan pasangan kromosom sehingga menghasilkan kromosom-kromosom baru, dan mutasi digunakan untuk mengubah satu atau lebih gen pada suatu kromosom.

Crossover (perkawinan silang) bertujuan menambah keanekaragaman string dalam populasi dengan penyilangan antar-string yang diperoleh dari sebelumnya. Tujuan dari crossover adalah untuk memperoleh keturunan yang lebih baik dengan memanfaatkan informasi orang tua. Dalam penelitian ini, operator crossover yang baru berdasarkan karakteristik dari JSSP itu sendiri dirancang. Misalkan x dan y adalah dua individu induk, operator crossover yang dirancang dalam makalah ini dapat digambarkan sebagai berikut.

**Algoritma 2.**

**Langkah 1** : Bagilah mesin-mesin (1,2,..m) dalam dua himpunan A dan B.

**Langkah 2** : Untuk dua parent *x* dan *y*, tukar gen dalam substring dalam himpunan B dengan probabilitas pc untuk mendapatakan dua anak x’ dan y’.

Tukar gen dalam substring himpunan B yang setara dengan menukar urutan dari proses operasi pada mesin yang ada di himpunan B. Dengan cara ini, anak secara efektif dapat mewarisi urutan dari proses operasi pada mesin yang ada di himpunan A. Operator crossover yang diusulkan seperti multipoint crossover, yang secara acak membagi mesin-mesin menjadi dua himpunan. Operator crossover ini lebih fleksibel dan meningkatkan keragaman populasi.

Mutasi merupakan proses mengubah nilai dari satu atau beberapa gen dalam suatu kromosom. Operator mutasi selanjutnya melakukan beberapa penyesuaian bagi individu parent dan mendapatkan keturunan. Hal ini dapat meningkatkan keragaman penduduk dan sangat membantu untuk keluar dari solusi optimal lokal.

Menurut Dell'Amica dan Tubian [5], ada sifat-sifat untuk JSSP :

1. Jika x adalah solusi yang layak, kemudian membalikkan salah satu ujungnya berorientasi pada jalur kritis dari x tidak pernah dapat mengarah pada solusi yang tidak layak.

2. Jika pembalikan berorientasi ujung dari solusi x yang layak yang tidak termasuk jalur kritis menyebabkan solusi x' yang layak, maka jalur kritis di x' tidak dapat menjadi lebih pendek dari jalur kritis dalam x.

Menurut sifat-sifat itu, operator mutasi berdasarkan pada jalur kritis dirancang. Misal PM(Oij) dan SM(Oij) masing-masing menunjukkan pendahulu dan pengganti dari operasi Oij pada mesin yang sama jika itu ada, jika tidak PM(Oij)=0, SM(Oij)=1, dimana Oij menunjukkan operasi ke-j dari pekerjaan ke-i. Kemudian operasi mutasi yang diusulkan dapat dijelaskan sebagai berikut.

**Algoritma 3.**

**Langkah 1** : Hitung jalur kritis dari parent. Anggap k = 1, anak = parent.

**Langkah 2** : Jika operasi Oikjk dan Oik+1jk+1 adalah proses pada mesin yang sama, maka semua kemungkinan permutasi dibalik, secara acak memilih satu diantaranya, dan gunakan urutan operasi ini untuk mengganti urutan sebelumnya pada anak dengan probabilitas dari *pm*.

**Langkah 3** : Jika *k<q* maka *k=k+1* dan menuju langkah 2, jika tidak, hasilnya adalah anak itu.

Berdasarkan penjelasan sebelumnya, telah diusulkan algoritma genetik hibrida baru untuk menyelesaikan JSSP. Algoritma tersebut dapat diuraikan sebagai berikut

**Langkah 1**. Inisialisasi. Berikan populasi awal P(0) dan t=0. Evaluasi nilai fitness setiap individu di P(0).

**Langkah 2**. Crossover. Gabungkan semua individu dalam P(t) secara acak, dan gunakan algoritma 2 untuk setiap pasangan individu dan dapatkan keturunan, namai dengan P'(t).

**Langkah 3**. Mutasi. Gunakan algoritma 3 untuk semua individu di P'(t) dan dapatkan keturunan, namai dengan P''(t)

**Langkah 4**. Evaluasi nilai fitness. Evaluasi nilai fitness setiap individu di P''(t)

**Langkah 5**. Pencarian Lokal. Gunakan algoritma 5 (operator pencarian lokal) untuk setiap individu dalam P''(t) dengan probabilitas *pl* dan dapatkan keturunan, namai dengan P'''(t)

**Langkah 6**. Seleksi. Pilih N individu dari P'''(t) dengan algoritma 1 (operator seleksi campuran) untuk mendapatkan populasi, tetap namai dengan P'''(t), lalu gunakan strategi elitist untuk menggabungkan P'''(t) dan P(t) untuk mendapatkan generasi populasi selanjutnya P(t+1).

**Langkah 7**. Jika kriteria tercapai, berhenti. Jika tidak, maka t = t+1 dan kembali ke langkah 2.

1. **METODOLOGI**

Ada beberapa tahap dalam proses pengerjaan tugas akhir ini. Berikut tahap-tahap dalam pembuatannya :

1. **Studi literatur**

Pada tahap ini akan dipelajari sejumlah literatur mengenai konsep dan teknologi yang akan digunakan. Literatur yang digunakan meliputi paper referensi, buku referensi, ebook dan dokumentasi internet.

1. **Perancangan perangkat perangkat lunak dan desain sistem**

Pada tahap ini dilakukan proses perancangan program, berdasarkan literatur yang telah di kaji kemudian di buat desain model data, diagram alir proses-proses yang ada, dan desain antar muka aplikasi. Yang selanjutnya kemudian akan diimplemetasikan.

1. **Pengimplementasian perangkat perangkat lunak**

Pada tahap ini dilakukan proses pengimplemetasian, dengan berdasar pada rancangan awal perangkat lunak dan literature yang telah dikaji. Aplikasi telah mulai dibuat secara menyeluruh.

1. **Uji coba dan evaluasi**

Pada tahap ini akan diuji aplikasi yang telah selesai diimplemetasikan tersebut. Pengujian dan evaluasi akan dilakukan pada dokumen-dokumen.

1. **Penyusunan laporan tugas akhir**

Pada tahap ini disusun buku sebagai dokumentasi dari pelaksanaan tugas akhir, yang mencakup seluruh konsep, teori, implementasi, serta hasil yang telah dikerjakan.Laporan tugas akhir ini akan dibagi menjadi beberapa bab sebagai berikut:

1. Bab I, Pendahuluan, berisi latar belakang, permasalahan, tujuan, batasan permasalahan, metodologi, dan sistematika penulisan.
2. Bab II, Landasan Teori, akan dibahas dasar ilmu yang mendukung pembahasan tugas akhir ini.
3. Bab III, Desain Aplikasi.
4. Bab IV, Implementasi dari aplikasi yang telah dibuat, akan dilakukan pembuatan aplikasi yang dibangun dengan komponen-komponen yang telah ada yang sesuai dengan permasalahan dan batasannya yang telah dijabarkan pada bab pertama.
5. Bab V, Uji coba dan analisa hasil, akan dilakukan uji coba berdasarkan parameter-parameter yang ditetapkan, dan kemudian dilakukan analisa terhadap hasil uji coba tersebut.
6. Bab VI, Penutup, berisi kesimpulan yang dapat diambil dari Tugas Akhir ini beserta saran untuk pengembangan selanjutnya
7. **DAFTAR PUSTAKA**

[1] Betrianis, Teguh Putu Aryawan.Penerapan Algoritma Tabu Search Dalam Penjadwalan Job Shop, 2003.

[2] Ren Qing-dao-er-ji, Yuping Wang.A new hybrid genetic algorithm for job shop scheduling problem, 2011.

[3] Fisher H, Thompson GL. Probabilistic learning combinations of local job-shop scheduling rules. Englewood Cliffs, NJ: Prentice-Hall; 1963. p. 225–51.

[4] Lawrence S. Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques. Technical Report, GSIA, Carnegie Mellon University; 1984.

[5] Dell’Amico M, Trubian M. Applying tabu search to the job shop scheduling

problem. Annals of Operation Research 1993;41:231–52

1. **JADWAL PEMBUATAN TUGAS AKHIR**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Kegiatan** | **Bulan** | | | | | | | |
| **Maret** | | **April** | | **Mei** | | **Juni** | |
| Studi Literatur |  |  |  |  |  |  |  |  |
| Perancangan Perangkat Lunak dan Desain Sistem |  |  |  |  |  |  |  |  |
| Pengimplementasian Perangkat Lunak |  |  |  |  |  |  |  |  |
| Uji Coba dan Evaluasi |  |  |  |  |  |  |  |  |
| Penyusunan Laporan Tugas Akhir |  |  |  |  |  |  |  |  |

**LEMBAR PENGESAHAN**

###### **Surabaya, 06 Maret 2012**

Menyetujui,

Dosen Pembimbing I

# **(Yudhi Purwananto, S.Kom, M.Kom)**

# **( NIP.19700714 199703 1 002)**

Dosen Pembimbing II

(Rully Sulaiman, S.Kom, M.Kom.)

( NIP.19700213.199402.1001)