BAB I
PENDAHULUAN

1.1 Latar Belakang

Masalah *Flowshop Scheduling Problem* (FSP) berperan penting dalam sistem manufaktur. Teknik penjadwalan yang bagus secara signifikan dapat meningkatkan efisiensi produksi. Untuk mencapai urutan yang baik dalam persaingan pasar, metode penjadwalan yang efektif selalu dibutuhkan.*Permutation Flowshop Scheduling Problem* (PFSP) adalah salah satu masalah penjadwalan produksi yang paling populer, dan dapat merupakan versi sederhana dari FSP. Dalam PFSP, setiap mesin hanya dapat memproses satu pekerjaan pada satu waktu, dimana urutan pekerjaan yang sama diikuti di semua mesin. Meskipun pada umumnya masalah ini dipelajari sebagai permasalahan dengan satu fungsi tujuan, namun kenyataannya terdapat juga permasalahan PFSP yang memerlukan pengoptimalan pada dua fungsi tujuan (*bi-objective*), seperti terdapatnya waktu tenggat untuk suatu pekerjaan, sehingga jika pekerjaan tersebut terlambat untuk diselesaikan maka perusahaan akan dikenakan pinalti atau beban keterlambatan (*weighted tardiness*) yang merugikan. Oleh karena itu selain meminimumkan *makespan* diperlukan juga untuk meminimalkan total beban keterlambatan**(Wang dan Tang, 2016).**

Dalam mengatasi permasalahan penjadwalan, terdapat berbagai macam algoritma yang telah digunakan dalam menyelesaikan *Bi-objective Permutation Flowshop Scheduling Problem* (BPFSP) seperti algoritma *Branch and Bound* untuk mengoptimalkan *total flow time* dan *total tardiness***(Lee dan Wu, 2001)** ataupun menggunakan *Ant Colony Optimization* (ACO) untuk mengoptimalkan *makespan* dan *total flowtime***(Yenisey dan Yagmahan, 2010)**. Dalam beberapa tahun terakhir, beberapa algoritma metaheuristik telah diusulkan untuk memecahkan permasalahan penjadwalan flowshop antara lain adalah *Genetic Algorithm* (GA), *Ant Colony Optimization* (ACO) *Algorithm, Simulated Annealing* (SA) *Algorithm, Particle Swarm Optimization* (PSO) *Algorithm, dan Cuckoo Search Algorithm* (CSA) **(Marichelvam 2012)**.

Cuckoo Search Algorithm (CSA) adalah salah satu algoritma metaheuristik yang terinspirasi oleh alam, dikembangkan pada tahun 2009 oleh Xin-She Yang dan Suash Deb. Pada penelitian sebelumnya menunjukkan bahwa CSA mengungguli beberapa algoritma yang ada seperti algoritma genetika (GA) dan *Particle Swarm Optimization* (PSO) **(Kennedy dan Eberhart 1995)**. Cuckoo adalah burung yang menarik, bukan hanya karena suara indah yang mereka dapat buat, tetapi juga karena strategi reproduksi agresif burung tersebut. Meskipun burung *cuckoo* mungkin membuang telur spesieslain untuk meningkatkan kemungkinan penetasan telur mereka sendiri, cukup jumlah spesies melibatkan parasitisme induk yang bertanggung jawab dengan bertelur di dalam sarang burung tuan rumah lainnya**(Yang dan Deb 2009)**.

Pada penelitian sebelumnya, ada beberapa ahli seperti **Civicioglu dan Besdok(2011)** yang telah membuktikan bahwa*Cuckoo Search Algorithm* (CSA) mempunyai hasil lebih baik dari *Particle Swarm Optimization* (PSO) dan *Artificia Bee Colony* (ABC). Serta **Gandomi dkk., 2011** telah memecahkan berbagai masalah optimasi dan menyimpulkan bahwa *Cuckoo Search Algorithm* (CSA) mempunyai hasil lebih baik dari *Particle Swarm Optimization* (PSO) dan *Genetic Algorithm* (GA).Keuntungan lebih lanjut dari *Cuckoo Search Algorithm* (CSA) adalah pencarian globalnya yang menggunakan *Lévy Flights Random Walk* (LFRW). Keuntungan ini, dikombinasikan dengan kemampuan pencarian lokal dan kovergensi global yang terjamin, membuat *Cuckoo Search Algorithm* (CSA) sangat efisien, berbagai penelitian dan aplikasi telah menunjukkan bahwa *Cuckoo Search Algorithm* (CSA) sangat efisien dalam mencari nilai optimal dengan tingkat keberhasilan lebih tinggi dibandingkan dengan PSO dan GA**(Yang dan Deb, 2009).** Oleh karena itu, untuk mengembangkanpenerapan *Cuckoo Search Algorithm* (CSA), dibuatlah penerapan *Cuckoo Search Algorithm* (CSA) untuk menyelesaikan *Bi-objective Permutation Flowshop Scheduling Problem* (BPFSP).

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka dalam penelitian ini membahas tentang :

1. Bagaimana menerapkan *Cuckoo Search Algorithm* (CSA) untuk menyelesaikan *Bi-objective Permutation Flowshop Scheduling Problem* (BPFSP)?

2. Bagaimana membuat program *Cuckoo Search Algorithm* (CSA) untuk menyelesaikan *Bi-objective Permutation Flowshop Scheduling Problem* (BPFSP)?

3. Bagaimana mengimplementasikan program *Cuckoo Search Algorithm* (CSA) untuk menyelesaikan *Bi-objective Permutation Flowshop Scheduling Problem*(BPFSP) pada contoh kasus?

1.3 Tujuan

Tujuan dalam penelitian ini adalah sebagai berikut:

1. Menerapkan *Cuckoo Search Algorithm* (CSA) untuk menyelesaikan *Bi-objective Permutation Flowshop Scheduling Problem* (BPFSP).
2. Membuat program *Cuckoo Search Algorithm* (CSA) untuk menyelesaikan *Bi-objective Permutation Flowshop Scheduling Problem* (BPFSP).
3. Mengimplementasikan program *Cuckoo Search Algorithm* (CSA) untuk menyelesaikan*Bi-objective Permutation Flowshop Scheduling Problem* (BPFSP) pada contoh kasus.

1.4 Manfaat

Manfaat dalam penelitian ini adalah sebagai berikut:

1. Menambah wawasan mahasiswa tentang cara menyelesaikan *Bi-Objective Permutation Flowshop Scheduling Problem* dengan menggunakan *Cuckoo Search Algorithm.*
2. Sebagai referensi dalam menerapkan algoritma lainnya untuk menyelesaikan *Bi-Objective Permutation Flowshop Scheduling Problem.*