ABSTRAK

Salah satu cara untuk meningkatkan efisiensi kerja adalah dengan penjadwalan kerja (produk/jasa) yang tepat. Banyaknya jenis industri yang ada membutuhkan variasi karakteristik penjadwalan yang berbeda-beda agar dapat disesuaikan. Problem penjadwalan yang sering terjadi adalah adanya *waiting time job* antar mesin dan adanya *unavailable interval* yang sudah diketahui sebelumnya tetapi belum diperhitungkan secara tepat dalam penjadwalan. Hal ini menyebabkan total waktu proses *jobs* menjadi semakin lama. Problem seperti ini dapat terjadi di berbagai jenis industri, misalnya pabrik pembuatan obat, pembuatan *ice cream*, dll. Untuk itu diperlukan penjadwalan *flowshop no-wait* m-mesin disertai *availability constraint* dengan tujuan minimum *makespan*. Penelitian-penelitian yang ada sebelumnya hanya membahas algoritma penjadwalan *flowshop no-wait* saja dan algoritma penjadwalan *flowshop no-wait* yang sudah memperhitungkan *availability constraint* tetapi untuk penjadwalan pada 2 mesin.

Dari perhitungan uji coba algoritma awal didapatkan kesimpulan bahwa algoritma *no-wait Traveling Salesman Problem* (TSP) selalu memberikan tingkat keberhasilan yang lebih besar dibandingkan 2 algoritma lainnya. Tingkat keberhasilan suatu algoritma dilihat berdasarkan *makespan* minimum yang dihasilkan (sesuai tujuan penjadwalan yang sudah ditetapkan). Sedangkan melalui uji coba algoritma awal untuk penjadwalan *no-wait* 2 mesin yang disertai *availability constraint* dapat disimpulkan *approximation scheme* selalu menghasilkan *makespan* yang lebih minimum. Berdasarkan kesimpulan akan dilakukan pengembangan algoritma.

Pengembangan selanjutnya dilakukan dengan mengganti algoritma *no-wait Gilmore and Gomory* (GG) dengan algoritma TSP2 pada *approximation scheme* untuk penjadwalan 2 mesin dengan memperhitungkan *availability constraint.* Algoritma yang baru dikembangkan ini dinamakan APP TSP2 (gabungan antara algoritma *no-wait* TSP untuk penjadwalan 2 mesin dan *approximation scheme*). Berdasarkan pengembangan awal ini didapatkan kesimpulan bahwa APP TSP2 hampir selalu menghasilkan persentase *makespan* minimum yang lebih besar. Untuk itu pengembangan berikutnya didasarkan pada APP TSP2 dengan cara mengganti algoritma TSP2 menjadi algoritma TSPm. Dengan pengembangan ini didapatkan algoritma usulan yang disebut APP TSPm.

Algoritma TSPm adalah algoritma usulan gabungan antara algoritma *no-wait* TSPm dan *approximation scheme* untuk penjadwalan *no-wait* m-mesin disertai *availability constraint* dengan tujuan minimum *makespan*. Berdasarkan perhitungan dan analisis pada algoritma usulan didapatkan kesimpulan bahwa algoritma APP TSPm hampir selalu menghasilkan *makespan* minimum untuk *jobs* dengan semua jenis data atau jenis proses. Persentase *makespan* minimum yang dihasilkan algoritma APP TSPm untuk *jobs* dengan jenis data *bottleneck* di awal proses, *bottleneck* di tengah proses dan *bottleneck* di akhir proses secara berturut-turut adalah 88,3%; 73,3% dan 73,3%. Sedangkan persentase *makespan* minimum yang dihasilkan untuk *jobs* yang diproses secara *resumable, semi-resumable* dan *non-resumable* secara berturut-turut adalah 78,3%; 76,6% dan 80%. Pengaruh perubahan fraksi pengulangan proses (α) pada *makespan* hanya sebesar lama pengulangan proses. Pengaruh posisi *availability constraint* terhadap *makespan* menjadi cukup signifikan apabila posisi *availability constraint* sama dengan posisi operasi *bottleneck* dan apabila *availability constraint* terjadi pada setiap mesin.