

## BAB V

### PEMBAHASAN

Penelitian ini dilakukan dengan tujuan untuk mengontrol *bottleneck* agar dapat melindungi laju produksi dari gangguan yang terjadi dalam sistem produksi dengan menambah suatu penyangga (buffer). Penambahan sejumlah *Time Buffer* dapat melindungi laju produksi dan akan mempengaruhi *makespan* yang sangat berkaitan erat dengan performansi *duedate*. Dari kondisi tersebut maka diperlukan suatu tindakan bagaimana mengestimasi *duedate*, agar diperoleh performansi *duedate* yang baik berdasarkan tardiness dan lateness dengan pendekatan *TOC Aggregate*.

Dari hasil pengolahan data pada bab IV, berdasarkan algoritma zijm, pada rantai produksi mengalami *bottleneck* pada mesin ke empat, ini dibuktikan dengan waktu proses sebesar 31.08 menit, selain waktu proses di mesin ke empat juga mengalami beban kerja terbesar yaitu 0.117011038 menit dan ekspektasi menunggu setiap job terbesar yaitu 0.199680303 menit. Selanjutnya setelah mendeteksi mesin *bottleneck* dengan menggunakan Algoritma Zijm kemudian menentukan *time buffer*, dengan cara menganalisa slack antara ETC, LTC dengan *completion time* di mesin empat. Pada penelitian ini perhitungan *makespan* dilakukan dengan tujuan mencari urutan job yang optimal, dengan metode FCFS yang hasilnya dapat dilihat pada tabel 4.11, EDD yang hasilnya dapat dilihat pada tabel 4.27, SPT yang hasilnya dapat dilihat pada tabel 4.43, LPT yang hasilnya dapat dilihat pada tabel 4.59 dan CR yang hasilnya dapat dilihat pada tabel 4.75. Penambahan sejumlah *Time Buffer* akan mempengaruhi *makespan* yang sangat berkaitan erat dengan performansi *duedate*. Performansi *duedate* dapat diketahui dengan menganalisa tardiness dan latenessnya, yang hasilnya dapat dilihat pada tabel 4.15, 4.31, 4.47, 4.63, 4.79.

Peningkatan performansi *duedate* melalui *TOC Aggregate* untuk menghindari ketidakpastian dari aktivitas produksi agar diperoleh performansi *duedate* yang lebih baik berdasarkan *tardiness* dan *lateness*. Algoritma yang digunakan pada *TOC Aggregate* adalah *Aggregate Time Buffer (ATB)*. Dalam pemakaian konsep *TOC Aggregate* hanya stasiun kerja yang *bottleneck* yang perlu diperhatikan. Performansi *duedate* yang lebih baik dapat diketahui dengan menghitung *tardiness* dan *lateness*nya, yang dapat dilihat pada tabel 4.25, 4.41, 4.57, 4.73, 4.89. Dari hasil analisa dengan menggunakan ATB dapat meningkatkan performansi *duedate* setiap job berdasarkan *Aggregate time buffer* pada masing-masing job yang hasilnya dilihat pada tabel 4.90.

Algoritma ATB dapat meningkatkan performansi *duedate* secara signifikan. Untuk metode FCFS terjadi peningkatan sebesar 95.522 %, metode LPT terjadi peningkatan sebesar 91.6607 %, metode CR terjadi peningkatan sebesar 37.9923 %, metode EDD terjadi peningkatan sebesar 37.9203 %, dan peningkatan SPT sebesar 35.458 %.

