

BAB 6

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Dengan demikian, kesimpulan yang dapat diambil adalah :

1. Cara menganalisis algoritma *earliest deadline first* yang ada pada perangkat lunak tahap perancangan salah satunya adalah dengan menganalisis algoritma yang memiliki fungsi yang sama, namun dengan cara kerja yang sedikit berbeda yaitu algoritma *shortest remaining time first*. Setelah menganalisis kedua algoritma tersebut, ditemukan bahwa pada algoritma *shortest remaining time first*, pekerjaan dengan waktu pengerjaan paling singkat dikerjakan terlebih dahulu oleh sistem penjadwalan, sedangkan pada algoritma *earliest deadline first*, pekerjaan dengan batas waktu pengerjaan paling awal dikerjakan terlebih dahulu oleh sistem penjadwalan.
2. Perancangan perangkat lunak penjadwalan dilakukan dengan cara menentukan cara kerja algoritma *earliest deadline first*, membangun rancangan antarmuka pengguna, membangun rancangan diagram *use case*, dan *activity diagram*.
3. Pengembangan perangkat lunak penjadwalan dilakukan dengan mengimplementasikan algoritma *earliest deadline first* untuk membandingkan dua pekerjaan berdasarkan *deadline* sehingga perangkat lunak dapat mengerjakan pekerjaan dengan *deadline* yang paling cepat.
4. Semakin banyak pekerjaan yang dijadwalkan oleh perangkat lunak, maka semakin besar pula nilai *average waiting timenya*. Nilai *average turnaround time* bervariasi (naik dan turun) tergantung dari selisih kapan sebuah pekerjaan selesai dijadwalkan dan *arrival time* dari pekerjaan tersebut. Dan juga, semakin besar selisih dari *burst time* dan *deadline time* pada pekerjaan, maka semakin jauh perbedaan AWT dan ATThya. Sebaliknya, semakin kecil selisih dari *burst time* dan *deadline time*, maka semakin seragam perbedaan AWT dan ATThya.

Dan juga, dengan melakukan pengujian eksperimental pada perangkat lunak, dapat dipastikan bahwa perangkat lunak penjadwalan dengan algoritma EDF dapat berjalan dengan benar karena hasil pengujian dari perangkat lunak secara keseluruhan sesuai dengan hasil harapan yang telah diuraikan pada bab 5. Adapun beberapa kondisi dari pengujian perangkat lunak yang ada pada bab 5 adalah sebagai berikut :

- (a) Tidak ada *miss* dan *idle*.
- (b) Ada *miss* dan tidak ada *idle*.
- (c) Tidak ada *miss* dan ada *idle*.
- (d) Ada *miss* dan *idle*.
- (e) Mode *hard real time* aktif dan ada *miss*.

6.2 Saran

Berdasarkan hasil yang telah ada, penulis memiliki beberapa saran untuk membuat perangkat lunak simulasi penjadwalan dengan menggunakan algoritma *earliest deadline first* ini dapat berjalan lebih baik lagi. Diantaranya adalah sebagai berikut :

1. Perangkat lunak dapat ditambahkan fitur untuk melakukan *pause* saat penjadwalan sedang berjalan. Pada saat *pause* dilakukan, pengguna dapat melihat ATT dan AWT sementara dari penjadwalan yang sedang berjalan.
2. Perangkat lunak dapat ditambahkan fitur untuk menambahkan hasil representasi selain dalam bentuk *gant chart*. Contohnya adalah dalam bentuk *kanban board*.

DAFTAR REFERENSI

- [1] Stallings, W. (2014) *Operating Systems: Internals and Design Principles, 8th Edition*, 8th edition. Pearson, US.
- [2] Tanenbaum, A. S. dan Bos, H. (2015) *Modern Operating Systems*, 4th edition. Pearson, Boston.
- [3] Maylor, H. (2001) Beyond the gantt chart:: Project management moving on. *European Management Journal*, **19**.
- [4] Baruah, B. M. B., Sanjoy dan Giorgio (2015) *Multiprocessor Scheduling for Real-Time Systems*, 1st edition. Springer, Switzerland.