

BAB II

LANDASAN TEORI

2.1. Penjadwalan

Menurut Pinedo (2002), penjadwalan adalah proses pengambilan keputusan yang mempunyai peran penting dalam proses manufaktur dan sistem produksi begitu juga dalam lingkungan pemrosesan informasi. Penjadwalan juga terdapat dalam transportasi dan distribusi serta dalam industri.

Menurut Farida (2008), Penjadwalan merupakan kumpulan kebijaksanaan dan mekanisme dalam sistem operasi yang berhubungan dengan urutan kerja yang dilakukan sistem komputer. Penjadwalan digunakan untuk memutuskan proses yang harus berjalan serta kapan dan selama berapa lama proses tersebut berjalan.

Sasaran utama proses penjadwalan:

1. Adil, tidak ada proses yang tidak kebagian layanan.
2. Efisien, pemroses dijaga tetap bekerja agar tidak ada waktu yang terbuang sia-sia.
3. Waktu tanggap, termasuk di dalamnya sistem waktu interaktif dan sistem waktu nyata.
4. *Turn around time*, waktu yang diperlukan untuk serangkaian satu proses.
5. *Throughput*, jumlah kerja yang dapat dilakukan dalam satu satuan waktu.

Permasalahan dalam proses penjadwalan dapat diselesaikan dengan menggunakan algoritma optimasi.

2.2. Algoritma Optimasi

Menurut Suyanto (2010), algoritma optimasi dapat didefinisikan sebagai algoritma atau metode numerik untuk menemukan nilai x sedemikian hingga menghasilkan $f(x)$ yang bernilai sekecil (atau sebesar) mungkin untuk suatu fungsi f yang diberikan, yang mungkin disertai dengan beberapa batasan pada x . Di sini, x bisa berupa skalar atau vektor dari nilai-nilai kontinu maupun diskrit.

Algoritma optimasi didefinisikan juga sebagai suatu cabang dari matematika terapan dan analisis numerik yang membahas optimasi dengan kriteria yang bersifat tunggal, ganda atau bahkan mungkin konflik. Hasil dari suatu proses optimasi adalah suatu himpunan masukan yang membuat fungsi-fungsi objektif menghasilkan nilai-nilai optimal (yang bisa berupa maksimal atau minimal). Dari beberapa algoritma optimasi yang ada, algoritma *Tabu Search* dapat digunakan untuk menyelesaikan masalah optimasi yang sulit.

2.3. *Tabu Search*

Menurut Glover dan Laguna (1998), *Tabu Search* pertama kali diperkenalkan oleh Glover sekitar tahun 1986. Glover menyatakan bahwa *tabu search* adalah salah satu prosedur metaheuristik tingkat tinggi untuk penyelesaian permasalahan optimasi kombinatorial. *Tabu search* ini dirancang untuk mengarahkan metode-metode lain (atau komponen proses *tabu search* itu sendiri) untuk keluar atau menghindari dari masuk ke dalam solusi optimal yang bersifat lokal. Kemampuan *tabu search* dalam menghasilkan solusi yang mendekati optimal telah dimanfaatkan dalam beragam permasalahan di berbagai bidang mulai penjadwalan hingga bidang telekomunikasi.

Menurut Suyanto (2010), *Tabu Search* adalah suatu metode optimasi matematis yang termasuk ke dalam *local search*. *Tabu search* memperbaiki performansi *local search* dengan memanfaatkan penggunaan struktur memori. Sebagian solusi yang pernah diberikan ditandai sebagai “*tabu*” (dalam ejaan lain adalah “*taboo*” yang berarti sesuatu yang terlarang), sehingga algoritma *tabu search* tidak akan mengangkat solusi tersebut secara berulang-ulang. Hal ini yang membuat *Tabu Search* menjadi lebih efisien dalam hal usaha dan waktu.

Tabu Search menggunakan struktur memori yang disebut *Tabu List* untuk menyimpan atribut dari sebagian move (langkah transisi dari satu solusi ke solusi yang lain) yang telah diterapkan pada iterasi-iterasi sebelumnya. *Tabu List* digunakan untuk menolak solusi-solusi yang memenuhi atribut tertentu agar proses pencarian tidak berulang-ulang pada daerah solusi yang sama dan untuk menuntun proses pencarian menelusuri solusi-solusi yang belum pernah dikunjungi.

Untuk efisiensi memori dan waktu proses, *Tabu List* hanya menyimpan langkah transisi (move) yang merupakan kebalikan dari langkah yang telah digunakan pada iterasi-iterasi sebelumnya. Dengan kata lain, *Tabu List* hanya berisi langkah-langkah yang mengembalikan solusi yang baru ke solusi yang lama. Dengan menggunakan *Tabu List*, *Tabu Search* dapat menerima solusi yang tidak memberikan peningkatan kualitas, sehingga *Tabu Search* dapat keluar dari optimum lokal. Nilai optimum lokal sendiri adalah nilai solusi optimal dalam beberapa kriteria solusi.

Menurut Kusumadewi dan Purnomo (2005), *Tabu Search* merupakan suatu metode optimasi yang menggunakan *short-term memory* untuk menjaga

agar proses pencarian tidak terjebak pada nilai optimum lokal. Metode ini menggunakan *Tabu List* untuk menyimpan sekumpulan solusi yang baru saja dievaluasi. Selama proses optimasi, pada setiap iterasi, solusi yang akan dievaluasi akan dicocokkan terlebih dahulu dengan isi *Tabu List* untuk melihat apakah solusi tersebut sudah ada pada *Tabu List*. Apabila solusi tersebut sudah ada pada *Tabu List*, maka solusi tersebut tidak akan dievaluasi lagi pada iterasi berikutnya. Apabila sudah tidak ada lagi solusi yang tidak menjadi anggota *Tabu List*, maka nilai terbaik yang baru saja diperoleh merupakan solusi yang sebenarnya.

Algoritma *tabu search* secara garis besar dapat ditulis sebagai berikut :

Langkah 1. Tentukan solusi awal i di dalam S .

Langkah 2. Cari j terbaik dalam $N(i)$ (seperti $f(j) \leq f(k)$ untuk k di dalam $N(i)$).

Langkah 3. Jika $f(j) \geq f(i)$ maka berhenti. Jika tidak maka $i=j$ dan pergi ke

Langkah 2.

Menurut Hertz dkk (1995), algoritma di atas berisikan langkah-langkah dalam membangun proses iterasi dari solusi awal(i), solusi berikutnya(j) dan memeriksa apakah proses iterasi harus dihentikan atau mengulangi proses iterasi(k) kembali. Setiap prosedur iterasi akan menghasilkan solusi yang layak dari solusi awal(i) dan solusi berikutnya(j).

2.4. Aplikasi

Menurut Pressman (2010), aplikasi adalah elemen sistem logis daripada fisik. Karena itu, aplikasi memiliki karakteristik yang jauh berbeda dibandingkan dengan perangkat keras:

- a. Aplikasi dikembangkan atau direkayasa
- b. Aplikasi tidak habis pakai
- c. Meskipun industri sekarang ini terus bergerak menuju pengembangan berbasis komponen, sebagian besar aplikasi terus diperbaharui sesuai dengan kebutuhan.

2.5. Siklus Hidup Pengembangan Sistem

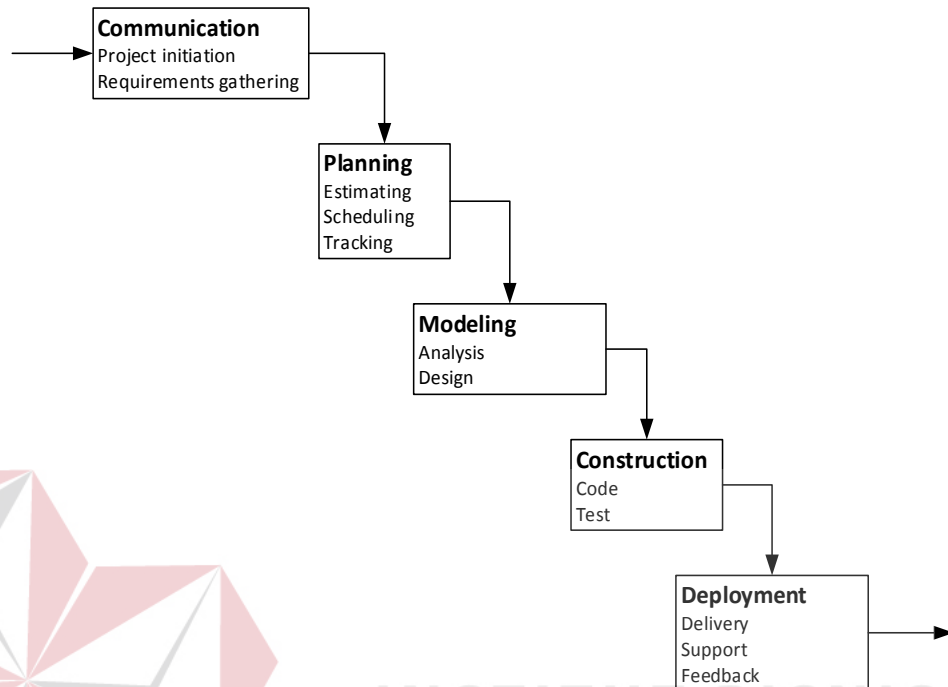
Menurut Laudon dan Laudon (2007), siklus hidup pengembangan sistem adalah metode pengembangan sistem informasi yang paling tua. Metodologi siklus hidup adalah pendekatan bertahap untuk membangun sistem, membagi pengembangan sistem menjadi tahapan-tahapan yang formal.

Sedangkan menurut Hoffer dkk (2011), siklus hidup pengembangan sistem adalah metodologi umum untuk pengembangan sistem di banyak organisasi. Di dalam siklus ini terdapat beberapa fase yang digunakan untuk menandai kemajuan analisa sistem dan desain. Siklus hidup dapat dianggap sebagai proses yang melingkar di mana akhir dari sebuah sistem dapat mengarah kepada proyek lain yang akan mengembangkan versi baru atau mengganti sistem yang ada.

2.6. Model *Waterfall*

Menurut Pressman (2010), model *waterfall* adalah siklus hidup klasik, menunjukkan secara sistematis, pendekatan sekuensial untuk pengembangan perangkat lunak yang diawali dengan spesifikasi persyaratan pelanggan dan berkembang melalui *planning*, *modeling*, *construction*, dan *deployment*,

berpuncak pada dukungan yang berkelanjutan dari perangkat lunak yang telah dibuat. Gambar model *waterfall* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Model *Waterfall* (Pressman, 2010)

Kerangka proses generik untuk rekayasa perangkat lunak meliputi lima kegiatan berikut:

a. *Communication*(Komunikasi)

Sebelum pekerjaan teknis dapat dimulai, sangat penting untuk berkomunikasi dan berkolaborasi dengan pelanggan atau *stakeholder*. Tujuannya adalah untuk memahami kemauan dari stakeholder untuk proyek ini dan untuk mengumpulkan kebutuhan yang membantu mendefinisikan fitur-fitur dan fungsi-fungsi aplikasi.

b. *Planning*(Perencanaan)

Proyek pembuatan aplikasi adalah perjalanan yang rumit, dan perencanaan adalah kegiatan yang membuat “peta” yang membantu dalam pengerjaan proyek. Peta ini disebut juga *software project plan* berisikan rencana kerja rekayasa aplikasi dengan menggambarkan tugas-tugas teknis yang akan dikerjakan, resiko apa saja yang bisa terjadi, sumber daya apa yang dibutuhkan, hasil yang diharapkan dan jadwal kerja.

c. *Modelling*(Pemodelan)

Proses yang dilakukan untuk menggambarkan fitur-fitur dan fungsi-fungsi dari aplikasi yang akan dibuat. Sehingga dapat memahami dan memenuhi kebutuhan dari stakeholder.

d. *Construction*(Konstruksi)

Proses yang menggabungkan pembuatan aplikasi baik menggunakan pembuatan secara manual maupun secara otomatis. Proses ini juga mencakup proses uji coba terhadap aplikasi yang sudah dibuat untuk mengetahui apakah masih ada kesalahan dalam pembuatan aplikasi.

e. *Deployment*

Aplikasi yang sudah dibuat akan diberikan kepada stakeholder sehingga dapat dievaluasi dan memberikan saran dan kritik berdasarkan evaluasi terhadap aplikasi.