

BAB II

TINJAUAN PUSTAKA

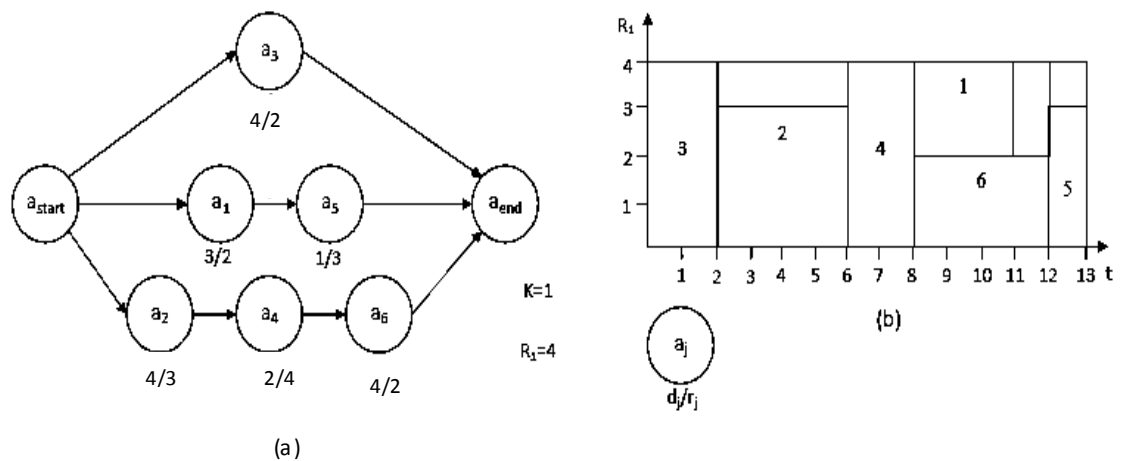
Dalam penulisan ini, diperlukan beberapa informasi yang berkaitan dengan persoalan yang akan diselesaikan.

2.1 *Resource Constrained Project Scheduling*

Resource Constrained Project Scheduling (RCPS) atau penjadwalan proyek dengan sumber daya terbatas dapat dijelaskan sebagai berikut. Sebuah proyek tunggal yang terdiri dari J aktivitas yaitu $J = \{ 0, 1, \dots, n, n+1 \}$ dengan n menyatakan banyaknya aktivitas. Aktivitas 0 dan $n+1$ menyatakan awal dan akhir aktivitas. Aktivitas saling terkait oleh dua jenis kendala. Pertama, penyelesaian beberapa kendala aktivitas pendahulu langsung atau *precedence constrained* memaksa kegiatan j tidak dapat dimulai sebelum semua aktivitas pendahulu terdekatnya telah selesai. Kedua, melangsungkan aktivitas membutuhkan sumber daya dengan kapasitas terbatas. Misalkan K adalah jenis sumber daya, yang diberikan oleh himpunan $K = \{1, \dots, k\}$. Sumber daya tipe k memiliki kapasitas terbatas b_k pada setiap titik waktu. Tujuan dari RCPS adalah untuk mencari penyelesaian beberapa aktivitas, sumber daya, dan waktu penyelesaian yang layak untuk semua kegiatan. Dengan demikian, waktu penyelesaian proyek terminimalkan.

(Hartmann dan Kolisch,1998)

Gambar 2.1 adalah contoh jaringan proyek dengan 6 aktivitas, 1 tipe sumber daya dengan kapasitas 4 unit. (a) menggambarkan jaringan aktivitas (b) menggambarkan jadwal yang optimal.



Gambar 2.1 Contoh Jaringan Proyek

Definisi 2.1 Proyek (*project*) adalah satu kombinasi kegiatan-kegiatan yang saling berkaitan dan harus dilakukan dalam urutan tertentu sebelum keseluruhan tugas dapat diselesaikan. (**Taha, 1997**)

Definisi 2.2 Aktivitas adalah sebuah tugas yang memerlukan waktu dan sumber daya untuk menyelesaikannya. (**Taha, 1997**)

Definisi 2.3 Sumber daya adalah hal-hal seperti pekerja, alat-alat bangunan dll, yang dapat diperoleh untuk memenuhi atau menyelesaikan suatu kegiatan dalam sebuah proyek. (**Suyanto, 2010**)

Definisi 2.4 *Precedence Constraints* adalah kendala aktivitas pendahulu langsung artinya sebuah aktivitas bisa dikerjakan jika aktivitas tepat sebelumnya selesai dikerjakan.

Menurut **Gen dan Cheng, 1997**, penjadwalan proyek dengan sumber daya terbatas diidentifikasi sebagai berikut :

1. Sebuah proyek yang terdiri atas sejumlah aktivitas yang diketahui durasinya.
2. Waktu mulai (*start time*) dari setiap aktivitas berhubungan dengan penyelesaian beberapa kendala aktivitas pendahulu langsung (*precedence constraints*).
3. Kebutuhan sumber daya tetap selama penjadwalan.
4. Sumber daya tersedia dalam jumlah yang terbatas tetapi dapat diperbarui dari waktu ke waktu.
5. Tidak ada substitusi antar sumber daya.
6. Aktivitas tidak dapat diganggu.
7. Hanya ada satu mode pengerjaan untuk setiap aktivitas.
8. Bertujuan untuk meminimalkan durasi proyek.

Secara matematis, masalah ini dapat dituliskan sebagai berikut :

$$\text{Min } t_n \quad (2.1)$$

Kendala :

$$t_j - t_i \geq d_i ; \forall j \in S_i \quad (2.2)$$

$$\sum_{t_i \in A_{t_i}} r_{ik} \leq b_k ; k = 1, 2, \dots, p \quad (2.3)$$

$$t_i \geq 0, i = 1, 2, \dots, n. \quad (2.4)$$

Dengan :

t_i : Waktu mulai (*starting time*) dari aktivitas i .

d_i : Durasi (*processing time*) dari aktivitas i .

S_i : Kumpulan aktivitas yang bisa segera dijadwalkan setelah aktivitas i .

r_{ik} : Banyaknya sumber daya tipe k yang dibutuhkan oleh aktivitas i .

b_k : Total ketersediaan sumber daya tipe k .

A_{t_i} : Kumpulan aktivitas yang sedang dikerjakan pada saat t_i .

p : Banyaknya tipe sumber daya

n : Banyaknya aktivitas

Persamaan (2.1) merepresentasikan fungsi tujuan dari RCPSP. Persamaan (2.2) merepresentasikan *precedence constraints* terpenuhi. Persamaan (2.3) merepresentasikan pada suatu waktu, jumlah sumber daya tipe k yang digunakan oleh seluruh aktivitas yang sedang dikerjakan tidak melebihi jumlah sumber daya tipe k yang tersedia.

2.2 *Priority Scheduling Method*

Menurut **Lee dan Kim (1996)**, *Priority Scheduling Method* merupakan metode penjadwalan menggunakan aturan prioritas untuk menyelesaikan konflik antara aktivitas dalam memperebutkan sumber daya yang tersedia. Nilai prioritas dari aktivitas biasanya ditentukan menggunakan informasi dari jaringan proyek dan aktivitas. Berikut ini prosedur penjadwalan berdasarkan prioritas :

1. Menentukan nilai prioritas dari setiap aktivitas. Menetapkan waktu sekarang t_{now} menjadi 0. Menginisialisasi vektor ketersediaan sumber daya dengan

komponen ke- i (r_i) menyatakan jumlah sumber daya tipe i yang tersedia pada saat t_{now} .

2. Jika semua aktivitas telah dijadwalkan, berhenti. Jika tidak, atur ulang E yaitu kumpulan aktivitas yang bisa dijadwalkan, dengan kata lain aktivitas yang semua aktivitas tepat sebelumnya telah dijadwalkan pada saat t_{now} dan membutuhkan sumber daya tidak lebih dari r_i untuk setiap i . Jika E kosong, lanjutkan ke langkah 3. Jika tidak, lanjutkan ke langkah 4.
3. Mengatur ulang t_{now} menjadi waktu tercepat dari aktivitas yang sedang diproses dan akan selesai. Memperbarui vektor ketersediaan sumber daya. Kembali ke langkah 2.
4. Memilih sebuah aktivitas dengan prioritas tertinggi dari seluruh aktivitas di E dan jadwalkan dengan cara menentukan waktu mulai dan waktu selesai dan perbarui vektor ketersediaan sumber daya. Kembali ke langkah 2.

Dengan metode penjadwalan berdasarkan prioritas diatas, jadwal yang lengkap dapat diperoleh jika nilai prioritas dari tiap aktivitas telah diberikan. Metode ini menggunakan *solution encoding scheme* dimana sebuah solusi diwakili dengan rangkaian angka. Posisi dari angka-angka dalam rangkaian menyatakan indeks dari aktivitas dan nilai dari angka-angka tersebut menyatakan prioritas dari aktivitas yang bersesuaian. Nilai prioritas dari solusi awal diperoleh dari membangkitkan secara random antara 0 sampai 1.

2.3 *Firefly Algorithm*

Algoritma (*Algorithm*) merupakan suatu himpunan aturan atau instruksi yang telah dirumuskan dengan baik (*well defined*) untuk memperoleh suatu keluaran khusus (*specific output*) dari suatu masukan khusus (*specific input*) dalam langkah-langkah yang jumlahnya berhingga.

(Chartrand & Oellerman, 1993)

2.3.1 **Inspirasi dan Perilaku Kunang-kunang**

Firefly atau Kunang-kunang, termasuk dalam keluarga *Lampyridae*, kumbang bersayap kecil yang mampu memproduksi *cold light* (cahaya dingin atau sebuah cahaya yang dihasilkan kecil) berkedip untuk menarik pasangan.

(Encyclopedia Britannica, 2011)

Ciri-ciri khas dari kunang-kunang yaitu kilauan cahaya yang dihasilkan dari proses kimia dalam tubuhnya yaitu proses Bioluminisensi (*Bioluminisense*). Bioluminisensi berasal dari kata “bio” yang berarti hidup dan “luminisensi” yang berarti emisi cahaya. Luminisensi adalah fenomena dimana materi memancarkan cahaya dalam kisaran cahaya tampak.

(Holsa, 2009)

Kilauan cahaya pada kunang-kunang dapat berfungsi sebagai sinyal utama untuk menarik pasangan (proses kawin), selain itu kilauan cahaya yang dihasilkan oleh kunang-kunang juga dapat berfungsi untuk memperingatkan kemungkinan adanya bahaya dan melindungi diri dari predator. Untuk menarik pasangan, kunang-kunang jantan akan memberikan sinyal pertama kepada kunang-kunang betina. Sebagai respon sinyal tersebut, kunang-kunang betina

memancarkan cahaya terus menerus atau berkedip. Kunang-kunang yang telah siap untuk proses kawin akan menghasilkan pola sinyal cahaya yang berbeda sebagai tanda bahwa mereka telah siap untuk kawin. Biasanya kunang-kunang akan tertarik pada kunang-kunang lain yang memiliki intensitas cahaya yang lebih cerah. Hal ini juga diketahui bahwa intensitas cahaya pada kunang-kunang bervariasi terhadap jarak dari sumber. Perilaku berkedip kunang-kunang tersebut berfungsi sebagai landasan biologis utama untuk mengembangkan algoritma kunang-kunang (*Firefly Algorithm*).

(Iztok, dkk, 2013)

2.3.2 Definisi *Firefly Algorithm*

Algoritma kunang-kunang merupakan satu dari metode metaheuristik terbaru, yang dikembangkan oleh Dr. Xin-She Yang di Universitas Cambridge pada tahun 2007.

Menurut (Yang, 2010) perumusan *firefly algorithm* adalah sebagai berikut :

1. Semua kunang-kunang itu unisex, jadi suatu kunang-kunang akan tertarik pada kunang-kunang yang lain.
2. Daya tarik sebanding dengan tingkat kecerahan kunang-kunang, kunang-kunang dengan tingkat kecerahan lebih rendah akan tertarik dan bergerak ke kunang-kunang dengan tingkat kecerahan lebih tinggi, kecerahan dapat berkurang seiring dengan bertambahnya jarak dan adanya penyerapan cahaya akibat faktor udara.

3. Kecerahan atau intensitas cahaya kunang-kunang ditentukan oleh nilai fungsi tujuan dari masalah yang diberikan. Untuk masalah maksimisasi, intensitas cahaya sebanding dengan nilai fungsi tujuan.

Untuk masalah maksimisasi, kecerahan hanya sebanding dengan nilai fungsi tujuan. Bentuk lain dari kecerahan dapat didefinisikan dengan cara yang identik dengan fungsi fitness dalam algoritma genetika.

Berikut beberapa istilah yang digunakan dalam Firefly Algorithm (Yang, 2010) :

1. Populasi adalah sebuah kumpulan populasi yang direpresentasikan dengan *fireflies*.
2. *Firefly* adalah individu dalam populasi yang terdiri dari kumpulan symbol atau kode yang merepresentasikan solusi dari permasalahan.
3. Intensitas cahaya adalah nilai untuk mengevaluasi *firefly*.
4. *Attractiveness* (Daya Tarik) adalah daya tarik *firefly* yang dinilai oleh *firefly* lainnya berdasarkan intensitas cahayanya.
5. *Distance* adalah jarak antar 2 *firefly*.
6. *Movement* adalah pergerakan yang dilakukan masing-masing *firefly* menuju *firefly* lain yang intensitas cahayanya lebih terang.

2.3.3 Keatraktifan *Firefly*

Didalam *firefly algorithm* terdapat 2 hal penting yaitu variasi intensitas cahaya dan daya tarik kunang-kunang (*attractiveness*). Untuk mempermudah, dapat diasumsikan bahwa daya tarik kunang-kunang ditentukan oleh kecerahan atau intensitas cahaya yang sebanding dengan nilai fungsi tujuan. Dalam kasus

yang paling sederhana contohnya masalah optimasi maksimum, tingkat intensitas cahaya pada sebuah kunang-kunang x dapat dilihat sebagai,

$$I(x) \approx f(x) \quad (2.5)$$

dengan nilai I merupakan tingkat intensitas cahaya pada x kunang-kunang yang sebanding terhadap solusi fungsi tujuan permasalahan yang akan dicari $f(x)$. Daya tarik β adalah relatif, karena intensitas cahaya yang harus dilihat dan dinilai oleh kunang-kunang lain. Dengan demikian, hasil penilaian akan berbeda tergantung dari jarak antara kunang-kunang yang satu dengan yang lainnya (r_{ij}). Selain itu, intensitas cahaya akan menurun seiring dengan jarak dari sumbernya, dan cahaya juga diserap oleh udara dengan jarak yang berlebih, hal ini akan memungkinkan adanya daya tarik (β) yang bervariasi dengan berbeda-beda tingkat penyerapannya.

Fungsi keatraktifan ialah sebagai berikut :

$$\beta(r) = \beta_0 * e^{-\gamma r^2} \quad (2.6)$$

dengan :

β_0 : nilai keatraktifan kunang-kunang awal

γ : koefisien penyerapan cahaya

r : jarak antar kunang-kunang

(Yang, 2010)

2.3.4 *Distance dan Movement*

Jarak antara dua kunang-kunang i dan j pada x_i dan x_j , masing-masing, adalah sebagai berikut :

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^n (x_{i,k} - x_{j,k})^2}, \quad (2.7)$$

dengan $x_{i,k}$ adalah komponen atau lokus ke-k dari untaian kunang-kunang i .

Dalam kasus 2-dimensi maka,

$$r_{ij} = \sqrt{(x_{i,1} - x_{j,1})^2 + (x_{i,2} - x_{j,2})^2}. \quad (2.8)$$

Pergerakan kunang-kunang i yang tertarik ke kunang-kunang j yang lebih menarik (terang) ditentukan oleh persamaan berikut

$$x_t = x_i + \beta_0 * \exp(-\gamma r_{ij}^2) * (x_j - x_i) + \alpha * (rand - \frac{1}{2}) \quad (2.9)$$

dengan variabel awal x_i menunjukkan posisi awal kunang-kunang yang berada pada lokasi x , kemudian persamaan kedua menyatakan daya tarik *firefly* kepada *firefly* yang memiliki intensitas cahaya yang lebih besar, yang terdiri dari $\beta_0 = 1,0$ parameter ini merupakan nilai keatraktifan awal pada *firefly*, (exp) bilangan eksponensial, parameter $\gamma \in [0, \infty)$ biasa ditentukan antara 0,1 sampai 10 merupakan nilai untuk tingkat penyerapan pada lingkungan sekitar *firefly* yaitu udara dan terakhir r_{ij} merupakan selisih jarak awal antara *firefly* i dan j . Selanjutnya persamaan ketiga terdiri dari selisih nilai solusi pada *firefly* terhadap *firefly* j . Kemudian fungsi persamaan pergerakan *firefly* secara *random* menunjukkan adanya bilangan *random* yang kisarannya antara [0,1]. variabel α yang memiliki kisaran antara [0,1] biasa ditentukan dengan nilai 0,2.

(Broersma, 2010)

2.3.5 Proses *Firefly Alorithm*

Menurut **Yang (2010)**, proses optimasi masalah dengan *Firefly Algorithm* adalah sebagai berikut:

1. Input parameter *Firefly Algorithm*.

2. Membangkitkan secara random populasi awal sebanyak m *fireflies*.
3. Menghitung intensitas cahaya tiap *firefly* $I(x)$ berdasarkan nilai fungsi tujuan $f(x)$.
4. Membandingkan intensitas cahaya tiap *firefly* dengan *firefly* lainnya. Apabila terdapat *firefly* yang intensitas cahayanya lebih besar, lakukan update pergerakan *firefly* menggunakan persamaan *movement*.
5. Menentukan *G-best*. Untuk iterasi pertama, *firefly* terbaik (*firefly* dengan intensitas cahaya terbesar) adalah *G-best*.
6. Membandingkan *firefly* terbaik tiap iterasi dengan *G-best* yang diperoleh. Apabila intensitas cahaya *firefly* terbaik saat itu lebih besar daripada *G-best* maka *firefly* tersebut menjadi *G-best*.
7. Melakukan proses *movement* seperti pada persamaan 2.9 kepada *firefly* terbaik dengan memasukkan nilai $\beta = 0$ dan menggabungkannya dengan *firefly* yang lain untuk menjadi populasi awal pada iterasi selanjutnya.
8. Melakukan proses di atas sampai batas iterasi dipenuhi.

2.4 C++

Berbicara tentang C++ biadsanya tidak lepas dari bahasa C, sebagai bahasa pendahulunya. Pencipta C adalah Brian W. Kerninghan dan Dennis M. Ritchie pada sekitar tahun 1972. C adalah bahasa pemrograman terstruktur, yang membagi program dalam bentuk sejumlah blok. Tujuannya adalah untuk memudahkan dalam pembuatan dan pengembangan program.

C++ diciptakan satu dekade setelah C diciptakan oleh Bjarne Stroustrup, Laboratorium Bell, AT & T, pada tahun 1983. Bahasa ini bersifat kompatibel dengan bahasa pendahulunya, yaitu C. Pada mulanya C++ disebut “a better C”. Nama C++ sendiri diberikan oleh Rick Masciti pada musim panas 1983. Adapun tanda ++ berasal dari nama operator kenaikan bahasa C.

Keistimewaan yang sangat berarti pada C++ adalah karena bahasa ini mendukung pemrograman yang berorientasi obyek. Tujuan utama pembuatan C++ adalah untuk meningkatkan produktivitas pemrograman dalam membuat aplikasi.

2.4.1 Istilah dalam C++

Kelas merupakan struktur data dari obyek. Sebuah contoh yang sangat sederhana dari kelas :

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
```

```
Class nama_kelas
{
  public :
    nama_kelas ();
    ~nama_kelas();

  private :
};

void main ()
{

}

nama_kelas :: nama_kelas ()
{
```

```

}

nama_kelas::~nama_kelas()
{
}

```

Public : menyatakan bahwa yang terdapat di dalamnya bersifat publik atau dapat diakses diluar kelas.

Private : untuk memproteksi anggota-anggota tertentu pada kelas, agar tidak dapat diakses di luar kelas secara langsung.

Fungsi main () : fungsi yang menjadi awal dan akhir eksekusi program C++.

Tanda () : digunakan untuk mengapit argumen fungsi, yaitu nilai yang akan dilewatkan ke fungsi.

Tanda { : menyatakan awal eksekusi program.

Tanda } : merupakan akhir eksekusi program.

Void main () : dipakai untuk menyatakan bahwa fungsi ini tidak memiliki nilai balik.

Pernyataan : instruksi untuk menyuruh komputer melakukan sesuatu. Setiap pernyataan harus diakhiri dengan tanda titik koma.

Cout : sebuah obyek di dalam C++ yang digunakan untuk mengarahkan data ke *standart output*.

#include : salah satu pengarah praprosesor yang tersedia pada C++. Bentuk umum # include bisa berupa # include <namafile> atau # include "namafile".

Clrscr : perintah untuk menghapus layar.

(Kadir, 1995)

