

Implementasi Program Dinamis pada Pencarian *Critical Path* dalam Metode Manajemen Proyek *Critical Path Method*

Hasna Roihan Nafiisah 13518008
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13518008@std.stei.itb.ac.id

Abstract—*Critical Path Method* adalah salah satu metode manajemen proyek berbasis analisis matematika. Metode ini biasa digunakan pada proyek-proyek yang memiliki daftar pekerjaan yang independen satu sama lain dalam pelaksanaannya serta memiliki durasi yang diperkirakan. Metode ini akan menghasilkan sebuah *critical path* atau rute pekerjaan yang memiliki durasi paling lama sebagai prediksi waktu minimal suatu proyek dapat diselesaikan. Program dinamis adalah algoritma yang biasa digunakan untuk persoalan optimisasi dan dapat memberikan lebih dari satu solusi. Algoritma ini dapat diimplementasikan untuk mencari satu atau lebih *critical path* pada suatu proyek untuk kemudian dievaluasi lebih lanjut oleh manajer proyek,

Kata Kunci—*Critical Path Method*, Program Dinamis, project scheduling

I. PENDAHULUAN

Dalam merencanakan suatu proyek, dibutuhkan sistem dan teknik penjadwalan yang tepat untuk membantu mengatur pelaksanaan setiap *milestone* atau aktivitas yang diperlukan untuk menyelesaikan proyek tersebut. Pengaturan ini digunakan untuk dapat mengintegrasikan keterbatasan waktu, sumber daya, serta memudahkan dalam merencanakan proyek-proyek yang besar dan kompleks.

Salah satu teknik koordinasi yang dapat digunakan untuk merencanakan penjadwalan proyek adalah *Critical Path Method*, atau biasa disingkat sebagai CPM. *Critical Path Method* adalah salah satu teknik yang menggunakan analisis matematika, bersama dengan metode PERT.

Critical Path Method pertama kali dikembangkan oleh Angkatan Laut AS pada tahun 1950^[1]. Metode ini digunakan untuk mencari durasi terpanjang suatu proyek dapat dikerjakan dari awal mulai hingga akhir proyek. Dengan metode ini, manajer proyek dapat menentukan *critical path* dari proyek tersebut dan mengatur alokasi sumber daya serta waktu dengan lebih efektif, agar pengerjaan proyek tidak memakan waktu yang terlalu lama akibat penundaan suatu *milestone* dari proyek tersebut. *Critical path* tersebut juga dapat menentukan skala prioritas *milestone*.

Metode CPM ini biasa digunakan pada proyek-proyek yang memiliki kegiatan yang berkesinambungan dan bercabang. Maksud berkesinambungan di sini adalah, untuk melaksanakan satu kegiatan, harus ada kegiatan lain yang sudah selesai terlebih dahulu. Misal, pada proyek-proyek seperti pembangunan rumah, konstruksi sipil, proses manufaktur, serta proyek-proyek lain yang memiliki prosedur bertahap. Selain berkesinambungan, proyek yang dianalisis juga harus merupakan proyek yang durasinya dapat dideskripsikan secara kuantitatif, karena CPM adalah teknik yang menggunakan analisis matematika.

Program dinamis dapat diterapkan pada metode CPM untuk menemukan *critical path* pada proyek tersebut. Program dinamis dapat memberikan hasil yang optimal, serta memberikan himpunan solusi untuk dipilih oleh manajer proyek. Program dinamis tidak terpaku pada satu solusi, sehingga memudahkan manajer proyek untuk mempertimbangkan *critical path* di dalam himpunan solusi dengan nilai-nilai kualifikasi yang belum bisa dilakukan oleh komputer. *Critical path* yang dipilih kemudian dapat diolah lebih lanjut oleh manajer proyek.

II. LANDASAN TEORI

A. Program Dinamis

Program dinamis adalah algoritma yang ditemukan oleh seorang matematikawan asal Amerika Serikat bernama Richard Bellman, pada tahun 1950. Program dinamis ini adalah sebuah metode optimisasi pencarian solusi dengan memecah masalah menjadi beberapa tahapan masalah yang lebih kecil^[2].

Program dinamis biasa digunakan pada masalah-masalah yang memiliki sub-masalah yang saling tumpang tindih karena hubungan rekurens sub-masalah tersebut dengan sub-masalah lain yang lingkungannya lebih kecil. Program dinamis tidak menyelesaikan sub-masalah yang tumpang tindih tersebut berulang kali, tapi hanya menyelesaikan sekali kemudian mencatat hasilnya dalam sebuah

tabel, yang kemudian digunakan dalam menemukan solusi untuk masalah yang lebih besar.

Penyelesaian solusi masalah secara bertahap dari sub-sub masalah ini kemudian menghasilkan serangkaian solusi yang saling berkaitan. Dikutip dari Rinaldi Munir (2018), persoalan yang dapat diselesaikan dengan program dinamis adalah persoalan yang memiliki karakteristik sebagai berikut:

1. Terdapat sejumlah berhingga pilihan yang mungkin.
2. Persoalan dibagi menjadi beberapa tahap, dan pada setiap tahap hanya diambil satu keputusan.
3. Setiap tahap memiliki beberapa status yang berkaitan dengan tahap tersebut. Status adalah masukan-masukan yang mungkin ada pada tahap tersebut.
4. Hasil keputusan yang diambil pada setiap tahap ditransformasikan dari status yang berkaitan ke status berikutnya pada tahap berikutnya.
5. *Cost* pada setiap tahap meningkat secara bertahap dan teratur juga, sesuai dengan jumlah pembagian tahap.
6. *Cost* pada suatu tahap bergantung pada *cost* di tahap sebelumnya.
7. Keputusan terbaik suatu tahap bersifat independen terhadap keputusan pada tahap sebelumnya.
8. Hubungan rekursif mengakibatkan keputusan terbaik pada tiap status pada tahap k memberikan keputusan terbaik pada tiap status pada tahap $k+1$.
9. Prinsip optimalitas berlaku pada persoalan tersebut.

Program dinamis dan algoritma Greedy adalah dua algoritma yang sama-sama mencari solusi secara bertahap. Namun perbedaannya, algoritma Greedy hanya memberikan satu buah solusi, sementara program dinamis memberikan himpunan solusi sehingga solusi yang diberikan lebih beragam^[3].

Pendekatan program dinamis dibagi menjadi dua, yaitu pendekatan secara maju dan pendekatan mundur. Misalkan tahap dari awal hingga akhir diurutkan dengan $1, 2, 3, \dots, n$, maka program dinamis dengan pendekatan maju akan dimulai dari tahap ke-1, tahap ke-2, dan seterusnya hingga tahap ke- n . Sementara program dinamis dengan pendekatan mundur akan bergerak sebaliknya, yaitu dari tahap ke- n , tahap ke- $n-1$, dan seterusnya hingga tahap ke-1.

B. Prinsip Optimalitas

Program dinamis adalah algoritma yang biasa digunakan untuk mengatasi persoalan-persoalan optimisasi. Oleh sebab itu, dalam program dinamis

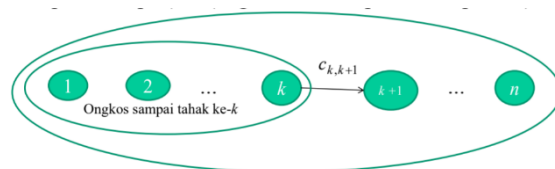
ini terdapat sebuah prinsip yang berlaku, yang disebut sebagai Prinsip Optimalitas.

Prinsip Optimalitas berbunyi sebagai berikut: solusi optimal dari suatu tahap dalam suatu masalah adalah gabungan dari solusi-solusi optimal dari tahap-tahap sebelumnya^[2].

Prinsip Optimalitas ini menjelaskan bahwa setiap solusi optimal pada k berarti solusi pada tahap ke- $k-1$ adalah tahap optimal. Ini juga berarti, jika hendak mencari solusi optimal pada tahap ke- $k+1$, maka bisa memakai solusi optimal dari tahap ke- k , sehingga tidak perlu mencari solusi dari tahap awal lagi.

Prinsip Optimalitas pada program dinamis dengan dua pendekatan adalah sebagai berikut^[3]:

1. Dengan pendekatan maju, *cost* ke- $k+1 = cost$ ke- $k + cost$ dari tahap- k ke tahap $k+1$.



Gambar 2.1 Prinsip Optimalitas

Sumber: Diktat Kuliah IF2211 Strategi Algoritma, Rinaldi Munir, 2018

2. Dengan pendekatan mundur, *cost* ke- $k = cost$ ke- $k-1 + cost$ dari tahap $k+1$ ke tahap ke- k .

C. Langkah Pengembangan Algoritma Program Dinamis pada Pencarian Lintasan Terpendek

Berikut adalah langkah-langkah untuk mengembangkan algoritma program dinamis^[3]:

1. Mengarakteristikan struktur solusi optimal.
2. Mendefinisikan secara rekursif nilai solusi optimal.
3. Menghitung nilai solusi optimal secara maju atau mundur, tergantung pendekatan yang dipakai.
4. Mengonstruksikan solusi optimal.

Misalkan terdapat sebuah graf berarah yang berawal di simpul-1 dan berakhir di simpul- n . Kemudian, graf akan dibagi menjadi beberapa tahap, dari tahap 1 hingga tahap m . Misalkan x_1, x_2, \dots, x_m adalah simpul-simpul yang dikunjungi pada tahap ke- k , dengan $k = 1, 2, \dots, m$. Status s adalah simpul-simpul di dalam graf.

Relasi rekurens untuk menyatakan lintasan terpendek dari status s ke x_m dijabarkan dalam fungsi berikut:

$$\text{Basis: } f_1(s) = c_{x_1s}$$

$$\text{Rekurens: } f_k(s) = \min_{x_k} \{ c_{sx_k} + f_{k-1}(x_k) \}$$

$$\text{Dengan } k = 2, 3, \dots, m.$$

Program dinamis kemudian akan menggunakan fungsi-fungsi tersebut di atas dalam sebuah tabel untuk menentukan fungsi optimal.

D. Critical Path Method

Critical Path Method (CPM) adalah salah satu metode manajemen proyek berbasis analisis matematika. Metode ini pertama kali dikembangkan oleh Angkatan Laut Amerika Serikat pada tahun 1950. Metode ini menentukan rute terlama yang berisi kegiatan atau aktivitas yang bersifat *critical* dari awal dimulai proyek hingga akhir proyek^[1].

Dikutip dari Harvard Business Review^[4], beberapa karakteristik penting suatu proyek untuk dapat dianalisis oleh *Critical Path Method* adalah sebagai berikut:

1. Proyek terdiri atas kumpulan pekerjaan yang terdefinisi secara jelas, yang ketika selesai maka menandakan akhir dari proyek tersebut.
2. Setiap pekerjaan mulai dan berakhir secara independen dari pekerjaan yang lain.
3. Setiap pekerjaan dapat disusun secara terurut, dengan menggunakan Fulkerson rule.

Fulkerson's Rule

Fulkerson's rule adalah sebuah aturan dalam memberi nomor atau mengurutkan peristiwa maupun pekerjaan. Prosedur dalam menerapkan Fulkerson's rule adalah sebagai berikut:

1. Setiap graf memiliki satu pekerjaan yang menginisialisasi. Pekerjaan ini tidak memiliki pekerjaan pendahulu (*immediate predecessor*), tapi memiliki pekerjaan yang dilakukan setelahnya.
2. Urutkan pekerjaan-pekerjaan yang dikerjakan setelah pekerjaan inisialisasi. Beri nomor secara sekuensial.
3. Ulangi prosedur 1 dan 2 sampai pekerjaan terakhir yang tidak lagi memiliki pekerjaan setelahnya.

Konsep dari CPM adalah menyusun pekerjaan atau *milestone* dari proyek dalam sebuah graf berarah. Pertama, setiap pekerjaan harus didefinisikan secara terpisah, diidentifikasi dengan penomoran unik (misal dengan nomor atau huruf). Setiap pekerjaan juga harus didefinisikan secara jelas waktu yang dibutuhkan untuk menyelesaikan pekerjaan tersebut, serta *predecessors* dari pekerjaan tersebut, jika ada. Setelah itu, pekerjaan-pekerjaan ini diurutkan dan dimasukkan ke dalam graf sesuai Fulkerson's rule di atas.

Graf yang dihasilkan dari daftar pekerjaan itu kemudian akan menghasilkan beragam rute yang mengikuti alur panah. Jumlah waktu yang dibutuhkan untuk menyelesaikan rute tersebut adalah jumlah durasi dari setiap pekerjaan yang dilewati oleh rute tersebut.

Rute yang memiliki jumlah waktu terbanyak adalah

critical path, dan waktu menelusuri rute tersebut adalah waktu minimal suatu proyek dapat selesai. Pekerjaan-pekerjaan yang dilewati rute tersebut adalah pekerjaan-pekerjaan yang bersifat *critical*.

Pekerjaan yang bersifat *critical* berarti pekerjaan tersebut memiliki prioritas yang lebih tinggi terhadap alokasi sumber daya maupun alokasi waktu. Sementara pekerjaan yang bersifat *non-critical* berada di rute yang *non-critical* dan dapat diberikan *slack* atau waktu *delay*.

Project Network Diagram

Diagram ini merepresentasikan graf pekerjaan-pekerjaan secara visual. Terdapat dua jenis diagram^[1]:

1. Pekerjaan pada panah. Diagram jenis ini merepresentasikan tiap pekerjaan pada panah, sehingga simpul pangkal menunjukkan awal mulai, dan simpul ujung menunjukkan akhir pekerjaan. Diagram jenis ini membutuhkan *dummy job* untuk menyambungkan *predecessors*.
2. Pekerjaan pada simpul. Diagram jenis ini merepresentasikan tiap pekerjaan pada simpul, serta durasinya. Diagram ini tidak membutuhkan *dummy job*.

III. IMPLEMENTASI ALGORITMA PROGRAM DINAMIS PADA PENCARIAN *CRITICAL PATH*

A. Persiapan Data

Pertama, disiapkan terlebih dahulu data pekerjaan-pekerjaan yang akan dilaksanakan di dalam proyek tersebut. Proyek harus dipastikan memiliki pekerjaan-pekerjaan dengan kriteria seperti yang sudah disebutkan di dalam bab 2, yaitu memiliki durasi yang dapat ditetapkan, independen satu sama lain, dan berkesinambungan.

Pada contoh implementasi kali ini, akan digunakan contoh proyek untuk melakukan *branding* publikasi sebuah acara. Adapun daftar pekerjaan, durasi, dan *immediate predecessors* tertera di tabel di bawah ini:

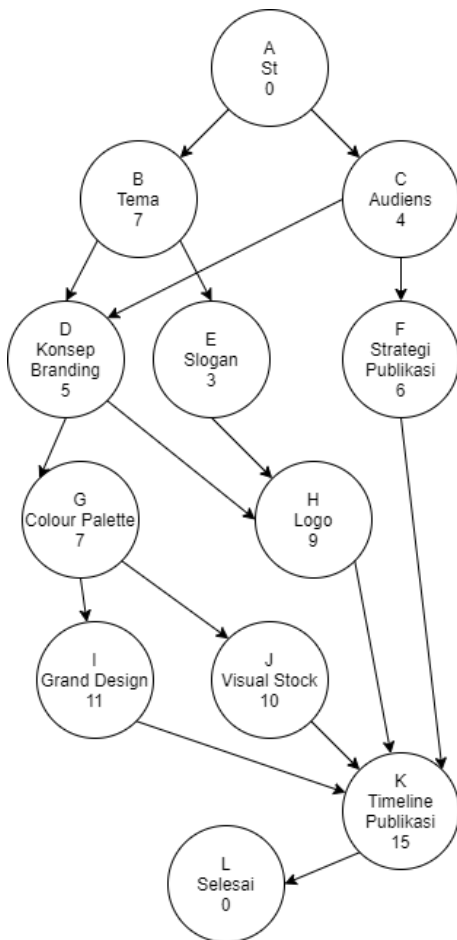
ID Pekerjaan	Deskripsi	<i>Immediate Predecessors</i>	Durasi (hari)
A	Mulai	-	0
B	Penentuan tema	A	7
C	Segmentasi audiens	A	4
D	Penentuan konsep <i>branding</i>	B, C	5
E	Pembuatan Slogan	B	3
F	Penyusunan strategi publikasi	B	6

G	Penentuan <i>colour palette</i>	D	7
H	Pembuatan logo	D, E	9
I	Panduan <i>grand design</i>	G	11
J	Pembuatan <i>visual stock</i>	G	10
K	Penyusunan Timeline Publikasi	F, H, I, J	15
L	Selesai	K	0

Tabel 3.1 Daftar Pekerjaan sesuai Fulkerson's Rule

Dari tabel di atas, dapat dilihat bahwa pekerjaan sudah dituliskan berurut sesuai aturan Fulkerson. Kemudian, daftar pekerjaan ini akan dimasukkan ke dalam graf untuk memberikan representasi visual yang lebih mudah dibaca dan dipahami. Pada implementasi kali ini, akan digunakan diagram yang merepresentasikan tiap pekerjaan dengan simpul, untuk mempermudah visualisasi dan mengurangi *dummy job*.

Graf dari tabel di atas adalah sebagai berikut:



Gambar 3.1 Diagram Alur Kerja

Diagram di atas merupakan graf berarah dari tabel 3.1. Pekerjaan dimulai dari pekerjaan A dengan durasi 0, dan berakhir di pekerjaan L dengan durasi 0.

B. Identifikasi Masalah

Pada persoalan *critical path* ini, kasus mirip dengan persoalan lintasan terpendek dari suatu simpul ke simpul yang lain. *Critical path* mencari rute dengan jumlah durasi terlama, dimulai dari simpul mulai dan berakhir di simpul selesai.

Oleh karena itu, fungsi pada kasus lintasan terpendek dapat dimodifikasi untuk menyelesaikan persoalan *critical path*. Hasil modifikasi fungsi rekurens adalah sebagai berikut:

Dalam setiap skema pencarian, selalu ada kasus terbaik dan kasus terburuk. Jika data yang dimasukkan menghasilkan BST yang seimbang, maka:

$$\text{Basis: } f_1(s) = c_{x1s}$$

$$\text{Rekurens: } f_k(s) = \max_{xk} \{ c_{xks} + f_{k-1}(x_k) \}$$

Dengan fungsi tersebut, maka graf dapat ditelusuri dengan program dinamis untuk mencari rute dengan durasi terpanjang sebagai *critical path*.

C. Langkah-Langkah Penerapan Program Dinamis

Langkah pertama untuk menerapkan algoritma program dinamis pada graf berarah yang sudah ada adalah dengan membagi persoalan menjadi beberapa tahap penyelesaian. Berikut adalah hasil pembagian tahap dari graf pada gambar 3.1:

Gambar 3.2 Diagram Alur Kerja bertahap

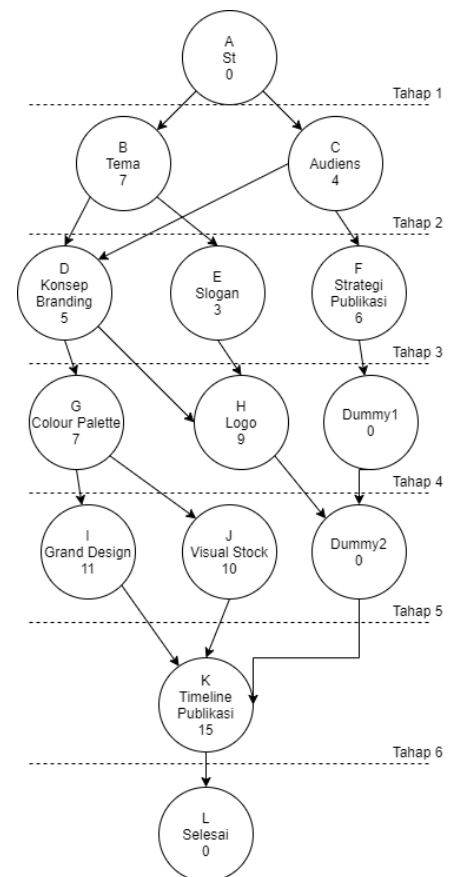


Diagram di atas adalah diagram yang sudah dibagi menjadi beberapa tahap untuk kemudian diproses dalam program dinamis. Dapat dilihat bahwa ditambahkan dua *dummy job* di dalam diagram. Hal ini dilakukan agar kesinambungan alur tidak terputus ketika dibagi ke dalam tahap-tahap, serta memudahkan analisis dengan program dinamis.

Berikutnya, pemrosesan dengan program dinamis. Pada contoh kali ini, program dinamis akan menggunakan pendekatan maju. Sehingga, pemrosesan program dinamis akan dimulai dari tahap 1. Ongkos (*cost*) yang dipakai adalah durasi yang tertera pada simpul yang akan dituju. Pada fungsi rekurens, pekerjaan terkait akan ditambahkan ke rute yang rute terakhirnya ada di solusi optimal.

Tahap 1 – Basis

s	Solusi Optimum	
	$f1(s)$	$x1^*$
B	7	A
C	4	A

Tabel 3.2 Tahap 1

Pada tahap 1, fungsi basis menghasilkan rute {(A->B), (A->C)} dengan nilai maksimum masing-masing adalah 7 dan 4.

Tahap 2

s	$f2(x2,s)$		Solusi Optimum	
	B	C	$f2(s)$	$x2^*$
D	$7+5 = 12$	$4+5 = 9$	12	B
E	$7+3 = 10$	-1	10	B
F	-1	$4+6 = 10$	10	C

Tabel 3.3 Tahap 2

Pada tahap 2, fungsi rekurens menghasilkan rute {(A->B->D), (A->B->E), (A->C->F)} dengan nilai maksimum masing-masing adalah 12, 10, dan 10. Pada rute yang tidak memiliki hubungan pada graf berarah, seperti B->F, maka program akan otomatis memberikan nilai -1.

Tahap 3

s	$f3(x3,s)$			Solusi Optimum	
	D	E	F	$f3(s)$	$x3^*$
G	$12+7 = 19$	0	0	19	D
H	$12+9 = 21$	$10+9 = 19$	0	21	D
D1	-1	-1	$10+0 = 10$	10	F

Tabel 3.4 Tahap 3

Pada tahap 3, fungsi rekurens menghasilkan rute {(A->B->D->G), (A->B->D->H), (A->C->F->D1)} dengan nilai maksimum masing-masing adalah 19, 21, dan 10.

Tahap 4

s	$f4(x4,s)$			Solusi Optimum	
	G	H	D1	$f4(s)$	$x4^*$
I	$19+11 = 30$	-1	-1	30	G
J	$19+10 = 29$	-1	-1	29	G
D2	-1	$21+0 = 21$	$10+0 = 10$	21	H

Tabel 3.5 Tahap 4

Pada tahap 4, fungsi rekurens menghasilkan rute {(A->B->D->G->I), (A->B->D->G->J), (A->B->D->H->D2)} dengan nilai maksimum masing-masing adalah 30, 29, dan 21.

Tahap 5

s	$f5(x5,s)$			Solusi Optimum	
	I	J	D2	$f4(s)$	$x5^*$
K	$30+15 = 45$	$29+15 = 44$	$21+15 = 36$	45	I

Tabel 3.6 Tahap 5

Pada tahap 5, fungsi rekurens menghasilkan rute {(A->B->D->G->I->K)} dengan nilai maksimum masing-masing adalah 45.

Tahap 6

s	Solusi Optimum	
	K	$x6^*$
L	$45+0 = 45$	K

Tabel 3.7 Tahap 6

Pada tahap 5, fungsi rekurens menghasilkan rute {(A->B->D->G->I->K->L)} dengan nilai maksimum masing-masing adalah 45.

Dari penelusuran dengan program dinamis di atas, didapat himpunan solusi rute dengan durasi terlama adalah {(A->B->D->G->I->K->L)}. Pada himpunan solusi hanya terdapat satu solusi, yaitu *critical path* dengan pekerjaan yang bersifat *critical* sebagai berikut:

- A – Mulai – 0 hari
- B – Penentuan tema – 7 hari
- D – Penentuan konsep *branding* – 5 hari
- G – Penentuan *colour palette* – 7 hari
- I – Panduan *grand design* – 11 hari
- K – Penyusunan *timeline* publikasi – 15 hari
- L – Selesai – 0 hari

Total durasi dari keseluruhan *critical path* adalah 45 hari.

Implementasi pada proyek lain memungkinkan adanya lebih dari satu solusi di dalam himpunan solusi, sehingga manajer proyek dapat mengevaluasi *critical path* yang mana yang lebih memungkinkan untuk diterapkan, dengan mempertimbangkan kondisi lapangan serta ketersediaan sumber daya.

Selanjutnya, *critical path* yang sudah ditentukan dapat diolah lebih lanjut untuk keberjalanan perencanaan penjadwalan proyek yang ada.

IV. KESIMPULAN DAN SARAN

Program dinamis adalah sebuah algoritma yang mampu menyelesaikan persoalan-persoalan optimisasi. Algoritma ini dapat memberikan lebih dari satu solusi optimal yang mungkin pada suatu persoalan. Hal ini membuat program dinamis lebih tepat digunakan untuk persoalan-persoalan yang membutuhkan lebih dari satu solusi, daripada menggunakan algoritma Greedy yang belum tentu memberikan solusi optimal.

Critical Path Method adalah sebuah teknik pengaturan perencanaan suatu proyek. Metode ini digunakan untuk mencari durasi terlama yang mungkin untuk mengerjakan suatu proyek dari mulai hingga selesai. Program dinamis tepat digunakan untuk mencari *critical path* pada proyek, karena bisa menghasilkan lebih dari satu solusi untuk dievaluasi oleh manajer proyek terkait kondisi lapangan dan ketersediaan sumber daya.

Untuk selanjutnya, konsep pencarian *critical path* pada suatu proyek dengan menggunakan program dinamis ini diharapkan dapat:

1. Digunakan bagi manajer proyek untuk membantu mengevaluasi *critical path* yang dihasilkan.
2. Konsep implementasi ini juga dapat diterapkan pada metode manajemen proyek lainnya seperti PERT, atau bentuk penjadwalan lainnya seperti pada *Gantt chart*.
3. Dikembangkan sehingga program yang ada dapat memilih rute dengan mempertimbangkan prioritas yang mungkin diterapkan pada pekerjaan-pekerjaan di dalam proyek

V. LINK VIDEO YOUTUBE

Penjelasan lebih sederhana terkait makalah ini dapat diakses di situs Youtube. Link video adalah sebagai berikut:

<https://youtu.be/T3ERxIpo0Pg>

VI. UCAPAN TERIMA KASIH

Terima kasih yang sebesar-besarnya penulis ucapkan kepada Allah swt. Tuhan Seluruh Alam, yang telah melimpahkan rahmat-Nya sehingga penulis dapat menyelesaikan makalah ini dengan tepat waktu. Tak lupa penulis ucapkan juga terima kasih kepada dosen mata kuliah IF2211 Strategi Algoritma yang telah meluangkan waktunya untuk mengajarkan ilmu yang bermanfaat tersebut. Penulis juga berterima kasih kepada Panitia Pelaksana Program Ramadhan dan Idul Adha Salman ITB 1441 H yang telah berkontribusi besar dalam menginspirasi saya selama penyusunan makalah ini.

REFERENCES

- [1] *Critical Path Method (CPM): A Coordinating Tool*. Nisha Bishnoi. 2018 .1,s.l.: IRJMST,2018, International Research Journal of Management Sciences & Technology ,Vol.9 www.IRJMS.com.
- [2] Levitin, Anany. 2011. Introduction to The Design and Analysis of Algorithms. London: Pearson
- [3] Munir, Rinaldi. Diktat Kuliah IF2211 Strategi Algoritma. Program Studi Teknik Informatika ITB. 2018.
- [4] Levy, F. K., Thompson, G. L. and Wiest, J. D. (1963, September). The ABCs of The Critical Path Method. *Harvard Business Review*. Diakses dari <https://hbr.org/1963/09/the-abcs-of-the-critical-path-method>.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 3 Mei 2020



Hasna Roihan Nafiisah 13518008