

Analisis waktu dan biaya percepatan proyek dengan menggunakan algoritma *crash time limit* (CTL)

Erni Setyowati ¹, Sapti Wahyuningsih ², Lucky Tri Oktoviana ³
Universitas Negeri Malang

Email: erniset12@yahoo.co.id; sapti.wahyuningsih.fmipa@um.ac.id;
lucky.tri.fmipa@um.ac.id

ABSTRAK: Algoritma *Crash Time Limit* (CTL) merupakan salah satu algoritma yang dapat diterapkan dalam mempercepat waktu penyelesaian proyek. Pada proses percepatan waktu proyek, Algoritma *Crash Time Limit* (CTL) membutuhkan data berupa kegiatan, kegiatan pendahulu, biaya normal, biaya *crash*, waktu normal, waktu *crash*, dan biaya tidak langsung. Setelah memperoleh data tersebut, proses yang dilakukan adalah memilih kegiatan kritis dengan biaya *cost-slope* terendah sebagai kegiatan yang akan dipercepat, menentukan kegiatan yang dipercepat dengan memanfaatkan lintasan dalam jaringan, dan menentukan waktu percepatan dengan menggunakan nilai *crash time limit*. Hasil yang diperoleh dari proses penyelesaian percepatan proyek dengan Algoritma *Crash Time Limit* (CTL) yaitu waktu penyelesaian proyek dapat diselesaikan lebih cepat dengan biaya yang lebih minimum.

Kata Kunci: *Crash Time Limit*, *crashing*, Percepatan Proyek, Lintasan

ABSTRACT: Crash Time Limit (CTL) Algorithms is one of the algorithms that can be applied to accelerate project completion time. In the process of acceleration project time, Crash Time Limit (CTL) Algorithms require data such as activity, activity predecessor, normal time, crash time, normal cost, crash cost, and indirect costs. After obtaining such data, the process is choose critical activities with least of the cost of cost-slope as activity accelerated, determine the activities that are accelerated by utilizing the path in the network, and determine acceleration time with value of crash time limit. The results obtained from the settlement process with Crash Time Limit (CTL) Algorithms is project completion time can solve more quickly with minimum cost.

Key Words: *Crash Time Limit*, *crashing*, project acceleration, path

Penjadwalan proyek merupakan salah satu permasalahan yang dapat diselesaikan dengan menggunakan teori graph. Salah satu metode penjadwalan yang berkaitan dengan teori graph adalah *network planning* (jaringan kerja). Metode *network planning* (jaringan kerja) merepresentasikan kegiatan dan hubungan ketergantungan antar kegiatan dengan menggunakan titik dan panah atau sisi berarah. Representasi jaringan tersebut dapat menunjukkan hubungan ketergantungan antar kegiatan dalam proyek melalui panah atau sisi berarah. Dalam penerapannya, penjadwalan proyek berguna untuk memperkirakan waktu penyelesaian proyek. Akan tetapi pada prosesnya, terkadang suatu proyek harus

-
1. Erni Setyowati adalah mahasiswa jurusan Matematika FMIPA Universitas Negeri Malang
 2. Sapti Wahyuningsih adalah dosen jurusan Matematika FMIPA Universitas Negeri Malang
 3. Lucky Tri Oktoviana adalah dosen jurusan Matematika FMIPA Universitas Negeri Malang

diselesaikan lebih cepat dari waktu yang telah diperkirakan karena harus mengejar suatu target atau kegiatan tertentu serta untuk mengantisipasi adanya suatu keterlambatan proyek. Beberapa hal yang dapat dilakukan untuk mempercepat waktu penyelesaian proyek adalah dengan menambah sumber daya atau jam lembur yang memungkinkan terjadinya peningkatan anggaran biaya proyek. Untuk itu dibutuhkan alternatif penyelesaian masalah percepatan proyek yang dapat menghasilkan biaya proyek yang minimum.

Algoritma *Crash Time Limit* (CTL) merupakan salah satu algoritma yang dapat dimanfaatkan untuk menyelesaikan permasalahan percepatan proyek. Algoritma *Crash Time Limit* (CTL) tersebut dikembangkan oleh Kharde & Patil (2012). Menurut Kharde & Patil (2012) *Unit Time Method* (UTM) merupakan salah satu metode *crashing* yang selalu memberikan hasil optimum akan tetapi membutuhkan proses iterasi yang lebih banyak karena pada setiap proses percepatan tiap iterasinya hanya mempercepat satu satuan waktu proyek. Untuk itu, Kharde & Patil (2012) mengembangkan Algoritma *Crash Time Limit* (CTL) yang menggunakan prinsip *Unit Time Method* (UTM) tetapi memiliki jumlah iterasi yang lebih sedikit. Selain itu, karena dalam proses perhitungan waktu proyek dengan algoritma ini memanfaatkan panjang lintasan, maka tidak diperlukan perhitungan maju mundur.

Dalam artikel ini, akan dikaji mengenai Algoritma *Crash Time Limit* (CTL) yang diterapkan pada dua contoh. Hasil penerapan pada contoh pertama akan dibandingkan dengan hasil yang diperoleh Zaenalzadeh (2011) dalam *Australian Journal of Basic and Applied Sciences* yang menggunakan program Lingo 12.0 untuk menyelesaikan model matematika permasalahan percepatan proyek. Hasil penerapan contoh kedua akan dibandingkan dengan hasil yang diperoleh Shanmugasundaram & Kumar (2015) dalam *International Journal of in Multidisciplinary and Academic Research* yang menggunakan metode *crashing* optimum untuk menyelesaikan permasalahan percepatan proyek.

PERCEPATAN PROYEK

Dalam proses penyelesaian percepatan proyek, terdapat beberapa komponen yang menjadi pertimbangan yaitu lintasan kritis, kekritisian lintasan, waktu proyek, biaya proyek, dan hubungan antara waktu dan biaya proyek. Lintasan kritis merupakan lintasan dengan panjang maksimum dan membutuhkan waktu penyelesaian paling lama dibandingkan lintasan lain yang terdapat pada suatu jaringan kerja (*network planning*) (Tsui, 2004: 291-292). Penentuan panjang lintasan diperoleh dengan menjumlahkan waktu penyelesaian kegiatan yang terdapat dalam lintasan (Hiller & Lieberman, 2001:475). Dalam proses mempercepat kegiatan, lintasan kritis awal maupun lintasan kritis yang telah terbentuk sebelumnya harus tetap dipertahankan (Uher dan Zantis, 2011: 121). Selain itu, dalam proses mempercepat waktu penyelesaian kegiatan, waktu penyelesaian kegiatan paling cepat adalah hingga waktu *crash*, sedangkan waktu penyelesaian paling lambat adalah hingga waktu normal (Zeinalzadeh, 2011). Biaya dalam proyek terdiri dari biaya langsung dan biaya tidak langsung (Kasana & Kumar, 2004: 284). Biaya langsung merupakan biaya yang berhubungan dengan kegiatan dalam proyek secara individual sedangkan biaya tidak langsung merupakan biaya yang berhubungan dengan pengawasan, administrasi, asuransi, atau perijinan. Dengan mempercepat waktu penyelesaian proyek, biaya langsung akan meningkat, biaya tidak langsung akan menurun, dan diantara waktu percepatan yang diperoleh akan dipilih penyelesaian proyek dengan total biaya yang lebih minimum (Sen, 2010: 214).

PEMBAHASAN

Algoritma *Crash Time Limit* (CTL)

Tujuan penyelesaian permasalahan percepatan proyek dengan Algoritma *Crash Time Limit* (CTL) adalah untuk mempercepat waktu penyelesaian proyek dengan biaya yang minimum atau dapat dinyatakan sebagai berikut,

$$\text{Minimum } TC = ID + DC = (CPL \times ID_{pt}) + \sum NC + \sum(CTL_A \times CS_A).$$

Beberapa kendala yang perlu diperhatikan dalam penyelesaian permasalahan dengan Algoritma *Crash Time Limit* (CTL) adalah sebagai berikut.

- (i) Kegiatan dapat diselesaikan paling cepat hingga waktu *crash* dan paling lama diselesaikan dengan waktu normal.

$$CT_A \leq t_A \leq NT_A$$

- (ii) Lintasan NP merupakan lintasan terpanjang kedua setelah lintasan kritis dan lintasan NNP merupakan lintasan terpanjang ketiga setelah lintasan kritis.

$$NNPL < NPL < CPL$$

- (iii) Nilai waktu, biaya, dan panjang lintasan adalah nonnegatif.

$$NT_A \geq 0, CT_A \geq 0, NNPL \geq 0, NPL \geq 0, CPL > 0, ID \geq 0, DC \geq 0$$

dengan keterangan yaitu,

TC	:= Total Biaya Proyek	CT_A	:= Waktu <i>crash</i> kegiatan A
$IDpt$:= Biaya tidak langsung tiap satuan waktu	t_A	:= Waktu penyelesaian kegiatan A
ID	:= Biaya tidak langsung	NT_A	:= Waktu normal kegiatan A
DC	:= Biaya Langsung	CPL	:= Panjang lintasan kritis
NC	:= Biaya normal	NPL	:= Lintasan terpanjang pertama setelah lintasan kritis
CTL_A	:= Nilai <i>crash time limit</i> kegiatan A		
CS_A	:= Biaya <i>cost-slope</i> kegiatan A	$NNPL$:= Lintasan terpanjang kedua setelah lintasan kritis

Dalam menyelesaikan permasalahan percepatan proyek dengan Algoritma *Crash Time Limit* (CTL), diperlukan beberapa pertimbangan komponen berikut ini.

1. *Critical Path Float Limit*

Critical path float limit merupakan selisih antara panjang lintasan kritis dengan lintasan non kritis. Pada Algoritma *Crash Time Limit* (CTL), terdapat dua jenis *critical path float limit* yang digunakan yaitu NF dan F dengan definisi sebagai berikut,

$$F = CPL - NPL$$

$$NF = CPL - NNPL$$

2. *Crash Time Limit*

Crash time limit merupakan batas waktu maksimum percepatan suatu kegiatan pada satu iterasi. Terdapat dua jenis *crash time limit* yang digunakan pada Algoritma *Crash Time Limit* (CTL) yaitu CTL_A dan $NCTL_A$ dengan definisi seperti berikut (Kharde & Patil, 2012).

$$CTL_A = \min \{F, \Delta t_A\}$$

$$NCTL_A = \min \{NF, \Delta t_A\}$$

Pada persamaan tersebut, Δt_A merupakan selisih antara waktu normal dan waktu *crash* kegiatan A. Apabila kegiatan kritis yang dipercepat terletak pada lintasan CP dan NP, maka nilai *crash time limit* yang digunakan adalah $NCTL_A$, sedangkan apabila tidak pada lintasan CP dan NP maka nilai *crash time limit* yang digunakan adalah CTL_A .

Langkah-langkah Algoritma *Crash Time Limit* (CTL)

Langkah-langkah penyelesaian percepatan proyek dengan Algoritma *Crash Time Limit* (CTL) adalah sebagai berikut.

1. Menyiapkan data dan menentukan biaya *cost-slope* masing-masing kegiatan dengan menggunakan persamaan berikut,

$$CS_A = \frac{CC_A - NC_A}{NT_A - CT_A} = \frac{CC_A - NC_A}{\Delta t}$$

2. Menyiapkan tabel iterasi untuk mencatat hasil yang diperoleh tiap iterasinya berupa kegiatan yang dipercepat, waktu percepatan kegiatan, biaya percepatan, biaya langsung, biaya tidak langsung, dan total biaya.
3. Menentukan representasi jaringan dan mendaftar lintasan beserta panjangnya untuk mengetahui waktu proyek.
4. Menentukan waktu proyek dan total biaya proyek pada kondisi normal.
5. Melakukan pertimbangan berikut,

- (a) jika kegiatan kritis pada salah satu lintasan kritis tidak ada yang dapat dipercepat, maka proses berhenti dan diperoleh hasil yang optimum yaitu waktu proyek yang memiliki biaya paling minimum,
 - (b) jika terdapat lebih dari satu lintasan kritis, maka langkah yang dapat dilakukan adalah langkah ke 6 ,
 - (c) apabila keadaan (a) dan (b) tidak terpenuhi, maka langkah yang dipilih adalah langkah yang ke 7
6. Menentukan kegiatan yang dipercepat dengan pertimbangan berikut,
- (a) Menentukan kegiatan yang termuat pada setiap lintasan kritis dengan biaya *cost-slope* terendah. Biaya *cost-slope* kegiatan tersebut dinotasikan dengan CSC.
 - (b) Menentukan satu kegiatan dengan biaya *cost-slope* terendah untuk masing-masing lintasan kritis dan jumlah dari biaya *cost-slope* tersebut dinotasikan dengan CS Total.
 - (c) Apabila $CSC < CS \text{ Total}$, maka langkah selanjutnya adalah langkah ke 8. Apabila tidak, maka langkah selanjutnya adalah sebagai berikut.
 - (i) Menentukan *crash time limit* (CTL) untuk setiap kegiatan yang diperoleh dari proses (b).
 - (ii) Menentukan nilai *crash time limit* (CTL) yang minimum.
 - (iii) Mengurangi setiap kegiatan yang diperoleh dari proses (b) dengan nilai minimum *crash time limit* (CTL) .
 - (iv) Memperbaiki tabel lintasan.
 - (v) Memperbaiki tabel *cost-sheet*.
 - (vi) Menuju langkah ke 9.

Pada langkah 6 ini terdapat beberapa hal yang perlu diperhatikan yaitu,

- (a) Misalkan kegiatan B merupakan kegiatan yang termuat pada setiap lintasan kritis dan merupakan salah satu kegiatan yang dapat dipercepat dengan biaya *cost-slope* terendah untuk masing-masing lintasan kritis, maka dapat disimpulkan bahwa pertidaksamaan $CSC < CS \text{ Total}$ terpenuhi.
- (b) Misalkan kegiatan B merupakan kegiatan yang termuat pada setiap lintasan kritis dan kegiatan C merupakan salah satu kegiatan yang memiliki

biaya *cost-slope* terendah untuk masing-masing lintasan kritis dengan $B \neq C$. Apabila kegiatan C juga termuat pada beberapa lintasan kritis lain yang memiliki kegiatan dengan biaya *cost-slope* terendah yang berbeda, maka penentuan kegiatan yang dipercepat pada lintasan yang memuat kegiatan C tersebut perlu dipertimbangkan dengan menggunakan langkah ke 6 Algoritma *Crash Time Limit* (CTL) seperti berikut.

- (i) Menentukan lintasan kritis yang memuat kegiatan C sebagai lintasan yang akan dipertimbangkan lebih lanjut.
 - (ii) Menentukan nilai $CSC_i = CS_C$.
 - (iii) Menentukan kegiatan kritis dengan biaya *cost-slope* terendah untuk masing-masing lintasan kritis yang dipertimbangkan dan menentukan $CS_i Total$ yang merupakan jumlah biaya *cost-slope* kegiatan tersebut
 - (iv) Jika $CSC_i < CS_i Total$ terpenuhi, maka kegiatan C yang dipercepat. Jika tidak terpenuhi, maka gabungan kegiatan kritis dengan biaya *cost-slope* terendah untuk masing-masing lintasan kritis yang dipercepat.
7. Menentukan kegiatan kegiatan kritis dengan biaya *cost-slope* terendah yang dapat dipercepat dan kemudian menentukan nilai *crash time limit* (CTL) dengan syarat yaitu apabila kegiatan kritis terletak pada CP dan NP, maka kegiatan kritis dipercepat dengan NCTL. Sedangkan apabila tidak maka kegiatan kritis dipercepat dengan CTL.
 8. Pada langkah ini terdapat beberapa tahapan yaitu
 - (a) menentukan nilai CTL atau NCTL,
 - (b) mengurangi kegiatan yang dipercepat dengan CTL atau NCTL,
 - (c) memperbarui tabel lintasan,
 - (d) memperbarui tabel *cost-sheet*.
 9. Jika total biaya proyek mengalami peningkatan, maka proses berhenti dan total biaya yang paling minimum merupakan nilai optimum, sedangkan apabila sebaliknya maka dilakukan langkah yang ke 5.

Penerapan Algoritma *Crash Time Limit* (CTL)

Dalam proses penerapan Algoritma *Crash Time Limit* (CTL) digunakan dua contoh kasus dengan data yang digunakan untuk contoh kasus 1 diperoleh dari

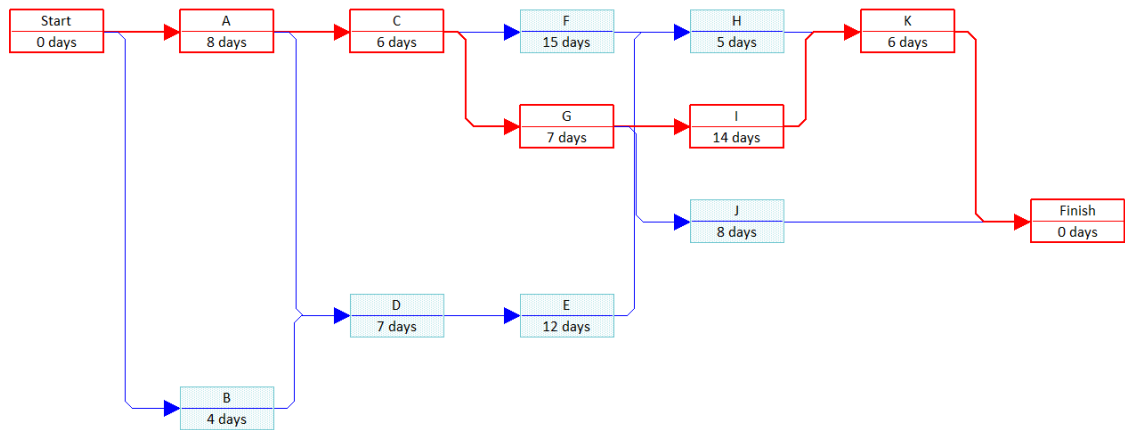
Zaenalzadeh (2011) dalam jurnal *Australian Journal of Basic and Applied Sciences* yang menggunakan program Lingo 12.0 untuk menyelesaikan model matematika permasalahan percepatan proyek dan untuk contoh kasus 2 diperoleh dari Shanmugasundaram & Kumar (2015) dalam jurnal *International Journal of in Multidisciplinary and Academic Research* yang menggunakan metode *crashing* optimum untuk menyelesaikan permasalahan percepatan proyek. Kedua contoh kasus dipilih dari kedua jurnal tersebut karena memiliki data lengkap yang dibutuhkan dalam proses penyelesaian permasalahan percepatan proyek dengan menggunakan Algoritma *Crash Time Limit* (CTL) yaitu berupa kegiatan, kegiatan pendahulu, biaya normal, biaya *crash*, waktu normal, waktu *crash*, dan biaya tidak langsung. Untuk ilustrasi pengerjaan menggunakan Algoritma *Crash Time Limit* (CTL) dapat diamati pada Tabel 1 berikut.

Tabel 1 Ilustrasi

Kegiatan	IP	Waktu (Minggu)		Biaya (Rs)		Δt	Cost-Slope
		Normal	Crash	Normal	Crash		
A	-	8	3	7000	10000	5	600
B	-	4	2	6000	8000	2	1000
C	A	6	1	9000	11500	5	500
D	A,B	7	5	2500	3000	2	250
E	D	12	8	10000	16000	4	1500
F	C	15	10	12000	16000	5	800
G	C	7	6	12000	14000	1	2000
H	E,F	5	5	10000	10000	0	-
I	G	14	7	6000	7400	7	200
J	G	8	5	6000	12000	3	2000
K	H,I	6	4	6000	7800	2	900
TOTAL				86500			

Tabel 1 tersebut merupakan tabel data disertai biaya *cost-slope* masing-masing kegiatan dan total biaya proyek pada contoh kasus 2 dengan biaya tidak langsung adalah Rs. 1000 tiap minggu. Untuk itu dengan diperolehnya tabel tersebut maka langkah 1 algoritma telah dilakukan. Pada langkah 2 disiapkan tabel iterasi untuk mencatat hasil yang diperoleh tiap iterasinya berupa kegiatan yang dipercepat, waktu percepatan kegiatan, biaya percepatan, biaya langsung, biaya tidak

langsung, dan total biaya. Dari kegiatan dan kegiatan pendahulu, dapat dilakukan langkah 3 algoritma yaitu membuat representasi jaringan kerja (*network planning*) seperti Gambar 1 berikut.



Gambar 1 Jaringan Kerja

Melalui gambar tersebut diperoleh lintasan terpanjang adalah (Start, A, C, G, I, K, Finish) dengan panjang 41, lintasan terpanjang kedua adalah (Start, A, C, F, H, K, Finish) dengan panjang 40, dan lintasan terpanjang ketiga adalah (Start, A, D, E, H, K, Finish). Untuk itu diperoleh nilai *crash time limit* yaitu $F = 1$ dan $NF = 3$. Untuk selanjutnya yaitu langkah 4, karena $CPL = 41$, maka diperoleh waktu penyelesaian proyek adalah 41 minggu dengan biaya langsung sebesar 86500 dan biaya tidak langsung adalah sebesar $41 \times 1000 = 41000$, sehingga diperoleh total biaya yaitu $86500 + 41000 = 127500$. Selanjutnya adalah proses percepatan kegiatan iterasi 1 dengan menggunakan langkah 5 algoritma yaitu karena banyaknya lintasan kritis yang diperoleh hanya satu maka langkah selanjutnya adalah langkah 7 algoritma. Pada langkah 7 ini, diperoleh bahwa kegiatan kritis dengan *cost slope* terendah yaitu kegiatan I dan karena kegiatan tersebut tidak terletak pada CP dan NP maka nilai *crash time limit* ditentukan oleh CTL_I . Dengan menggunakan langkah 8 algoritma, diperoleh bahwa kegiatan I dapat dipercepat sebesar $CTL_I = \min\{F, \Delta t_I\} = \min\{1, 7\} = 1$ minggu. Akibat dari percepatan kegiatan tersebut, lintasan yang memuat kegiatan I yaitu (Start, A, C, G, I, K, Finish) berkurang 1 satuan menjadi 40 hari. Dengan mempercepat kegiatan I waktu proyek menjadi 40 hari, biaya langsung sebesar $86500 + 200 =$

86700, dan biaya tidak langsung adalah sebesar $40 \times 1000 = 40000$, sehingga diperoleh total biaya pada iterasi 1 yaitu $86700 + 40000 = 126700$. Dengan menggunakan langkah 9 algoritma, karena total biaya pada iterasi 1 lebih kecil dari total biaya awal yaitu 127500 maka proses penyelesaian dilanjutkan dengan mengulang langkah 5.

Hasil Analisis

Dengan menerapkan Algoritma *Crash Time Limit* (CTL) dan penyelesaian dengan software QM for Windows V4 pada dua contoh kasus permasalahan percepatan proyek, diperoleh bahwa hasil yang diperoleh sama dengan hasil yang telah diperoleh Zaenalzadeh (2011) dan Shanmugasundaram & Kumar (2015) yaitu pada contoh kasus 1 waktu penyelesaian proyek dari 20 hari dengan total biaya sebesar 2044000 rial dapat dipercepat menjadi 16 hari dengan total biaya 1990000 rial. Pada contoh kasus 2 dengan menggunakan Algoritma *Crash Time Limit* (CTL) dan penyelesaian dengan software QM for Windows V4 diperoleh bahwa waktu penyelesaian proyek 41 minggu dengan total biaya sebesar Rs. 127500 dapat dipercepat menjadi 30 minggu dengan total biaya Rs. 123400.

PENUTUP

Kesimpulan

Berdasarkan pembahasan dapat disimpulkan beberapa hal berikut,

1. Proses penyelesaian permasalahan percepatan proyek dengan Algoritma *Crash Time Limit* (CTL) memerlukan data berupa kegiatan, kegiatan pendahulu, waktu normal, waktu *crash*, biaya normal, biaya *crash*, dan biaya tidak langsung. Algoritma *Crash Time Limit* (CTL) menentukan kegiatan yang dipercepat dengan memanfaatkan lintasan dalam jaringan, kegiatan yang dipercepat adalah kegiatan kritis yang memiliki biaya *cost-slope* terendah, dan waktu percepatan kegiatan ditentukan dengan nilai *crash time limit*. Untuk hasil yang diperoleh dari penyelesaian dengan Algoritma *Crash Time Limit* (CTL) adalah waktu penyelesaian proyek yang lebih cepat dengan total biaya yang minimum.

2. Dari hasil penerapan Algoritma *Crash Time Limit* (CTL) dan penyelesaian menggunakan QM for Windows V4 diperoleh bahwa contoh kasus 1 yang memiliki waktu penyelesaian 20 hari dengan total biaya sebesar 2044000 rial dapat dipercepat menjadi 16 hari dengan total biaya 1990000 rial. Contoh kasus 2 yang memiliki waktu penyelesaian proyek 41 minggu dengan total biaya sebesar Rs. 127500 dapat dipercepat menjadi 30 minggu dengan total biaya Rs. 123400
3. Algoritma *Crash Time Limit* (CTL) dapat digunakan untuk menentukan waktu dan biaya percepatan proyek dengan memanfaatkan lintasan dalam jaringan dan nilai *crash time limit*.

Saran

Permasalahan percepatan proyek dapat diselesaikan dengan menggunakan Algoritma *Crash Time Limit* (CTL) sebagai salah satu alternatif penyelesaian masalah. Bagi yang ingin menyelesaikan permasalahan percepatan proyek dengan metode lain dapat digunakan metode penyelesaian dengan program dinamik atau metode fuzzy sebagai penelitian lebih lanjut.

DAFTAR RUJUKAN

- Hillier, F. S. & Lieberman. G. J. 2001. *Introduction To Operations Research Seventh Edition*. New York: McGraw-Hill Companies, Inc.
- Kasana, H. S., & Kumar, K. D. 2004. *Introductory Operations Research*. New York: Springer.
- Kharde, B. R., & Pathil, G. J. V. 2012. An Efficient Algorithm For Crashing, *International Journal of Scientific & Technology Research*, 1 (4): 94-101.
- Sen, R. P. 2010. *Operation Research: Algorithms and Applications*. New Delhi: PHI Learning Private Limited.
- Shanmugasundaram, S., & Kumar, V. M. 2015. Application of project scheduling and Cost analysis Using PERT and CPM. *International Journal of in Multidisciplinary and Academic Research (SSIJMAR)*, 4 (2): 1-9.
- Tsui, F. 2004. *Managing Software Projects*. London: Jones and Bartlett Publishers, Inc.
- Uher, T. E., & Zantis, A. S. 2011. *Programming And Scheduling Techniques*. Canada: Spon Press.
- Zeinalzadeh, A. 2011. An Application of Mathematical Model to Time-cost Trade off Problem (Case Study). *Australian Journal of Basic and Applied Sciences*, 5 (7): 208-214.

