

# OPTIMASI PENJADWALAN PROYEK MENGGUNAKAN ALGORITMA GENETIKA

Wiwik Anggraeni, Amalia Utamima, Khairil Juhdi Siregar

Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember  
Kampus Keputih, Sukolio Surabaya 60111  
Telp: (031) 5999944, Fax: (031) 5964965  
E-mail: [wiwik@is.its.ac.id](mailto:wiwik@is.its.ac.id)

---

## Abstract

*Scheduling problems belong to the class of NP-hard problems with complex combinatorial optimization problems. Scheduling project consists of several activities in which they are interconnected. The number and the relationship between these activities make the search space solution for project scheduling problems is very broad. It needs an optimization method to find the objective function of scheduling problem, which is to minimize the duration of the project. Genetic algorithm is quite flexible and accurate for dealing with optimization problem. By utilizing Genetic Algorithm method, this research seeks to optimize project scheduling problem with the objective function to minimize the duration of the project. PERT method was also conducted to become a comparer. Based on the calculation, the optimal result is obtained with a minimum duration of the project completion.*

## Abstrak

*Masalah penjadwalan termasuk ke dalam kelas permasalahan NP-hard dengan masalah optimasi kombinatorial yang sangat kompleks. Penjadwalan proyek terdiri dari beberapa aktivitas dimana antara aktivitas satu dengan yang lainnya saling berhubungan. Jumlah dan hubungan antar aktivitas ini membuat ruang pencarian solusi untuk permasalahan penjadwalan proyek sangat luas. Untuk itu, dibutuhkan sebuah metode optimasi untuk menemukan fungsi tujuan dari permasalahan penjadwalan yaitu meminimalkan durasi proyek. Algoritma Genetika dianggap cukup fleksibel dan akurat untuk menyelesaikan masalah optimasi. Dengan memanfaatkan metode Algoritma Genetika, penelitian ini berusaha menyelesaikan permasalahan penjadwalan proyek dengan fungsi tujuan meminimalkan durasi proyek. Metode PERT juga dijalankan sebagai pembanding hasil. Berdasarkan perhitungan tersebut, didapatkan hasil yang optimal dengan durasi penyelesaian proyek paling minimum..*

**Kata kunci:** penjadwalan, algoritma genetika, optimasi, manajemen proyek

## 1. PENDAHULUAN

Penjadwalan proyek memerlukan waktu untuk penugasan kegiatan dalam perencanaan proyek dengan tujuan meminimalkan durasi proyek. Masalah tersebut muncul dalam berbagai bidang termasuk konstruksi, manufaktur, pengembangan perangkat lunak dan sejumlah aplikasi. Karena pentingnya bidang-bidang tersebut, masalah penjadwalan proyek telah banyak diteliti dengan baik oleh kalangan OR (Operational Research) dan AI (Artificial Intelligence) (Carl, 2004).

Selanjutnya, penjadwalan proyek termasuk ke dalam kelas optimasi bertipe NP-hard, yang berarti bahwa tidak mungkin menemukan sebuah algoritma yang paling efisien untuk menyelesaikan pengoptimalan dengan data yang berukuran besar saat melakukan komputasi. Sehingga dibutuhkan penggunaan heuristik ketika menyelesaikan permasalahan dengan ukuran data yang besar (Omara, 2009). Penelitian-penelitian yang bersangkutan dengan

permasalahan ini membahas tentang bagaimana membuat jadwal yang bagus dengan mengutamakan durasi proyek seminimal mungkin. Generalisasi ini banyak dikenal dan dipelajari dengan baik dalam kelas masalah penjadwalan (termasuk masalah job-shop dengan beberapa variasinya).

1. Metode Algoritma Genetika juga telah banyak dilakukan oleh peneliti-peneliti terdahulu. Beberapa kelebihan dari algoritma genetika dalam penyelesaian masalah antara lain dapat mengoptimasi variabel diskrit maupun kontinyu, tidak bermasalah dengan jumlah data yang besar, tidak hanya menyajikan satu solusi namun daftar solusi permasalahan. Untuk itu, Algoritma Genetika banyak digunakan untuk menyelesaikan permasalahan penjadwalan proyek, baik algoritma genetika murni, maupun digabungkan dengan metode-metode yang lain (Beasley, 1993).

2. Alba dan Chicano telah menunjukkan bahwa Algoritma Genetika cukup fleksibel dan akurat untuk penjadwalan proyek, dan dianggap sebagai alat penting untuk manajemen proyek otomatis. Mereka menyediakan ide dasar pada penerapan GA untuk tugas-tugas otomatis (Alba, 2004). Latar belakang permasalahan penjadwalan proyek yang membutuhkan sebuah algoritma untuk mengoptimalkan durasi dalam pengerjaan proyek, dan dengan kelebihan-kelebihan yang dimiliki Algoritma Genetika dalam penyelesaian masalah, maka penelitian ini akan mengimplementasikan metode Algoritma Genetika.

Pada penelitian ini, kromosom direpresentasikan dengan sebuah matriks yang berisi urutan aktivitas, waktu mulai, dan waktu selesai dari aktivitas. Pembentukan populasi dilakukan secara acak sepanjang aktivitas yang ada. Kemudian dilakukan perhitungan waktu mulai dan waktu selesai dari setiap aktivitas. Pada tahap pembentukan generasi selanjutnya dilakukan proses perkawinan silang sederhana dan dilanjutkan dengan proses mutasi.

### 1.1 Penjadwalan

Proyek dapat diartikan sebagai kumpulan dari beberapa tugas, orang, atau sumber daya dengan tujuan untuk meminimalkan waktu atau untuk meminimalkan biaya (A. Lawrence, 2002). Tugas-tugas dari sebuah proyek disebut aktivitas. Untuk mengestimasi penyelesaian proyek selalu dihubungkan dengan setiap aktivitas – aktivitas yang ada dalam proyek tersebut. Aktivitas dalam sebuah proyek bisa saja sedikit maupun banyak, tergantung dari situasi dan jenis proyeknya. Dalam sebuah proyek, ada beberapa aktivitas yang harus terlebih dahulu dikerjakan sebelum aktifas sebelumnya dimulai. Ada juga aktivitas-aktifas yang dilakukan secara bersamaan. Ketepatan relasi atau hubungan aktivitas yang satu dengan aktivitas yang lainnya sangatlah penting guna mengoptimalkan jadwal untuk setiap aktivitas itu sendiri, seperti pada Gambar 1.

Untuk menghitung waktu mulai tercepat (ES) dan waktu selesai tercepat (EF) dari aktivitas yang tidak memiliki penghubung yang menuju kepada aktivitas tersebut. Dalam hal ini dimulai dengan aktivitas A. aktivitas A memiliki nilai  $ES(A) = 0$ , karena merupakan aktivitas pemula. Sedangkan untuk nilai  $EF(A) = 90$ , yaitu durasi waktu yang diperlukan untuk menyelesaikan aktivitas tersebut. Aktivitas B, F, dan I berhubungan langsung dengan aktivitas A, sehingga nilai ES dari aktivitas-aktivitas tersebut adalah EF dari aktivitas A. Karena aktivitas B, F, dan I tidak

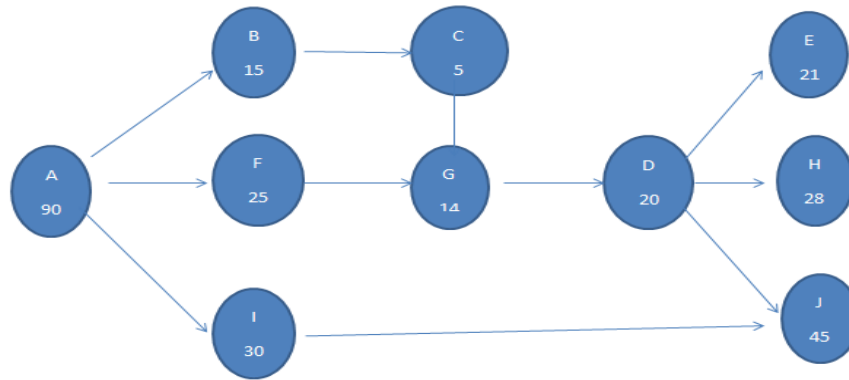
akan bisa dimulai sebelum aktivitas A selesai. Sehingga  $ES(B), (F), (I) = 90$ . Sedangkan untuk nilai EF dari masing-masing aktivitas B, F, dan I adalah nilai ES dari masing-masing aktivitas dijumlahkan dengan durasi waktu yang diperlukan untuk menyelesaikan aktivitas tersebut. Jadi  $EF(B) = 90 + 15 = 105$ ,  $EF(F) = 90 + 25 = 115$ , dan  $EF(I) = 90 + 30 = 120$ . Adapun untuk aktivitas yang terhubung dengan lebih dari satu aktivitas maka dalam penentuan ES dari aktivitas tersebut memilih EF yang paling besar/maksimum dari aktifitas-aktivitas yang terhubung dengannya. Seperti aktivitas G yang terhubung dengan aktivitas C dan F. Maka untuk penentuan  $ES = \max(ES(C), ES(F)) = \max(110, 115) = 115$ . Sedangkan untuk EF dari aktivitas G  $= 115 + 14 = 129$ .

$ES$  = Maksimum EF dari masing-masing aktivitas yang menuju kepada aktivitas tersebut  
 $EF$  =  $ES + (\text{durasi waktu penyelesaian aktivitas})$

### 1.2 Algoritma Genetika

Sebuah solusi yang dibangkitkan dalam algoritma genetika disebut sebagai kromosom, sedangkan kumpulan kromosom-kromosom tersebut disebut sebagai populasi. Sebuah kromosom dibentuk dari komponen-komponen penyusun yang disebut sebagai gen dan nilainya dapat berupa bilangan numerik, biner, simbol ataupun karakter tergantung dari permasalahan yang ingin diselesaikan. Kromosom-kromosom tersebut akan berevolusi secara berkelanjutan yang disebut dengan generasi. Dalam tiap generasi kromosom-kromosom tersebut dievaluasi tingkat keberhasilan nilai solusinya terhadap masalah yang ingin diselesaikan (fungsi objektif) menggunakan ukuran yang disebut dengan fitness.

Untuk memilih kromosom yang tetap dipertahankan untuk generasi selanjutnya dilakukan proses yang disebut dengan seleksi. Proses seleksi kromosom menggunakan konsep aturan evolusi Darwin yang telah disebutkan sebelumnya yaitu kromosom yang mempunyai nilai fitness tinggi akan memiliki peluang lebih besar untuk terpilih lagi pada generasi selanjutnya. Kromosom-kromosom baru yang disebut dengan offspring, dibentuk dengan cara melakukan perkawinan antar kromosom dalam satu generasi yang disebut sebagai proses crossover. Mekanisme perubahan susunan unsur penyusun makhluk hidup akibat adanya faktor alam yang disebut dengan mutasi direpresentasikan sebagai proses berubahnya satu atau lebih nilai gen dalam kromosom dengan suatu nilai acak.



Gambar 1. aktivitas yang saling berhubungan

## 2. METODOLOGI

Fungsi tujuan dari permasalahan penjadwalan proyek dapat dituliskan sebagai berikut :

$$\text{Meminimalkan } f_j = \max[EF_n] \quad (1)$$

Keterangan :

$f_j$  = Waktu selesai keseluruhan proyek.

$EF_n$  = *Earliest Finish* dari aktivitas terakhir dari proyek.

Dengan beberapa batasan yang ada, diantaranya:

a. Mode menjalankan aktivitas

Memastikan bahwa setiap aktivitas hanya dijalankan pada 1 mode.

$$\sum_{m=1}^M y_{jm} = 1 \quad \forall j \in \{1, \dots, J\} \quad (2)$$

Keterangan :

$m$  = Mode aktivitas.

$M_j$  = Jumlah mode yang dimiliki oleh aktivitas  $j$ .

$y_{jm}$  = Nilai yang diberikan pada mode  $m$ .

b. Urutan Pengerjaan Aktivitas

Memastikan bahwa waktu selesai suatu aktivitas lebih kecil dari atau sama dengan waktu mulai aktivitas berikutnya

$$f_i \leq s_j \quad \forall i \in \{1, \dots, J\}, \quad \forall i \in P_j \quad (3)$$

Keterangan :

$s_j$  = Waktu mulai dari sebuah aktivitas.

$f_i$  = Waktu selesai dari aktivitas sebelumnya.

$P_j$  = *Predecessor* dari sebuah aktivitas.

c. Penggunaan Sumber daya

Membatasi jumlah sumber daya pada setiap aktivitas agar kurang dari atau sama dengan kapasitas sumber daya pada suatu periode

$$\sum_{j=2}^{J-1} \sum_{m=1}^M (k_{jmt} \times y_{jm}) \leq K_{rt} \quad \forall r \in \{1, \dots, R\}, \quad \forall t \in \{1, \dots, T\} \quad (4)$$

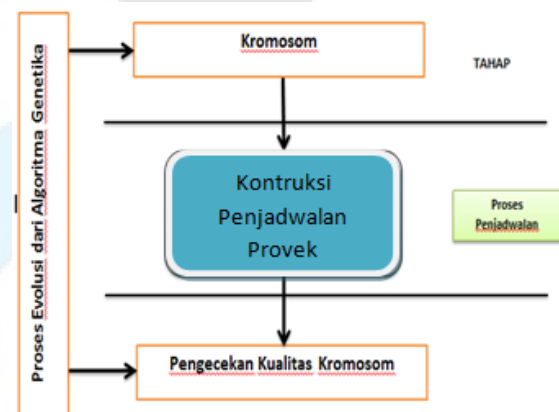
Keterangan :

$y_{jm}$  = Nilai yang diberikan pada mode  $m$ .

$k_{jmt}$  = Kebutuhan sumber daya  $r$  pada aktivitas  $j$  jika aktivitas tersebut dijalankan pada mode  $m$ .

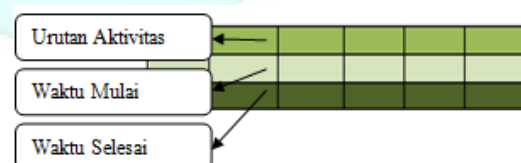
$K_{rt}$  = Kapasitas sumber daya yang tersedia pada periode  $t$ .

Konstruksi penjadwalan proyek diperlukan untuk membuat batasan-batasan pada pengoptimalan yang dilakukan oleh algoritma genetika. Konstruksi penjadwalan yang perlu diperhatikan yaitu aturan dalam penghitungan durasi proyek. Desain global dari algoritma genetika untuk penjadwalan proyek pada penelitian ini ditunjukkan oleh Gambar 2.



Gambar 2. Desain Algoritma Genetika

Representasi kromosom dari algoritma ini akan dibentuk seperti pada Gambar 3.



Gambar 3. Kromosom

Berikut adalah langkah-langkah yang dilakukan dalam menghitung penjadwalan proyek :

Langkah 1: Inisialisasi

Langkah 2: Dilakukan perulangan untuk sepanjang predesesor aktivitas.

Pengecekan terhadap jumlah predesesor aktivitas  
Jika aktivitas  $j$  memiliki 1 predesesor, maka ES aktivitas  $j$  adalah EF dari aktivitas sebelumnya.  
Jika aktivitas  $j$  memiliki predesesor lebih dari 1, maka ES dari aktivitas  $j$  adalah  $\max(Ef_{ij})$ .

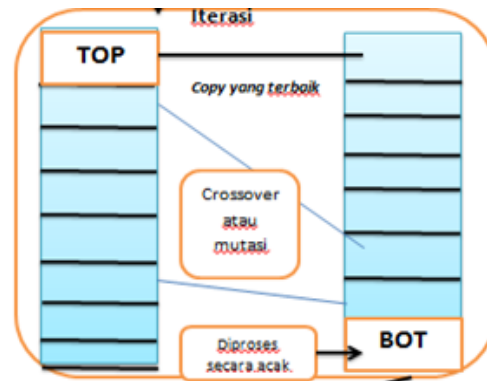
Jika aktivitas  $j$  memiliki tidak memiliki predesesor, maka ES aktivitas  $j$  adalah 0, dan aktivitas ini merupakan aktivitas pertama. EF dari aktivitas adalah ES + durasi aktivitas. Langkah 3. Menghitung Completion time/ CT dari seluruh aktivitas:  
 $f_n = \max (EF)$

Data yang dipakai dalam penelitian ini berupa salah satu data proyek yang disediakan oleh salah satu situs web yaitu *Project Scheduling Problem Library* (PSPLIB), [www.ombd.wi.tum.de/psplib/data.html](http://www.ombd.wi.tum.de/psplib/data.html). (Kolisch, 1996). Penelitian ini mengadopsi jenis genetic algorithm yang digunakan oleh Jairo et al (2010), kemudian menerapkan dataset yang berbeda, yaitu dari PSPLIB. Penelitian ini menggunakan perangkat lunak Matlab. Berikut langkah-langkah penyelesaiannya dengan program komputer.

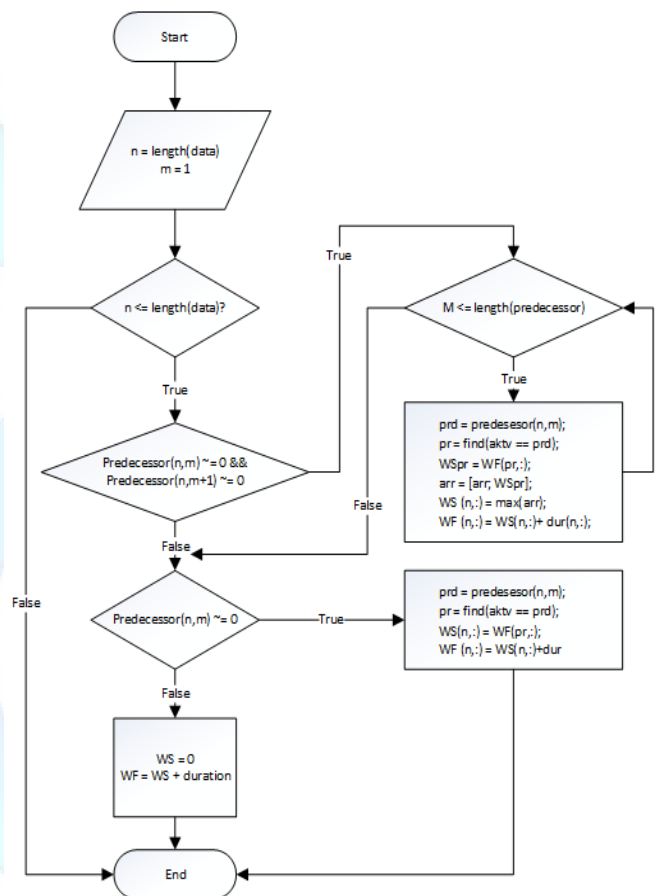
- Inisialisasi data  
Data dan variabel harus diinisialisasikan terlebih dahulu. Terdapat tiga data yang akan dimasukkan ke dalam program aplikasi, antara lain aktivitas, durasi, dan *predesesor*.
- Memasukkan data awal
- Mencari durasi aktivitas (Gambar 5)
- Menentukan cost(biaya) setiap individu (Gambar 6)
- Menentukan parameter genetika
- Inisialisasi Populasi
- Perkawinan Silang
- Mutasi
- Memasukkan Batasan model

Untuk mendapatkan generasi atau populasi selanjutnya, seperti ditunjukkan Gambar 4, langkah pertama yang dilakukan adalah dengan memindahkan individu yang berada pada urutan pertama pada populasi pertama. Urutan populasi pertama terdiri dari yang terbaik hingga yang terburuk, dari urutan atas ke bawah. Jadi urutan individu yang berada pada tingkat teratas akan dipindahkan pada populasi selanjutnya sebanyak yang ditentukan. Jika kemungkinan kawin silang adalah 0,7, maka yang akan dipindahkan adalah sekitar 10% hingga 15%.

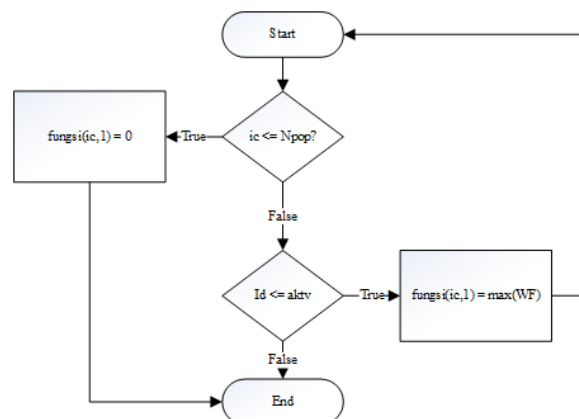
Proses kawin silang atau *crossover* akan menggunakan metode kawin silang sederhana. Misalkan anggota induk adalah  $pa = [x_1, x_2, \dots, x_n]$  dan  $ma = [y_1, y_2, \dots, y_n]$  dan  $r$  adalah bilangan acak diskrit yang bernilai antara 1 dan panjang vektor  $n$  (panjang aktivitas). Kemudian untuk individu-individu yang tersisa kan diproses secara acak, kemudian dimasukkan ke dalam populasi selanjutnya. Jika terjadi konvergensi dini di dalam proses evolusi, maka akan dilakukan proses mutasi.



Gambar 4. Proses Evolusi Generasi selanjutnya



Gambar 5. Diagram alur mencari durasi aktivitas



Gambar 6 Diagram alur cost kromosom

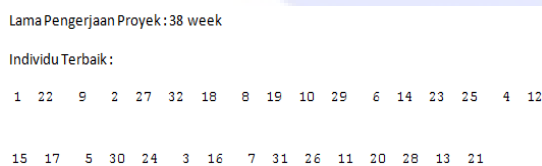


Konvergensi dini ini mengakibatkan pencarian terhadap solusi/generasi yang lebih baik melemah. Mutasi bekerja secara spontan dengan cara mengubah kromosom tanpa melalui persilangan dengan kromosom lain. Proses mutasi memiliki peranan penting saat populasi terjebak pada situasi konvergensi dini. Fungsi objektif dari penyelesaian permasalahan ini adalah untuk meminimalkan *makespan* dari proyek.

### 3. HASIL DAN PEMBAHASAN

Waktu mulai dari sebuah aktivitas tergantung pada aktivitas yang berhubungan dengannya. Aktivitas yang memiliki predesesor sama bisa dikerjakan secara bersama-sama bisa juga tidak. Agar setiap aktivitas yang berjalan tidak saling tumpang tindih pengerjaannya, maka dibutuhkan sebuah aturan agar aktivitas berjalan sesuai dengan predesesor masing-masing. Untuk menghindari adanya ketidaktersediaan kapasitas sumber daya tertentu, maka dibutuhkan aturan yang membatasi pemakaian sumber daya yang ada.

Uji coba dilakukan terhadap sebuah dataset dari proyek pada PSPLIB. Gambar 7 merupakan hasil dari output biaya per kromosom, dengan nilai yaitu 38. Oleh karena itu lama pengerjaan proyek adalah 38 minggu. Bagian selanjutnya adalah individu terbaik yang diambil dari peringkat pertama sepanjang 20 populasi.



Lama Pengerjaan Proyek: 38 week																			
Individu Terbaik:																			
1	22	9	2	27	32	18	8	19	10	29	6	14	23	25	4	12			
15	17	5	30	24	3	16	7	31	26	11	20	28	13	21					

Gambar 7. Output dari Algoritma Genetika

Individu terbaik yang ditunjukkan pada gambar diatas merupakan individu baru yang telah mengalami proses genetika. Proses mutasi telah menghasilkan individu yang tidak sama dengan individu sebelumnya yaitu urutan aktivitas 1 sampai aktivitas 32. Individu baru ini tetap menghasilkan durasi yang optimal karena dalam proses genetika yang dilakukan tetap mengacu pada proses perhitungan ES dan EF antara aktivitas yang berhubungan. Proses validasi dilakukan dengan membandingkan keluaran antara metode algoritma genetika dan PERT yang telah dibuat pada perangkat lunak Matlab.

Tabel 1 Perbandingan Output GA dan PERT

Metode	Hasil
Algoritma Genetika	38 minggu
PERT	38 minggu

Tabel 2 Perbandingan Runtime GA dan PERT

Percobaan ke-	Algoritma Genetika	PERT
1	0.0156	0.5148
2	0.0468	0.4680
3	0.0156	0.078001
4	0.0156	0.093601
5	0.0156	0.078001
6	0.0156	0.0624
7	0.0156	0.078001
8	0.0156	0.078001
9	0.0156	0.078001
10	0.0156	0.078001

Berdasarkan Tabel 1 yang merupakan hasil keluaran program Algoritma Genetika dan program PERT maka dapat disimpulkan bahwa model dan program telah valid karena nilai yang dikeluarkan sama.

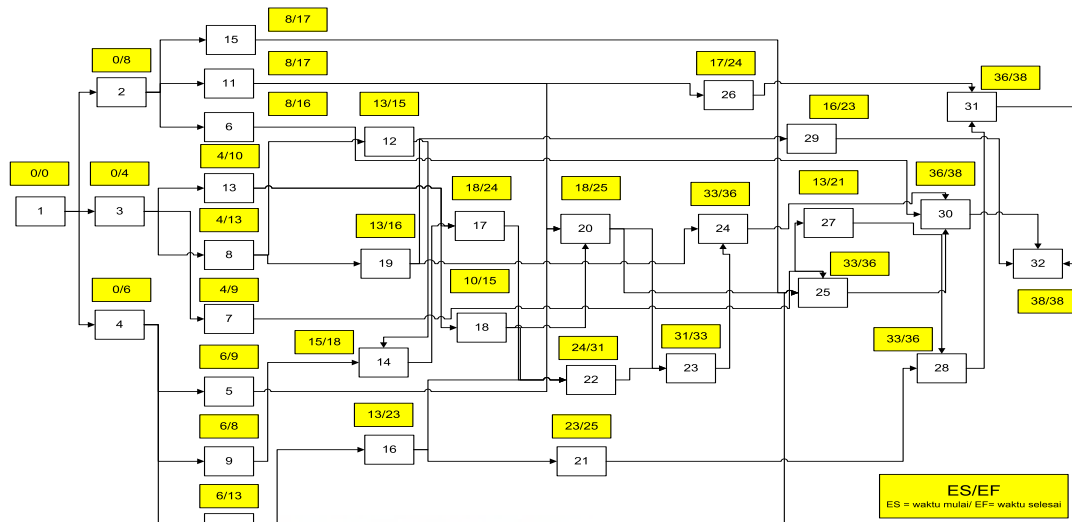
Untuk validasi yang dilakukan terhadap penggunaan metode optimasi yaitu dengan membandingkan *running time* yang dibutuhkan metode dalam mengeluarkan hasil yang diharapkan. Tabel 2 menunjukkan perbandingan *running time* antara metode algoritma Genetika dan metode PERT dalam 10 kali percobaan yang dijalankan di perangkat lunak Matlab. Hasil perbandingan dapat dilihat bahwa dalam 9 dari 10 percobaan, *running time* Algoritma Genetika lebih kecil dari metode PERT. Pada setiap percobaan, hasil output dari Algoritma Genetika juga sama dengan PERT, yaitu 38. Sehingga dapat disimpulkan bahwa dengan ruang lingkup permasalahan yang kompleks, Algoritma Genetika lebih efisien daripada metode PERT.

Nilai optimal yang didapatkan dengan meminimalkan durasi proyek menggunakan algoritma genetika adalah 38 minggu. Hasil optimal telah divalidasi dengan dibandingkannya dengan metode lain yaitu PERT.

### 4. SIMPULAN dan SARAN

Berdasarkan proses-proses yang telah dilakukan dalam pengerjaan penelitian ini maka ada beberapa kesimpulan yang dapat diambil, diantaranya adalah:

1. Metode Algoritma Genetika mampu menghasilkan durasi optimal dari permasalahan penjadwalan proyek yang diselesaikan pada penelitian ini yaitu 38 minggu.
2. Dari uji coba yang telah dilakukan, algoritma genetika lebih efisien daripada metode PERT karena memiliki runtime yang lebih cepat.



Gambar 8. Waktu mulai dan waktu selesai setiap aktivitas

## 5. DAFTAR RUJUKAN

- A. Lawrence, J. &. (2002). *"Applied Management Science: Modelling, Spreadsheet Analysis, and Communication for Decision Making"*. California: California State University Fullerton: Jhon Wiley & Son. Inc.
- Beasley, D. B. (1993). "An overview of genetic algorithms: Part 1, fundamentals". *University Computing* , 58-69.
- Carl, K. &. (2004). "Time-line based model for software project scheduling with genetic algorithms". *European Journal of Operational Research. Elsevier B.V.*
- Jairo R, Edgar G, Carolina P (2010). "Project scheduling with limited resources using a genetic algorithm". *International Journal of Project Management*. vol 28, 619-628.
- Kolisch, R. a. (1996). " PSPLIB - A project scheduling library". *European Journal of Operational Research* , 205-216.
- Omara, F. A. (2009). " Genetic algorithms for task scheduling problem ". *European Journal of Operational Research. Elsevier B.V.*
- E Alba, J. C. (2004). "Genetic and Evolutionary Computation". *International Journal of Operational Research* , 852-863.