

**USULAN TUGAS AKHIR**

**1. IDENTITAS PENGUSUL**

**NAMA** : TEGUH SURYO SANTOSO  
**NRP** : 5110100164  
**DOSEN WALI** : Ir. Muchammad Husni, M.Kom  
**DOSEN PEMBIMBING** : 1. Ahmad Saikhu, S.SI., MT.  
2. Ir. Rully Soelaiman, M.Kom

**2. JUDUL TUGAS AKHIR**

“Desain dan Analisis Algoritma Modifikasi *Hungarian* untuk Permasalahan Penugasan Dinamis pada Studi Kasus Permasalahan SPOJ Klasik 12749”

**3. LATAR BELAKANG**

Masalah penugasan adalah salah satu masalah optimasi kombinatorial mendasar yang merupakan cabang dari ilmu optimasi atau riset operasi. Secara umum, masalah penugasan dapat dinyatakan dengan adanya sejumlah agen dan sejumlah tugas dimana setiap agen dapat ditugaskan untuk melakukan beberapa tugas dengan masing-masing penugasan memakan biaya tertentu. Hasil akhir yang diperoleh adalah penetapan tepat satu agen untuk setiap tugas dan tepat satu tugas kepada setiap agen sedemikian sehingga didapat susunan penugasan yang optimal dimana total biaya yang dikeluarkan adalah sekecil-kecilnya atau sebesar-besarnya sesuai dengan kebutuhan. Beberapa algoritma dapat digunakan untuk menyelesaikan masalah penugasan ini.

Masalah penugasan dapat diterapkan untuk menyelesaikan masalah transportasi atau masalah-masalah lain pada kehidupan nyata. Dalam prakteknya, masalah penugasan dapat bersifat dinamis, yaitu, terdapat kemungkinan perubahan biaya pada beberapa penugasan atau penambahan agen dan tugas yang baru akibat faktor tertentu. Ketika hal tersebut terjadi, susunan penugasan yang optimal mungkin saja berubah dari susunan penugasan sebelumnya. Salah satu solusi yang dapat dilakukan untuk mendapatkan susunan penugasan yang optimal adalah dengan melakukan kembali proses komputasi dari awal dengan menggunakan algoritma yang telah ditetapkan. Hal tersebut akan menimbulkan masalah jika perubahan biaya seringkali terjadi. Proses komputasi akan dilakukan berulang kali dari tahap awal dimana untuk

algoritma yang ada sampai saat ini, proses tersebut membutuhkan waktu yang tidak sedikit. Maka dari itu perlu adanya algoritma untuk menyelesaikan masalah penugasan dinamis, yaitu melakukan proses komputasi untuk mendapatkan susunan penugasan yang optimal ketika terjadi perubahan biaya atau penambahan agen dan tugas yang baru dengan memanfaatkan solusi yang telah ada agar waktu yang dibutuhkan untuk melakukan proses komputasi tersebut lebih cepat daripada melakukan proses komputasi dari awal.

Beberapa algoritma yang dapat digunakan untuk menyelesaikan masalah ini adalah algoritma *Hungarian* dan algoritma *dynamic Hungarian*. Algoritma *Hungarian* dapat menyelesaikan masalah penugasan dengan kompleksitas waktu  $O(n^4)$  dimana  $n$  adalah banyaknya *vertices* pada salah satu bagian dari *bipartite graph*. Namun dengan menggunakan struktur data yang tepat, kompleksitas waktu dapat diturunkan menjadi  $O(n^3)$ . Sedangkan algoritma *dynamic Hungarian* dapat menyelesaikan masalah penugasan dinamis dengan kompleksitas waktu  $O(kn^2)$  dimana  $k$  adalah ukuran banyaknya perubahan yang terjadi.

Dari penjelasan di atas, hasil yang diharapkan dalam tugas akhir ini adalah proses komputasi untuk menyelesaikan masalah penugasan dinamis dengan dengan konsumsi waktu dan *memory* yang efisien. Tentunya dilakukan analisis yang komprehensif terlebih dahulu dalam menentukan algoritma yang digunakan untuk menyelesaikan masalah ini.

#### 4. RUMUSAN MASALAH

Beberapa permasalahan yang akan diselesaikan pada tugas akhir ini adalah:

- Bagaimana menganalisis dan merancang algoritma yang efisien untuk menyelesaikan masalah penugasan dinamis?
- Bagaimana mengimplementasikan algoritma yang telah terancang untuk menyelesaikan masalah penugasan dinamis secara efisien?

#### 5. BATASAN MASALAH

Beberapa batasan masalah yang menjadi batas pada tugas akhir ini adalah:

- Jenis graf yang digunakan adalah *complete bipartite graph*.
- Jumlah *vertices* pada masing-masing partisi *bipartite graph* tidak lebih dari 100 buah termasuk penambahan *vertices* pada nantinya.
- Jumlah *vertices* pada masing-masing partisi *bipartite graph* harus sama.
- Jumlah perubahan biaya dan penambahan *vertices* baru tidak lebih dari 10000 kali.
- Biaya dari tiap penugasan lebih dari atau sama dengan 0 dan mampu ditampung dalam tipe data *integer* 32 bit bertanda.
- Bahasa pemrograman yang digunakan adalah C++.

#### 6. TUJUAN PEMBUATAN TUGAS AKHIR

Tujuan dari pembuatan tugas akhir ini adalah:

- a. Menganalisis dan merancang algoritma yang efisien untuk menyelesaikan masalah penugasan dinamis.
- b. Mengimplementasikan algoritma yang telah terancang untuk menyelesaikan masalah penugasan dinamis secara efisien.

## 7. MANFAAT TUGAS AKHIR

Manfaat dari tugas akhir ini adalah untuk menghasilkan kajian yang komprehensif serta implementasi algoritma yang digunakan untuk menyelesaikan masalah penugasan dinamis dengan konsumsi waktu dan *memory* yang efisien.

## 8. TINJAUAN PUSTAKA

### a. *Bipartite Graph*

Dalam teori graf, *bipartite graph* adalah graf yang *vertices*-nya dapat dibagi menjadi dua *disjoint set*  $U$  dan  $V$  sehingga setiap *edge* menghubungkan sebuah *vertex* di  $U$  ke sebuah *vertex* di  $V$ , dimana  $U$  dan  $V$  masing-masing adalah set yang independen. Sebuah *bipartite graph* adalah graf yang tidak mengandung *cycle* dengan panjang yang berjumlah bilangan ganjil.

*Bipartite graph* dapat ditulis dengan notasi  $G = (U, V, E)$  dimana  $U$  dan  $V$  adalah dua partisi *vertices* dan  $E$  yang menunjukkan *edges*. Jika *bipartite graph* tidak terhubung, graf tersebut memiliki lebih dari satu bipartisi, dalam hal ini notasi  $G = (U, V, E)$  sangat membantu dalam menentukan salah satu bipartisi yang kemungkinan penting untuk dibedakan pada kasus tertentu. Jika sebuah *bipartite graph* memiliki properti  $|U| = |V|$ , dengan kata lain jika dua himpunan *vertices* pada *bipartite graph* memiliki kardinalitas yang sama, maka graf tersebut disebut *balanced bipartite graph*. Jika *vertices* pada partisi yang sama pada sebuah *bipartite graph* memiliki derajat yang sama, maka graf tersebut disebut *biregular*.

### b. Algoritma *Hungarian*

Algoritma *Hungarian* adalah algoritma optimasi kombinatorial yang dapat menyelesaikan masalah penugasan dalam waktu polinomial. Algoritma ini dikembangkan oleh Harold Kuhn pada tahun 1955 dan memberi nama algoritma ini sebagai algoritma *Hungarian* karena algoritma ini sebagian besar didasarkan pada karya sebelumnya dari dua ahli matematika asal Hungaria: Dénes König dan Jenő Egerváry.

James Munkres membahas algoritma ini pada tahun 1957 dan menemukan bahwa kompleksitas waktu dari algoritma ini adalah polinomial. Sejak itu algoritma ini dikenal juga sebagai algoritma Kuhn-Munkres atau algoritma *Munkres assignment*. Kompleksitas waktu asli dari algoritma ini adalah  $O(n^4)$ , namun Edmonds dan Karp serta Tomizawa mendapati bahwa algoritma tersebut dapat dimodifikasi untuk mencapai kompleksitas waktu  $O(n^3)$ .

### c. Pemrograman Linier

Metode Pemrograman linier pertama kali ditemukan oleh ahli statistika Amerika Serikat yang bernama Prof. George Dantzig. Pemrograman Linier disingkat PL merupakan metode matematik dalam mengalokasikan sumber daya yang terbatas untuk mencapai suatu tujuan seperti memaksimumkan keuntungan dan meminimumkan biaya. PL banyak diterapkan dalam masalah ekonomi, industri, militer, sosial dan lain-lain. PL berkaitan dengan penjelasan suatu kasus dalam dunia nyata sebagai suatu model matematik yang terdiri dari sebuah fungsi tujuan linier dengan beberapa kendala linier.

Tahapan dalam penyelesaian optimasi dari Pemrograman linier ini adalah sebagai berikut:

- 1) Menentukan *decision of variables*.
- 2) Membuat *objective function*.
- 3) Memformulasikan *constraints*.
- 4) Menggambarkan dalam bentuk grafik.
- 5) Menentukan daerah kemungkinan atau *feasible*.
- 6) Menentukan solusi optimal.

Dua jenis pendekatan yang sering digunakan dalam metode pemrograman linier ini, yaitu:

- 1) Metode Grafik
  - a) Digunakan untuk menyelesaikan optimasi dengan maksimum dua variabel.
  - b) Untuk variabel lebih dari dua, penyelesaiannya menggunakan metode kedua.
- 2) Metode Simpleks
  - a) Digunakan untuk proses dengan jumlah variabel lebih dari dua.
  - b) Tahapan dalam metode simpleks ini lebih kompleks dibandingkan dengan metode grafik.

### d. Dualitas

Ditinjau dari teori dan praktek, maka dualitas merupakan konsep pemrograman linier yang penting. Ide dasar dari teori dualitas adalah bahwa setiap persoalan pemrograman linier mempunyai suatu program linier yang berkaitan yang disebut dual. Solusi dan persoalan asli pemrograman linier (primal) juga memberikan solusi pada dualnya.

Secara sistematis, dualitas merupakan alat bantu masalah pemrograman linier, yang secara langsung didefinisikan dari persoalan aslinya atau dari model pemrograman linier primal. Secara umum dualitas sangat tergantung pada primal dalam hal tipe *constraint*, *decision of variables* dan kondisi optimal. Oleh karena itu teori dualitas tidak diharuskan penggunaannya. Primal-dual menunjukkan hubungan secara simetris dengan ketentuan sebagai berikut:

- 1) Koefisien fungsi tujuan primal menjadi konstanta ruas kanan dual.
- 2) Konstanta ruas kanan primal menjadi koefisien fungsi tujuan dual.
- 3) Semua kolom primal menjadi *constraint* dual.
- 4) Semua *constraint* primal menjadi *decision of variables* dual.
- 5) Koefisien *constraint* dan variabel primal menjadi koefisien yang berkorespondensi dengan *constraint* dual.

## 9. RINGKASAN TUGAS AKHIR

Beberapa algoritma yang dapat digunakan untuk menyelesaikan masalah penugasan adalah algoritma *Hungarian*. Namun untuk dapat menyelesaikan masalah penugasan dinamis, algoritma *Hungarian* relatif belum efisien. Algoritma yang relatif lebih baik dalam menyelesaikan masalah penugasan dinamis adalah algoritma *dynamic Hungarian*. Namun demikian algoritma yang dapat digunakan untuk menyelesaikan masalah penugasan dinamis tidak terbatas hanya pada algoritma itu saja. Maka dari itu akan dilakukan kajian komprehensif untuk menyelesaikan masalah penugasan dinamis.

Kajian komprehesif akan menghasilkan rancangan algoritma yang efisien dalam menyelesaikan masalah penugasan dinamis. Salah satu situs *Online Judge* telah menyediakan fasilitas yang dapat digunakan untuk memeriksa kebenaran dan efisiensi algoritma yang telah dirancang [1]. Dalam laman situs tersebut dijelaskan permasalahan yang karakteristiknya sama dengan masalah penugasan dinamis. Diberikan masukan berupa matriks bobot berukuran  $N \times N$ . Terdapat  $M$  operasi dimana tiap satu operasi adalah salah satu dari operasi berikut:

- a. Mengubah nilai bobot satu elemen pada matriks.
- b. Mengubah semua nilai bobot pada satu kolom matriks.
- c. Mengubah semua nilai bobot pada satu baris matriks.
- d. Menambah ukuran matriks menjadi  $(N + 1) \times (N + 1)$  dimana  $N \times N$  adalah ukuran matriks yang sekarang.
- e. Mendapatkan total biaya susunan penugasan yang optimal.

Beberapa batasan dari permasalahan tersebut adalah sebagai berikut:

- a. Nilai  $N$  tidak akan pernah melebihi 100 termasuk pada saat proses  $M$  operasi berjalan.
- b. Nilai  $M$  tidak akan pernah melebihi 10000.
- c. Waktu maksimal dari program untuk menyelesaikan permasalahan ini adalah 10 detik.

Dengan adanya fasilitas tersebut diharapkan proses pengujian kebenaran dan efisiensi rancangan algoritma yang dihasilkan berjalan dengan lancar dan hasil yang didapat adalah yang seakurat mungkin.

## 10. METODOLOGI

### a. Penyusunan proposal tugas akhir

Penyusunan proposal tugas akhir merupakan tahap awal dalam proses pengerjaan tugas akhir. Dalam proposal ini diajukan bagaimana menganalisis dan merancang algoritma yang efisien untuk menyelesaikan masalah penugasan dinamis.

### b. Studi literatur

Tahap berikutnya adalah mencari informasi dan studi literatur apa saja yang bisa dijadikan referensi untuk melakukan pengerjaan tugas akhir. Literatur yang digunakan adalah beberapa *paper* dan buku yang berhubungan dengan topik yang dibahas.

### c. Implementasi perangkat lunak

Implementasi merupakan tahap untuk menerapkan algoritma yang dipilih ke dalam bahasa pemrograman. Algoritma yang telah dipilih diimplementasikan dengan bahasa pemrograman C++.

### d. Pengujian dan evaluasi

Proses pengujian akan dilakukan dengan mengunggah *source code* hasil implementasi pada situs *Online Judge* SPOJ. Diharapkan hasil yang didapat adalah implementasi yang baik dan benar dengan konsumsi waktu dan *memory* yang lebih efisien dari implementasi beberapa *user* yang telah menyelesaikan masalah ini sebelumnya. Sampai pada saat proposal ini diajukan, masih terdapat dua *user* yang telah menyelesaikan masalah ini.

### e. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan
  - a. Latar Belakang
  - b. Rumusan Masalah
  - c. Batasan Tugas Akhir
  - d. Tujuan
  - e. Metodologi
  - f. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

## 11. JADWAL KEGIATAN

Tahapan	2013 – 2014															
	Oktober - November				Desember - Januari				Februari - Maret				April - Mei			
Penyusunan Proposal	■															
Studi Literatur	■	■	■	■	■	■										
Implementasi							■	■	■	■	■	■				
Pengujian dan Evaluasi										■	■	■	■			
Penyusunan Buku											■	■	■	■	■	■

## 12. DAFTAR PUSTAKA

- [1] SPOJ. (2012, November) Dynamic Assignment Problem. [Online].  
<http://www.spoj.com/problems/DAP/>
- [2] Wikipedia. (2013, October) Linear Programming. [Online].  
[http://en.wikipedia.org/wiki/Linear\\_programming](http://en.wikipedia.org/wiki/Linear_programming)
- [3] Wikipedia. (2013, October) Hungarian Algorithm. [Online].  
[http://en.wikipedia.org/wiki/Hungarian\\_algorithm](http://en.wikipedia.org/wiki/Hungarian_algorithm)
- [4] Wikipedia. (2013, July) Bipartite Graph. [Online].  
[http://en.wikipedia.org/wiki/Bipartite\\_graph](http://en.wikipedia.org/wiki/Bipartite_graph)
- [5] Wikipedia. (2013, August) Assignment Problem. [Online].  
[http://en.wikipedia.org/wiki/Assignment\\_problem](http://en.wikipedia.org/wiki/Assignment_problem)
- [6] Christos H. Papadimitriou and Kenneth Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. New York: Dover Publication, 1998.
- [7] Eugene L. Lawler, *Combinatorial Optimization: Networks and Matroids*. New York: Holt, Reinehart and, 1976.
- [8] G. Ayorkor Korsah, Anthony Stentz, and M Bernardine Dias, "The Dynamic Hungarian Algorithm for the Assignment Problem with Changing Costs," Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-07-27, July 2007.