

# Penerapan Algoritma *Greedy by Density* dalam Pengerjaan Tugas Besar di Teknik Informatika ITB sebagai Persoalan *Binary Knapsack*

Tubagus Andhika Nugraha (13510007)<sup>1</sup>

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

<sup>1</sup>andhika.nugraha@gmail.com

**Abstrak**—Tugas besar adalah salah satu ciri khas mata kuliah di program studi Teknik Informatika ITB. Seorang mahasiswa Teknik Informatika dituntut untuk menyelesaikan lebih dari satu tugas besar untuk setiap mata kuliah yang diambilnya di sela-sela kegiatannya yang lain. Makalah ini membahas bagaimana strategi algoritma dapat diterapkan untuk menjadwalkan pengerjaan tugas besar tersebut dengan memanfaatkan persoalan *knapsack*.

**Index Terms**—Strategi Algoritma, Tugas Besar, Teknik Informatika, *Greedy*, *Knapsack*

## I. PENDAHULUAN

Setiap program studi di Institut Teknologi Bandung memiliki ciri khasnya sendiri, baik dari segi kemahasiswaan di himpunan mahasiswanya maupun dari segi akademik. Dari segi kemahasiswaan, misalnya, himpunan mahasiswa program studi Teknik Pertambangan, Teknik Perminyakan, Teknik Geologi, dan Teknik Geofisika dikenal sebagai himpunan mahasiswa yang ‘keras’, dan mahasiswanya seringkali distereotipekan sesuai dengan citra yang didapatkan oleh himpunan mahasiswanya. Namun, karena pada dasarnya perguruan tinggi adalah tempat menuntut ilmu di mana mahasiswa kuliah untuk mendapatkan *hard skills*, kegiatan akademik yang diselenggarakan sebuah program studi berdampak pada citra mahasiswa yang kuliah di program studi tersebut. Salah satu program studi yang memiliki ciri khas yang kental adalah Teknik Informatika.

Ciri khas kegiatan akademik di Teknik Informatika ITB adalah tugas besar. Meski banyak mata kuliah di program studi lain yang menggunakan tugas besar sebagai salah satu parameter penilaian dan/atau kaskas pengajaran, tugas besar di Teknik Informatika memiliki reputasi tersendiri karena hampir seluruh mata kuliah wajib di Teknik Informatika menugaskan mahasiswanya untuk mengerjakan tugas besar. Seringkali, di akhir semester, batas waktu pengumpulan (*deadline*) untuk beberapa tugas besar mata kuliah bertumpuk dan menjadi masalah bagi mahasiswa yang mengerjakannya. Bertumpuknya batas waktu pengumpulan tersebut bukanlah sebuah hal

yang tidak disengaja, karena salah satu tujuan pemberian tugas besar di Teknik Informatika, selain untuk menumbuhkan pemahaman tentang mata kuliah, adalah melatih *soft skills* manajemen waktu. Selain manajemen waktu, pemberian tugas besar juga melatih kemampuan bekerja dalam kelompok dan pengaturan beban kerja, karena hampir seluruh tugas besar di Teknik Informatika dikerjakan secara berkelompok. Sebagian mata kuliah bahkan membagikan kelompok secara acak sehingga kemampuan adaptasi mahasiswa juga diasah.

Namun, tugas besar bukanlah satu-satunya tanggung jawab yang harus diambil oleh seorang mahasiswa Teknik Informatika ITB. Layaknya seorang mahasiswa pada umumnya, seorang mahasiswa Teknik Informatika juga dituntut untuk aktif berorganisasi dan berkegiatan di luar perkuliahan. Kegiatan-kegiatan tersebut juga membutuhkan waktu dan kemampuan manajemen waktu yang baik. Namun, tidak jarang mahasiswa Teknik Informatika dengan segala tanggung jawab akademik dan nonakademik yang harus dijalaninya dapat menangani semua itu dengan baik. Manajemen waktu menjadi sebuah hal yang sulit, dan pengerjaan tugas besar pun menjadi tidak ideal.

Namun, jika diamati, persoalan manajemen waktu tidak jauh berbeda dengan persoalan *knapsack*. Terdapat sejumlah kegiatan atau pekerjaan yang masing-masing memiliki nilai-nilai tertentu yang harus dimuat dalam jangka waktu yang tersedia dengan tujuan mendapatkan hasil terbaik – serupa dengan persoalan *knapsack* di mana sejumlah barang dengan berat tertentu harus dimasukkan ke dalam karung (*knapsack*) untuk mendapatkan hasil tertentu. Makalah ini mencoba menganalogikan pengerjaan tugas besar di Teknik Informatika sebagai persoalan *binary knapsack* dan bagaimana strategi algoritma dapat diterapkan untuk menyelesaikan persoalan tersebut.

## II. TUGAS BESAR DI TEKNIK INFORMATIKA ITB

Terdapat setidaknya satu tugas besar untuk sebagian besar mata kuliah wajib dan pilihan utama di Teknik

Informatika ITB. Sebuah ‘tugas besar’ dalam lingkup makalah ini berarti tugas yang oleh dosen pengajar mata kuliah disebut secara harfiah sebagai ‘tugas besar’. Beberapa mata kuliah, di antaranya IF3051 Strategi Algoritma, juga memberikan tugas-tugas lain yang disebut tugas kecil. Tugas besar dapat dikerjakan berkelompok maupun perseorangan, namun kebanyakan dikerjakan berkelompok. Satu mata kuliah dapat memberikan lebih dari satu tugas besar selama satu semester, di antaranya IF3051 Strategi Algoritma.

Misalkan sebuah mahasiswa Teknik Informatika 2010 Semester 5 ber-NIM ganjil yang mengambil mata kuliah yang diselenggarakan program studi Teknik Informatika sebagai berikut:

| KODE KULIAH | NAMA MATA KULIAH                 |
|-------------|----------------------------------|
| KU1071      | Pengenalan Teknologi Informasi A |
| IF2091      | Struktur Diskrit                 |
| IF2092      | Probabilitas & Statistika        |
| IF2030      | Algoritma & Struktur Data        |
| EL2010      | Organisasi & Arsitektur Komputer |
| IF2032      | Pemrograman Berorientasi Objek   |
| IF2034      | Basis Data                       |
| IF2036      | Rekayasa Perangkat Lunak         |
| IF2050      | Logika Informatika               |
| IF2052      | Teori Bahasa dan Otomata         |
| IF3035      | Sistem Basis Data                |
| IF3051      | Strategi Algoritma               |
| IF3055      | Sistem Operasi                   |
| IF3057      | Sistem Informasi                 |
| IF3097      | Jaringan Komputer                |

Terdapat lima belas mata kuliah pada daftar mata kuliah di atas. Jika diklasifikasikan berdasarkan ada atau tidaknya tugas besar, maka didapatkan tabel sebagai berikut:

| KODE KULIAH       | NAMA MATA KULIAH                         |
|-------------------|------------------------------------------|
| KU1071            | Pengenalan Teknologi Informasi A         |
| <del>IF2091</del> | <del>Struktur Diskrit</del>              |
| <del>IF2092</del> | <del>Probabilitas &amp; Statistika</del> |
| IF2030            | Algoritma & Struktur Data                |
| EL2010            | Organisasi & Arsitektur Komputer         |
| IF2032            | Pemrograman Berorientasi Objek           |
| <del>IF2034</del> | <del>Basis Data</del>                    |

|                   |                              |
|-------------------|------------------------------|
| IF2036            | Rekayasa Perangkat Lunak     |
| IF2050            | Logika Informatika           |
| IF2052            | Teori Bahasa dan Otomata     |
| <del>IF3035</del> | <del>Sistem Basis Data</del> |
| IF3051            | Strategi Algoritma           |
| IF3055            | Sistem Operasi               |
| IF3057            | Sistem Informasi             |
| IF3097            | Jaringan Komputer            |

Berdasarkan tabel di atas, hanya empat dari lima belas mata kuliah Teknik Informatika hingga semester 5 yang tidak memberikan tugas besar. Ini berarti 73.33% mata kuliah pada kasus di atas menyediakan tugas besar.

Secara umum, terdapat dua jenis *deliverables* tugas besar di Teknik Informatika ITB:

#### Dokumen

Deliverables yang berupa dokumen dapat berupa dokumen untuk rekayasa perangkat lunak atau sistem, atau laporan. Dokumen rekayasa perangkat lunak diberikan oleh di antaranya mata kuliah IF2036 Rekayasa Perangkat Lunak, IF3037 Rekayasa Perangkat Lunak Lanjut. Dokumen rekayasa sistem informasi diberikan oleh di antaranya mata kuliah IF3057 Sistem Informasi. Laporan umumnya dikumpulkan bersama dengan *deliverables* aplikasi. Ada juga mata kuliah seperti IF2052 Teori Bahasa dan Otomata yang memberikan tugas membuat makalah sebagai tugas besar. Mata kuliah IF2091 Struktur Diskrit juga memberikan tugas makalah sebagai tugas akhir mata kuliah tersebut, namun tidak secara tersurat disebut sebagai tugas besar.

Sebuah dokumen terdiri atas beberapa bab dan umumnya didasarkan pada sebuah contoh yang disediakan oleh dosen, yang tataletaknya umumnya bersumber dari IEEE. Sebuah bab dapat memiliki subbab, sebuah subbab dapat memiliki subsubbab, dan seterusnya, hingga membuat sebuah *document outline*.

#### Aplikasi

Tugas besar dengan deliverables berupa aplikasi diberikan pada banyak mata kuliah, dengan kaskas dan bahasa pemrograman yang berbeda-beda, mulai dari Assembly pada mata kuliah EL2010 Organisasi dan Arsitektur Komputer, C++ pada mata kuliah IF2032 Pemrograman Berorientasi Objek, hingga C# pada mata kuliah IF3051 Strategi Algoritma. Umumnya bahasa pemrograman yang digunakan untuk tugas besar tidak diajarkan di dalam kelas. Melalui mata kuliah KU1071 Pengenalan Teknologi Informasi A dan IF2030 Algoritma dan Struktur Data, mahasiswa diajarkan untuk membuat aplikasi secara modular, dan modularitas tersebut merupakan ciri khas dari *deliverables* aplikasi. Kode sumber sebuah aplikasi dapat dipisahkan menjadi beberapa modul, yang masing-masing modul dapat dipisahkan menjadi berkas masing-masing atau digabung

menjadi sebuah modul yang baru. Sebuah modul bisa dimaksudkan sebagai sebuah pustaka, sebuah kelas, sebuah fungsi, atau bagian aplikasi lainnya. Jika modularitas dilaksanakan dengan baik, sebuah modul tidak saling bergantung pada modul lainnya.

Berdasarkan dua jenis deliverables di atas, sebuah tugas besar di Teknik Informatika ITB dapat dipisahkan menjadi dua jenis: dokumen saja serta dokumen dan aplikasi. Karena dokumen dan aplikasi sama-sama bersifat modular, serta untuk pembagian kerja di kelompok umumnya memanfaatkan modularitas tugas besar juga, sebuah tugas besar di Teknik Informatika dapat digeneralisasi menjadi sebuah struktur data yang mengandung beberapa modul. Masing-masing modul memiliki tingkat kesulitan dan waktu penyelesaian sendiri. Sebuah tugas besar yang menjadi induk modul mengandung sebuah tanggal batas pengumpulan.

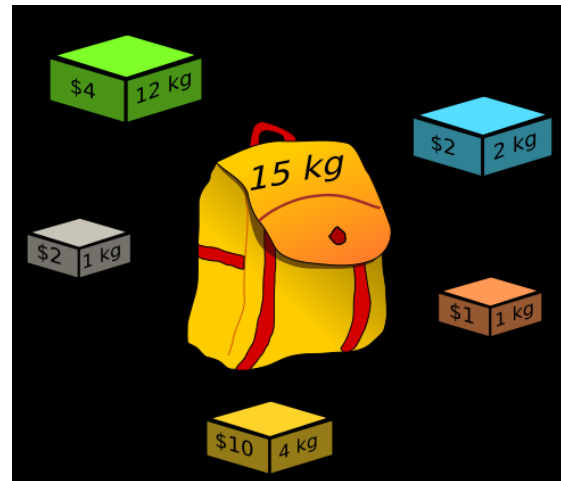
Jika pada saat tugas besar diberikan, tidak ada tugas besar lain yang perlu dikerjakan, maka persoalan pengerjaan tugas besar menjadi relatif mudah. Namun, pada akhir semester, skenario tersebut hampir tidak mungkin. Semester 5 di Teknik Informatika, misalnya, dikenal sebagai semester terberat di Teknik Informatika, karena banyaknya tugas besar yang diberikan. Bagian yang paling ‘menantang’ dari Semester 5 adalah dua minggu terakhir perkuliahan. Pada dua minggu ini, setidaknya untuk Semester I-2012/2013, terdapat bukan satu, bukan dua, bukan tiga, tapi empat tugas besar yang batas waktu pengumpulannya berdekatan. Keempat tugas besar tersebut adalah:

1. Tugas Besar 3 IF3051 Strategi Algoritma:  
Aplikasi Algoritma Pencocokan String pada  
*Simple ChatBot*.  
*Deadline: 7 Desember 2012*
2. Revisi akhir Tugas Besar IF3055 Sistem  
Informasi: Pengembangan sistem informasi untuk  
sebuah perusahaan  
*Deadline: 9 Desember 2012*
3. Tugas besar IF3091 Sistem Operasi:  
Implementasi *filesystem* sederhana  
*Deadline: 8 Desember 2012*
4. Tugas besar IF3097 Jaringan Komputer:  
Implementasi FTP sederhana  
*Deadline: 7 Desember 2012*

Keempat tugas tersebut memiliki waktu kurang lebih dua minggu mulai dari pemberian tugas hingga batas waktu pengumpulan. Keempat tugas tersebut dikerjakan secara berkelompok, dan bagi sebagian besar orang, kelompok setiap Tugas Besar berbeda. Persoalannya adalah, bagaimana menentukan tugas besar apa yang sebaiknya dikerjakan dalam satu kurun waktu tertentu?

### III. PERSOALAN *BINARY KNAPSACK*

Persoalan *knapsack* adalah sebuah persoalan optimisasi kombinatorial. Terdapat beberapa jenis persoalan *knapsack*, salah satu di antaranya 1/0 *knapsack*, yang juga dikenal sebagai persoalan *binary knapsack*.



Diberikan  $n$  buah objek dan sebuah knapsack dengan kapasitas bobot  $K$ . Setiap objek memiliki properti bobot (*weight*)  $w_i$  dan keuntungan (*profit*)  $p_i$ .

Bagaimana memilih memilih objek-objek yang dimasukkan ke dalam knapsack sedemikian sehingga memaksimalkan keuntungan. Total bobot objek yang dimasukkan ke dalam knapsack tidak boleh melebihi kapasitas knapsack.

Solusi persoalan *binary knapsack* dinyatakan sebagai

$$X = \{x_1, x_2, \dots, x_n\}$$

di mana

$x_i = 1$ , jika objek ke- $i$  dipilih,

$x_i = 0$ , jika objek ke- $i$  tidak dipilih. [1]

Pemecahan persoalan *binary knapsack* dapat menggunakan beberapa strategi algoritma, antara lain *exhaustive search*, algoritma Greedy, dan pemrograman dinamis. Secara umum, formalisasi persoalan adalah sebagai berikut:

$$\text{Maksimasi } F = \sum_{i=1}^n p_i x_i$$

dengan kendala (*constraint*)

$$\sum_{i=1}^n w_i x_i \leq K$$

yang dalam hal ini,  $x_i = 0$  atau  $1$ ,  $i = 1, 2, \dots, n$

#### IV. PENJADWALAN Pengerjaan Tugas Besar SEBAGAI PERSOALAN *KNAPSACK*

Untuk memecahkan persoalan yang dinyatakan dalam Bab II, diperlukan analogi antara tugas besar di Teknik Informatika ITB dengan persoalan *knapsack*. Komponen persoalan tugas besar adalah:

1. Modul, bab, atau bagian tugas besar yang harus dikerjakan
  - a. Waktu yang dibutuhkan untuk mengerjakan modul
  - b. Waktu yang tersisa sebelum tenggat waktu tugas besar
2. Waktu yang dimiliki oleh mahasiswa yang dialokasikan untuk mengerjakan tugas besar

Komponen pada persoalan *knapsack* adalah:

1. Barang yang ingin diambil
  - a. Beratnya
  - b. Keuntungannya
2. Kapasitas *knapsack*

Terdapat kemiripan antara komponen pada persoalan tugas besar dan persoalan *knapsack*. Modul, bab, atau bagian tugas besar yang harus dikerjakan dapat dianalogikan sebagai barang yang diambil pada persoalan *knapsack*, dan waktu yang tersedia dapat dianalogikan dengan kapasitas *knapsack*. Waktu pada persoalan tugas besar dapat dianalogikan sebagai berat pada persoalan *knapsack*. Waktu yang tersisa sebelum tenggat waktu dapat dianalogikan sebagai keuntungan pada persoalan *knapsack*, karena tugas besar yang tenggat waktunya lebih dekat lebih urgen untuk dikerjakan.

Namun, agar analogi tersebut dapat direalisasikan, ada dua asumsi yang diambil. Asumsi pertama adalah bahwa pengerjaan sebuah modul atau bab bersifat atomik, yaitu tidak dapat dipecah. Tentunya pada kenyataannya sebuah modul dapat dicicil menjadi pengerjaan yang lebih singkat namun banyak. Agar analogi dapat berjalan, istilah modul diubah menjadi pecahan tugas besar, dalam ukuran apapun, yang bersifat atomik. Sebuah modul dapat bersifat dependen pada modul lainnya, dalam hal di mana jika sebuah modul berkegantungan pada modul lain yang belum dikerjakan, modul tersebut belum dapat dikerjakan.

Asumsi kedua adalah bahwa mahasiswa dapat dengan tepat mengira-ngira waktu yang dibutuhkan untuk mengerjakan sebuah modul tersebut. Meski sulit mendapatkan perkiraan yang tepat, namun dengan banyaknya tugas besar yang sudah dikerjakan oleh mahasiswa sebelumnya, mahasiswa seharusnya memiliki kemampuan yang cukup untuk mendapatkan perkiraan waktu tersebut.

Asumsi ketiga adalah bahwa mahasiswa memiliki

kemampuan untuk bertukar moda pikiran antara satu tugas besar dengan tugas besar lainnya dengan baik, atau waktu untuk melakukan pertukaran moda pikiran tersebut sudah dihitung dalam peubah waktu.

Jika kedua asumsi tersebut kita generalisasi, asumsi yang digunakan hanya satu, yaitu bahwa mahasiswa memiliki kemampuan manajemen waktu dasar untuk mempartisi sebuah tugas dengan baik dengan mengetahui waktu yang dibutuhkan untuk mengerjakannya.

Dengan demikian, analogi persoalan tugas besar sebagai persoalan *binary knapsack* dapat berlaku. Persoalan tersebut menjadi:

$$\text{Maksimasi } F = \sum_{i=1}^n p_i x_i$$

dengan kendala (*constraint*)

$$\sum_{i=1}^n w_i x_i \leq K$$

yang dalam hal ini,  $x_i = 0$  atau  $1$ ,  $i = 1, 2, \dots, n$

$p$  adalah waktu yang tersisa sebelum tenggat waktu tugas, dan  $w$  adalah waktu yang dibutuhkan untuk mengerjakan tugas.

#### V. ALGORITMA *GREEDY BY DENSITY*

Greedy adalah algoritma pemecahan masalah yang fokus untuk mendapatkan solusi optimal untuk langkah tertentu namun tidak memikirkan apakah solusi tersebut optimal untuk rangkaian solusi secara keseluruhan.

*Greedy by density* adalah algoritma Greedy yang penentuan bobot nilainya bergantung pada densitas yang ditentukan. Densitas adalah perbandingan nilai dan bobot. Nilai dan bobot tergantung pada masing-masing kasus, di mana dalam persoalan *knapsack* mewakili keuntungan dan berat. [1] [2]

Nilai yang dimaksimasi dalam algoritma Greedy by Density adalah:

$$F = \sum_{i=1}^n \frac{p_i}{w_i} x_i$$

#### VI. PEMECAHAN PERSOALAN

Untuk memecahkan persoalan tugas besar, pertama perlu diketahui kasus yang akan dipecahkan. Misalkan tanggal saat ini adalah 30 November 2012 dan seorang mahasiswa bernama Budi mengambil mata kuliah Strategi Algoritma, Sistem Informasi, Jaringan Komputer, dan Sistem Informasi. Keempat mata kuliah tersebut memiliki tenggat waktu dalam sembilan hari ke depan, dengan rincian sebagai berikut:

1. Tugas Besar 3 IF3051 Strategi Algoritma:  
Aplikasi Algoritma Pencocokan String pada

Simple ChatBot.

Deadline: 7 hari lagi

2. Revisi akhir Tugas Besar IF3055 Sistem Informasi: Pengembangan sistem informasi untuk sebuah perusahaan  
Deadline: 9 hari lagi
3. Tugas besar IF3091 Sistem Operasi: Implementasi *filesystem* sederhana  
Deadline: 8 hari lagi
4. Tugas besar IF3097 Jaringan Komputer: Implementasi FTP sederhana  
Deadline: 7 hari lagi

Budi bersama kelompoknya telah menentukan bagian apa saja yang harus dikerjakan oleh Budi, dan Budi sudah membuat perkiraan waktu yang dibutuhkan untuk mengerjakan masing-masing bagian tersebut.

Bagian-bagian tersebut adalah:

| Bagian                                        | Waktu |
|-----------------------------------------------|-------|
| 1. Antarmuka dasar ChatBot (Stima)            | 3 jam |
| 2. Struktur data dasar <i>filesystem</i> (OS) | 5 jam |
| 3. Struktur dasar server FTP (Jarkom)         | 2 jam |
| 4. Implementasi fitur pada SI                 | 4 jam |
| 5. Pemeriksaan editorial dokumen SI           | 1 jam |

Budi perlu mengerjakan bagian-bagian lain juga pada masing-masing tugas besar, namun keempat bagian di atas adalah bagian-bagian yang bisa dikerjakan saat ini. Bagian-bagian tugas tersebut berkorespondensi dengan tenggat waktu tugas besar sebagai berikut:

| Bagian                                        | Tenggat waktu |
|-----------------------------------------------|---------------|
| 1. Antarmuka dasar ChatBot (Stima)            | 7 hari        |
| 2. Struktur data dasar <i>filesystem</i> (OS) | 8 hari        |
| 3. Struktur dasar server FTP (Jarkom)         | 7 hari        |
| 4. Implementasi fitur pada SI                 | 9 hari        |
| 5. Pemeriksaan editorial dokumen SI           | 9 hari        |

Budi ingin menentukan tugas apa saja yang harus dia kerjakan dalam 10 jam ke depan ( $K = 10$ ). Dengan memanfaatkan algoritma Greedy by Density, didapatkan tabel berikut:

| Bagian | p | w | p/w  | X |
|--------|---|---|------|---|
| Stima  | 7 | 3 | 2,33 | 1 |
| OS     | 8 | 5 | 1,6  | 0 |
| Jarkom | 7 | 2 | 3,5  | 1 |
| SI 1   | 9 | 4 | 2,25 | 1 |
| SI 2   | 9 | 1 | 9    | 1 |

Berdasarkan perhitungan tersebut, bagian yang harus dikerjakan Budi dalam sepuluh jam adalah:

1. Antarmuka dasar ChatBot
2. Struktur dasar server FTP

3. Implementasi fitur pada SI
4. Pemeriksaan editorial dokumen SI

Waktu yang dibutuhkan untuk mengerjakan semuanya adalah 10 jam.

## V. KESIMPULAN DAN SARAN

Berdasarkan paparan di atas, dapat disimpulkan bahwa algoritma Greedy by Density dapat digunakan untuk menentukan bagian tugas besar apa sajakah yang harus dikerjakan dalam kurun waktu tertentu. Namun, makalah ini belum dapat menjawab apakah keputusan yang dihasilkan bersifat baik atau buruk, sehingga ada baiknya jika dilakukan kajian mengenai dampak keputusan yang diambil dari metode yang dijelaskan dalam makalah ini.

## VII. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Bapak Rinaldi Munir dan Bu Masayu Leilia Khodra sebagai dosen pengajar mata kuliah IF3051 Strategi Algoritma. Penulis juga mengucapkan terima kasih kepada dosen-dosen di Teknik Informatika ITB yang telah memberikan tugas besar kepada mahasiswanya sehingga menjadi inspirasi untuk makalah ini.

## REFERENSI

- [1] Rinaldi Munir, "Diktat Kuliah Strategi Algoritma", 2011
- [2] Rendy Bambang Junior, "Penerapan *Greedy by Density* dalam Pemilihan Menu Warteg sebagai Permasalahan *Knapsack*", 2011

## 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, 21 Desember 2012



Tubagus Andhika Nugraha  
13510007