

Penerapan Algoritma *Greedy* dalam Pengisian Rencana Studi Semester di ITB sebagai Persoalan *Binary Knapsack*

Muhammad Ikhsan (13511064)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13511064@std.stei.itb.ac.id

Abstrak—ITB sebagai perguruan tinggi mengadakan pendidikan tinggi dan diwakili oleh kuliah tiap semester. Mahasiswa selaku peserta didik yang mengikuti kegiatan perkuliahan mengambil mata kuliah yang akan diambil dalam semester itu di awal semester. Namun mata kuliah di program studi tertentu seperti Program Studi Teknik Informatika memiliki karakteristik mata kuliah ‘paket’ sehingga mahasiswa tidak dapat memilih mata kuliah secara bebas dan bertanya-tanya apakah ‘paket’ yang dipilihkan oleh program studi ini adalah yang tercepat dan dapat sebanyak-banyaknya dikarenakan beberapa mahasiswa ingin lulus lebih cepat. Banyaknya mata kuliah yang dapat diambil dapat dianalogikan sebagai persoalan *binary Knapsack problem* dan dapat diselesaikan dengan Algoritma *Greedy*.

Index Terms—Pengisian Rencana Studi, SKS, Knapsack, Greedy

I. PENDAHULUAN

Institut Teknologi Bandung adalah salah satu perguruan tinggi di Indonesia yang mendapatkan amanat untuk melakukan pendidikan tinggi. ITB terdiri dari program Sarjana, Magister dan Doktor. Kuliah di Program Sarjana ITB secara normal dilaksanakan selama 4 tahun atau 8 semester, dan maksimal sebanyak 6 tahun, dibagi atas bagian Tahap Persiapan Bersama selama 1 tahun (maksimal 2 tahun), dan Program Sarjana selama 3 tahun (maksimal 4 tahun).

Setiap awal semester, semua program studi membuka pendaftaran mata kuliah, atau biasa disebut Pengisian Rencana Studi. Rencana Studi dapat diisi oleh daftar mata kuliah yang akan diambil mahasiswa pada semester tersebut. Mata kuliah ada yang memiliki prasyarat mata kuliah lainnya (*prerequisite*) dan ada yang memiliki syarat semisal minimal jumlah SKS yang sudah diambil. Kemudian juga, ada batas maksimal jumlah SKS yang dapat diambil dalam satu semester, yaitu maksimal 22 SKS apabila IP semester terakhir > 2.75 dan maksimal 24 SKS apabila IP semester terakhir > 3.15 . Setiap program studi memiliki karakteristik yang berbeda terhadap mata kuliah yang harus diambil, salah satunya yaitu dapat dilihat di program studi Teknik Informatika.

Program Studi Teknik Informatika memiliki beberapa tipe mata kuliah, yaitu mata kuliah TPB, mata kuliah wajib program studi, mata kuliah pilihan program studi, mata kuliah pilihan nonprogram studi dan mata kuliah wajib ITB. Setiap semester memiliki karakteristik, apakah sudah bisa mengambil mata kuliah wajib, dan lainnya.

Pengambilan mata kuliah biasanya diambil pada awal semester, tepatnya beberapa hari sebelum hari pertama semester baru dimulai. Rangkaian pendaftaran biasanya berisi tentang pertemuan dengan Dosen Wali (yang berikutnya pertemuan ini disebut *perwalian*) dan pengisian form rencana studi, serta persetujuan dari wali mengenai rencana studi.

Dalam memilih mata kuliah, sangat banyak pertimbangan, salah satunya yaitu beban SKS yang diambil, serta jenis mata kuliah (wajib prodi, wajib ITB, dsb), mata kuliah yang dibuka semester tersebut, total SKS yang diperbolehkan dalam satu semester yang berarti bergantung pada total Indeks Prestasi (IP) yang didapat pada semester terakhir, serta jadwal kuliah.

Mahasiswa terkadang bingung untuk memilih mata kuliah apa saja yang harus diambilnya dalam semester, dikarenakan biasanya program studi sudah memberikan ‘paket’ SKS dalam satu semester. Tetapi disisi lain, mahasiswa tersebut ragu apakah dengan ‘paket’ yang sudah dijadwalkan oleh Program Studi seperti itu, mahasiswa tersebut dapat lulus cepat. Mahasiswa pun ingin memilih sendiri mata kuliahnya, sebanyak-banyaknya dalam satu semester agar dapat lulus dengan cepat.

Namun jika diamati, memilih sebanyak-banyaknya adalah perwujudan dari Algoritma *Greedy*. Algoritma *Greedy* biasa digunakan untuk masalah optimasi, bagaimana memaksimalkan. Selain itu, memilih sebanyak-banyaknya pilihan dengan sebuah bobot (dalam hal ini SKS) dengan batasan tertentu cukup mirip dengan persoalan *knapsack binary (0/1) problem* yang pada dasarnya memilih sebanyak-banyaknya barang dengan profit paling menguntungkan kedalam sebuah tas. Makalah ini berusaha membahas pemilihan rencana studi di Program Studi Teknik Informatika dengan pendekatan *algoritma greedy* dan *knapsack binary problem*.

II. DASAR TEORI

1. Algoritma Greedy

Algoritma *Greedy* berasal dari kata *Greedy*. Secara harafiah *greedy* artinya rakus atau tamak, yaitu sifat yang berkonotasi negatif. Orang yang tamak biasanya akan mengambil sebanyak mungkin makanan (atau harta lainnya) yang tersedia tanpa memikirkan konsekuensi kedepan. Prinsip *greedy* adalah : "*take what you can get now*". Ambil apa yang dapat anda peroleh sekarang! Prinsip ini juga diadopsi dalam pemecahan masalah optimasi, tetapi tentu dalam konteks positif. (Munir, Rinaldi. 2009)

Algoritma *greedy* membentuk solusi langkah per langkah (step by step) Terdapat banyak pilihan yang perlu dieksplorasi pada setiap langkah solusi. Oleh karena itu, pada setiap langkah harus dibuat keputusan terbaik dalam menentukan pilihan. Keputusan yang telah diambil pada suatu langkah tidak dapat diubah lagi pada langkah selanjutnya. (Munir, Rinaldi. 2009)

Pendekatan yang digunakan di dalam algoritma *greedy* adalah membuat pilihan yang "tampaknya" memberikan perolehan terbaik, yaitu dengan membuat pilihan optimum lokal (local optimum) pada setiap langkah dengan harapan bahwa sisanya mengarah ke solusi optimum global (global optimum). (Munir, Rinaldi. 2009)

Skema Umum Algoritma Greedy

Persoalan optimasi dalam konteks algoritma *greedy* disusun oleh elemen-elemen sebagai berikut :

1. Himpunan kandidat, C.
Himpunan ini berisi elemen-elemen pembentuk solusi. Contohnya adalah himpunan koin, himpunan job yang akan dikerjakan, himpunan simpul dalam graf, dan lain-lain. Pada setiap langkah, satu buah kandidat diambil dari himpunannya.
2. Himpunan solusi, S.
Berisi kandidat-kadidat yang terpilih sebagai solusi persoalan. Dengan kata lain, himpunan solusi adalah himpunan bagian dari himpunan kandidat.
3. Fungsi seleksi - dinyatakan dengan predikat SELEKSI
Fungsi yang pada setiap langkah memiliki kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertmbangkan lagi

pada langkah selanjutnya. Biasanya setiap kandidat, x, di-assign sebuah nilai numerik, dan fungsi seleksi memilih x yang mempunyai nilai terbesar atau memilih x yang mempunyai nilai terkecil.

4. Fungsi kelayakan (*feasible*) - dinyatakan dengan predikat LAYAK
Fungsi yang memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbenetuk tidak melanggar kendala (*constraints*) yang ada. Kandidat yang layak dimasukkan ke dalam himpunan solusi, sedangkan kandidat yang tidak layak dibuang dan tidak pernah dipertmbangkan lagi.
5. Fungsi obyektif
Fungsi yang memaksimumkan atau meminimumkan nilai solusi (misalnya panjang lintasan, keuntungan, dan lain-lain)
(Munir, Rinaldi. 2009)

```
function greedy(input C: himpunan_kandidat)→  
  himpunan_kandidat  
{ Mengembalikan solusi dari persoalan  
  optimasi dengan algoritma greedy  
  Masukan: himpunan kandidat C  
  Keluaran: himpunan solusi yang bertipe  
  himpunan_kandidat  
}  
Deklarasi  
x : kandidat  
S : himpunan_kandidat  
Algoritma:  
S ← {} { inisialisasi S dengan kosong  
}  
while (not SOLUSI(S)) and (C ≠ {}) do  
  x ← SELEKSI(C) { pilih sebuah  
  kandidat dari C }  
  C ← C - {x} { elemen  
  himpunan kandidat berkurang satu }  
  if LAYAK(S ∪ {x}) then  
    S ← S ∪ {x}  
  endif  
endwhile  
{SOLUSI(S) or C = { } }  
  
if SOLUSI(S) then  
  return S  
else  
  write('tidak ada solusi')  
endif
```

Skema Umum Algoritma Greedy

2. Binary Knapsack (Knapsack 0/1 Problem)

Knapsack berarti karung, kantung, atau buntelan. Knapsack merupakan salah satu persoalan di bidang Strategi Algoritma.

Diberikan n buah objek dan sebuah knapsack dengan kapasitas bobot K . Setiap objek memiliki properti bobot (weight) w_i dan keuntungan (profit) p_i . Objektif persoalan adalah 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 dinyatakan sebagai vektor n -tupel $X = \{x_1, x_2, \dots, x_n\}$, yang dalam hal ini, $x_i = 1$ jika objek ke- i dimasukkan ke dalam knapsack, atau $x_i = 0$ jika objek ke- i tidak dimasukkan.

Secara matematis, persoalan 0/1-Knapsack dapat dirumuskan sebagai berikut :

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 , $i = 1, 2, \dots, n$

Penyelesaian dengan algoritma *greedy*

Masukkan objek satu per satu ke dalam *knapsack*. Sekali objek dimasukkan ke dalam *knapsack*, objek tersebut tidak bisa dikeluarkan lagi.

Terdapat beberapa strategi *greedy* yang heuristik yang dapat digunakan untuk memilih objek yang akan dimasukkan ke dalam *knapsack*:

1. Greedy by profit.

Pada setiap langkah, pilih objek yang mempunyai keuntungan terbesar.

Mencoba memaksimalkan keuntungan dengan memilih objek yang paling menguntungkan terlebih dahulu.

2. Greedy by weight.

Pada setiap langkah, pilih objek yang mempunyai berat teringan.

Mencoba memaksimalkan keuntungan dengan memasukkan sebanyak mungkin objek ke dalam *knapsack*.

3. Greedy by density.

Pada setiap langkah, *knapsack* diisi dengan objek

yang mempunyai p_i/w_i terbesar.

Mencoba memaksimalkan keuntungan dengan memilih objek yang mempunyai keuntungan per unit berat terbesar.

Pemilihan objek berdasarkan salah satu dari ketiga strategi di atas tidak menjamin akan memberikan solusi optimal.

Algoritma *greedy* tidak selalu berhasil menemukan solusi optimal untuk masalah 0/1 *Knapsack*.

3. Rencana Studi Semester di ITB

ITB menyelenggarakan program pendidikan yang dapat diselesaikan oleh mahasiswa secara tepat waktu dengan kemampuan normal, sesuai dengan kurikulum. Mahasiswa diharapkan selesai tepat waktu.

Pasal 6.1 Waktu Studi Program Sarjana

Waktu studi normal untuk pendidikan Program Sarjana terdiri dari:

- Tahap Tahun Pertama dijadwalkan dalam 2 (dua) semester atau 1 (satu) tahun.
- Tahap Sarjana dijadwalkan dalam 6 (enam) semester atau 3 (tiga) tahun, setelah Tahap Tahun Pertama.

Program Studi Teknik Informatika sebagai salah satu Program Studi yang ada di ITB yang dijadikan bahan eksperimen dalam makalah ini, memiliki struktur kurikulum normal sebagai berikut ini :

Mata kuliah wajib dari Program Studi

Informatics/Computer Science Program Structures

(See Freshman Year)

SOPHOMORE YEAR

Third Semester

No	Code	Subject	Credit
1	IF2110	Algorithm & Data Structure	4
2	IF2120	Discrete Mathematics	3
3	IF2121	Logic of Informatics	3
4	IF2122	Probability & Statistics	3
5	IF2123	Geometric Algebra	3
6	IF2130	Organization & Computer Architecture	3
Total			19

Fourth Semester

No	Code	Subject	Credit
1	IF2210	Object Oriented Programming	3
2	IF2211	Algorithm Strategies	3
3	IF2220	Formal Language Theory and Automata	3
4	IF2230	Operating System	3
5	IF2240	Database	3
6	IF2250	Software Engineering Fundamental	2
Total			17

JUNIOR (THIRD) YEAR

Fifth Semester

No	Code	Subject	Credit
1	IF3110	Web based Application Development	2
2	IF3111	Platform-based Development	2
3	IF3130	Computer Networks	3
4	IF3140	Database Management	2
5	IF3151	Human Computer Interaction	3
6	IF3170	Artificial Intelligence	4
7	IF3150	Project Management of Software	2
Total			18

Sixth Semester

No	Code	Subject	Credit
1	IF3230	Parallel and Distributed Systems	3
2	IF3240	Information System	3
3	IF3250	Project of Software	4
4	IF3260	Computer Graphics	3
5	IF3280	Socio-informatics and Professionalism	3
6	KU206x	Religion & Ethics	2
7	XXxxxx	Environmental Elective	2
Total			20

SENIOR (FOURTH) YEAR

Seventh Semester

No	Code	Subject	Credit
1	IF4090	Industrial Practice	2
2	IF4091	Final Project I & Seminar	2
3	IF4150	Domain Specific Software Engineering	2
4	KU2071	Pancasila and Civic Education	2
5	IF4xxx	Major Elective	3
6	XXxxx	Free Electives	6
Total			17

Eighth Semester

No	Code	Subject	Credit
1	IF4092	Final Project II	4
2	XXxxxx	Non Major Elective	3
3	IF4xxx	Major Elective	3
4	XXxxxx	Free Electives	8
Total			18

Mata kuliah pilihan dari Program Studi :

Electives for Informatics/Computer Science

No	Code	Subject	Credit
1	IF4020	Cryptography	3
2	IF4021	Modeling and Simulation	3
3	IF4022	Advanced Computational Science and Engineering	3
4	IF4030	Advanced Computer Network	3
5	IF4031	Development of Distributed Application	3
6	IF4032	Multimedia System	3
7	IF4033	Information Assurance and Security	3
8	IF4040	Advanced Data Modelling	3
9	IF4041	Database Technology	3
10	IF4042	Information Retrieval Systems	3
11	IF4043	Advanced Information System	3
12	IF4050	Service Oriented Software Development	3
13	IF4051	Component Based Software Engineering	3
14	IF4060	Interaction Engineering	3
15	IF4061	Information and Data Visualization	3
16	IF4062	Development of 3D Graphical Application	3
17	IF4070	Knowledge Representation and Reasoning	3
18	IF4071	Machine Learning	3
19	IF4072	Natural Language (Text and Speech) Processing	3
20	IF4073	Interpretation and Image Processing	3

III. PEMODELAN MASALAH

Dalam persoalan Pengisian Rencana Studi ini, dianalogikan sebagai persoalan Knapsack. Dikarenakan tujuan dari permasalahan ini yaitu bagaimana mengambil

sebanyak-banyaknya mata kuliah dalam satu semester (satu semester dianggap satu knapsack). Persoalan ini dapat diselesaikan dengan algoritma *Greedy* dikarenakan *Greedy* bersifat 'tamak' yaitu mengambil sebanyak-banyaknya yang dapat diambil.

Selain itu juga, dalam *greedy by weight* dan *greedy by profit*, *weight* diwakili oleh jumlah SKS dan *profit* diwakili oleh jarak semester dari semester seharusnya sampai semester kelulusan (semester 8), jadi makin menuju akhir, mata kuliah memiliki nilai *profit* semakin kecil. Hal ini dapat dikatakan keuntungan karena waktu yang dibutuhkan untuk mengambil mata kuliah semakin banyak.

Namun, dalam persoalan ini, asumsi yang diambil yaitu IP mahasiswa yang akan mengambil mata kuliah dianggap selalu stabil yaitu 3.00 tiap semester sehingga . Kuliah yang diambil selalu lulus sehingga tidak mempengaruhi pengambilan. Serta jadwal untuk kuliah apapun tidak akan bertabrakan. Kemudian juga diasumsikan semua mata kuliah dibuka dalam satu semester.

IV. PEMECAHAN PERSOALAN

Awalnya, kita buat dalam tabel, yang berisi mengenai mata kuliah beserta bobot SKS dan keuntungan.

Matakuliah	W	P
Algoritma Struktur Data	4	6
Matematika Diskrit	3	6
Logika Informatika	3	6
Organisasi dan Arsitektur Komputer	3	6
Aljabar Geometri	3	6
Probabilitas dan Statistika	3	6
Pemrograman Berorientasi Objek	3	5
Strategi Algoritma	3	5
Teori Bahasa Formal dan Otomata	3	5
Basis Data	3	5
Sistem Operasi	3	5
Rekayasa Perangkat Lunak	3	5
Interaksi Manusia dan Komputer	3	4
Pengembangan Aplikasi pada Platform Khusus	3	4
Pengembangan Aplikasi berbasis Web	3	4
Jaringan Komputer	3	4
Manajemen Basis Data	3	4
Manajemen Proyek Perangkat Lunak	3	4
Intelegensia Buatan	3	4
Sistem Paralel dan Terdistribusi	3	3
Sistem Informasi	3	3
Proyek Perangkat Lunak	4	3
Grafika	3	3
Sosio-Informatika dan Profesionalisme	3	3
Agama dan Etika	2	3

Mata Kuliah Lingkungan	2	3
Pendidikan Kewarganegaraan	2	2
Kerja Praktek	3	2
Tugas Akhir I	2	2
Rekayasa Perangkat Lunak Spesifik Domain	3	2
Tugas Akhir II	4	1

Beberapa kuliah memiliki prasyarat, sebagaimana dikutip dari STEI Undergraduate Handbook,

- Dasar Pemrograman
 - Algoritma dan Struktur Data
 - Strategi Algoritma
 - Pemrograman Berorientasi Objek
 - Pengembangan Aplikasi Berbasis Web
 - Pengembangan Aplikasi pada Platform khusus
 - Sistem Multimedia
 - Pengembangan Aplikasi 3D
 - Teori Bahasa Formal dan Otomata
 - Inteligencia Buatan
 - Representasi Pengetahuan dan Penalaran
 - Machine Learning
 - Pemrosesan Bahasa Natural
 - Grafika
 - Sistem Multimedia
 - Keamanan dan Asuransi Informasi
 - Informasi dan Visualisasi Data
 - Pengembangan Aplikasi 3D
 - Pemrosesan Citra dan Interpretasi
 - Kriptografi
 - Simulasi dan Modelling
 - Matematika Diskrit
 - Strategi Algoritma
 - Inteligencia Buatan
 - Representasi Pengetahuan dan Penalaran
- Machine Learning
- Pemrosesan Bahasa Natural
- Logika Informatika
 - Inteligencia Buatan
 - Representasi Pengetahuan dan Penalaran
 - Machine Learning
 - Pemrosesan Bahasa Natural
 - Basis Data
 - Manajemen Basis Data
 - Sistem Informasi
 - Pemodelan Data Lanjut
 - Teknologi Basis Data
 - Representasi Pengetahuan dan Penalaran
 - Teori Bahasa Formal dan Otomata
 - Inteligencia Buatan
 - Representasi Pengetahuan dan Penalaran
 - Machine Learning
 - Pemrosesan Bahasa Natural
 - Kriptografi
- Kalkulus IA
 - Probabilitas dan Statistika
 - Inteligencia Buatan
 - Representasi Pengetahuan dan Penalaran
 - Machine Learning
 - Pemrosesan Bahasa Natural
 - Simulasi dan Pemodelan
 - Sistem Penerimaan Informasi
 - Informasi dan Visualisasi Data
 - Machine Learning
 - Aljabar Geometri
 - Grafika
 - Sistem Multimedia
 - Keamanan dan Asuransi Informasi
 - Informasi dan Visualisasi Data
 - Pengembangan Aplikasi 3D
 - Pemrosesan Citra dan Interpretasi
 - Sistem Penerimaan Informasi
 - Pengembangan Aplikasi 3D
 - Organisasi dan Arsitektur Komputer

- Grafika
 - Sistem Multimedia
 - Keamanan dan Asuransi Informasi
 - Informasi dan Visualisasi Data
 - Pengembangan Aplikasi 3D
 - Pemrosesan Citra dan Interpretasi
- Sistem Operasi
 - Jaringan Komputer
 - Sistem Paralel dan Terdistribusi
 - Pengembangan Aplikasi terdistribusi
 - Jaringan Komputer Lanjut
 - Sistem Multimedia
 - Keamanan dan Asuransi Informasi
 - Teknologi Basis Data
- Basis Data
 - Manajemen Basis Data
 - Sistem Informasi
 - Pemodelan Data Lanjut
 - Teknologi Basis Data

Dimana jika sebuah kuliah Y terdapat dibawah penomoran kuliah X dan Z maka dapat dikatakan mata kuliah X dan mata kuliah Z adalah prasyarat mata kuliah Y.

Mata kuliah Y tidak boleh diambil apabila matakuliah X dan Z belum diambil.

VI. PEMBAHASAN

Semisal kita ingin mengambil mata kuliah dalam semester 3, dengan IP = 3.00 dan tidak ada matakuliah sebelumnya yang tidak lulus, dan maksimal dalam satu semester maksimal dapat mengambil 20 SKS, maka didapatkan perhitungan Knapsack sebagai berikut

Matakuliah	P	W	P/W	P	W	P/W	X
Matdis	6	3	2	1	1	1	1
Probstat	6	3	2	1	1	1	1
AlStrukDat	6	4	6/4	1	0	0	0
LogIF	6	3	2	1	1	1	1
OrKom	6	3	2	1	1	1	1
Stima	5	3	5/3	0	1	1	1
OOP	5	3	5/3	0	0	0	1
TBO	5	3	5/3	0	0	0	1
Kriptografi	1	3	1/3	0	0	0	1
AlGeo	6	3	2	1	0	0	1
Total Bobot				19	18	18	19
Total Keuntungan				40	39	39	40

Maka dari hasil diatas didapatkan bahwa dalam semester 3, seorang mahasiswa harus mengambil Matematika Diskrit (Matdis), Probabilitas dan Statistika (Probstat), Algoritma Struktur Data (AlStrukDat), Logika Informatika (LogIF), Organisasi dan Arsitektur Komputer (OrKom) dan Aljabar Geometri (AlGeo) dengan total SKS = 19. Kondisi ini masih memenuhi persyaratan dimana maksimal dalam satu semester pengambilan SKS yaitu berjumlah 20.

VII. KESIMPULAN DAN SARAN

Berdasarkan hasil perhitungan diatas, disimpulkan bahwa Algoritma Greedy untuk Knapsack dapat menentukan matakuliah apa saja yang dapat diambil sebanyak-banyaknya dalam satu semester. Selain itupun, jika dicocokkan dengan struktur kurikulum asli yang berada di bab III, didapat hasil perhitungan Knapsack sama dengan struktur kurikulum yang ditentukan Program Studi. Namun belum dapat disimpulkan apakah cara penyusunan mata kuliah 'paket' dari program studi adalah yang terbaik dan mampu mengambil sebanyak-banyaknya, dan perlu dilakukan pembahasan lebih lanjut.

Penulis menyadari akibat keterbatasan waktu dan bahasan, hasil yang didapat barulah sebatas ini, akan tetapi penulis berharap dimasa depan program ini dapat dikembangkan lebih lanjut dan dapat bermanfaat bagi program pendidikan di ITB, agar mahasiswa dapat lebih bebas memilih mata kuliahnya. Kritik dan saran yang membangun sangat diharapkan.

REFERENCES

- [1] Munir, Rinaldi. 2009. *Diktat Kuliah IF2211 Struktur Algoritma*, Program Studi Teknik Informatika ITB. Bandung
- [2] Tubagus Andhika Nugraha, "Penerapan *Greedy by Density* dalam Pengerjaan Tugas Besar di Teknik Informatika ITB sebagai Persoalan *Binary Knapsack*", 2012.
- [3] "Peraturan Akademik dan Kemahasiswaan Institut Teknologi Bandung", 2013
- [4] "Undergraduate-Handbook-STEI-2013", 2013

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, 19 Desember 2012



Muhammad Ikhsan
13511064