

Penggunaan Algoritma Greedy pada Lapisan Kesadaran Kecerdasan Buatan Total

Muhammad Diaztonto Haryaputra, 13514002

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

13514002@std.stei.itb.ac.id

Abstract—Algoritma Greedy adalah salah satu algoritma yang mangkus dalam memecahkan masalah optimasi. Tidak semua masalah dapat ditemukan solusi terbaik, tetapi algoritma ini berjalan cepat dan tetap menemukan solusi yang baik. Kecerdasan buatan umumnya tidak menggunakan algoritma ini untuk pemecahan solusi karena mesin kecerdasan buatan dibuat untuk mencari solusi terbaik atau bahkan yang lebih baik. Tapi masalah mencari solusi itu adalah masalah mesin *artificial narrow intelligence*. Untuk tahap yang lebih tinggi, *artificial general intelligence*, diperlukan sebuah lapisan kesadaran yang mengontrol kerja semuanya, yang belum tentu optimal dengan menggunakan algoritma yang sama untuk pemecahan solusi pada *artificial narrow intelligence*. Algoritma Greedy dapat dijadikan implementasi yang baik untuk lapisan kesadaran pada *artificial general intelligence* dengan mengutamakan aktifitas yang lebih mudah terlebih dahulu.

Keywords—*algoritma, artificial, general, greedy, intelligence, kecerdasan, mesin, narrow.*

I. PENDAHULUAN

Algoritma Greedy adalah algoritma yang pada setiap tahapnya mengambil nilai yang paling optimal dengan harapan menemukan nilai optimal global. Pada banyak kasus, algoritma Greedy tidak berhasil menemukan solusi optimal, tetapi algoritma ini tetap berjalan dengan cepat. Ini dikarenakan algoritma Greedy tidak memproses semua data, yang akibatnya mungkin saja tidak menemukan nilai optimal global.

Meski pada banyak kasus algoritma Greedy tidak menemukan solusi optimal, tidak sedikit masalah yang dapat dipecahkannya, dan dengan waktu yang singkat pula. Algoritma ini ampuh dalam memecahkan masalah optimasi, yaitu maksimasi dan minimasi, karena kecepatannya yang sangat cepat dibanding solusi optimasi lain seperti program dinamis. Salah satu aplikasi dari algoritma Greedy adalah decision-tree making.

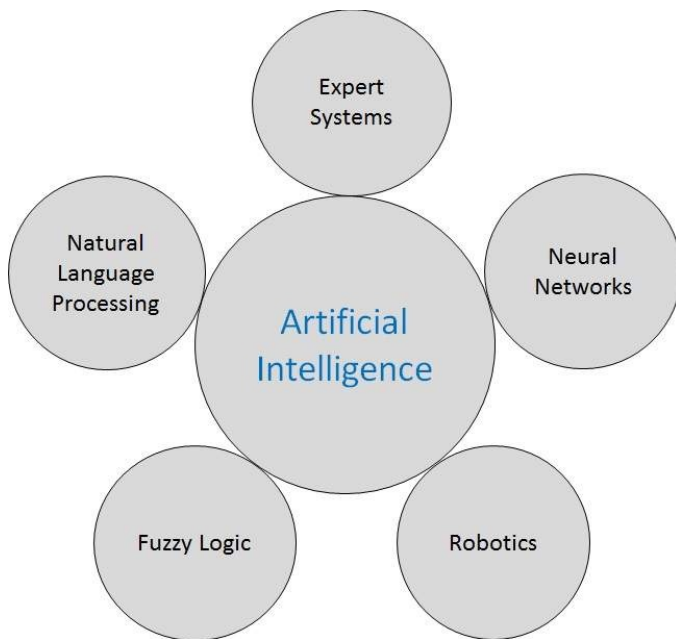
Tanpa disadari kehidupan jaman sekarang telah dipenuhi oleh mesin-mesin pintar yang membantu kita dalam berbagai aktivitas sehari-hari. Mulai dari berkomunikasi hingga transportasi, kini sebuah alat bantu bisa disebut sebagai entitas yang intelijen dalam melaksanakan fungsinya. Dengan

perkembangan teknologi, kini alat-alat bantu tersebut dapat mengkoleksi data penggunaan penggunanya dan mengubah dirinya sedemikian rupa agar lebih disukai oleh pemiliknya. Ya, kita sekarang hidup di tengah-tengah keberadaan mesin intelijen, atau artificial intelligence.

Seperti yang dikemukakan oleh Alan Mathison Turing, pada dasarnya artificial intelligence adalah program yang dapat menulis kembali kodenya untuk membuat dirinya lebih baik. [1] Sebuah tahap lanjut dalam ilmu rekayasa komputasi konvensional yang mengandalkan program yang "statis". Artificial intelligence sendiri memiliki beberapa tahap, yaitu artificial narrow intelligence, artificial general intelligence, dan artificial super intelligence.

- **Artificial narrow intelligence (ANI)** adalah mesin intelijen yang menguasai satu bidang khusus, hanya satu dan satu bidang yang ia kuasai. Contohnya sudah banyak sekali di masa kini, seperti self-driving car, penyusun playlist musik Spotify, chatbot, penyusun news feed Facebook, pemain catur handal, dan lain-lain. Jika misal, sebuah mesin intelijen pemain catur handal ditanyakan tentang prediksi cuaca, maka mesin tersebut tidak bisa memberi respon apa-apa.
- **Artificial general intelligence (AGI)** adalah mesin intelijen yang memiliki kepintaran setara dengan manusia dalam segala bidang.
- **Artificial super intelligence (ASI)** adalah mesin intelijen dengan kecerdasan melebihi manusia. Ini adalah produk superior dari AGI, yang memiliki potensi untuk menciptakan knowledge singularity di mana pola pikir dan kecerdasan mesin seperti ini tidak lagi dimengerti oleh manusia.[2]

Jika kecerdasan buatan dibagi, maka ada beberapa topik utama, seperti dalam gambar di bawah.



Gambar 1.1 Lima topik utama artificial intelligence. Sumber: http://www.tutorialspoint.com/artificial_intelligence/images/research_areas_of_ai.jpg

Kecerdasan buatan di masa kini barulah mencapai tahap *artificial narrow intelligence*, di mana setiap mesin intelijen hanya memiliki satu keahlian khusus di salah satu dari lima topik pada gambar 1.1.

Pada tahap berikutnya, *artificial general intelligence*, setidaknya dibutuhkan tiga dari lima bidang di atas untuk menjadi satu mesin AGI yang utuh. Belum pula suatu sistem untuk mengontrol semuanya, sebuah lapisan kesadaran.

Sebenarnya selain lapisan kesadaran, sebuah mesin AGI juga membutuhkan sesuatu yang disebut kecerdasan. Hanya kita sendiri masih kesulitan untuk mendefinisikan kecerdasan secara global.[2][3] Tetapi, untuk makalah ini saya hanya akan membahas lapisan kesadaran pada sebuah mesin AGI.

Bahasan topik makalah ini adalah *artificial narrow intelligence* dan *artificial general intelligence*. Salah satu masalah dalam implementasi *artificial general intelligence* adalah bagaimana ia harus bisa mengambil keputusan untuk berbagai situasi, dan tidak hanya untuk berbagai masalah, karena situasi dapat membawa masalah baru dengan bobot lebih kritis secara temporer. Dengan itu saya mencoba menggunakan algoritma Greedy untuk memecahkan skala prioritas pemecahan masalah berbasis waktu.

II. DASAR TEORI

A. Pohon

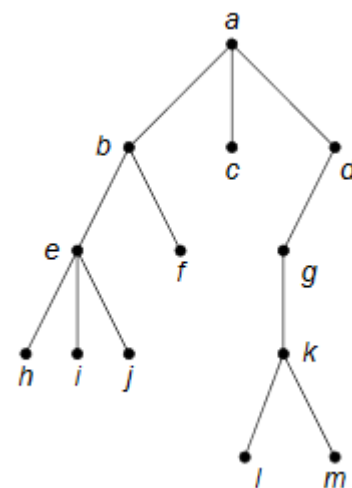
1. Pohon

Pohon adalah graf tak-berarah terhubung yang tidak mengandung sirkuit [4]. Misalkan $G = (V, E)$ adalah graf tak-berarah sederhana dan jumlah simpulnya n . Maka, semua pernyataan di bawah ini adalah ekuivalen:

1. G adalah pohon.
2. Setiap pasang simpul di dalam G terhubung dengan lintasan tunggal.
3. G terhubung dan memiliki $m = n - 1$ buah sisi.
4. G tidak mengandung sirkuit dan memiliki $m = n - 1$ buah sisi.
5. G tidak mengandung sirkuit dan penambahan satu sisi pada graf akan membuat hanya satu sirkuit.
6. G terhubung dan semua sisinya adalah jembatan.

2. Pohon berakar

Pohon yang satu buah simpulnya diperlakukan sebagai akar dan sisi-sisinya diberi arah sehingga menjadi graf berarah dinamakan pohon berakar (rooted tree). Ada beberapa terminologi pohon berakar:



Gambar 2.1 Pohon Berakar

• Anak dan Orang tua

b , c , dan d adalah anak dari simpul a . a adalah orangtua dari b , c , dan d .

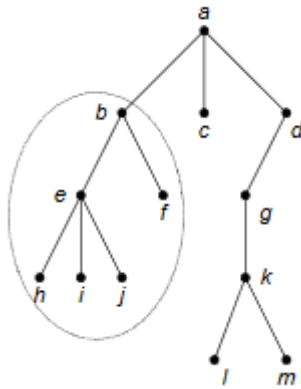
• Lintasan

Berdasarkan gambar 2.1, lintasan dari a ke j adalah a, b, e, j , dan panjang lintasan dari a ke j adalah 3.

• Sibling

Berdasarkan gambar 2.1, f adalah saudara kandung e , tetapi g bukan saudara kandung e , karena orangtua mereka berbeda.

• Upapohon



Gambar 2.2 Pohon berakar dengan upapohon yang dilingkari

- Derajat

Derajat sebuah simpul adalah jumlah upapohon (atau jumlah anak) pada simpul tersebut.

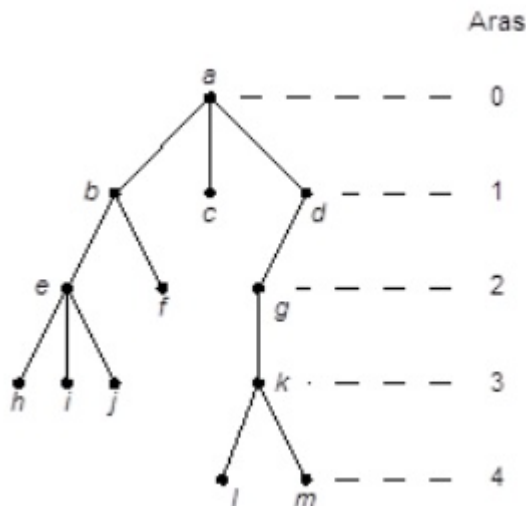
- Daun

Simpul yang berderajat nol (atau tidak mempunyai anak) disebut daun. Berdasarkan gambar 2.1, simpul h, i, j, c, l, dan m adalah daun.

- Simpul dalam

Simpul yang mempunyai anak disebut simpul dalam. Berdasarkan gambar 2.1, simpul b, d, e, g, dan k adalah simpul dalam.

- Aras/level



Gambar 2.3 Pohon berakar dengan aras yang ditunjukkan.

- Tinggi

Aras maksimum dari suatu pohon disebut tinggi atau kedalaman pohon tersebut. Pohon di atas, gambar 2.3, mempunyai tinggi 4.

3. Pohon n-ary

Pohon berakar yang setiap simpul cabangnya mempunyai paling banyak n buah anak disebut pohon n-ary. Pohon n-ary dikatakan teratur atau penuh (full) jika setiap simpul cabangnya mempunyai tepat n anak.

B. Algoritma Greedy

Algoritma greedy merupakan metode yang paling populer untuk memecahkan persoalan optimasi. Persoalan optimasi (optimization problems) adalah persoalan mencari solusi optimum.

Hanya ada dua macam persoalan optimasi:

1. Maksimasi (maximization)
2. Minimasi (minimization)

Contoh persoalan optimasi salah satunya adalah Persoalan Penukaran Uang:

->diberikan uang senilai A. Tukar A dengan koin-koin uang yang ada. Berapa jumlah minimum koin yang diperlukan untuk penukaran tersebut?

Ini adalah contoh persoalan minimasi.

1. Prinsip algoritma Greedy

Greedy berarti rakus, tamak, loba, dll. Prinsip greedy adalah, “take what you can get now!”. Algoritma greedy membentuk solusi langkah per langkah (step by step). Pada setiap langkah, terdapat banyak pilihan yang perlu dievaluasi. Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Pada setiap langkah, kita membuat pilihan optimum lokal (local optimum) dengan harapan bahwa langkah sisanya mengarah ke solusi optimum global (global optimum).

Algoritma greedy adalah algoritma yang memecahkan masalah langkah per langkah; pada setiap langkah:

1. mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan (prinsip “take what you can get now!”)
2. berharap bahwa dengan memilih optimum lokal pada setiap langkah akan berakhir dengan optimum global.

Tinjau masalah penukaran uang:

Strategi greedy:

Pada setiap langkah, pilihlah koin dengan nilai terbesar dari himpunan koin yang tersisa.

Misal: A = 32, koin yang tersedia: 1, 5, 10, dan 25

- Langkah 1: pilih 1 buah koin 25 (Total = 25)
- Langkah 2: pilih 1 buah koin 5 (Total = 25 + 5 = 30)
- Langkah 3: pilih 2 buah koin 1 (Total = 25+5+1+1= 32)

Solusi: Jumlah koin minimum = 4 (solusi optimal!)

2. Elemen-elemen algoritma greedy:

1. Himpunan kandidat, C.

Merupakan himpunan berisi elemen-elemen pembentuk solusi.

2. Himpunan solusi, S.

Merupakan himpunan berisi kandidat-kandidat terpilih sebagai solusi persoalan. Himpunan solusi adalah himpunan bagian dari himpunan kandidat.

3. Fungsi seleksi (selection function)

Merupakan fungsi yang memilih kandidat yang paling memungkinkan mencapai solusi optimal.

4. Fungsi kelayakan (feasible)

Merupakan fungsi yang memeriksa apakah kandidat yang terpilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala yang ada.

5. Fungsi obyektif

Merupakan fungsi yang memaksimumkan atau meminimumkan nilai solusi.

Dengan kata lain, algoritma Greedy melibatkan pencarian sebuah himpunan bagian, S, dari himpunan kandidat, C; yang dalam hal ini, S harus memenuhi beberapa kriteria yang ditentukan, yaitu menyatakan suatu solusi dan S dioptimasi oleh fungsi obyektif.

3. Pseudocode

```
procedure greedy(input C:
himpunan_kandidat; output S :
himpunan_solusi)
{ menentukan solusi optimum dari
persoalan optimasi dengan algoritma greedy

    Masukan: himpunan kandidat C
    Keluaran: himpunan solusi S
}
Deklarasi
x : kandidat;

Algoritma:

S←{}
{ inisialisasi S dengan kosong }
while (belum SOLUSI(S)) and (C ≠ {} )
do
    x←SELEKSI(C);
    { pilih sebuah kandidat dari C}
    C← C - {x}
    { elemen himpunan kandidat
    berkurang satu }

    if LAYAK(S U {x})
        then S← S U {x}
    endif
endwhile

{SOLUSI(S) sudah diperoleh or C = {} }
```

III. ANALISIS DAN PEMBAHASAN

Artificial intelligence pada masa kini hanyalah terbatas pada artificial narrow intelligence (ANI) yang memiliki keahlian masing-masing, dan tidak dapat merespon apapun untuk input di luar bidang keahliannya. Tahap berikutnya untuk artificial intelligence adalah artificial general intelligence (AGI) yang memiliki kecerdasan setara manusia dan memiliki berbagai keahlian. Misal, dapat menerima input suara dengan voice recognition, yang juga merupakan sebuah ANI, lalu, menjawab input tersebut dengan pengetahuannya yang luas dan beragam. Bisa dibilang, AGI adalah gabungan dari berbagai ANI yang menjadi suatu kesatuan utuh.

Dengan AGI sebagai kumpulan dari ANI yang menjadi suatu kesatuan yang utuh, wajarnya sebuah AGI memiliki wujud layaknya seperti manusia, dan memiliki kemampuan sensori serta motorik layaknya manusia. Ini membawa berbagai masalah baru di luar kemampuan berpikir mesin tersebut layaknya manusia, seperti kemampuan survival mesin tersebut di berbagai situasi dan kondisi. Misalnya, sebuah mesin AGI dibawa oleh pemiliknya pergi menaiki pesawat pribadi. Selagi perjalanan, mesin tersebut bermain catur dengan rekan pemiliknya. Tiba-tiba pesawat memasuki keadaan darurat dan penumpang diharapkan menyelamatkan dirinya. Di

sini mesin tersebut haruslah mengerti bahwa keadaan darurat untuk proses penyelamatan dirinya lebih penting dibandingkan permainan catur yang sedang dimainkannya.

Implementasi artificial intelligence sangatlah beragam. Mulai dari penggunaan pengelolaan basis data hingga proses penulisan kode ulang secara real time, keduanya ampuh sebagai implementasi sebuah mesin intelijen. Namun untuk AGI yang merupakan gabungan dari mesin-mesin intelijen sebagai satu entitas yang utuh dibutuhkan suatu lapisan kesadaran untuk menyatukan semua komponen mesin intelijen agar dapat bekerja dengan padu. Karena itu tentunya bentuk implementasi lapisan kesadaran ini akan berbeda dengan bagaimana setiap komponen pada ANI menjalankan tugasnya.

1. Pohon kesadaran

Untuk makalah ini saya mencoba menggunakan sebuah pohon untuk merepresentasikan kesadaran sebuah AGI.

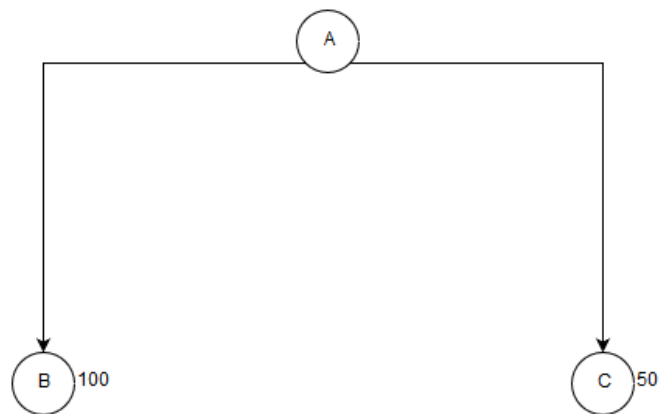
Misalkan sebuah mesin AGI berada pada kondisi yang saya sudah jelaskan sebelumnya. Mesin AGI ini misalnya, dibuat untuk sebuah misi memperbaiki keadaan umat manusia dalam segala bidang. Maka, pada dasarnya, pada pohon kesadarannya mesin ini memiliki sebuah cabang untuk misinya. Karena misi ini terbilang berat, maka bobotnya sangat tinggi. Tentunya misi ini dapat dipecah menjadi bagian-bagian yang lebih kecil. Lalu, dengan suatu algoritma penelusuran pohon ini, akan dicari daun yang paling kecil bobotnya, agar pengerjaan lebih efisien.



Gambar 3.1 Sebuah pohon kesadaran dasar mesin AGI pada contoh

Pada gambar ini A menggambarkan root node, dan B menggambarkan misi utama mesin AGI.

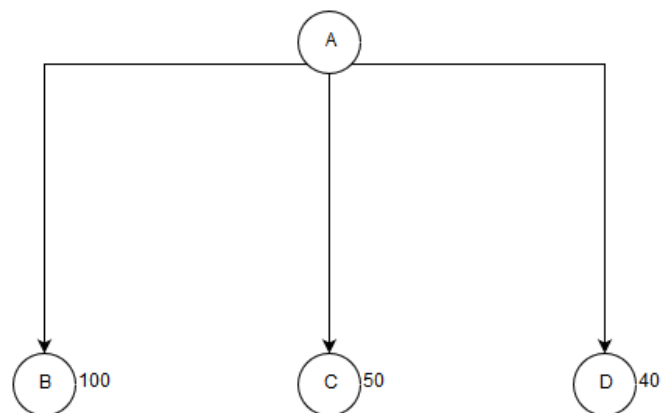
Lalu seperti pada cerita sebelumnya, mesin AGI ini diajak bermain catur bersama rekan pemiliknya. Maka muncul cabang baru pada titik akar paling atas untuk merepresentasikan proses permainan catur. Tentu saja bobot permainan catur lebih ringan dibanding keseluruhan misi mesin AGI ini, tetapi tentunya ada tugas yang lebih kecil bobotnya di bawah cabang B, dan seharusnya mesin melakukan tugasnya yang lebih kecil dibanding permainan catur ini jika mengikuti aturan sebelumnya di mana algoritma akan mencari daun dengan bobot paling kecil untuk dijalankan.



Gambar 3.2 Pohon kesadaran mesin yang sudah diperbaharui. Daun C menggambarkan proses permainan catur.

Tetapi karena faktor waktu (mesin AGI sedang dalam perjalanan untuk tugas yang lain) dan moral (menolak ajakan pertandingan catur dari seorang teman pemilik) maka mesin ini harus bisa menjalankan daun C. Dengan algoritma Greedy untuk pencarian cabang terkecil, tentunya daun C akan terpilih dan mesin AGI dapat melaksanakan permainan catur.

Kondisi berikutnya, semisal pesawat memasuki keadaan darurat di mana penumpangnya harus menyelamatkan diri dengan cara terjun bebas menggunakan parasut. Mesin AGI harus dapat menyesuaikan kesadarannya dan beralih untuk melaksanakan proses penyelamatan, dan jika mungkin, membantu orang-orang di sekitarnya. Muncul cabang baru di bawah root node.



Gambar 3.3 Pohon kesadaran mesin yang sudah diperbaharui.

Proses penyelamatan diri ini, digambarkan oleh cabang D, tentunya lebih mudah dibanding bermain catur, yang sudah lebih mudah dibanding misi utama mesin AGI ini, maka bobotnya paling ringan di antara ketiganya. Pemilihan cabang ini mengalami masalah yang sama dengan sebelumnya, yaitu algoritma penelusuran pohon bisa saja pergi ke cabang B dan mencari daun yang lebih ringan. Tetapi, dengan algoritma Greedy, tentunya cabang D akan dipilih terlebih dahulu dan mesin akan mulai mengalihkan kesadarannya untuk

melaksanakan proses penyelamatan diri dari kondisi darurat tersebut.

IV. KESIMPULAN

Artificial general intelligence membutuhkan sebuah lapisan kesadaran untuk mengelola aktifitas yang dijalankan pada proses komputasinya. Sebagai tahap lanjut dari *artificial narrow intelligence*, sebuah mesin AGI memiliki potensi untuk berada pada berbagai kondisi lingkungan yang bermacam-macam. Untuk mengatasi masalah prioritas pengerjaan ini algoritma biasa yang digunakan pada ANI untuk menelusuri pohon tidak cukup dikarenakan berbagai faktor eksternal yang dihadapi AGI lebih banyak. Masalah ini dapat dipecahkan dengan

V. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih sebesar-besarnya kepada Allah SWT karena berkat rahmatnya penulis bisa menyelesaikan makalah pertama penulis. Tidak lupa penulis ucapkan terima kasih untuk orangtua yang senantiasa mendukung dan mendoakan penulis. Terima kasih juga penulis ucapkan untuk dosen pengajar Mata Kuliah IF2211 Strategi Algoritma Diskrit yaitu Ibu Nur Ulfa Maulidevi dan Dr. Ir. Rinaldi Munir, M.T. atas segala bimbingannya selama perkuliahan di semester IV. Ucapan terima kasih juga penulis ucapkan kepada teman-teman mahasiswa yang membantu menyemangati serta memberi bantuan secara langsung maupun tidak langsung selama pengerjaan makalah.

REFERENSI

- [1] http://www.alanturing.net/turing_archive/pages/reference%20articles/what_is_AI/What%20is%20AI02.html diakses 3 Mei 2016, pukul 20:52.
- [2] <http://waitbutwhy.com/2015/01/artificial-intelligence-revolution-1.html> diakses 4 Mei 2016, pukul 08:22.
- [3] http://www.tutorialspoint.com/artificial_intelligence/artificial_intelligent_systems.htm diakses 5 Mei 2016, pukul 07:43.
- [4] Munir, Rinaldi. Matematika Diskrit. Bandung : Penerbit Informatika, Palasari.

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, 5 Mei 2016



Muhammad Diaztanto Haryaputra