



# LAPORAN TUGAS PROGRAM

## **ARTIFICIAL INTELLIGENCE**

### **Goal State Planning**

Dipersiapkan oleh :

Raden Muhammad Imam (1301154106)

Universitas Telkom

Bandung

2017

## 1. *Goal State Planning*

### A. *Pengertian Goal State Planning*

Planning berbeda dengan Search-Based Problem Solving dalam hal representasi goals, states, dan actions, juga berbeda dalam representasi dan pembangunan urutan-urutan action. Planning berusaha untuk mengatasi kesulitan-kesulitan yang dialami dalam Search-Based Problem Solving. Planning adalah suatu metode penyelesaian masalah dengan cara memecah masalah ke dalam sub-sub masalah yang lebih kecil, menyelesaikan sub-sub masalah satu demi satu, kemudian menggabungkan solusi-solusi dari sub-sub masalah tersebut menjadisebuah solusi lengkap dengan tetap mengingat dan menangani interaksi yang ada antar sub masalah.

Pengujian keberfungsian suatu metode perencanaan dapat dilakukan pada suatu domain yang dinamakan Dunia Balok (Blocks-World). Dunia Balok dinilai cukup layak sebagai lahan pengujian karena tingkat kompleksitas permasalahan yang mungkin timbul di dalamnya. Di dalam Dunia Balok dikenal istilah initial-state dan goal-state yang masing-masing direpresentasikan oleh suatu komposisi dari sejumlah balok. Kemudian, ada satu set operator yang dapat diaplikasikan oleh sebuah tangan robot untuk memanipulasi balok. Permasalahan yang ada di dalam Dunia Balok adalah: Rangkaian operator seperti apa yang dapat mengubah initial-state menjadi goal-state? Rangkaian operator tersebut biasa disebut sebagai Rencana Penyelesaian

Dua metode perencanaan yang cukup populer dan sudah pernah diuji pada Dunia Balok adalah Goal-Stack-Planning (GSP) dan Constraint-Posting (CP). GSP dan CP memiliki kelemahan dan keunggulan masing-masing. Dari segi kemudahan implementasi dan biaya komputasi, GSP lebih unggul dibanding CP. Sedangkan, dari segi efisiensi Rencana Penyelesaian yang dihasilkan, CP pada umumnya lebih unggul dibanding GSP.

Tetapi, dari seluruh kemungkinan permasalahan yang timbul pada Dunia Balok, meskipun GSP dan CP mampu menghasilkan rencana-rencana penyelesaian, namun rencana-rencana penyelesaian yang dihasilkan pada umumnya tidak efisien [RIC91]. Padahal, rencana penyelesaian yang efisien adalah salah satu hal yang penting, terutama pada sistem-sistem raksasa. Pada domain tertentu, pemecahan masalah menggunakan komputer perlu dilakukan dengan cara bekerja pada bagian-bagian kecil dari masalah secara terpisah kemudian menggabungkan solusi-solusi per bagian kecil tersebut menjadi sebuah solusi lengkap dari masalah. Karena, jika hal-hal tersebut tidak dilakukan, jumlah kombinasi state dari komponen masalah menjadi terlalu besar untuk dapat ditangani dipandang dari segi waktu yang tersedia.

#### **Dua syarat untuk melakukan dekomposisi di atas adalah :**

- Menghindari penghitungan ulang dari seluruh state masalah ketika terjadi perubahan dari suatu state ke state lainnya.
- Membagi masalah ke dalam beberapa sub masalah yang relatif lebih mudah untuk diselesaikan.

Penggunaan metode-metode yang terfokus pada cara mendekomposisi masalah ke dalam sub-sub masalah yang sesuai dan cara untuk mengingat dan menangani interaksi antar sub masalah ketika terjadi proses penyelesaian masalah tersebut diistilahkan dengan Perencanaan.

### **Pemecahan masalah dengan menggunakan perencanaan pada umumnya:**

- Modus Goal-Directed, yaitu pencarian solusi dilakukan dari kondisi goal-state sampai ke kondisi initial-state yang dapat dicapai.
- Runut-Balik-Terpandu-Kebergantungan (Dependency-Directed-Backtracking) ketika menemukan jalan buntu.

Untuk berpindah dari satu state ke state lainnya, perencana diperkenankan untuk mengaplikasikan sejumlah operator. Sebuah sistem perencanaan pada umumnya perlu untuk mampu:

- Memilih operator
- Mengaplikasikan operator
- Mendeteksi ketika solusi telah tercapai
- Mendeteksi jalan-jalan buntu
- Mendeteksi ketika solusi yang hampir benar telah dicapai dan melakukan teknik khusus untuk membuat solusi tersebut menjadi benar

### **Pemilihan operator pada umumnya dilakukan dengan:**

- Pertama, mengumpulkan kondisi pembeda goal-state dan current –state (current – state adalah suatu state yang menunjukkan kondisi saat ini).
- Kedua, mengidentifikasi operator yang dapat mengurangi perbedaan tersebut.

Pendeteksian bahwa solusi telah ditemukan dilakukan dengan cara menguji rangkaian operator yang telah dihasilkan. Jika rangkaian operator tersebut dapat mengubah kondisi initial-state menjadi goal-state, maka solusi telah ditemukan. Pendeteksian jalan-jalan buntu dilakukan dengan: jika suatu rangkaian operator menghasilkan state yang diyakini bukan state antara goal-state dan initial-state atau jika kemajuan yang dicapai dianggap terlalu kecil atau menimbulkan permasalahan yang lebih sulit, maka rangkaian operator tersebut dianggap telah menemukan atau akan menuju jalan buntu.

Membuat solusi yang hampir benar menjadi benar dilakukan jika teknik yang dipakai dalam pencarian solusi hanya menghasilkan rangkaian solusi yang hampir benar. Perencana kemudian akan mengaplikasikan suatu metode tertentu untuk membuat solusi yang hampir benar tersebut menjadi benar.

Untuk menghindari pencatatan kondisi keseluruhan di setiap titik persimpangan (node) dalam pencarian solusi, pengaplikasian operator memerlukan tiga daftar predikat untuk mendeskripsikan perubahan kondisi, yaitu:

- PRECONDITION : predikat – predikat yang harus bernilai benar sebelum pengaplikasian operator
- ADD: predikat-predikat yang bernilai benar setelah pengaplikasian suatu operator.
- DELETE : predikat-predikat yang bernilai salah setelah pengaplikasian suatu operator

Ketiga daftar predikat tersebut selanjutnya disebut dengan daftar-PAD.

## B. Studi Kasus *Goal State Planning*

Diberikan suatu data Kurs Penukaran Mata Uang pada Tabel 1, di mana kurs penukaran dibaca dari baris ke kolom. Sebagai contoh, USD 1 = IDR 13.200, SGD 1 = IDR 9.450, GBP 1 = IDR 17.700, EUR 1 = IDR 15.100, IDR 1 = USD 0,0000788, dan seterusnya. Tugas Anda adalah mendesain *Goal Stack Planning* untuk menemukan urutan penukaran mata uang, dimulai dari IDR, kemudian ke sejumlah mata uang lainnya (masing-masing satu-dan-hanya-satu kali penukaran), dan akhirnya kembali ke IDR yang menghasilkan keuntungan paling maksimum.

- Definisikan semua *state* (kondisi) yang ada pada kasus tersebut disertai penjelasan yang lengkap dan detail
- Definisikan himpunan operator yang berisi semua operator yang diperlukan disertai penjelasan yang lengkap dan detail
- Lakukan penelusuran *Goal Stack Planning*, disertai penjelasan yang lengkap dan detail, sampai dihasilkan urutan penukaran mata uang dimulai dari IDR, kemudian ke sejumlah mata uang lainnya (masing-masing satu-dan-hanya-satu kali penukaran), dan akhirnya kembali ke IDR yang menghasilkan keuntungan paling maksimum. Jika Si A memiliki uang tunai sebesar IDR 100.000.000 (seratus juta rupiah), berapa rupiah yang akan didapat Si A jika dia menukarkan semua uangnya dengan mengikuti urutan penukaran yang menghasilkan keuntungan paling maksimum tersebut?

Tabel 1. Kurs Penukaran Mata Uang

Kurs	IDR	USD	SGD	GBP	EUR
IDR	1	0,000078 8	0,000100 5	0,000057 6	0,000057 6
USD	13.200	1	1,368888 9	0,708474 6	0,708474 6
SGD	9.450	0,687272 7	1	0,517881 4	0,517881 4
GBP	17.700	0,738305 1	1,854285 7	1	0,836045 2
EUR	15.100	0,865430 5	1,581904 8	1,581904 8	1

## C. Jawaban Studi Kasus *Goal State Planning*

Goal State = IDR → EUR → GBP → SGD → USD → IDR (IDR 100.000.000 x EUR 0,0000576 x GBP 1,5819048 x SGD 1,8542857 x USD 0,6872727 x IDR 13.200)

$$\begin{aligned}
 &100000000 \times 0,0000576 \times 1,5819048 \\
 &\times 1,8542857 \times 0,6872727 \times 13200 = \\
 &\mathbf{153.278.944,3410042}
 \end{aligned}$$

Goal State : Merupakan goal dengan hasil paling menguntungkan

Contoh : Initial state dengan uang = IDR 100.000.000 melalui perubahan dan goal state menjadi : IDR 153.278.944,3410042

Initial State : Hold\_Money (IDR) & Check\_Box

- a. State = Hold\_Money (x)                      Change (x)  
                    Check\_Box                      Inbox (x)  
                    Disable\_Option (x)

\*Keterangan:

Hold\_Money        : Memegang mata uang tersebut

Check\_Box         : Mengecek kotak dalam keadaan kosong

Change             : Melakukan procedure tukar mata uang

Disable\_Option : Mengset mata uang agar tidak bisa digunakan untuk kedua kali

Inbox                : Memasukkan uang ke dalam kotak

- b. Operator =

Put_Money (x)
$P = \text{Hold\_Money (x)} \wedge \text{Check\_Box}$
$A = \text{Inbox (x)}$
$D = \text{Hold\_Money (x)} \wedge \text{Check\_Box}$

Take_Money (x)
$P = \text{Inbox (x)}$
$A = \text{Hold\_Money (x)} \wedge \text{Check\_Box}$
$D = \text{Inbox (x)}$

Exchange (x , y)
$P = \text{Inbox (x)}$
$A = \text{Change (y)} \wedge \text{Disable\_Option (y)} \wedge \text{Inbox (y)}$
$D = \text{Inbox (x)}$

c. Goal Stack Planning

Current		Queue
Hold_Money (IDR)	Put_Money	(IDR)
Check_Box	Exchange	(IDR , EUR)
Inbox (IDR)	Exchange	(EUR , GBP)
Change (EUR)	Exchange	(GBP , SGD)
Disable_Option (EUR)	Exchange	(SGD , USD)
Inbox (EUR)	Exchange	(USD , IDR)
Change (GBP)	Take_Money	(IDR)
Disable_Option (GBP)		
Inbox (GBP)		
Change (SGD)		
Disable_Option (SGD)		
Inbox (SGD)		
Change (USD)		
Disable_Option (USD)		
Inbox (USD)		
Change (IDR)		
Disable_Option (IDR)		
Inbox (IDR)		
Check_Box		