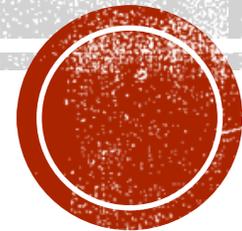


# **PLANNING (TEKNIK DEKOMPOSISI MASALAH)**

Oleh:  
Renna Yanwastika Ariyana, S.T., M.Kom



## Pokok Bahasan dalam Materi Planning:

1. Pengertian Planning
2. Penyelesaian masalah menggunakan Planning
3. Goal-Starck-Planning (GSP)
4. Constraint



# PLANNING

**Plan (Rencana)**

**Planning (Perencanaan)**

Planning → aksi atau proses membuat plans untuk sesuatu



Definisi yang mengacu pada istilah plan dalam tata bahasa (Kamus Oxford):

- ❖ Ide atau metode yang telah dipikirkan secara detail sebelum menyelesaikan suatu pekerjaan.
- ❖ Diagram atau peta detail tentang bagian – bagian penting suatu benda.
- ❖ Cara penyusunan suatu benda.
- ❖ Penyusunan keuangan sehingga seseorang bisa mendapatkan keuntungan.



# PLANNING DALAM KECERDASAN ARTIFISIAL

Planning → suatu penyelesaian masalah dengan:

1. memecah masalah menjadi bagian-bagian yang lebih kecil (sub bab),
2. Menyelesaikan masalah dalam sub bab satu demi satu
3. Menggabungkan solusi dari sub – sub masalah menjadi sebuah solusi lengkap.
4. Mengingat dan menangani interaksi yang terdapat pada setiap bagian sub – sub masalah yang telah di bagi.



**INGAT !!!**

**PLANNING DALAM KECERDASAN  
ARTIFISIAL BERBEDA DENGAN  
SEARCHING**



Teknik planning berbeda dengan searching dalam hal :

- ❖ Representasi goals,
- ❖ State
- ❖ Action,
- ❖ Representasi dan pembangunan urutan – urutan dalam action.

Planning berusaha mengatasi kesulitan – kesulitan dalam waktu proses dan kebutuhan memori.



# PENGUJIAN PLANNING

Proses pengujian planning yaitu:

- ❖ Menganalisa masalah yang dihadapi dan mengungkapkan masalah tersebut dala satu sistem simbol.
- ❖ Sistem dapat merupakan diagram, skema, graf atau simbol" yang lain.
- ❖ Sistem simbol ini harus diterjemahkan dala bahasa pemgrograman AI.
- ❖ Terdapat Initial state dan Goal State



Secara umum pendefinisian masalah sebagai suatu ruang keadaan meliputi 3 hal :

- ❖ Posisi Awal (initial State)
- ❖ Aturan (Rule )
- ❖ Tujuan (Goal)



# CONTOH:

Misal: permainan catur , maka State Space Search (SSS) nya adalah :

- ❖ Menspesifikasikan posisi awal dari papan catur
- ❖ Peraturan (rules) yang mendefinisikan langkah-langkah yang legal
- ❖ Posisi papan yang merepresentasikan pemenang dari satu sisi atau sisi lainnya.
- ❖ Tujuan (Goal) dari permainan adalah : memenangkan permainan.



# SIFAT PEMECAHAN MASALAH PLANNING

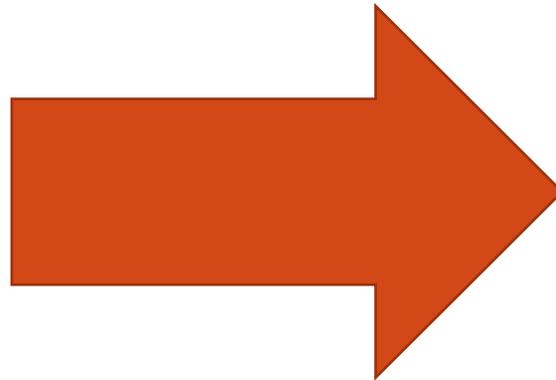
❖ Goal – Directed → pencarian solusi dilakukan dari kondisi goal – state sampai ke kondisi initial - state yang dapat dicapai

❖ Bersifat:

Dependency

Directed

Backtracking



ketika  
Menemukan  
jalan buntu.



# KEMAMPUAN PLANNING

Untuk berpindah dari suatu state ke state yang lain , sistem planning diperkenankan untuk mengamplikasikan sejumlah operator.

Suatu sistem Planning memiliki kemampuan :

- ❖ 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 2 cara

- ❖ Mengumpulkan kondisi pembeda goal-state dan current-state(kondisi saat ini)
- ❖ Mengidentifikasi operator yang dapat mengurangi perbedaan tersebut.



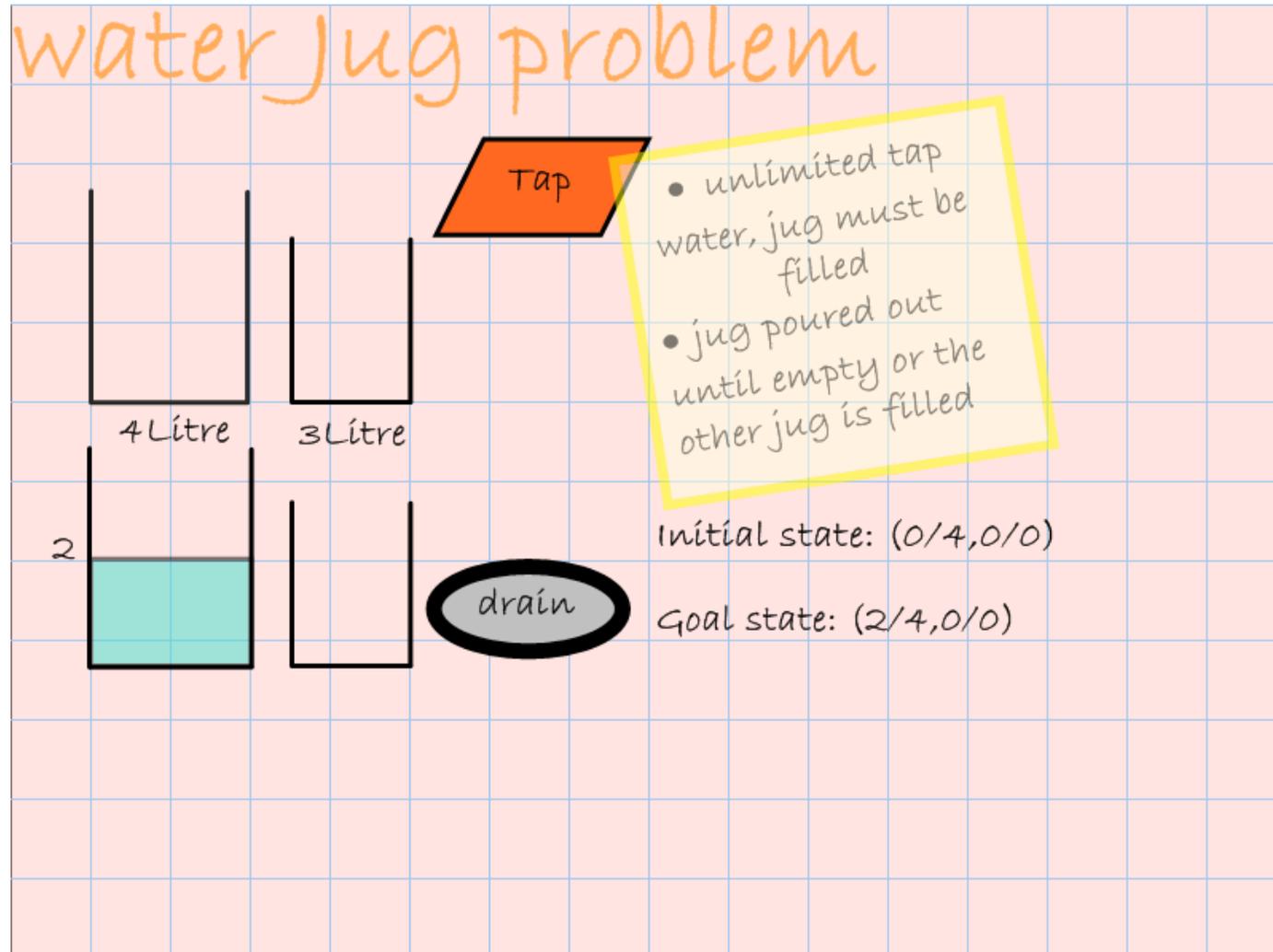
# DETEKSI

Proses pendeteksian jika:

- ❖ Solusi telah ditemukan → dilakukan dengan cara menguji rangkaian operator yang telah dihasilkan. Jika rangkaian operator dapat mengubah kondisi initial state menjadi goal state, berarti solusi telah ditemukan.
- ❖ Proses menemukan jalan buntu → dilakukan dengan cara, setiap rangkaian operator menghasilkan state yang diyakini bukan state antara goal state dan initial state atau jika kemajuan yang 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.



# CONTOH KASUS



# MENDEFINISIKAN MASALAH SEBAGAI “STATE SPACE SEARCH” (SSS)

Kasus : A permasalahan menuangkan air

- ❖ Initial state : Diketahui dua buah ember masing-masing berkapasitas 3 gallon dan 4 gallon, dan sebuah pompa air.
- ❖ Goal state : Isi ember yang berkapasitas 4 gallon dengan 2 gallon air!
- ❖ Solusi :

Buat asumsi dengan :

X : ember berkapasitas 4 gallon

Y : ember berkapasitas 3 gallon



- ❖ Ruang masalah untuk masalah di atas dapat digambarkan sebagai himpunan pasangan bilangan bulat  $(x,y)$  yang terurut, sedemikian hingga :
  - $x = 0, 1, 2, 3, \text{ atau } 4$
  - $y = 0, 1, 2, \text{ atau } 3;$
- ❖  $x$  menyatakan jumlah air dalam gelas ukuran 4 galon, dan  $y$  menyatakan jumlah air dalam gelas ukuran 3 galon. Dengan keadaan mula-mula adalah  $(0,0)$ .
- ❖ *State* tujuan adalah  $(2,n)$  untuk setiap nilai  $n$ .



# PRODUCTION RULES:

Sistem Produksi terdiri dari:

- ❖ Sekumpulan Aturan (a set of rules)
- ❖ Knowledge Base /Data Base
- ❖ Sebuah strategi pengontrol (Control Strategy)
- ❖ Urutan yang dipakai (a rule applier)



1.	$(x,y)$ If $x < 4$	$\rightarrow$	$(4,y)$	Isi penuh gelas 4 galon
2.	$(x,y)$ If $y < 3$	$\rightarrow$	$(x,3)$	Isi penuh gelas 3 galon
3.	$(x,y)$ If $x > 0$	$\rightarrow$	$(x-d,y)$	Buang sebagian air dari gelas 4 galon
4.	$(x,y)$ If $y > 0$	$\rightarrow$	$(x,y-d)$	Buang sebagian air dari galon ukuran 3 galon
5.	$(x,y)$ If $x > 0$	$\rightarrow$	$(0,y)$	Kosongkan gelas 4 galon



6.	$(x,y)$ If $y > 0$	$\rightarrow$	$(x,0)$	Kosongkan gelas 3 galon
7.	$(x,y)$ If $x+y \geq 4$ and $y > 0$	$\rightarrow$	$(4,y-(4-x))$	Tuangkan air dari gelas 3 galon ke gelas 4 galon sampai gelas 4 galon penuh
8.	$(x,y)$ If $x+y \geq 3$ and $x > 0$	$\rightarrow$	$(x-(3-y),3)$	Tuangkan air dari gelas 4 galon ke gelas 3 galon sampai gelas 3 galon penuh
9.	$(x,y)$ If $x+y \leq 4$ and $y > 0$	$\rightarrow$	$(x+y,0)$	Tuangkan seluruh air dari gelas 3 galon ke gelas 4 galon



10.	(x,y) If $x+y \leq 3$ and $x > 0$	→	(0,x+y)	Tuangkan seluruh air dari gelas 4 galon ke gelas 3 galon
11.	(0,2)	→	(2,0)	Tuangkan 2 galon air dari gelas 3 galon ke gelas 4 galon
12.	(2,y)	→	(0,y)	Buang 2 galon dalam gelas 4 galon sampai habis.



# Solusi

<b>Jumlah galon dalam gelas 4 galon</b>	<b>Jumlah galon dalam gelas 3 galon</b>	<b>Aturan yang dilakukan</b>
0	0	-
0	3	2
3	0	9
3	3	2
4	2	7
0	2	5 atau 12
2	0	9 atau 11



# 3 DAFTAR DESKRIPSI PERUBAHAN KONDISI

Untuk menghindari pencatatan kondisi keseluruhan di setiap titik persimpangan dalam pencarian solusi, pengaplikasian operator memerlukan 3 daftar predikat untuk mendiskripsikan 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.



# DUNIA BALOK (BLOCKS-WORLD)

Dunia Balok adalah suatu masalah yang dapat didekomposisi dengan karakteristik sebagai berikut :

- ❖ Memiliki sebuah permukaan datar tempat menyimpan balok, umumnya disebut meja.
- ❖ Memiliki sejumlah balok kotak yang berukuran sama.
- ❖ Memiliki sebuah lengan robot yang dapat memanipulasi balok.



Untuk memudahkan pendefinisian kondisi balok pada suatu state dalam dunia balok, kita bisa menggunakan predikat – predikat berikut ini:

- ❖  $ON(A,B)$  : Balok A menempel di permukaan meja.
- ❖  $ONTABLE(A)$  : Balok A berada di permukaan meja.
- ❖  $CLEAR(A)$  : Tidak ada balok yang sedang menempel diatas balok A.

Sedangkan, untuk lengan robot digunakan predikat – predikat sebagai berikut:

- ❖  $HOLDING(A)$  : Lengan Robot memegang balok A.
- ❖  $ARMEMPTY$  : Lengan robot tidak sedang memegang balok.



# FIRST ORDER LOGIC

Untuk merepresentasikan kondisi suatu keadaan didalam Dunia Balok digunakan notasi First Order Logic

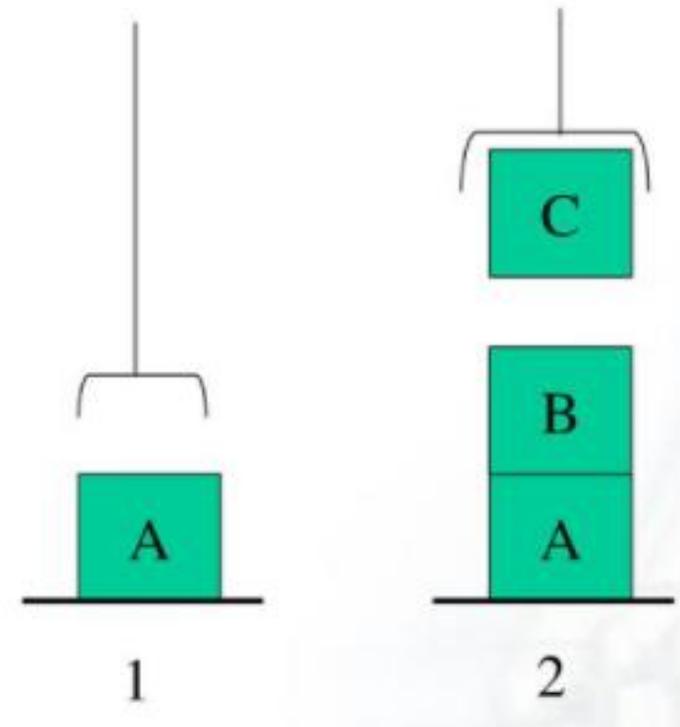
## CONTOH:

**Gbr 1. Notasi FOL :**

**ONTABLE(A)  $\wedge$  CLEAR(A)  $\wedge$   
ARMEMPTY**

**Gbr 2. Notasi FOL :**

**ONTABLE(A)  $\wedge$  ON(B,A)  $\wedge$   
CLEAR(B)  $\wedge$  HOLDING(C)  $\wedge$   
CLEAR(C)**



Untuk mengubah keadaan menjadi keadaan yang lain, diperlukan operator, Dalam dunia balok , dapat didefinisikan 4 operator sbb:

*Daftar operator untuk lengan Robot*

Operator	Hal yang dilakukan
STACK(A,B)	Meletakkan balok A diatas balok B
UNSTACK(A,B)	Mengangkat balok A yang menempel diatas balok B
PICKUP(A)	Mengangkat balok A dari permukaan meja
PUTDOWN(A)	Meletakkan balok A di permukaan meja



Agar suatu operator bisa diaplikasikan tentu saja ada persyaratan yang harus dipenuhi, seperti:

- ❖ STACK(A,B) bisa diaplikasikan jika kedua kondisi :CLEAR(B) dan HOLDING(A) adalah benar.
- ❖ Persyaratan yang harus dipenuhi agar suatu operator dapat diaplikasikan disebut dengan Precondition.
- ❖ Setelah suatu operator diaplikasikan , maka keadaan akan berubah: Keadaan yang benar setelah operator diaplikasikan adalah **Add**, dan keadaan yang salah disebut **Delete**.
- ❖ Jadi setelah STACK(A,B) diaplikasikan maka keadaan yang benar adalah :CLEAR(B) DAN HOLDING(A).



Daftar yang lengkap  
tentang Precondition,  
Add dan Delete untuk  
keempat operator  
lengan robot  
ditunjukkan oleh  
notasi

### **STACK(A,B)**

- P:HOLDING(A)  $\wedge$  CLEAR(B)
- A:ON(A,B)  $\wedge$  ARMEMPTY
- D:HOLDING(A)  $\wedge$  CLEAR(B)

### **UNSTACK(A,B)**

- P:ON(A,B)  $\wedge$  CLEAR(A)  $\wedge$  ARMEMPTY
- A:HOLDING(A)  $\wedge$  CLEAR(B)
- ON(A,B)  $\wedge$  ARMEMPTY

### **PICKUP(A)**

- P:ONTABLE(A)  $\wedge$  CLEAR(A)  $\wedge$  ARMEMPTY
- A:HOLDING(A)
- D:ONTABLE(A)  $\wedge$  ARMEMPTY

### **PUTDOWN(A)**

- P:HOLDING(A)
- A:ONTABLE(A)  $\wedge$  ARMEMPTY
- D:HOLDING(A)



# PERMASALAHAN

Permasalahan yang hendak dicari penyelesaiannya adalah permasalahan yang memenuhi spesifikasi sbb:

- ❖ Jumlah balok sama.
- ❖ Balok yang ada pada initial-state juga berada pada goal state.
- ❖ Kondisi terdefinisi dengan benar.
- ❖ Lengan robot sedang tidak memegang balok.

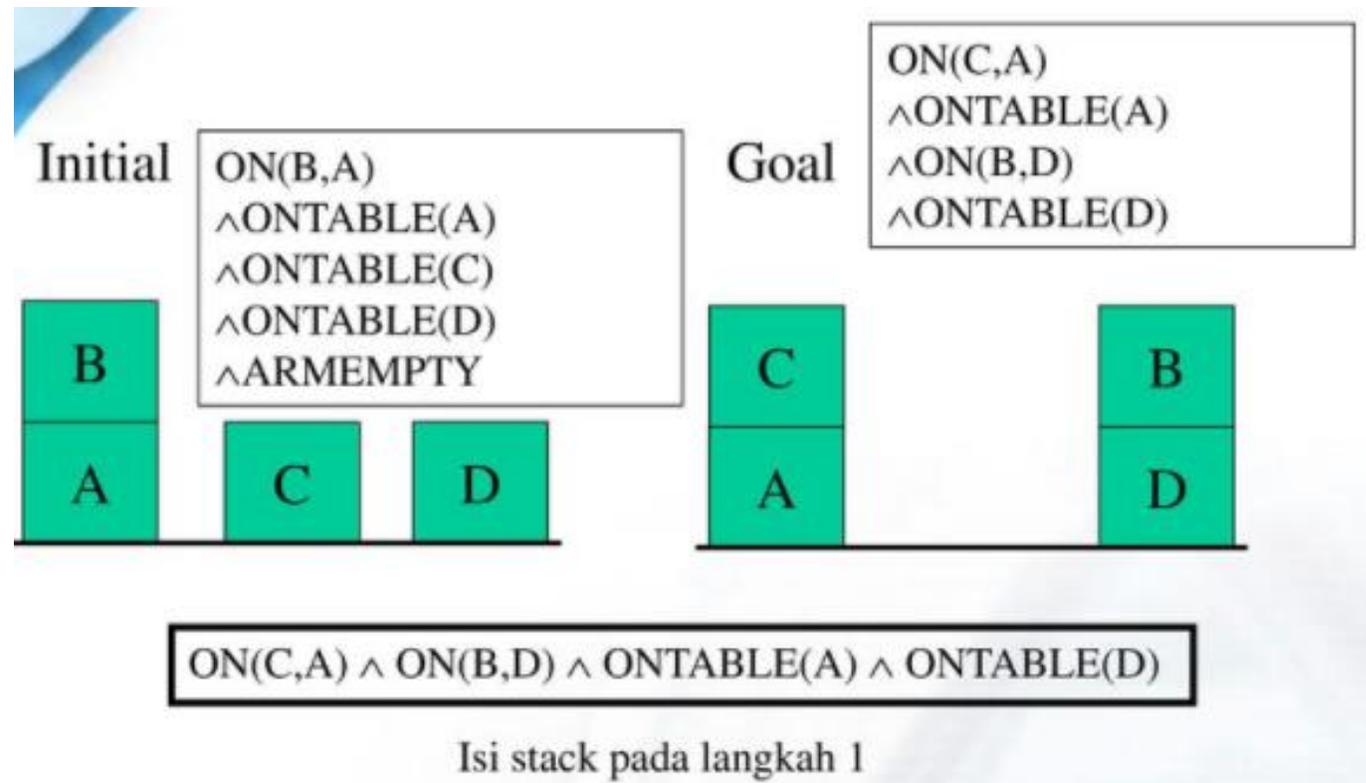


# GOAL-STACK-PLANNING(GSP)

- ❖ Untuk menyelesaikan suatu masalah . GSP menggunakan **STACK (Tumpukan)** untuk menampung keadaan – keadaan tujuan (**Goal State**) dan keadaan – keadaan lain yang mungkin terjadi selama proses pencarian solusi).



Langkah pertama yang dilakukan GSP untuk menyelesaikan suatu masalah adalah memasukkan kondisi – kondisi yang merupakan goal-state ke dalam stack.kondisi – kondisi tersebut akan tersimpan dalam stack.



Langkah 2 → Mengacu pada current state, kondisi – kondisi goal state yang belum tercapai dimasukkan dalam stack, masing – masing menempati sebuah slot. Tidak ada aturan khusus untuk proses urutan pemasukan kondisi – kondisi yang belum tercapai dalam stack.

ON(C,A)  
ON(B,D)  
ON(C,A)  $\wedge$  ON(B,D)  $\wedge$  OTAD

[1]

ON(B,D)  
ON(C,A)  
ON(C,A)  $\wedge$  ON(B,D)  $\wedge$  OTAD

[2]

Isi stack pada langkah 2



CLEAR(A)  
HOLDING(C)  
CLEAR(A)  $\wedge$  HOLDING(C)  
**STACK(C,A)**  
ON(B,D)  
ON(C,A)  $\wedge$  ON(B,D)  $\wedge$  OTAD

Langkah 3

ON(B,A)  
CLEAR(B)  
ARMEMPTY  
ON(B,A)  $\wedge$  CLEAR(B)  $\wedge$  ARMEMPTY  
**UNSTACK(B,A)**  
HOLDING(C)  
CLEAR(A)  $\wedge$  HOLDING(C)  
**STACK(C,A)**  
ON(B,D)  
ON(C,A)  $\wedge$  ON(B,D)  $\wedge$  OTAD

Langkah 4

**ONTABLE(A)  $\wedge$  ONTABLE(C)  $\wedge$   
ONTABLE(D)  $\wedge$  HOLDING(B)**

HOLDING(C)  
CLEAR(A)  $\wedge$  HOLDING(C)  
**STACK(C,A)**  
ON(B,D)  
ON(C,A)  $\wedge$  ON(B,D)  $\wedge$  OTAD

Langkah 5

ONTABLE(C)  
CLEAR(C)  
ARMEMPTY  
ONTABLE(C)  $\wedge$  CLEAR(C)  $\wedge$  ARMEMPTY  
**PICKUP(C)**  
CLEAR(A)  $\wedge$  HOLDING(C)  
**STACK(C,A)**  
ON(B,D)  
ON(C,A)  $\wedge$  ON(B,D)  $\wedge$  OTAD

Langkah 6

CLEAR(D)  
HOLDING(B)  
CLEAR(D)  $\wedge$  HOLDING(B)  
**STACK(B,D)**  
ONTABLE(C)  $\wedge$  CLEAR(C)  $\wedge$  ARMEMPTY  
**PICKUP(C)**  
CLEAR(A)  $\wedge$  HOLDING(C)  
**STACK(C,A)**  
ON(B,D)  
ON(C,A)  $\wedge$  ON(B,D)  $\wedge$  OTAD

Langkah 7



Rencana Penyelesaian Masalah pada proses tersebut:

- ❖ UNSTACK(B,A)
- ❖ STACK(B,D)
- ❖ PICKUP(C)
- ❖ STACK(C,A)



# KARAKTERISTIK MASALAH DALAM AI

- ❖ Apakah masalahnya dapat didekomposisi menjadi himpunan sub masalah yang (hampir) independen lebih kecil atau lebih mudah ?
- ❖ Dapatkah langkah penyelesaian diacuhkan paling tidak dibatalkan ketika dapat dibuktikan hal tersebut tidak bijaksana ?
- ❖ Apakah universe masalahnya dapat diprediksi ?
- ❖ Apakah solusi yang baik dari masalah tertentu jelas tanpa membandingkan dengan seluruh solusi lain yang mungkin ?
- ❖ Apakah solusi yang diinginkan sebuah keadaan dari dunia atau sebuah jalur dari keadaan ?
- ❖ Apa peran dari pengetahuan ?
- ❖ Apakah pekerjaan memerlukan interaksi dengan manusia ?



**TERIMA KASIH**

