AI | PLANNING

IF-40-01 | Kelompok 7

M. Farras Hafis 1301164348

Rezza Nafi I. 1301160425

Nanda Tri H. 1301160443

Bayu Arifat F. 1301164176

Pengertian (Bahasa)

planning NOUN

The process of making plans for something.

Oxford Dictionaries

Planning pada AI

1. Metode penyelesaian masalah dengan cara memecah masalah ke dalam **sub-sub masalah** yang lebih kecil
2. Menyelesaikan sub-sub masalah satu demi satu
3. Menggabungkan solusi-solusi dari sub-sub masalah tersebut menjadi sebuah solusi lengkap dengan tetap mengingat dan menangani interaksi yang ada antar sub masalah.

Perbedaan Planning dan Searching

- Teknik planning berbeda dengan searching dalam hal representasi goals, state dan action
- Berbeda dalam hal representasi pembangunan urutan-urutan dalam action.
- Planning berusaha mengatasi kesulitan–kesulitan dalam waktu proses dan kebutuhan memori.

Sifat Pemecahan Masalah Planning

- **Goal – Directed**, yaitu pencarian solusi dilakukan dari kondisi goal – state sampai ke kondisi initial - state yang dapat dicapai
- **Dependency- Directed-Backtracking** ketika menemukan jalan buntu.

Kemampuan Planning

Untuk berpindah dari suatu state ke state yang lain, sistem planning diperkenankan untuk mengaplikasikan sejumlah **operator**. Suatu sistem planning pada umumnya perlu memiliki kemampuan :

a. Memilih operator
b. Mengaplikasikan operator
c. Mendeteksi ketika solusi telah tercapai
d. Mendeteksi jalan – jalan buntu
e. Mendeteksi ketika solusi yang hampir benar telah dicapai dan melakukan teknik khusus untuk membuat solusi tersebut menjadi benar

Predikat Operator (PAD)

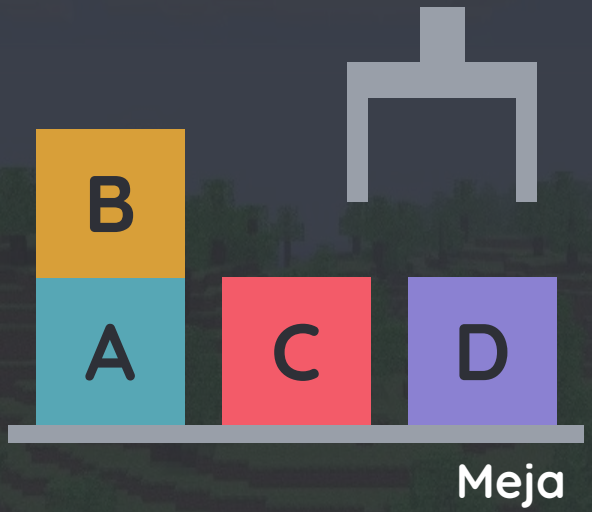
1. **Precondition**

   Predikat yang bernilai benar sebelum pengaplikasian operator
2. **Add**

   Predikat yang bernilai benar setelah pengaplikasian operator
3. **Delete**

   Predikat yang bernilai salah setelah pengaplikasian operator

Dunia Balok (Blocks-World)



1. Terdapat meja (permukaan datar) untuk menaruh balok.
2. Terdapat beberapa balok berukuran sama.
3. Terdapat lengan robot untuk memanipulasi balok.

Predikat Dunia Balok

ON (*x*, *y*)

Balok *x* menempel di atas balok *y*

ONTABLE (*x*)

Balok *x* berada di permukaan meja

CLEAR (*x*)

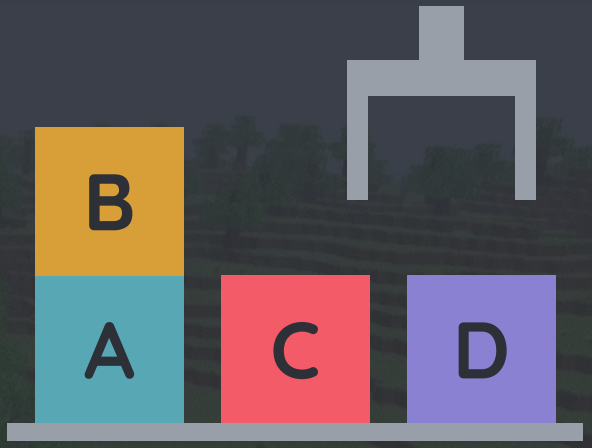
Tidak ada balok di atas balok *x*

ARMEMPTY

Lengan robot tidak memegang balok

HOLDING (*x*)

Lengan robot memegang balok *x*



ON (B, A)
ONTABLE (A)
ONTABLE (C)
ONTABLE (D)
CLEAR (B)
ARMEMPTY

Daftar operator

STACK (A,B)

Meletakkan balok A di atas balok B

UNSTACK (A,B)

Mengangkat balok A yang menempel di atas balok B

PICKUP (A)

Mengangkat balok A dari permukaan meja

PUTDOWN (A)

Meletakkan balok A di permukaan meja

Daftar-PAD untuk operator (1)

STACK (A,B)

P: HOLDING(A) ∧ CLEAR(B)

A: ON(A,B) ∧ ARMEMPTY

D: HOLDING(A) ∧ CLEAR(B)

UNSTACK (A,B)

P: ON(A,B) ∧ CLEAR(A) ∧ ARMEMPTY

A: HOLDING(A) ∧ CLEAR(B)

D: ON(A,B) ∧ ARMEMPTY

Daftar-PAD untuk operator (2)

PICKUP (A)

P: ONTABLE(A) ∧ CLEAR(A) ∧ ARMEMPTY

A: HOLDING(A)

D: ONTABLE(A) ∧ ARMEMPTY

PUTDOWN (A)

P: HOLDING(A)

A: ONTABLE(A) ∧ ARMEMPTY

D: HOLDING(A)

- Goal-Stack Planning (GSP)

Goal-Stack Planning

Menggunakan sebuah **stack** yang menampung:

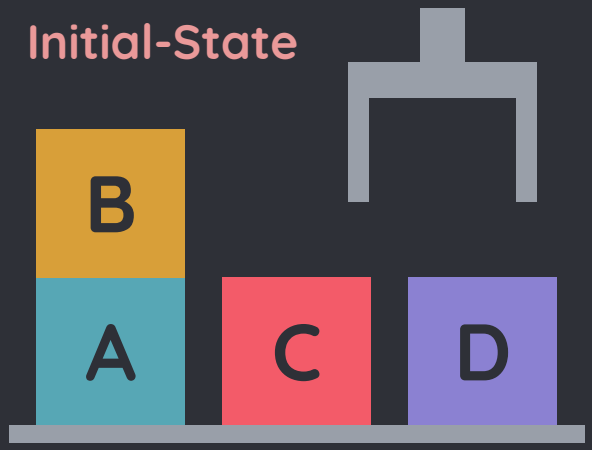
1. Goal State
2. State lain yang mungkin terjadi
3. Operator

Bergantung pada basis data berupa:

1. current-state
2. daftar PAD

Langkah ke-1

Initial-State



Goal-State

Isi **current state** dengan **initial state**:

current state:

ON(B,A) ∧
ONTABLE(A) ∧
ONTABLE(C) ∧
ONTABLE(D) ∧
ARMEMPTY

Masukkan **goal-state** ke dalam **stack:**

STACK
ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D)

Langkah ke-2

Tiap kondisi **goal-state** yang belum tercapai pada **current-state** dimasukkan ke dalam **stack** (urutan tidak ditentukan)

current state

ON(B,A) ∧ONTABLE(A) ∧ONTABLE(C) ∧ONTABLE(D) ∧ARMEMPTY

goal-state

ON(C,A) ∧ **ON(B,D)** ∧ ONTABLE(A) ∧ ONTABLE(D)

STACK
ON(C,A)
ON(B,D)
ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D)

Langkah ke-3 (1)

TOP pada **stack** akan diperiksa. Kondisi yang mungkin terjadi:

Kondisi 1 | **TOP** berisi kondisi yang memenuhi **current state**, tetapi tidak **TOP** tidak berada di dasar **stack** dan **TOP** tidak terletak di atas operator:

1. **TOP** **di-*pop*** dari **stack**.
2. Pemeriksaan dilanjutkan pada **TOP** yang baru.

Kondisi 2 | **TOP** berisi kondisi yang belum memenuhi **current state**:

1. **TOP** **di-*pop*** dari **stack**.
2. Operator yang sesuai untuk memenuhi kondisi pada **TOP** akan **di-*push*** ke **stack**.
3. Tiap predikat **precondition** dari operator tersebut **di-*push*** secara terurut.

Langkah ke-3 (2)

Kondisi 3 | TOP berisi kondisi atau rangkaian kondisi dan berada di atas operator:

1. **TOP di-*pop*** dari **stack**.
2. Operator di bawahnya ***di-pop*** dan dimasukkan ke **rencana penyelesaian**.
3. **Current state di-*update*** dengan mengaplikasikan operator tersebut.

Kondisi 4 | TOP berada di dasar **stack** (elemen **stack** tinggal satu):

1. Uji kesamaan antara **current-state** dan **goal-state**.
2. Jika sama, **TOP di-*pop*** dari **stack** ⇒ **[SELESAI]**
3. Jika beda, langkah ke-2 diulangi.

Jika setelah tindakan berdasarkan tiap kondisi dilakukan dan yang terjadi bukan **kondisi 4**, maka langkah ke-3 diulangi.

Langkah ke-3 (Iterasi ke-1)

STACK
ON(C,A)
ON(B,D)
ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D)

current state

ON(B,A) ∧ ONTABLE(A) ∧
ONTABLE(C) ∧ ONTABLE(D) ∧
ARMEMPTY

STACK
CLEAR(A)
HOLDING(C)
CLEAR(A) ∧ HOLDING(C)
STACK(C,A)
ON(B,D)
ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D)

(Kondisi 2)

Langkah ke-3 (Iterasi ke-2)

STACK
CLEAR(A)
HOLDING(C)
CLEAR(A) ∧ HOLDING(C)
STACK(C,A)
ON(B,D)
ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D)

current state

ON(B,A) ∧ ONTABLE(A) ∧ ONTABLE(C) ∧ ONTABLE(D) ∧ ARMEMPTY

STACK
ON(B,A)
CLEAR (B)
ARMEMPTY
ON(B,A) ∧ CLEAR (B) ∧ ARMEMPTY
UNSTACK(B,A)
HOLDING(C)
CLEAR(A) ∧ HOLDING(C)
STACK(C,A)
ON(B,D)
ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D)

(Kondisi 2)

Langkah ke-3 (Iterasi ke-3-5)

STACK
ON(B,A)
CLEAR (B)
ARMEMPTY
ON(B,A) ∧ CLEAR (B) ∧ ARMEMPTY
UNSTACK(B,A)
HOLDING(C)
CLEAR(A) ∧ HOLDING(C)
STACK(C,A)
ON(B,D)
ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D)

STACK
ON(B,A) ∧ CLEAR (B) ∧ ARMEMPTY
UNSTACK(B,A)
HOLDING(C)
CLEAR(A) ∧ HOLDING(C)
STACK(C,A)
ON(B,D)
ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D)

current state

ON(B,A) ∧ ONTABLE(A) ∧ ONTABLE(C) ∧ ONTABLE(D) ∧ ARMEMPTY

(Kondisi 1)

Langkah ke-3 (Iterasi ke-6)

STACK
ON(B,A) ∧ CLEAR (B) ∧ ARMEMPTY
UNSTACK(B,A)
HOLDING(C)
CLEAR(A) ∧ HOLDING(C)
STACK(C,A)
ON(B,D)
ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D)

current state

ON(B,A) ∧ ONTABLE(A) ∧ ONTABLE(C) ∧ ONTABLE(D) ∧ ARMEMPTY

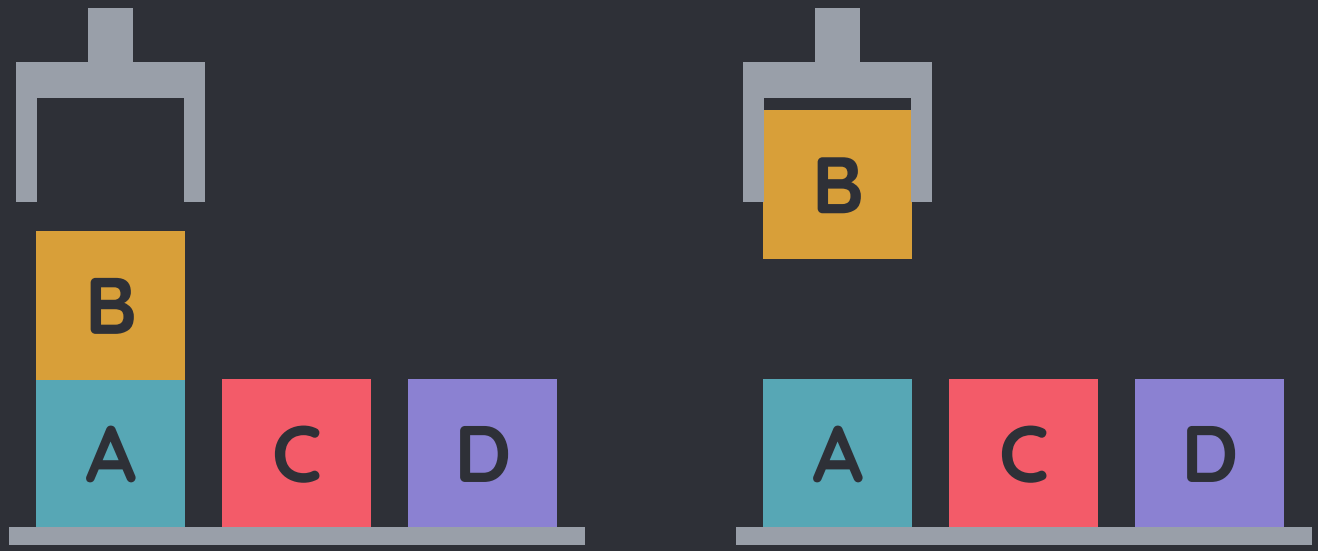
STACK
HOLDING(C)
CLEAR(A) ∧ HOLDING(C)
STACK(C,A)
ON(B,D)
ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D)

new current state setelah operasi UNSTACK(B,A):

ONTABLE(A) ∧ ONTABLE(C) ∧ ONTABLE(D) ∧ HOLDING(B)

(Kondisi 3)

OPERASI UNSTACK(B,A)



Langkah ke-3 (Iterasi ke-7)

STACK
HOLDING(C)
CLEAR(A) ∧ HOLDING(C)
STACK(C,A)
ON(B,D)
ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D)

STACK
ONTABLE(C)
CLEAR(C)
ARMEMPTY
ONTABLE(C) ∧ CLEAR(C) ∧ ARMEMPTY
PICKUP(C)
CLEAR(A) ∧ HOLDING(C)
STACK(C,A)
ON(B,D)
ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D)

current state

ONTABLE(A) ∧ ONTABLE(C) ∧ ONTABLE(D) ∧ HOLDING(B)

(Kondisi 2)

Langkah ke-3 (Iterasi ke-8-9)

STACK
ONTABLE(C)
CLEAR(C)
ARMEMPTY
ONTABLE(C) ∧ CLEAR(C) ∧ ARMEMPTY
CLEAR(A) ∧ HOLDING(C)
STACK(C,A)
ON(B,D)
ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D)

current state

ONTABLE(A) ∧ ONTABLE(C) ∧ ONTABLE(D) ∧ HOLDING(B)

STACK
ARMEMPTY
ONTABLE(C) ∧ CLEAR(C) ∧ ARMEMPTY
CLEAR(A) ∧ HOLDING(C)
STACK(C,A)
ON(B,D)
ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D)

(Kondisi 1)

Langkah ke-3 (Iterasi ke-10)

STACK
ARMEMPTY
ONTABLE(C) ∧ CLEAR(C) ∧ ARMEMPTY
PICKUP(C)
CLEAR(A) ∧ HOLDING(C)
STACK(C,A)
ON(B,D)
ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D)

current state

ONTABLE(A) ∧ ONTABLE(C) ∧ ONTABLE(D) ∧ HOLDING(B)

STACK
CLEAR(D)
HOLDING(B)
CLEAR(D) ∧ HOLDING(B)
STACK(B,D)
ONTABLE(C) ∧ CLEAR(C) ∧ ARMEMPTY
PICKUP(C)
CLEAR(A) ∧ HOLDING(C)
STACK(C,A)
ON(B,D)
ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D)

(Kondisi 2)

Langkah ke-3 (Iterasi ke-11-13)

STACK
CLEAR(D)
HOLDING(B)
CLEAR(D) ∧ HOLDING(B)
STACK(B,D)
ONTABLE(C) ∧ CLEAR(C) ∧ ARMEMPTY
PICKUP(C)
CLEAR(A) ∧ HOLDING(C)
STACK(C,A)
ON(B,D)
ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D)

current state

ONTABLE(A) ∧ ONTABLE(C) ∧ ONTABLE(D) ∧ HOLDING(B)

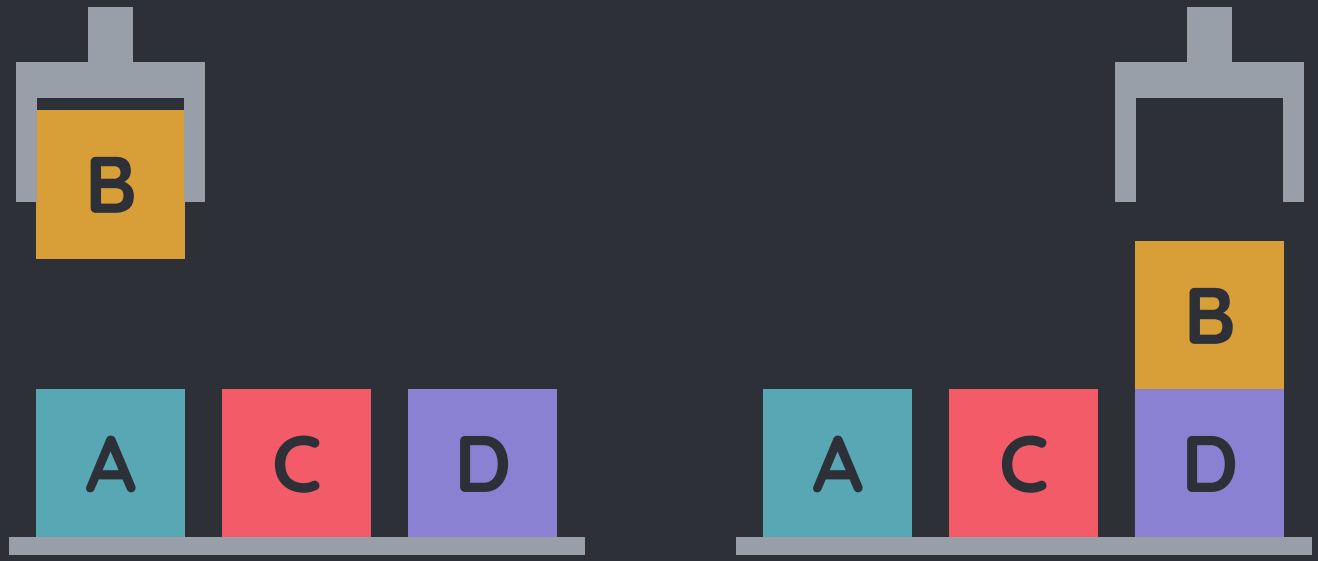
STACK
ONTABLE(C) ∧ CLEAR(C) ∧ ARMEMPTY
PICKUP(C)
CLEAR(A) ∧ HOLDING(C)
STACK(C,A)
ON(B,D)
ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D)

new current state setelah operasi STACK(B,D):

ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(C) ∧ ONTABLE(D) ∧ ARMEMPTY

(Kondisi 3)

OPERASI STACK(B,D)



Langkah ke-3 (Iterasi ke-14)

STACK
ONTABLE(C) ∧ CLEAR(C) ∧ ARMEMPTY
PICKUP(C)
CLEAR(A) ∧ HOLDING(C)
STACK(C,A)
ON(B,D)
ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D)

current state

ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(C) ∧ ONTABLE(D) ∧ ARMEMPTY

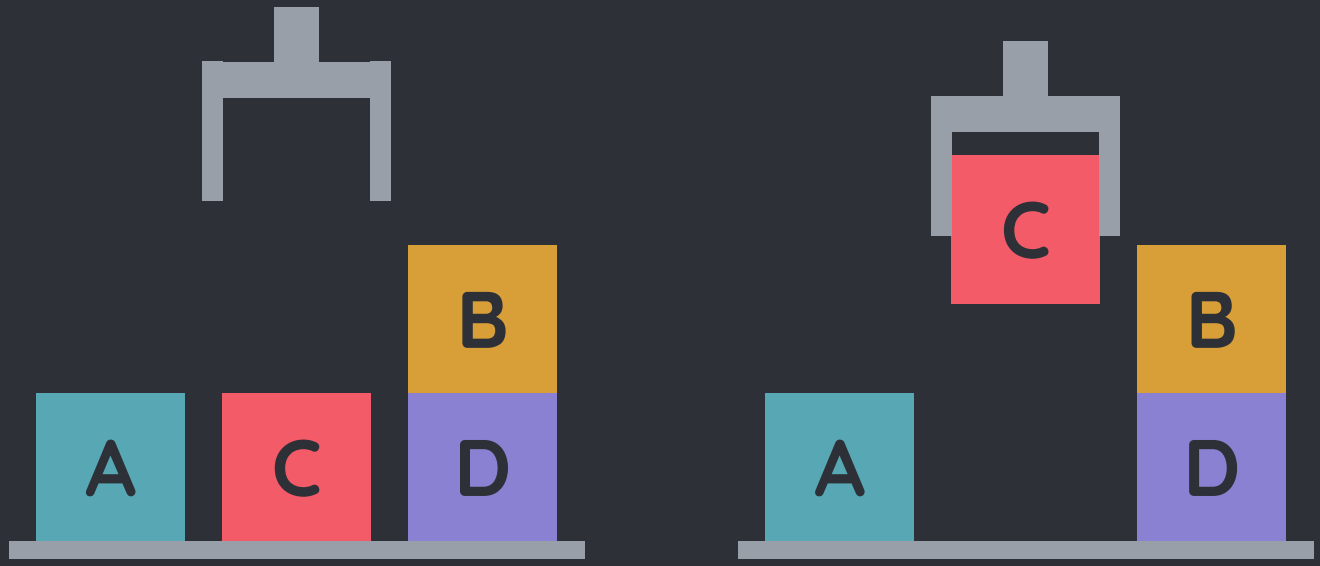
STACK
CLEAR(A) ∧ HOLDING(C)
STACK(C,A)
ON(B,D)
ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D)

new current state setelah operasi PICKUP(C):

ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D) ∧ HOLDING(C)

(Kondisi 3)

OPERASI PICKUP(C)



Langkah ke-3 (Iterasi ke-15)

STACK
CLEAR(A) ∧ HOLDING(C)
STACK(C,A)
ON(B,D)
ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D)

STACK
ON(B,D)
ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D)

current state

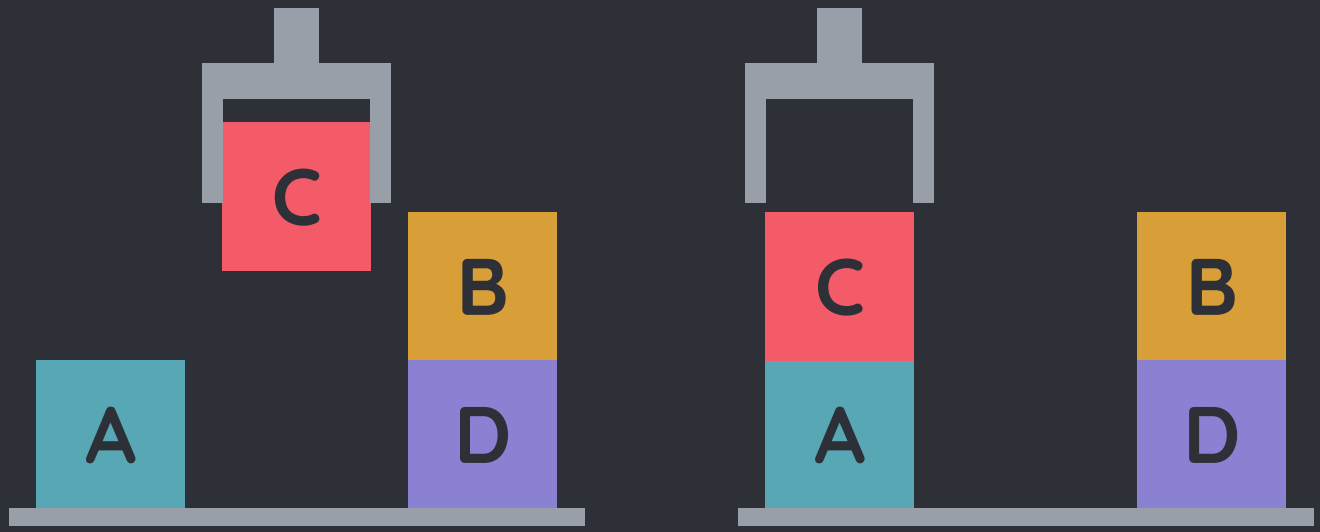
ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D) ∧ HOLDING(C)

new current state setelah operasi STACK(C,A):

ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D) ∧ ARMEMPTY

(Kondisi 3)

OPERASI STACK(C,A)



Langkah ke-3 (Iterasi ke-16)

STACK
ON(B,D)
ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D)

SELESAI

STACK
<EMPTY>

current state

ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D) ∧ ARMEMPTY

goal state

ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D)

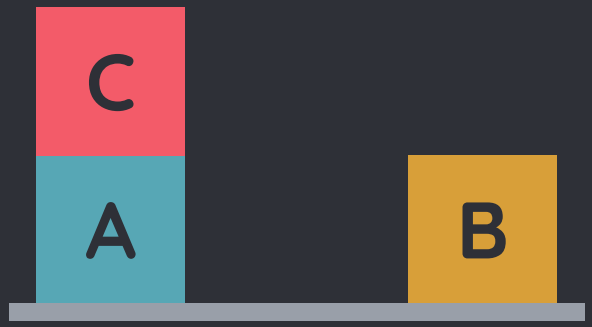
(Kondisi 4)

Rencana Penyelesaian

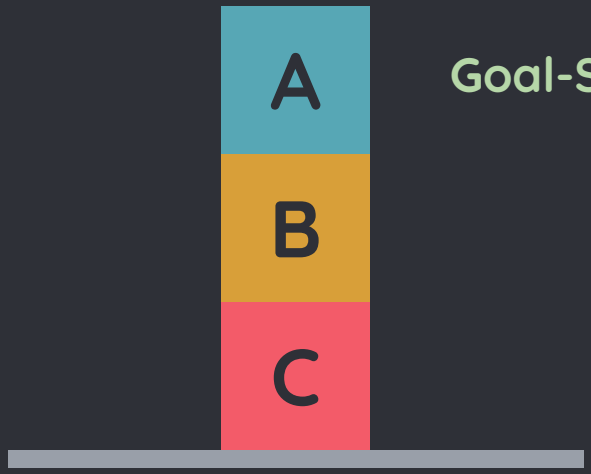
1. UNSTACK(B,A)
2. STACK(B,D)
3. PICKUP(C)
4. STACK(C,A)

Sussman Anomaly

Initial-State



Goal-State



- Mengilustrasikan kelemahan dari GSP
- Pada rencana penyelesaian, terdapat operasi yang "menggagalkan" operasi sebelumnya
- Terjadi karena GSP secara "naif" mengejar suatu *subgoal* X setelah memenuhi *subgoal* Y
- Langkah untuk mencapai *subgoal* X dapat menggagalkan *subgoal* Y
- Contoh: *subgoal* **ON(A,B)** dan *subgoal* **ON(B,C)**

Rencana Penyelesaian

1. UNSTACK(C,A)
2. PUTDOWN(C)
3. **PICKUP(A)**
4. **STACK(A,B)**
5. **UNSTACK(A,B)**
6. **PUTDOWN(A)**
7. PICKUP(B)
8. STACK(B,C)
9. PICKUP(A)
10. STACK(A,B)

Terima Kasih

ADA PERTANYAAN?