

Nama : Ukinda Feriando Setiawan  
NIM : 312210510  
Kelas : TI.22.C6  
Prodi : Teknik Informatika  
Matkul : Kecerdasan Buatan



## Ulangan Tengah Semester

### 1. Pengertian dan Penerapan AI

**Artificial Intelligence (AI)** atau **Kecerdasan Buatan** adalah sebuah bidang ilmu komputer yang bertujuan untuk menciptakan sistem atau mesin yang mampu meniru, mensimulasikan, dan bahkan melebihi kemampuan kognitif manusia, seperti belajar, memecahkan masalah, membuat keputusan, mengenali pola, dan memahami bahasa.

#### Dua Contoh Penerapan AI Sehari-hari:

1. **Asisten Suara Pribadi:** (e.g., Google Assistant/Siri) Menggunakan Natural Language Processing (NLP) untuk memahami perintah suara.
2. **Rekomendasi Konten/Produk:** Digunakan oleh platform *streaming* (Netflix, YouTube) atau *e-commerce* untuk memprediksi minat dan menyarankan item.

### 2. Intelligent Agent

**Intelligent Agent (Agen Cerdas)** adalah entitas otonom yang dapat **mengamati** (perceive) lingkungannya melalui sensor, **memproses** informasi tersebut, dan **bertindak** (act) terhadap lingkungan tersebut melalui aktuator untuk mencapai tujuannya.

### 3. Analisis Agent Cerdas: Mobil Self-Driving

| Kategori                       | Deskripsi (Mobil Otonom)  |
|--------------------------------|---|
| <b>P (Performance Measure)</b> | Keamanan (menghindari kecelakaan), kecepatan, kenyamanan, kepatuhan terhadap hukum lalu lintas. |
| <b>E (Environment)</b>         | Jalan raya, lalu lintas lainnya, pejalan kaki, rambu dan marka jalan, kondisi cuaca.            |
| <b>A (Actuators)</b>           | Setir/kemudi, rem, pedal gas/akselerator, lampu sinyal/sein, klakson.                           |
| <b>S (Sensors)</b>             | Kamera, LiDAR, Radar, GPS, Odometer.  |

#### Klasifikasi Lingkungan:

- **Sebagian Dapat Diamati** (Partially Observable)
- **Multi-Agent**
- **Stokastik** (Stochastic)
- **Sekuensial** (Sequential)
- **Dinamis** (Dynamic)

### 4. Perbandingan Algoritma Blind Search

| Algoritma                         | Strategi  | Memori (Memory)   | Efektifitas (Completeness & Optimality)  |
|-----------------------------------|---|---|--|
| <b>Breadth-First Search (BFS)</b> | Ekspansi simpul secara <b>horizontal</b> ; mencari solusi dengan kedalaman paling dangkal terlebih dahulu (tingkat demi tingkat). | <b>Buruk</b> (membutuhkan memori yang besar karena menyimpan semua simpul pada tingkat saat ini). | <b>Lengkap dan Optimal</b> (jika biaya langkah seragam).                         |
| <b>Depth-First Search (DFS)</b>   | Ekspansi simpul secara <b>vertikal</b> ; mencari solusi hingga  | <b>Baik</b> (hanya menyimpan jalur yang sedang dieksplorasi).                                     | <b>Tidak Lengkap</b> (jika ada <i>infinite path</i> ) dan <b>Tidak Optimal</b> . |

| Algoritma                        | Strategi  | Memori (Memory)   | Efektifitas (Completeness & Optimality)   |
|----------------------------------|---|---|---|
|                                  | kedalaman maksimum sebelum kembali ( <i>backtrack</i> ).  |   |   |
| <b>Uniform Cost Search (UCS)</b> | Ekspansi simpul berdasarkan <b>biaya jalur terendah</b> dari <i>initial state</i> ke simpul saat ini. | <b>Buruk</b> (menyimpan semua simpul yang belum diekspansi dalam <i>priority queue</i> ). | <b>Lengkap dan Optimal</b> (selalu menemukan jalur solusi dengan biaya terendah). |

## 5. Global Search vs. Local Search

### Perbedaan Utama:

- **Global Search:** Bertujuan menemukan solusi **terbaik secara keseluruhan** (*global optimum*). Melacak dan menyimpan keseluruhan jalur eksplorasi.
- **Local Search:** Bertujuan menemukan solusi **terbaik di sekitar** konfigurasi saat ini (*local optimum*). Hanya beroperasi pada satu atau beberapa solusi saat ini dan tidak menyimpan jalur/riwayat.

### Contoh dan Cara Kerja:

| Tipe Pencarian       | Contoh Algoritma     | Cara Kerja Singkat   |
|----------------------|----------------------|--|
| <b>Global Search</b> | <b>A* Search</b>     | Mencari jalur dengan biaya terendah menggunakan formula $f(n) = g(n) + h(n)$ , di mana $g(n)$ adalah biaya aktual yang sudah dikeluarkan, dan $h(n)$ adalah estimasi biaya tersisa ( <i>heuristik</i> ) ke tujuan. |
| <b>Local Search</b>  | <b>Hill Climbing</b> | Memulai dari simpul acak dan berulang kali bergerak ke simpul <b>tetangga</b> yang <b>meningkatkan</b> nilai fungsi tujuan. Berhenti ketika mencapai <i>local optimum</i> (tidak ada tetangga yang lebih baik).    |