

## UTS AI

Nama : Arya Saepul Hakim  
NIM : 312210608  
Kelas : TI.22.C.SE.1  
Dosen : Yogi Yulianto, M.Kom

1. Jelaskan pengertian Artificial Intelligence menurut Anda sendiri dan sebutkan dua contoh penerapan AI yang sering digunakan dalam sehari-hari.

Artificial Intelligence (AI) atau Kecerdasan Buatan adalah cabang ilmu komputer yang berfokus pada pembuatan sistem yang dapat meniru kecerdasan manusia, seperti berpikir, belajar, memahami bahasa, serta membuat keputusan secara mandiri.

AI memungkinkan mesin untuk memproses data, mengenali pola, dan menyesuaikan perilaku berdasarkan pengalaman.

Contoh penerapan AI dalam kehidupan sehari-hari:

1. Asisten virtual seperti Google Assistant atau Siri, yang memahami perintah suara dan menjawab pertanyaan pengguna.
2. Sistem rekomendasi di YouTube, Netflix, atau Shopee yang mempelajari preferensi pengguna untuk menampilkan konten atau produk yang sesuai.

## 2. Apa yang dimaksud dengan Intelligent Agent?

Intelligent Agent adalah entitas perangkat lunak atau sistem yang mampu mengamati lingkungannya melalui sensor, mengambil keputusan berdasarkan informasi yang diterima, dan melakukan aksi untuk mencapai tujuan tertentu secara mandiri (*autonomous*).

Agent yang cerdas dapat belajar dari pengalaman serta menyesuaikan tindakannya untuk meningkatkan kinerja dalam situasi yang dinamis.

Contoh: Robot pembersih lantai (seperti Roomba) yang memetakan ruangan, mengenali rintangan, dan menyesuaikan rute pembersihan secara otomatis.

## 3. Analisa 1 Agent cerdas lalu tentukan PEAS framework dan klasifikasi lingkungannya.

Contoh agent: Mobil otonom (*Self-Driving Car*).

### PEAS Framework

Komponen	Deskripsi
Performance Measure	Keamanan, efisiensi bahan bakar, waktu tempuh minimal, dan kenyamanan penumpang.
Environment	Jalan raya, kendaraan lain, pejalan kaki, rambu lalu lintas, dan kondisi cuaca.
Actuators	Kemudi, pedal gas, rem, lampu, klakson.

Komponen	Deskripsi
Sensors	Kamera, LIDAR, radar, GPS, sensor jarak, dan sensor kecepatan.

#### Klasifikasi Lingkungan

- Fully Observable: Tidak sepenuhnya, karena ada elemen yang tidak selalu terlihat (contoh: kondisi cuaca atau kendaraan di tikungan).
- Dynamic: Lingkungan berubah setiap waktu karena adanya kendaraan dan pejalan kaki.
- Continuous: Variabel lingkungan seperti kecepatan dan arah berubah secara kontinu.
- Multi-Agent: Berinteraksi dengan pengemudi atau kendaraan lain di jalan raya.

4. Bandingkan tiga algoritma Blind Search: Breadth-First Search (BFS), Depth-First Search (DFS), dan Uniform Cost Search (UCS) dalam hal strategi, memori, dan efektivitas.

Aspek	Breadth-First Search (BFS)	Depth-First Search (DFS)	Uniform Cost Search (UCS)
Strategi	Menelusuri node secara melebar dari root ke level berikutnya.	Menelusuri node sedalam mungkin sebelum kembali ( <i>backtrack</i> ).	Menelusuri node berdasarkan biaya lintasan paling rendah.

Aspek	Breadth-First Search (BFS)	Depth-First Search (DFS)	Uniform Cost Search (UCS)
Penggunaan Memori	Tinggi, karena semua node pada level yang sama disimpan di memori.	Rendah, hanya menyimpan node pada satu jalur saat ini.	Sedang, tergantung banyaknya node dengan biaya kecil.
Efektivitas	Optimal jika semua langkah memiliki biaya sama.	Tidak optimal dan bisa mengalami <i>infinite loop</i> tanpa batas kedalaman.	Optimal untuk semua lintasan dengan biaya berbeda.

5. Apa perbedaan utama antara Global Search dan Local Search? Berikan satu contoh algoritma dari masing-masing dan jelaskan cara kerjanya.

Perbedaan utama:

- Global Search menjelajahi seluruh ruang pencarian secara sistematis untuk menemukan solusi optimal global.
- Local Search berfokus pada perbaikan solusi saat ini dengan menjelajahi tetangga terdekat (local neighborhood) untuk mencari solusi yang lebih baik.

### 1. Global Search

- Ciri: Mengeksplorasi seluruh kemungkinan solusi dari awal.
- Contoh: *Breadth-First Search (BFS)*
  - Bekerja dengan menelusuri setiap level dari node akar secara melebar.

- Menjamin solusi yang ditemukan adalah solusi optimal jika semua langkah memiliki biaya yang sama.

## 2. Local Search

- Ciri: Tidak menjelajahi seluruh ruang, hanya memperbaiki solusi saat ini.
- Contoh: *Hill Climbing Algorithm*
  - Mulai dari solusi acak.
  - Mengevaluasi solusi tetangga dan berpindah ke yang memiliki nilai lebih baik.
  - Dapat berhenti pada *local optimum* jika tidak ada solusi yang lebih baik di sekitar.