

1. Rangkuman Video
 - *Introduction to AI*

Contoh penerapan AI dalam keseharian yaitu “*Path Finding*” atau pencarian rute dari satu titik ke titik yang lain. *Path Finding* akan menghitung jarak tiap rute yang selanjutnya memberikan rekomendasi jarak terdekat di antara semua alternatif yang ada. Selain itu juga ada “*AI Assistant*” yang membantu melakukan tugas atau *task* dan juga layanan berdasarkan perintah manusia. Contoh aplikasi yang sudah ada diantaranya Siri, Cortana, Maluba, dan Google Assistant. Contoh lainnya dalam bidang biometrik yaitu pengenalan sidik jari dan *face verification* untuk mengaktifkan *smartphone*. Pemanfaatan AI lainnya yang mulai populer yaitu pembuatan sketsa wajah baru berdasarkan satu wajah saja. Contoh lain penerapan AI ada *game digital free to play* pada mobile. Dalam dunia *entertainment* ada aplikasi Spotify yang mampu menyimpan *preference* dari pengguna. AI juga merambah ke kendaraan tanpa pengemudi dengan penentuan keputusan berdasarkan kondisi dari sekeliling kendaraan.

Definisi AI sendiri dapat dibedakan berdasarkan 2 sudut pandang, yakni dari cara berpikir (*Thinking*) atau cara berlaku (*Acting*) dan dari cara peniruan perbuatan manusia (*Humanly*) atau bertindak secara rasional (*Rationally*). Dari kedua sudut pandang tersebut terdapat 4 pendekatan definisi AI, yaitu:

- a. *Acting Humanly*

Menurut Kurzweil, 1990 AI adalah seni membentuk mesin dimana fungsi tersebut memerlukan kecerdasan ketika dilakukan oleh manusia. Contoh aplikasinya adalah komputer yang bisa mendengarkan perintah, mengenali gambar, dan lain – lain. Sistem harus melewati “*Turing Test*” yang membutuhkan manusia sebagai *Human Interpregator*, manusia sebagai *Human*, dan AI System untuk memberikan respon kepada *Human Interpregator*. Posisi *Human Interpregator* harus terpisah dengan *Human* dan *System* agar *Human Interpregator* mampu membedakan apakah inputan berasal dari *Human* atau *System*. Aplikasi yang menerapkan *Acting Humanly* misalnya PXDES, aplikasi yang melakukan diagnosa x-ray untuk penyakit paru – paru yang disebabkan oleh debu.

- b. *Thinking Humanly*

Menurut Bellman, 1978 AI adalah proses otomatis dari aktivitas yang berhubungan dengan proses berpikir manusia seperti pengambilan keputusan, pemecahan masalah, belajar, dan lain sebagainya. Sistem dikatakan sebagai AI jika sistem tersebut merupakan model komputasi dari proses berpikir manusia. Sistem ini menirukan cara berpikir manusia dengan mengamati dari dalam. Aplikasi yang menerapkan *Thinking Humanly* misalnya Sighthound Video, *software* pengawasan kamera agar lebih cerdas yang dapat membedakan manusia, kendaraan, dan hewan.

- c. *Thinking Rationally*

Menurut Charniak dan McDermott, 1985 AI adalah ilmu yang mempelajari kemampuan mental dengan memanfaatkan model komputasional. Sistem dikatakan AI jika dapat mengkodekan proses pemikiran yang benar sesuai dengan aturan – aturan berpikir rasional atau hukum filosofi.

Contohnya Vampire Theorem Prover, aplikasi untuk membuktikan teorema. *Theorem Prover* sendiri banyak digunakan untuk komponen sebagai otak dari peralatan elektronik.

d. *Acting Rationally*

Menurut Poole dan teman – teman 1998, AI adalah ilmu yang mempelajari desain dari suatu agen berinteleksi. Sistem dikatakan AI jika mampu berperilaku secara rasional, artinya berbekal pengetahuan dan tujuan yang ingin dicapai, agen bisa mencapai tujuan tersebut. Sistem ini berfokus pada pencarian, misalnya *Path Finding* yang sudah dijelaskan sebelumnya dan beberapa algoritma seperti Greedy, A-star, DFS, BFS, Genetic Algorithm, dan lain sebagainya. Contoh aplikasinya adalah Jobshop Visual Production Scheduling, aplikasi yang membantu dalam hal penjadwalan yang penyelesaiannya mempertimbangkan batasan yang telah ditetapkan mulai awal.

- *Intelligent Agent*

a. *Agent & Environment*

Agent menangkap atau *percept* keadaan yang ada pada lingkungan melalui sensor. Keadaan yang ditangkap oleh *Agent* bergantung pada desain yang dibuat oleh desainer. Selanjutnya desainer juga mendefinisikan aksi untuk *Agent*. Setelah proses penentuan aksi selesai kemudian *Agent* akan melakukan aksi melalui *Actuators* yang menempel pada *Agent*. Proses ini berkelanjutan hingga *Agent* mencapai tujuan yang diinginkan. Sehingga *Agent* dapat didefinisikan sebagai *Agent* komputasi yang bisa bertindak secara *autonomous*, artinya *Agent* dapat bertindak sendiri tanpa dipandu langkah per langkah oleh manusia atau entitas lainnya. Contoh sensor pada robot lego untuk menyelesaikan kubus rubik adalah kamera untuk membaca warna pada rubik. Sedangkan *Actuators*-nya berupa motor – motor.

b. *Agent Model*

Pada pembahasan ini mengambil contoh *Agent* sebagai penghisap debu. *Agent* dirancang untuk menangkap lokasi dan isi dari lingkungan. *Percept* yang ditangkap yaitu lokasi dan kondisi lingkungan yaitu kotor. Aksi yang dilakukan yaitu bergerak ke kiri dan ke kanan, menghisap kotoran, dan tidak melakukan apapun. Untuk membuat model *Agent* harus memiliki kumpulan persepsi, aksi apa saja yang bisa dilakukan, *state* atau kondisi dalam program, fungsi untuk memetakan *state*, dan dinamika dunia *Agent*. Untuk menilai *Agent* sudah sesuai harapan atau belum dapat dengan menggunakan nilai utilitas yang memetakan *state* ke dalam bilangan real tertentu, *Agent* dapat memilih aksi yang benar.

c. *Rational Agent*

Rational Agent adalah setiap *sequence percept* yang mungkin dan pengetahuan yang dimiliki oleh *Agent* sehingga *Agent* dapat memilih aksi yang dapat memaksimalkan pengukuran kinerja. Rasional yang digunakan pada AI adalah komputasi yang terbatas karena adanya batasan komputasi. Untuk mengoptimalkan *Agent* yang dibangun perlu adanya definisi mengenai kinerja, lingkungan, *actuator* dan sensor.

d. PEAS

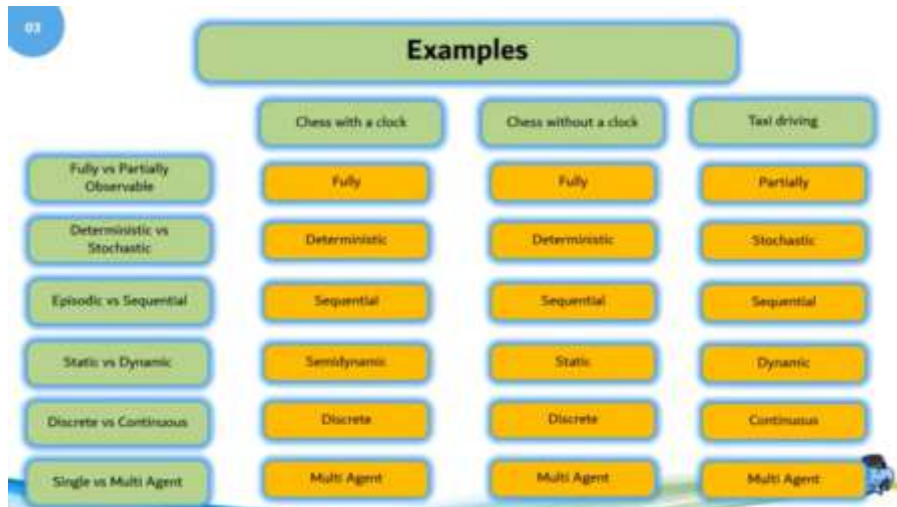
PEAS merupakan hal yang harus didefinisikan mulai awal dalam membangun *Rational Agent*. Berikut pengertian dan contoh PEAS pada pengemudi taxi *autonomous*.

- *Performance Measure*, yaitu pengukuran kinerja untuk menilai apakah *Agent* yang dibangun sudah bekerja dengan baik atau belum. *Performance Measure* yang bisa ditentukan misalnya keamanan mengemudi, kecepatan sampai di tujuan, tidak melanggar peraturan lalu lintas, kenyamanan penumpang dan keuntungan yang didapatkan maksimal.
- *Environment*, yaitu lingkungan dimana *Agent* akan ditempatkan. *Environment* yang bisa ditentukan misalnya jalan, kendaraan sekitar, pejalan kaki, dan penumpang taksi.
- *Actuators*, yaitu bagian dari *Agent* yang dapat melakukan aksi dan berdampak pada lingkungan. *Actuators* bukan aksi yang dilakukan *Agent*. *Actuators* yang dapat ditentukan misalnya kemudi untuk menentukan arah kendaraan, pedal gas, rem, tuas untuk mengaktifkan lampu ke kanan atau ke kiri, dan klakson.
- *Sensor*, bertugas menangkap *percept* dari lingkungan. *Sensor* yang bisa ditentukan misalnya kamera, sonar, GPS, keyboard, dan lain – lain.

e. *Task Environment*

Jenis *Task Environment* yaitu:

- *Fully vs Partially Observable*. *Fully Observable* ditandai dengan sensor *Agent* mampu menangkap semua *state* lingkungan setiap saat, sedangkan *Partially Observable* sebaliknya.
- *Deterministic vs Stochastic*. *Deterministic* ditandai jika *state* lingkungan selanjutnya hanya bergantung pada *state* saat ini dan aksi yang dilakukan *Agent*. Sedangkan *Stochastic* terdapat ketidakpastian yang membuat *state* berikutnya memiliki peluang yang berbeda.
- *Episodic vs Sequential*. *Episodic* ditandai dengan kinerja *Agent* yang dibagi menjadi episode – episode *atomic* yang tidak saling bergantung episodenya. Sedangkan *Sequential* aksi *Agent* saat ini dipengaruhi oleh aksi *Agent* sebelumnya.
- *Static vs Dynamic*. *Static* ditandai dengan lingkungan *Agent* yang tidak berubah saat *Agent* berpikir aksi apa yang akan dilakukan. Sedangkan *Dynamic* sebaliknya.
- *Discrete vs Continuous* penentuannya bergantung lingkungan, bagaimana waktu ditangani, *percept*, dan aksi.
- *Single vs Multi Agent*. *Single Agent* ditandai dengan perubahan *state* selanjutnya dipengaruhi oleh *Agent* yang melakukan aksi. Sedangkan *Multi Agent* dipengaruhi oleh *Agent* yang lain.



Struktur *Agent* terdiri dari *Architecture* yang menyatakan perangkat komputasi yang dilengkapi sensor serta aktuator dan Program yang dibedakan menjadi:

- *Simple Reflex Agent*, langsung melakukan aksi dari *percept* yang sudah didefinisikan.
- *Model-based Reflex Agent*, bisa membentuk model lingkungan ketika *Agent* berhadapan dengan *Partially Observable*.
- *Goal-based Agent*, pemilihan aksinya bukan hanya dari kondisi lingkungan saat ini tapi juga berdasarkan informasi goal yang akan dicapai.
- *Utility-based Agent*, perbaikan dari *Goal-based Agent* saat ada 2 pilihan yang sama.
- *Learning Agent*, mampu memperbaiki kinerjanya dari waktu ke waktu berdasarkan pengalaman yang disimpan.
- f. *Agent Level*
 - Level 1: *Problem Solving Agent*

Mendefinisikan semua *state* persoalan yang ditangkap *Agent* dan semua aksi yang bisa dipilih untuk mencapai tujuan. Kemudian *Agent* melakukan pencarian aksi apa yang akan dilakukan. Tidak ada penalaran khusus karena yang dilakukan adalah memilih aksi yang bisa membuat *Agent* mendekati tujuannya. Algoritma yang bisa diterapkan misalnya DFS, BFS, IDS, dan lain – lain.

- Level 2: *Knowledge Based Agent*

Tidak mendefinisikan semua *state* namun memberikan premis yang bisa dimanfaatkan untuk penalaran.

- Level 3: *Learning Agent*

Tidak ada informasi yang diberikan kepada *Agent* mengenai *state* atau premis, namun yang diberikan adalah kumpulan data hasil observasi selama beberapa waktu. *Agent* akan menentukan pengetahuan dasarnya sendiri berdasarkan data yang diberikan. Berdasarkan ketersediaan data terdapat 3 jenis pembelajaran, yakni:

- 1) *Supervised Learning*, ada data hasil observasi dan nilai yang dicari.

- 2) *Unsupervised Learning*, ada data hasil observasi tapi tidak ada kelas dari setiap data yang ada.
- 3) *Reinforcement Learning*, tidak ada data dan *Agent* harus belajar sendiri secara berulang – ulang.

- *Introduction to Knowledge Based Agent*

Knowledge-based System (KBS) dapat diartikan sebagai mengaplikasikan pengetahuan. Jika yang dilibatkan pengetahuan dari pakar disebut dengan *Knowledge-based Expert System* atau *Expert System*. Di dalam KBS dilakukan penalaran berbasis logika. KBS memiliki 2 komponen utama yaitu *Knowledge Base* yang berisi kumpulan pengetahuan dan *Inference Engine*. *Knowledge Base* di sini belum tentu *Expert System*. 4 pendekatan pada *Pattern Recognition* yaitu:

- a. *Template-based*, *Knowledge Base* berupa kumpulan template dan inferensinya berupa *template matching*.
- b. *Statistical-based*, *Knowledge Base* berupa model hasil pembelajaran dan inferensinya sesuai dengan representasi model tersebut.
- c. *Structural / Syntactic*, *Knowledge Base* berupa struktur yang dibangun oleh sejumlah fitur dan inferensinya menelusuri struktur tersebut.
- d. *Deep Learning-based*, *Knowledge Base* berupa model ANN



Logical Reasoning pada *Knowledge-based Agent* pada setiap langkahnya *Agent* akan melakukan *matching* antara *Knowledge Base* yang dimiliki dengan data spesifik yang didapatkan sehingga dihasilkan *Conclusion* dari data yang ada. Development pada *Knowledge-based Agent* dibagi menjadi 3 tahap yaitu: dimulai dari *Knowledge Base* yang kosong (*Starting with an empty Knowledge Base*) tapi masih bisa digunakan, *Agent* desainer dapat memasukkan pengetahuan satu per satu (*Agent Designer Can Tell Sentence One by One*), *Agent* menjadi tahu bagaimana beroperasi di dalam lingkungannya (*Agent Knows How to Operate in its Environment*). *Knowledge Representation* adalah suatu bahasa yang dipakai untuk merepresentasikan *Knowledge* berupa sintaks dan semantik untuk merepresentasikan apa yang ada pada pengetahuan tersebut. Contohnya adalah logika proposisi dan predikat. *Requirements* yang dibutuhkan di antaranya tidak ada kontradiksi, setiap simbol harus unik, bisa menjelaskan objek; relasi; dan atribut,

manipulasinya efisien. Cara memilih representasi: cocok untuk domainnya, kecocokan dengan *task*, dan kecocokan dengan *user*.

3 alasan mengapa *Agent* membutuhkan komponen *Learning* yaitu: ketika *Agent* akan beroperasi pada lingkungan yang tidak diketahui dan desainer juga tidak mendefinisikannya, ketika *Learning* dapat menggantikan peran dari *Agent* desainer untuk menambahkan pengetahuan dalam proses pengembangan sistem, dan ketika penambahan pengetahuan oleh komponen *Learning* dapat memperbaiki performa sistem. *Knowledge-based Agent with Learning*, modul *Learning*-nya dibagi menjadi beberapa modul, yaitu:

- a. *KBA with Learning*, sama dengan *Learning Agent*. Memiliki 4 komponen penting yakni *Performance Element* yang menentukan aksi apa yang akan dilakukan *Actuators* berdasarkan persepsi yang diterima sensor, *Learning Element* untuk melakukan perubahan pengetahuan terhadap *Performance Element*, *Critic* untuk mengukur seberapa baik aksi yang dilakukan *Agent*, dan *Problem Generator* untuk menyarankan aksi eksplorasi yang mungkin dibutuhkan *Agent*. *Learning Element* memberikan perubahan pada sistem sehingga bersifat adaptif dalam artian *task* yang diberikan menjadi lebih efisien dan efektif. Dalam mendesain *Learning Element* perlu 4 hal yang perlu dipertimbangkan, di antaranya komponen apa dari *Performance Element* yang akan diperbaiki kinerjanya, ketersediaan *Prior Knowledge*, *feedback* yang tersedia, dan representasi yang digunakan.
- b. *Supervised Learning*, tanpa ada *feedback* hanya diberikan pasangan *input output*.
- c. *Concept Learning*.
- d. *kNN & Naive Learning*.
- e. *Unsupervised Learning*, tanpa ada *feedback* hanya diberikan kumpulan data tanpa label lalu mencari kluster terbaik dari data tersebut.
- f. *Reinforcement Learning*, *Agent* akan belajar dari series *Reinforcement* yang didapatkan berupa *Reward* atau *Punishment*.

Knowledge-based System merupakan gabungan dari *Problem-solving method* yaitu algoritma yang menentukan bagaimana cara menyelesaikan persoalan, *Knowledge*, dan Data. *Knowledge-based System* dibedakan menjadi 2 jenis, yaitu:

- a. *Interactive*, *user* berhubungan langsung dengan KBS melalui *user interface* yang ada
- b. *Embedded*, sistem berbasis pengetahuan berinteraksi dengan subsistem lain dari sistem yang lebih besar atau mengakses eksternal basis data. Sistem yang lebih besar ini menyediakan *interface*.

Knowledge Acquisition merupakan proses mengekstraksi pengetahuan pakar dan mengkonversinya ke dalam bentuk yang bisa digunakan komputer untuk menyelesaikan persoalan. *Knowledge Acquisition* dibagi menjadi:

- a. *Direct*, pakar melakukan formalisasi dan strukturalisasi pengetahuan secara langsung. Hasil pengujian KBS diberikan ke pakar untuk verifikasi.
- b. *Indirect*, menggunakan *Knowledge Engineer* untuk mengekstraksi elisitasi pengetahuan domain dan strategi dari pakar dalam bentuk dialog atau *interview* yang berulang.

Knowledge Engineer juga melakukan representasi dalam bentuk program deklaratif untuk mengisi *Knowledge Base*, lalu pengujiannya diberikan kepada pakar.

General Architecture of KBS: Domain Specific Knowledge Base (DSKB) yang berisi pengetahuan domain, *Case Specific Facts (CSF)* untuk menyimpan fakta – fakta, dan *Intermediate Result and Problem Solution* untuk menyimpan hasil sementara dari inferensi. *Control System* terdiri dari *Problem Solving Component* sebagai mesin inferensi dan *Interface* ke *user* berupa *submodul interviewer, explanation, dan Knowledge Aquisition*. Ketika *user* berinteraksi melalui *interviewer* fakta – fakta yang didapat kemudian disimpan di *CSF*. Kemudian dilakukan *matching* oleh *Problem Solving Component* dan hasilnya disimpan di *Intermediate Results*. Selanjutnya hasil ditampilkan oleh *interviewer* ke *user*.

2. P (*Performance Measure*): kecepatan membersihkan kotoran, melewati halangan.

E (*Environment*): ruangan, halangan, kotoran,

A (*Actuators*): berupa roda untuk bergerak ke kanan, kiri, depan, belakang, berputar. Alat untuk menghisap.

S (*Sensor*): sensor ultrasonik, mikrokontroller, LCD.

3. Sistem Deteksi Masker

Sistem ini dibangun untuk mendeteksi orang – orang apakah menggunakan masker atau tidak. Mengingat saat ini menggunakan masker merupakan kewajiban maka sistem ini dapat dimanfaatkan untuk memberikan himbauan atau teguran kepada orang yang tidak menggunakan masker. Teknik yang digunakan yaitu *Knowledge* dengan memberikan pengetahuan kepada *Agent* bagaimana tampilan orang yang bermasker dan tidak bermasker. Sistem ini menggunakan pengenalan wajah sehingga mampu membedakan orang dengan masker dan tanpa masker. Selanjutnya sistem akan menyimpan datanya untuk selanjutnya bisa diolah oleh pihak yang berwenang.

4. Metodologi dalam beberapa Jurnal

- Sihombing, Eka N.A.M. & Syaputra, Muhammad Yusrizal Adi. (2020). *Implementasi Penggunaan Kecerdasan Buatan Dalam Pembentukan Peraturan Daerah*. 14(3), 419-434.

Metode penelitian diawali dengan tahap Pendekatan yang didalamnya menerapkan metode penelitian hukum yuridis normatif, suatu penelitian yang menganalisis hukum baik yang tertulis di dalam buku (*Law as it is Written in the Book*). Kemudian metode Pengumpulan Data menggunakan teknik studi pustaka dengan melakukan studi dokumen pada perpustakaan pribadi dan perpustakaan hukum. Selanjutnya teknik Analisa Data yang merupakan proses pengorganisasian dan mengurutkan data ke dalam kategori – kategori dan satuan uraian dasar, sehingga ditemukan tema dan dapat dirumuskan hipotesis kerja, seperti disaranka oleh data.

- Fauzan, Ivan. (2020). *Artificial Intelligence (Ai) Pada Proses Pengawasan Dan Pengendalian Kepegawaian – Sebuah Eksplorasi Konsep Setelah Masa Pandemi Berakhir*. 14(1), 31 – 42.

Metode penelitian menggunakan pendekatan kualitatif yang selanjutnya metode penelitian yang digunakan adalah metode deskriptif. Pada proses pengumpulan data penelitian kualitatif deskriptif berfokus pada penemuan sebuah ciri atau sifat dari sebuah peristiwa spesifik yang diteliti.

- Nugraha, Dedi & Winiarti, Sri. (2014). *Pengembangan Media Pembelajaran Sistem Pelacakan Pada Mata Kuliah Kecerdasan Buatan Berbasis Multimedia*. Jurnal Sarjana Teknik Informatika, 2(1), 67 – 77.

Metode penelitian diawali dengan Pengumpulan Data menggunakan *Study Literature* dengan cara membaca buku – buku referensi yang berkaitan dengan Kecerdasan Buatan, Pengumpulan Data dari Internet dengan cara mencari data dan informasi berupa citra, Wawancara dengan cara tanya jawab langsung kepada dosen pengampu dan mahasiswa, dan Observasi dengan cara mengamati secara langsung terhadap cara pembelajaran. Kemudian Analisis Kebutuhan dan dilanjutkan Perancangan Sistem dengan cara membuat perancangan sistem terhadap permasalahan yang dibahas.