



**LABORATORIUM PEMBELAJARAN ILMU KOMPUTER**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS BRAWIJAYA**

BAB : AGEN CERDAS  
NAMA : REZA AZZUBAIR WIJONARKO  
NIM : 155150200111182  
TANGGAL : 22/03/2017  
JENIS : TUGAS  
ASISTEN : - ANNISA FITRIANI NUR  
- RISKI PUSPA DEWI D. P.

ACC

**A. DEFINISI MASALAH**

1. Carilah 2 aplikasi dari 5 tipe agen
  - Simple reflex agents
  - Model-based reflex agents
  - Goal-based reflex agents
  - Utility-based reflex agents dan
  - Learning agents

Berikan review masing-masing aplikasi, kenapa aplikasi tersebut masuk pada tipe agen tertentu ?

2. Pilih satu aplikasi dari 5 tipe agen pada nomor 1, kemudian tentukan PEAS dari aplikasi tersebut!

3. Modifikasi kode program pada Tabel 1.1 menjadi model-based reflex agents dengan model sampah-sampah yang bisa dibersihkan oleh vacuum cleaner, simpan sebagai database yang diketahui oleh agen, buat task environment seperti pada latihan dengan detail yang lebih jelas pada tiap-tiap lokasi (contoh : lokasi 1 memiliki sampah kertas, plastic, debu, rambut, kulit buah).

## B. JAWAB

### 1. Tipe-tipe agen

#### a. Simple reflex agents

- 1) Sistem penerima tingkat kepuasan pasien pada pasien psikiatri. Wujud dan cara kerja sistem: sistem terdiri atas aktuator berupa monitor layar sentuh dan komputer. Setelah pasien selesai menemui dokter, sistem akan menampilkan tulisan “Apakah Anda puas dengan layanan dokter X?” dengan “X” merupakan nama dokter yang memeriksanya. Di bawah tulisan itu ada dua gambar besar yaitu gambar centang hijau dan silang merah. Bila pasien mengetuk tanda centang hijau, sistem akan menampilkan “Terima kasih atas input Anda” pada monitor. Bila pasien mengetuk tanda silang merah, sistem akan menampilkan “Mohon bila Anda berkenan, isikan keluhan Anda pada baris-baris di bawah ini” dengan menampilkan baris-baris yang bisa diisi kata dengan virtual keyboard yang dimunculkan seketika. Data input pasien tersebut akan dikirimkan oleh sistem ke sistem yang lebih besar untuk dilakukan penilaian pada si dokter. Karena sistem hanya bekerja menyediakan output yang sederhana dan memproses input secara sederhana, sistem dikatakan sebagai simple reflex agent.
- 2) Sistem pendeteksi logam pada bandara. Wujud dan cara kerja sistem: sistem berbentuk kotak yang berdiri yang bagian tengahnya kosong. Tinggi dan lebar cukup sehingga orang dewasa bisa mudah melewatinya. Sistem didirikan tegak lurus terhadap lantai sehingga orang-orang yang hendak masuk bandara harus melewatinya terlebih dahulu. Bila seseorang membawa benda logam, sistem akan berbunyi dan menyalakan kedipan-kedipan berwarna merah agar pihak keamanan dapat segera menanggapi kejadian tersebut. Karena sistem bekerja dengan mendeteksi logam dan melakukan sesuatu ketika mendeteksinya secara sederhana, sistem dikatakan sebagai simple reflex agent.

#### b. Model-based agents

- 1) Sistem pemanas ruangan pada negara-negara empat musim. Wujud dan cara kerja sistem: sistem terdiri atas aktuator sensor suhu dalam dan luar ruangan, dan pemanas (boiler) di ruang bawah tanah sistem. Sistem menaikkan dan menurunkan suhu ruangan dengan mengecek suhu luar ruangan. Sistem juga menyimpan history pengukuran suhu luar ruangan agar sistem bisa menebak musim apa yang sedang terjadi; mengukur tingkat

berkembangnya dunia. Dengan history itu, sistem bisa lebih akurat dalam menentukan waktu menaikkan dan menurunkan suhu ruangan serta menentukan suhu yang tepat. Karena sistem bekerja atas percept history yang ia kumpulkan untuk memprediksi bagaimana dunia berubah dan melakukan aksi-aksi atas percept history-nya, sistem dikatakan sebagai model-based reflex agent.

- 2) Sistem peramal hujan. Wujud dan cara kerja sistem: sistem terdiri atas sensor-sensor pendeteksi intensitas cahaya matahari, angin, kelembapan, dan hujan. Sensor-sensor itu terhubung dengan komponen kalkulasi dan storage. Data yang dikumpulkan tersimpan dalam storage. Sistem memprediksi apakah hujan akan terjadi atau tidak dengan melakukan kalkulasi terhadap data yang ia miliki. Karena sistem bekerja atas percept history yang ia kumpulkan untuk memprediksi bagaimana dunia berubah dan melakukan aksi-aksi atas percept history-nya, sistem dikatakan sebagai model-based reflex agent.

c. Goal-based reflex agents

- 1) Sistem pengatur lalu lintas jalan raya. Wujud dan cara kerja sistem: sistem terdiri atas sensor kamera di jalan-jalan yang ditentukan untuk mengukur tingkat kepadatan lalu lintas, dan komponen-komponen komputasi dan storage. Sistem akan terhubung dengan pengatur lampu lalu lintas untuk mengatur tingkat kepadatan lalu lintas itu. Dalam praktiknya, sistem memiliki tujuan (goal) bila jalan raya sangat padat, panjang kemacetan tidak akan lebih dari 1 km. Dengan tujuan seperti itu, sistem akan menyesuaikan lama lampu lalu lintas yang menyala di tiap pertigaan atau perempatan agar tujuan tetap tercapai. Karena sistem memiliki informasi tujuan yang hendak dicapai dengan beradaptasi menggunakan pengetahuan yang dimilikinya, sistem dikatakan sebagai goal-based reflex agent.
- 2) Sistem pengatur lalu lintas bandar udara. Wujud dan cara kerja sistem: sistem memiliki kemampuan untuk melacak keberadaan pesawat, waktu pergi, terbang, dan mendarat pesawat, serta tingkat kepadatan bandara. Dengan prinsip kerja yang mirip pada nomor c.1), sistem akan mengatur lalu lintas sedemikian rupa sehingga sistem akan memperoleh tujuan tidak ada pesawat yang terbang atau datang dari atau pada tempat dan/atau waktu yang bersamaan. Karena sistem memiliki informasi tujuan yang hendak dicapai dengan beradaptasi menggunakan pengetahuan yang dimilikinya, sistem dikatakan sebagai goal-based reflex agent.

d. Utility-based reflex agents dan

- 1) Sistem pemilih rute jalan yang diimplementasikan pada robot yang bisa berjalan. Wujud dan cara kerja sistem: sistem terdiri atas sensor kamera untuk melihat hambatan di jalan yang sedang ditempuh dan aktuator kaki besar tahan air untuk berjalan. Hambatan yang dimaksud berarti genangan air kecil, batu besar, atau tembok. Sistem bisa memilih jalan lain untuk menghindari suatu hambatan untuk berjalan ke tempat yang dituju. Namun, ketika jalannya terhambat (misal oleh tembok) dan jalan-jalan alternatifnya terdapat hambatan yang lain (misal oleh batu besar dan genangan air kecil), sistem akan bisa memutuskan memilih alternatif yang mana agar ia bisa sampai ke tujuannya dengan baik. Sistem akan dapat memutuskan untuk berjalan melalui air karena kakinya tahan air dan tidak memutuskan untuk berjalan melalui batu besar karena bisa saja dia terjatuh ketika akan melewati batu besar itu. Karena sistem bisa memikirkan jalan mana yang terbaik untuk kebaikan dirinya, sistem dikatakan sebagai utility-based agent.
- 2) Sistem pemilih tempat pendaratan yang diimplementasikan pada drone yang terbang tanpa bantuan manusia (autoflight drone). Wujud dan cara kerja sistem: sistem terdiri atas sensor kamera untuk melihat sekelilingnya dan aktuator baling-baling untuk terbang. Sistem dapat terbang dan mendarat kapan pun dan dari mana pun, selama tempat pendaratannya aman bagi dirinya. Menggunakan sensor kamera, sistem akan menentukan apakah tempat yang akan ia darati aman. Arti aman berarti rata, tidak basah, tidak ramai, dan tidak berlubang. Karena sistem bisa memikirkan tempat pendaratan yang baik untuk kebaikan dirinya, sistem dikatakan sebagai utility-based agent.

e. Learning agents

- 1) Sistem asisten pribadi Google Assistant yang diimplementasikan pada smartphone Android. Wujud dan cara kerja sistem: sistem terdiri atas semua sensor input yang ada pada smartphone dan aktuator speaker dan layar untuk menunjukkan suatu aksi atau kejadian (event). Sistem dapat melakukan penjadwalan keberangkatan dan kepulangan si pengguna berdasarkan jadwal terkini, penyedia berita sesuai dengan preferensi pengguna, penyedia layanan peramal cuaca, dan penyedia hasil pencarian melalui Google yang sesuai dengan preferensi pengguna. Asisten pribadi ini melakukan tugasnya menggunakan pengetahuan awal yang dimilikinya, dan lambat laun menjadi semakin akurat dan berguna bagi penggunaannya berdasarkan evaluasi penggunaannya.

Evaluasi penggunaannya bisa berbentuk input layar yang mengatakan bahwa asisten pribadi sudah/tidak memberi waktu perjalanan yang baik, sudah/tidak memberi berita yang tepat, sudah/tidak memberi informasi cuaca yang benar, sudah/tidak memberi hasil pencarian yang relevan. Karena sistem bisa belajar dari pengguna dan mengevaluasi dirinya berdasarkan preferensi pengguna, sistem dikatakan sebagai learning agent.

- 2) Sistem olah vokal (bicara) yang diimplementasikan pada robot yang dapat berbicara. Wujud dan cara kerja sistem: sistem terdiri atas sensor pengenalan suara sebagai input dan komponen komputasi, storage, learning element, dan problem generator untuk keperluan belajar, dan aktuator speaker atau sejenisnya sebagai output suara. Pengkritik (dalam hal ini manusia) akan mengajarkan robot cara berbicara dengan menyampaikan robot satu kata tiap waktunya agar robot mengenal kata itu. Robot akan menyebutkan kembali kata itu setelah mendengarnya. Agar si robot alami dalam mengucapkan kata itu, si pengkritik akan menyebutkan kata itu berulang-ulang hingga pengkritik merasa bahwa robot sudah dapat menyebutkan suatu kata dengan jelas. Bila sudah melewati waktu tertentu yang cukup, robot kemudian diajari oleh pengkritik untuk berbicara dalam kalimat dan proses belajar dan mengulang di atas dimulai lagi agar kemampuan bicara robot menjadi alami seperti manusia. Karena sistem bisa belajar dari pengguna dan mengevaluasi dirinya berdasarkan preferensi pengguna, sistem dikatakan sebagai learning agent.

2. PEAS dari sistem pendeteksi logam di bandara.

- a. Performance measure: kecepatan menilai logam, kecepatan menyalakan lampu dan suara ketika mendeteksi logam, kemampuan membedakan logam dengan barang lain yang mirip logam (besi dengan teflon, seng dengan karbon,...)
- b. Environment: bandara, orang yang melaluinya
- c. Actuator: suara deteksi logam, lampu deteksi logam
- d. Sensor: pendeteksi logam

### 3. Tugas

AgenCerdasEx.java

```
package praktikumkb1;
public class AgenCerdasEx {

    private String lingkungan[][];
    private int row, kolom = 3;

    AgenCerdasEx(int location) {
        lingkungan = new String[location][4];
        row = location / kolom;
    }

    AgenCerdasEx(int baris, int kol) {
        kolom = kol;
        row = baris;
        lingkungan = new String[row * kolom][4];
    }

    public void setKondisi(int i, String Nama,
String location, String status) {
        lingkungan[i - 1][0] = Nama;
        lingkungan[i - 1][1] = location;
        lingkungan[i - 1][2] = status;
    }

    public void vacumm(int i, String status) {
        int next = cekAr(i);
        if ("kotor".equals(status.toLowerCase())) {
            System.out.println(lingkungan[i][0] + "
("+lingkungan[i][1]+")");
            System.out.println("----bersih-bersih--
--");
            status = "bersih";
            lingkungan[i][2] = status;
            System.out.println(lingkungan[i][1]+ "
sudah bersih");
            geser(i, next);
        }
    }

    private boolean[] cekGerak(int i) {
        int Ver = lingkungan.length / row;
        int index = lingkungan.length - 1;
        boolean[] move = new boolean[4];
        for (int j = 0; j < move.length; j++) {
            move[j] = true;
        }
        if ((i + 1) % Ver == 0) {
            move[0] = false; //move 0 geser kiri
        }
        if ((i + 1) % Ver == 1) {
            move[1] = false; //move 1 geser kanan
        }
        if ((i + Ver) > index) {
            move[2] = false; //move 2 geser atas
        }
    }
}
```

		<pre>         if ((i - Ver) &lt; 0) {             move[3] = false; //move 3 geser bawah         }         return move;     }      public int cekAr(int i) {         boolean[] move = cekGerak(i);         int[] next = {1, -1, kolom, (-1 * kolom)};         try {             for (int j = 0; j &lt; move.length; j++) {                 if (move[j]) {                     if ("kotor".equalsIgnoreCase(lingkungan[i + next[j]][2])) {                         return next[j];                     }                 }             }             return 0;         } catch (Exception e) {             return 0;         }     }      public void geser(int i, int next) {         if (next == 0) {          } else {             if (next == kolom) {                 System.out.println("geser atas");             } else if (next == (-1 * kolom)) {                 System.out.println("geser bawah");             } else if (next == 1) {                 System.out.println("geser kiri");             } else {                 System.out.println("geser kanan");             }             next += i;             vacumm(next, lingkungan[next][2]);         }     }      public void bersihkan(int i) {         vacumm(i, lingkungan[i][2]);         cek();     }      public void bersihkan() {         bersihkan(0);         System.out.println("-----         -----");         System.out.println("SEMUA SUDAH BERSIH         KECUALI SAMPAH YANG TIDAK BISA DIBERSIHKAN");         System.out.println("-----         -----");     } </pre>	
--	--	---	--

```

public void cek() {
    System.out.println("-----
    -----");
    System.out.println("Cek Dari Awal");
    System.out.println("-----
    -----");
    for (int i = 0; i < lingkungan.length; i++)
    {
        if (!"bersih".equals(lingkungan[i][2]))
        {
            bersihkan(i);
            break;
        } else
        if ("bersih".equals(lingkungan[i][2])) {
            System.out.println(lingkungan[i][1]+ " sudah
            bersih");
        }
    }
}

public static void main(String[] args) {
    AgenCerdasEx a = new AgenCerdasEx(6);
    a.setKondisi(2, "Lokasi 2", "tengah-bawah",
    "bersih");
    a.setKondisi(3, "Lokasi 3", "kiri-bawah",
    "kotor");
    a.setKondisi(5, "Lokasi 5", "tengah-atas",
    "bersih");
    a.setKondisi(6, "Lokasi 6", "kiri-atas",
    "kotor");
    a.setKondisi(1, "Lokasi 1", "kanan-bawah",
    "kotor");
    a.setKondisi(4, "Lokasi 4", "kanan-atas",
    "kotor");
    a.bersihkan();
}
}

```

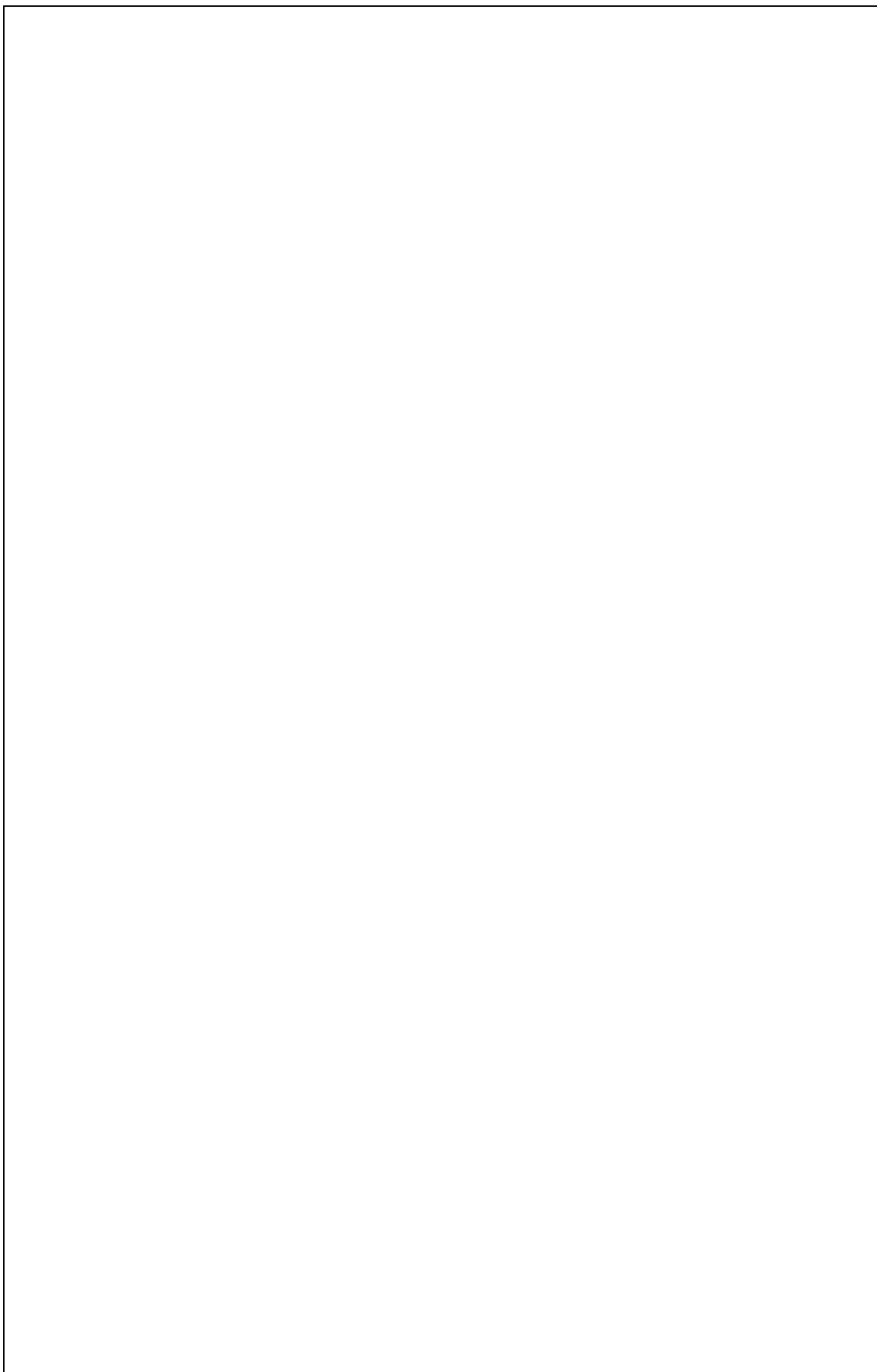
#### Penjelasan

- Deklarasi variable row dan kolom untuk membantu pengarahan pergerakan cleaner
- Constructor AgenCerdasEx untuk menginisialisasi nilai yang dibutuhkan
- Contructor kedua untuk cleaner dengan area yang memiliki kolom selain 3
- Deklarasi method setKondisi untuk mengeset kondisi Area
- Deklarasi method vacumm digunakan untuk membersihkan area yang kotor dan menggerakkan cleaner ke area berikutnya dengan bantuan method cekAr untuk menentukan kemana cleaner akan bergeser dan method geser untuk menggeser cleaner
- Deklarasi method array boolean cekGerak digunakan untuk mengetahui apakah cleaner bisa bergerak keatas, bawah, kiri maupun kanan, dengan



prioritas gerak kiri, kanan, atas, dan bawah berturut-turut. Untuk bergerak ke kiri atau kanan, dicek apakah posisi sekarang +1 jika di modulo dengan kolom memiliki hasil 1 (tidak bisa ke kanan) atau 0 (tidak bisa ke kiri). Untuk bergerak ke atas atau bawah dicek jika posisi sekarang ditambah kolom apakah melebihi panjang indeks atau kurang dari 0.

- Deklarasi method cekAr yang berguna untuk menentukan gerakan cleaner. Proses penentuan pergerakan diawali dengan menjalankan method cekGerak, lalu mengecek status dari area yang bisa dikunjungi, jika kotor maka cleaner akan bergerak ke arah tertentu dengan mengembalikan nilai untuk dijumlahkan dengan posisi sekarang, nilai tersebut: 1 (kiri,) -1 (kanan), kolom (atas) -kolom (bawah). Jika bersih maka nilai = 0 (tidak bergerak)
  - Deklarasi method geser untuk menggeser cleaner sesuai nilai next diparameter. Cara menggesernya adalah dengan menambahkan posisi sekarang dengan nilai next lalu memberi keterangan bergeser kemana, jika next 0, method tidak melakukan apapun.
  - Deklarasi method bersihkan digunakan untuk mulai pembersihan dari posisi i dengan method vacumm. Setelah proses vacumm berhenti dilakukan pengecekan ulang dengan method cek.
  - Method bersihkan tanpa parameter adalah method bersihkan yang dipanggil di main method dan memberi status kalau pembersihan selesai
  - Method cek digunakan untuk mengecek ulang apakah area sudah bersih. Jika belum mulai bersihkan dari area tersebut, jika sudah cetak keterangan kalau area tersebut sudah bersih.
- Main method:
- Menjalankan prosedur cleaner dengan mengeset jumlah area dan kondisi masing-masing area lalu membersihkannya dengan method bersihkan.



### C. KESIMPULAN

1. Apa yang dimaksud Agen Cerdas?

Agen adalah perangkat yang dapat menilai lingkungannya melalui sensor-sensor yang dimilikinya serta bertindak sesuai penilaiannya itu menggunakan aktuator-aktuatornya. Agen cerdas adalah agen yang dapat bertindak seolah-olah dirinya adalah manusia sehingga dalam bekerja ia akan menggunakan rasionalitas-rasionalitas yang mirip dengan manusia.

2. Apa saja yang termasuk agen cerdas? Jelaskan dan beri contoh.

a. Agen simple-reflex

Agen jenis ini adalah agen yang hanya melakukan aksi dari hasil penilaian lingkungan terkini dan tidak melihat hasil penilaiannya yang lalu. Contoh agen jenis ini adalah agen penerima tingkat kepuasan pasien psikiatri dan sistem pendeteksi logam di bandara.

b. Agen model-based

Agen jenis ini adalah pengembangan agen simple-reflex. Bila agen simple-reflex menilai hanya dari kondisi terkini, agen model-based menilai dari kondisi-kondisi lampau (percept history) dalam melakukan aksinya. Contoh agen ini adalah sistem pemanas ruangan pada negara-negara empat musim dan sistem peramal hujan.

c. Agen goal-based

Agen jenis ini adalah pengembangan agen model-based. Agen ini tidak hanya menyimpan kondisi-kondisi lampau saja, tetapi agen juga mengetahui tujuan yang hendak ia capai sehingga ia bisa beradaptasi dalam suatu kondisi tertentu menggunakan pengetahuan yang dimiliki untuk mencapai tujuan (goal state)-nya. Contoh agen ini adalah agen pengatur lalu lintas jalan raya dan agen pengatur lalu lintas bandara.

d. Agen utility-based

Agen jenis ini adalah pengembangan agen goal-based. Selain mengetahui tujuan yang hendak ia capai, agen juga dapat mengukur tingkat “kepuasan”-nya terhadap aksi yang ia telah lakukan sehingga ia benar-benar dapat memilih aksi-aksi yang cocok untuk kebbaikannya sendiri. Contoh agen ini adalah sistem pemilih rute jalan terbaik yang diimplementasikan pada robot yang bisa berjalan dan sistem pemilih tempat pendaratan yang diimplementasikan pada drone yang terbang tanpa bantuan manusia (autoflight drone).

e. Agen belajar

Agen jenis ini adalah agen yang memiliki rasionalitas yang dapat menyaingi manusia. Agen ini dibuat untuk mempelajari pola-pola atau perbuatan-perbuatan tertentu agar ia cakap dalam melakukan pekerjaan yang menjadi tujuannya diciptakan. Agen ini memiliki empat komponen penting: learning element (komponen penting untuk belajar berdasarkan pengkritik yang biasanya adalah manusia), performance element (komponen penting untuk

menyeksi aksi-aksi eksternal yang hendak dilakukannya), critic (penilai kinerja sistem), dan problem generator (komponen untuk memberi aksi-aksi yang dapat menghasilkan pengalaman-pengalaman baru dan informatif). Contoh agen ini adalah Google Assistant dan sistem olah vokal yang diimplementasikan pada robot yang dapat berbicara.