

INTELLIGENT AGENTS

Dosen Pengampu : Riki Rulli A Siregar

Kode Mata Kuliah : C31050204

Tugas Ini Disusun Untuk Memenuhi Tugas Mata Kuliah Kecerdasaan Buatan



Disusun Oleh :

1. Fitto Marcelindo (202131001)

**Fakultas Telematika Energi
Program Studi Teknik Informatika
Institut Teknologi PLN
2022/2023**

INTELLIGENT AGENTS

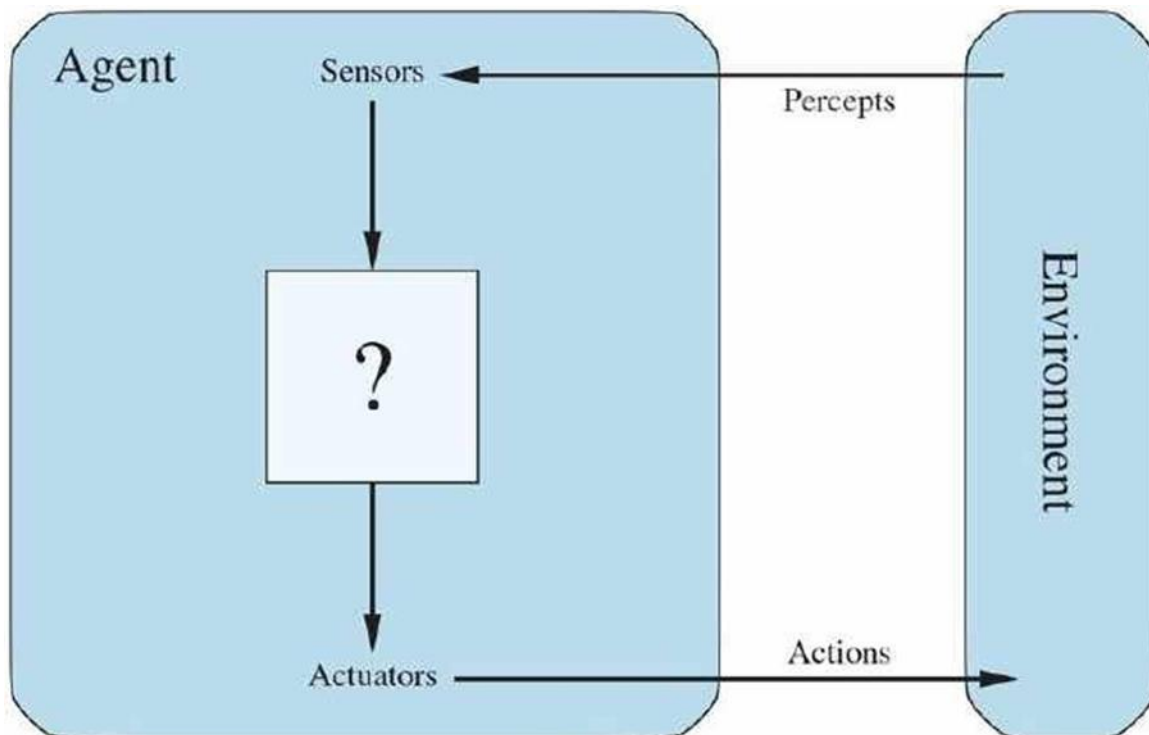
Dalam konsep *Intelligent Agents* akan membahas sifat agen, sempurna atau tidak, keragaman lingkungan dan jenis jenis agen yang akan dihasilkan. Konsep agen rasional sebagai pusat pendekatan kecerdasan buatan konsep rasionalitas diterapkan pada berbagai jenis agen yang beroperasi dilingkungan yang dapat dibayangkan.

Hal pertama yang akan dibahas adalah memeriksa age, lingkungan, dan hubungan di antara keduanya pengamtan bahwa beberapa agen berperilaku lebih baik daripada yang kaun membaca pada gagasan agen yang rasional, selanjutnya kita akan desain agen, kerangka dasar agen.

Secara keseluruhan dalam pembahasan kali ini tentang konsep agen rasional, lingkungan , sifat lingkungan yang mempengaruhi prilaku agen dan beberapa jenis desain kerangka dasar agen yang relevan.

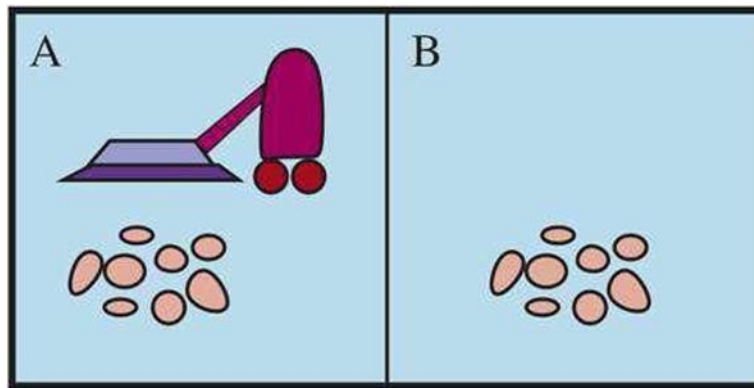
1. Agents and Enivornments

Agen adalah segala sesuatu yang dapat dilihat sebagai presepsi lingkungan melalui sensor dan bertindak atas lingkungan tersebut melalui akuator.



Pada gambar diatas menjelaskan bahwasanya agen akan dipengaruhi oleh lingkungannya dan agen akan merespon inputan tersebut dengan keluaran berdasarkan lingkungan tersebut. Akutor sebagai alat mekanis yang memberikan ouput atau actions yang telah di respon oleh agen melaluu penyulingan atau filtrasi dari sensor.

Agen cerdas pasti memiliki sensor dan motor untuk aktuator karena agen adalah perangkat lunak yang menerima content, file, paket jaringan dan input manusia



Gambar diatas mengilustrasikan Vacuum Cleaner yang sedang bekerja, gambar diatas terdapat 2 kotak yang di wakili oleh variabel name A dan B, dan Vacuum tersebut mengecek setiap square atau kotak tersebut terdapat kotoran atau tidak. Apabila didalam kotak tersebut terdapat kotoran maka Vacuum tersebut akan membersihkan, dan apabila tidak terdapat kotoran maka vacuum tersebut akan move atau bergerak ke kotak lain yang terdapat kotoran.

2. Good Behavior The Concept of Rationality

Agen rasional adalah agen yang melakukan hal yang benar. Dalam hal ini melakukan hal yang benar adalah suatu jawaban yang relevan berdasarkan apa yang diminta oleh user.

Contoh apabila kita meminta jawaban pada chatgpt “berikan contoh perilaku baik manusia ketiak sedang berbicara kepada orang yang lebih tua?”
=> jawaban yang diberikan Chatgpt pasti berdasarkan sumber di internet yang ada misal, Wikipedia, Jurnal-jurnal ilmiah dan sebagainya.

Ada 4 tahapan rasional pada waktu tertentu yaitu sebagai berikut

- Ukuran kinerja yang mendefinisikan
- Kriteria keberhasilan pengetahuan agen sebelumnya tentang lingkungan
- Tindakan yang dapat dilakukan oleh agen
- Urutan persepsi agen hingga saat ini

3. The Nature of Environments

Sifat lingkungan memainkan peran pada agen beroperasi, dan sangat memengaruhi cara agen memandang, berinteraksi, dan merespons lingkungannya penting dalam membentuk perilaku dan kemampuan agen cerdas. Lingkungan mencakup faktor dan kondisi eksternal tempat

PEAS adalah singkatan dari Performance measure (ukuran kinerja), Environment (lingkungan), Actuators (aktuator), dan Sensors (sensor). Ini adalah konsep yang digunakan untuk mengidentifikasi dan menggambarkan sifat-sifat agen dalam suatu lingkungan. Berikut adalah contoh deskripsi PEAS dalam lingkungan sehari-hari beserta tabelnya:

Contoh: Asisten Pribadi Virtual (seperti Siri atau Google Assistant)

1. Performance Measure (Ukuran Kinerja):

- Kecepatan dan keakuratan dalam memberikan informasi yang relevan
- Kemampuan untuk memahami dan menafsirkan perintah pengguna
- Kualitas dalam menjalankan tugas-tugas yang diberikan (misalnya, mengatur alarm, mengirim pesan)

2. Environment (Lingkungan):

- Lingkungan audio-visual yang mencakup suara, teks, dan gambar
- Interaksi dengan pengguna melalui perangkat seperti smartphone atau speaker pintar
- Akses ke koneksi internet untuk mencari informasi

3. Actuators (Aktuator):

- Speaker untuk memberikan respon suara kepada pengguna
- Teks atau pesan yang ditampilkan di layar perangkat
- Akses ke perangkat lain, seperti pengaturan alarm atau mengirim pesan

4. Sensors (Sensor):

- Mikrofon untuk mendeteksi suara pengguna
- Pengenalan teks untuk memahami pesan teks
- Akses ke koneksi internet untuk mencari informasi

Tabel PEAS untuk Asisten Pribadi Virtual:

PEAS	Deskripsi
Performance Measure	Kecepatan, keakuratan, dan kualitas dalam memberikan informasi yang relevan, serta kemampuan untuk menjalankan tugas-tugas yang diberikan oleh pengguna.
Environment	Lingkungan audio-visual dengan interaksi melalui perangkat seperti smartphone atau speaker pintar, dan akses ke koneksi internet.
Actuators	Speaker, tampilan teks, dan akses ke perangkat lain seperti pengaturan alarm atau pengiriman pesan.
Sensors	Mikrofon untuk mendeteksi suara pengguna, pengenalan teks untuk memahami

Lingkungan dapat sangat bervariasi dalam karakteristiknya, dan karakteristik ini memiliki dampak langsung pada desain dan fungsi agen cerdas. Berikut adalah beberapa aspek utama dari sifat lingkungan yang memengaruhi agen cerdas:

1. Dapat Diamati Sepenuhnya vs Dapat Diamati Sebagian: Dalam lingkungan yang dapat diamati sepenuhnya, agen memiliki akses ke informasi yang lengkap dan akurat tentang kondisi lingkungan saat ini. Sebaliknya, lingkungan yang dapat diamati sebagian membatasi pengetahuan agen, mengharuskannya untuk membuat asumsi atau menggunakan metode probabilistik untuk menyimpulkan keadaan. Tingkat keteramatan mempengaruhi proses pengambilan keputusan agen dan strategi yang digunakannya.
2. Deterministik vs Stokastik: Lingkungan deterministik memiliki hasil yang terdefinisi dengan baik dan dapat diprediksi untuk setiap tindakan yang diambil oleh agen. Di sisi lain, lingkungan stokastik memperkenalkan keacakan, menghasilkan hasil yang tidak pasti bahkan untuk tindakan yang sama. Agen di lingkungan stokastik perlu menangani ketidakpastian dan menyesuaikan perilaku mereka.
3. Episodik vs Sekuensial: Dalam lingkungan episodik, setiap episode atau tugas tidak bergantung pada episode atau tugas lainnya, dan tindakan agen hanya memengaruhi hasil langsung. Dalam lingkungan berurutan, tindakan memiliki konsekuensi yang terbawa ke interaksi berikutnya, dan agen harus mempertimbangkan perencanaan jangka panjang dan pengambilan keputusan strategis.
4. Statis vs Dinamis: Lingkungan statis tetap tidak berubah ketika agen mempertimbangkan dan membuat keputusan. Sebaliknya, lingkungan yang dinamis mengalami perubahan bahkan

ketika agen tidak bertindak, mengharuskan agen untuk responsif dan adaptif terhadap kondisi yang berkembang.

5. Diskrit vs Kontinu: Sifat tindakan agen dan keadaan lingkungan dapat bersifat diskrit atau kontinu. Lingkungan diskrit memiliki sejumlah keadaan dan tindakan yang berbeda, sedangkan lingkungan kontinu memiliki rentang keadaan atau tindakan yang tak terbatas. Agen cerdas perlu dirancang sesuai untuk menangani kompleksitas dan perincian lingkungan.

6. Agen Tunggal vs Multi-Agen: Sebuah lingkungan dapat melibatkan satu agen cerdas atau beberapa agen yang beroperasi dan berinteraksi secara bersamaan. Lingkungan multi-agen memberikan tantangan tambahan seperti koordinasi, negosiasi, dan persaingan di antara para agen.

Memahami sifat lingkungan sangat penting untuk merancang agen cerdas yang dapat menavigasi dan berinteraksi secara efektif dengan lingkungannya. Dengan mempertimbangkan observabilitas, determinisme, sifat episodik atau sekuensial, dinamisme, diskretitas atau kontinuitas, dan keberadaan agen lain, perancang dapat mengembangkan algoritme, strategi, dan arsitektur yang tepat untuk memungkinkan agen membuat keputusan rasional dan menunjukkan perilaku cerdas di lingkungannya.

4. Program Agents

Pada program agents merujuk pada program computer yang dirancang untuk mengimplementasikan kecerdasan buatan dan bertindak sebagai agen cerdas. Pada program ini mencakup algoritma, pemrosesan data, dan pemodelan yang memungkinkan agen cerdas dapat menganalisis dan berinteraksi terhadap lingkungan.

Ada 4 jenis program agen dasar yang mewujudkan prinsip-prinsip yang mendasari hampir semua sistem cerdas :

- Agen Refleks Sederhana
- Agen Refleks berbasis model
- Agen berbasis tujuan
- Agen berbasis utilitas

A. Agen Refleks Sederhana

Jenis agen yang paling sederhana yang bertindak secara refleksif berdasarkan stimulus yang diterima dari lingkungan agen ini tidak memiliki pemahaman mendalam tentang lingkungan atau kemampuan untuk merencanakan tindakan jangka Panjang.

Contoh :

```
def SimpleReflexAgent(percept):
```

```
# Input: percept (current percept or sensor input)
# Output: action (action to be taken)

# Rules:
# Rule 1: IF percept = "dirty" THEN action = "suck"
# Rule 2: IF percept = "clean" THEN action = "move"
```

```
# Initialize rules
```

```
action = None
```

```
if percept == "dirty":
    action = "suck"
elif percept == "clean":
    action = "move"
```

```
return action
```

```
# Contoh penggunaan:
```

```
percept = "dirty"
action = SimpleReflexAgent(percept)
print(action) # Output: "suck"
```

```
//bahasa python
```

```
//hasil program
```

```
def SimpleReflexAgent(percept):
    # Input: percept (current percept or sensor input)
    # Output: action (action to be taken)

    # Rules:
    # Rule 1: IF percept = "dirty" THEN action = "suck"
    # Rule 2: IF percept = "clean" THEN action = "move"

    # Initialize rules
    action = None

    if percept == "dirty":
        action = "suck"
    elif percept == "clean":
        action = "move"

    return action

# Contoh penggunaan:
percept = "dirty"
action = SimpleReflexAgent(percept)
print(action) # Output: "suck"
```

[1] ✓ 0.0s

... suck

B. Agen refleks berbasis model

Agen ini bisa dibilang cara yang paling efektif untuk menangani keteramatan parsial adalah agen harus melacak bagian dunia yang tidak dapat dilihatnya saat ini artinya agen harus mempertahankan semacam keadaan internal yang bergantung pada riwayat persepsi.

Contoh source code :

```
class ModelBasedReflexAgent:
    def __init__(self):
        self.current_state = None
        self.model = {}

    def update_model(self, percept):
        # Update the model based on the current percept
        if(percept == "dirty"):
            self.model[percept] = True
        else:
            self.model[percept] = False

    def update_state(self, percept):
        # Update the current state based on the percept and the model
        if percept in self.model:
            self.current_state = self.model[percept]
```



```

def select_action(self):
    # Select the action based on the current state
    if self.current_state == True:
        action = "clean"
    else:
        action = "move"

    return action

def model_based_reflex_agent(percept):
    agent = ModelBasedReflexAgent()
    agent.update_model(percept)
    agent.update_state(percept)
    action = agent.select_action()

    return action

# Contoh penggunaan:
percept = input("")
action = model_based_reflex_agent(percept.lower())
print(action)

```

//Hasil ouput

```

D:\SEMESTER 4 > KECERDASAN BUATAN > agentberbasimodel.py > ...
1 class ModelBasedReflexAgent:
2     def __init__(self):
3         self.current_state = None
4         self.model = {}
5
6     def update_model(self, percept):
7         # Update the model based on the current percept
8         if(percept == "dirty"):
9             self.model[percept] = True
10        else:
11            self.model[percept] = False

```

PS C:\Users\ACER> & C:\Users\ACER\AppData\Local\Programs\Python\Python311\python.exe "d:/SEMESTER 4/KECERDASAN BUATAN/agentberbasimodel.py"

dirty
clean
PS C:\Users\ACER>

C. Agen Berbasis Tujuan

Goals based agent adalah jenis agen kecerdasan buatan yang bertindak berdasarkan tujuan yang telah ditetapkan sebelumnya tujuan mereka keadaan atau kondisi yang diinginkan atau ingin dicapai oleh agen

Cara kerja agen berbasis tujuan umumnya melibatkan langkah-langkah berikut:

1. Penetapan Tujuan: Agen menetapkan tujuan yang ingin dicapai berdasarkan pengetahuan atau kebutuhan yang telah ditentukan sebelumnya. Tujuan ini bisa bersifat jangka pendek maupun jangka panjang.
2. Pemahaman Lingkungan: Agen mengumpulkan informasi tentang lingkungan melalui sensor atau percepts. Informasi ini digunakan untuk memahami keadaan lingkungan saat ini.

3. Analisis Keadaan: Agen menganalisis keadaan lingkungan saat ini dan membandingkannya dengan tujuan yang telah ditetapkan. Agen mengevaluasi kesenjangan antara keadaan saat ini dan tujuan yang diinginkan.

4. Perencanaan: Agen menggunakan pengetahuan dan pemahaman tentang lingkungan serta tujuan yang ingin dicapai untuk merencanakan serangkaian tindakan yang harus diambil untuk mencapai tujuan tersebut. Perencanaan ini melibatkan pemilihan dan urutan tindakan yang dianggap paling efektif.

5. Tindakan: Agen mengambil tindakan berdasarkan perencanaan yang telah dibuat. Tindakan ini dapat berupa interaksi fisik dengan lingkungan atau komunikasi dengan entitas lain.

6. Evaluasi dan Koreksi: Agen terus memantau lingkungan dan mengevaluasi apakah tindakan yang diambil membawa agen lebih dekat atau menjauh dari tujuan. Jika diperlukan, agen melakukan koreksi atau memperbarui perencanaan dan tindakan berikutnya.

Contoh Study Case :

Algorithm GoalBasedAgent:

Input: percept (current percept or sensor input)

Output: action (action to be taken)

State: current_state (current state of the environment)

Goals: goals_list (list of goals to be achieved)

Procedure UpdateState(percept):

Update the current state based on the percept

current_state = percept

Procedure SelectAction(current_state, goals_list):

Select the action based on the current state and goals

if current_state == goals_list[0]:

 action = "achieve_goal"

else:

 action = "explore"

return action

```
UpdateState(percept)
```

```
action = SelectAction(current_state, goals_list)
```

```
RETURN action
```

Berikut ini adalah contoh pseudocode program untuk sistem agen berbasis tujuan:

```
'''
```

Algorithm GoalBasedAgent:

Input: percept (current percept or sensor input)

Output: action (action to be taken)

State: current_state (current state of the environment)

Goals: goals_list (list of goals to be achieved)

Procedure UpdateState(percept):

```
# Update the current state based on the percept
```

```
current_state = percept
```

Procedure SelectAction(current_state, goals_list):

```
# Select the action based on the current state and goals
```

```
if current_state == goals_list[0]:
```

```
    action = "achieve_goal"
```

```
else:
```

```
    action = "explore"
```

```
return action
```

```
UpdateState(percept)
```

```
action = SelectAction(current_state, goals_list)
```

RETURN action

'''

Dalam pseudocode, memiliki algoritma untuk sistem agen berbasis tujuan. Agen menerima input percept, yang merepresentasikan persepsi atau masukan sensor saat ini. Agen juga memiliki state (current_state), yang merepresentasikan keadaan lingkungan saat ini, dan goals_list, yang merupakan daftar tujuan yang ingin dicapai.

Algoritma tersebut terdiri dari dua prosedur utama. Pertama, prosedur UpdateState digunakan untuk memperbarui state agen berdasarkan percept saat ini. Kemudian, prosedur SelectAction digunakan untuk memilih tindakan berdasarkan current_state dan goals_list. Dalam contoh ini, jika current_state sama dengan tujuan pertama dalam goals_list, agen akan memilih tindakan "achieve_goal" (mencapai tujuan). Jika tidak, agen akan memilih tindakan "explore" (menjelajah atau mencari).

Setelah memperbarui state dan memilih tindakan, agen mengembalikan tindakan yang dipilih dengan menggunakan pernyataan RETURN.

D. Agents Berbasis Utilitas

Agen berbasis utilitas (utility-based agent) adalah jenis agen kecerdasan buatan yang bertindak berdasarkan nilai utilitas atau preferensi terhadap berbagai keadaan atau hasil yang mungkin terjadi. Agen ini mempertimbangkan manfaat atau keuntungan yang diharapkan dari setiap tindakan yang dapat diambil, dan memilih tindakan yang dianggap memiliki nilai utilitas tertinggi berdasarkan preferensi atau tujuan yang telah ditentukan sebelumnya.

Cara kerja agen berbasis utilitas umumnya melibatkan langkah-langkah berikut:

1. Penetapan Preferensi: Agen menetapkan preferensi atau skala nilai utilitas untuk berbagai keadaan atau hasil yang mungkin terjadi. Preferensi ini dapat berupa bobot numerik atau peringkat relatif yang menunjukkan tingkat keinginan atau manfaat yang dihubungkan dengan setiap keadaan atau hasil.
2. Pemahaman Lingkungan: Agen mengumpulkan informasi tentang lingkungan melalui sensor atau percepts. Informasi ini digunakan untuk memahami keadaan lingkungan saat ini.
3. Evaluasi Tindakan: Agen mengevaluasi manfaat atau utilitas yang diharapkan dari setiap tindakan yang dapat diambil berdasarkan preferensi yang telah ditetapkan. Hal ini melibatkan memperkirakan konsekuensi dari tindakan yang berpotensi terjadi.
4. Pemilihan Tindakan: Agen memilih tindakan yang dianggap memiliki nilai utilitas tertinggi berdasarkan evaluasi tindakan sebelumnya. Tindakan ini dianggap sebagai yang paling menguntungkan atau menghasilkan manfaat yang paling besar sesuai dengan preferensi yang ditetapkan.
5. Tindakan: Agen mengambil tindakan yang telah dipilih berdasarkan penilaian utilitas. Tindakan ini dapat berupa interaksi fisik dengan lingkungan atau komunikasi dengan entitas lain.

6. Evaluasi dan Koreksi: Agen terus memantau lingkungan dan mengevaluasi apakah tindakan yang diambil memberikan hasil yang sesuai dengan nilai utilitas yang diharapkan. Jika diperlukan, agen melakukan koreksi atau memperbarui preferensi dan perencanaan tindakan berikutnya.

Fungsi Utilitas agen pada dasarnya adalah internalisasi dari ukuran kinerja. Agen berbasis utilitas rasional memilih Tindakan yang memaksimalkan utilitas yang diharapkan dari hasil Tindakan tersebut yaitu utilitas yang diharapkan agen secara rata”.

// Contoh : study kasus

Algorithm UtilityBasedAgent:

Input: percept (current percept or sensor input)

Output: action (action to be taken)

State: current_state (current state of the environment)

Actions: possible_actions (list of possible actions)

Utility: utility_function (function to calculate the utility of actions)

Procedure UpdateState(percept):

Update the current state based on the percept

current_state = percept

Procedure SelectAction(current_state, possible_actions, utility_function):

Select the action with the highest utility based on the current state

best_action = None

highest_utility = -Infinity

for action in possible_actions:

utility = utility_function(action, current_state)

if utility > highest_utility:

highest_utility = utility

best_action = action

return best_action

```
UpdateState(percept)
```

```
action = SelectAction(current_state, possible_actions, utility_function)
```

```
RETURN action
```

Berikut adalah contoh pseudocode program untuk sistem agen berbasis utilitas:

```
'''
```

Algorithm UtilityBasedAgent:

Input: percept (current percept or sensor input)

Output: action (action to be taken)

State: current_state (current state of the environment)

Actions: possible_actions (list of possible actions)

Utility: utility_function (function to calculate the utility of actions)

Procedure UpdateState(percept):

```
# Update the current state based on the percept
```

```
current_state = percept
```

Procedure SelectAction(current_state, possible_actions, utility_function):

```
# Select the action with the highest utility based on the current state
```

```
best_action = None
```

```
highest_utility = -Infinity
```

```
for action in possible_actions:
```

```
    utility = utility_function(action, current_state)
```

```
    if utility > highest_utility:
```

```
        highest_utility = utility
```

```

        best_action = action

    return best_action

def UpdateState(percept):
    action = SelectAction(current_state, possible_actions, utility_function)

    RETURN action
'''

```

Dalam pseudocode di atas, kita memiliki algoritma untuk sistem agen berbasis utilitas. Agen menerima input percept, yang merepresentasikan persepsi atau masukan sensor saat ini. Agen juga memiliki state (`current_state`), yang merepresentasikan keadaan lingkungan saat ini, actions (`possible_actions`), yang merupakan daftar tindakan yang mungkin dilakukan, dan `utility_function`, yang merupakan fungsi untuk menghitung utilitas dari setiap tindakan.

Algoritma tersebut terdiri dari dua prosedur utama. Pertama, prosedur `UpdateState` digunakan untuk memperbarui state agen berdasarkan percept saat ini. Kemudian, prosedur `SelectAction` digunakan untuk memilih tindakan dengan utilitas tertinggi berdasarkan `current_state`, `possible_actions`, dan `utility_function`. Dalam contoh ini, agen melakukan iterasi melalui setiap tindakan yang mungkin, menghitung utilitas untuk masing-masing, dan memilih tindakan dengan utilitas tertinggi.

Setelah memperbarui state dan memilih tindakan, agen mengembalikan tindakan terbaik dengan menggunakan pernyataan `RETURN`.

Pseudocode dapat diimplementasikan dalam bahasa pemrograman seperti Python dengan menambahkan detail implementasi yang sesuai, seperti definisi fungsi `utility_function`, inisialisasi variabel, pemanggilan fungsi, dan perulangan jika diperlukan.

KESIMPULAN

- Agen adalah sesuatu yang merasakan dan bertindak dalam suatu lingkungan,
- Fungsi agen untuk sebuah agen menentukan Tindakan yang diambil oleh agen sebagai respons terhadap urutan persepsi.
- Agen yang rasional bertindak untuk memaksimalkan nilai yang diharapkan dari ukuran kinerja, mengingat urutan persepsi yang telah dilihatnya sejauh ini.
- Lingkungan tugas mencakup ukuran kinerja, lingkungan eksternal, aktuator dan sensor. Dalam mendesain agen.

Terdapat beberapa jenis agen kecerdasan buatan, antara lain:

- - Agen refleksi sederhana, yang berperilaku berdasarkan kondisi saat ini tanpa mempertimbangkan keadaan masa depan.
- - Agen berbasis model, yang membangun model lingkungan dan menggunakannya untuk mengambil keputusan.
- - Agen berbasis tujuan, yang bertindak berdasarkan tujuan atau preferensi yang ditetapkan sebelumnya.
- - Agen berbasis utilitas, yang mempertimbangkan nilai utilitas atau manfaat yang dihubungkan dengan tindakan yang mungkin diambil.

- Program agen dapat diimplementasikan dalam bahasa pemrograman seperti Python untuk menggambarkan logika atau aturan yang diikuti oleh agen.

Harapan Terhadap Pembuatan makalah dimasa yang akan datang

Dengan demikian, harapan kedepannya terhadap pembuatan makalah yang akan datang adalah untuk menggali lebih dalam tentang konsep dan prinsip-prinsip desain agen kecerdasan buatan. Makalah tersebut dapat mengeksplorasi berbagai jenis agen, menggambarkan lebih rinci tentang lingkungan dan sifat-sifatnya, serta memberikan contoh-contoh kasus yang lebih kompleks dan implementasi yang lebih lengkap. Selain itu, makalah tersebut dapat mengeksplorasi kemungkinan pengembangan lebih lanjut dalam bidang kecerdasan buatan, seperti integrasi dengan teknologi baru, perumusan metode evaluasi yang lebih baik, atau penerapan pada domain atau industri tertentu.