Bab 3: Pernyataan Berkondisi dalam Pemrograman Artificial Intelligence

3.1. Pendahuluan

Pernyataan berkondisi adalah salah satu konsep dasar dalam pemrograman yang memungkinkan program untuk membuat keputusan berdasarkan kondisi tertentu. Konsep ini sangat penting dalam **Artificial Intelligence (AI)** dan **Machine Learning (ML)**, karena sebagian besar algoritma memerlukan kemampuan untuk melakukan pengambilan keputusan berdasarkan input atau data yang diberikan.

Pernyataan berkondisi digunakan untuk menentukan apakah sebuah blok kode dieksekusi atau tidak, berdasarkan kondisi yang diberikan. Dengan memahami cara kerja **pernyataan berkondisi**, kita dapat mengontrol alur eksekusi program, membuat model yang lebih cerdas, dan mengembangkan algoritma yang dapat merespons berbagai situasi.

Bab ini akan menjelaskan konsep **pernyataan berkondisi**, struktur kontrol alur yang digunakan dalam pemrograman, serta penerapannya dalam **pemrograman Artificial Intelligence (AI)**. Pembahasan ini mencakup penggunaan **if**, **else**, **elif**, serta **ternary operators** untuk menangani kondisi dalam program. Selain itu, kita juga akan melihat contoh penerapan pernyataan berkondisi dalam pengembangan model AI.

3.2. Pernyataan Berkondisi (Conditional Statements)

Pernyataan berkondisi adalah pernyataan yang memungkinkan kita untuk mengeksekusi blok kode tertentu jika kondisi yang diberikan terpenuhi, dan blok kode lain jika kondisi tersebut tidak terpenuhi. Pernyataan ini sangat berguna ketika kita ingin membuat keputusan berdasarkan input atau situasi tertentu dalam program.

3.2.1. Struktur Pernyataan Berkondisi (If Statement)

Pernyataan **if** adalah bentuk paling dasar dari pernyataan berkondisi. Dengan **if**, kita dapat memeriksa apakah suatu kondisi benar atau salah. Jika kondisi tersebut benar, maka kode dalam blok **if** akan dieksekusi. Jika kondisi tersebut salah, maka blok kode tersebut akan dilewati.

```
x = 10


if x > 5:
   print("x lebih besar dari 5")
```

Penjelasan:

- Program akan memeriksa apakah **x > 5**.
- Karena **x = 10**, yang lebih besar dari 5, maka pernyataan dalam blok **if** akan dijalankan dan outputnya adalah x lebih besar dari 5.

3.2.2. Pernyataan Else

Kadang-kadang kita ingin menentukan aksi alternatif jika kondisi dalam pernyataan **if** tidak terpenuhi. Untuk itu, kita menggunakan **else**. **else** digunakan untuk mengeksekusi kode ketika kondisi dalam **if** salah.

```
x = 3


if x > 5:
  print("x lebih besar dari 5")
else:
  print("x lebih kecil atau sama dengan 5")
```

Penjelasan:

- Jika **x = 3**, kondisi **x > 5** salah, maka blok **else** akan dieksekusi, dan outputnya adalah x lebih kecil atau sama dengan 5.

3.2.3. Pernyataan Elif (Else If)

Seringkali kita memiliki lebih dari dua kondisi yang perlu diperiksa. Dalam kasus ini, kita bisa menggunakan **elif** untuk memeriksa kondisi tambahan setelah kondisi **if** pertama. Pernyataan **elif** memungkinkan kita untuk memeriksa beberapa kondisi secara berurutan.

```
x = 10


if x < 5:
  print("x lebih kecil dari 5")
elif x == 10:
  print("x sama dengan 10")
else:
  print("x lebih besar dari 5 tetapi tidak sama dengan 10")
```

Penjelasan:

- Program pertama memeriksa apakah **x < 5**, yang salah.
- Kemudian, **elif x == 10** diperiksa dan benar, sehingga outputnya adalah x sama dengan 10.

3.2.4. Nested If Statements (Pernyataan If Bertingkat)

Pernyataan **if** juga dapat digunakan dalam blok **if** lainnya, yang disebut **nested if**. Ini memungkinkan kita untuk membuat keputusan yang lebih kompleks.

```
x = 8

y = 12


if x > 5:
  if y > 10:
    print("x lebih besar dari 5 dan y lebih besar dari 10")
  else:
    print("x lebih besar dari 5 dan y tidak lebih besar dari 10")
else:
  print("x tidak lebih besar dari 5")
```

Penjelasan:

- Program pertama memeriksa apakah **x > 5**.
- Jika benar, maka program memeriksa apakah **y > 10**. Jika kedua kondisi benar, maka outputnya adalah x lebih besar dari 5 dan y lebih besar dari 10.

3.3. Pernyataan Ternary (Conditional Expression)

Pernyataan ternary (atau conditional expression) adalah bentuk singkat dari **if-else** yang digunakan untuk membuat keputusan sederhana. Ini memungkinkan kita untuk menetapkan nilai berdasarkan kondisi dalam satu baris kode.

```
x = 10

y = "Lebih besar dari 5" if x > 5 else "Tidak lebih besar dari 5"

print(y)
```

Penjelasan:

- Jika **x > 5**, maka nilai **y** akan menjadi "Lebih besar dari 5".
- Jika **x <= 5**, maka nilai **y** akan menjadi "Tidak lebih besar dari 5".
- Pada contoh ini, outputnya adalah Lebih besar dari 5.

3.4. Pernyataan Berkondisi dalam Konteks Artificial Intelligence

Dalam **Artificial Intelligence** dan **Machine Learning**, pernyataan berkondisi sering digunakan dalam berbagai algoritma dan aplikasi, terutama dalam pengambilan keputusan dan pembuatan aturan.

3.4.1. Pernyataan Berkondisi dalam Algoritma Decision Trees

Pada algoritma **decision tree**, pernyataan berkondisi digunakan untuk membagi dataset ke dalam cabang-cabang pohon keputusan. Setiap cabang akan mewakili kondisi yang diperiksa, dan pembagian ini akan terus berlangsung sampai keputusan akhir dibuat.

Contoh kode sederhana dengan decision tree:

```
from sklearn.tree import DecisionTreeClassifier


# Data contoh
X = [[3, 4], [2, 6], [4, 2], [5, 8]]
y = [0, 1, 0, 1]


# Membuat model DecisionTree
model = DecisionTreeClassifier()


# Melatih model
model.fit(X, y)


# Memprediksi kelas untuk data baru
print(model.predict([[4, 5]]))
```

Penjelasan:

- Pada model **decision tree**, setiap langkah dalam pembentukan pohon akan menggunakan pernyataan berkondisi untuk memisahkan data sesuai dengan fitur yang diberikan.

3.4.2. Pernyataan Berkondisi dalam Pengolahan Bahasa Alami (NLP)

Dalam **Natural Language Processing (NLP)**, pernyataan berkondisi digunakan untuk membuat keputusan berdasarkan teks yang diproses. Misalnya, dalam **sentiment analysis**, pernyataan berkondisi digunakan untuk memeriksa kata-kata atau frasa dalam teks yang menunjukkan sentimen positif atau negatif.

```
def sentiment_analysis(text):
  if "baik" in text or "senang" in text:
    return "Positif"
  elif "buruk" in text or "sedih" in text:
    return "Negatif"
  else:
    return "Netral"


print(sentiment_analysis("Saya merasa sangat senang hari ini!"))
```

Penjelasan:

- Dalam contoh ini, pernyataan berkondisi digunakan untuk menentukan apakah teks yang diberikan menunjukkan sentimen positif, negatif, atau netral.

3.5. Pernyataan Berkondisi dalam Pengembangan AI dan Machine Learning

Pernyataan berkondisi digunakan dalam **Artificial Intelligence** untuk menangani **rule-based systems**, **decision trees**, dan bahkan dalam **algoritma machine learning** untuk pengambilan keputusan berdasarkan data atau hasil prediksi.

Contoh penerapannya antara lain:

1. **Decision Making Systems**: Misalnya, dalam robotika, pernyataan berkondisi digunakan untuk menentukan langkah berikutnya berdasarkan lingkungan yang diamati.
2. **Model Machine Learning**: Dalam **classification models**, seperti **Logistic Regression** atau **SVM**, kondisi digunakan untuk menentukan apakah prediksi tersebut benar atau salah, dan untuk memperbarui model.

3.6. Kesimpulan

Pernyataan berkondisi adalah dasar dalam pemrograman yang memungkinkan kita untuk membuat keputusan berdasarkan input atau kondisi yang ada. Dalam **Artificial Intelligence** dan **Machine Learning**, pernyataan berkondisi digunakan secara luas dalam berbagai algoritma dan sistem pengambilan keputusan. Dengan memahami bagaimana menggunakan pernyataan berkondisi, kita dapat membangun program yang lebih cerdas yang dapat merespons berbagai situasi secara otomatis, memungkinkan aplikasi AI untuk bekerja lebih efisien dan efektif dalam menangani data yang lebih kompleks.