

PEMILIHAN

7



Overview

Program dapat merepresentasikan situasi pemilihan yang sering dihadapi dalam dunia nyata. Berdasarkan satu atau beberapa kondisi, dapat ditentukan satu atau sejumlah aksi yang akan dilakukan. Dengan adanya struktur pemilihan, program dapat berjalan dengan jalur yang berbeda, berdasarkan hasil pengecekan kondisi yang dipenuhi.



Tujuan

- Memahami struktur pemilihan dalam program
- Mengenal struktur IF dan CASE yang dapat digunakan dalam pemilihan
- Memahami konsep kondisi dan aksi dalam struktur pemilihan
- Menerapkan pemilihan dalam menyelesaikan berbagai kasus

Dalam kehidupan nyata, seringkali dihadapkan pada beberapa pilihan. Pada saat menghadapi pilihan, satu atau beberapa kondisi menjadi bahan pertimbangan dalam memutuskan untuk melakukan aksi tertentu. Contoh:

Jika cuaca mendung, maka saya membawa payung.

Pada contoh tersebut, ‘cuaca mendung’ merupakan kondisi yang menjadi bahan pertimbangan untuk melakukan aksi ‘saya membawa payung’. Jika kondisi ‘cuaca mendung’ terpenuhi (bernilai benar), maka aksi ‘saya membawa payung’ dilakukan

Sebuah program komputer juga dapat mengenali situasi pemilihan. Pernyataan dalam contoh di atas dapat dituliskan dalam struktur pemilihan sebagai berikut:

```
IF cuaca mendung THEN  
    saya membawa payung  
END IF
```

Untuk selengkapnya penggunaan bentuk pemilihan akan dijelaskan berikut ini.

7.1 Bentuk Umum IF dan Variasinya

7.1.1 Pernyataan if

Pernyataan if dapat dipakai untuk mengambil keputusan berdasarkan suatu kondisi. Bentuk pernyataan ada dua macam :

- ↳ if
- ↳ if else

7.1.2 Pernyataan if sederhana

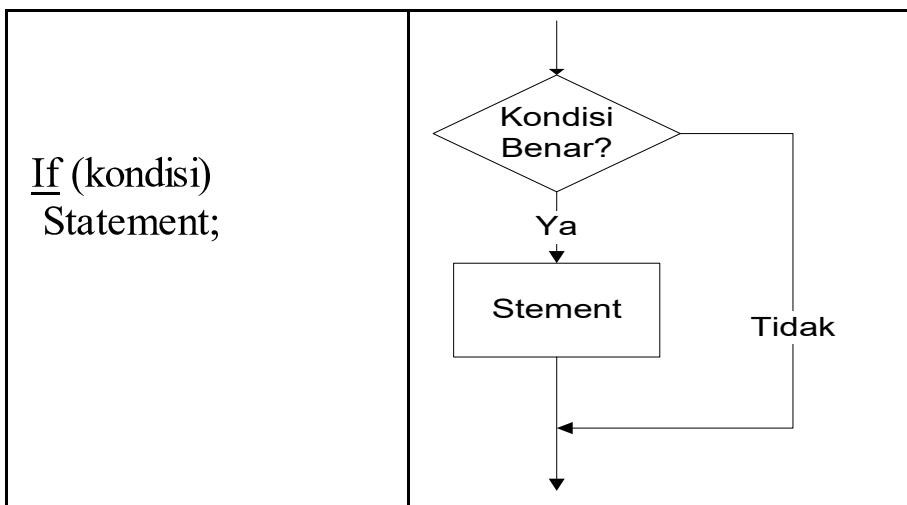
Bentuk IF yang juga dikenal dengan istilah IF Statement, memiliki bentuk umum sebagai berikut. Pernyataan if paling sederhana berbentuk :

if (kondisi)
pernyataan

Keterangan :

- ↳ Kondisi digunakan untuk menentukan pengambilan keputusan
- ↳ Pernyataan dapat berupa sebuah pernyataan-pernyataan majemuk.
- ↳ Nilai yang dihasilkan adalah True (Benar)

Flowchart if sederhana berikut :



Pada variasi ini, apabila kondisi bernilai benar maka Statement dikerjakan dan apabila kondisi bernilai salah maka Statement tidak dikerjakan.

Penerapan **if** misalnya untuk menentukan seseorang boleh atau tidak menonton pertunjukkan bioskop. Kondisi yang digunakan seseorang boleh menonton kalau sudah berusia 17 tahun.

Contoh 20	Hasil
<pre>/*-----* /* Contoh 20 : Penggunaan if dalam pengambilan * /* keputusan * /*-----* #include <iostream.h> #include <conio.h> void main() { int usia; clrscr(); // Hapus layar cout << "Berapa usia anda ? "; cin >> usia; if (usia < 17) cout << " Anda tidak diperkenankan menonton" << endl; }</pre>	Barapa usia anda ? 16 ↴ Anda Tidak diperkenankan menonton

Apabila program dieksekusi lagi untuk memasukkan usia diatas 17 maka :

Tampak diatas bila dimasukkan usia diatas 17 maka pesan tidak akan ditampilkan. Untuk mengatasi hal ini dapat dilakukan dengan menggunakan pernyataan kondisi **if else**.

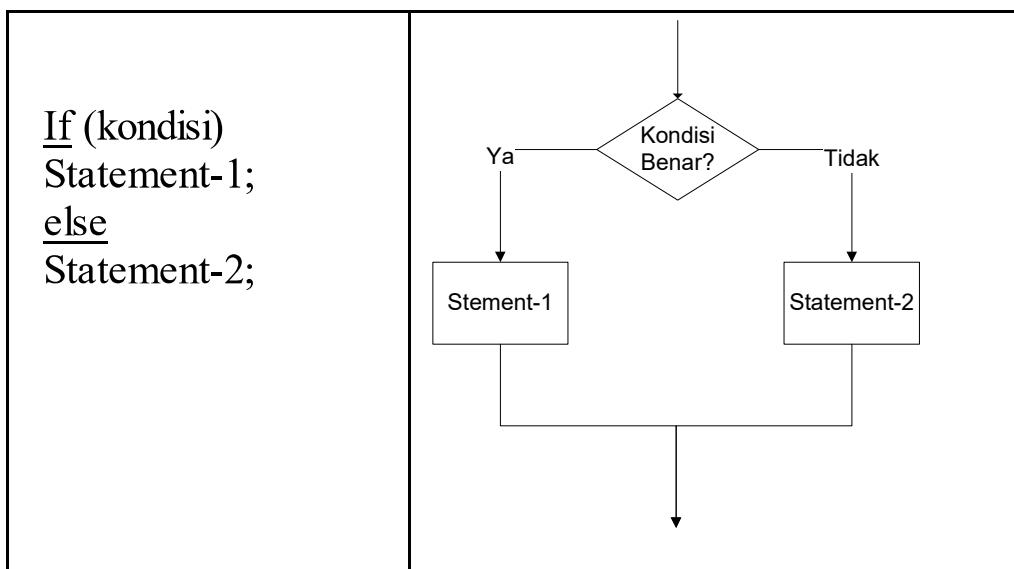
Contoh 20	Hasil
<pre>/* * Contoh 20 : Penggunaan if dalam pengambilan * * keputusan * */ #include <iostream.h> #include <conio.h> void main() { int usia; clrscr(); // Hapus layar cout << "Berapa usia anda ? "; cin >> usia; if (usia < 17) cout << " Anda tidak diperkenankan menonton" << endl; }</pre>	Barapa usia anda ? 17 ↴

7.1.3 Pernyataan if else

Pernyataan **if else** mempunyai bentuk sebagai berikut :

```
if (kondisi)
    Pernyataan 1;
else
    Pernyataan 2;
```

Flowchart if else berikut :



Pada variasi ini, apabila kondisi bernilai benar maka Statement-1 yang dikerjakan dan apabila kondisi bernilai salah maka Statement-2 yang dikerjakan (tidak pernah 2 statement ini dikerjakan semua).

Untuk **if else** kita dapat menggunakan contoh 21 untuk melihat perbedaan dengan if sederhana.

Contoh 21	Hasil
<pre>/* _____ /* Contoh 21 : Penggunaan if else dalam * /* pengambilan keputusan * _____ _____ #include <iostream.h> #include <conio.h> void main() { int usia; clrscr(); // Hapus layar cout << "Berapa usia anda ? "; cin >> usia; if (usia < 17) cout << " Anda tidak diperkenankan menonton" << endl; else cout << " Selamat menonton" << endl; }</pre>	Barapa usia anda ? 16 ↴ Anda Tidak diperkenankan menonton

Apabila kita memasukkan umur lebih dari 17 maka hasil eksekusi yang didapat adalah :

Contoh 21	Hasil
<pre>/* -----* * Contoh 21 : Penggunaan if else dalam * * pengambilan keputusan * * -----* #include <iostream.h> #include <conio.h> void main() { int usia; clrscr(); // Hapus layar cout << "Berapa usia anda ? "; cin >> usia; if (usia < 17) cout << " Anda tidak diperkenankan menonton" << endl; else cout << " Selamat menonton" << endl; }</pre>	Barapa usia anda ? 17 ↴ Selamat menonton

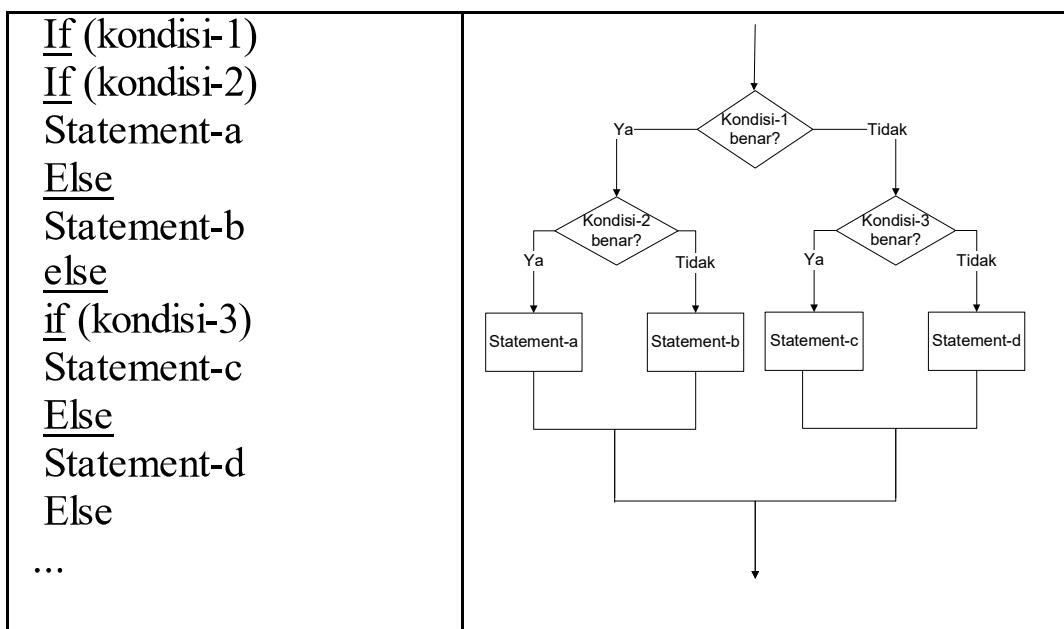
7.1.4 Pernyataan if dalam if

Pernyataan **if** yang terletak dalam **if** sering disebut *nested if* atau **if** bersarang. Salah satu bentuknya adalah :

```
if (kondisi1)
    pernyataan1;
else if (kondisi2)
    pernyataan2;
else if (kondisi3)
    pernyataan3;
if (kondisiM)
    pernyataanM;
else /*Opsional*/
    pernyataanN; /*Opsional*/
```

Bentuk pernyataan if seperti ini bermanfaat untuk menyeleksi sejumlah kemungkinan tindakan. Penyeleksian dilakukan secara bertingkat.

Flowchart if dalam if berikut :



Pada variasi ini, apabila kondisi-1 bernilai benar maka dilanjutkan memeriksa kondisi-2. Apabila kondisi-2 bernilai benar maka Statement-a yang dikerjakan dan apabila kondisi-2 bernilai salah maka Statement-b yang dikerjakan. Sedangkan apabila kondisi-1 salah maka dilanjutkan memeriksa kondisi-3. Apabila kondisi-3 bernilai benar maka Statement-c yang dikerjakan dan apabila kondisi-3 bernilai salah maka Statement-d yang dikerjakan. (dari 4 statement yang ada hanya salah satu yang dikerjakan).

Contoh 22	Hasil
<pre>/* _____ /* Contoh 22 : Pemakaian if bertingkat untuk * /* menentukan nama hari * /*_____ _____ #include <iostream.h> #include <conio.h> void main () { int kode_hari;</pre>	<p>Menentukan hari 1 = Senin 3 = Rabu 5 = Jum'at 7 = Minggu 2 = Selasa 4 = Kamis 6 = Sabtu Kode hari [1..7] : 2 ↴ Selasa</p>

```

clrscr(); // Hapus layar
cout << "Menentukan hari " << endl;
cout << "1 = Senin 3 = Rabu 5 = Jum'at
= Minggu " << endl;
cout << "2 = Selasa 4 = Kamis 6 = Sabtu "
<< endl;
cout << "Kode hari [1..7] : ";
cin >> kode_hari;
// Proses seleksi;
if (kode_hari == 1)
    cout << "Senin" << endl;
if (kode_hari == 2)
    cout << "Selasa" << endl;
if (kode_hari == 3)
    cout << "Rabu" << endl;
if (kode_hari == 4)
    cout << "Kamis" << endl;
if (kode_hari == 5)
    cout << "Jum'at" << endl;
if (kode_hari == 6)
    cout << "Sabtu" << endl;
if (kode_hari == 7)
    cout << "Minggu" << endl;
else
    cout << "Kode hari salah" << endl;
}
}

```

Program diatas pertama-tama meminta kode hari dimasukkan dari keyboard. Kemudian **if** dan **else** secara bertingkat akan menyeleksi nilai tersebut dan mememberikan nama hari. Bila anda memasukkan kode hari yang salah maka :

Contoh 22	Hasil
<pre> /* _____ _____ /* Contoh 22 : Pemakaian if bertingkat untuk * /* menentukan nama hari * _____ _____ #include <iostream.h> #include <conio.h> void main () { int kode_hari; clrscr(); // Hapus layar </pre>	Menentukan hari 1 = Senin 3 = Rabu 5 = Jum'at 7 = Minggu 2 = Selasa 4 = Kamis 6 = Sabtu Kode hari [1..7] : 9 ↴ Kode hari salah

```

cout << "Menentukan hari " << endl;
cout << "1 = Senin 3 = Rabu 5 = Jum'at 7
= Minggu " << endl;
cout << "2 = Selasa 4 = Kamis 6 = Sabtu "
<< endl;
cout << "Kode hari [1..7] : " ;
cin >> kode_hari;
// Proses seleksi;
if (kode_hari == 1)
    cout << "Senin" << endl;
if (kode_hari == 2)
    cout << "Selasa" << endl;
if (kode_hari == 3)
    cout << "Rabu" << endl;
if (kode_hari == 4)
    cout << "Kamis" << endl;
if (kode_hari == 5)
    cout << "Jum'at" << endl;
if (kode_hari == 6)
    cout << "Sabtu" << endl;
if (kode_hari == 7)
    cout << "Minggu" << endl;
else
    cout << "Kode hari salah" << endl;
}
}

```

7.1.5 Terapan bentuk-bentuk IF

Sebuah masalah terkadang dapat diselesaikan dengan berbagai cara, seperti penggunaan “if tanpa else” dan “if dengan else”. Sebagai contoh dapat dilihat pada kasus berikut:

Kasus 7.1 : Menentukan apakah bilangan yang diinput positip atau negatip

Solusi : ada beberapa cara berikut

<i>Solusi-1</i>	<i>Solusi-2</i>
Input(bil) If (bil>=0) Output('positip') Else Output('negatip')	Input(bil) If (bil<0) Output('negatip') Else Output('positip')

Input(bil) Ket \leftarrow ‘positip’ If (bil<0) Ket \leftarrow ‘negatip’ Output(Ket)	Input(bil) Ket \leftarrow ‘negatip’ If (bil \geq 0) Ket \leftarrow ‘positip’ Output(Ket)
<i>Solusi-5</i> Input(bil) If (bil \geq 0) Output(‘positip’) If (bil<0) Output(‘negatip’)	<i>Solusi-6</i> Input(bil) If (bil<0) Output(‘negatip’) If (bil \geq 0) Output(‘positip’)
<i>Solusi-7</i> Input(bil) positip \leftarrow bil \geq 0 If (positip=true) Output(‘positip’) else Output(‘negatip’)	<i>Solusi-8</i> Input(bil) positip \leftarrow bil \geq 0 If (positip) Output(‘positip’) else Output(‘negatip’)

Ulasan dari beberapa solusi:

Solusi-1 dan Solusi-2 adalah solusi yang sama, digunakan kondisi berkebalikan sehingga posisi perintah tampilan ditukar.

Solusi-3 dan Solusi-4 juga sama, keduanya menggunakan “if tanpa else”, dengan cara variabel Ket diinisialisasi (diberi nilai awal) dengan salah satu kemungkinan hasilnya, kemudian diubah bila memenuhi kondisi.

Solusi-5 dan Solusi-6 juga sama, pada solusi ini dibuat 2 buah “if tanpa else” secara terpisah. Dengan cara ini, berarti akan dilakukan pemeriksaan kondisi 2 kali (padahal sebenarnya cukup satu kali).

Solusi-7 dan Solusi-8 keduanya menggunakan variabel bertipe boolean bernama positip untuk mencatat hasil pembandingan bil \geq 0. Penulisan “if (positip=true)” sama saja dengan menuliskan “if (positip)” cara yang terakhir lebih cepat waktu eksekusinya.

Berikut beberapa kasus yang lain:
Kasus 7.2 : Terbesar dari 3 bilangan

<p><i>Solusi-1</i></p> <pre> Input(A,B,C) <u>If</u> (A>B) <u>If</u> (A>C) Output('terbesar =',A) <u>Else</u> Output('terbesar =',C) <u>else</u> <u>if</u> (B>C) Output('terbesar =',B) <u>Else</u> Output('terbesar =',C) </pre>	<p><i>Solusi-2</i></p> <pre> Input(A,B,C) <u>If</u> (A<B) <u>If</u> (B<C) Output('terbesar =',C) <u>Else</u> Output('terbesar =',B) <u>else</u> <u>if</u> (A<C) Output('terbesar =',C) <u>Else</u> Output('terbesar =',A) </pre>
<p><i>Solusi-3</i></p> <pre> Input(A,B,C) <u>If</u> (A>B and A>C) Output('terbesar =',A) <u>Else</u> <u>if</u> (B>A and B>C) Output('terbesar =',B) <u>Else</u> <u>if</u> (C>A and C>B) Output('terbesar =',C) </pre>	<p><i>Solusi-4</i></p> <pre> Input(A,B,C) <u>If</u> (A>B and A>C) Output('terbesar =',A) <u>Else if</u> (B>A and B>C) Output('terbesar =',B) <u>Else if</u> (C>A and C>B) Output('terbesar =',C) </pre>
<p><i>Solusi-5</i></p> <pre> Input(A,B,C) Max ← A <u>If</u> (B>Max) Max ← B <u>End if</u> <u>If</u> (C>Max) Max ← C Output('terbesar = ',Max) </pre>	<p><i>Solusi-6</i></p> <pre> Input(A,B,C) <u>If</u> (A>B) Max ← A <u>else</u> Max ← B <u>If</u> (C>Max) Max ← C Output('terbesar = ',Max) </pre>

Ulasan dari beberapa solusi:

Solusi-1,Solusi-2 dan Solusi-3 menggunakan 3 buah kondisi dan setiap hasil yang didapat akan melalui pemeriksaan 2 buah kondisi.

Solusi-4 menggunakan kondisi yang terdiri dari 2 pembandingan, dengan rata-rata melakukan pemeriksaan 2 kondisi (4 pembandingan)

Solusi-5 dan Solusi-6 digunakan 2 buah if yang terpisah, dimana hasil sementara nilai terbesar dicatat di tempat baru (Max), cara ini lebih praktis terutama kalau dikembangkan untuk mencari terbesar dari banyak bilangan.

Kasus 7.3 : Pembayaran air minum PDAM

PDAM menerapkan pembayaran air minum perumahan dengan cara perhitungan sebagai berikut :

- ↳ Tarif per m³ untuk 10 m³ pertama (1-10) adalah 2.000
- ↳ Tarif per m³ untuk 10 m³ kedua (11-20) adalah 3.000
- ↳ Tarif per m³ untuk 10 m³ ketiga (21-30) adalah 4.000
- ↳ Tarif per m³ untuk 10 m³ selanjutnya (31 ke atas) adalah 5.000
- ↳ Pemakaian air dihitung minimal 10 m³ (kurang dari 10 m³ dianggap 10 m³)
- ↳ Biaya administrasi bulanan sebesar 10.000

Bagaimana membuat algoritma untuk menghitung biaya tersebut?

Contoh kasus

Penggunaan air 5 m³ dengan biaya :

$$10 \times 2.000 + 10.000 = 30.000$$

Penggunaan air 15 m³ dengan biaya :

$$10 \times 2.000 + 5 \times 3.000 + 10.000 = 45.000$$

Penggunaan air 75 m³ dengan biaya :

$$10 \times 2.000 + 10 \times 3.000 + 10 \times 4.000 + 45 \times 5.000 + 10.000 = 325.000$$

Solusi :

Pemakaian air dibagi menjadi 4 area pemakaian (misal area a,b,c,d), baru dihitung total biaya

Solusi-1

```
Input(pakai)
If (pakai>30)
a← 10
b← 10
c← 10
d← pakai - 30
Else If (pakai>20)
a← 10
b← 10
c← pakai - 20
d← 0
Else If (pakai>10)
a← 10
b← pakai - 10
c← 0
d← 0
Else
a← 10
b← 0
c← 0
d← 0
biaya← a * 2000 + b * 3000 +
c * 4000 + d * 5000 +
10000
Output('biaya =',biaya)
```

Solusi-2

```
Input(pakai)
a← 10
b← 0
c← 0
d← 0
If (pakai>30) then
b← 10
c← 10
d← pakai - 30
Else If (pakai>20)
b← 10
c← pakai - 20
Else If (pakai>10) then
b← pakai - 10
biaya← a * 2000 + b * 3000 +
c * 4000 + d * 5000 +
10000
Output('biaya =',biaya)
```

Ulasan solusi :

Pada solusi-1, tiap aksi dari if , terdiri dari 4 statement mengisi a,b,c dan d. Bentuk solusi ini disederhanakan pada solusi-2 dengan cara memberikan nilai awal sebelum masuk if.

Kasus 7.4 : Indeks Nilai Kuliah

Indeks nilai sebuah matakuliah didapat dengan cara menghitung nilai akhir berdasarkan persentase komponen-komponennya kemudian

ditentukan indeks nilainya. Misal digunakan ketentuan sebagai berikut:

- ↳ Nilai Akhir dihitung dari 30% UTS, 40%UAS, 20% Tugas dan 10% kehadiran
- ↳ Indeks Nilai ditentukan berdasarkan Nilai Akhir (NA),
 - Bila $NA \geq 85$ maka Indeksnya A
 - Bila $85 > NA \geq 70$ maka Indeksnya B
 - Bila $70 > NA \geq 55$ maka Indeksnya C
 - Bila $55 > NA \geq 40$ maka Indeksnya D
 - Bila $NA < 40$ maka Indeksnya E

Bagaimana membuat algoritma untuk menentukan Indeks Nilai tersebut?

Solusi-1

```

Input(UTS,UAS,Tugas,Abs)
NAS ← 0.3*UTS + 0.4*UAS +
0.2*Tugas + 0.1*ABS
If (NAS>=85)
Indeks ← 'A'
ElseIf (NAS>=70 and NAS<85)
Indeks ← 'B'
ElseIf (NAS>=55 and NAS<70)
Indeks ← 'C'
ElseIf (NAS>=40 and NAS<55)
Indeks ← 'D'
Else
Indeks ← 'E'
Output('Indeks Nilai =',Indeks)
```

Solusi-2

```

Input(UTS,UAS,Tugas,Abs)
NAS ← 0.3*UTS + 0.4*UAS +
0.2*Tugas + 0.1*ABS
If (NAS>=85)
Indeks ← 'A'
Else If (NAS>=70)
Indeks ← 'B'
Else If (NAS>=55)
Indeks ← 'C'
Else If (NAS>=40)
Indeks ← 'D'
Else
Indeks ← 'E'
Output('Indeks Nilai =',Indeks)
```

Ulasan solusi :

Pada solusi-2 lebih baik dari solusi-1 karena pemeriksaan kondisi yang serupa tidak dilakukan dua kali. Pada solusi-1, pembandingan NAS dengan 85 dilakukan dua kali, yang pertama $NAS \geq 85$ dan yang kedua $NAS < 85$ adalah pembandingan yang serupa.

7.1.6 Pernyataan Case

Case adalah pernyataan yang digunakan untuk menjalankan salah satu pernyataan dari beberapa kemungkinan pernyataan, berdasarkan nilai dari sebuah ungkapan dan nilai penyeleksi. Sebenarnya semua bentuk pemilihan dapat ditulis dengan IF, namun penulisan dengan IF untuk banyak pilihan terasa kurang praktis. Bentuk CASE adalah cara lain penulisan bentuk pemilihan yang lebih sederhana, namun bentuk ini hanya dapat menggantikan IF apabila memenuhi syarat:

- ↳ kondisi berupa pembandingan kesamaan (dengan tanda “==”)
- ↳ nilai yang dibandingkan bertipe ordinal (integer,char dan boolean)

Bentuk CASE yang juga dikenal dengan istilah CASE Statement, memiliki bentuk umum sebagai berikut :

```
switch (ungkapan)
{
    case ungkapan1;
        pernyataan_1;
        break;
    case ungkapan2;
        pernyataan_2;
        break;
    .....
    default : /*Opsinal*/
        pernyataan_x; /*Opsinal*/
}
```

Pada pernyataan switch, *ungkapan* dapat berupa ungkapan, konstanta ataupun variabel. Adapun *ungkapan1*, *ungkapan2* dan seterusnya dapat berupa sembarang konstanta bertipe **int** atau **char**.

Contoh 23	Hasil
<pre data-bbox="228 305 909 1998"> /* _____ /* Contoh 23 : Pemakaian switch untuk menentukan * /* nama hari * _____ _____ #include <iostream.h> #include <conio.h> void main() { int kode_hari; clrscr(); // Hapus layar cout << "Menentukan Hari"" << endl; cout << "1 = Senin 3 = Rabu 5 = Jum'at 7 = Minggu << endl; cout << "2 = Selasa 4 = Kamis 6 = Sabtu " << endl; cout << "Kode hari [1..7] : " ; cin >> kode_hari; // Proses seleksi dengan switch switch (kode_hari) { case 1: cout << "Senin" << endl; break; case 2 : cout << "Selasa" << endl; break; case 3 : cout << "Rabu" << endl; break; case 4 : cout << "Kamis" << endl; break; case 5 : cout << "Jum'at" << endl; break; case 6 : cout << "Sabtu" << endl; break; case 7 : cout << "Minggu" << endl; break; default : cout << "Kode hari salah" << endl; break; } // akhir switch } </pre>	<p>Menentukan hari 1 = Senin 3 = Rabu 5 = Jum'at 7 = Minggu 2 = Selasa 4 = Kamis 6 = Sabtu Kode hari [1..7] : 2 ↴ Selasa</p>

Kehadiran **break** pada setiap **case** sangat penting. Sebab **break** akan menyebabkan keluar dari **switch**.

Program diatas pertama-tama meminta kode hari dimasukkan dari keyboard. Kemudian **if** dan **else** secara bertingkat akan menyeleksi nilai tersebut dan memeberikan nama hari. Bila anda memasukkan kode hari yang salah maka :

Contoh 23	Hasil
<pre>/* *-----* /* Contoh 23 : Pemakaian switch untuk menentukan * /* nama hari * *-----* #include <iostream.h> #include <conio.h> void main() { int kode_hari; clrscr(); // Hapus layar cout << "Menentukan Hari"" << endl; cout << "1 = Senin 3 = Rabu 5 = Jum'at 7 = Minggu "<< endl; cout << "2 = Selasa 4 = Kamis 6 = Sabtu " << endl; cout << "Kode hari [1..7] : " ; cin >> kode_hari; // Proses seleksi dengan switch switch (kode_hari) { case 1: cout << "Senin" << endl; break; case 2 : cout << "Selasa" << endl; break; case 3 : cout << "Rabu" << endl; break; case 4 : cout << "Kamis" << endl; break; case 5 : cout << "Jum'at" << endl; break; case 6 : cout << "Sabtu" << endl; break; case 7 : cout << "Minggu" << endl;</pre>	Menentukan hari 1 = Senin 3 = Rabu 5 = Jum'at 7 = Minggu 2 = Selasa 4 = Kamis 6 = Sabtu Kode hari [1..7] : 9 ↴ Kode hari salah

```

break;
default :
cout << "Kode hari salah" << endl;
break;
} // akhir switch
}

```

7.1.7 Terapan bentuk-bentuk CASE

Kasus 7.5 : Menentukan nama hari dari nomor hari yang diinput
Dinput nomor hari, ditampilkan namanya, bagaimana algoritmanya?

Solusi dengan IF dan CASE

<i>Solusi-If</i>	<i>Solusi-Case</i>
Input(NoHari) <u>If</u> (NoHari=1) <u>then</u> NmHari ← ‘Senin’ <u>Else If</u> (NoHari=2) NmHari ← ‘Selasa’ <u>Else If</u> (NoHari=3) NmHari ← ‘Rabu’ <u>Else If</u> (NoHari=4) NmHari ← ‘Kamis’ <u>Else If</u> (NoHari=5) NmHari ← ‘Jumat’ <u>Else If</u> (NoHari=6) NmHari ← ‘Sabtu’ <u>Else If</u> (NoHari=7) <u>then</u> NmHari ← ‘Minggu’ <u>Output</u> (NmHari)	Input(NoHari) <u>Case</u> NoHari 1: NmHari ← ‘Senin’;Break; 2: NmHari ← ‘Selasa’;Break; 3: NmHari ← ‘Rabu’;Break; 4: NmHari ← ‘Kamis’;Break; 5: NmHari ← ‘Jumat’;Break; 6: NmHari ← ‘Sabtu’;Break; 7: NmHari ← ‘Minggu’;Break; <u>Output</u> (NmHari)

Pada solusi-2 terlihat lebih sederhana dan mudah dibaca dibanding dengan solusi-1.

Kasus 7.6 : Merubah angka menjadi kalimat

Dinput bilangan/angka (angka dibatasi 1-99), ditampilkan kata-kata/kalimat dari bilangan tersebut, bagaimana algoritmanya?

Solusi

Solusi-Case

Input(bil)

pul \leftarrow bil div 10

sat \leftarrow bil mod 10

Kalimat \leftarrow “

Case sat

1: Kalimat \leftarrow ‘Satu’

2: Kalimat \leftarrow ‘Dua’

3: Kalimat \leftarrow ‘Tiga’

4: Kalimat \leftarrow ‘Empat’

5: Kalimat \leftarrow ‘Lima’

6: Kalimat \leftarrow ‘Enam’

7: Kalimat \leftarrow ‘Tujuh’

8: Kalimat \leftarrow ‘Delapan’

9: Kalimat \leftarrow ‘Sembilan’

Case pul

1: Case sat

0: Kalimat \leftarrow ‘Sepuluh’

1: Kalimat \leftarrow ‘Sebelas’

Otherwise: Kalimat \leftarrow Kalimat + ‘ belas’

2: Kalimat \leftarrow ‘Dua Puluh’ + Kalimat

3: Kalimat \leftarrow ‘Tiga Puluh’ + Kalimat

4: Kalimat \leftarrow ‘Empat Puluh’ + Kalimat

5: Kalimat \leftarrow ‘Lima Puluh’ + Kalimat

6: Kalimat \leftarrow ‘Enam Puluh’ + Kalimat

7: Kalimat \leftarrow ‘Tujuh Puluh’ + Kalimat

8: Kalimat \leftarrow ‘Delapan Puluh’ + Kalimat

9: Kalimat \leftarrow ‘Sembilan Puluh’ + Kalimat

Output(Kalimat)

Pada solusi di atas, satuan diproses dengan case pertama, selanjutnya puluhan diproses CASE kedua. Pada puluhan=1 (angka belasan) dibagi lagi menjadi 3 kemungkinan, karena bunyi kalimatnya ada 3 macam,

7.2 Konversi Struktur IF dan CASE ke Bahasa C

Berikut ini diberikan pedoman konversi dari algoritma ke dalam bahasa C untuk struktur IF dan CASE:

Algoritma	Bahasa C
If kondisi then Aksi End if	if (kondisi) { Aksi; }
If kondisi then Aksil Else Aksi2 End if	If (kondisi) { Aksil; } else { Aksi2; }
If kondisil then Aksil Else if kondisi2 Aksi2 Else Aksi3 End if	if (kondisil) { Aksil; } else if (kondisi2){ Aksi2; } else { Aksi3; }
Case ekspresi Nilai1: Aksil Nilai2: Aksi2 Nilai3: Aksi3 End case	switch (ekspresi) { case Nilai1: Aksil; Break; case Nilai2: Aksi2; Break; case Nilai3: Aksi3; }
Case ekspresi Nilai1: Aksil Nilai2: Aksi2 Nilai3: Aksi3 Otherwise: Aksi4 End case	switch (ekspresi) { case Nilai1: Aksil; Break; case Nilai2: Aksi2; Break; case Nilai3: Aksi3; Break; default: Aksi4;

<u>Case ekspresi</u> Nilai-1,Nilai-2,Nilai-3: Aksi1 Nilai-4,Nilai-5: Aksi2 Nilai-6..Nilai-8: Aksi3 Otherwise: Aksi4 <u>End Case</u>	<code>}</code> <pre>switch (ekspresi) { case Nilai1: case Nilai2: case Nilai3: Aksi1; Break; case Nilai4: case Nilai5: Aksi2; Break; case Nilai6: case Nilai7: case Nilai8: Aksi3; Break; default: Aksi4; }</pre>
--	--

Catatan:

- penulisan kondisi pada IF dan ekspresi pada CASE dalam bahasa C harus digunakan tanda kurung ().
- aksi berupa satu perintah atau lebih, masing-masing diakhiri titik koma.
- apabila aksi hanya berupa satu perintah, penggunaan { } dapat dihilangkan.
- kata “if”, “else”, “switch”, “case” dan “default” dalam bahasa C, harus ditulis dengan huruf kecil semua.
- dalam bahasa C tidak ada kata “then”, “end if” dan “end case” tetapi digantikan pasangan kurung kurawal { dan }
- hati-hati dengan penggunaan kesamaan, yaitu dengan “==” bukan “=”.
- string digunakan kutip dua (seperti “test”) bukan kutip satu (‘test’).

Contoh penulisan algoritma selengkapnya dan hasil konversinya ke bahasa C.

Diambil dari contoh pada Kasus 7.3

Algoritma	Bahasa C
<u>Algoritma</u> PDAM /* menghitung biaya pemakaian air*/ <u>Kamus Data</u> pakai,a,b,c,d : integer biaya : integer <u>Begin</u> Input(pakai) a← 10 b← 0 c← 0 d← 0 <u>If</u> (pakai>30) b← 10 c← 10 d← pakai - 30 <u>Else If</u> (pakai>20) b← 10 c← pakai - 20 <u>Else If</u> (pakai>10) b← pakai - 10 biaya← a * 2000 + b * 3000 + c * 4000 + d * 5000 + 5000 Output('biaya =',biaya)	<pre>#include <stdio.h> #include <conio.h> /*menghitung biaya pemakaian air*/ int main() { //Kamus Data int pakai,a,b,c,d; int biaya; //Begin printf("Masukkan pemakaian air: "); scanf("%d",&pakai); a=10; b=0; c=0; d=0; if (pakai>30) { b=10; c=10; d=pakai - 30; } else if (pakai>20) { b=10; c=pakai - 20; } else if (pakai>10) { b=pakai - 10; } biaya = a * 2000 + b * 3000 + c * 4000 + d * 5000 + 10000; printf("biaya = %d",biaya); getche(); return 0; //End }</pre>



Rangkuman

- Struktur pemilihan dapat digunakan untuk membuat program melakukan aksi tertentu sesuai nilai dari kondisi yang dipertimbangkan.
- Struktur pemilihan dapat berupa struktur IF ... THEN ..., struktur IF ... THEN ... ELSE ..., struktur CASE, maupun pemilihan bersarang IF atau CASE.
- Struktur IF ... THEN ... (tanpa ELSE) digunakan pada situasi dengan pilihan mengerjakan aksi atau tidak.
- Struktur IF ... THEN ... ELSE ... digunakan untuk memilih salah satu aksi dari berdasarkan nilai kondisi.
- Struktur CASE merupakan bentuk penyederhanaan dari struktur IF dengan persyaratan tertentu, yaitu kondisi berupa pembandingan kesamaan dan nilai yang dibandingkan harus ordinal (integer,char atau boolean).
- Klausu OTHERWISE pada struktur CASE bersifat opsional, seperti halnya ELSE pada struktur CASE.
- Struktur pemilihan bersarang dan struktur CASE dapat menyederhanakan program yang menyelesaikan kasus dengan nilai kondisi berjumlah lebih dari dua.