

6 STRUKTUR KAWALAN ATURCARA

Bab 2 kita telah mempelajari bagaimana hendak menyelesaikan masalah dgn menggunakan struktur jujukan, struktur pemilihan, struktur perulangan & gelung.

C mempunyai beberapa struktur kawalan utk mengawal aliran aturcara bagi melakukan pemilihan, perulangan & gelung. Struktur² ini berguna utk membuat keputusan.

Katakunci² yg digunakan dlm suruhan kawalan:

<i>if</i>	<i>else</i>	<i>goto</i>
<i>switch</i>	<i>case</i>	<i>while do</i>
<i>for</i>		

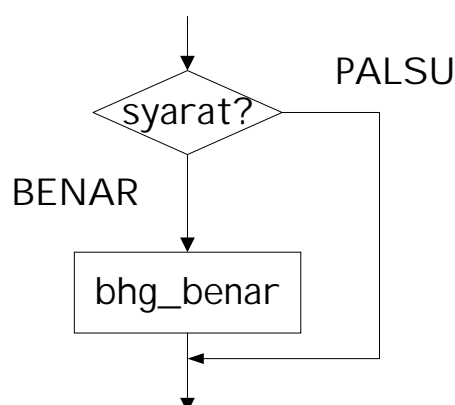
6.1 Struktur Pemilihan

- Dlm struktur pilihan, arahan² tertentu akan dipilih utk dilaksanakan, pemilihan ini bergantung kpd syarat pemilihan.
- C melaksanakan struktur pemilihan dgn 3 cara:
 - I) Pemilihan dgn *if* & *else*
 - II) Pemilihan dgn operator@pengolah bersyarat
 - III) Pemilihan dgn *switch*, *case* & *break*

6.1.1 Pemilihan dgn *if* & *else*

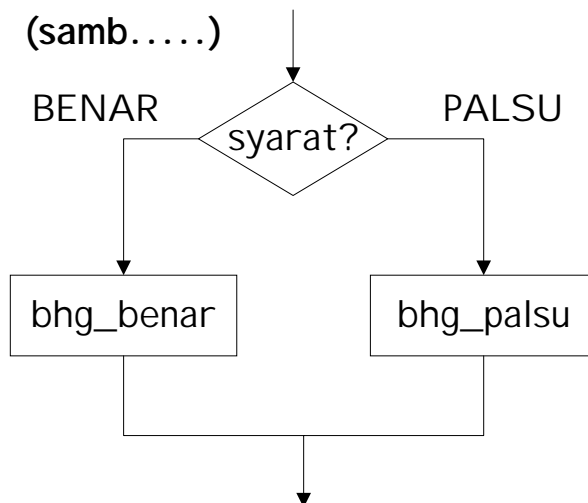
- Katakunci *if* & *else* digunakan utk memilih aliran aturcara @ membuat keputusan. Ada 3 format penggunaan *if* & *else* dlm aturcara:
- Ada 3 format penggunaan *if* & *else* dlm aturcara:

a) *if* (syarat)
 bhg_benar;

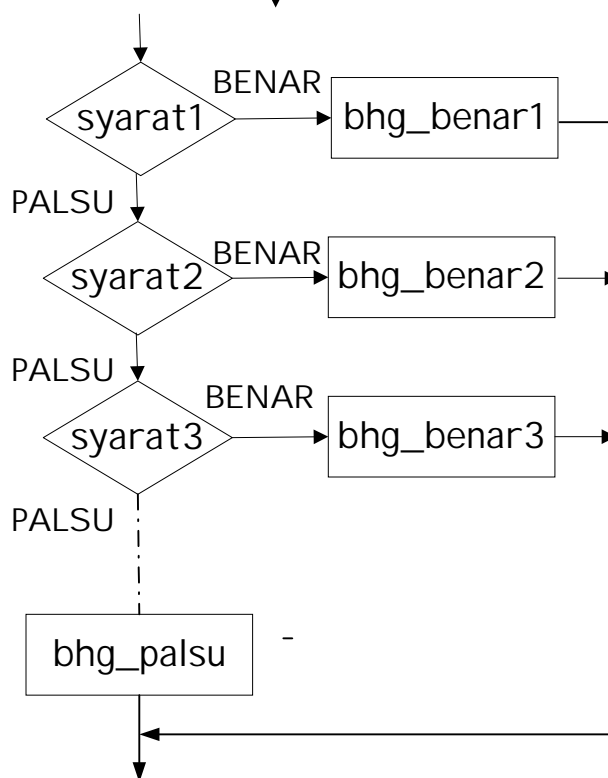


6.1.1 Pemilihan dgn if & else (samb.....)

b) `if (syarat)`
`bhg_benar;`
`else`
`bhg_palsu;`



c) `if (syarat1)`
`bhg_benar1;`
`else if (syarat2)`
`bhg_benar2;`
`else if (syarat3)`
`bhg_benar3;`
`:`
`else`
`bhg_palsu;`



○ Di dlm struktur a) - c) di atas:

syarat, syarat1 ... adalah ungkapan² logik dan hubungan mudah/kompleks.

bhg_benar ... terdiri dari 1 @ lebih kenyataan² C yg akan dilaksanakan jika **syarat...** adalah BENAR. Jika **bhg_benar** mempunyai lebih dari satu kenyataan, ia **mesti** diletakkan di antara tanda { }.

bhg_palsu ... terdiri dari 1 @ lebih kenyataan² C yg akan dilaksanakan jika **syarat...** adalah PALSU. Jika **bhg_palsu** mempunyai lebih dari satu kenyataan, ia mesti diletakkan di antara tanda { }.

Soalan: Di dlm struktur c) di atas, jika syarat1 & syarat3 adalah BENAR, bahagian benar manakah yg akan dilaksanakan ?

6.1.1 Pemilihan dgn if & else (samb.....)

Contoh penggunaan if & else:

- | | |
|---|--|
| <p>a) <code>if (x == 10)</code>
 <code>printf("sama dgn 10");</code></p> <p>b) <code>if (x < 0)</code>
 <code>printf("negati f");</code>
 <code>else</code>
 <code>printf("posi ti f");</code></p> <p>c) <code>if (x == 0)</code>
 <code>printf("si far");</code>
 <code>else if (x < 0)</code>
 <code>printf("negati f");</code>
 <code>else</code>
 <code>printf("posi ti f");</code></p> | <p>d) <code>if (x > 10)</code>
 <code>{</code>
 <code>printf("lebi h dari 10");</code>
 <code>++ki ra;</code>
 <code>}</code>
 <code>else</code>
 <code>{</code>
 <code>printf("keci l dari 10");</code>
 <code>--ki ra;</code>
 <code>}</code></p> <p>e) <code>if ((x>10) (y>20))</code>
 <code>printf("Memenuhi syarat");</code></p> |
|---|--|

- Contoh pengujian fail: Untuk memastikan tiada ralat dlm proses membuka fail, pengujian boleh dilakukan terutama bagi fail yg hendak dibaca. Semasa fail stor12. dat hendak dibuka & dibaca pengujian dilakukan.

```
#include <stdio.h>

main()
{
    FILE *failptr;
    failptr = fopen("stor12.dat", "r");

    if (failptr ==NULL)
    {
        printf("Ralat dalam pembukaan fail.\n")
        exit(-1); /*Tamatkan Aturcara*/
    }

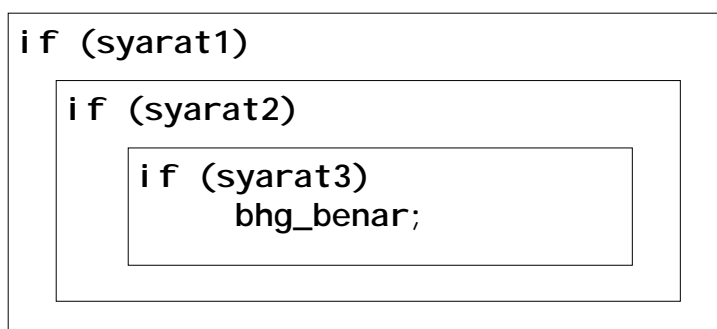
    return 0;
}
```

Kod ini boleh ditukar spt berikut - memberi maksud yg sama:

```
if(failptr = fopen("stor12.dat", "r"))== NULL
{
    puts("Ralat dalam pembukaan fail.");
    exit(-1); /*Tamatkan Aturcara*/
}
```

6.1.1 Pemilihan dgn if & else (samb.....)

- Jika dlm satu pemilihan if hanya ada 1@2 bahagian pilihan sahaja dinamakan struktur pemilihan mudah.
- Utk lebih dpd 2 pilihan **if bersarang** perlu digunakan. 2 cara if bersarang digunakan:
 - i) Keputusan Berbilang Alternatif
Telah diguna sebelum ini iaitu if .. else if ... else rujuk if format c) ms 2.
 - ii) Syarat if yang diberi dlm beberapa aras. Contoh 3 aras if bersarang:



if bersarang keputusan berbilang alternatif digunakan bila hanya satu data(1 pembolehubah) shj digunakan utk menentukan bahagian yang benar.

Contoh if bersarang menggunakan kep.berbilang alternatif:

Jika kadar kebisingan bunyi dalam desibel dihubungkan dgn kesannya terhadap tanggapan manusia. Jadual di bawah menghubungkan kadar bunyi dgn tanggapan manusia:

kebisingan dlm desibel (db)	Tanggapan manusia
50 atau kurang	Senyap
51-70	Menggangu
71-90	Menyakitkan hati
91-110	Sangat menyakitkan hati
lebih dpd 110	Tidak selesa

6.1.1 Pemilihan dgn if & else (samb.....)

Penyelesaian:

```

i f (bi si ng_db<=50)
    printf("%d db adalah Senyap\n", bi si ng_db);
el se i f (bi si ng_db<=70)
    printf("%d db adalah Mengganggu\n", bi si ng_db);
el se i f (bi si ng_db<=90)
    printf("%d db adalah Menyaki tkan hati \n", bi si ng_db);
el se i f (bi si ng_db<=110)
    printf("%d db adalah Sangat menyaki tkan hati \n", bi si ng_db);
el se
    printf("%d db adalah Ti dak sel esa\n", bi si ng_db);

```

Contoh syarat i f yang diberi dlm beberapa aras:

Tulis satu aturcara utk mengira & memaparkan luas satu segiempat. Saiz sisi² segiempat ini diberikan oleh pengguna melalui papan kekunci. Aturcara anda mesti memastikan saiz sisi² yg diberi bukanlah sifar atau negatif. Jika saiz yg diberikan adalah sifar, cetak mesej amaran kpd pengguna & jadikan luas sifar. Jika saiz yg diberikan adalah negatif, ambil magnitudnya shj.

Penyelesaian:

i) Analisa Masalah

Input:

media : papan kekunci
data : panjang & lebar

Output:

media : skrin
data : luas segiempat

Proses:

1. utk mencari magnitud sesuatu nombor:
 - jika nombor adalah negatif, magnitud = -(nombor)
 - jika nombor adalah positif, magnitud = nombor
2. luas segiempat = panjang x lebar

6.1.1 Pemilihan dgn if & else (samb.....)

Penyelesaian:

ii) Kod pseudo:

```

baca panjang & lebar dari pengguna
if panjang = 0 atau lebar = 0 then
  begin
    cetak mesej amaran
    luas := 0
  end
else
  begin
    if panjang negatif then
      panjang:= magnitud panjang
    endif
    if lebar negatif then
      lebar:= magnitud lebar
    endif
    luas := panjang * lebar
  end
endif
cetak luas

```

iii) Aturcara:

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    float luas, panjang, lebar;
```

```
    printf ("masukkan nilai panjang dan lebar\n");
```

```
    scanf ("%f %f", &panjang, &lebar);
```

```
    if (panjang== 0 || lebar ==0)
```

```
    {
```

```
        printf("%*@$%^#***@^\n");
```

```
        luas=0;
```

```
    }
```

```
    else
```

```
    {
```

```
        if (panjang < 0)
```

```
            panjang = -panjang; /*cari magnitud panjang*/
```

```
        if (lebar < 0 )
```

```
            lebar = -lebar; /* cari magnitud lebar*/
```

```
        luas = panjang*lebar;
```

```
    }
```

```
    printf ("luas segi empat ialah %f\n", luas);
```

```
}
```

if bersarang 2 aras

6.1.1 Pemilihan dgn `if` & `else` (samb.....)

Contoh syarat `if` yang diberi dlm beberapa aras:

Kementerian pertahanan hendak menyenaraikan nama-nama kakitangan lelaki yg berumur 20 hingga 26 tahun dan belum berkahwin.

Kenyataan `if` bersarang berikut boleh digunakan:

```
if (jantina == 'L' )
    if (taraf_perkahwinan == 'B' )
        if (umur <=26)
            if (umur>=20)
                printf("%s\n", nama );
```

ATAU kenyataan berikut juga boleh digunakan:

```
if (jantina == 'L' )&& (taraf_perkahwinan == 'B' )
    && (umur <=26) && (umur>=20)
        printf("%s\n", nama );
```

6.1.2 Pemilihan dgn operator@pengolah bersyarat

- C mempunyai singkatan bagi mengungkap kenyataan *if-else* menggunakan operator bersyarat `?` :

Format penggunaan operator bersyarat dlm aturcara:

syarat ? ungkapan_benar : ungkapan_palsu

Jika syarat adalah BENAR, maka keseluruhan ungkapan ini akan memberikan nilai ungkapan_benar. Sebaliknya, jika syarat adalah PALSU, maka keseluruhan ungkapan ini akan memberikan nilai ungkapan_palsu.

6.1.2 Pemilihan dgn pengolah bersyarat (Samb..)

○ Contoh:

i) `harga_tiket = (umur < 12) ? 10 : 20;`

pembolehubah harga_tiket akan diberikan nilai 10 jika nilai pembolehubah umur kurang dari 12 sebaliknya ia akan diberikan nilai 20 jika nilai pembolehubah umur lebih dari@ sama dgn 12. Kenyataan if-else yg setara:

```
if (umur < 12)
    harga_tiket = 10;
else
    harga_tiket = 20;
```

ii) Aturcara mencari luas segi4 menggunakan operator bersyarat:

```
#include <stdio.h>
main()
{
    float luas, panjang, lebar;

    printf ("masukkan nilai panjang dan lebar\n");
    scanf ("%f %f", &panjang, &lebar);
    if (panjang== 0 || lebar ==0)
    {
        printf("%*@$%^#***@^\n");
        luas=0;
    }
    else
    {
        panjang = (panjang < 0) ? -panjang : panjang;
        lebar = (lebar < 0) ? -lebar : lebar;
        luas = panjang*lebar;
    }
    printf ("luas segi empat ialah %f\n", luas);
}
```


Latihan

1. Bentukkan ungkapan² dlm C bagi mengungkap syarat² berikut:
 - a) **nombor** sama dgn atau lebih besar dari 1 tetapi lebih kecil dari 9
 - b) pembolehubah hrf bukan huruf q atau k
 - c) **nombor** adalah di antara 1 dan 9 tetapi bukan 5
 - d) **nombor** bukan di antara 1 dan 9

2. Tulis satu aturcara untuk memaparkan mesej samada satu huruf yg dibaca dari papan kekunci adalah huruf vokal @ konsonan.

3. Tukar aturcara mencari luas segiempat dlm contoh untuk mengira juga luas satu bulatan yg jejaringnya dibaca dari pengguna. Pengguna boleh memilih samada mencari luas bulatan atau luas segiempat.

4. Tulis aturcara yang memberi grade bagi satu mata pelajaran. Pengguna memasukkan peratusan markah pelajar dari papan kekunci dan aturcara akan menukarkan markah tersebut dalam bentuk gred dan dipaparkan ke skrin.

6.1.3 Pemilihan dgn switch, case & break

- Struktur *if-else-if-...-else* digunakan utk memilih aliran aturcara yg terdiri dari banyak alternatif. Kadang-kadang ini lebih mudah dilakukan dgn menggunakan struktur *switch-case-break*.
Format penggunaan *switch-case-break* dlm aturcara:

```
switch (ungkapan_integer)
{
    case pemalar1: kenyataan2A; break;
    case pemalar2: kenyataan2B; break;
    case pemalar3: kenyataan2C; break;
    :
    default: kenyataan2D;
}
```

Di mana *ungkapan_integer* adalah ungkapan yg akan menghasilkan nilai jenis integer @ char.

Cara operasi:

- 1) Aturcara akan mencari nilai *ungkapan_integer*.
- 2) Nilai ini akan dibandingkan dgn nilai² pemalar1, pemalar2 ... dlm kenyataan case pemalar1, case pemalar2 dsbnya satu demi satu sehingga aturcara jumpa nilai pemalar yg sama dgn nilai *ungkapan_integer*. Jika sama, kenyataan² di antara case .. break tersebut akan dilaksanakan. Misalnya jika *ungkapan_integer* == pemalar2, maka kenyataan^{2B} akan dilaksanakan.
- 3) Jika tiada *ungkapan_integer* tidak sama dgn kesemua pemalar² dlm senarai case maka kenyataan^{2D} di bhg default akan dilaksanakan. Bhg default adalah pilihan-boleh ditinggalkan jika tak perlu.
- 4) Katakunci **break** akan menyebabkan perlaksanaan selanjutnya akan bermula luar dari blok kenyataan `switch() { ... }`. Jika break ditinggalkan, kenyataan² selepasnya akan dilaksanakan.

6.1.3 Pemilihan dgn switch, case & break (Samb..)

i) Aturcara ini akan mencetak nombor 1, 2, @3 yg dimasukkan melalui papan kekunci dlm perkataan- satu, dua @ tiga

```
main()
{
    int x;
    scanf("%d", &x);
    switch (x)
    {
        case 1: puts ("satu"); break;
        case 2: puts ("dua"); break;
        case 3: puts ("tiga"); break;
        default: puts ("bukan satu dua tiga ");
    }
}
```

ii) Perhatikan output jika break dikeluarkan spt berikut:

```
main()
{
    int x;
    scanf("%d", &x);
    switch (x)
    {
        case 1: puts ("satu"); break;
        case 2: puts ("dua");
        case 3: puts ("tiga");
        default: puts ("bukan satu dua tiga ");
    }
}
```

<p>Masukan: 1 Jawapan: satu</p>	<p>Masukan: 3 Jawapan: tiga bukan satu dua tiga</p>
<p>Masukan: 2 Jawapan: dua tiga bukan satu dua tiga</p>	<p>Masukan: selain dpd 1,2 & 3 Jawapan: bukan satu dua tiga</p>

Soalan? Lukiskan carta alir utk bahagian swi tch, 2 kod di atas .

6.1.3 Pemilihan dgn switch, case & break (Samb..)

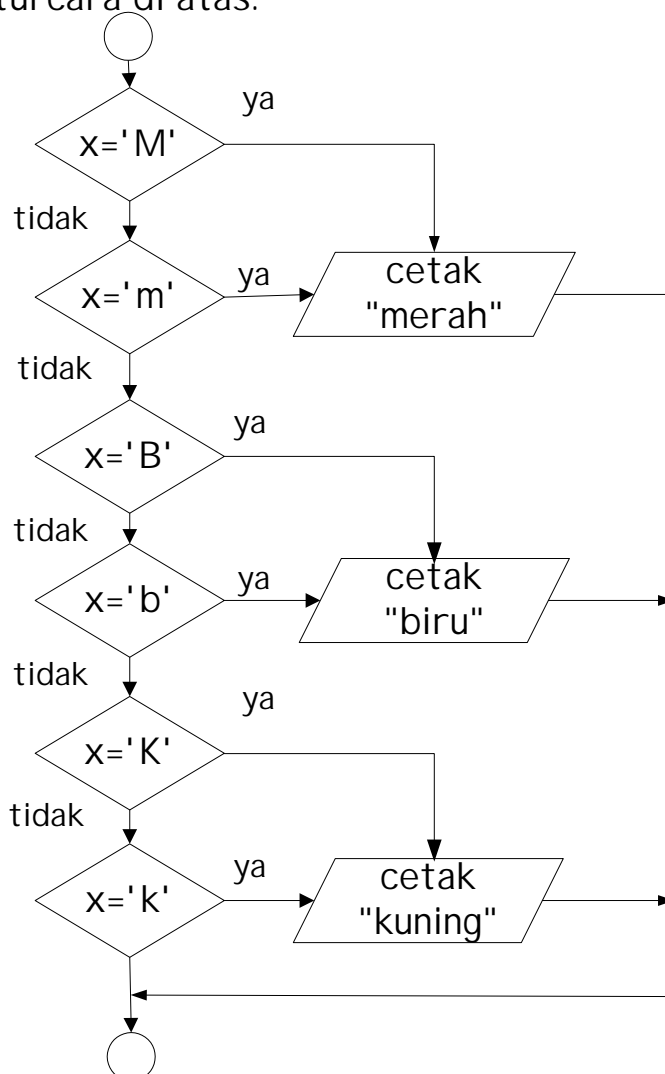
iii) Aturcara ini akan mencetak warna pada skrin, akan mencetakan "merah" jika aksara 'M' @ 'm' dimasukan, "biru" jika aksara B @ b dimasukkan & "kuning" jika aksara 'K' @ 'k' dimasukkan melalui papan kekunci.

```

main()
{
    char x;
    scanf("%d", &x);
    switch (x)
    {
        case 'M' :
        case 'm' : printf ("merah\n"); break;
        case 'B' :
        case 'b' : printf ("bi ru\n"); break;
        case 'K' :
        case 'k' : printf ("kuni ng\n"); break;
    }
}

```

Carta alir bagi aturcara di atas.

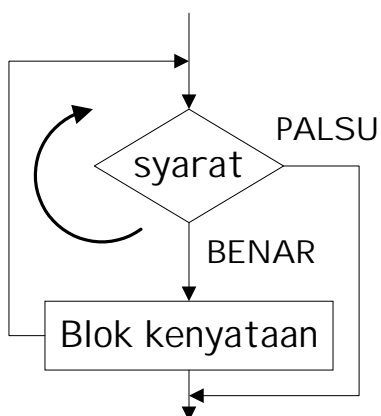


6.2.1 Gelung dgn while & do...while (samb...)

○ Perbezaan *while* dan *do ... while*:

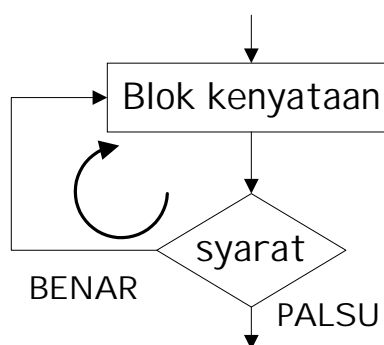
while

syarat diuji sebelum masuk ke gelung.
Blok kenyataan tidak akan dilaksanakan jika syarat adalah PALSU semasa masuk ke gelung.



do ... while

syarat diuji setelah masuk ke gelung
Blok kenyataan akan dilaksanakan sekurang²nya sekali jika syarat adalah PALSU semasa masuk ke gelung.



○ Contoh 1: Tulis satu aturcara utk memcetak nama Pn. Dayang di Skrin sehingga hujan. Perhatikan operasi dgn i/p & o/p fail berikut: while [i/p: y, o/p:TI ADA] do...while [i/p:yy, o/p:1)Pn. Dayang]

```
#include <stdio.h> #include <conio.h>
void main()
{
    FILE *keluar; int pengira=1; char hujan='n';
    keluar = fopen("dayang.out", "w");

    printf("Dah hujan? Tekan aksara y jika hujan>");
    hujan = getch();
    while ( hujan != 'y' )
    {
        fprintf (keluar, "%d) Pn. Dayang \n", pengira);
        printf("\nDah hujan? Tekan aksara y jika hujan>");
        hujan = getch();
        ++pengira;
    }
}
```

```
do {
    fprintf (keluar, "%d) Pn. Dayang \n", pengira);
    printf("\nDah hujan? Tekan aksara y jika hujan>");
    hujan = getch();
    ++pengira;
} while (hujan != 'y' );
```

6.2.1 Gelung dgn while & do...while (samb...)

- **Contoh 1:** Tulis satu aturcara utk mempamirkan sifir darab bagi 11. Corak output yg dikehendaki adalah spt berikut:

```

11 X 1 = 11
11 X 2 = 22
      :
11 X 12 = 121

```

Penyelesaian:

```

#include <stdio.h>
main()
{
    int nomb_sifir,
        pengira, kiraan_max;

    nomb_sifir = 11;
    pengira = 1; kiraan_max = 12;

    while ( pengira <= kiraan_max )
    {
        printf ("%d X %d = %d \n", nomb_sifir, pengira,
                pengira*nomb_sifir);
        ++pengira;
    }
}

```

- **Contoh 2:** Tulis satu aturcara permainan meneka nombor. Komputer akan memilih satu nombor & pengguna diminta meneka nombor tersebut. Selagi nombor yg diteka tidak sama dgn nombor yg dipilih oleh komputer, pengguna mesti terus meneka. Paparkan mesej samada tekaan adalah tepat atau sebaliknya.

Penyelesaian:

```

#include <stdio.h>
main()
{
    int nomb_komputer, nomb_teka;

    nomb_komputer = 17;
    do {
        puts ("sila teka nombor yg saya pilih (1-20)\n" );
        scanf ("%d", nomb_teka);
        if (nomb_teka != nomb_komputer)
            puts ("Anda salah, cuba lagi \n");
    } while ( nomb_teka != nomb_komputer );

    puts ("Tahni ah, anda betul \n");
}

```

6.2.2 Gelung dgn for

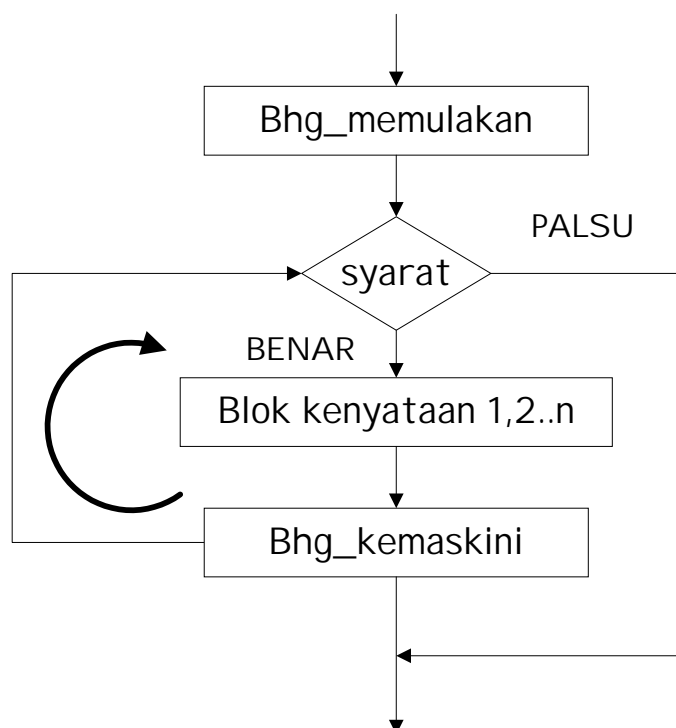
- Format penggunaan suruhan **if** dlm aturcara:

```
for (bhg_memulakan ; syarat ; bhg_kemaskini )
{
    kenyataan1;
    kenyataan2;
    :
    kenyataann;
}
```

Di mana syarat adalah satu ungkapan Boolean yg akan menghasilkan BENAR (1) @ PALSU (0).

Cara operasi:

- 1) ungkapan di bhg_memulakan dilaksanakan. Bhg ini biasanya digunakan utk mengumpuk nilai awalan kpd sesuatu pembolehubah. Dilaksanakan sekali sahaja.
- 2) syarat akan diuji. Jika syarat adalah BENAR, Blok kenyataan² 1,2...n akan dilaksanakan. Jika PALSU, gelung akan ditamatkan, dan perlaksanaan bermula selepas blok suruhan for.
- 3) ungkapan di bhg_kemaskini dilaksanakan dan perlaksanaan akan ulang langkah 2).



- Ungkapan² bhg_memulakan @ bhg_kemaskini boleh diabaikan ttp tanda semkolon mesti di kekalkan. Contohnya
for (; syarat ;)

6.2.2 Gelung dgn for (samb....)

- Contoh 1: Ulang contoh 1 m/s 15 menggunakan for.

Penyelesaian:

```
#include <stdio.h>
main()
{
    int nomb_sifir,
        pengira, kiraan_max;

    nomb_sifir = 11;
    kiraan_max =12;

    for ( pengira = 1; pengira <= kiraan_max; pengira++ )
    {
        printf ("%d X %d = %d \n", nomb_sifir, pengira,
                pengira*nomb_sifir);
    }
}
```

- Contoh 2: Apakah output bagi aturcara berikut ?

```
#include <stdio.h>
main()
{
    int i,j;

    for ( i = 6; i ; i-- )
    {
        for ( j = 0; j < i ; j ++ )
        {
            putchar ('#');
        }
        putchar ('\n');
    }
}
```

Penyelesaian:

```
#####
#####
####
###
##
#
```

6.2.2 Gelung dgn for (samb....)

- **Contoh 3:** Aturcara sifir dgn mengabaikan ungkapan bhg_memulakan & bhg_kemaskini .

Penyelesaian:

```
#include <stdio.h>
main()
{
    int nomb_sifir,
        pengira, kiraan_max;

    nomb_sifir = 11;
    kiraan_max = 12;
    pengira = 1;

    for ( ; pengira <= kiraan_max; )
    {
        printf ("%d X %d = %d \n", nomb_sifir, pengira,
                pengira*nomb_sifir);
        pengira++;
    }
}
```

6.3 Contoh Penggunaan Struktur Kawalan Pemilihan & Gelung

Tulis satu aturcara utk mengira & memaparkan jumlahlah markah pelajar di skrin. Markah² dibaca dari fail markah.dat.

Isi fail markah. dat

```
12
56
89
66
78
```

1. Ada ulangan dlm PM?

YA

2. Jika soalan 1 Ya, tahu ke, berapa kali perlu diulang?

Tidak

3. Jika soalan 2 Tidak, bila nak berhenti @ apakah syarat utk berhenti.

Sehingga habis semua markah dibaca di dlm fail

2 cara untuk memeriksa penghujung fail:

1. EOF - pemalar penghujung fail

2. feof - macro yg memulangkan nilai 0 selagi pemalar EOF belum dijumpai

```
#include<stdio.h>
#include<process.h>

void main (void)
{
    FILE *masuk;
    int jumlah = 0, markah, status;

    if ((masuk = fopen("markah.dat", "r"))== NULL)
    {
        puts("Ralat dalam pembukaan fail.");
        exit(-1); /*Tamatkan Aturcara*/
    }

    printf("Markah-markah Saya\n");
    while ((fscanf(masuk, "%d\n", &markah))!=EOF)
    {
        printf("%5d\n", markah);
        jumlah+=markah;
    }
    printf("Jumlah markah %d", jumlah);
}
```

6.3 Contoh Penggunaan Struktur Kawalan Pemilihan & Gelung(samb..)

Gelung utk membaca markah, mencetak markah & mengira jumlah markah boleh ditukar menggunakan macro feof spt berikut:

```
while (!feof(masuk))
{
    fscanf(masuk, "%d\n", &markah);
    printf("%5d\n", markah);
    jumlah+=markah;
}
```

Gelung while tersebut juga boleh digantikan dgn gelung for spt berikut:

```
for (status=fscanf(masuk, "%d\n", &markah);
     status != EOF; status=fscanf(masuk, "%d\n", &markah))
{
    printf("%5d\n", markah);
    jumlah+=markah;
}
```

Output di skrin:

<pre>Markah-markah Saya 12 56 89 66 78 Jumlah markah 301</pre>
--

6.4 Gelung Bersarang

- Gelung bersarang - pernyataan gelung yg mengandung gelung lain.

Tulis satu aturcara utk mempamirkan sifir darab bagi 1 hingga 5 & hanya hasil darab yg kurang & sama dgn 5 sahaja akan dicetak.

Penyelesaian:

```
#include <stdio.h>
main()
{
    int nomb_sifir, hasil_darab,
        pengira, sifir_max;

    sifir_max=5; hasil_darab=0; pengira = 1;

    for ( nomb_sifir=1; nomb_sifir<=sifir_max ;nomb_sifir++)
    {
        printf("\nSifir %d\n", nomb_sifir);
        hasil_darab = pengira*nomb_sifir;

        Gelung aras 1
        while (hasil_darab<=5)
        {
            Gelung aras 2
            printf ("%d X %d = %d \n", nomb_sifir,
                pengira, hasil_darab);

            pengira++;
            hasil_darab = pengira*nomb_sifir;
        }
        pengira=1;
    }
    return 0;
}
```

Output pada skrin dari
aturcara di atas

```
Sifir 1
1 X 1 = 1
1 X 2 = 2
1 X 3 = 3
1 X 4 = 4
1 X 5 = 5

Sifir 2
2 X 1 = 2
2 X 2 = 4

Sifir 3
3 X 1 = 3

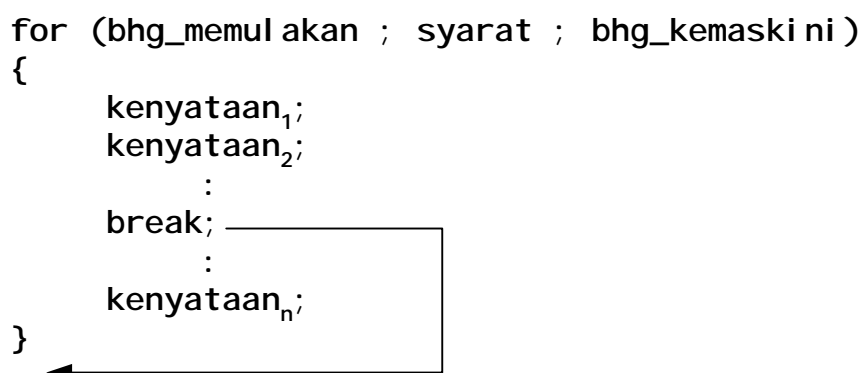
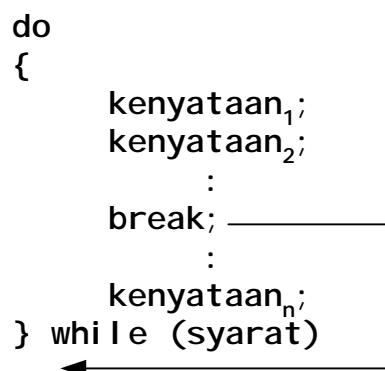
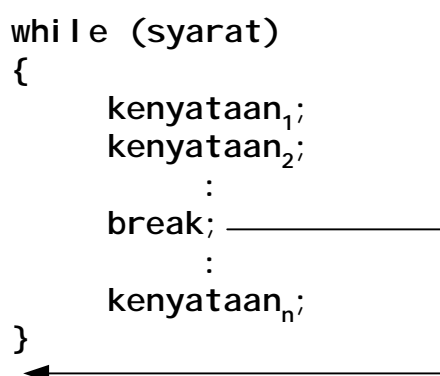
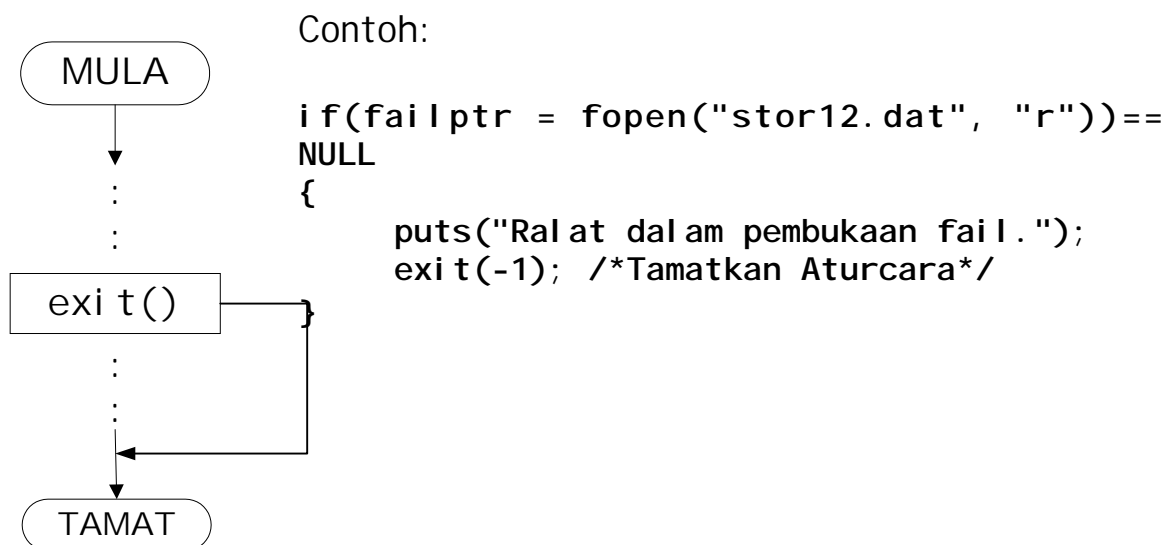
Sifir 4
4 X 1 = 4

Sifir 5
5 X 1 = 5
```

6.5 Pintas Struktur

- 3 struktur PM struktur jujukan, struktur pemilihan, struktur perulangan & gelung menentukan aliran pelaksanaan aturcara. Kadang-kadang wujud situasi keluar dari struktur tertentu. C menyediakan arahan utk tujuan ini, antaranya:

1. fungsi `exit()`
2. pernyataan `break`
3. pernyataan `continue`



6.5 Pintas Struktur (samb....)

```

while (syarat)
{
    kenyataan1;
    kenyataan2;
    :
    continue;
    :
    kenyataann;
}

do
{
    kenyataan1;
    kenyataan2;
    :
    continue;
    :
    kenyataann;
} while (syarat)

```

```

for (bhf_memul akan ; syarat ; bhg_kemaski ni)
{
    kenyataan1;
    kenyataan2;
    :
    continue;
    :
    kenyataann;
}

```

Contoh 1:

```

for (i=1; i<=3; i++)
{
    printf("%d\n", i);
    if (i==2)
        break;
    printf("Bahagi an bawah gel ung\n");
}
printf("Keluar dari gel ung");

```

Output di skrin:

```

1
Bahagi an bawah gel ung
2
Keluar dari gel ung

```

6.5 Pintas Struktur (samb....)

Contoh 2:

```
for (i=1; i<=3; i++)
{
    printf("%d\n", i);
    if (i==2)
        continue;
    printf("Bahagian bawah gelung\n");
}
printf("Keluar dari gelung");
```

Output di skrin:

```
1
Bahagian bawah gelung
2
3
Bahagian bawah gelung
Keluar dari gelung
```

Contoh 3:

```
i=0;
while (i<5)
{
    if (i<3)
    {
        i+=2;
        printf("%d\n", i);
        continue;
    }
    else{
        printf("%d\n", ++i);
        break;
    }
    printf("Bahagian bawah gelung\n");
}
printf("Keluar dari gelung");
```

Output di skrin:

```
2
4
5
Keluar dari gelung
```