



# TEKNIK MEMBINA ATUR CARA DENGAN BAHASA C

DAYANG NORHAYATI ABANG JAWAWI  
ROSBI MAMAT



## Bab 4: Fungsi Input dan Output

© Copyright Universiti Teknologi Malaysia

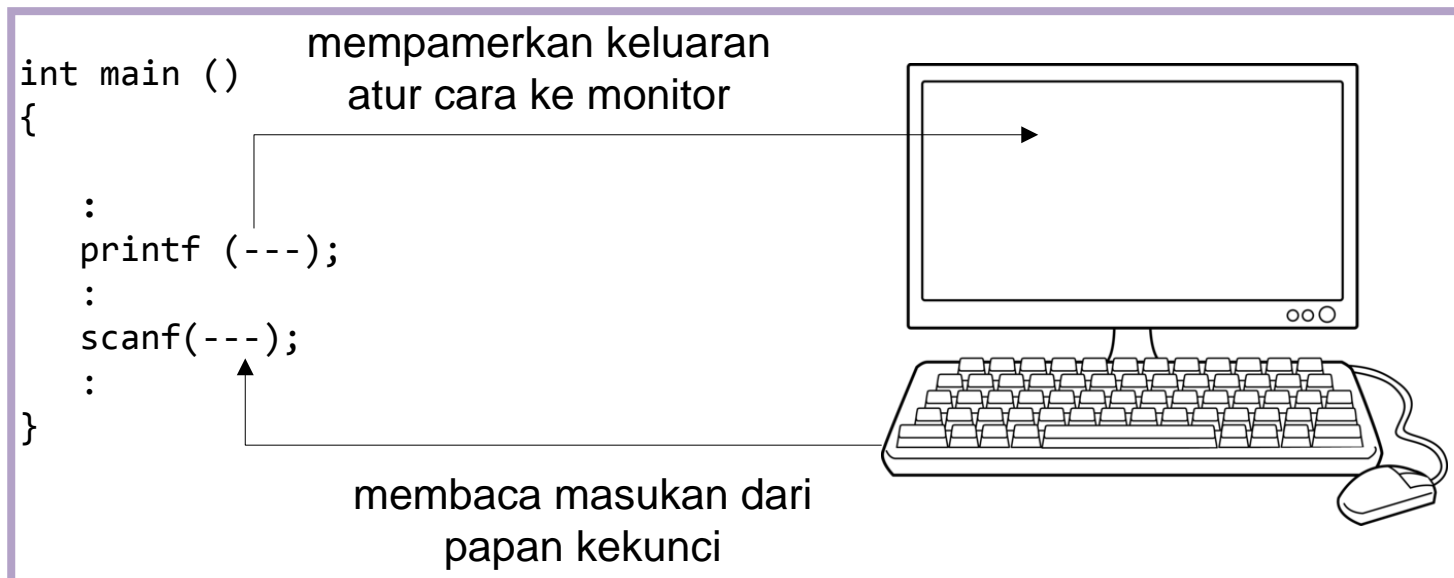
innovative • entrepreneurial • global

## 4.1 Pengenalan

- fungsi utama pada input dan output:

- printf() = pamer atau cetak mesej ke skrin  
- scanf() = baca masukan dari papan kekunci

- Penggunaan fungsi printf() dan scanf():



## 4.1 Pengenalan

- Fungsi `printf()` dan `scanf()` membolehkan atur cara komputer berhubung dengan pengguna.
- Perpustakaan C juga menyediakan fungsi-fungsi lain untuk operasi input dan output. Fungsi-fungsi ini boleh dicapai dari perpustakaan C melalui fail kepala piawai `<stdio.h>` dan `<conio.h>`.
- Antara fungsi input dan output yang terdapat di dalam dua perpustakaan diatas:

```
conio.h : clrscr()   getch()   inport()  
stdio.h : scanf()   getchar() feof()
```

## 4.2 Fungsi Output `printf()`

- ❑ Fungsi `printf()` mengambil data dari ingatan komputer dan mencetakannya sebagai output ke skrin komputer.
- ❑ Perpustakaan `stdio.h` mentakrif fungsi `printf()` dan format sintaks memanggil fungsi `printf()` ialah :

```
printf(kawalan_format_rentetan, senarai_cetakan);
```

- ❑ Jika tiada nilai perlu dicetak, maka fungsi `printf()` tidak memerlukan format pengenalan pasti pada kawalan format rentetan dan argumen kedua `printf()` iaitu senarai cetakan. Contohnya seperti kenyataan `printf()` berikut:

```
printf("Atur Cara C Pertama Saya");
```

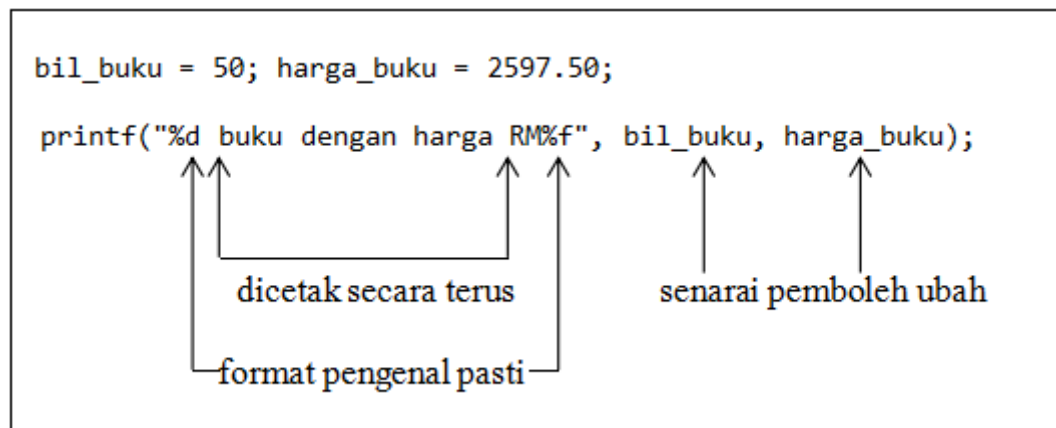
## 4.2 Fungsi Output printf()

- Kawalan format pengenalan pasti:

Tentukan format pengenalan pasti	Jenis output
%d	integer decimal
%c	satu aksara
%s	rentetan aksara
%f %lf	nombor nyata dalam decimal nombor nyata dalam decimal panjang
%e %E	nombor nyata dalam eksponen nombor nyata eksponen E dalam huruf besar
%g	guna %d atau %e mana yang lebih pendek
%u	integer desimal tak bertanda
%x	integer hexa tak bertanda
%X	integer hexa tak bertanda dalam huruf besar
%O	integer oktal tak bertanda
%%	cetak tanda %

## 4.2 Fungsi Output printf()

- Contoh kenyataan printf() mencetak bilangan dan harga buku:



output:50 buku dengan harga RM2599.15

## 4.2 Fungsi Output `printf()`

- Contoh atur cara menggunakan ungkapan aritmetik terus pada senarai cetakan.

```
1:  /* Contoh atur cara mencetak luas segiempat dan bulatan */
2:  /* Penulis : Dayang Norhayati Tarikh: 28.06.2020 */
3:
4:  #include <stdio.h>
5:
6:  int main () {
7:      const float pi = 3.1415;
8:      int panjang, lebar, luas_segi4;
9:      float jejari, luas_bulat;
10:
11:      panjang = 12;
12:      jejari = 3.5;
13:      lebar = 3;
14:
15:      printf("Luas bulatan = %f Luas segi4 = %d\n",
16:          pi * jejari * jejari, panjang * lebar );
17:      return 0;
18: }
```

## 4.2 Fungsi Output printf()

□ Pengubahsuaian dan makna:

Pengubahsuaian	Makna
<b>Digit</b>	Lebar medan minimum bagi mencetak nilai. Medan yang lebih lebar akan digunakan jika nombor yang dicetak tidak muat dalam medan yang diberi.
<b>.digit</b>	Ketepatan bagi nombor nyata. Mewakili bilangan digit yang dicetak selepas titik perpuluhan.
<b>-</b>	Nilai akan dicetak bermula dari kiri lebar medannya.
<b>L</b>	Nilai yang akan dicetak adalah jenis panjang.

© Copyright Universiti Teknologi Malaysia



## 4.2 Fungsi Output printf()

□ Contoh pengubahsuaian format pengenalan pasti:

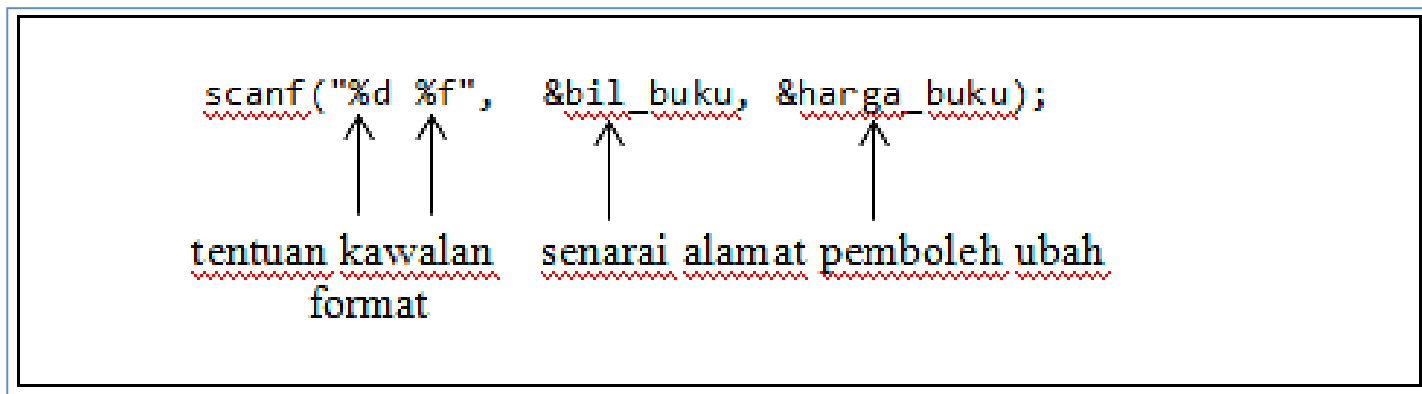
No	Kenyataan	Output di skrin
1	<code>printf("%d\n", 987);</code>	987
2	<code>printf("%2d\n", 987);</code>	987
3	<code>printf("%8d\n", 987);</code>	987
4	<code>printf("%-8d\n", 987);</code>	987
5	<code>printf("%f\n", 9876.54);</code>	9876.54
6	<code>printf("%0.2f\n", 9876.54);</code>	9876.54
7	<code>printf("%4.2f\n", 9876.54);</code>	9876.54
8	<code>printf("%3.1f\n", 9876.54);</code>	9876.5
9	<code>printf("%10.3f\n", 9876.54);</code>	9876.540
10	<code>printf("%10.3e\n", 9876.54);</code>	9.876e+03

## 4.3 Fungsi Output scanf()

- Fungsi scanf() membolehkan input gabungan nilai berangka, aksara dan rentetan dibuat dengan peranti piawai. Format memanggil fungsi scanf() ialah:

```
scanf(kawalan_format_rentetan, senarai_alamat_pemboleh_ubah);
```

- Contoh kenyataan scanf() mencetak bilangan dan harga buku:



## 4.3 Fungsi Output scanf()

- Tiga keperluan data yang dihasilkan daripada fasa analisis masalah mengira dan mencetak luas dan ukur lilit satu bulatan:

### Input

*jejari* nilai masukan jejari

### Output

*Luas* hasil proses luas bulatan

*ukurlilit* hasil proses ukur lilit bulatan

### Proses

*Luas* = 3.1415 x jejari x jejari

*ukurlilit* = 2 X 3.1415 X jejari

## 4.3 Fungsi Output scanf()

- ❑ Atur cara kira luas bulatan:

```
1: /* Contoh atur cara kira luas dan ukurlilit bulatan */
2: #include <stdio.h>
3:
4: int main () {
5:     const float PI = 3.1415;
6:     double jejari, luas, ukurlilit;
7:
8:     printf("Masukkan jejari bagi bulatan : ");
9:     scanf("%f", &jejari);
10:
11:     luas = PI*jejari*jejari;
12:     ukurlilit = 2*PI*jejari;
13:
14:     printf("Jejari bulatan ialah %0.2f\n", jejari);
15:     printf("Luas bulatan ialah %0.2f\n", luas);
16:     printf("Ukurlilit bulatan ialah %0.2f", ukurlilit);
17:
18:     return 0;
19: }
```

Output:

Masukkan jejari bagi bulatan : 5.4  
Jejari bulatan ialah 5.40  
Luas bulatan ialah 91.61  
Ukurlilit bulatan ialah 33.93

## 4.4 Fungsi-Fungsi Input dan Output Mudah

- ❑ Terdapat empat set fungsi mudah yang digunakan untuk operasi input dan output satu aksara dan satu rentetan, iaitu:

- (1) Fungsi `getchar()` dan `putchar()`
- (2) Fungsi `getch()` dan `putch()`
- (3) Fungsi `gets()` dan `puts()`
- (4) Fungsi `gets()` dan `puts()`

## 4.4.1 Fungsi `getchar()` dan `putchar()`

- ❑ Fungsi `getchar()` digunakan untuk membaca satu aksara daripada peranti input piawai seperti papan kekunci dan menunggu sehingga aksara <ENTER> dimasukkan.
- ❑ dicapai dari perpustakaan piawai <stdio.h>
- ❑ Format memanggil fungsi `getchar()`:

```
Pemboleh_ubah = getchar();
```

- ❑ contoh kenyataan memanggil fungsi `getchar()`:

```
char ch;  
  
ch = getchar();
```

## 4.4.1 Fungsi `getchar()` dan `putchar()`

- ❑ fungsi `putchar()` digunakan untuk mencetak satu aksara ke peranti output piawai seperti skrin.
- ❑ Format memanggil fungsi `putchar()`:

```
putchar(pengenal_pasti);
```

- ❑ Contoh kenyataan berikut memanggil `putchar(ch)` untuk mencetak kandungan aksara yang disimpan pada pemboleh ubah `ch`.

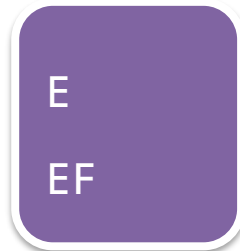
```
putchar(ch);
```

## 4.4.1 Fungsi `getchar()` dan `putchar()`

- Contoh membaca satu aksara dari papan kekunci dan mencetak aksara yang dibaca dan aksara selepasnya :

```
char huruf;  
huruf = getchar();  
putchar(huruf);      putchar(huruf+1);
```

- Paparan skrin dengan input aksara 'E' dengan fungsi `getchar()` dan `putchar()`



```
E  
EF
```

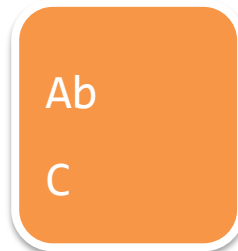


## 4.4.1 Fungsi getchar() dan putchar()

- Contoh lain:

```
char hurufA = 0x41;  
putchar(hurufA);      putchar('b');  
putchar('\n');        putchar('C');
```

- Paparan hasil fungsi dengan pelbagai cara argumen parameter:



```
Ab  
C
```

## 4.4.2 Fungsi `getch()` dan `putch()`

- ❑ Kedua-dua fungsi `getch()` dan `putch()` boleh dicapai dari perpustakaan piawai `<conio.h>`.
- ❑ fungsi `getch()` digunakan untuk membaca satu aksara daripada papan kekunci, tetapi masukan aksara dengan fungsi `getch()` tidak menunggu aksara `<ENTER>` dimasukkan dan aksara yang dimasukkan tidak akan dipamer.
- ❑ Contoh dua kenyataan panggilan fungsi `getch()` dan `putch()` adalah seperti berikut:

```
ch = getch();  
putch(ch);
```

## 4.4.2 Fungsi `getch()` dan `putch()`

- ❑ Jika contoh dari `getchar()` dan `putchar()` ditukarkan kepada `getch()` dan `putch()` ia akan memberi keluaran yang sama, hanya cara masukan berbeza.

- ❑ Contoh:

```
char huruf;  
huruf = getch();  
putch(huruf); putch(huruf+1);
```

- ❑ Paparan skrin dengan input aksara 'E' fungsi `getch()` dan `putch()`:



EF

## 4.4.3 Fungsi `getc()` dan `putc()`

- ❑ dicapai dari perpustakaan piawai `<stdio.h>`
- ❑ fungsi `getc()` dan `putc()` digunakan untuk membaca atau mencetak satu aksara dengan peranti input dan output piawai.
- ❑ Format sintaks memanggil fungsi `getc()` dan `putc()` :

```
getc(jenis_peranti)  
putc (kendalian, jenis_peranti);
```

## 4.4.3 Fungsi `getc()` dan `putc()`

- ❑ Contoh memanggil fungsi `getc()` dan `putc()`:

```
ch = getc(stdin);  
putc(ch, stdout);
```

- ❑ Contoh lain:

```
char huruf;  
huruf = getc(stdin);  
putch(huruf, stdout); putch(huruf+1, stdout);
```

## 4.4.4 Fungsi `gets()` dan `puts()`

- ❑ dicapai dari perpustakaan piawai `<stdio.h>`
- ❑ Fungsi `gets()` digunakan untuk membaca rentetan aksara dari papan kekunci sehingga aksara baris baru `'\n'` dimasukkan atau papan kekunci `<ENTER>` ditekan.
- ❑ Fungsi `puts()` digunakan untuk mencetak rentetan di skrin memasukan aksara baris baru `'\n'` pada penghujung rentetan.
- ❑ format panggilan `gets()` dan `puts()` :

```
gets(pemboleh_ubah);  
puts(pemboleh_ubah);
```

## 4.4.4 Fungsi gets () dan puts ()

□ Contoh atur cara:

```
1: #include <stdio.h>
2:
3: int main ()
4: {
5:     char bandar[30];
6:
7:     printf("Masukkan nama bandar >> ");
8:     gets(bandar);    /*baca rentetan*/
9:
10:    printf("Selamat Datang ke ");
11:    puts(bandar);    /*cetak rentetan*/
12:
13:    return 0;
14: }
```

Output:  
Masukkan nama bandar >> Johor Bahru  
Selamat Datang ke Johor Bahru

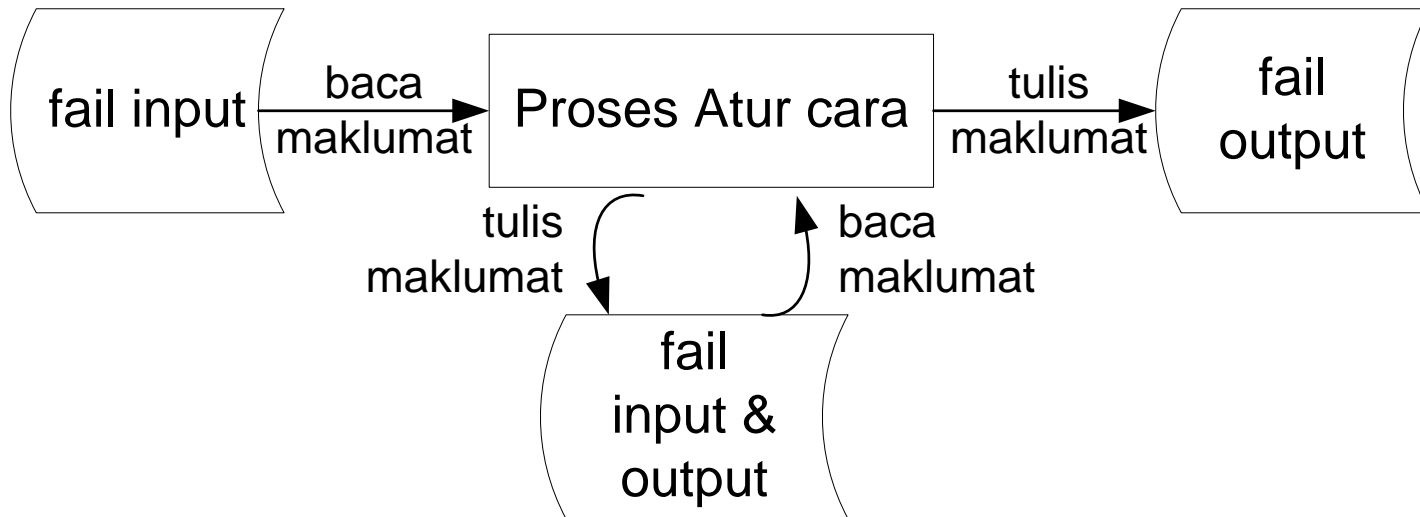
## 4.5 Fail sebagai Input dan Output

- ❑ 2 ragam operasi komputer adalah:
  - berkelompok (batch)
  - interaktif (interactive)
  
- ❑ Aturcara C yang ditulis setakat ini adalah ragam interaktif. Untuk menulis aturcara yang beroperasi secara berkelompok fail perlu digunakan.
  
- ❑ 3 operasi yg dilakukan utk memproses fail:
  1. membina fail - wajib
  2. membaca data dari fail
  3. menulis data ke fail
  
- ❑ operas 2 dan 3 dilakukan - salah satu atau kedua2



## 4.5 Fail sebagai Input dan Output

□ Proses fail data:



© Copyright Universiti Teknologi Malaysia

# 4.5 Fail sebagai Input dan Output

- ❑ Contoh Tiga operasi memproses fail data untuk membaca input fail bulat.dat dan menulis output ke fail bulat.out:

membaca data dari fail input

```
#include <stdio.h>
#define PI 3.14159

int main(void)
{
    double jejari,luas,ukurlilit;

    FILE *inp, *outp;
    inp = fopen("bulat.dat", "r");
    outp = fopen("bulat.out", "w");

    fscanf(inp,"%lf", &jejari);

    luas = PI*jejari*jejari; ukurlilit = 2*PI*jejari;

    fprintf(outp,"Jejari bulatan ialah %.2f\n", jejari);
    fprintf(outp,"Luas bulatan ialah %.2f\n", luas);
    fprintf(outp,"Ukurlilit bulatan ialah %.2f", ukurlilit);

    fclose(inp);
    fclose(outp);

    return 0;
}
```

membina fail

- Isi fail input bulat.dat:

5.0

-Isi fail output bulat.out:

Jejari bulatan ialah 5.00  
Luas bulatan ialah 78.54  
Ukurlilit bulatan ialah 31.42

menulis data ke fail output

## 4.5.1 Membina Fail

- ❑ Pengisytiharan fail menggunakan pemboleh ubah penuding fail, format pengisytiharan adalah seperti berikut:

```
FILE *pembpenudingfail;
```

- ❑ Untuk membolehkan fail digunakan, fail tersebut mesti dibuka terlebih dahulu. Format membuka fail:

```
pembpenudingfail = fopen("namafail", "jenismodcapaian");
```

↑  
pembolehubah penuding  
fail yang disytiharkan  
sebelum

↑  
nama fail  
yang sah

↑  
jenis mod capaian menentukan  
samada fail tersebut adalah  
fail input, output atau keduanya

## 4.5.1 Membina Fail

- ❑ Mod capaian fail:

Mod	Jika fail wujud	Jika fail tidak wujud
“r”	buka fail untuk dibaca	Ralat
“w”	buka fail untuk ditulis	Bina fail baru
“a”	buka fail untuk ditambah di hujung fail	Bina fail baru
“r+”	buka fail untuk dibaca dan ditulis	Ralat
“w+”	buka fail untuk dibaca dan ditulis	Bina fail baru
“a+”	buka fail untuk dibaca dan ditambah di hujung fail	Bina fail baru

- ❑ Jika fail tidak digunakan lagi fail perlu ditutup. Format utk menutup fail:

```
fclose(pembpenudingfail);
```

## 4.5.2 Membaca dan Menulis Fail

- Banyak cara boleh dilakukan untuk membaca dan menulis data ke atau dari fail, Ini bergantung kepada cara data hendak diproses:
  - sama ada aksara demi aksara pada satu masa
  - satu baris data (satu rentetan aksara) pada satu masa
- format sintaks panggilan:

`fscanf()` dan `fprintf()` untuk memproses pelbagai jenis data pada satu masa. Format panggilan dengan fail data adalah:

```
fprintf(pembpenudingfail, kawalan_format_rentetan, senarai_cetakan);
```

```
fscanf(pembpenudingfail, kawalan_format_rentetan, senaraAlamatPemb);
```

`fgetc()` dan `fputc()` untuk memproses satu aksara demi satu aksara pada satu masa dengan data berjenis char. Format panggilan dengan fail data menjadikan pemboleh ubah penuding sebagai jenis peranti:

```
fgetc(jenis_peranti);
```

```
fputc(kendalian, jenis_peranti);
```

## 4.5.2 Membaca dan Menulis Fail

- format sintaks panggilan (sambungan):

`fgets()` dan `fputs()` untuk memproses satu baris data dalam bentuk rentetan aksara pada satu masa.

Format panggilan dengan fail data adalah:

```
fgets(pemboleh_ubah, saiz, pembpenudingfail);
```

```
fputs(kendalian, pembpenudingfail);
```

- Contoh aturcara menggunakan fail, membaca fail `stor12.dat` dan menulis output di fail yg mana nama fail dimasukkan oleh pengguna melalui papan kekunci.
  - Isi fail `stor12.dat` adalah seperti berikut:

```
MAKMAL KAWALAN  
ABC  
10 45.20  
23 17.39  
4 100.14
```

## 4.5.2 Membaca dan Menulis Fail

- Hasil dari aturcara (slide seterusnya), fail output yg terhasil.

Laporan Bulanan Stor MAKMAL KAWALAN

Kod Komponen	Bil	Harga
A	10	45.20
B	23	17.39
C	4	100.14

Nilai stok 1252.53

Purata harga satu komponen 54.24

## 4.5.2 Membaca dan Menulis Fail

- aturcara Mencetak laporan bulanan stor.

```
1:  #include <stdio.h>
2:  int main(void)
3:  {
4:      char namastor[30], failLaporan[12];
5:      char kompA, kompB, kompC;
6:      int bilkompA, bilkompB, bilkompC;
7:      float harga1kompA, harga1kompB, harga1kompC,
8:          nilaistok, purataharga;
9:      FILE *stor, *laporan;
10:
11:     stor = fopen("stor12.dat", "r");
12:     printf("Masukkan nama fail laporan > ");
13:     gets(failLaporan);
14:     laporan = fopen(failLaporan, "w");
15:
16:     fgets(namastor, 30, stor);
17:     kompA = fgetc(stor); kompB = fgetc(stor); kompC = fgetc(stor);
18:     fscanf(stor, "%d %f", &bilkompA, &harga1kompA);
19:     fscanf(stor, "%d %f", &bilkompB, &harga1kompB);
```



## 4.5.2 Membaca dan Menulis Fail

- aturcara Mencetak laporan bulanan stor( sambungan).

```
20: fscanf(stor,"%d %f", &bilkompC, &harga1kompC);
21:
22: nilaistok = (bilkompA*harga1kompA) + (bilkompB*harga1kompB)+
23:           (bilkompC*harga1kompC);
24: purataharga= (harga1kompA + harga1kompB +harga1kompC)/3;
25:
26: fprintf(laporan,"Laporan Bulanan Stor ");
27: fputs(namastor, laporan);
28: fprintf(laporan,"\nKod Komponen\tBil\t\tHarga\n");
29: fprintf(laporan,"\t%c\t\t\t%d\t\t%0.2f\n", kompA, bilkompA,
30:           harga1kompA);
31: fprintf(laporan,"\t%c\t\t\t%d\t\t%0.2f\n", kompB, bilkompB,
32:           harga1kompB);
33: fprintf(laporan,"\t%c\t\t\t%d\t\t%0.2f\n", kompC, bilkompC,
34:           harga1kompC);
35: fprintf(laporan,"\nNilai stok %0.2f ", nilaistok);
36: fprintf(laporan,"\nPurata harga satu komponen %0.2f",purataharga);
37:
38: fclose(stor);
39: fclose(laporan);
40:
41: return 0;
42: }
```

Contoh paparan interaktif :  
Masukkan nama fail laporan > laporan032016.txt

© Copyright Universiti Teknologi Malaysia

innovative • entrepreneurial • global