

Pointer

- **Pointer** (variabel penunjuk) adalah suatu variabel yang berisi alamat memori dari suatu variabel lain. Alamat ini merupakan lokasi dari obyek lain (biasanya variabel lain) di dalam memori.
- Operator Pointer ada dua, yaitu :
 - Operator &
Operator & bersifat unary (hanya memerlukan satu operand saja)
Operator & menghasilkan alamat dari operandnya.
 - Operator *
Operator * bersifat unary (hanya memerlukan satu operand saja).
Operator * menghasilkan nilai yang berada pada sebuah alamat.

- **Deklarasi Pointer**

Tipe_data *nama_pointer;

Contoh :

```
int *px;  
char *sh;
```

Contoh Program :

```
#include "stdio.h"  
#include "conio.h"  
void main()  
{ int x, y;      /* x dan y bertipe int */  
  int *px;      /* px pointer yang menunjuk objek */  
  clrscr();  
  x = 87;  
  px = &x;     /* px berisi alamat dari x */  
  y = *px;     /* y berisi nilai yang ditunjuk px */  
  printf("Alamat x = %p\n", &x);  
  printf("Isi px = %p\n", px);  
  printf("Isi x = %i\n", x);  
  printf("Nilai yang ditunjuk oleh px = %i\n", *px);  
  printf("Nilai y = %i\n", y);  
  getch();  
}
```



Operasi File

- **File** adalah sebuah organisasi dari sejumlah record. Masing-masing record bisa terdiri dari satu atau beberapa field. Setiap field terdiri dari satu atau beberapa byte.
- **Membuka File** menggunakan fungsi **fopen()**
 - File dapat berupa file biner atau file teks.
 - File biner adalah file yang pola penyimpanan di dalam disk dalam bentuk biner, yaitu seperti bentuk pada memori (RAM) computer
 - File teks adalah file yang pola penyimpanan datanya dalam bentuk karakter.
 - Penambahan yang perlu dilakukan untuk menentukan mode teks atau biner adalah “t” untuk file teks dan “b” untuk file biner
 - Prototype fungsi fopen() ada di header fungsi “**stdio.h**”
 - **Bentuk umum :**
file *fopen(char *namafile, char *mode);
- **Keterangan :**
 - **namafile** adalah nama dari file yang akan dibuka/diaktifkan.
 - **mode** adalah jenis operasi file yang akan dilakukan terhadap file.
- **Jenis-jenis operasi file :**
 - r : menandakan file hanya dapat dibaca (file harus sudah ada)
 - w : menyatakan file baru akan dibuat/diciptakan (file yang sudah ada akan dihapus)
 - a : untuk membuka file yang sudah ada dan akan dilakukan proses penambahan data (jika file belum ada, otomatis akan dibuat)
 - r+ : untuk membuka file yang sudah ada dan akan dilakukan proses pembacaan dan penulisan.
 - w+ : untuk membuka file dengan tujuan untuk pembacaan atau penulisan. Jika file sudah ada, isinya akan dihapus.
 - a+ : untuk membuka file, dengan operasi yang akan dilakukan berupa perekaman maupun pembacaan. Jika file sudah ada, isinya akan dihapus.

Contoh :

```
pf = fopen("COBA.TXT", "w");
```

- **Menutup File** menggunakan fungsi **fclose()**.
 - Prototype fungsi fclose() ada di header file “**stdio.h**”
 - **Bentuk Umum :**
int fclose(FILE *pf);
atau
int fcloseall(void);
- **Melaksanakan Proses File**
- **Menulis Karakter**
bentuk yang digunakan adalah :
putc(int ch, file *fp)
fp adalah pointer file yang dihasilkan oleh fopen()
ch adalah karakter yang akan ditulis



Contoh Program :

```
#include "stdio.h"
#include "conio.h"
#define CTRL_Z 26
void main()
{ file *pf; /* pointer ke file */
char kar;
if((pf = fopen("COBA.TXT", "w")) == NULL) /* ciptakan file */
{ cputs("File tak dapat diciptakan !\n");
exit(1); /* selesai */
}
while((kar=getche()) != CTRL_Z)
putc(kar, pf); /* tulis ke file */
fclose(pf); /* tutup file */
}
```

- **Membaca Karakter**

fungsi yang digunakan adalah :

```
getc(file *fp);
```

fp adalah pointer file yang dihasilkan oleh fopen()
Fungsi feof(), digunakan untuk mendeteksi akhir file
Pada saat membaca data foef(file *fp)

Contoh Program :

```
#include "stdio.h"
#include "conio.h"
void main()
{ file *pf; /* pointer ke file */
char kar;
clrscr();
if((pf = fopen("COBA.TXT", "r")) == NULL) /* buka file */
{ cputs("File tak dapat dibuka !\n");
exit(1); /* selesai */
}
while((kar=getc(pf)) != EOF)
putch(kar); /* tampilkan ke layar */
fclose(pf); /* tutup file */
}
```

- **Membaca dan Menulis String**

Fungsi untuk membaca dan menulis string adalah : fgets() dan fputs()

Bentuk Umum :

```
fgets(char *str, int p, file *fp)
fputs(char *str, file *fp)
```

- **Membaca dan Menulis Blok Data**

Fungsi untuk membaca dan menulis blok data adalah : fread() dan fwrite()

Bentuk umum :

```
fread(void *buffer, int b_byte, int c, file *fp);
```



```
fwrite(void *buffer, int b_byte, int c, file *fp);
```

Keterangan :

buffer adalah pointer ke sebuah area di memori yang menampung data yang akan dibaca dari file.

b_byte adalah banyaknya byte yang akan dibaca atau ditulis ke file

c adalah banyaknya item dibaca/ditulis.

Contoh Program :

```
#include "stdio.h"
#include "conio.h"
void main()
{ file *f_struktur;
char jawaban;
struct data_pustaka
{ char judul[26];
char pengarang[20];
int jumlah;
} buku; /* variabel buku bertipe struktur */
/* buka file */
if((f_struktur = fopen("DAFBUKU.DAT", "wb")) == NULL)/* buka file */
{ cputs("File tak dapat diciptakan !\r\n");
exit(1); /* selesai */
}
do
{ clrscr();
cputs("Judul Buku : ");
gets(buku.judul);
cputs("Nama Pengarang : ");
gets(buku.pengarang);
cputs("Jumlah buku : ");
scanf("%i", $buku.jumlah);
fflush(stdin); /* Hapus isi penampung keyboard */
/*Rekam sebuah data bertipe struktur */
fwrite(&buku, sizeof(buku), 1, f_struktur);
cputs("\r\nMau merekam data lagi (Y/T) ?");
jawaban = getch();
}
while(jawaban == 'Y' || jawaban == 'y');
fclose(f_struktur); /* tutup file */
}
```

- **Membaca dan Menulis File yang Terformat**

Jika diinginkan, data bilangan dapat disimpan ke dalam file dalam keadaan terformat.

Fungsi yang digunakan adalah :

```
fprintf(ptr_file, "string control", daftar argument);
fscanf(pts_file, "string control", daftar argument);
```



Contoh Program :

```
#include "stdio.h:
#include "conio.h"
void main()
{ FILE *pformat;
char jawaban;
struct
{ int x;
int y;
} koordinat;
/* Buka dan ciptakan file. Periksa kalau gagal dibuka */
if((pformat = fopen("KOORDINAT.TXT", "w")) == NULL) /* buka file */
{ cputs("File tak dapat dibuka !\r\n");
exit(1); /* selesai */
}
do
{ clrscr();
cputs("Masukkan data koordinat (bilangan integer)\r\n");
cputs("Format : posisi x posisi y\r\n");
cputs("Contoh : 20 30 [ENTER]\r\n");
scanf("%i %i, &koordinat.x, &koordinat.y);
fflush(stdin);
/* Rekam ke file */
fprintf(pformat, "%5i %5i\n", koordinat.x, koordinat.y);
cputs("\r\nMenyimpan data lagi (Y/T) ??");
jawaban = getch();
}
while(jawaban == 'y' || jawaban == 'Y');
fclose(pformat);
getch();
}
```

Contoh Program 2 :

```
#include <stdio.h>
FILE *in;
void BACA( int[ ] );
void CETAK( int[ ] );
void main() {
int tabel[26] = {0};
BACA(tabel);
CETAK(tabel);
}
void BACA ( int huruf[] ) {
char c;
if (( in = fopen("data.txt" , "r")) == NULL)
printf ("File tidak bisa dibaca\n");
else
while ( (ch = fgetc(in)) != EOF ) {
c = ( (( c >= 97) || ( c <= 122)) ? c - 32 : c );
if ( ( c >= 65) || ( c <= 90) )
++huruf [ c - 65 ];
```



```
}  
fclose(in);  
}  
void CETAK ( int huruf[] ) {  
int counter;  
for ( counter = 0 ; counter <= 25 ; counter++ )  
printf ("\n%c%5d", counter + 65, huruf[counter] );  
}
```

