PERCOBAAN 9

* 1. JUDUL: VARIABEL POINTER & PENGGUNAANNYA DALAM FUNGSI
  2. TUJUAN: Mempelajari penggunaan variable pointer yang dipadukan dengan variable bertipe integer, array dan string serta penggunaan pointer dalam suatu fungsi, baik sebagai penunjuk fungsi atau sebagai argument fungsi. Dipelajari juga pointer yang menunjuk pointer lainnya.
  3. TEORI: Suatu pointer adalah suatu variable yang menyimpan alamat dari suatu variable yang ditunjuk oleh pointer yang bersangkutan. Karena data yang ditunjuk oleh pointer bervariasi, maka tipe dari pointer harus disesuikan dengan tipe data yang ditunjuk, untuk memperkirakan ukuran data yang ditunjuk, sehingga alokasi memori terjaga dengan baik. Pointer berguna untuk mentransfer data yang mempunyai kapasitas besar, melalui suatu fungsi. Pointersangat erat kaitannya dengan array, sehingga variable pointer dapat menggantikan fungsi dari variable array.
  4. PROGRAM PERCOBAAN
     1. Mengakses isi suatu variabel melalui pointer. Untuk membedakan antara pointer dengan isi dari pointer maka dibedakan dengan penggunaan tanda ‘&’ (ampersand) dan ‘\*’ (asterisk), tanda ‘&’ digunakan pada variable yang bukan pointer, bila suatu variable bukan pointer diawali dengan tanda & didepannya maka akan didapatkan alamat dari variable tersebut. Demikian sebaliknya bila variable pointer diawali dengan tanda ‘\*’, maka akan didapatkan nilai isi dari variable yang ditunjuk oleh pointer tersebut, dan bukan didapatkan alamatnya.

/\* Nama File : POINTER1.C \*/

void main()

{

int y, x = 2002; /\* x & y bertipe int \*/

int \*px; /\* var pointer menunjuk data tipe int \*/

px = &x; y = \*px;

/\* px diisi alamat dari variabel x \*/

/\* y diisi nilai yang ditunjuk ole- px \*/

printf("Alamat x = %p\n", &x); printf("Isi px = %p\n", px); printf("Isi x = %d\n", x); printf("Nilai \*px = %d\n", \*px); printf("Nilai y = %d\n", y);

getc-();

}

* + 1. Mengamati persamaan penggunaan variable index pada array dan variable index pada pointer, untuk menunjuk suatu nilai data didalam suatu variable array.

/\* Nama File : STRING2.C \*/

void main()

{

int nilai[10]={86,75,98,66,56,76,80,95,70,60};

int index, \*ip;

printf("Mencetak menggunakan array\n"); printf("Daftar nilai siswa\n\n"); for(index=0; indexc10; index++)

printf("%3d",nilai[index]);

puts("\n");

printf("Mencetak menggunakan pointer dan index\n"); printf("Daftar nilai siswa\n\n");

for(index=0; indexc10; index++) printf("%3d",\*(nilai+index));

puts("\n");

printf("Mencetak menggunakan pointer\n"); printf("Daftar nilai siswa\n\n"); ip=&nilai;

for(index=0; indexc10; index++) printf("%3d",\*ip++);

getc-();

}

* + 1. Pointer yang menunjuk ke pointer yang lain.

/\* Program : POINTER3.c \*/

void main()

{

int a, \*b, \*\*c;

a = 1975;

b = &a; c = &b;

printf("Nilai a = %d atau %d atau %d\n", a, \*b, \*\*c); printf("b = %p = alamat a di memori\n", b);

printf("c = %p = alamat b di memori\n", c); printf("alamat c di memori = %p\n", &c);

getc-();

}

* + 1. Fungsi dengan argumen berupa pointer.

/\* Nama File : POINTER4.C \*/

void naikkan\_nilai(int \*x, int \*y); main()

{

int a = 3, b = 7;

printf("SEMULA : a = %d b = %d\n", a, b); naikkan\_nilai(&a, &b);

printf("SEKARANG : a = %d b = %d\n", a, b); getc-();

}

void naikkan\_nilai(int \*x, int \*y)

{

\*x = \*x + 2;

\*y = \*y + 3;

}

* + 1. Suatu fungsi dapat mempunyai nilai balik bertipe pointer, sehingga dapat digunakan untuk mengembalikan nilai yang panjang seperti halnya nilai suatu string.

/\* Nama File : POINTER5.C \*/ #include cstdio.-S

c-ar \*nama\_bulan(int n)

{

static c-ar \*bulan[] = { "Kode bulan sala-",

"Januari","Februari","Maret","April","Mei","Juni", "Juli","Agustus","September","Oktober","November", "Desember"

};

return ((nc1||nS12) ? bulan[0] : bulan[n]);

}

void main()

{

int bl;

printf("Masukkan kode Bulan [1..12] : "); scanf("%d", &bl);

printf("Bulan ke-%d adala- %s\n", bl, nama\_bulan(bl));

getc-();

}

* + 1. Penggunaan fungsi \*strlwr(str) yang digunakan untuk mengubah dari string huruf BESAR menjadi huruf kecil, dan fungsi strupr() yang digunakan untuk sebaliknya. Fungsi calloc() digunakan untuk memesan sejumlah memori (dinamis) yang akan digunakan untuk menampung hasil konversi, karena variable \*ptrx tidak diketahui panjang data yang ditunjuk olehnya.

/\* Nama File : POINTER6.C \*/ #include cstdlib.-S #includecstring.-S

void main()

{

c-ar str1[80], str2[80], \*ptrx;

strcpy(str1,"INI ADALAH HURUF BESAR SEMUA!!!");

strcpy(str2,"ini adala- -uruf kecil semua!!!"); ptrx = NULL;

cputs("Isi str1 = "); puts(str1); cputs("Isi str2 = "); puts(str2); cputs("Isi ptrx = "); puts(ptrx);

ptrx = (c-ar \*) calloc(80, sizeof(c-ar)); ptrx = strlwr(str1);

puts("\nSeletela- 'str1' diproses dengan strlwr()\n"); cputs("Isi str1 = "); puts(str1);

cputs("Isi ptrx = "); puts(ptrx);

ptrx = strupr(str2);

puts("\nSeletela- 'str2' diproses dengan strupr()\n"); cputs("Isi str2 = "); puts(str2);

cputs("Isi ptrx = "); puts(ptrx);

getc-();

}

* 1. TUGAS-TUGAS (Dikumpulkan 1 minggu setelah praktikum dilaksanakan)
     1. Buatlah program untuk menyimpan bilangan pecahan acak sebanyak 12 bilangan, yang disimpan dalam suatu array. Urutkan bilangan-bilangan tersebut dan tampilkan sebelum dan sesudah diurutkan menggunakan index array dan pointer.
     2. Buatlah program yang menyimpan data karakter ‘X’ pada suatu array

berdimensi 12 x 12, kemudian tampilkan menggunakan index pointer.