POINTER I

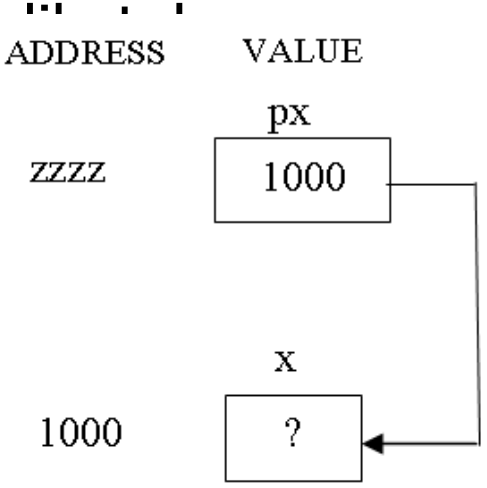
Oleh :
Mike Yuliana
PENS

SASARAN

- Menjelaskan tentang konsep dari variabel pointer
- Menjelaskan tentang pointer array
- Menjelaskan tentang pointer string

Konsep Dasar Pointer

- Pointer adalah variabel yang **khusus digunakan untuk menampung address.**
- Pointer sering dikatakan sebagai variabel yang menunjuk ke obyek/variabel lain.
- Kenyataan sebenarnya, variabel pointer berisi alamat dari suatu obyek lain (yaitu obyek yang ditunjuk oleh pointer).
- Misalnya:
  - **px** adalah variabel pointer
  - **x** adalah variabel yang ditunjuk oleh **px**
  - Kalau **x** berada pada alamat memori 1000, maka **px** akan berisi 1000



Deklarasi Variabel Pointer

- Bentuk umum:

```
tipe_data *nama_variabel;
```

- Contoh:

```
int *px;
```

menyatakan bahwa `px` adalah variabel pointer yang menunjuk ke suatu data tertentu yang bertipe *int*

- Mengatur pointer agar menunjuk ke variabel lain:

```
px = &x;
```

- Mengakses isi suatu variabel melalui pointer:

```
y = *px;
```

Contoh #1

```
Alamat x = 0012FF78
Isi px   = 0012FF78
Isi x    = 87
Nilai px = 87
Nilai y  = 87
```

```
#include <stdio.h>
main()
{
  int y, x = 87;
  int *px;
  px = &x;
  y = *px;
  printf("Alamat x     = %p\n", &x);
  printf("Isi px       = %p\n", px);
  printf("Isi x        = %d\n", x);
  printf("Nilai yang ditunjuk oleh px =
  %d\n", *px);
  printf("Nilai y      = %d\n", y);
}
```

Analisa

- Pada program di atas, dua pernyataan

```
px = &x;
y = *px;
```

- sebenarnya dapat digantikan dengan sebuah pernyataan berupa

```
y = x;
```

- Seandainya pada program di atas pernyataan

```
px = &x;
```

diganti dengan

```
px = x;
```

- kemudian

```
y = *px;
```

- mengakibatkan komputer tidak dapat dikendalikan *(hang)*.

Contoh #2

```
#include <stdio.h>
main()
{
   int *pu;
   int nu;
   int u = 1234;
   pu = &u;
   nu = *pu;
   printf("Alamat dari u = %p\n", &u);
   printf("Isi pu        = %p\n", pu);
   printf("Isi u         = %d\n", u);
   printf("Nilai yang ditunjuk pu = %d\n", *pu);
   printf("Nilai nu      = %d\n", nu);
}
```

```
Alamat dari u = 0028ff14
Isi pu        = 0028ff14
Isi u         = 1234
Nilai yang ditunjuk pu = 1234
Nilai nu      = 1234
```

Mengubah isi variabel pointer - Contoh #3 -

```
#include <stdio.h>
main()
{
  int d = 100;
  int *pd;
  printf("Isi d mula-mula = %d\n",
  d);
  pd = &d;
  *pd = *pd + 10;
  printf("Isi d sekarang  = %d\n",
  d);
}
```

Latihan

Untuk setiap program di bawah ini,

- **gambarkan ilustrasi alokasi memori dari setiap baris pernyataan yang diproses**
- **perkirakan hasil eksekusinya**

```
1. main(){
      int z = 20, s = 30, *pz, *ps;
      pz = &z;
      ps = &s;
      *pz += *ps;
      printf("z = %d\n", z);
      printf("s = %d\n", s);
   }
```

Latihan

2.
```
main() {
        int  count = 10, *temp,sum=7;
        temp = &count;
        *temp = 32;
        temp = &sum;
        *temp = count;
        sum = *temp * 4;
        printf("count = %d, *temp = %d, sum = %d\n",
   count,*temp,        sum );
```

3.
```
main(){
        int i1, i2, *p1, *p2;
        i1 = 9;
        p1 = &i1;
        i2 = *p1 / 2 - 2 * 3;
        p2 = p1;
        printf("i1=%d,i2=%d,*p1=%d,*p2=%d\n",i1,i2,*p1,*p2);
   }
```

Pointer to Array

- Hubungan antara pointer dan array pada C sangatlah erat.
- Ingat bahwa sesungguhnya array secara internal akan diterjemahkan dalam bentuk pointer

array yang dituliskan tanpa kurung sikunya ⇔ alamat dari elemen pertama array tsb.

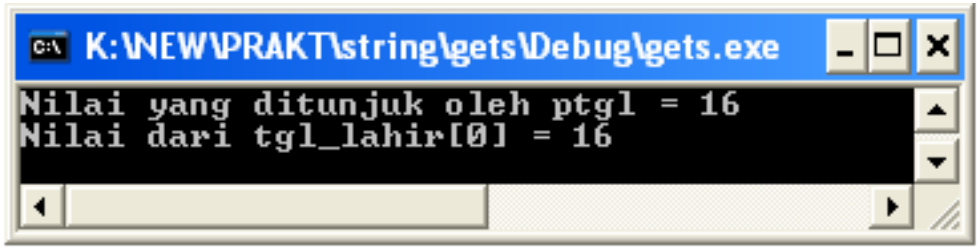
Pointer dan Array

- Deklarasi variabel:
```
int tgl_lahir[3] = { 01, 09, 64 };
int *ptgl;
```
- Kemudian diberi instruksi:
```
ptgl = &tgl_lahir[0];
```
→ maka `ptgl` akan berisi alamat dari elemen array **tgl_lahir** yang berindeks nol.
- Instruksi di atas bisa juga ditulis menjadi:
```
ptgl = tgl_lahir;
```
→ nama array tanpa tanda kurung menyatakan alamat awal dari array.

Pointer to Array

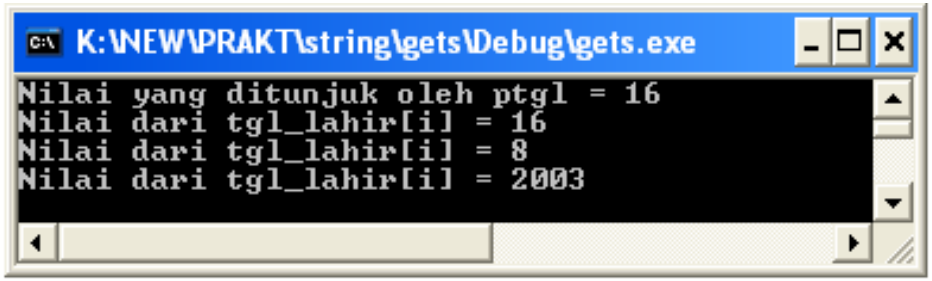
```
#include <stdio.h>

main()
{
   int tgl_lahir[] = {16, 8, 2003};
   int *ptgl;

   ptgl = tgl_lahir;
   printf("Nilai yang ditunjuk oleh ptgl = %d\n",*ptgl);
   printf("Nilai dari tgl_lahir[0] = %d\n", tgl_lahir[0]);
}
```

ptgl = tgl_lahir;
artinya sama dengan
ptgl = &tgl_lahir[0];
→pointer to array of integer



Pointer to Array

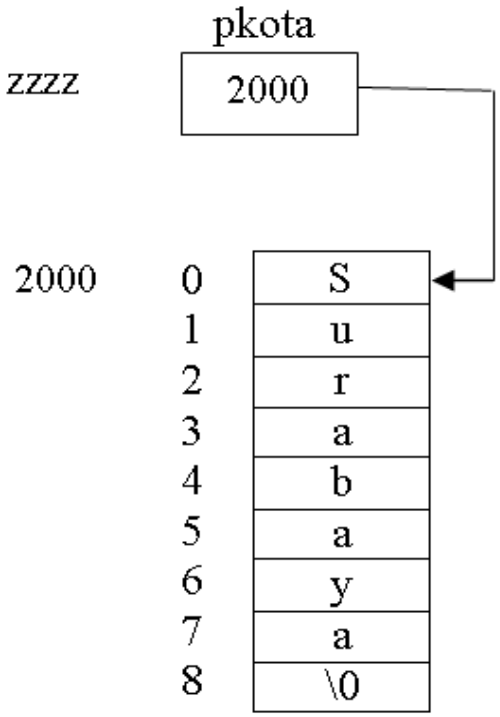
```
#include <stdio.h>
main()
{
    int tgl_lahir[] = {16, 8, 2003};
    int *ptgl, i;

    ptgl = tgl_lahir;
    printf("Nilai yang ditunjuk oleh ptgl = %d\n", *ptgl);
    for (i=0; i<3; i++)
         printf("Nilai dari tgl_lahir[i] = %d\n",*(ptgl+i));
}
```



Tambahkan nilai ptgl(berisi alamat) dengan i kali ukuran dari obyek yang ditunjuk oleh ptgl

Pointer dan String

- Pointer juga dapat digunakan untuk mendeklarasikan variabel string.
- Contoh:
  `char *pkota = "surabaya";`
- Hampir sama dengan:
  `char kota[] = "surabaya";`
- Tetapi sebenarnya tidak tepat sama.
  - pkota adalah pointer (menyatakan alamat) yang menunjuk ke string "SEMARANG",
  - kota adalah array (array menyatakan alamat yang konstan, tak dapat diubah).



Contoh #4

```
/* Program : ptr4.c */
#include <stdio.h>
main()
{
   // pkota menunjuk konstanta string "surabaya"
   char *pkota = "surabaya";

   printf("String yang ditunjuk oleh pkota = ");
   puts(pkota);   // printf("%s\n", pkota)
}
```

Hampir sama dengan sbb :

```
char *pkota;
char data [] = "surabaya";
pkota = data;
printf ("%c\n",*pkota);
```

LATIHAN

1. Buat program untuk menampilkan sebaris string seperti contoh berikut ;
   “Selamat Pagi“
   menggunakan variable pointer *(pointer to string)*, kemudian tampilkan karakter per karakter.

2. Buat potongan program untuk mencetak kalimat diatas menjadi :
   “amat Pagi “
   dengan menggunakan variabel pointer .

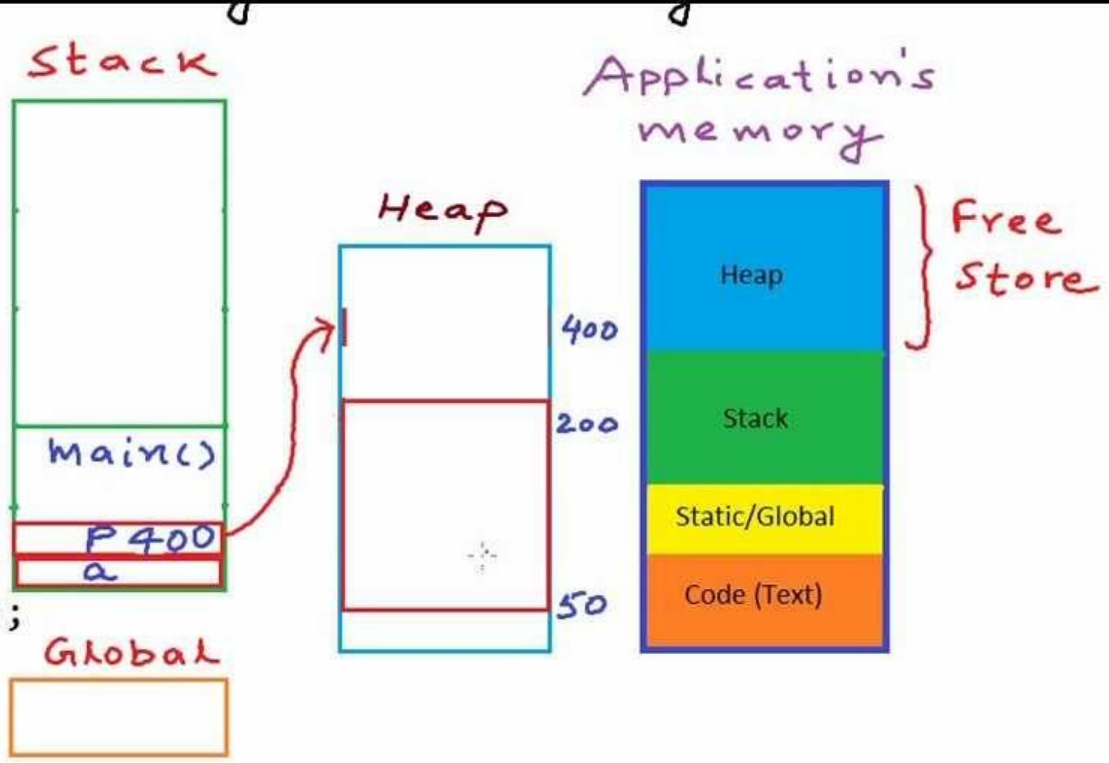
LATIHAN

3. Buat program untuk menghitung total nilai dari int x[7]={10,75,50,12,15,55,25} dengan menggunakan pointer

4. Perkirakan hasil eksekusi dari program dibawah ini

```
main()
{
    int x[5], *p;
    p=x;
    *p=5;
    *(p+1)=x[0];
    *(x+2)=*p+2;
    *(p+3)=*(p+1)-3;
    *(x+4)=*(x+2);
    printf("%d %d %d %d\n", x[0], x[1], x[2],
    x[3], x[4]);
}
```

```c
#include<stdio.h>
#include<stdlib.h>
int main()
{
  int a; // goes on stack
  int *p;
  p = (int*)malloc(sizeof(int));
  *p = 10;
  free(p);
 p = (int*)malloc(20*sizeof(int));
}
```



```
main()
{
        int x[5], *p;
        p=x;
        *p=5;
        *(p+1)=x[0];
        *(x+2)=*p+2;
        *(p+3)=*(p+1)-3;
        *(x+4)=*(x+2);
        printf("%d %d %d %d %d\n", x[0], x[1], x[2], x[3], x[4]);
        printf("%p %p %p %p %p\n", p, (p+1), (p+2), (p+3),(p+4));
        printf("%p %p %p %p %p\n", &x, &x[1], &x[2], &x[3],&x[4]);
}
```