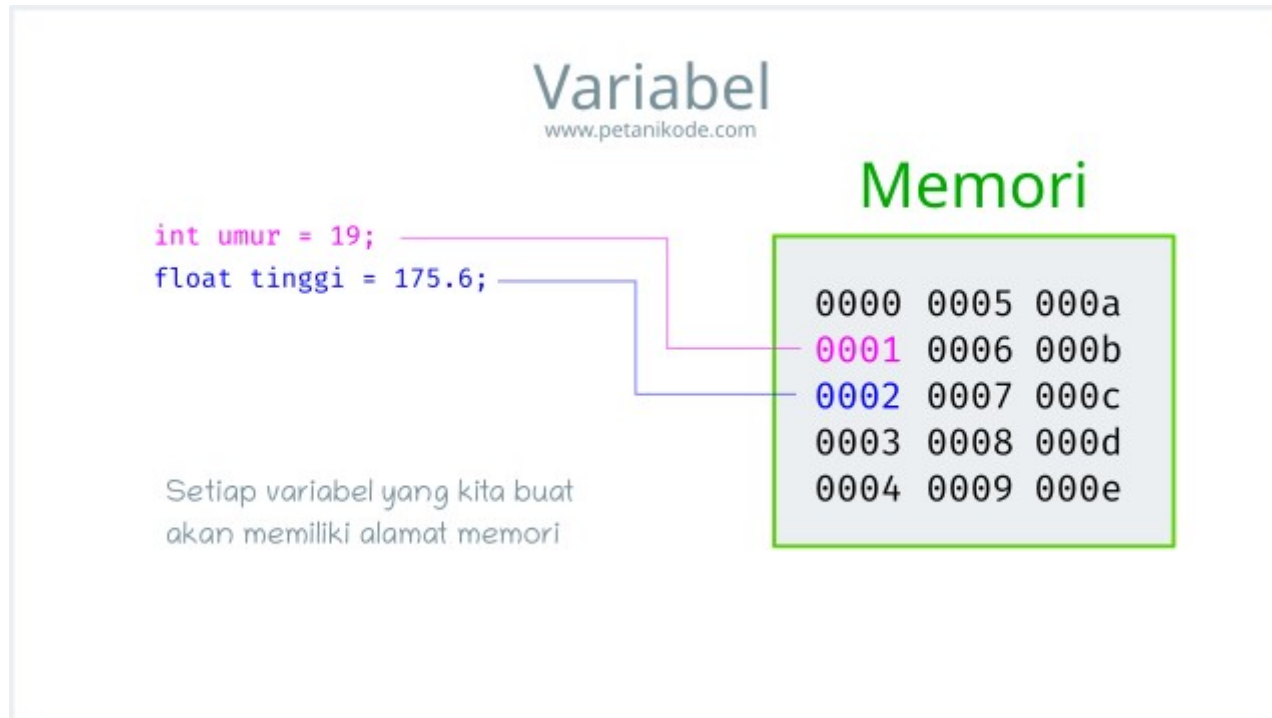


Pointer & Memori

Kharisma Monika

Informatika

Institut Teknologi Telkom Surabaya



setiap kita membuat variabel pasti akan punya alamat memori di RAM.

untuk melihat alamat memori yang digunakan pada variabel, kita bisa pakai **simbol & (emphasis)**.

%x : bilangan hexadecimal

```
#include <stdio.h>

void main () {

    int a;
    char b[10];

    printf("Alamat memori variabel a: %x\n", &a);
    printf("Alamat memori variabel b: %x\n", &b);

}
```

Pointer adalah sebuah variabel yang berisi **alamat memori** dari variabel yang lain.

Fungsi : mengakses data yang ada di suatu alamat memori.

Membuat Pointer

Pointer dibuat dengan menambahkan **simbol**
***** (**asterik**) di depan namanya, kemudian
diisi dengan alamat memori yang akan
digunakan sebagai referensi

```
int *pointer1 = 00001;
```

Pointer

www.petanikode.com

Memori

```
int umur = 19;  
float tinggi = 175.6;
```

```
int *pointer_umur = &umur;  
int *p_tinggi = &tinggi;  
int *pointer2 = &umur;
```

0000	0005	000a
0001	0006	000b
0002	0007	000c
0003	0008	000d
0004	0009	000e

```
#include <stdio.h>

void main(){
    // membuat variabel
    int umur = 19;
    float tinggi = 175.6;

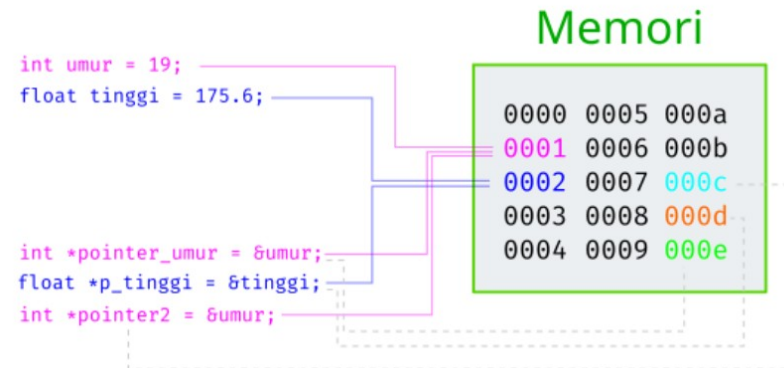
    // membuat pointer
    int *pointer_umur = &umur;
    int *pointer2 = &umur;
    float *p_tinggi = &tinggi;

    // mencetak alamat memori variabel
    printf("alamat memori variabel 'umur' = %d\n", &umur);
    printf("alamat memori variabel 'tinggi' = %d\n", &tinggi);
    // mencetak referensi alamat memori pointer
    printf("referensi alamat memori *pointer_umur = %d\n", pointer_umur);
    printf("referensi alamat memori *pointer2 = %d\n", pointer2);
    printf("referensi alamat memori *p_tinggi = %d\n", p_tinggi);
}
```

pointer juga punya alamat
memorinya sendiri.

Alamat memori pointer
adalah alamat memori
yang digunakan untuk
menyimpan pointer.

alamat referensi adalah
alamat variabel yang
menjadi referensi dari
pointer




```
#include <stdio.h>

void main(){
    // membuat variabel
    int umur = 19;
    float tinggi = 175.6;

    // membuat pointer
    int *pointer_umur = &umur;
    int *pointer2 = &umur;
    float *p_tinggi = &tinggi;

    // mencetak alamat memori variabel
    printf("alamat memori variabel 'umur' = %d\n", &umur);
    printf("alamat memori variabel 'tinggi' = %d\n", &tinggi);
    // mencetak referensi alamat memori pointer
    printf("referensi alamat memori *pointer_umur = %d\n", pointer_umur);
    printf("referensi alamat memori *pointer2 = %d\n", pointer2);
    printf("referensi alamat memori *p_tinggi = %d\n", p_tinggi);

    // mencetak alamat memori pointer
    printf("alamat memori *pointer_umur = %d\n", &pointer_umur);
    printf("alamat memori *pointer2 = %d\n", &pointer2);
    printf("alamat memori *p_tinggi = %d\n", &p_tinggi);
}
```

Mengakses Data dengan Pointer

```
#include <stdio.h>

void main(){
    // membuat variabel score
    int score = 0;

    // membuat pointer dan referensikan dengan alamat
    // dari variabel score
    int *p_score = &score;

    // mengakses data variabel score dari pointer
    printf("isi *p_score = %d\n", *p_score);

    printf("isi score awal = %d\n", score);

    // mengubah data variabel score dari pointer
    *p_score = 5;

    // melihat isi variabel score
    printf("isi score setelah diubah = %d\n", score);
}
```

Pointer Pass by Reference

```
#include <stdio.h>

void add_score(int *score){
    *score = *score + 5;
    printf("score diubah ke: %d\n", *score);
}

void main(){
    int score = 0;

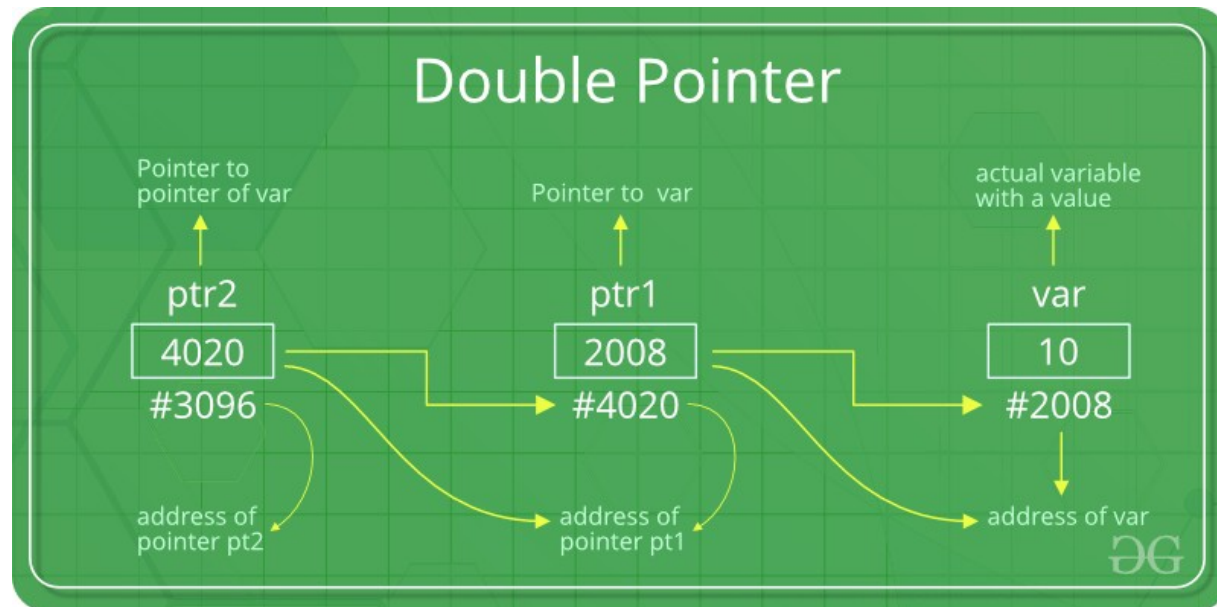
    printf("score sebelum diubah: %d\n", score);
    add_score(&score);
    printf("score setelah diubah: %d\n", score);
}
```

Pointer untuk Mengakses Array

```
char no_antrian[5] = {'A', 'B', 'C', 'D', 'E'};  
  
// menggunakan pointer  
char *ptr_current = &no_antrian;
```

Double Pointer

`int **ptr;`



Example double Pointer

```
// C program to demonstrate pointer to pointer
int main()
{
    int var = 789;

    // pointer for var
    int *ptr2;

    // double pointer for ptr2
    int **ptr1;

    // storing address of var in ptr2
    ptr2 = &var;

    // Storing address of ptr2 in ptr1
    ptr1 = &ptr2;

    // Displaying value of var using
    // both single and double pointers
    printf("Value of var = %d\n", var );
    printf("Value of var using single pointer = %d\n", *ptr2 );
    printf("Value of var using double pointer = %d\n", **ptr1);

    return 0;
}
```