

Algoritme dan Pemrograman

Kuliah #9
Pointer

Address and Pointers

Address (alamat)

- Tiap variabel memiliki dua atribut: alamat (address) dan nilai (value).
- Alamat adalah lokasi dalam memori komputer.
- Pada lokasi tersebut, suatu nilai disimpan.
- Operator untuk mendapatkan alamat adalah *ampersand* (&).
- Format untuk menuliskan alamat adalah %p.

Pointers

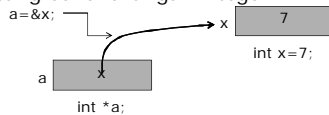
- Pointer adalah variabel yang nilainya berupa alamat.
- Pointer ke integer adalah variabel yang dapat menyimpan alamat dari suatu bilangan integer.
- Operator untuk mendeklarasikan pointer adalah asterisk (*)

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Pointer

- Pointer digunakan untuk memanipulasi memori komputer.
- Pointer dideklarasikan menggunakan asterisk (*). Contoh:

`int *a;`
berarti variabel **a** berisi suatu alamat memori, yang berisi bilangan integer.



DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Contoh penggunaan operator * dan &

Alamat x adalah 0028FF44
Nilai x adalah 7
Alamat a adalah 0028FF40
Nilai a adalah 0028FF44
Nilai *a adalah 7

```
#include <stdio.h>
main()
{
    int x;      /* x adalah integer */
    int *a;    /* a adalah pointer ke integer */

    x = 7;
    a = &x;   /* a = alamat dari x */

    printf("Alamat x adalah %p"
           "\nNilai x adalah %d"
           "\nAlamat a adalah %p"
           "\nNilai a adalah %p"
           "\nNilai *a adalah %d\n", &x, x, &a, a, *a);

    return 0;
}
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Pointer ke NULL

Mendefinisikan pointer tidak menginisialisasi alamat yang ditunjuk. Agar jelas bahwa pointer tidak menunjuk ke alamat manapun, gunakan NULL.

```
#include <stdio.h>
Main()
{
    int n = 0;
    int *p = NULL;

    n = 10;
    printf("\nAlamat n = %p", &n); /* Output berupa alamat */
    printf("\nNilai n = %d", n);  /* Output berupa nilai */

    return 0;
}
```

Alamat n = 0022FF74
Nilai n = 10

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Mengubah nilai suatu variabel melalui pointer

Apa output program berikut?

```
#include <stdio.h>
main()
{
    long int num1=0, num2 = 0;
    long int *p = NULL;

    p = &num1; *p = 2; ++num2; num2 += *p;
    p = &num2; ++*p;

    printf ("%ld %ld %ld %ld\n", num1, num2, *p, *p+num2);
    return 0;
}
```

2 4 4 8

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Pointer dan array

- Array dapat didefinisikan sebagai pointer ke elemen pertama dari array.
- Oleh karena itu, pointer dan array merupakan dua struktur data yang saling berkaitan dan dapat saling dipertukarkan penggunaannya.
- Deklarasi array dapat menggunakan pointer seperti contoh berikut:

```
int *x;
int *(*y);
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Contoh pointer dan array

Apa output program berikut?

```
#include <stdio.h>
#define N 3

int main(){
    int i, a[N];
    int *b;
    for(i=0; i<N; i++) a[i]=i;

    b=a; /* pointer ke elemen pertama dari a */
    for(i=0; i<N; i++){
        printf("%d ",*b);
        b++; /* pointer ke elemen berikutnya */
    }
    return 0;
}
```

0 1 2

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Pemanggilan fungsi

- Pemanggilan fungsi:
 - Call by value
 - Call by reference
- Call by value, suatu fungsi dipanggil dengan mengirimkan argumen berupa nilai variabel. Contoh:


```
int a=5, b=10, c1, c2, c3;
c1=jumlah(5,20);
c2=jumlah(a,b);
c3=jumlah(3, b);
```
- Nilai variabel argumen tidak berubah selama dideklarasikan sebagai variabel lokal.

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Call by Value dan Variabel Lokal

```
#include <stdio.h>
int tukar(int a, int b) {
    int t=a;
    a=b;
    b=t;
    return (a+b);
}

main() {
    int a=5, b=10;
    printf("%d %d %d", a, b, tukar(a,b));
    return 0;
}
```

APA KELUARAN PROGRAM TERSEBUT?

Call by value dan Variabel Global

```
#include <stdio.h>
int a=5, b=10;

int tukar() {
    int t=a;
    a=b; b=t;
    return (a+b);
}

main() {
    printf("%d %d %d", a, b, tukar(a,b));
    return 0;
}
```

APA KELUARAN PROGRAM TERSEBUT?

Pemanggilan fungsi: call by reference

- Call by reference, suatu fungsi dipanggil dengan mengirimkan argumen berupa alamat dari variabel.
- Oleh karena itu, parameter yang menerima di dalam fungsi haruslah berupa pointer.

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Contoh pemanggilan fungsi: call by reference

Apa output program berikut?

```
#include <stdio.h>

void misteri(int x, int *y) {
    int t = x;
    x = *y;
    *y = t;
}

main() {
    int a=5, b=10;

    misteri(a, &b); /* call by reference */

    printf("%d %d\n", a, b);
    return 0;
}
```

5 5

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Apa keluaran program berikut?

```
#include <stdio.h>
int misteri(int, int *);
int main() {
    int a=5, b=10, c=15;
    a += misteri(b, &c);
    printf("%d %d %d", a, b, c);
    return 0;
}

int misteri(int a, int *b) {
    int c=3;
    *b = a + c;
    return c;
}
```

Latihan 1: apa output program berikut?

```
#include <stdio.h>
#define SIZE 4
main() {
    int a[][SIZE] = {{0,1,2,3}, {4,5,6}, {7,8,9,10}};
    int *p = &a[0][0];
    int *q = a[0];
    int *r = a[1];
    int *s = a[2];

    printf("%d ", *(p + SIZE + 1));
    printf("%d ", p[SIZE + 1]);
    printf("%d ", p[2 * SIZE + 1]);
    printf("%d ", *(q + 2 * SIZE + 2));
    printf("%d ", *r);
    printf("%d ", *(r - 2));
    printf("%d\n", s[3]);
    return 0;
}
```

5 5 8 9 4 2 10

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Latihan 2: apa output program berikut?

```
#include <stdio.h>

int x = 2, y;

void misteri(int a, int *b) {
    a = 2*x; *b = y; y = a;
}

main() {
    x = 5; y = 7;
    misteri(y, &x);
    printf("%d %d\n", x, y);
    return 0;
}
```

7 10

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Latihan 3: apa output program berikut?

```
#include <stdio.h>
#define N 10
void what(int *b, int n) {
    if (n) {
        printf("%d ", *b++);
        what(&b[0], n-1);
    }
}

main() {
    int x[N]={10, 20, 30, 40, 50};
    what(x, 5);
    return 0;
}
```

10 20 30 40 50

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Array Dinamis

- Deklarasi array adalah pemesanan ruang di memori
- Bagaimana jika ukuran yang dibutuhkan belum diketahui?
 - Ukuran berdasarkan input pengguna
 - Ukuran berdasarkan hasil perhitungan
- Array dinamis
 - pesan memori sesuai kebutuhan

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Alokasi Memori

- Alokasi memori dapat dilakukan dengan fungsi:
void *malloc (size_t size)

tsize = ukuran yang diinginkan; bisa memanfaatkan fungsi **sizeof**

- Setelah tidak diperlukan, dapat dibebaskan dengan fungsi:
void free(void *ptr)

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Contoh

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int n, i;
    int *a, *b;
    scanf("%d", &n);
    a = (int*) malloc (n * sizeof(int));
    b = (int*) malloc (n * sizeof(int));
    for (i = 0; i<n; i++) {a[i] = rand(); b[n-i-1] = a[i];}
    for (i = 0; i<n; i++) printf("%d ", a[i]); printf("\n");
    for (i = 0; i<n; i++) printf("%d ", b[i]); printf("\n");
    free(a);
    return 0;
}
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR