

Algoritme dan Pemrograman

Kuliah #9 Pointer

Alamat dan *Pointer*

Address (alamat)

- Tiap variabel memiliki dua atribut: alamat (*address*) dan nilai (*value*).
- Alamat adalah lokasi dalam memori komputer.
- Lokasi tersebut digunakan untuk menyimpan nilai.
- Operator untuk mendapatkan alamat adalah *ampersand* (&).
- Format string* 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 dapat digunakan untuk memanipulasi memori komputer.
- Pointer dideklarasikan menggunakan asterisk (*). Contoh:


```
int *a;
```

 berarti variabel **a** berisi suatu alamat memori, yang berisi bilangan integer.
- Operator * digunakan untuk mendapatkan nilai yang ada di alamat yang tersimpan dalam pointer

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Contoh penggunaan operator

Perhatikan bahwa nilai &x, &a, dan a akan tergantung pada alamat memori yang dialokasikan. Tetapi, a akan sama dengan &x, dan *a akan sama dengan x.

```
#include <stdio.h>
int 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\n", &x);
    printf( "\nNilai x adalah %d\n", x);
    printf( "\nAlamat a adalah %p\n", &a);
    printf( "\nNilai a adalah %p\n", *a);
    printf( "\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. Penggunaan *pointer* NULL akan menimbulkan *runtime-error* (berguna untuk *debugging*)

```
#include <stdio.h>
int 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 */
    printf("\nAlamat p = %p", &p); /* Output berupa alamat */
    printf("\nNilai p = %p", p); /* Output berupa nilai */
    return 0;
}
```

Apa yang terjadi jika dilakukan pencetakan *p?

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Mengubah nilai suatu variabel melalui *pointer*

Apa output program berikut?

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

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

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

2 4 4 8

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Pointer dan array

- Array dapat diperlakukan seperti *pointer* ke elemen pertama (indeks 0).
- *Pointer* juga dapat digunakan untuk membuat *array* (terutama yang dinamis)
- Deklarasi array dapat menggunakan pointer seperti contoh berikut:

```
int *x; // 1D
int *(*y); // 2D
```

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

Manipulasi *Pointer*

```
#include <stdio.h>
```

```
int main() {
    long int i[2] = {0, 1}, *pi = &i;
    long long j[2] = {0, 1}, *pj = &j;

    printf ("%p %ld\n", pi, *pi);
    printf ("%p %lld\n", pj, *pj);
    pi++, pj++;
    printf ("%p %ld\n", pi, *pi);
    printf ("%p %lld\n", pj, *pj);
    return 0;
}
```

Perhatikan perbedaan selisih antara pi dan pi++ serta pj dan pj++

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Pemanggilan fungsi

- Pemanggilan fungsi:
 - Call by value
 - Call by reference
- *Call by value*: nilai argumen yang dikirimkan disalin ke variabel di fungsi. Contoh:


```
int a=5, b=10, c1, c2, c3;
c1=jumlah(5, 20);
c2=jumlah(a, b);
c3=jumlah(3, b);
```
- Perubahan variabel lokal tidak berpengaruh pada nilai variabel yang dikirim ke fungsi.

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);
}

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

APA KELUARAN PROGRAM TERSEBUT? 5 10 15

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);
}

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

APA KELUARAN PROGRAM TERSEBUT? 10 5 15

Variabel Lokal vs Global

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

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

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

APA KELUARAN PROGRAM TERSEBUT? 10 5 15

Pemanggilan fungsi: *call by reference*

- *Call by reference*: argumen dikirimkan berupa alamat.
- 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;
}

int 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;
}
```

8 10 13

Latihan 1: Keluaran?

```
#include <stdio.h>
#define SIZE 4
int 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: Keluaran?

```
#include <stdio.h>

int x = 2, y;

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

int 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: Keluaran?

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

int 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 masukan 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** misalnya **sizeof(int)** akan mengembalikan ukuran yang dibutuhkan oleh sebuah integer

- Manipulasi dapat dilakukan dengan notasi *array*
- Setelah tidak diperlukan, perlu 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()%10; 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