

Konsep Pemrograman

11. Pointer 1

Umi Sa'adah

Entin Martiana Kusumaningtyas

Tri Hadiah Muliawati

2021



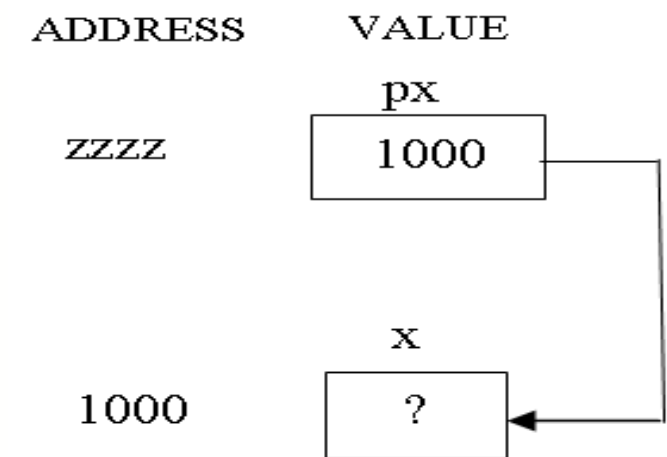
Politeknik Elektronika Negeri Surabaya
Departemen Teknik Informatika dan Komputer

Overview

- Konsep Dasar Pointer
- Deklarasi Variabel Pointer
- Mengatur Pointer agar Menunjuk ke Variabel Lain
- Akses INDIRECT melalui pointer

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 dikatakan 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

- Pointer dideklarasikan dengan bentuk sbb:

```
tipe_data *nama_variabel;
```

- **tipe_data** dapat berupa sembarang tipe yang sudah dibahas pada bab-bab sebelumnya, maupun bab-bab berikutnya.
- **nama_variabel** adalah nama dari variabel pointer.
- Jika dideklarasikan :

```
int *px;
```

 - nama var = `px`; tipenya = *pointer to int*
 - menyatakan bahwa `px` adalah variabel pointer yang menunjuk ke suatu data tertentu yang bertipe *int*



Mengarahkan Pointer ke Variabel Lain

- Mula-mula pointer diisi dengan alamat dari variabel yang akan ditunjuk.
- Untuk menyatakan alamat dari suatu variabel, digunakan operator & (operator alamat, bersifat *unary*) yang ditempatkan di depan nama variabel.
- Jika dideklarasikan :

```
int *px, x = 10;
```

Maka `&x` berarti “alamat dari variabel **x**”.

```
px = &x;
```

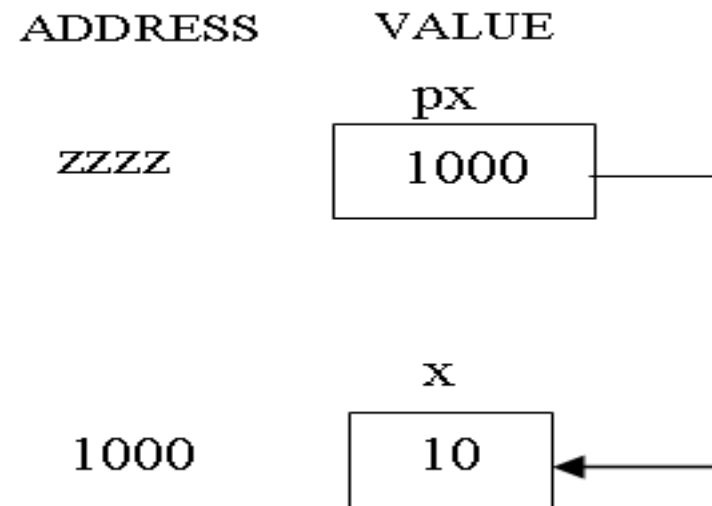
berarti bahwa `px` diberi nilai berupa alamat dari variabel **x**.

- Setelah pernyataan tersebut dieksekusi barulah dapat dikatakan bahwa **px** menunjuk ke variabel **x**.



Mengarahkan Pointer ke Variabel Lain

- Hasilnya :



- Suatu lokasi/address yg telah ditunjuk oleh sebuah pointer, maka lokasi tsb value-nya bisa diakses baik secara DIRECT maupun INDIRECT melalui pointernya

Akses INDIRECT melalui pointer

- Pengaksesan tak langsung dilakukan dengan menggunakan operator *indirection* (tak langsung) berupa simbol ***** (bersifat *unary*).

`*px`

menyatakan “nilai atau value dari variabel/data yang ditunjuk oleh pointer **px**” .

- Contoh

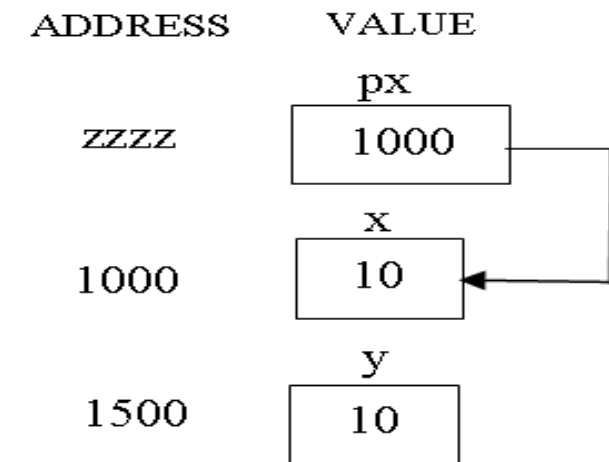
```
int *px, x = 10, y;
```

```
px = &x;
```

```
y = *px;
```

maka `y` akan berisi 10 yaitu

nilai yang sama dengan nilai `x`



Akses INDIRECT melalui pointer

ATURAN PEMBACAAN

- **&x** = alamat/address dari variabel x
- ***px** = nilai/value yang ada pada address/lokasi px
 - Baca ulang dengan cara mengganti px dengan value yang ada pada px (dalam contoh di atas px berisi 1000)
 - merupakan cara akses indirect

Keterangan:

- x adalah nama variabel yang bertipe *int*
- px adalah nama variabel yang bertipe *pointer to int*

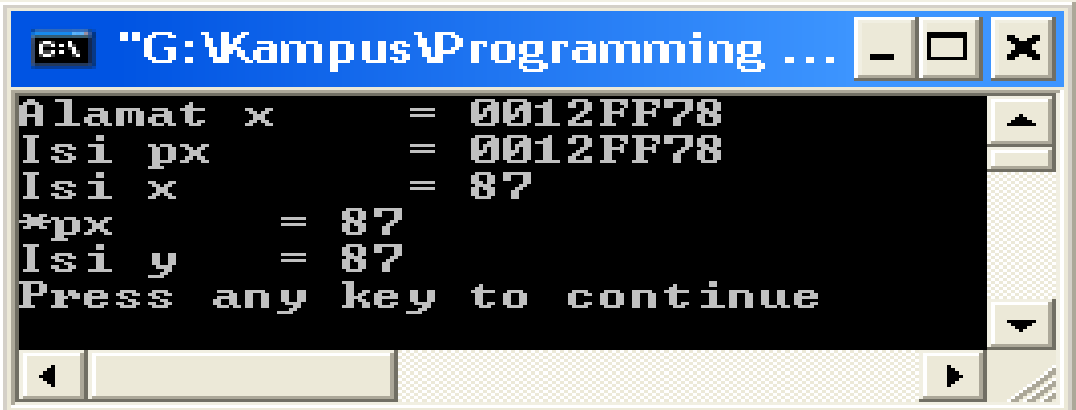


Contoh

```
#include <stdio.h>
main()
{
    int y, x = 87, *px;

    px = &x;
    y = *px;

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



A screenshot of a Windows command prompt window titled "G:\Kampus\Programming ...". The window displays the output of the C program shown on the left. The output is as follows:

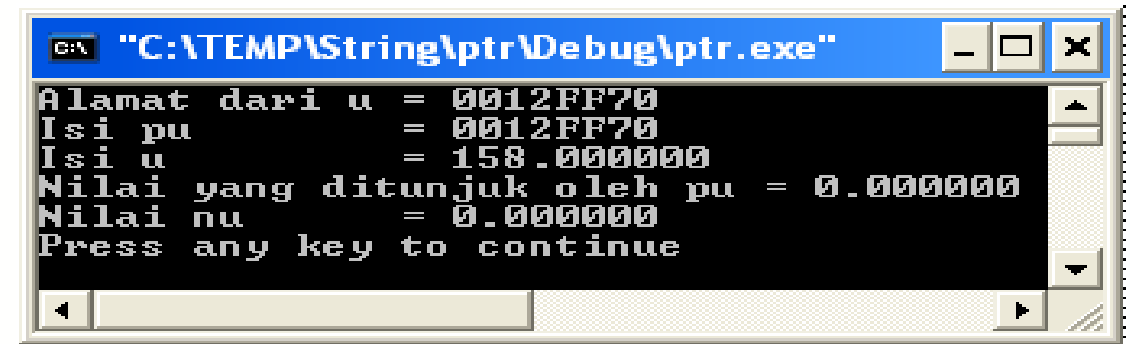
```
Alamat x      = 0012FF78
Isi px        = 0012FF78
Isi x          = 87
*px           = 87
Isi y          = 87
Press any key to continue
```

Akses INDIRECT melalui pointer

- Tipe variabel pointer dan tipe data yang ditunjuknya harus sejenis. Bila tidak sejenis, akan terjadi hasil yang tidak diinginkan

Contoh

```
main() {  
    float *pu, nu;  
    double u = 158.0;  
  
    pu = &u;  
    nu = *pu;  
  
    printf("Alamat dari u = %p\n", &u);  
    printf("Isi pu          = %p\n", pu);  
    printf("Isi u           = %lf\n", u);  
    printf("Nilai yang ditunjuk oleh pu = %f\n", *pu);  
    printf("Nilai nu           = %f\n", nu);  
}
```



```
C:\TEMP\String\ptr\Debug\ptr.exe  
Alamat dari u = 0012FF70  
Isi pu          = 0012FF70  
Isi u           = 158.000000  
Nilai yang ditunjuk oleh pu = 0.000000  
Nilai nu           = 0.000000  
Press any key to continue
```

Latihan

Untuk setiap program di bawah ini,

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

```
1. 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);  
}
```

Referensi

1. Brian W. Kernighan, Dennis M. Ritchie (2012): The C Programming Language : Ansi C Version 2 Edition, PHI Learning
2. Byron Gottfried (2010) : Programming with C, Tata McGraw - Hill Education
3. Kochan Stephen (2004) : Programming in C, 3rd Edition, Sams
4. K. N. King (2008) : C Programming: A Modern Approach, 2nd Edition, W. W. Norton & Company
5. Abdul Kadir (2012) : Algoritma & Pemrograman Menggunakan C & C++, Andi Publisher, Yogyakarta
6. <http://www.gdsw.at/languages/c/programming-bbrown/>
7. <https://www.petanikode.com/tutorial/c/>
8. <http://www.cprogramming.com/tutorial/c-tutorial.html>



bridge to the future

<http://www.eepis-its.edu>