PENGERTIAN VARIABEL POINTER

Pathul Khorib

185100030

Fakultas Komputer

Pathulkhorib.student@umitra.ac.id

Abstract

Pointer dapat didefinisikan sebagai suatu variabel yang menyimpan alamat memori. Beda dengan variabel biasa yang menyimpan sebuah nilai, pointer ini berisikan alamat saja.

Jika kita mempunyai sebuah variabel dengan tipe data tertentu, maka untuk mendapatkan alamat dari variabel tersebut adalah dengan menggunakan operator &. Alamat inilah yang kemudian akan disimpan ke dalam variabel yang bertipe pointer. Untuk mendeklarasikan variabel sebagai pointer, kita hanya menambahkan tanda asterik (*) di depan nama variabel. Berikut ini bentuk umum dari pendeklarasian variabel yang bertipe pointer.

mendeklarasikan pointer P yang menunjuk ke tipe long
long *P ; Jika kita mempunyai sebuah variabel yang bertipe long (misalnya X), maka kita dapat memerintahkan pointer P di atas untuk menunjukkan ke alamat yang ditempati oleh variabel X.

Kata Kunci : Penerimaan dan Pengeluaran Kas.

atau alamat memory. Variabel pointer berarti variabel yang berisi alamat memori suatu nilai. Sebagai contoh sebuah variabel bertipe integer memiliki nilai 4, maka yang dimaksud pointer adalah alamat memori dimana nilai 4 disimpan, bukan nilai 4 nya sendiri.

Variabel-variabel yang memiliki reference atau alamat pointer yang sama, saling berhubungan satu sama lain dan nilainya pasti sama. Ketika ada perubahan nilai, maka akan memberikan efek kepada variabel lain (yang referensinya sama) yaitu nilainya ikut berubah.

B. PEMBAHASAN / STUDI KASUS

A. Pengertian Pointer

Pointer adalah sebuah variabel yang berisi alamat lain. Suatu pointer dimaksudkan untuk menunjukan ke suatu alamat memori sehingga alamat dari suatu variabel dapat diketahui dengan mudah.

Intinya :

- Pointer adalah suatu variabel yang menunjuk ke alamat memory variabel yang lainnya.
- Variabel pointer berisi suatu alamat (alokasi memory).

B. Fungsi Pointer

Kegunaan pointer yang utama adalah untuk menyimpan alamat memori dari sebuah variabel dan alamat dari sebuah fungsi. Pointer dapat meningkatkan kinerja untuk operasi yang dilakukan secara berulang.

Ada 2 operator pointer yang dikenal secara luas, yaitu operator "&" dan operator "*".

a. Operator &

Operator & merupakan operator alamat. Pada saat pendeklarasian variabel, user tidak diharuskan menentukan lokasi sesungguhnya pada memory, hal ini akan dilakukan secara otomatis oleh kompiler dan operating sysem pada saat run-time.

Jika ingin mengetahui dimana suatu variabel akan disimpan, dapat dilakukan dengan memberikan tanda ampersand (&) didepan variable , yang berarti "address of".

Contoh :

Misalkan variabel DATA_1 diletakkan pada alamat memory 1770, kemudian dituliskan instruksi

sbb :

```
DATA_1 = 27;
```

Variabel DATA_1 berisi data 27

```
DATA_2 = DATA_1;
```

Variabel DATA_2 diberi isi

seperti DATA_1, yaitu 27
DATA_3 = &DATA_1;
Variabel DATA_3 berisi
alamat memory DATA_1,
yaitu 1770

b. Operator *

Operator * merupakan
operator reference. Dengan
menggunakan pointer, kita
dapat mengakses
nilai yang tersimpan
secara langsung dengan
memberikan awalan
operator asterisk (*) pada
identifier pointer, yang
berarti "value pointed by".

Contoh :

Melanjutkan deklarasi
sebelumnya, jika ada
penulisan variabel berikut
DATA_4 = *DATA_3
Dapat dikatakan bahwa
DATA_4 sama dengan nilai
yang ditunjuk oleh DATA_3.
DATA_3 berisi alamat
memory 1770, sementara
memory 1770 menampung
data bernilai 27.
Jadi DATA_4 berisi nilai
yang berada pada alamat
1770, yaitu 27.

C. ID SECURITY

QWTD4452377-ASP-
5244107

D. KESIMPULAN

Ada 2 operator pointer yang
dikenal secara luas, yaitu
operator "&" dan operator "*".

a. Operator &

Operator & merupakan
operator alamat. Pada

saat pendeklarasian
variabel, user tidak
diharuskan menentukan
lokasi sesungguhnya pada
memory, hal ini akan
dilakukan secara
otomatis oleh kompiler dan
operating system pada saat
run-time.

Jika ingin mengetahui
dimana suatu variabel akan
disimpan, dapat dilakukan
dengan memberikan
tanda ampersand (&)
didepan variable, yang
berarti "address of".

Contoh :

Misalkan variabel DATA_1
diletakkan pada alamat
memory 1770, kemudian
dituliskan instruksi

sbb :

DATA_1 = 27;

Variabel DATA_1 berisi data
27

DATA_2 = DATA_1;

Variabel DATA_2 diberi isi
seperti DATA_1, yaitu 27

DATA_3 = &DATA_1;

Variabel DATA_3 berisi
alamat memory DATA_1,
yaitu 1770

b. Operator *

Operator * merupakan
operator reference. Dengan
menggunakan pointer, kita
dapat mengakses

nilai yang tersimpan
secara langsung dengan
memberikan awalan
operator asterisk (*) pada
identifier pointer, yang
berarti "value pointed by".

Contoh :
Melanjutkan deklarasi sebelumnya, jika ada penulisan variabel berikut
`DATA_4 = *DATA_3`
Dapat dikatakan bahwa `DATA_4` sama dengan nilai yang ditunjuk oleh `DATA_3`.
`DATA_3` berisi alamat memori 1770, sementara memori 1770 menampung data bernilai 27.
Jadi `DATA_4` berisi nilai yang berada pada alamat 1770, yaitu 27.
Function pointer telah digunakan sejak dikenalkannya bahasa C, dan banyak digunakan untuk sebuah fungsi callback atau untuk meningkatkan readability dari sebuah code. Anda dapat memperlakukan function pointer seperti pointer biasa (pointer ke datatype / object), anda dapat menyimpan, mengirimkan, merubah address, atau meng-evaluasi address dari pointer ke fungsi ini dengan sifat tambahan anda dapat memanggil fungsi yang ditunjuk oleh function pointer.

E. DISKUSI

Saya :apa itu variable pointer
Yay :variable pointer adalah sebuah variable yang berisi alamat lain.yang dimasukan untuk menunjukan ke suatu alamat memori sehingga dapat di ketahui dengan mudah.

Risky :apa kegunaan utama pointer.

Pathul :untuk menyimpan alamat memori dari sebuah variable.

F. REFERENCE

- [1] O. M. Febriani and A. S. Putra, "Sistem Informasi Monitoring Inventori Barang Pada Balai Riset Standardisasi Industri Bandar Lampung," *J. Inform.*, vol. 13, no. 1, pp. 90–98, 2014.
- [2] A. S. Putra, "Paperplain: Execution Fundamental Create Application With Borland Delphi 7.0 University Of Mitra Indonesia," 2018.
- [3] A. S. Putra, "2018 Artikel Struktur Data, Audit Dan Jaringan Komputer," 2018.
- [4] A. S. Putra, "ALIAS MANAGER USED IN DATABASE DESKTOP STUDI CASE DB DEMOS."
- [5] A. S. Putra, "COMPREHENSIVE SET OF PROFESSIONAL FOR DISTRIBUTE COMPUTING."
- [6] A. S. Putra, "DATA ORIENTED RECOGNITION IN BORLAND DELPHI 7.0."
- [7] A. S. Putra, "EMBARCADERO DELPHI XE 2 IN GPU-POWERED FIREMONKEY APPLICATION."
- [8] A. S. Putra, "HAK ATAS KEKAYAAN INTELEKTUAL DALAM DUNIA TEKNOLOGY BERBASIS REVOLUSI INDUSTRI 4.0."

- [9] A. S. Putra, "IMPLEMENTASI PERATURAN PERUNDANGAN UU. NO 31 TAHUN 2000 TENTANG DESAIN INDUSTRI BERBASIS INFORMATION TECHNOLOGY."
- [10] A. S. Putra, "IMPLEMENTATION OF PARADOX DBASE."
- [11] A. S. Putra, "IMPLEMENTATION OF TRADE SECRET CASE STUDY SAMSUNG MOBILE PHONE."
- [12] A. S. Putra, "IMPLEMENTATION PATENT FOR APPLICATION WEB BASED CASE STUDI WWW. PUBLIKLAMPUNG.COM."
- [13] A. S. Putra, "IMPLEMENTATION SYSTEM FIRST TO INVENT IN DIGITALLY INDUSTRY."
- [14] A. S. Putra, "MANUAL REPORT & INTEGRATED DEVELOPMENT ENVIRONMENT BORLAND DELPHI 7.0."
- [15] A. S. Putra, "PATENT AS RELEVAN SUPPORT RESEARCH."
- [16] A. S. Putra, "PATENT FOR RESEARCH STUDY CASE OF APPLE. Inc."
- [17] A. S. Putra, "PATENT PROTECTION FOR APPLICATION INVENT."
- [18] A. S. Putra, "QUICK REPORT IN PROPERTY PROGRAMMING."
- [19] A. S. Putra, "REVIEW CIRCUIT LAYOUT COMPONENT REQUIREMENT ON ASUS NOTEBOOK."
- [20] A. S. Putra, "REVIEW TRADEMARK PATENT FOR INDUSTRIAL TECHNOLOGY BASED 4.0."
- [21] A. S. Putra, "TOOLBAR COMPONENT PALLETTE IN OBJECT ORIENTED PROGRAMMING."
- [22] A. S. Putra, "WORKING DIRECTORY SET FOR PARADOX 7."
- [23] A. S. Putra, "ZQUERY CONNECTION IMPLEMENTED PROGRAMMING STUDI CASE PT. BANK BCA Tbk."
- [24] A. S. Putra, D. R. Aryanti, and I. Hartati, "Metode SAW (Simple Additive Weighting) sebagai Sistem Pendukung Keputusan Guru Berprestasi (Studi Kasus: SMK Global Surya)," in *Prosiding Seminar Nasional Darmajaya*, 2018, vol. 1, no. 1, pp. 85–97.
- [25] A. S. Putra and O. M. Febriani, "Knowledge Management Online Application in PDAM Lampung Province," in *Prosiding International conference on Information Technology and Business (ICITB)*, 2018, pp. 181–187.
- [26] A. S. Putra, O. M. Febriani, and B. Bachry, "Implementasi Genetic Fuzzy System Untuk Mengidentifikasi Hasil Curian Kendaraan Bermotor Di Polda Lampung," *SIMADA (Jurnal Sist. Inf. dan Manaj. Basis Data)*, vol. 1, no. 1, pp. 21–30, 2018.
- [27] A. S. Putra, H. Sukri, and K. Zuhri, "Sistem Monitoring Realtime Jaringan Irigasi Desa (JIDES) Dengan

Konsep Jaringan Sensor Nirkabel," *IJEIS (Indonesian J. Electron. Instrum. Syst.*, vol. 8, no. 2, pp. 221–232.

[28] D. P. Sari, O. M. Febriani, and A. S. Putra, "Perancangan Sistem Informasi SDM Berprestasi pada SD Global Surya," in *Prosiding Seminar Nasional Darmajaya*, 2018, vol. 1, no. 1, pp. 289–294.