



BAB I ALOKASI DAN REPRESENTASI DATA

Tujuan Pembelajaran Umum :

Setelah menyelesaikan pembahasan ini, mahasiswa :

1. Dapat membedakan cara alokasi statis dan dinamis berdasarkan ciri khas dan sifat alokasi memory;
2. Mengetahui kapan implementasi yang tepat dari jenis alokasi yang dipilih;
3. Memahami tahapan representasi data dengan acuan type alokasi memory yang dipilihnya berdasarkan kebutuhan dunia nyata;
4. Dapat menyimpulkan perbedaan keempat type data elementer terutama pemahaman akan type data pointer;
5. Dapat memberikan contoh tentang penerapan pointer baik sebagai type data elementer dan type data komposit.

Tujuan Pembelajaran Khusus :

Setelah menyelesaikan pembahasan ini, mahasiswa :

1. Dapat menyebutkan dan menjelaskan perbedaan type data elementer dan komposit;
2. Dapat menyebutkan ciri khas alokasi statis dan dinamis;
3. Dapat menyebutkan contoh penerapan representasi data baik representasi fungsional, logikal dan fisik;
4. Dapat menyusun, merancang dan menelusuri type data pointer;
5. Dapat mempraktekkan deklarasi type pointer, assignment terhadap type pointer dan menyusun algoritma insert, delete dan searching pada Single Linked List Non sirkuler.

1.1 Pendahuluan

Software komputer digital selalu berhubungan dengan kreasi, operasi, akses, dan menghapus suatu struktur data dalam kaitan penyelesaian masalah. Komputasi untuk penyelesaian masalah dapat dikategorikan sebagai : matematis, grafik, text, voice, image, dan multimedia. Dalam istilah komputer, data merupakan nama yang diberikan pada suatu item yang dapat dioperasikan oleh program komputer. Setiap item data yang digunakan dalam suatu program di komputer mempunyai beberapa atribut berikut :

1. Alamat (address)
2. Nilai (value) yang berkaitan dengan representasi dan interpretasi
3. Sekumpulan operasi yang diberlakukan terhadap data
4. Pengenal (identifikasi) yang menyatukan alamat, nilai, dan operasi

Nature atribut tersebut diwujudkan dalam bentuk tipe dari suatu item data. Tipe data merupakan algoritma yang berfungsi menginterpretasikan nilai bit string menjadi nilai yang representatif sesuai dengan tipe data yang didefinisikan. Pada

Struktur data

halaman : 1

umumnya tipe data dikategorikan sebagai (a). tipe data primitif : integer, real, boolean, character, dan pointer; (b) tipe data komposit : array, string, record, enumerasi, set, dan list; dan (c) user defined data types.

1.1.1 Alamat (address)

Alamat atau referensi suatu item data merupakan lokasi dalam memori (computer store) suatu item data. Banyaknya memori yang dibutuhkan suatu item data ditentukan oleh tipe datanya.

1.1.2 Nilai (value)

Bilangan biner disimpan pada suatu alamat yang diasosiasikan dengan suatu item data disebut dengan nilai (value). Interpretasi terhadap suatu nilai ditentukan oleh tipe datanya.

Misalkan 01011010, dapat diinterpretasikan sebagai :

- Integer desimal 90
- Integer heksadesimal 5A
- Integer oktal 132
- Karakter ASCII Z (huruf besar)

1.1.3 Operasi

Tipe dari suatu item data berperan juga dalam mendikte operasi-operasi yang sah diperbolehkan. Misalkan tipe data numerik mempunyai standar operasi matematis *, /, +, -, ^; sedangkan tipe data string tidak dapat diperlakukan seperti halnya tipe data numerik.

1.1.4 Nama (identifier)

Keseluruhan alamat, nilai, dan operasi harus dikemas dengan diberikan suatu nama identifikasi (label/ simbol) sesuai dengan aturan standar penamaan. Dikebanyakan bahasa pemrograman aturan penamaan antara lain :

- Nama dibentuk dari kombinasi karakter alpha numerik dan garis bawah (_).
- Spasi tidak diperbolehkan.
- Huruf kecil atau huruf besar significant.
- Tidak diperbolehkan kata kunci/reserved sebagai identifier.

1.2 Deklarasi Data, Assignment, Inisialisasi

Deklarasi data dimaksudkan untuk memperkenalkan suatu item data. Dalam lingkungan bahasa pemrograman JAVA, C, PASCAL, Visual BASIC dilakukan dengan cara mengasosiasikan identifier item dengan suatu tipe data :

int integerItem;	(JAVA, C)
var integerItem : integer;	(PASCAL)
DIM integerItem AS integer	(Visual BASIC)

Disini dilakukan deklarasi item data `integerItem` yang bertipe `int` (integer/ bilangan asli). Deklarasi item data dapat dilakukan dalam program utama, prosedur, dan fungsi.

Assignment merupakan proses mengasosiasikan suatu nilai terhadap suatu item data. Proses Assignment biasanya menggunakan operator assignment (`=`, `:=`), sebagai contoh dalam berbagai bahasa :

integerItem = 5;	(JAVA, C)
integerItem := 5;	(PASCAL)
integerItem = 5	(Visual BASIC)

Operasi di atas dimaksudkan untuk memberikan nilai numerik 5 ke item data `integerItem`. Dapat juga disebelah kanan operator assignment diisikan identifier, ekspresi, atau pemanggilan fungsi, sebagai contoh dalam PASCAL :

integerItem := myDataItem;
integerItem := (myDataItem+2) * someIncrement;
integerItem := myFunction(argument1,argument2);

Inisialisasi merupakan proses memberikan nilai awal suatu item data pada saat deklarasi.

1.3 Range dan Presisi

Range mereferensikan nilai minimum dan maksimum yang diperbolehkan pada tipe numerik.

Presisi mereferensikan banyaknya *significant figures* pada tipe data numerik.

1.4 Parameter dan Variabel

Untuk memahami bagaimana data tersimpan dalam *data storage* komputer, maka perlu dibahas lebih lanjut mengenai keadaan fisik penyimpanan data tersebut.

Sebagaimana telah dibahas dalam materi sebelumnya (di Dasar-dasar pemrograman), secara logik data dikelompokkan dalam jenis data :

- Elementer, yaitu jenis data yang sudah tidak dapat dipecah menjadi elemen yang lebih kecil;
- Komplementer/Gabungan, yaitu jenis data yang terdiri atas gabungan elemen-elemen type data yang lebih kecil (data elementer).
-

Dari keberadaan data dalam memori , penggolongan data dibagi menjadi:

1. Variabel, yaitu suatu penamaan data yang mempunyai isi / nilai dengan tipe tertentu dan bisa diubah
2. Parameter , yaitu suatu Variabel yang pada kurun waktu tertentu isinya tetap

Keberadaan Variabel dalam memory ditandai oleh beberapa unsur berikut :

- Alamat = Posisi (fisik) dari variabel
- Type = Jenis variabel
- Isi = Kandungan data dari variabel
- Retensi = Waktu keberadaan suatu variabel dialokasikan dalam memory
- Alokasi = Sifat pengalokasian variabel dalam memory

Cara alokasi variabel :

- **Statis** = pengalokasian variabel yang tidak dapat diubah lagi (lokasinya tetap)
 Contoh : Array/matriks
 Keuntungannya :
 - . Proses lebih cepat (karena sudah pesan tempat)
 - . Memory harus dapat diatur sesuai kebutuhan (tahu besar memory yang dibutuhkan/tidak boros memory)
- **Dinamis** = Alokasi variabel yang dapat diubah saat eksekusi;
 Contoh : Pointer

1.5 Pointer

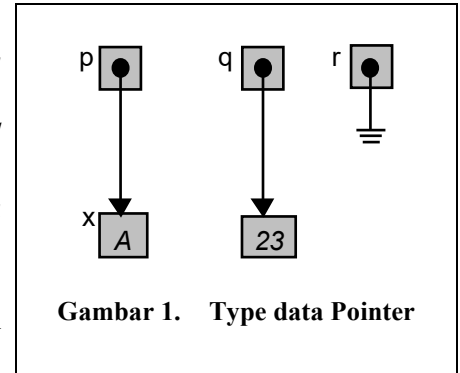
- Merupakan salah satu tipe data elementer, nilainya merepresentasikan *ALAMAT (ADDRESS) data lain*. Secara konsep sebenarnya tidak dikaitkan dengan bentuk fisik nya, apakah alamat pada *primary-memory* atau *secondary-storage*.
- Variabel jenis Pointer : variabel yang nilainya merepresentasikan *alamat data atau variabel lain, atau NIL* (yang berarti "**tidak menunjuk kemana mana**").

Contoh:

misalkan p, q dan r merupakan variabel tunggal jenis pointer, dan :

- p berisi alamat (atau menunjuk ke) variabel tunggal karakter yang bernama x , bernilai "A".
- q berisi alamat (atau menunjuk ke) suatu data numerik integer yang bernilai 23.
- r tidak menunjuk kemanapun, atau bernilai *NIL*.

atau dapat direpresentasikan melalui gambar bagan disamping



- Pointer dapat dipergunakan sebagai alat untuk merujuk data yang ditunjuk olehnya.

misalnya :

Pada kasus seperti pada gambar diatas kita dapat merujuk data dengan nilai 23 (lihat gambar) dengan menyatakannya sebagai "data yang ditunjuk oleh pointer q " atau dengan notasi resmi sbb :

$q \uparrow$ (atau dalam bahasa PASCAL q^{\wedge})

Jika kemudian dilaksanakan operasi sbb :

$q \uparrow \leftarrow q \uparrow + 1$

maka efeknya adalah :

nilai data yang tersimpan pada alamat yang ditunjuk oleh pointer q akan bertambah dengan 1, sehingga dari 23 menjadi 24.

- Assignment terhadap variabel jenis pointer :

a) disalin dari nilai pointer lain :

contoh :

Misalkan ada 2 buah pointer, ialah p dan q dengan state awal sbb :

- q menunjuk kesuatu data integer bernilai 101
- p tidak terdefinisi

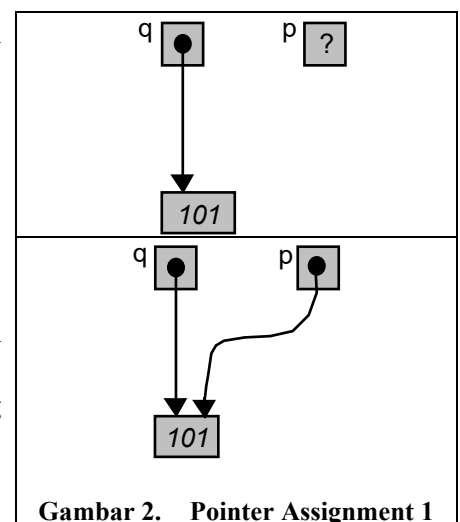
Kemudian setelah dilakukan operasi sbb :

$p \leftarrow q$

State akhirnya adalah :

- q tetap, dan
- p menunjuk ke tempat data yang ditunjuk oleh q .

Jadi sekarang pointer p dan q menunjuk tempat yang sama.



b) diisi langsung oleh alamat variabel lain :

contoh :

Misalkan ada sebuah pointer, ialah p ; dan sebuah variabel integer bernama x ; dengan state awal sbb :

- p tidak terdefinisi

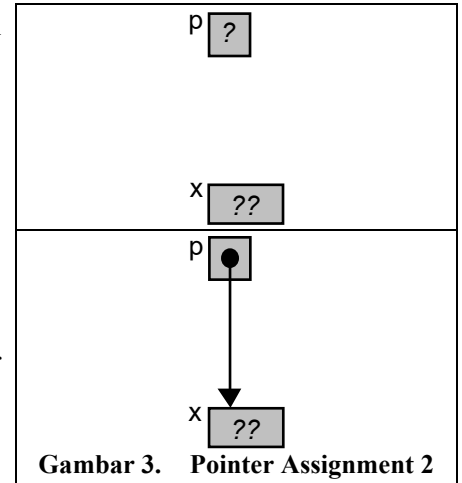
Kemudian setelah dilakukan operasi sbb :

$p \leftarrow \text{address}(x)$

Maka, state akhirnya adalah :

- x tidak terganggu isinya, dan

- p sekarang berisi *alamat* x , jadi digambar menunjuk ke x .



1.6 Tipe Kombinasi Dan Abstrak.

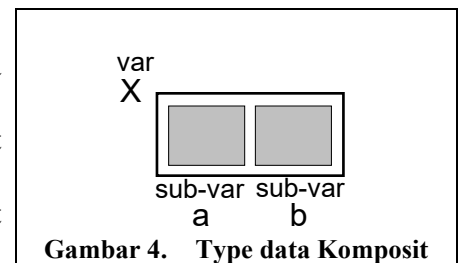
Tipe data tidak terbatas pada tipe data elementer saja. Dapat dilakukan kombinasi dan bahkan pendefinisian tipe data yang "baru". Dapat ditempuh beberapa cara untuk melakukan hal ini.

- Cara ke 1 : pendefinisian suatu variabel yang didalamnya mengandung "sub-variabel".

contoh :

Variabel x merupakan variabel yang didalamnya mengandung 2 bagian (atau subvariabel) sbb :

- subvariabel ke 1 berjenis karakter tunggal disebut a ,
- subvariabel ke 2 berjenis integer tunggal disebut b .



referensi dapat dilakukan seperti misalnya dalam operasi sbb :

$x.a \leftarrow 211$

yang berarti : isilah subvariabel a dari variabel x dengan nilai 211.

atau misalnya :

write(layar) $x.a, x.b$

- Cara ke 2 : pendefinisian terlebih dahulu suatu jenis data abstrak (yang dibangun dari jenis data elementer). Sesudah itu baru didefinisikan suatu variabel dengan jenis data abstrak tersebut.

contoh :

Misalnya didefinisikan suatu jenis data yang disebut " ss ", yang terdiri dari 2 bagian sbb :

- bagian ke 1 berjenis karakter tunggal, disebut a .
- bagian ke 2 berjenis integer tunggal, disebut b .

Kemudian didefinisikan suatu variabel bernama x dari jenis data ss tersebut.

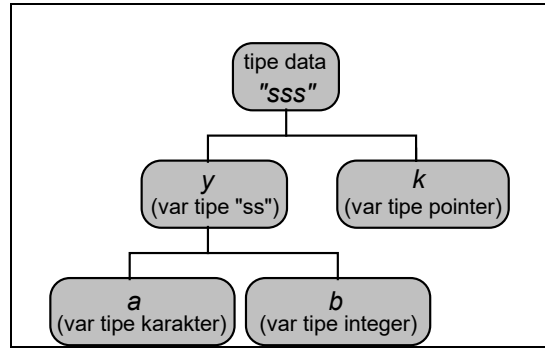
Maka, contoh referensi seperti pada cara ke 1 dapat pula dipergunakan dengan maksud dan efek yang sama.

Dapat pula dibentuk suatu tipe data abstrak yang mengandung hirarki. Sebagai contoh :

Tipe data "sss" dibangun dari 2 bagian sbb:

- bagian ke 1 berjenis "ss" (lihat contoh sebelumnya, terdiri dari 2 bagian, *a* dan *b*), tunggal dan disebut sebagai *y*.
- bagian ke 2 berjenis *pointer*, tunggal dan disebut *k*.

maka akan terlihat adanya hirarki seperti pada gambar.



Gambar 5. Hirarki tipe komposit

Jika ada 2 buah variabel bernama *z* dan *zz* yang berjenis "sss", maka rujukan dibawah ini sah :

$z.y \leftarrow zz.y$
 $z.k \leftarrow zz.k$
 $z.y.a \leftarrow zz.y.a$
 $z.y.b \leftarrow zz.y.b$

Dan operasi dibawah inipun sah :

$z.y \leftarrow zz.y$
 $z.k \leftarrow \text{address}(zz)$
 $z.y.a \leftarrow zz.y.a$
 $zz.y.a \leftarrow z.y.a + 2$

Dalam pemrograman, data yang diproses seringkali sangat kompleks, sedangkan struktur dalam tempat penyimpanan komputer sangat sederhana. Karena itu dibutuhkan suatu "abstraksi" dari perepresen-tasian data yang kompleks, supaya pemrogram dapat memusatkan struktur datanya terhadap "logik" dari program.

Maka dikenal tiga struktur data, yaitu :

1. Definisi Fungsional, yaitu pendefinisian struktur data dan operator-operatornya yang berlaku pada struktur tersebut.
2. Representasi Logik, yaitu spesifikasi dari struktur, yang menyangkut nama type dan spesifikasi semua operator.
Representasi logik tidak bergantung pada memori komputer. Struktur ini memudahkan pemrogram untuk merancang data dan algoritma.
3. Representasi (implementasi) fisik, yaitu spesifikasi dari struktur data sesuai dengan implemantasinya dalam memory komputer.

Pada dasarnya hanya ada dua macam implementasi fisik, yaitu :

- Representasi Kontigu, yaitu sekumpulan data yang penempatannya dalam memory benar-benar secara fisik adalah kontigu, setiap elemen ditaruh secara berurutan dengan elemen lainnya. Struktur ini biasa disebut dengan struktur yang statis.
- Representasi fisik berkait , yaitu sekumpulan data yang penempatannya pada memory komputer dapat terpencar-pencar , namun dapat ditelusuri berkat adanya informasi berupa alamat yang menghubungkan elemen yang satu dengan yang lainnya. Struktur ini biasa disebut dengan struktur yang dinamis.

1.7 Struktur Data Abstrak

Adalah struktur data dengan definisi fungsional dan representasi logik tertentu, yang implementasi fisiknya disesuaikan dengan persoalannya. Struktur data abstrak ini diperlukan untuk mempermudah merancang struktur data dari suatu persoalan dalam terminologi yang lebih mudah dimengerti oleh manusia, namun sulit direpresentasikan langsung oleh mesin riil.

Contoh struktur data abstrak yang “standard” di dalam bidang informatika

- List Linier
- Multi Linked List
- Antrian (Queue)
- Tumpukan (stack)
- Pohon (tree)
- Graph

1.8 Kesimpulan

Dalam pemrograman, data yang diproses seringkali sangat kompleks, sedangkan struktur dalam tempat penyimpanan komputer sangat sederhana. Karena itu dibutuhkan suatu “abstraksi” dari perepresentasian data yang kompleks, supaya pemrogram dapat memusatkan struktur datanya terhadap “logik” dari program.

Programmer profesional harus mampu berpikir kritis, detail, dan algoritmis dalam konteks representasi, deklarasi, dan operasi dalam komputasi menyelesaikan masalah. Komputasi dapat dikategorikan dalam bentuk: matematis, grafik, text, voice, image, dan multimedia.

(Samaved)

Data Structures + Algorithms = Programs

(Niklaus Wirth)

Dalam menyusun representasi data sesuai dengan kebutuhan dunia nyatanya, maka dikenal tiga struktur data berikut :

1. Definisi Fungsional, yaitu pendefinisian struktur data dan operator-operatornya yang berlaku pada struktur tersebut.
2. Representasi Logik, yaitu spesifikasi dari struktur, yang menyangkut nama type dan spesifikasi semua operator.
Representasi logik tidak bergantung pada memori komputer. Struktur ini memudahkan pemrogram untuk merancang data dan algoritma.
3. Representasi (implementasi) fisik, yaitu spesifikasi dari struktur data sesuai dengan implemantasinya dalam memory komputer.

Pada dasarnya hanya ada dua macam implementasi fisik, yaitu :

- Representasi Kontigu, yaitu sekumpulan data yang penempatannya dalam memory benar-benar secara fisik adalah kontigu, setiap elemen ditaruh secara berurutan dengan elemen lainnya. Struktur ini biasa disebut dengan struktur yang statis.

- Representasi fisik berkait , yaitu sekumpulan data yang penempatannya pada memory komputer dapat terpencar-pencar , namun dapat ditelusuri berkat adanya informasi berupa alamat yang menghubungkan elemen yang satu dengan yang lainnya. Struktur ini biasa disebut dengan struktur yang dinamis.

1.9 Studi Kasus Representasi, Deklarasi, dan Operasi

Pra studi kasus, peserta akan diberikan pemahaman representasi, deklarasi, dan operasi untuk setiap kategori tipe data : primitif, komposit, dan user defined data types.

Studi kasus diarahkan dengan memanfaatkan hasil pemahaman tipe data primitif : integer, real, character, boolean, dan pointer serta tipe data komposit : string, enumerasi, array, dan record terhadap modus pemrosesan data matematis, grafik, dan pemrosesan teks sesuai dengan struktur kendali algoritma yang sudah diberikan di materi Dasar-dasar Pemrograman.