

Pertemuan 01

Tipe dan Struktur Data

Pada akhir pertemuan ini, diharapkan mahasiswa akan mampu :

- menerangkan konsep Tipe Data & Struktur Data
- memberikan contoh Tipe Data & Struktur Data

MATERI

- Terminologi
- Kelas tipe data
- Level abstraksi tipe data
- Contoh

Terminologi

- Data : segala sesuatu yg terukur yg bisa diolah menjadi informasi.
- Struktur: Pengaturan atau hubungan antar elemen dalam suatu sistem.
- Nilai Data: Suatu data yg dipandang sbg kesatuan tunggal (single entity).
- Tipe Data merupakan kombinasi antara:
 - ♦ Himpunan nilai data (Set of value)
 - ♦ Himpunan operasi terhadap nilai-nilai data tersebut (Set of operation).
- Contoh deklarasi :

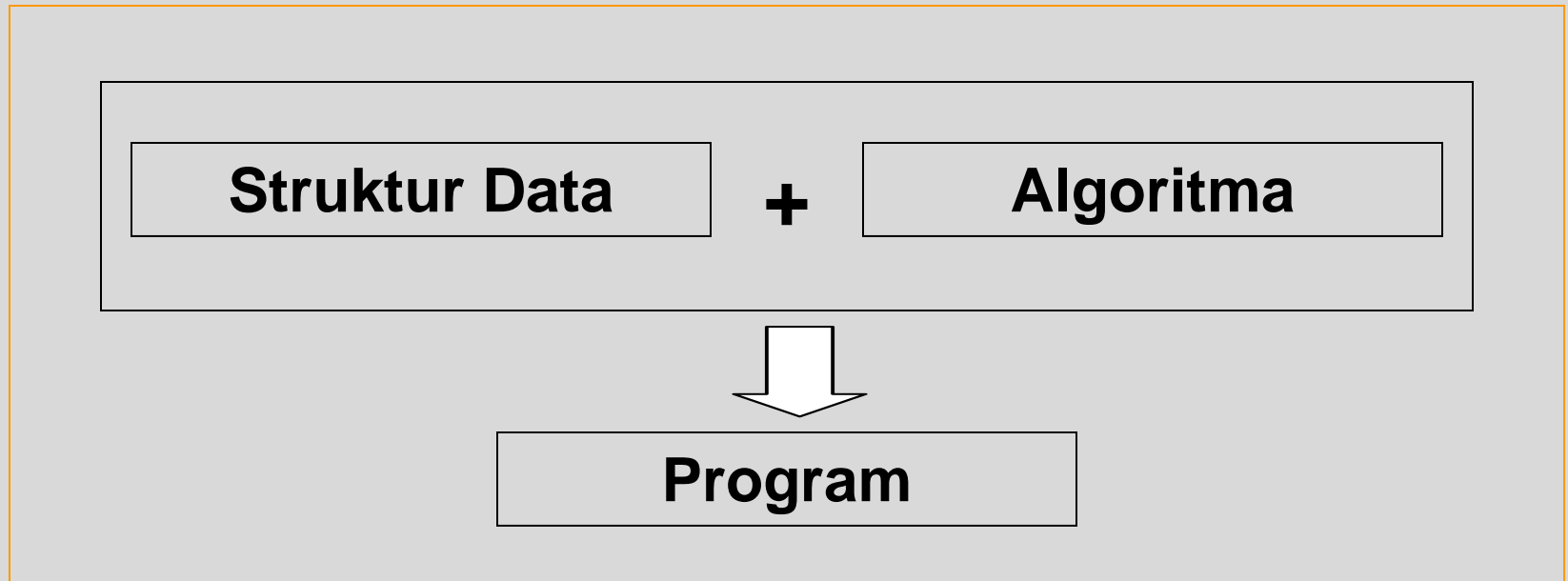
```
int X;  
char C;
```

X : menyimpan nilai bilangan bulat
(integer)
operasinya a.l.: tambah, kali, bagi

C : menyimpan nilai karakter
Operasinya a.l.: gabung,
salin

Hubungan

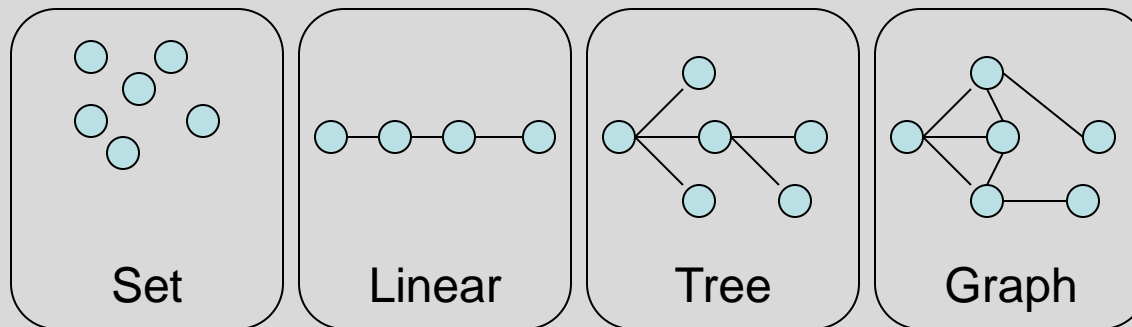
- Hubungan antara Struktur Data, Algoritma, dan Program dapat digambarkan s.b.b.:



Struktur Dasar Hubungan Elemen

Empat struktur dasar hubungan antar elemen :

- Set
- Linear
- Hierarki (tree)
- Graph (network)



Kelas Tipe Data

Kelas Tipe Data :

- ♦ Atomik
- ♦ Berstruktur / Struktur Data

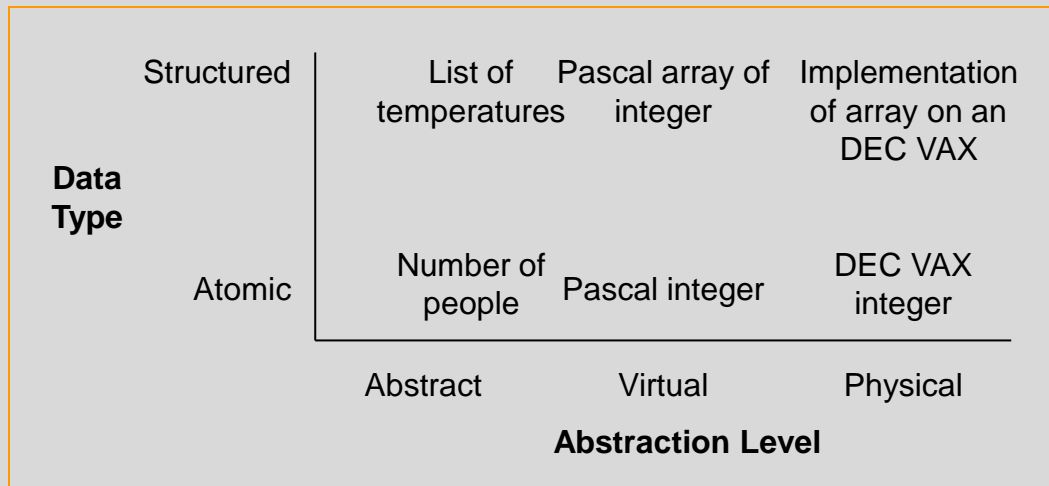
Tipe Data Atomik : Suatu data yg dipandang sbg satu kesatuan tunggal dan tdk dpt dipecah-pecah lagi (non decomposable entity). Contoh : Integer, Char, dll.

Tipe Data Berstruktur : Suatu data yg dipandang sbg satu kesatuan tunggal dan dpt dipecah-pecah lagi (decomposable entity). Contoh : Array, Structure, dll.

Level Abstraksi Tipe Data

■ Level Abstraksi Tipe Data :

- Abstrak : hasil imajinasi
- Virtual : ada dalam virtual processor/bahasa pemrograman
- Fisik : ada scr fisik/nyata dalam main processor



Tipe Data Abstrak

“ADT is a data type that organized in such a way that the specification of the objects and the specification of the operations on the objects is separated from the representation of the objects and implementation of the operations”.

Karakteristik

- ♦Modular
- ♦Penyembunyian Informasi (Information hiding)

Keuntungan

- ♦Penyederhanaan Masalah (Simplicity)
- ♦Integritas (Integrity)
- ♦Kebebasan Pengimplementasian (Implementation Independence)

Desain & Spesifikasi TDA

Horowitz, E., Sahni, S., Freed, S.A. 1993, **Fundamentals of Data Structures in C**, Computer Science Press, P11-21

ADT Natural_Number

Structure Natural_Number is

Objects : an ordered subrange of the integers starting at zero and ending at the maximum integer (*INT_MAX*) on the computer

function :

For all $x, y \in \text{Nat_Number}$; $\text{TRUE}, \text{FALSE} \in \text{Boolean}$ and where $+$, $-$, $<$, and $==$ are the usual of integer operations

Nat_No Zero() ::= 0

Boolean Is_Zero(X) ::= if (X) return TRUE else return FALSE

Nat_No Add(x,y) ::= if $((x+y) \leq \text{INT_MAX})$ return $x+y$ else return *INT_MAX*

Boolean Equal(x,y) ::= if $(x==y)$ return TRUE else return FALSE

Nat_No Successor(x) ::= if $(x==\text{INT_MAX})$ return *INT_MAX* else return $x+1$

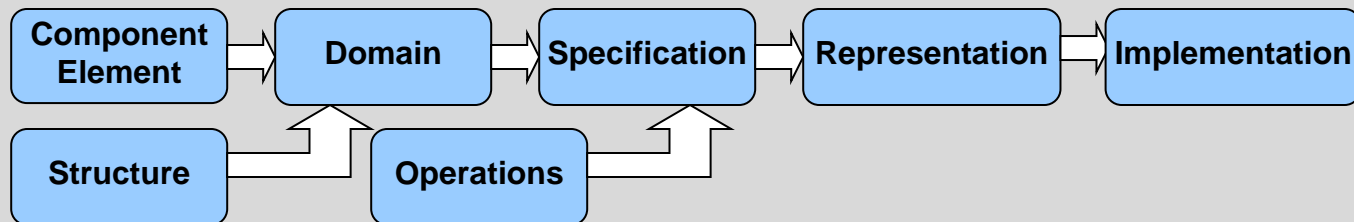
Nat_No Substruct(x,y) ::= if $(x < y)$ return 0 else return $x-y$

In Fact, the nature of and ADT argues that we avoid implementation detail. Therefore, we will usually use a form of structured English to explain the meaning of the function

Desain & Spesifikasi TDA (2)

Daniel F. Stubbs & Neil W. Webre (1985). ***Data Structure with Abstract Data Types and Pascal***. Book / Cole Publishing Company, P11-21

□ Struktur :



Contoh Spesifikasi TDA

Spesifikasi untuk Tipe Data Abstrak *letterstring* :

Elements :The component elements are the characters 'a'-'z','A'-'Z', and the space character. We refer to them as letters.

Structure :There is a linear relationship (structure) among the letters in each value of the letter string.

Domain : There are between 0 and 80 letter in any such letterstring. The domain of the type *letterstring* is all such letter strings that satisfy these rules.

Operatio
n: In specifying the operations, we occasionally have to refer to value of letter string before and after execution of an operation. We call the former s-pre and the letter s-post.

Contoh Spesifikasi TDA (2)

letter leftletter(letterstring *s)

pre - The number of letters in the input letter string s greater than 0.

post -leftletter is the first (leftmost) letter in the input letter string (s-pre). s-post is s-pre less its leftmost letter.

append(letter l; letterstring *s)

pre - The number of letters in s-pre is less than 80.

post -The string s (s-post) is longer by one letter than s-pre, and the letter in l is its new last (rightmost) letter.

boolean empty(letterstring s)

pre - None.

post -If s contains 0 letters than empty is true else empty is false.

boolean full(letterstring s)

pre - None.

post -If s contains 80 letters than full is true else full is false.

reverse (letterstring *s)

pre - None.

post -The sequence of the letters in the string is reversed so that the first and last have changed places, the second and next-to-last have changed places, and so on.

Contoh Spesifikasi TDA (3)

▪Representasi

```
struct letterstring {  
    int n;  
    letter str[80];  
}
```

▪Implementasi

```
void reverse ( letterstring *s ) {  
    int i; letterstring temp;  
    for(i=0;i<80;i++) temp.str[i] = temp.str[80-i+1];  
    for(i=0;i<80;i++) s.str[i] = temp.str[i];  
}
```

Selesai