



MODUL II Struktur Data

Judul	STRUKTUR DATA DAN TYPE DATA ABSTRAK	
Penyusun	Distribusi	Perkuliahan
Nixon Erzed	Teknik Informatika Universitas Esa Unggul	Pertemuan – II online

Tujuan :

1. Mahasiswa memahami pengertian dan posisi struktur data dalam pemrograman
2. Mahasiswa mengenal dan memahami tipe data dan urgensi representasi data dalam pemrograman
3. Mahasiswa memahami representasi data dan dapat mendeklarasikan struktur data secara lengkap

Materi:

1. Definisi Formal Struktur Data
2. Pembuatan Struktur Data
3. Spesifikasi Non Formal
4. Pengertian Deklarasi
5. Deklarasi Type dan Function
6. Deklarasi Struktur Data

Referensi :

1. Algoritma dan Pemrograman dalam Bahasa Pascal, C, dan C++, Rinaldi Munir 2016
2. Struktur Data, Bambang Hariyanto, Informatika, Bandung, 2003
3. Algoritma dan Struktur Data 1 & 2 , Moh. Syukani, Mitra Wacana Media, 2012
4. Fundamental of Data Structure, Ellis Horowitz, Pitman International Text, 1978

STRUKTUR DATA

Perhatikan rantai berikut : Type data – objek data – struktur data

Type Data

Type data adalah jenis data yang ditangani oleh bahasa pemrograman. Tiap bahasa pemrograman memiliki kumpulan type data *built-in*, sehingga :

- Memungkinkan deklarasi variabel bertipe tersebut.
- Menyediakan kumpulan operasi untuk manipulasi variabel bertipe tersebut.

Beberapa type data dasar sudah disediakan oleh pemroses, sehingga secara mudah bahasa pemrograman mengadopsinya, yaitu : *boolean, integer dan real*. Sedangkan type lain merupakan type bentukan yang memerlukan rutin khusus untuk mengimplementasikannya dalam bahasa pemrograman, misalnya : *string, character dan lain-lain*.

Contoh

Bahasa Pascal : *boolean, char, integer, real, string*
Bahasa C : *int, short int, long int, float, double, character, string*
Bahasa Fortran : *character, integer, real, logical, double precision*
Bahasa Snobol : *string, character*
Bahasa Lisp : *list (ekspresi S)*

Ukuran dan jangkauan data dari suatu tipe data tergantung dari [kompilator](#) dan [komputer](#) yang digunakan. Berikut adalah jangkauan dan ukuran tipe integer pada beberapa bahasa pemrograman.

Objek Data

Objek data adalah mengacu kumpulan elemen D (Domain)

Misalnya :

Objek data *integer* mengacu domain berikut :

$$D = \{ 0, \pm 1, \pm 2, \pm 3, \dots \}$$

Objek data string karakter dengan panjang kurang dari 3 mengacu kepada domain berikut:

$$D = \{ ' ', 'A', 'B', \dots, 'Z', 'AA', \dots, 'ZZ', 'AAA', 'BBB', \dots \}$$

D dapat berjumlah berhingga atau tak berhingga. Jika D sangat besar maka diperlukan cara khusus untuk merepresentasikan elemen-elemen Domain D di komputer.

Struktur Data

Untuk membuat menjadi struktur data, kita harus melakukan aktifitas terhadap objek data, yaitu :

- Mendeskripsikan kumpulan operasi sah yang diterapkan terhadap elemen-elemen objek data
- Menunjukkan mekanisme kerja operasi-operasi

Secara terstruktur : struktur data adalah objek data ditambah himpunan operasi untuk memanipulasi objek.

$$\text{Struktur data} = \text{objek data} + \{\text{operasi manipulasi}\}$$

Operasi manipulasi harus dapat menjamin terpenuhinya spesifikasi relasi antar elemen-elemen data atau menjamin invarian.

Contoh :

1. Struktur data integer:

$$\begin{aligned} \text{Integer} : D & \{ 0, \pm 1, \pm 2, \pm 3, \dots \} \\ \text{Operasi} & = \{ +, -, *, \text{mod}, \text{div} \} \end{aligned}$$

Operasi manipulasi +, -, * menjamin terpenuhinya spesifikasi relasi antar elemen-elemen data integer. Sementara itu operasi bagi (/) tidak dapat menjamin spesifikasi relasi dan invarian, karena hasil operasi bagi (/) antara 2 integer tidak lagi menghasilkan type real, bukan integer. Operasi yang memenuhi syarat adalah DIV mengambil nilai bulat pembagian dan MOD mengambil sisa pembagian.

2. Struktur Data Waktu : $t = (j : m : d)$

$$\begin{aligned} \text{Waktu} : \text{Domain } D & \{ j, m, d \mid 0 \leq j \leq 23, 0 \leq m \leq 59, 0 \leq d \leq 59, j, m, d \in \mathbb{Z} \} \\ \text{Operasi} & \{ \text{totaldetik} = f(t), \text{lama} = f(t_1, t_2) \} \end{aligned}$$

DEFINISI FORMAL STRUKTUR DATA

Struktur data mendeskripsikan :

- Kumpulan Objek data
- Operasi dasar terhadap objek data
- Hubungan (relasi) antar objek data.

Secara formal :

Struktur data dapat dinyatakan sebagai *triple* (D, F, A) , yaitu :

- a. Domain $D \rightarrow d \in D$
- b. Himpunan fungsi F ,
yaitu kumpulan operasi-operasi yang legal terhadap anggota domain D
- c. Himpunan aksiom A ,
yaitu kumpulan relasi atau invarian yang dipenuhi oleh anggota di domain D .

Triple (D,F,A) : disebut tipe data abstrak/TDA (*Abstract Data Type*), karena implementasinya tersembunyi dari pemakai, baik implementasi datanya maupun implementasi operasinya.

TDA \rightarrow pemodelan matematis terhadap objek data

TDA merupakan pemodelan matematis terhadap objek data dengan kumpulan operasi yang didefinisikan pada model tersebut. TDA merupakan generalisasi tipe data primitif.

Kegunaan spesifikasi struktur data yang digambarkan dengan TDA adalah :

- Untuk membuktikan kebenaran program yang menggunakan struktur data.
- Untuk membuktikan kebenaran implementasi struktur data.

Tujuan pembentukan struktur data adalah untuk mengkapsulkan informasi, yaitu :

- Perubahan implementasi struktur data tidak mengubah program yang menggunakan struktur data bila interface pada struktur data tersebut dirancang secara luwes sehingga independen terhadap program.
- Dengan pembakuan interface, pemakaian dan pembuatan struktur data dapat dilakukan secara terpisah.
- Struktur data adalah sarana pemrograman modular yang dapat dijadikan basis pembentukan team programmer.

Contoh :

1. TDA Type Integer

Domain $D = \{ 0, \pm 1, \pm 2, \pm 3, \dots \}$

Function $F = \{ +, -, *, \text{mod}, \text{div} \}$

Aksioma $A = \{ \}$

2. TDA Type String

Domain $D = \{ s \mid s[x] = \text{string ASCII}, 1 < x < 256 \ \& \ x \in \mathbb{Z} \}$

Function $F = \{ \text{penggalan} = f(s[x], n, m) \}$

Aksioma $A = \{ \}$

3. TDA Type Jam

Domain $D = \{ j, m, d \mid 0 \leq j \leq 23, 0 \leq m \leq 59, 0 \leq d \leq 59, j, m, d \in \mathbb{Z} \}$

Function $F = \{ \text{Lama} = f_1(t_0, t_1), \text{TotalDetik} = f_2(t) \}$

Aksioma $A = \{ \text{perhitungan lama dikoreksi jika tangga berbeda} \}$

4. TDA Type Nilai Matakuliah UMB

Domain $D = \{ n, \text{abs} \mid 0 \leq n \leq 100, 0 \leq \text{abs} \leq 14, n \in \mathbb{R} \ \& \ \text{abs} \in \mathbb{Z} \}$

Function $F = \{ n\text{-uts} = f_1(\text{mhs}^{(i)}, n),$

$n\text{-uas} = f_2(\text{mhs}^{(i)}, n),$

$n\text{-tugas} = f_3(\text{mhs}^{(i)}, n),$

$n\text{-prakt} = f_4(\text{mhs}^{(i)}, n),$

$n\text{-abs} = f_5(\text{mhs}^{(i)}, n)$

$n\text{-akhir} = f_6(n\text{-uts}, n\text{-uas}, n\text{-tugas}, n\text{-prakt}, n\text{-abs}) \}$

Aksioma $A = \{ n\text{-abs} \leq 75 \Rightarrow n\text{-akhir} = \text{"E"} \}$

Soal Latihan :

1. Sajikan TDA Type Tanggal

2. Anda dihadapkan pada masalah **pengolahan data transaksi** di kasir sebuah super market. Barang dikodekan dengan barcode 12 digit

- Definisikan objek-objek yang merepresentasikan persoalan.
- identifikasi type data yang diperlukan, dan sajikan TDA nya

3. Anda dihadapkan pada persoalan pengolahan data parkir mobil di area Blok Z Plaza, untuk mengetahui berapa biaya yang harus dibayar.

- Definisikan objek-objek yang merepresentasikan persoalan.
- identifikasi type data yang diperlukan, dan sajikan TDA nya

IV. PEMBUATAN STRUKTUR DATA

Terdapat tiga tahap pembuatan struktur data :

- Tahap pertama : spesifikasi
- Tahap ke dua : implementasi
- Tahap ke tiga : pemrograman

Tahap Pertama : Spesifikasi

Pendeskripsian atau spesifikasi struktur data yang merupakan batasan-batasan struktur data. Pendefinisian melibatkan logik sehingga sering digunakan ketetapan/teorema matematika untuk menyatakan sifat-sifat struktur data yang dikehendaki.

Spesifikasi dapat disajikan dengan dua cara :

- Spesifikasi secara formal
Diwujudkan dalam bentuk Type Data Abstrak
- Spesifikasi secara informal
Diwujudkan dengan membuat deskripsi dalam bahasa alami.

Tahap ke dua : Implementasi

Implementasi menyatakan cara penerapan struktur data dengan struktur data yang telah ada. Implementasi struktur data merupakan proses pendefinisian type data abstrak sehingga semua operasi dapat diekspresikan dengan fungsi-fungsi yang dapat dieksekusi oleh komputer.

Implementasi berisi deklarasi struktur penyimpanan item-item data serta algoritma-algoritma untuk implmentasi operasi-operasi yang menjadi karakteristik struktur data.

Implementasi struktur data adalah pemetaan fungsi **d** ke kumpulan struktur data lain, **e**, yaitu :

- Pemetaan domain : merepresentasikan objek **d** dengan objek **e**
- Pemetaan Fungsi **n** : menggunakan fungsi-fungsi pada struktur data **e**

Tahap ke tiga : Pemrograman

Pemrograman struktur data adalah penterjemaahan menjadi pernyataan dalam bahasa pemrograman tertentu.

Terdiri dari proses :

- Deklarasi yang mendefinisikan objek-objek data dan hubungannya

- Pembuatan prosedur atau rutin untuk operasi-operasi dasar
- Perancang harus memilih tipe-tipe data yang telah ada untuk merepresentasikan struktur data.
- Struktur data dibangun menggunakan fasilitas pembentukan struktur data yang disediakan bahasa pemrograman.

Spesifikasi Non Formal

Untuk kebutuhan praktis, struktur data dapat didefinisikan dengan bahasa alami (bukan matematis), yang melibatkan pendefinisian komponen-komponen berikut:

Deklarasi struktur data dengan tipe yang telah ada

Operasi-operasi terhadap tipe data sehingga dapat mengakomodir fungsi atau aksi yang dikehendaki.

Pembentukan struktur data adalah pembentukan tipe data lengkap, dengan properti sebagai berikut :

- Nama : sebagai identifier tipe data
- Domain : himpunan semesta nilai bertipe data
- Penyebutan anggota : cara penyebutan anggota-anggota tipe data
- Operasi terhadap tipe : daftar operasi terhadap anggota tipe data sesuai spesifikasi

DEKLARASI STRUKTUR DATA

I. PENGERTIAN DEKLARASI STRUKTUR DATA

Deklarasi struktur data adalah suatu pernyataan formal tentang representasi data dalam pemrograman. Terdapat dua tahapan deklarasi :

- Deklarasi Type Data
- Deklarasi Struktur Data

a. Deklarasi Type Data

Deklarasi type data dimaksudkan untuk mengkapsulkan informasi dasar tentang suatu type dalam suatu batasan objek (domain) dan operasi yang legal untuk type tersebut. Tujuan deklarasi type data, agar pemrogram tidak dirumitkan dengan penulisan secara berulang detil fungsi-fungsi untuk suatu type data.

Pada awalnya hanya tersedia type-type data yang didefinisikan secara built-in oleh bahasa pemrograman. Type built-in berorientasi pada cara data disimpan dalam sistem memory komputer, pada keadaan tertentu hanya memiliki hubungan yang kecil dengan permasalahan yang akan diselesaikan. Hal ini disebabkan karena type tersebut tidak didefinisikan dengan orientasi domain persoalan yang dihadapi pemrograman.

Untuk mengadaptasi berbagai keunikan persoalan dan kebutuhan pemrogram, diperlukan fasilitas untuk dapat mengekspresikan dan mendefinisikan type data sesuai dengan domain persoalan. Bahasa pemrograman yang ada, saat ini telah menyediakan fasilitas yang memungkinkan pemrogram mendefinisikan sendiri type-type data yang mendekati abstraksi type pada domain persoalan.

b. Deklarasi Struktur Data

Deklarasi struktur data mengacu pada objek permasalahan yang akan diselesaikan dengan suatu program/aplikasi. Dalam hal ini terfokus pada tata cara mewadahi data-data yang merupakan representasi permasalahan. Wadah data diimplementasikan dalam bentuk pendeklarasian variabel-variabel.

Deklarasi struktur data dilakukan setelah menganalisa himpunan data/objek dan himpunan kuasanya (*power set*). Analisa terhadap data/objek akan menghasilkan pengklusteran/pengelompokan data/objek.

Untuk objek data yang direpresentasikan oleh item data tunggal, akan dideklarasikan dengan menggunakan variabel bertipe data tunggal. Sedangkan objek data yang membutuhkan sekumpulan item data (bundel data) diimplementasikan dalam deklarasi type record.

II. DEKLARASI TYPE DAN FUNCTION

Untuk type-type dasar sudah dideklarasikan oleh bahasa pemrograman, misalnya :

Type integer

Type Integer berlaku bagi objek data bilangan bulat $\rightarrow 0, \pm 1, \pm 2, \dots$

Terhadap objek data ini hanya diizinkan operasi $+$, $-$, $$, DIV , dan MOD*

Operasi matematis lainnya akan ditolak. Terdapat pengecualian jika type tersebut hanya sebagai operand (mengambil nilai / get value / read only), misalnya :

A & B adalah integer,

Operasi bagi ($/$) diizinkan selama hasilnya disimpan ke variabel lain yang bertipe real $\rightarrow C \leftarrow A / B$ dimana C adalah Real

Contoh lain adalah type String.

Secara formal type integer dan type string disajikan dengan TDA sebagai berikut :

- o TDA Type Integer

Domain $D = \{ 0, \pm 1, \pm 2, \pm 3, \dots \}$

Function $F = \{ +, -, *, \text{mod}, \text{div} \}$

Aksioma $A = \{ \}$

- o TDA Type String

Domain $D = \{ s \mid s[x] = \text{string ASCII}, 1 < x < 256 \ \& \ x \in \mathbb{Z} \}$

Function $F = \{ \text{penggalan} = f(s[x], n, m) \}$

Aksioma $A = \{ \}$

Begitu juga untuk beberapa type rakitan, sudah banyak dideklarasikan oleh bahasa pemrograman. Misalnya type *date*/tanggal dan type *time*/waktu saat ini sudah built-in dalam bahasa pemrograman.

Yang harus diingat ketika sebuah type data didefinisikan, maka operasi-operasi yang melekat pada type tersebut juga harus didefinisikan. Berikut ini akan diberikan beberapa contoh Deklarasi Type dan Function.

5. Type Jam

Data jam merupakan kombinasi hirarki 3 komponen yang basisnya adalah bilangan bulat dengan *range* tertentu, yaitu Jam, Menit, Detik. Domain untuk masing-masing komponen data memiliki range berbeda. Terhadap Jam, operasi yang umum dilakukan adalah : menghitung selisih 2 waktu (P/lama), menghitung nilai detik (TS) dari data jam (data skalar), mengambil komponen jam (Hr) dari data jam, mengambil komponen menit (Mn) dan mengambil komponen detik (Sc). Operasi selisih 2 waktu hanya untuk waktu pada hari/tanggal yang sama.

Agar lebih jelas, berikut ini didefinisikan dengan TDA Type Jam, yaitu:

$$\text{Domain } D = \{ j,m,d \mid \begin{array}{l} 0 \leq j \leq 23, \\ 0 \leq m \leq 59, \\ 0 \leq d \leq 59, \\ j,m,d \in \mathbb{Z} \end{array} \}$$

$$\text{Function } F = \{ \begin{array}{l} \text{Periode} = f1(t_0, t_1), \\ \text{TotalDetik} = f2(t), \\ \text{DJam} = f3(t), \\ \text{DMenit} = f4(t), \\ \text{DDtk} = f5(t) \end{array} \}$$

$$\text{Aksioma } A = \{ \text{perhitungan lama dikoreksi jika tangga berbeda} \}$$

Deklarasi Type Jam

a. Type

```
jm = [0...23]
ms = [0...59]
wkt = record of
    |   j : jm
    |   m : ms
    |   d : ms
end-record
```

b. Function

Function TotalDetik (t : wkt) : Integer

Begin

```
    |   TotalDetik ← (t.j * 3600) + (t.m * 60) + t.d
```

End TotalDetik

Function Periode(w0, w1 : wkt) : wkt

Var dt0, dt1, dt : integer

Begin

dt0 \leftarrow w0.d + w0.m*60 + w0.j*3600

dt1 \leftarrow w1.d + w1.m*60 + w1.j*3600

If w1.j > w0.j

 Then dt \leftarrow dt1 – dt0

 Else dt \leftarrow dt0 – dt1

End-if

Periode.j \leftarrow dt div 3600

Periode.m \leftarrow (dt mod 3600) div 60

Periode.d \leftarrow (dt mod 3600) mod 60

End-waktu

Untuk Function lainnya : DJam = f3 (t), DMenit = f4 (t), dan DDtk = f5 (t), silahkan deklarasikan fuction sebagai latihan.

6. Type Tanggal

Sama seperti data jam, data tanggal juga merupakan kombinasi hirarki 3 komponen yang basisnya adalah bilangan bulat dengan *range* tertentu, yaitu tanggal, bulan, dan tahun. Domain untuk masing-masing komponen data memiliki range berbeda. Terhadap Tanggal, operasi yang umum dilakukan adalah : menghitung selisih 2 tanggal, menghitung jumlah hari(Total_hari) dari data jam (data skalar), mengambil komponen jam (Hr) dari data jam, mengambil komponen menit (Mn) dan mengambil komponen detik (Sc). Operasi selisih 2 waktu hanya untuk waktu pada hari/tanggal yang sama.

Mendefinisikan Type Tanggal

Contoh data \rightarrow 7 / 3 /2007

Identifikasi objek dan masalah-masalahnya

1. Objek

Tanggal \rightarrow 1, 2, 3, 4.. 31

Bulan \rightarrow 1, 2, .. 12

tahun \rightarrow integer positif

2. Operasi yang legal :

menghitung hari → umur
nama bulan

3. aksiom

tahun kabisat, max hari perbulan berbeda-beda

TDA Tanggal

$D : t = \{d \mid 1 \leq d \leq 31, d \in \mathbb{Z}\} \cup$
 $\{m \mid 1 \leq m \leq 12, m \in \mathbb{Z}\} \cup$
 $\{y \mid y \geq 0, y \in \mathbb{Z}\}$

$F : \text{fungsi} = \{ \text{hari} = f(t1, t2), \text{nabul} = g(t.m) \}$

$A : \text{batas} = \{ \text{Max}(t.m = 1,3,5,7,8,10,12) = 31;$
 $\text{Max}(t.m = 4,6,9,11) = 30;$
 $\text{Max}(t.m = 2) = 28;$
 $t.y \bmod 4 = 0 \Rightarrow \text{max}(t.m = 2) = 29 \}$

Deklarasi Type

Type date = [1...31]
month = [1...12]
tgl = record of
 d : date
 m : month
 y : integer
end-record

Deklarasi Fungsi

Function Hari(t1, t2 : tgl) : integer

Var h1, h2, h3, x : integer

Begin

```
Case t1.m of
    1 : h1 ← 365 – t1.d
    2 : h1 ← 334 – t1.d
    :
    12: h1 ← 31 – t1.d
end-case

IF (t1.y mod 4 = 0) and ( t1.m ≤ 2)
then
    h1 ← h1+1
end-IF

x = (t2.y – t1.y – 1)
h2 = x * 365 + x div 4

IF [(x mod 4 = 1) and (t1.y+1 mod 4 = 0)] or
    [(x mod 4 = 2) and ((t1.y+1 mod 4 = 0) or (t1.y+2 mod 4 = 0))] or
    [(x mod 4 = 3) and ((t1.y+1 mod 4 = 0) or (t1.y+2 mod 4 = 0) or (t1.y+3 mod 4 = 0)]
then
    h2 ← h2+1
end-IF

Case t2.m of
    1 : h3 ← t2.d
    2 : h3 ← t2.d + 31
    3 : h3 ← t2.d + 59
    :
    12: h3 ← t2.d +334
end-case

IF (t2.y mod 4 = 0) and (t2.m ≥ 3)
then
    h3 ← h3+1
end-IF

Hari ← h1 + h2 + h3
```

End-function

Function Hari-koreksi(t1, t2 : tgl) : integer

Var h1, h2, h3, x : integer

*{memperhentikan kemungkinan
menghitung hari pada tahun yang
sama }*

Begin

IF t2.y > t1.y

Then

Case t1.m of

1 : h1 \leftarrow 365 - t1.d

2 : h1 \leftarrow 334 - t1.d

:

12: h1 \leftarrow 31 - t1.d

end-case

IF (t1.y mod 4 = 0) and (t1.m \leq 2) then h1 \leftarrow h1+1 end-IF

x = (t2.y - t1.y - 1)

h2 = x * 365 + x div 4

IF [(x mod 4 = 1) and (t1.y+1 mod 4 = 0)] or

[(x mod 4 = 2) and ((t1.y+1 mod 4 = 0) or (t1.y+2 mod 4 = 0))] or

[(x mod 4 = 3) and ((t1.y+1 mod 4 = 0) or (t1.y+2 mod 4 = 0) or (t1.y+3 mod 4 = 0))]

then h2 \leftarrow h2+1

end-IF

Case t2.m of

1 : h3 \leftarrow t2.d

2 : h3 \leftarrow t2.d + 31

:

12: h3 \leftarrow t2.d +334

end-case

IF (t2.y mod 4 = 0) and (t2.m \geq 3) then h3 \leftarrow h3+1 end-IF

Hari \leftarrow h1 + h2 + h3

else

Case t1.m of

1 : h1 \leftarrow 365 - t1.d

2 : h1 \leftarrow 334 - t1.d

:

12: h1 \leftarrow 31 - t1.d

end-case

IF (t1.y mod 4 = 0) and (t1.m \leq 2) then h1 \leftarrow h1+1 end-IF

Case t2.m of

1 : h3 \leftarrow t2.d

2 : h3 \leftarrow t2.d + 31

:

12: h3 \leftarrow t2.d +334

end-case

IF (t2.y mod 4 = 0) and (t2.m \geq 3) then h3 \leftarrow h3+1 end-IF

IF t1.y mod 4 = 0

then Hari \leftarrow h1 + h3 -366

else Hari \leftarrow h1 + h3 -365

end-IF

end-IF

End-function

III. Deklarasi Struktur Data (REVIEW)

Kegunaan spesifikasi struktur data yang digambarkan dengan TDA adalah :

1. Untuk menjamin dan membuktikan kebenaran program yang menggunakan struktur data.
 2. Untuk menjamin dan membuktikan kebenaran implementasi struktur data.
- dengan mengetahui objek data, fungsi, aksioma

Pemrograman dapat diarahkan dalam batasan abstraksi data tersebut saja

Tujuan pembentukan struktur data adalah untuk mengkapsulkan informasi, yaitu :

1. Perubahan implementasi struktur data tidak mengubah program yang menggunakan struktur data bila interface pada struktur data tersebut dirancang secara luwes sehingga data independen terhadap program (prinsip independensi).
2. Dengan pembakuan interface, pemakaian dan pembuatan struktur data dapat dilakukan secara terpisah.
3. Struktur data adalah basis pemrograman modular yang dapat dijadikan dasar pembentukan team programmer atau pembagian kerja.

PEMBUATAN STRUKTUR DATA

Terdapat tiga tahap pembuatan struktur data :

- | | | |
|------------------|----------------|---|
| 1. Tahap pertama | : spesifikasi | } |
| 2. Tahap ke dua | : implementasi | |
| 3. Tahap ke tiga | : pemrograman | |

**SPESIFIKASI, DEKLARASI
STRUKTUR DATA DAN
DEKLARASI FUNGSI**

Tahap Pertama : Spesifikasi

Pendeskripsian atau spesifikasi struktur data yang merupakan batasan-batasan struktur data.

Pendefinisian melibatkan logik sehingga sering digunakan ketetapan/ teorema matematika untuk menyatakan sifat-sifat struktur data yang dikehendaki.

Spesifikasi dapat disajikan dengan dua cara :

1. Spesifikasi secara formal
Diwujudkan dalam bentuk **Type Data Abstrak**
2. Spesifikasi secara non formal
Diwujudkan dengan membuat deskripsi **dalam bahasa alami.**

Tahap ke dua : Implementasi

Implementasi berisi deklarasi struktur penyimpanan item-item data (nama variabel dan typenya), serta algoritma-algoritma (algoritma fungsi) untuk implmentasi operasi-operasi yang menjadi karakteristik struktur data.

Contoh : objek jam pada kasus parkir kendaraan

Objek jam : hh:mm:ss

Type h = [0 ... 23]

m = [0...59]

s = [0...59]

TJam = record

jam : h

mnt : m

dtk : s

end-Tjam

var

Jmasuk, Jkeluar : Tjam

Function lamaparkir (jMasuk, jkeluar : tJam) : Tjam

{asumsi bukan parkir menginap}

Var dtkmasuk, dtkkeluar, dtklama : integer

Begin

Dtkmasuk <= jmasuk.jam*3600 + jmasuk.mnt*60 + jmasuk.dtk

Dtkkeluar <= jkeluar.jam*3600 + jkeluar.mnt*60 + jkeluar.dtk

Dtklama <= dtkkeluar – dtkmasuk

Lamaparkir.jam <= dtklama div 3600

Lamaparkir.mnt <= (dtklama mod 3600) div 60

Lamaparkir.dtk <= (dtklama mod 3600) mod 60

End-function

Implementasi objek nopol kendaraan adalah :

Var nopol : string[9]

Mengacu pada masalah yang harus diselesaikan → tidak memerlukan deklarasi fungsi

Tahap ke tiga : Pemrograman

Pemrograman struktur data adalah penterjemahan menjadi pernyataan dalam bahasa pemrograman tertentu.

Terdiri dari proses :

1. Deklarasi yang mendefinisikan objek-objek data dan hubungannya
2. Pembuatan prosedur atau rutin untuk operasi-operasi dasar

Perancang harus memilih tipe-tipe data yang telah ada untuk merepresentasikan struktur data.

Struktur data dibangun menggunakan fasilitas pembentukan struktur data yang disediakan bahasa pemrograman.