

# **KUG1C3**

## **Dasar Algoritma dan Pemrograman**



## **Enumeration and Record Data Type**

## Data Type

- ▶ Logic / Boolean
  - ▶ Integer
  - ▶ Real
  - ▶ Character
- 
- ▶ Enumerated data type
  - ▶ Record data type

→ **Primitive Data Type**

## Enumerated Data Type

- ▶ A collection of ordered items
- ▶ A data type consisting of a set of named values called elements
- ▶ Enumerated data type variables can only assume values which have been previously declared

## A Deck of Card

- ▶ For example, the four suits in a deck of playing cards may be four enumerators named CLUB, DIAMOND, HEART, SPADE, belonging to an enumerated type named suit



**type**

suit : (club,  
diamond,  
heart,  
spade)

## A Deck of Card

- ▶ The card variable can only be assigned with the value from the value previously designed



### type

```
suit : (club, diamond,  
        heart, spade)
```

### dictionary

```
card1 : suit  
card2 : suit
```

### algorithm

```
card1 ← club  
card2 ← spade  
card1 ← clover //error
```

## Enumeration Keyword

- ▶ To easily access the enumeration value, there are several keyword that can be used
- ▶ First
  - Return the first value of enumerated type
- ▶ Last
  - Return the last value of enumerated type
- ▶ Succ(element)
  - Return the next value from an element (successor)
- ▶ Pred(element)
  - Return the previous value of an element (predecessor)

## List of Day

- ▶ For example, we have an enumerated type of day
- ▶ We can easily access the value with the keyword



**type**

days: (Sunday,  
Monday,  
Tuesday,  
Wednesday,  
Thursday,  
Friday,  
Saturday)

# Enumeration Example

## Try enumeration

### Type

day\_name : (Monday, Tuesday, Wednesday,  
Thursday, Friday, Saturday, Sunday)

### Dictionary

day : day\_name

### Algorithm

```
day ← Wednesday
output(day)
output( succ(day) )
output( pred(day) )

output( first( day_name ) )
output( last( day_name ) )
```

Output

Wednesday  
Thursday  
Tuesday

Monday  
Sunday



## Record Data Type

- ▶ Custom data type to store multiple type of variables in a single variable
- ▶ Used for collects or grouping variables into a particular structure that can contain
  - Heterogeneous Data
  - Multiple Components
  - Various Types
- ▶ For better understanding, let's see the illustration

## A Student

- ▶ Let's say we want to store a student data that has the properties :

- id
- name
- age



### **dictionary**

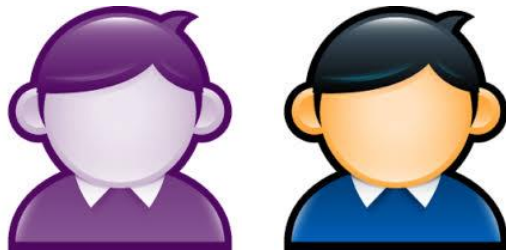
id : string  
name : string  
age : integer

### **Algorithm**

id ← **3101501**  
name ← **Andy**  
age ← **19**

## 2 Students

- ▶ Now, supposedly we have 2 students to store
- ▶ We need 2 variable for each properties



### **dictionary**

id1 : string

id2 : string

name1 : string

name2 : string

age1 : integer

age2 : integer

...

## 2 Students

- ▶ Then we store each data of the 2 students
- ▶ Notice the difficulty in managing a separate variables



...

### Algorithm

```
id1 ← 3101501  
name1 ← Andy  
age1 ← 19
```

```
id2 ← 3101502  
name2 ← Mark  
age2 ← 19
```

## Student Type

- ▶ Thus we can group the properties and collect it into a single custom type called 'student'



```
Type student <  
  id : string,  
  name : string,  
  age : integer  
>
```

## Student Type

- ▶ Now we have a new record data type called student
- ▶ To store students' data, we just need to create a student variable



```
Type student <  
  id : string,  
  name : string,  
  age : integer  
>  
dictionary  
  s1 : student
```

## Student Type

- ▶ Variable `s1` is automatically having elements of `id`, `name`, and `age`
- ▶ To access the element we use dot (`.`)



### **dictionary**

`s1 : student`

### **Algorithm**

`s1.id ← 3101501`

`s1.name ← Andy`

`s1.age ← 19`

## 2 Students

- ▶ Now if we want to store 2 data students, we can just create new variable for student2



### dictionary

s1,s2 : student

### Algorithm

s1.id ← **3101501**

s1.name ← **Andy**

s1.age ← **19**

s2.id ← **3101502**

s2.name ← **Mark**

s2.age ← **19**



## Nested Record Data Type

- ▶ We can also create a nested record data type so a record data type can contain another record data type



```
Type student <  
  id : string,  
  name : string,  
  age : integer  
>
```

## Nested Record Data Type

- ▶ Supposed we want to store the birthdate of the student



```
Type student <  
  id : string,  
  name : string,  
  age : integer  
>  
  day_birth : integer  
  month_birth : integer  
  year_birth : integer  
>
```

## Nested Record Data Type

- ▶ We can collect the element of birthdate into another record data type



```
Type date <  
    day, month, year :  
        integer  
>
```

```
Type student <  
    id : string,  
    name : string,  
    age : integer,  
  
    birthdate : date  
>
```

## Nested Record Data Type

- ▶ Then we can use the same principle to access the birthdate



### **dictionary**

s1 : student

### **Algorithm**

s1.name ← **Andy**

s1.birthdate.day ← **16**

s1.birthdate.month ← **04**

# Question?



## Exercise 0

- A. Definisikan tipe data bentukan Indeks Nilai, dimana nilai yang mungkin untuk Indeks Nilai tersebut adalah A, AB, B, BC, C, D, E, T
- B. Definisikan tipe data bentukan Mahasiswa yang terdiri dari NIM dan nilai bertipe Indeks Nilai yang telah didefinisikan pada bagian A.

## Excercise 1

- ▶ Bilangan kompleks adalah bilangan yang dinyatakan sebagai  $a + bi$  dengan  $a$  dan  $b$  riil sedangkan  $i = \sqrt{-1}$ . Contoh:  $4.0 + 3.8i$ . Tuliskan tipe bentukan bilangan kompleks tsb dengan  $a$  dan  $b$  sebagai *field*-nya.

## Excercise 2

- ▶ Definisikan tipe bentukan untuk kendaraan bermotor roda empat yang diberi nama KB4 yang terdiri atas:
  - Tahun
  - Merk kendaraan yang terdiri atas 'toyota', 'honda', 'hyundai', 'suzuki'.
  - Model



## Excercise 3

- ▶ Definisikan tipe bentukan data pegawai yang diberi nama PEG terdiri atas:
  - Nama
  - Jabatan fungsional: (AA, Lektor, Lektor kepala, Guru besar)
  - NIP
  - Tanggal lahir, yang terdiri atas
    - Tanggal
    - Bulan
    - Tahun

## Excercise 4

- ▶ Didefinisikan tipe bentukan yang mewakili tanggal dalam kalender Masehi. Tanggal dinyatakan sebagai tanggal, bulan, dan tahun. Misalkan tipe bentukan diberi nama TGL.
  - Tanggal memiliki domain  $[1..31]$
  - Bulan memiliki domain  $[1..12]$
  - Tahun memiliki domain  $> 0$

## Excercise 5

- ▶ Didefinisikan tipe bentukan jadwal kereta api yang terdiri atas:
  - No KA
  - Kota asal
  - Jam keberangkatan
  - Kota tujuan
  - Jam kedatangan
- ▶ Jam keberangkatan dan kedatangan dinyatakan sebagai waktu yang terdiri atas jam dan menit.

## Excercise 6

- ▶ Definisikan sebuah **type record pecahan** yang terdiri dari pembilang dan penyebut berupa bilangan integer, kemudian buat flow chart & algoritma yang menerima dua buah **pecahan** dan menghasilkan hasil kali berupa **pecahan!**

## Excercise 7

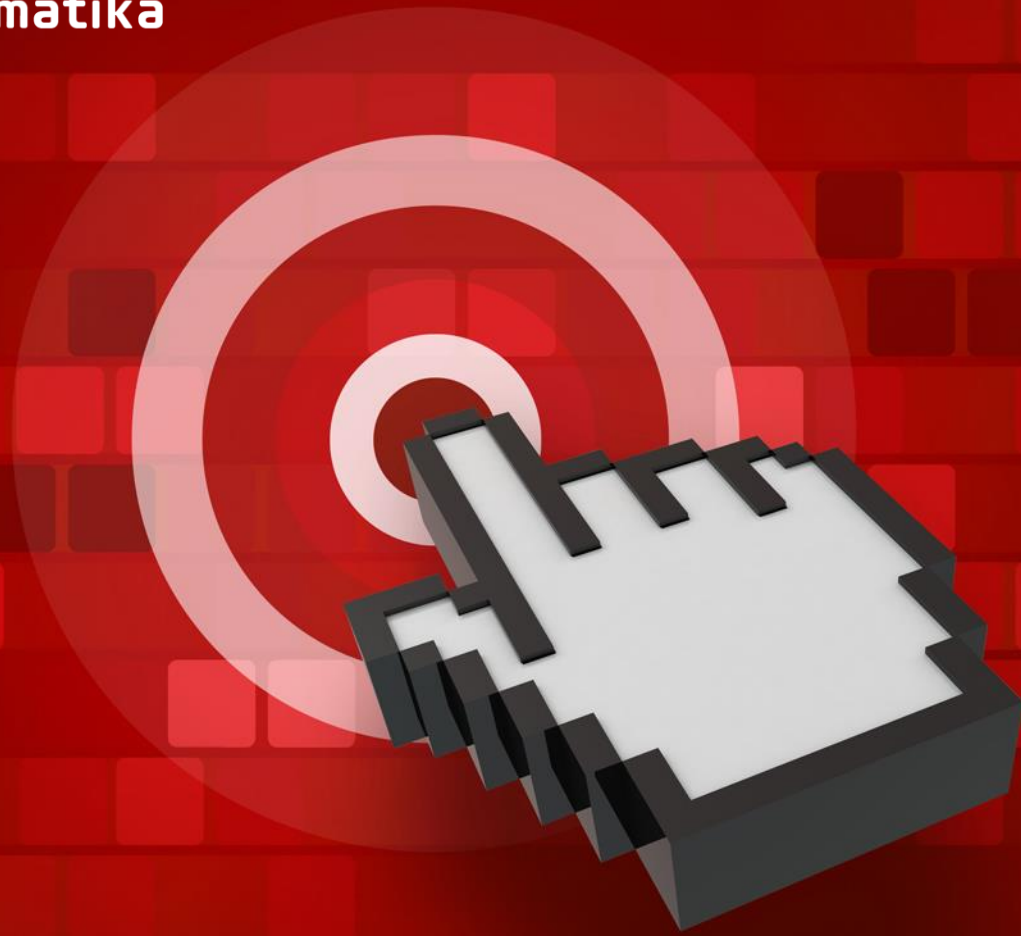
- ▶ Diberikan data siswa berupa **type record** yang terdiri dari nomor induk, nama, nilai ujian, dan nilai rata-rata.

Nilai ujian juga merupakan **type record** yang terdiri dari 3 elemen nilai yaitu nilai matematika, nilai bahasa indonesia, dan nilai bahasa inggris.  
(Nilai = 0 .. 100)

Tuliskan algoritma untuk menghitung nilai rata-rata dari ketiga nilai ujian tersebut!



Fakultas Informatika  
School of Computing  
Telkom University



**THANK YOU**