



# Ekspresi, Tipe Data dan Operator

**Dasar Pemrograman**  
Program Studi S1 Informatika  
Universitas Diponegoro  
Semester Gasal 2020/2021

# Outline



Konsep Fungsi

Tipe Data dan Operator

Ekspresi

Notasi Fungsional

# Paradigma Fungsional

- Didasari oleh konsep pemetaan dan fungsi pada matematika.
- Fungsi dapat berbentuk: fungsi "primitif" atau komposisi.
- Setiap fungsi adalah "kotak hitam", yang menjadi perhatiannya hanya keadaan awal dan akhir.
- Paradigma ini tidak mempermasalahkan memorisasi dan struktur data.
  - Tidak ada pemisahan antara data dan program.
  - Tidak ada istilah "variabel".

# Fungsi dalam Matematika

---

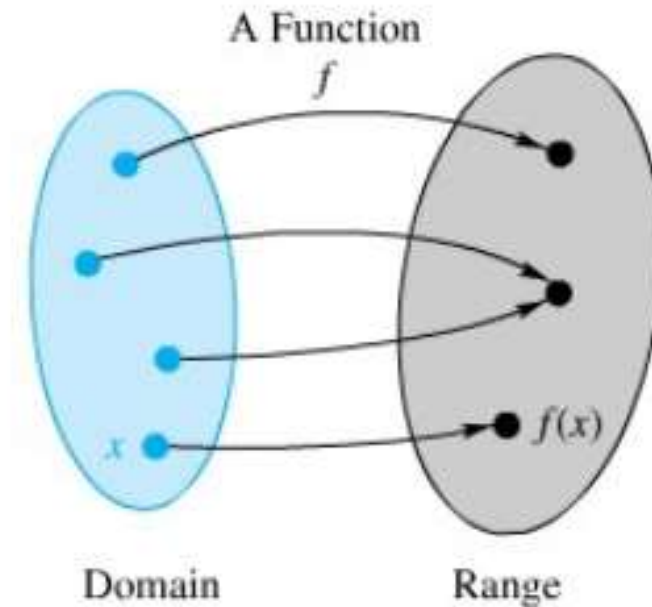
- Fungsi merupakan pemetaan sebuah anggota dari sebuah himpunan A (disebut sebagai domain) ke anggota himpunan B (disebut sebagai range), yang dapat dinotasikan dengan:

$$f : A \rightarrow B$$

$$f(x)$$

- x mewakili sebuah anggota dari himpunan A (domain)  $\rightarrow$  *independent variable*.
- f disebut sebagai nama fungsi.
- $f(x)$  adalah hasil pemetaan fungsi dari himpunan B (range)

## Fungsi dalam Matematika (lanj.)



- Setiap anggota di domain dipetakan tepat satu ke anggota di range.
- Beberapa anggota dapat memiliki pasangan yang sama di range.

## Domain Fungsi

- Saat menspesifikasikan sebuah fungsi, kita juga harus menyebutkan **domain** untuk fungsi tersebut.
- Sebagai contoh sebuah fungsi:

$$f(x) = x^2 - 1$$

mempunyai domain  $\{-1, 0, 1, 2, 3\}$ ,  
maka range-nya adalah  $\{1, 2, 5, 10\}$ .

- Dalam fungsi matematika, jika domain tidak dinyatakan secara eksplisit, maka diasumsikan bahwa domain berasal dari bilangan himpunan bilangan real yang sangat besar → ***natural domain*** dengan syarat tidak menyebabkan pembagian dengan 0 dan akar dari bilangan negatif.

# Domain Fungsi

---

Contoh:

- fungsi  $f(x) = 1 / (x-3)$  mempunyai domain  $\{x : x \neq 3\}$ .
- Fungsi  $g(t) = \sqrt{9 - t^2}$  mempunyai domain dalam interval  $(-3,3)$  atau dapat dituliskan dengan  $\{t : -3 < t < 3\}$ .

Kesimpulan: kita harus menyatakan domain yang sah (valid) untuk sebuah fungsi.

## Fungsi dalam Pemrograman

---

- Fungsi dalam pemrograman digunakan untuk menyelesaikan persoalan tertentu.
- Fungsi menerima masukan, kemudian akan menerapkan serangkaian proses terhadap masukan tersebut agar dapat menghasilkan keluaran sesuai dengan yang diharapkan.
- Fungsi dapat menerima sebuah masukan maupun lebih dari satu masukan.
- Setiap masukan dapat memiliki domain masing-masing.



# Fungsi dalam Pemrograman

Fungsi dengan 1 parameter input

$$f(x)$$
$$f : A \rightarrow B$$

- $x$  : input argument / parameter
- $f(x)$ : output fungsi
- $A$  : domain untuk  $x$
- $B$ : range untuk  $f(x)$

Domain menunjukkan himpunan nilai yang diijinkan untuk  $x$ .

Range menunjukkan himpunan nilai dari output fungsi  $f(x)$

# Fungsi dalam Pemrograman

Fungsi dengan 2 parameter input.

$$f(x, y)$$
$$f : \langle A, B \rangle \rightarrow C$$

- $x, y$  : input argument / parameter
- $f(x, y)$  : output fungsi
- $A$  : domain untuk  $x$
- $B$  : domain untuk  $y$
- $C$  : range untuk  $f(x, y)$

Domain dan range dalam pemrograman dinyatakan dengan tipe data (dan batasan khusus dalam tipe data tersebut).

## Tipe Data

---

**Tipe data** digunakan untuk menyatakan tipe dari sebuah nilai (*value*).

Dalam pemrograman tipe data tidak terbatas pada nilai numerik saja, tapi dapat berupa:

- Tipe data dasar (*primitive data type*):
  - integer
  - real
  - boolean
  - character
  - string
- Tipe data komposisi/ bentukan

## 1. Tipe data integer

---

- Tipe data integer berupa bilangan bulat.
- Contoh: -24, -12, 0, 34, 54546
- Meskipun bilangan bulat dapat memiliki nilai dari negatif tak hingga sampai positif tak hingga, implementasi bilangan integer dalam program dapat memiliki batasan.
- Batasan ini tergantung pada implementasi bahasa pemrograman.

## 2. Tipe data real

---

- Tipe data real berupa bilangan rasional (memiliki angka desimal).
- Contoh: -24.002, -12.01, 0.3, 34.5665
- Meskipun antara 1 dan -1 terdapat tak hingga bilangan rasional, implementasi bilangan real dalam program dapat memiliki batasan.
- Batasan ini tergantung pada implementasi bahasa pemrograman.

## 3. Tipe data boolean

---

- Digunakan untuk merepresentasikan nilai kebenaran (*truth value*).
- Nilai kebenaran dapat berupa True atau False.
- Sebuah fungsi yang menghasilkan keluaran bertipe boolean disebut sebagai **predikat**.

## 4. Tipe data character

- Tipe data character berupa satu karakter tunggal.
- Penulisan character diapik dengan tanda petik tunggal (single quote), yaitu “.”.
- Contoh:
  - ‘A’
  - ‘a’
  - ‘8’ berbeda dengan 8
- Tipe data character dapat juga dikenakan operator perbandingan, misalnya:
  - ‘A’ < ‘a’ jika dievaluasi menghasilkan True
  - ‘A’ = ‘a’ jika dievaluasi menghasilkan False
  - ‘b’ < ‘a’ jika dievaluasi menghasilkan False

## 5. Tipe data string

- Tipe data string terdiri atas kumpulan karakter.
- Penulisan character diapik dengan tanda petik ganda (*double quotes*), yaitu `""`.
- Contoh:
  - `"mahasiswa"`
  - `"AIK123"`
  - `"23"`
- Tipe string dapat juga dikenakan operator perbandingan, misalnya:
  - `"Anak" = "anak"` jika dievaluasi menghasilkan `False`.
  - `"Anak" ≠ "anak"` jika dievaluasi menghasilkan `True`.



## Operator

---

- Operator digunakan untuk mentransformasikan satu atau beberapa nilai untuk menghasilkan sebuah nilai yang baru.
- Operator yang diterapkan pada sebuah nilai bergantung pada tipe datanya.
- Contoh:
  - Operator aritmatika dapat diterapkan pada nilai numerik (integer atau real), namun tidak dapat diterapkan pada nilai bertipe character atau string.

## Operator Dasar

---

- **Operator dasar (*primitive operator*)** adalah operator yang paling sederhana yang telah disediakan oleh pemroses bahasa.
- Fungsi yang paling dasar dalam program fungsional disebut juga sebagai operator.
- Operator dasar terdiri atas:
  1. Operator aritmatika
  2. Operator perbandingan/ relasional
  3. Operator boolean

# 1. Operator aritmatika

| Operator | Keterangan                    | Contoh                 |
|----------|-------------------------------|------------------------|
| +        | Penjumlahan                   | $2 + 4 = 6$            |
| -        | Pengurangan                   | $5 - 1 = 4$            |
| *        | Perkalian                     | $3 * 2 = 6$            |
| /        | Pembagian                     | $5 / 4 = 1.25$         |
| ^        | Pemangkatan                   | $2^4 = 16$             |
| div      | Pembagian bilangan bulat      | $5 \text{ div } 3 = 1$ |
| mod      | Sisa pembagian bilangan bulat | $5 \text{ mod } 3 = 2$ |

- Semua operator tersebut merupakan binary operator.
- Operator – dapat pula menjadi unary operator, misanya: -4 (menandakan bilangan negatif).

## Precedence

- Urutan operator precedence dari tinggi ke rendah:

|                             |                             |
|-----------------------------|-----------------------------|
| $\wedge$                    | exponentiation              |
| $*$ / <b>div</b> <b>mod</b> | the 'multiplying' operators |
| $+$ $-$                     | the 'addition' operators    |

- Contoh:
  - $3^2 * 5 = \dots$
  - $2 + 5 * 2 = \dots$
  - $4 - 2 * 2^2 = \dots$
- Gunakan tanda kurung ( ) untuk mengubah precedence.

## Order of Association

---

- Urutan asosiasi untuk operator yang memiliki precedence sama adalah dari kiri ke kanan, kecuali untuk pangkat adalah kanan ke kiri.
- Contoh:
  - $5 - 2 - 3$  artinya adalah  $(5 - 2) - 3$
  - $2^3^4$  artinya adalah  $2^{(3^4)}$

## div & mod

- $(x \text{ div } y)$  dan  $(x \text{ mod } y)$  dapat didefinisikan sbb:

$$x = q \times y + r \text{ and } 0 \leq r < y$$

- $x$  adalah bilangan integer positif atau negatif.
- $y$  adalah bilangan integer positif.
- $q$  adalah hasil **div**, berupa bilangan integer positif atau negatif.
- $r$  adalah hasil **mod**, berupa bilangan integer positif.

## div & mod

---

- Contoh:
  - $7 \text{ div } 3 = \dots$
  - $(-7) \text{ div } 3 = \dots$
  - $7 \text{ mod } 3 = \dots$
  - $(-7) \text{ mod } 3 = \dots$

## 2. Operator Relasional

- Nilai boolean dapat dihasilkan dari **operator perbandingan/ relasional** berikut:

|   |                        |
|---|------------------------|
| = | equals                 |
| ≠ | not equals             |
| < | less than              |
| > | greater than           |
| ≤ | less than or equals    |
| ≥ | greater than or equals |

- Contoh:
  - $2 < 7$
  - $3.14 = 3.142$
  - $\text{True} = \text{False}$
  - $\text{True} = (5 < 11)$



## Equality on Real Number

- Bilangan real memiliki batasan pada memori komputer. Batasan ini bervariasi tergantung pada implementasi bahasa pemrograman.
- Hati-hati saat menyamakan 2 bilangan real.
- Contoh:

$3.142857 = 3.1428571$  akan menghasilkan nilai False.

Alternatif  $|3.142857 - 3.1428571| \leq 0.00001$

### 3. Operator boolean

- Nilai boolean dapat dikombinasikan dengan operator berikut:

| Simbol     | Keterangan | Contoh                      |
|------------|------------|-----------------------------|
| <u>and</u> | Konjungsi  | True <u>and</u> (3 = 7)     |
| <u>or</u>  | Disjungsi  | (7 > 9) <u>or</u> (0 = 0.1) |
| <u>not</u> | Negasi     | <u>not</u> True             |

| A            | B     | A <u>and</u> B | A <u>or</u> B |
|--------------|-------|----------------|---------------|
| <u>False</u> | False | False          | False         |
| <u>False</u> | True  | False          | True          |
| <u>True</u>  | False | False          | True          |
| <u>True</u>  | True  | True           | True          |

## Ekspresi

---

- Ekspresi digunakan untuk mendeskripsikan sebuah nilai (value).
- Sebuah ekspresi dapat tersusun dari nama, simbol, operator, fungsi, () yang dapat menghasilkan suatu nilai berkat evaluasi dari ekspresi tersebut.
- Ekspresi dalam paradigma fungsional terdiri atas 3 jenis, yaitu:
  - **ekspresi dasar** (menerapkan operator dasar)
  - ekspresi kondisional
  - ekspresi rekursif

# Penulisan Ekspresi

| Jenis   | Ekspresi Aritmatika | Ekspresi Boolean |
|---------|---------------------|------------------|
| Infix   | $(3 + 6) * 2$       | $6 > 9$          |
| Prefix  | $(* (+ 3 6) 2)$     | $> 6 9$          |
| Postfix | $(3 6 +) 2 *$       | $6 9 >$          |

- Ekspresi dalam notasi fungsional maupun bahasa Python dituliskan dalam bentuk **infix**.
- Evaluasi ekspresi bergantung pada presedensi.

## Konstruksi Program Fungsional

| Tahapan     | Deskripsi  |
|-------------|--|
| Definisi    | <ul style="list-style-type: none"><li>• Menentukan identitas fungsi, yaitu nama, domain dan range.</li><li>• Contoh: membuat fungsi pangkat tiga dari sebuah bilangan integer.</li><li>• Pangkat3 : integer <math>\rightarrow</math> integer</li></ul> |
| Spesifikasi | <ul style="list-style-type: none"><li>• Menentukan apa yang dilakukan oleh fungsi.</li><li>• Contoh: Fungsi bernama Pangkat3(x) artinya menghitung pangkat tiga dari x.</li></ul>  |

## Konstruksi Program Fungsional (lanj.)

| Tahapan   | Deskripsi   |
|-----------|---|
| Realisasi | <ul style="list-style-type: none"> <li>Menentukan bagaimana fungsi melakukan komputasi, yaitu mengasosiasikan pada nama fungsi, sebuah ekspresi fungsional dengan parameter formal yang cocok.</li> <li>Contoh: mengasosiasikan pada Pangkat3: <math>x*x*x</math> dengan <math>x</math> adalah nama parameter formal.</li> <li>Parameter formal adalah nama yang dipilih untuk mengasosiasikan domain dan range.</li> </ul> |
| Aplikasi  | <ul style="list-style-type: none"> <li>Fungsi untuk memakainya dalam suatu ekspresi, yaitu dengan menggantikan semua nama parameter normal dengan nilai.</li> </ul>   |

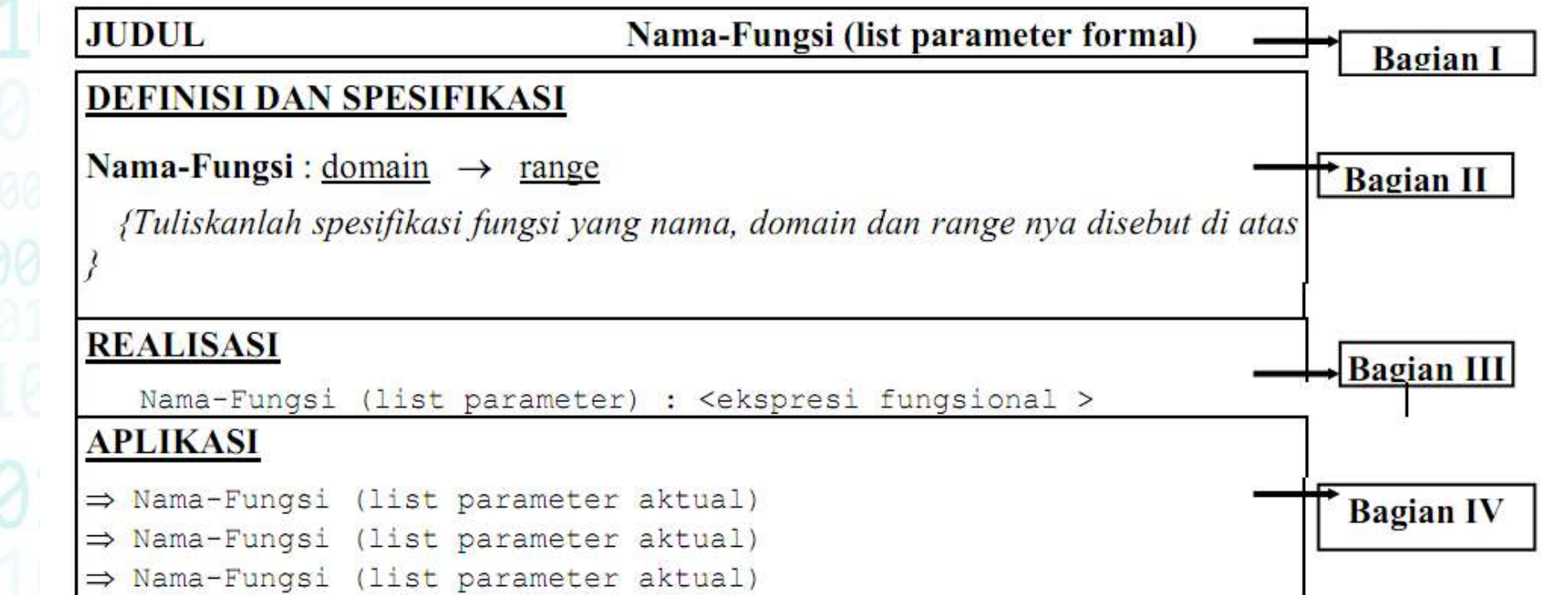
## Notasi Fungsional

---

- Notasi yang dipakai di kuliah ini untuk menuliskan program fungsional.
- Notasi ini tidak mempunyai eksekutor, sehingga harus ditranslasikan ke dalam bahasa pemrograman tertentu untuk mengeksekusinya.
- Translasi ke bahasa program dipelajari pada saat praktikum.
- Bahasa program yang digunakan untuk praktikum adalah bahasa Python.



# Notasi Fungsional



- Kurung kurawal {} digunakan untuk menuliskan komentar.
- Komentar tidak dieksekusi oleh pemroses bahasa, hanya digunakan untuk memberikan penjelasan.



# Notasi Fungsional

| Tahapan                   | Deskripsi  |
|---------------------------|--|
| Header                    | <ul style="list-style-type: none"><li>Berisi judul fungsi, nama dan parameter formalnya.</li></ul>   |
| Definisi dan Speisifikasi | <ul style="list-style-type: none"><li>Berisi identitas fungsi (nama, domain, range) dan deskripsi apa yang dilakukan oleh fungsi.</li></ul>  |
| Realisasi                 | <ul style="list-style-type: none"><li>Berisi realisasi fungsi, yaitu ekspresi fungsional yang ditulis untuk mencapai spesifikasi yang dimaksudkan.</li><li>Sebuah definisi dan spesifikasi yang sama dapat direalisasikan dalam beberapa ekspresi.</li><li>Pada bagian ini nama fungsi dituliskan beserta ekspresinya.</li></ul> |
| Aplikasi                  | <ul style="list-style-type: none"><li>Contoh aplikasi fungsi dapat disertai pula dengan hasil aplikasinya.</li><li>Bagian ini merupakan interaksi langsung dengan pemakai dalam konteks eksekusi.</li></ul>  |

# Notasi Fungsional

## Fungsi Pangkat Dua

**PANGKAT2**

**FX2(x)**

### DEFINISI DAN SPESIFIKASI

**FX2** : integer  $\rightarrow$  integer

*{FX2 (x) menghitung pangkat dua dari x, sebuah bilangan integer }*

### REALISASI

**FX2** (x) :  $x * x$

### APLIKASI

$\Rightarrow$  FX2 (1)

$\Rightarrow$  FX2 (0)

$\Rightarrow$  FX2 (-1)



# Thank you

Selamat Belajar dan Berlatih !!!