



- Dasar Pemrograman – Pertemuan 2

Tim Bahan Ajar Dasar Pemrograman
Teknik Informatika - S1
Fakultas Ilmu Komputer



Review Materi



Materi pertemuan 1

- Sudah mengenal teknologi informasi secara umum?
- Paham pemrograman komputer?
- Mengetahui bahasa pemrograman?
- Mencoba program dengan bahasa python?



Capaian Kuliah Pertemuan 2



Capaian Pembelajaran

- Setelah mengikuti matakuliah ini mahasiswa dapat menjelaskan proses rekayasa perangkat lunak sederhana dengan menerapkan suatu paradigma pemrograman prosedural dengan beberapa hal dasar seperti aksi sekuensial, tipe, variabel, konstanta, nilai dan assignment.



PEMROGRAMAN



Program penyimpanan Komputer

- Urutan dari **instruksi atau perintah tersimpan** di dalam komputer
 - Terdiri atas perintah-perintah dasar yang sudah didefinisikan sebelumnya
 1. Aritmatika dan logika
 2. Tes sederhana
 3. Perpindahan data
- Program spesial yang **mengeksekusi instruksi sesuai urutan** hingga selesai
 - **Intepreter**: menerjemahkan dan mengeksekusi setiap perintah dalam suatu aksi urutan (perintah demi perintah)
 - **Compiler**: menghasilkan kode objek yang kemudian di-link oleh linker menjadi kode yang dapat dieksekusi

Cara kerja komputer



- Manusia **memberikan perintah** pada komputer
- Komputer beberapa kali **menerjemahkan perintah**
- Terdapat beberapa tingkat Bahasa Pemrograman:
 - Bahasa tingkat tinggi / High level language
 - Contoh: Java, Python, PHP
 - Bahasa tingkat menengah / Mid level language
 - Contoh: C
 - Bahasa tingkat rendah / Low level language
 - Contoh: Assembly
 - Bahasa Mesin
 - Contoh: 01010101

Pemrograman

- Tidak sekedar membuat kode-kode perintah
- Belajar pemrograman **BUKAN** belajar bahasa pemrograman
- Pemrograman = Science atau Art?
- Kegiatan memprogram:
 - Utak atik, mencari solusi permasalahan, menganalisis, membuat spesifikasi, menulis kode, mengeksekusi kode, membaca program, menganalisis kompleksitas.
- Skala: rendah (1-2 file), sedang(100-200 file), besar(>200 file)
- Kompleksitas = ke-kompleks-an dari suatu program berkaitan dengan pembuatan kode program yang efektif dan efisien

Apa resep memprogram?

1. Urutan dari **langkah-langkah** sederhana.
2. **Aliran proses kontrol** yang menentukan kapan masing-masing langkah dieksekusi.
3. Tersedia sarana untuk menentukan **kapan harus berhenti**.

$1 + 2 + 3 = \text{Algoritma!}$

Secara formal, ALGORITMA adalah daftar instruksi yang terbatas yang mendeskripsikan suatu KOMPUTASI yang ketika dijalankan pada set input akan di proses melalui set kejadian yang terdefinisi dengan baik dan akhirnya menghasilkan output



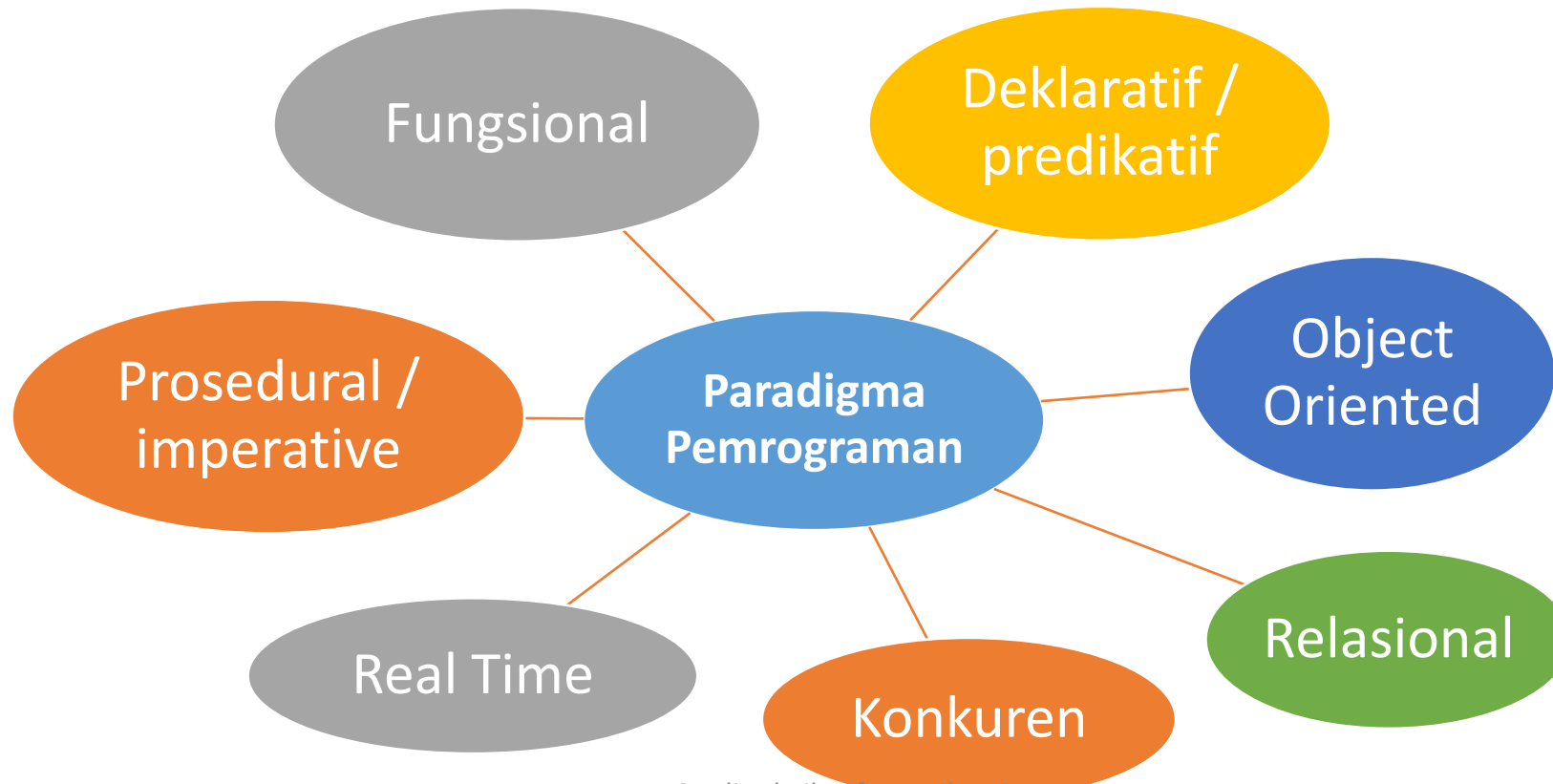
Tipe Pengetahuan



- Pengetahuan **deklaratif** adalah suatu pernyataan fakta
 - Seseorang kenyang setelah makan mie goreng instan.
- Pengetahuan **imperatif/prosedural** adalah resep atau “bagaimana caranya”.
 1. Beli mie di warung
 2. Buka kemasan
 3. Siapkan air di panci sampai mendidih
 4. Masak mie
 5. Siapkan bumbu di piring
 6. Tiriskan mie
 7. Campur bumbu dengan mie
 8. Adi makan mie, sampai akhirnya dia kenyang!

Paradigma Pemrograman

- Paradigma pemrograman adalah sudut pandang penyelesaian persoalan dengan program



Area Pemrograman

- Tekstual vs Visual
- Desktop vs Internet vs Mobile
- Client Server vs N-tier
- Online vs Batch vs Mini-Batch
- Lingkungan pemrograman:
 - Textual Programming
 - Visual Programming
- **Realisasi Program: Source code**

Programmer

- **Bill Gates (Microsoft co-founder)**
- Dennis Ritchie (C Programming language creator)
- Guido van Rossum (Python creator)
- Tim Beners-Lee (HTML and WWW inventor)
- **Steve Wozniak (Apple co-Founder)**
- Linus Torvalds (Linux Kernel Creator)
- Ruchi Sanghvi (FB's first female engineer)
- **Mark Zuckerberg (FB Creator)**



Bahasa Pemrograman

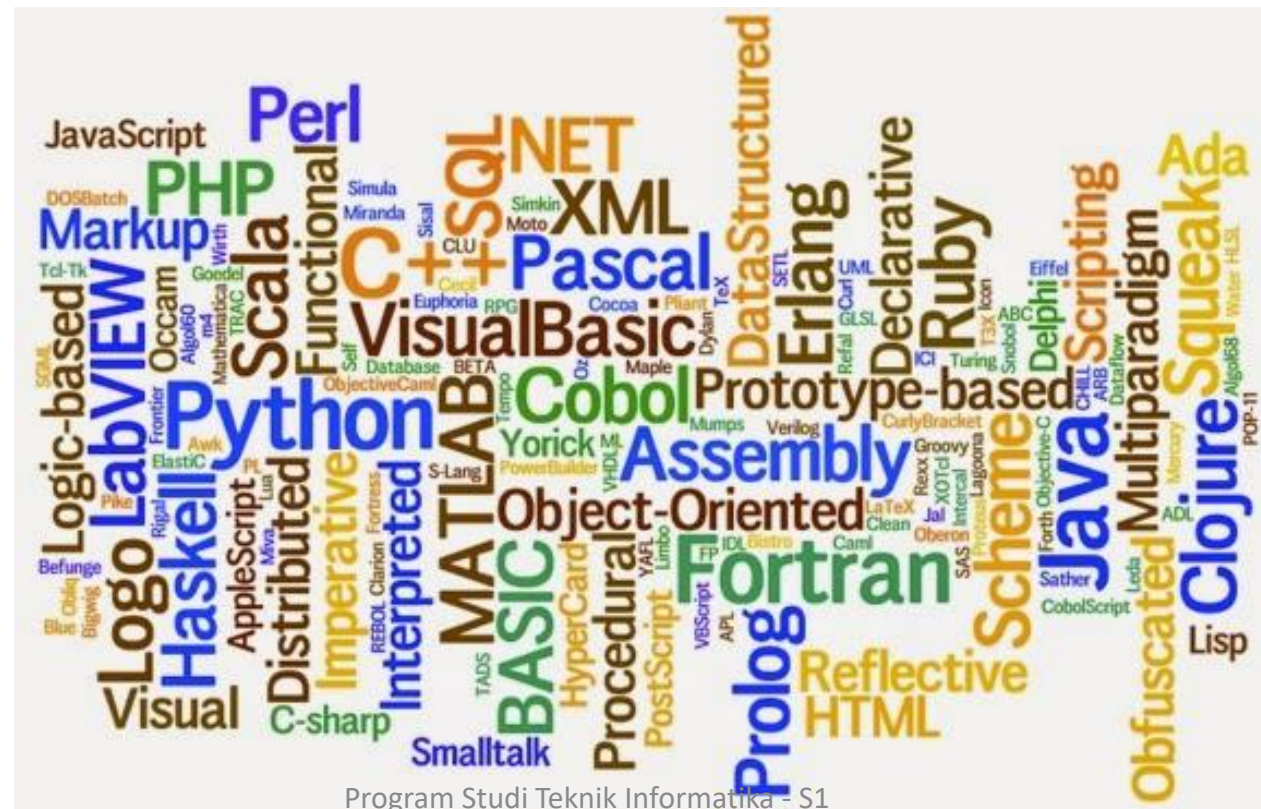


Bagaimana agar mesin paham dengan perintah kita?

- Menggunakan **bahasa mesin**,
 - Contoh: 10101011101010101110101
- Menggunakan **bahasa pemrograman!**
 - Contoh: `print("Halo")`

Macam-macam Bahasa Pemrograman

- Ribuan bahasa pemrograman di dunia...
 - Fortran, Cobol, LISP, Prolog, Pascal, Basic, C, C++, C#, Java, Javascript, Perl, PHP, Visual C++, **Python**, R, B, dll



-

float **
* < > bool
string >= !=
int /
NoneType —
= == +

18

Aspek Bahasa (Sintaks)

- Bahasa Indonesia:
 - “nasi ibu makan” → secara sintaksis tidak valid
 - “ibu makan nasi” → secara sintaksis valid
- Bahasa pemrograman:
 - “hi”5 → secara sintaksis tidak valid
 - $2+5*100$ → secara sintaksis valid

Aspek Bahasa (Semantik Statis)

- Suatu kalimat atau string yang valid secara sintaks dan mempunyai arti
- Bahasa Indonesia:
 - Ibu dimakan nasi → secara sintaksis valid, secara semantik statis tidak valid
- Bahasa Pemrograman:
 - $3+5$ → semantik statis valid
 - $3+"hi"$ → semantik statis tidak valid
- **Semantik** valid artinya makna yang terkait dengan rangkaian simbol/kata benar secara sintaksis tanpa kesalahan semantik statis

Kesalahan memprogram

- Error Sintaksis
 - Hal biasa bagi pemula, mudah dicari dan diperbaiki
- Error Semantik Statis
 - Bahasa pemrograman secara otomatis akan memeriksa ini
 - Dapat menimbulkan perilaku tidak terduga (hasil yang sulit diartikan atau tidak pasti)
- Tidak ada kesalahan sintaksis maupun Semantik statis tetapi **memiliki arti yang berbeda dengan yang diharapkan oleh programmer**
 - Crash/lag/infinite loop/memory leak
 - Output tidak seperti yang diharapkan
 - Istilah keren “**Bug**” → **Solusi Debugging/Metani** (proses untuk mencari bug/kutu)



Pemrograman Prosedural



Pemrograman Prosedural

- Istilah lain adalah pemrograman imperatif
- Dihasilkan berdasarkan suatu dekomposisi (pemecahan masalah) aksional, menjadi aksi-aksi yang dijalankan secara berurutan.
- Aksi-aksi:
 - Jelas ada suatu keadaan awal (*initial state*) dan keadaan akhir (*final state*) dengan waktu yang terbatas
 - Dapat dipecah dalam sub aksi
- Aksi diterjemahkan sebagai instruksi-instruksi yang dapat dijalankan oleh mesin.

Aksi Sekuensial

- Merupakan struktur control algoritma yang paling sederhana.
- Terdiri dari deretan instruksi/aksi/perintah dari aksi ke 1,2,3,..., sampai n.
- Aksi tersebut diawali dengan keadaan awal/*Initial State* yang harus dipenuhi dan keadaan akhir/*final state*.
- *Initial State* disingkat I.S
- *Final State* disingkat F.S

Aksi masak mie instan (versi singkat)

- *Initial State* (I.S): Niat
- *Final State* (I.F): Mie disajikan di piring
- Sub-aksi:
 1. Ambil uang untuk beli mie
 2. Beli Mie instan di warung
 3. Buka bungkus mie instan
 4. Masak mie instan
 5. Siapkan Bumbu
 6. Tiriskan Mie

Notasi Algoritmik

- Aksi sekuensial bisa dibuat dengan notasi algoritma
- Bentuk penulisan algoritma / teks algoritma
- Biasanya terdiri dari 3 bagian utama:
 1. Judul : Spesifikasi deskripsi algoritma secara umum
 2. Kamus : definisi konstanta, tipe data, variabel, spesifikasi prosedur / fungsi
 3. Algoritma : isi penyelesaian masalah dengan menggunakan elemen – elemen pada kamus, intinya program yang berisi instruksi-instruksi atau pemanggilan aksi

Template Notasi Algoritmik

JUDUL

{ Judul sebagai spesifikasi teks algoritmik secara umum }

Algoritma memasak air

KAMUS

{ Pada bagian ini, dilakukan pendefinisian nama konstanta, nama variabel, spesifikasi prosedur, spesifikasi fungsi yang akan digunakan dalam algoritma }

1 panci

1 liter air

DESKRIPSI ALGORITMA

{ Spesifikasi teks algoritmik secara detail }

Masukkan 1 liter air dalam sebuah panci

Panaskan air di dalam panci hingga mendidih

Keterangan

- Judul di isi dengan judul program untuk aksi sekuensial yang didefinisikan.
- Kamus digunakan untuk mendeklarasikan suatu variabel atau konstanta dengan suatu “tipe”
 - Contoh deklarasi: `a : integer`
- Algoritma merupakan bagian utama program dalam bentuk aksi-aksi sekuensial.
 - Bisa berisi bermacam-macam: instruksi perintah input/output, assignment, statemen sekuensial, analisis kasus, dan pengulangan ***kelak akan dipelajari**

Komentar

- Komentar dapat memudahkan dalam membaca alur sebuah program.
- Fungsi komentar:
 - Menerangkan apa yang ada dalam sejumlah baris kode di bawahnya
 - Mendokumentasikan siapa yang menulis kode tersebut atau informasi tambahan lainnya
 - Menonaktifkan sejumlah baris kode - mungkin hanya temporer
- Contoh komentar

Notasi Algoritmik	C++
{Ini komentar}	//ini komentar /*ini komentar */



Tipe



Tipe



- Tipe merupakan pola struktur informasi:
 - Memiliki nama tipe
 - Domain nilai atau batasan nilai yang dapat disimpan pada tipe tersebut
 - Terdapat cara untuk menuliskan tipe atau konstanta(literal)
 - Memiliki operasi
- Tipe ada dua jenis:
 - Primitif: tipe dasar yang disediakan oleh Bahasa pemrograman
 - Non-primitif: tipe bentukan yang dibuat oleh programmer

Tipe primitif

- Boolean
- Integer
- Real
- Character
- String
- Misalnya kita menggunakan Bahasa pemrograman C++:
 - Integer
 - Float
 - Char

Tipe Enumerasi dan Tipe Bentukan

- Tipe Enumerasi adalah tipe yang domainnya tidak dilakukan menurut suatu aturan, melainkan dengan menyebutkan satu persatu anggotanya.
- Tipe Bentukan adalah tipe yang dibentuk dari beberapa komponen tipe primitif

Apa kegunaan dari tipe?

- Untuk mempermudah proses berfikir dalam tingkatan abstraksi yang lebih tinggi.
- Abstraksi merupakan cara untuk melihat sesuatu dengan lebih mudah.
- Misalnya dengan tipe kita bisa mengabstrasikan suatu data mahasiswa yang sebetulnya terdiri dari beberapa data seperti nim, nama, d..ll



Variabel dan Konstanta



Variabel

- Nama yang merepresentasikan sebuah tempat di memori komputer untuk menyimpan data berupa nilai ber-tipe yang sifatnya dinamis (dapat berubah-ubah)
- Contoh :

$x \leftarrow 122$

$y \leftarrow 14$

$x \leftarrow 100$

Variabel

Nilai

x

~~122~~ 100

y

14

Penggunaan Variabel

- **Dideklarasikan** supaya nama dikenali mesin
- **Dialokasikan** pada memori komputer
- **Diinisialisasi** dengan memberi nilai awal agar siap dimanipulasi
- Contoh:
 - Deklarasi sekaligus alokasi: `i : integer`
 - Inisialisasi: `i ← 0`
 - Simbol `←` adalah **assignment** (nanti akan dijelaskan)

Tidak boleh menggunakan **nama variabel** yang digunakan oleh internal program (**reserved word**)

Aturan penulisan variabel [1]

- **Case sensitive**, penulisan dengan menggunakan **huruf kecil** dan **huruf kapital** memiliki arti yang **berbeda** (merupakan **variabel** yang **berbeda**).
- **Contoh :**
 - angka
 - Angka

Variabel 'angka' dan 'Angka' merupakan dua variabel yang berbeda.

Aturan penulisan variabel [2]

- Nama variabel **tidak** boleh menggunakan **spasi**.
- **Contoh :**
 - luas persegi** /*Salah karena menggunakan spasi*/
 - luas_persegi** /*Benar, karena tidak menggunakan spasi*/
 - luasPersegi** /*Benar, karena tidak menggunakan spasi*/

Aturan penulisan variabel [3]

- Nama variabel hanya boleh diawali dengan huruf atau garis bawah (**tidak boleh diawali dengan angka dan simbol lain**)
- Contoh :
 - `@angka` /*Salah, karena menggunakan simbol*/
 - `angk@` /*Salah, karena menggunakan simbol, meskipun berada di belakang*/
 - `a5gk4` /*Benar, karena diawali dengan huruf dan diikuti angka*/
 - `_angka` /* Benar, karena diawali dengan garis bawah*/
 - `4angka` /* Salah, karena diawali dengan angka*/

Konstanta

- Nilai tetap seperti angka, huruf, dan teks dinamakan dengan “konstanta” karena nilainya tidak berubah
- Konstanta numerik/angka seperti yang anda lihat apa adanya
- Contoh:
 - `constant pi : real = 3.14`

Nilai

- Nilai bisa disebut juga harga merupakan besaran suatu tipe
- Cara untuk mengisi nilai dengan mekanisme assignment.
- Contoh variabel a bertipe integer yang memiliki nilai 10:
 - `a : integer ← 10`



Assignment



Assignment

- **Pemberian nilai** terhadap suatu variabel
- **<ruas kiri> \leftarrow <ruas kanan>**
- **Ruas kiri** adalah **variabel**
- **Ruas kanan** dapat berupa **nilai, konstanta, ekspresi**
- **Contoh:**
 - $a \leftarrow 1$ {nilai diisi dengan konstanta}
 - $a \leftarrow 1+2$ {nilai diisi dengan ekspresi aritmatika}
 - $a \leftarrow a+b$

Perintah assignment pada program

- Realisasi dengan bahasa pemrograman, assignment memiliki kode simbol = **jangan dimaknai sebagai “sama dengan”** karena “sama dengan” yang digunakan untuk operator perbandingan yaitu dengan simbol ==
- Contoh:
 - a:integer {deklarasi a}
 - b:integer {deklarasi b}
 - a = 1
 - a = 1+2
 - a = a+b

Referensi



Utama :

1. Liem, Inggriani. Diktat Pemrograman Prosedural Informatika ITB. IF-ITB. 2007
2. Bjarne Stroustrup, 2014, Programming: Principles and Practice Using C++ (Second Edition), Addison-Wesley Professional

Pendukung :

1. Introduction to Computer Science and Programming in Python, MIT
<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016>
2. Introduction to Computer Science and Programming, MIT <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-00sc-introduction-to-computer-science-and-programming-spring-2011/index.htm>