

# **LECTURE NOTES**

**COMP6124**

**Program Design Methods**

**Minggu 1**

**Program Design and Pseudocode, Developing  
an Algorithm**

## LEARNING OUTCOMES

Setelah menyelesaikan pembelajaran ini, mahasiswa akan mampu:

- ☑ LO1 – Menjelaskan metode-metode perancangan program

Outline Materi (Sub-Topic) :

1. Steps in Program Development
2. Program Design Methodology
3. Procedural versus Object – Oriented Programming
4. An Introduction to Algorithms and Pseudocode
5. Program Data
6. How to Write Pseudocode
7. The Structure Theorem
8. Defining the Problem
9. Designing a Solution Algorithm
10. Checking the Solution Algorithm

## ISI MATERI

### 1. Steps in Program Development

Berdasarkan teori dari Anne Robertson, tahapan pengembangan software dibagi menjadi 7 langkah dasar:

#### 1. Mendefinisikan masalah.

Tahapan ini merupakan tahapan untuk memahami permasalahan yang ada dalam mengembangkan sebuah program, sehingga dihasilkan kebutuhan dari sistem yang dibagi menjadi tiga komponen utama; input, proses dan output. Input merupakan data-data yang dibutuhkan oleh program, proses menjelaskan langkah-langkah yang dibutuhkan oleh program untuk mengolah input dan menyelesaikan masalah yang ada, sehingga dihasilkan sebuah output. Hasil dari tahapan ini dapat berupa tabel definisi seperti contoh pada sub-topik kedelapan dari materi ini.

#### 2. Outline the solution

Setelah masalah didefinisikan, langkah selanjutnya adalah membagi masalah menjadi beberapa tugas yang lebih kecil, seperti:

- Langkah-langkah proses utama
- User interface
- Struktur control utama
- Logika utama, dll

#### 3. Mengembangkan outline menjadi algoritma

Solusi yang telah dibuat menjadi outline pada tahap kedua kemudian dikembangkan menjadi sebuah algoritma. Algoritma adalah urutan tahap yang secara tepat menunjukkan tugas-tugas yang harus dilakukan untuk dapat mencapai suatu tujuan tertentu. Salah satu cara untuk

merepresentasikan algoritma adalah dengan menggunakan pseudocode (bentuk dari bahasa Inggris yang terstruktur, akan dijelaskan lebih lanjut pada sub topic berikutnya) atau menggunakan flowchart.

#### 4. Menguji kebenaran dari algoritma

Sebelum algoritma diimplementasikan ke dalam bahasa pemrograman, kebenaran dari algoritma yang dibuat perlu diuji terlebih dahulu. Salah satu pengujian yang dapat dilakukan adalah dengan menggunakan desk checking.

#### 5. Menulis algoritma ke dalam bahasa pemrograman

Setelah keempat tahap awal dijalankan dan telah menunjukkan hasil yang diinginkan, langkah selanjutnya adalah melakukan pemrograman dengan mengkonversi pseudocode/flowchart yang telah dibuat sebelumnya menjadi bahasa pemrograman.

#### 6. Mengeksekusi program

Apabila program telah selesai dibuat, maka program akan dieksekusi untuk dalam mendemonstrasikan algoritma yang dibuat sebelumnya sehingga masalah yang dihadapi mendapatkan solusi tepat.

#### 7. Mendokumentasikan dan memelihara program

Dokumentasi bukan merupakan langkah terakhir dari proses pengembangan program, namun merupakan proses yang dilakukan secara parallel dari awal hingga program dapat dijalankan. Dokumentasi program dapat berupa eksternal (hierarchy chart, hasil uji data, dst) atau internal dokumentasi (komentar pada kode bahasa pemrograman).

## 2. Metodologi perancangan program

Terdapat tiga prinsip fundamental dari perancangan sebuah program, dimana program menerima input berupa data, melakukan proses terhadap data tersebut dan kemudian menampilkan hasil pemrosesan data kepada user. Berdasarkan prinsip tersebut, terdapat tiga pendekatan dalam merancang sebuah program:

### 1. Procedure-driven

Procedure driven merupakan pendekatan perancangan program yang berdasarkan ide bahwa hal yang paling penting dari sebuah program adalah **apa yang dilakukan oleh program**, baik proses maupun fungsinya. Sehingga dalam melakukan perancangan, difokuskan pada proses-proses apa saja yang harus ada dalam program. Setelah itu akan dianalisis data yang dibutuhkan dari setiap proses dan output yang dihasilkan.

### 2. Data-driven

Pada pendekatan data-driven, dasar utamanya adalah anggapan bahwa hal yang paling stabil dari sebuah program adalah **data** dibandingkan proses yang terjadi didalamnya. Sehingga perancangan program dimulai dengan analisis data apa saja yang dibutuhkan oleh program, dan hubungan-hubungan antar data tersebut. Setelah struktur data terbentuk, output dari program diuji dan terakhir proses apa saja yang dibutuhkan untuk mengkonversi input data menjadi output diidentifikasi.

### 3. Event-driven

Pada perancangan program dengan menggunakan pendekatan event-driven, hal utama yang diidentifikasi adalah **kegiatan-kegiatan** atau **interaksi** dengan dunia luar yang menyebabkan program berubah dari satu state ke state lainnya.

### 3. Pemrograman prosedural vs Object Oriented

Pemrograman procedural merupakan pemrograman terstruktur dari atas ke bawah yang berfokus pada proses apa saja yang dilakukan oleh program. Sedangkan focus dari pemrograman berorientasi objek adalah pada objek-objek yang ada dalam sistem yang membangun program secara keseluruhan.

### 4. Pengenalan Algoritma dan Pseudocode

Algoritma merupakan urutan dari langkah-langkah yang terurut secara tepat untuk menyelesaikan suatu tugas khusus, biasanya algoritma ditulis dalam bahasa Inggris. Berikut ini adalah beberapa syarat sebuah algoritma:

- Harus jelas, tepat dan tidak ambigu
- Memberikan solusi yang benar pada semua kasus yang mungkin terjadi
- Memiliki akhir

Berikut ini adalah contoh algoritma sederhana:

```
turn on calculator
clear calculator
repeat the following instructions
    key in dollar amount
    key in decimal point (.)
    key in cents amount
    press addition (+) key
until all prices have been entered
write down total price
turn off calculator
```

Pada contoh di atas, dijabarkan secara jelas tahap-tahap yang harus diikuti untuk menghitung total harga dari sekumpulan barang, dimana penginputan harga barang akan diulang secara terus-menerus hingga

barang habis, setelah itu akan dilakukan proses penjumlahan dan ditampilkan totalnya pada layar.

Sedangkan yang dimaksud dengan pseudocode adalah salah satu bentuk representasi dari algoritma berupa bahasa Inggris yang terstruktur. Karakteristik utama dari sebuah pseudocode adalah sebagai berikut:

- Semua statement ditulis dalam bahasa Inggris
- Setiap perintah ditulis dalam baris terpisah
- Kata kunci (keyword) dan lekuk tab ke dalam digunakan untuk menunjukkan struktur khusus dari suatu control
- Setiap perintah ditulis dari atas ke bawah, dengan satu masukan dan satu keluaran.
- Sekumpulan statement dalam dikelompokkan dalam satu modul dan diberi nama khusus.

## 5. Program Data

Pada program data akan dikenal beberapa istilah berikut:

- **Variable:** merupakan sekumpulan dari memory yang ditugaskan untuk menyimpan item data
- **Constant:** item data dengan nama dan nilai yang mutlak selama program dijalankan.
- **Literal:** merupakan constant yang namanya merepresentasikan nilai yang disimpan

Tipe data pada bahasa pemrograman pada umumnya dibagi menjadi dua jenis:

- **Tipe data dasar:** Integer, Real, Character, dan Boolean.
- **Tipe data terstruktur:** Record, File, Array, dan String.

## 6. Cara menulis Pseudocode

Ketika merancang sebuah solusi algoritma, computer yang akan menjalankan perintah yang diberikan, sehingga penggunaan kata-kata pada pseudocode yang berkorelasi dengan perintah pada bahasa pemrograman akan mempermudah proses konversi bahasa pemrograman dari pseudocode. Pada subtopic ini, akan dijelaskan enam perintah dasar computer dan kata kunci yang dapat digunakan dalam pseudocode untuk menunjukkan perintah tersebut.

### 1. Komputer dapat menerima informasi

Ketika computer membutuhkan informasi atau data dari user, computer akan meminta input data. Hal ini dapat diwakilkan oleh kata kunci:

**Read** atau **Get**. Read digunakan untuk meminta inputan data dari sebuah file, sedangkan Get digunakan untuk menerima input data dari user melalui keyboard.

### 2. Komputer dapat menampilkan informasi

Dalam menampilkan informasi, pada pseudocode dapat digunakan kata kunci sebagai berikut:

- **Print**, digunakan ketika output akan dikirim pada printer
- **Write**, digunakan ketika output akan ditulis ke dalam file
- **Put out**, **Output** dan **Display**, digunakan untuk output yang akan ditampilkan di layar.
- **Prompt**, digunakan untuk menampilkan informasi kepada user untuk meminta respond input dari user, dengan menggunakan kata kunci Get. Berikut adalah contoh penggunaan Prompt:

```
Prompt student_mark
```

```
Get student_mark
```

### 3. Komputer dapat menjalankan perhitungan aritmatika

Apabila program yang dibuat membutuhkan perhitungan matematika,



maka kata kunci berikut dapat digunakan: Compute dan Calculate. Sedangkan symbol matematika dapat digunakan symbol ataupun kata kunci berikut: +, -, \*, /, () atau **Add, Subtract, Multiply, and Divide**.

4. Komputer dapat menyimpan suatu nilai pada variable atau lokasi memori tertentu:
  - **Initialize** dan **Set** digunakan untuk memberikan nilai awal
  - **=** atau **←**, digunakan untuk menyimpan nilai dari suatu proses/perhitungan
  - **Save** dan **Store**, digunakan untuk menyimpan nilai pada suatu variable.
5. Komputer dapat membandingkan dua variable dan memilih satu dari beberapa alternative aksi yang ada.

Untuk menjalankan perintah perbandingan, dibutuhkan struktur seleksi. Berikut ini adalah salah satu contoh struktur seleksi IF.

```
IF score > 49 THEN
    Display "PASS"
ELSE
    Display "FAIL"
```

6. Komputer dapat melakukan perulangan terhadap sekumpulan aksi

Berikut ini adalah salah satu contoh perintah perulangan pada struktur DOWHILE:

```
DOWHILE student_total != 30
    Read student_record
    Print student_id, student_name, GPA to report
    student_total = student_total + 1
ENDDO
```

## 7. Teorema Struktur

Terdapat tiga struktur teorema yang digunakan dalam membuat sebuah program:

- **Sequence**, merupakan eksekusi program dari satu statement ke statement selanjutnya secara berurut.
- **Seleksi**, struktur yang memiliki pengecekan suatu kondisi, dimana kondisi tersebut akan mempengaruhi statement yang akan dijalankan dari beberapa pilihan statement yang ada.
- **Perulangan**, merupakan struktur yang memiliki statement yang akan dieksekusi secara berulang apabila kondisi tertentu terpenuhi.

## 8. Mendefinisikan masalah

Cara utama mendefinisikan suatu masalah adalah dengan membaca dan menganalisis secara teliti masalah tersebut sehingga didapatkan kebutuhan akan input, proses dan output dari sebuah program yang akan dikembangkan.

- **Input**: sekumpulan sumber data yang diberikan oleh sumber permasalahan.
- **Proses**: sekumpulan aksi yang dibutuhkan untuk menghasilkan output yang dibutuhkan.
- **Output**: sekumpulan output yang dibutuhkan dari program sebagai bentuk solusi yang dirancang.

## 9. Merancang algoritma solusi

Setelah masalah didefinisikan secara jelas, langkah selanjutnya adalah merancang sebuah algoritma solusi dengan menggunakan pseudocode yang telah dibahas sebelumnya.

## 10. Menguji algoritma solusi

**Desk checking** merupakan salah satu cara yang dapat digunakan untuk menguji ketepatan sebuah algoritma. Berikut ini adalah enam langkah dalam melakukan desk checking:

1. Memilih dua hingga tiga contoh input sederhana yang valid
2. Menentukan apakah output yang seharusnya dihasilkan dari setiap input yang telah dipilih.
3. Membuat tabel dengan memasukkan semua variable yang digunakan dalam algoritma pada sisi kolom
4. Mensimulasikan baris per baris dari setiap statement yang ada pada pseudocode dengan perubahan nilai setiap variable pada pada tabel.
5. Lakukan perulangan simulasi dengan menggunakan contoh input kedua dan seterusnya
6. Bandingkan output seharusnya yang telah ditentukan sebelumnya dengan hasil output pada table desk checking.

## SIMPULAN

1. Terdapat tiga metodologi dalam merancang sebuah program; procedure-driven, data-drive, dan event-drive. Sedangkan metodologi pemrogramannya terbagi menjadi dua, terstruktur dan berorientasi objek.
2. Algoritma merupakan langkah-langkah secara tepat untuk menyelesaikan suatu tugas, dimana algoritma dapat direpresentasikan menggunakan pseudocode ataupun flowchart.
3. Struktur teorema dibagi menjadi tiga; sekuensial, seleksi dan perulangan. Dengan menggunakan tiga dasar teorema tersebut, program apapun dapat dikembangkan.

## DAFTAR PUSTAKA

1. Simple Program Design : A Step by Step Approach., Lesley Anne Robertson, Fifth Edition, Course Technology. ISBN: 978-1-4239-0132-7.
2. The Structured Theorem, <http://www.docstoc.com/docs/4530185/Developing-an-Algorithm-The-Structure-Theorem-The-Structured-Theorem>
3. Vague Pseudocode Examples,  
[http://users.csc.calpoly.edu/~jdalbey/SWE/pdl\\_vague.html](http://users.csc.calpoly.edu/~jdalbey/SWE/pdl_vague.html)
4. Object Oriented Overview,  
[http://www.tutorialspoint.com/uml/uml\\_quick\\_guide.htm/SWE/pdl\\_vague.htm](http://www.tutorialspoint.com/uml/uml_quick_guide.htm/SWE/pdl_vague.htm)  
|