

Modul 2

Representasi Algoritma

Adam Bachtiar, S.Kom., M.MT

Tujuan Pembelajaran

Secara khusus setelah mempelajari modul ini, mahasiswa diharapkan mampu :

1. Merepresentasikan algoritma dalam bentuk *flowchart* dan *pseudocode*
2. Membuat flowchart dan pseudocode dengan menggunakan bahasa sehari-hari dengan sederhana dan dapat dimengerti oleh orang awam
3. Mengenali simbol-simbol flowchart yang umum digunakan
4. Menyusun struktur pseudocode yang tepat dan memberikan indentasi jika terdapat sub-proses
5. Menyusun pseudocode dengan logis, representatif dan ideal
6. Menyusun flowchart dengan aturan yang ada
7. Menganalisa permasalahan, dan menyusun penyelesaian dengan menggunakan flowchart dan pseudocode pada tingkat ideal

Modul 2

Flowchart (Diagram Alir)

Adam Bachtiar, S.Kom., M.MT

Pengertian Flowchart

Flowchart adalah jenis diagram (grafik atau simbolik) yang mewakili algoritma atau proses-proses tertentu

Keuntungan Menggunakan Flowchart

Komunikasi

Cara paling mudah untuk mengkomunikasikan logika atau alur berfikir sebuah sistem

Analisis Yang Efektif

Dengan bantuan flowchart, masalah dapat dianalisis lebih efektif

Dokumentasi

Berfungsi sebagai dokumentasi cetak biru program baik agar dapat ditelusur dengan mudah

Pengkodean Yang Efisien

Berfungsi cetak biru atau pedoman program baik selama proses analisis dan pengembangan program

Proper Debugging

Membantu mempercepat proses penelusuran kesalahan program dengan baik dan cepat

Pemeliharaan Program Yang Efisien

Membantu programmer dalam memelihara program dengan lebih efisien

Keterbatasan Menggunakan Flowchart

Alur Logika Yang Rumit atau Kompleks

Ketika berhadapan dengan logika yang kompleks akan lebih sulit untuk direpresentasi dengan flowchart

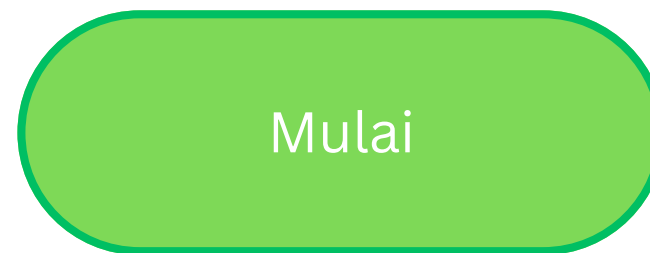
Perubahan / Modifikasi

Ketika terdapat perubahan kebutuhan dari program, akan berimbas pada perombakan flowchart

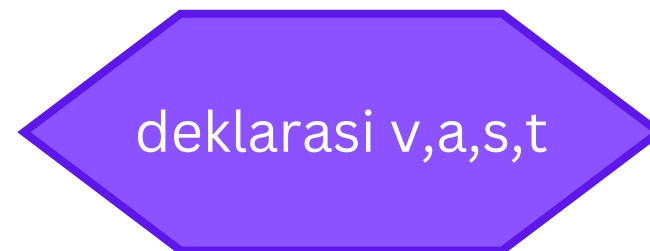
Simbol - Simbol Flowchart

Terdapat 4 unsur utama simbol dari flowchart yaitu (1) Input / masukan; (2) Percabangan; (3) Perulangan; (4) Output

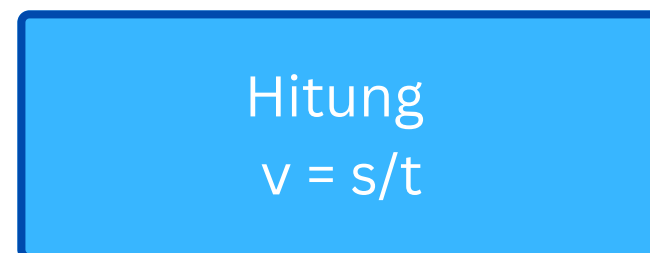
Simbol - Simbol Flowchart



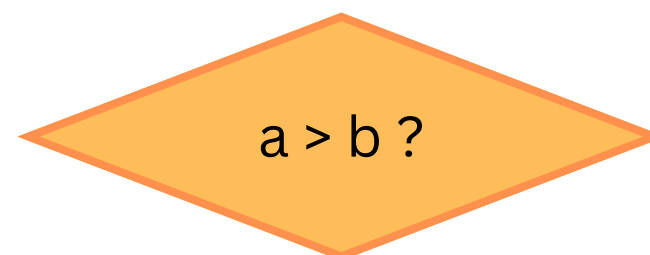
Terminator, berbentuk oval menunjukkan awal atau akhir dari proses dalam flowchart



Preperation (predifine process), digunakan sebagai penyimpanan atau pendeklarasian variabel yang dibutuhkan oleh proses dalam flowchart

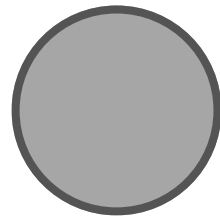


Process (proses), persegi panjang menunjukkan langkah aliran proses normal/generik

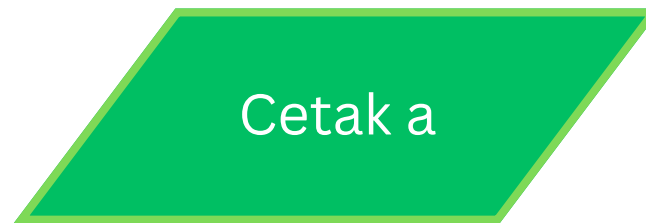


Decision (Keputusan), menunjukkan adanya percabangan dalam aliran proses dalam flowchart. Biasanya dalam bentuk pertanyaan dengan pilihan jawaban ya/tidak

Simbol - Simbol Flowchart



Connector (penghubung), digunakan untuk lompatan dalam aliran proses, biasanya untuk proses yang kompleks



Data (input/output), menunjukkan input ataupun output dari hasil dari suatu proses



Arrow (Tanda Panah), digunakan untuk menunjukkan aliran kontrol dalam suatu proses

Tugas Modul 2A

Buatlah sebuah algoritma dengan menggunakan Flowchart,
untuk menentukan :

1. Apakah sebuah bilangan adalah bilangan genap atau ganjil ?
2. Manakah rute dengan jalur terpendek, jika ada dua rute yang akan dibandingkan ?
3. Mengurutkan 3 bilangan yang diinputkan, mulai dari yang terkecil sampai yang terbesar !

Deadline : 7 Oktober 2024, maks. Pkl. 23.59 Wita
Upload ke GitHub masing-masing

Modul 2

Pseudocode

Adam Bachtiar, S.Kom., M.MT

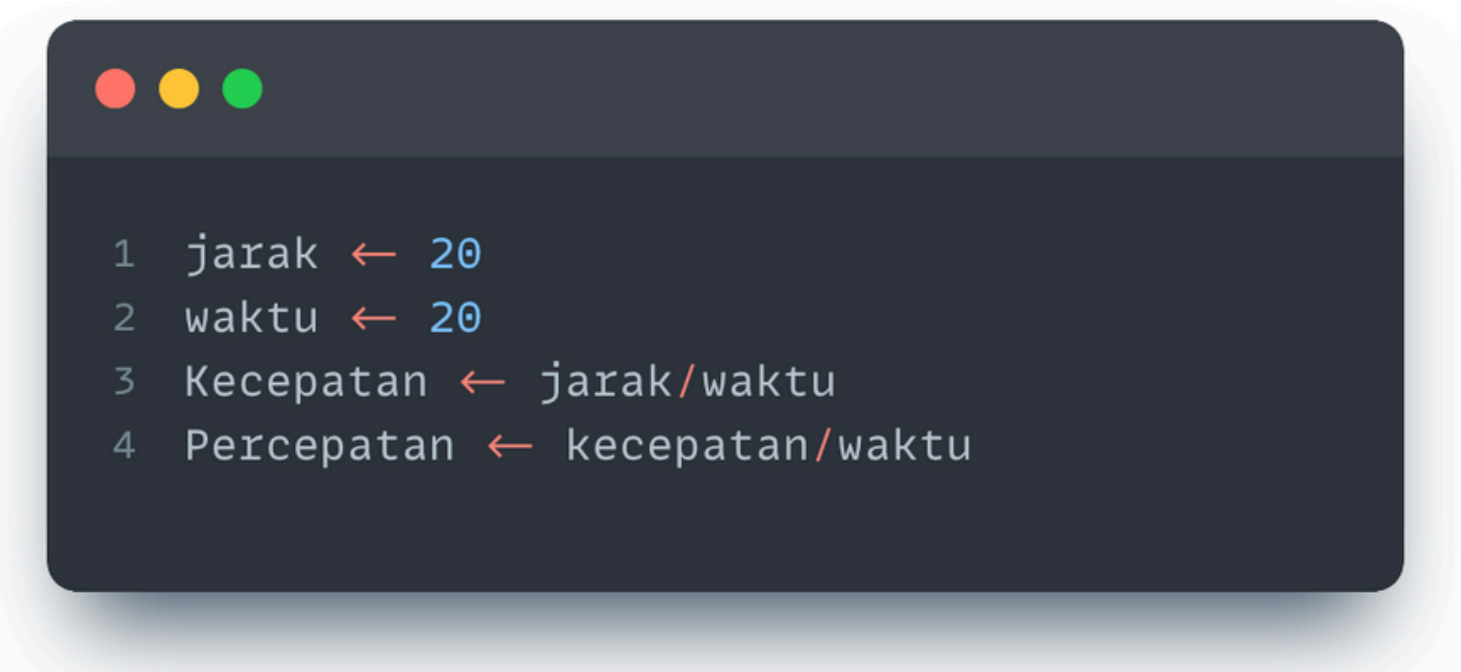
Pengertian Pseudocode

Pseudocode, adalah deskripsi tingkat tinggi dan singkat (ringkas) ditulis untuk dibaca oleh manusia dan bukan merupakan perintah untuk mesin (kompailer)

Cara Menulis Pseudocode

Ditulis dengan bahasa sederhana dan mudah untuk dimengerti, setiap pseudocode boleh dimulai dengan kata-kata “Mulai”, dan diakhiri dengan “Selesai”

Notasi \leftarrow digunakan untuk mengisi nilai pada sebuah variabel, contoh :

A code editor window with a dark background and three colored window control buttons (red, yellow, green) at the top left. It contains four lines of pseudocode, each starting with a line number and followed by an assignment statement using the left-pointing arrow notation.

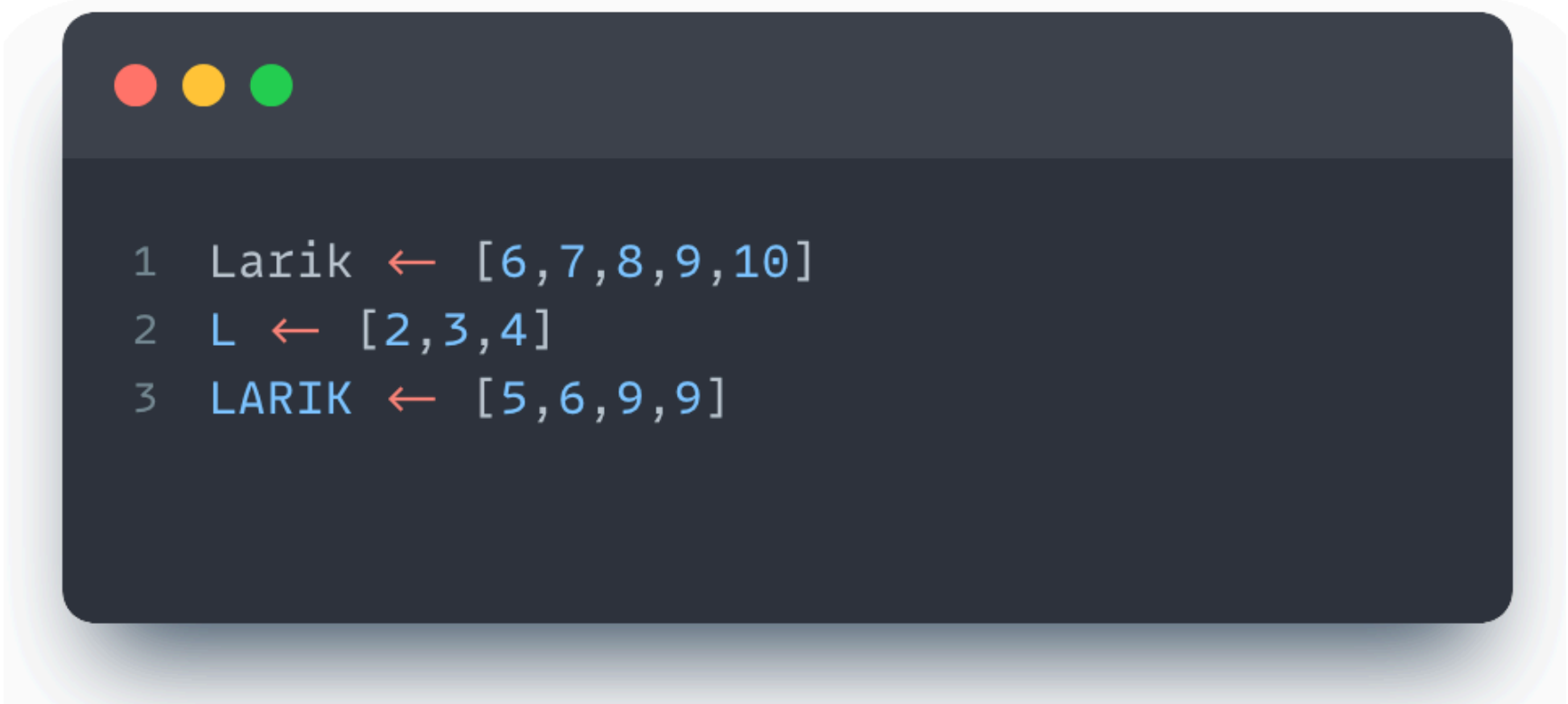
```
1 jarak  $\leftarrow$  20
2 waktu  $\leftarrow$  20
3 Kecepatan  $\leftarrow$  jarak/waktu
4 Percepatan  $\leftarrow$  kecepatan/waktu
```

Setiap instruksi ditulis secara independen dan ditulis dalam baris sendiri

Cara Menulis Psudocode

Disarankan untuk menulis variabel skalar menggunakan huruf kecil, dimana variabel skalar merupan variabel yang dapat menyimpan nilai yang dapat berubah

Untuk variabel yang menyimpan nilai array disarankan untuk menggunakan huruf besar didepannya




```
1 Larik ← [6,7,8,9,10]
2 L ← [2,3,4]
3 LARIK ← [5,6,9,9]
```

Notasi seperti $L[i]$, menunjukkan elemen ke-i, dari variabel larik L. Sedangkan $L[i,j]$, menunjukkan baris ke-i dan kolom ke-j

Cara Menulis Pseudocode

Variabel majemuk, yaitu variabel yang dapat menyimpan beberapa nilai dalam sebuah variabel

dan untuk mengisi data dapat dengan cara :



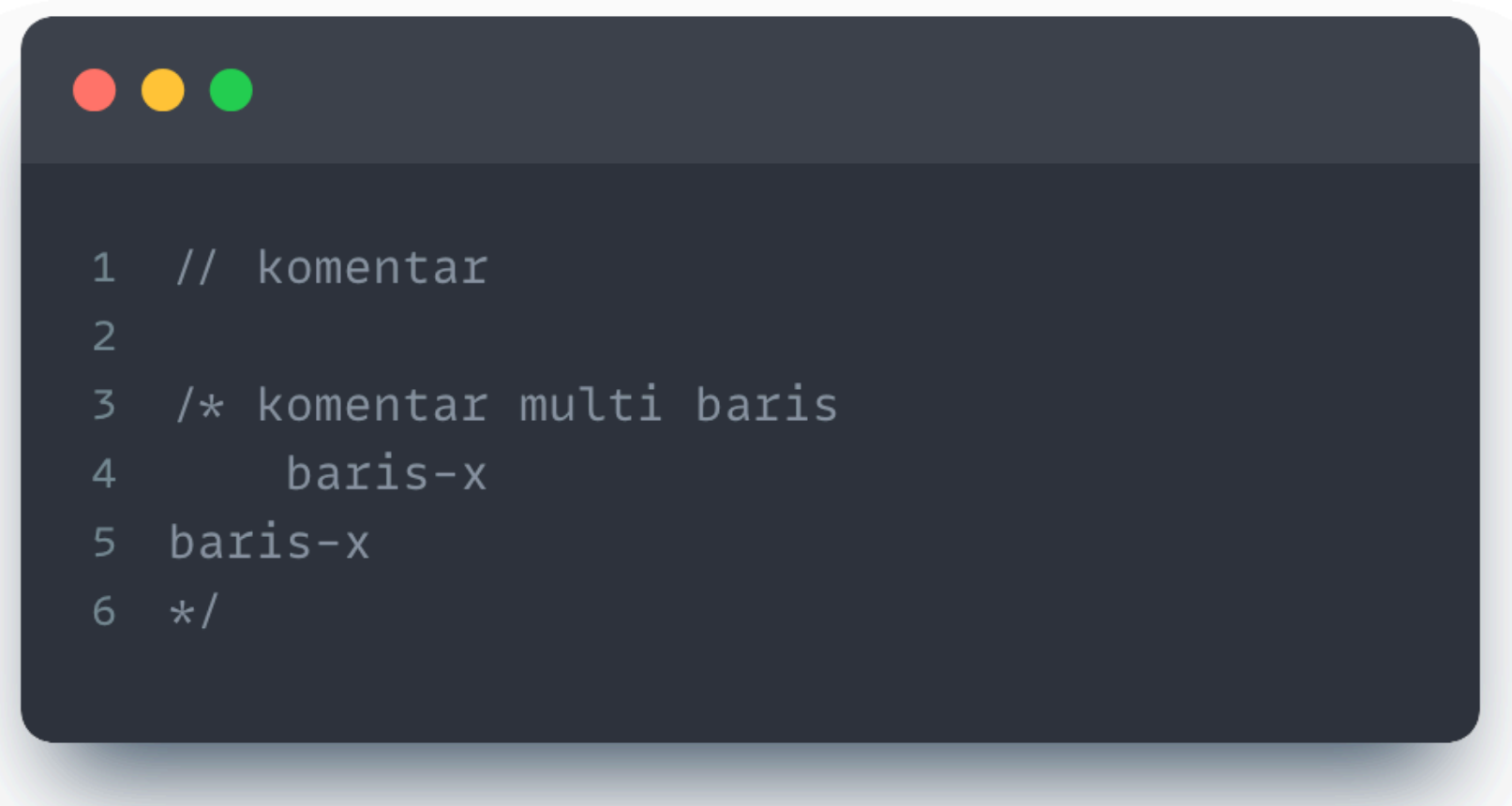
```
1 Kendaraan = GROUP
2   nama
3   jenis
4   penemu
5 AKHIR-GROUP
```

```
1 nama ← "Sepeda"
2 jenis ← "Sepeda Gunung"
3 penemu ← "Baron Karl Van Drais"
```

Dapat menyertakan nomor urut baris, dan jika terdapat sub-proses diwajibkan menggunakan indentasi

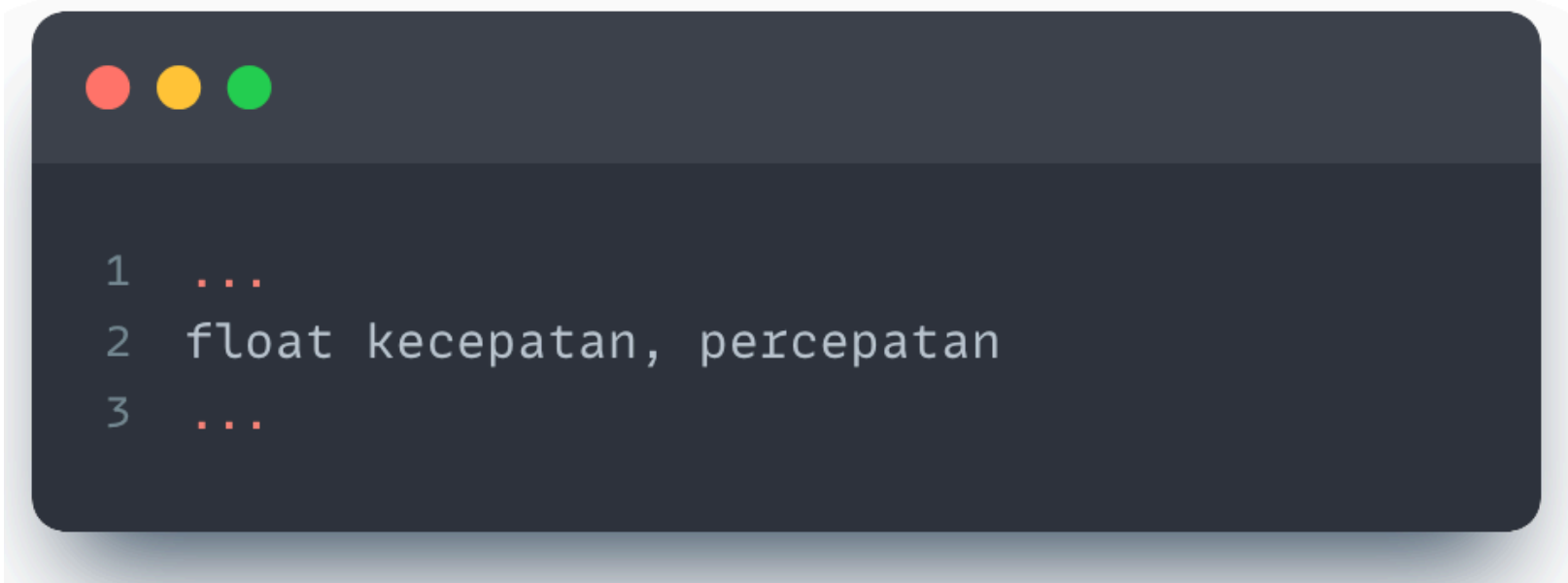
Cara Menulis Pseudocode

Simbol `//` atau `/* ... */` digunakan untuk menulis komentar



```
1 // komentar
2
3 /* komentar multi baris
4     baris-x
5 baris-x
6 */
```

Setiap baris instruksi harus jelas, misalnya menyertakan variabel dan tipe datanya



```
1 ...
2 float kecepatan, percepatan
3 ...
```


Cara Menulis Psudocode

Notasi masukan(), input(), cetak() atau print(), mewakili I/O, contohnya :



```
1 masukan()
2 input()
3 cetak()
4 print()
```

Untuk Operator logika, menggunakan notasi sebagai berikut :

Operator/Notasi	Keterangan
AND	Dan
OR	Atau

Untuk logika menggunakan true/benar dan false/salah, sementara operator pembandingan dapat menggunakan cara penulisan sebagai berikut :

Operator/Notasi	Keterangan
<	Kurang Dari
>	Lebih Dari
<=	Kurang Dari Sama Dengan
>=	Lebih Besar dari Sama Dengan
<>	Sama Dengan
!=	Tidak Sama Dengan

Cara Menulis Pseudocode

Notasi perulangan (while-loop), dimana selama kondisi bernilai benar (true), maka pernyataan akan dijalankan :

```
1 // Perulangan (while-loop)
2 ULANG-SELAMA (kondisi) :
3     pernyataan-1
4     pernyataan-2
5     ...
6 AKHIR-ULANG
```

Notasi perulangan (do-loop) blok perulangan berjalan 1 kali, baru mengecek kondisi jika benar (true), maka pernyataan akan dijalankan

```
1 // Perulangan (do-loop)
2 ULANG :
3     pernyataan-1
4     pernyataan-2
5     ...
6 SELAMA (kondisi)
```

Cara Menulis Pseudocode

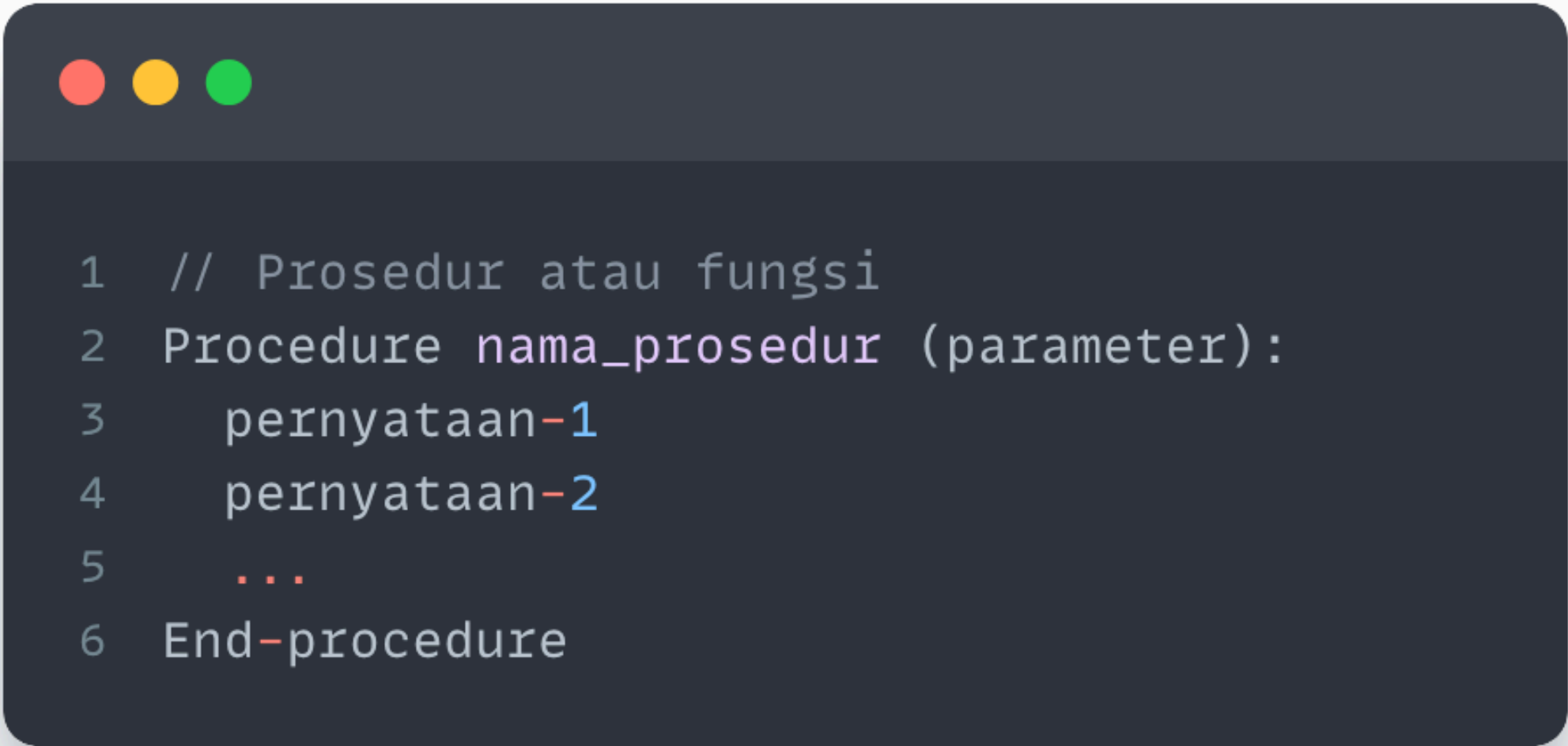
Notasi perulangan (for-loop), yaitu perulangan yang akan berhenti hanya jika batas akhir perulangan terpenuhi, notasinya:

```
1 // Perulangan (for-loop)
2 UNTUK (variabel = awal sampai akhir):
3     pernyataan-1
4     pernyataan-2
5     ...
6 AKHIR-UNTUK
```

```
1 // Perulangan (for-loop)
2 for (variabel = awal sampai akhir):
3     pernyataan-1
4     pernyataan-2
5     ...
6 End-for
```

Cara Menulis Psudocode

Notasi untuk membuat prosedur atau fungsi adalah sebagai berikut :



```
1 // Prosedur atau fungsi
2 Procedure nama_prosedur (parameter):
3     pernyataan-1
4     pernyataan-2
5     ...
6 End-procedure
```

Struktur Pseudocode

Judul, nama program, class

{Penjelasan tentang algoritma, berisi uraian singkat mengenai masalah yang akan diselesaikan}

Deskripsi

{deklarasi, semua nama yang dipakai, meliputi nama tipe, nama konstanta, nama peubah, nama prosedur, dan nama fungsi}

Implementasi

{semua langkah/aksi algoritma dituliskan disini}

Tugas Modul 2B

Buatlah sebuah algoritma dengan menggunakan Psudocode,
untuk menentukan :

1. Apakah sebuah bilangan adalah bilangan genap atau ganjil ?
2. Manakah rute dengan jalur terpendek, jika ada dua rute yang akan dibandingkan ?
3. Mengurutkan 3 bilangan yang diinputkan, mulai dari yang terkecil sampai yang terbesar !

Deadline : 7 Oktober 2024, maks. Pkl. 23.59 Wita
Upload ke GitHub masing-masing