PROGRAM DATA & PSEUDOCODE

PERTEMUAN 2

Objectives

- To introduce common words, keywords and meaningful names when writing pseudocode
- To define the three basic control structures as set out in the Structure Theorem
- To illustrate the three basic control structures using pseudocode
- To describe program data

Outline

2.1 Program data

2.2 How to write pseudocode
2.3 Meaningful names

2.4 The Structure Theorem

PROGRAM DATA

Karena program ditulis untuk memproses data, Anda harus memiliki pemahaman yang baik tentang sifat dan struktur data yang sedang diproses. Data dalam program dapat berupa variabel tunggal, seperti integer atau karakter, atau item grup (kadang-kadang disebut agregat), seperti array atau file.

Variabel, konstanta dan literal

- Variabel adalah nama yang diberikan untuk kumpulan sel memori yang dirancang untuk menyimpan item data tertentu. Disebut variabel karena nilai yang disimpan dalam sel memori tersebut dapat berubah atau bervariasi saat program dijalankan. Misalnya, variabel yang disebut *total_amount* mungkin berisi beberapa nilai selama eksekusi program

- Konstanta adalah item data dengan nama dan nilai yang tetap sama selama eksekusi program. Misalnya, nama lima puluh dapat diberikan untuk item data yang berisi nilai 50.

- Literal adalah konstanta yang namanya adalah representasi tertulis dari nilainya.

PROGRAM DATA

Tipe data

Di awal program, programmer harus secara jelas mendefinisikan bentuk atau jenis data. Tipe data dapat berupa item dasar data atau struktur data.

Item dasar data

- Item dasar data dasar adalah item yang berisi variabel tunggal yang selalu diperlakukan sebagai satu unit. Item data ini biasanya diklasifikasikan ke dalam tipe data. Tipe data terdiri dari satu set nilai data dan satu set operasi yang dapat dilakukan pada nilai-nilai tersebut. Tipe data dasar yang paling umum adalah:

PROGRAM DATA

Integer

mewakili satu set bilangan bulat, positif, negatif atau nol. Contoh: 3, 576, -5

Float

mewakili satu set angka, positif atau negatif, yang dapat mencakup nilai sebelum atau setelah titik desimal. Contoh: 19.2, 1.92E+01, –0.01

Char

mewakili set karakter pada keyboard, ditambah beberapa special karakter. Contoh: ‘A’, ‘b’, ‘$’

Boolean

mewakili bendera kontrol atau sakelar yang mungkin berisi salah satu dari hanya dua nilai yang mungkin, benar atau salah.

PROGRAM DATA

Struktur data

Struktur data adalah struktur yang terdiri dari item data lainnya. Item data yang dikandungnya adalah komponennya, yang mungkin berupa item data dasar atau struktur data lainnya. Dalam struktur data, data dikelompokkan bersama dengan cara tertentu, yang mencerminkan situasi yang terkait dengan program. Struktur data yang paling umum adalah:

record:

kumpulan item data atau bidang yang semuanya memiliki hubungan dengan satu sama lain. Misalnya, catatan siswa mungkin berisi catatan siswa, nomor, nama, alamat dan mata pelajaran yang terdaftar.

file:

kumpulan catatan terkait. Misalnya, file siswa mungkin berisi: kumpulan catatan tentang siswa.

PROGRAM DATA

Array:

Struktur data yang terdiri dari sejumlah variabel atau data item yang semuanya memiliki tipe data yang sama dan diakses oleh yang sama nama. Misalnya, sebuah array yang disebut skor mungkin berisi kumpulan nilai ujian siswa. Akses ke masing-masing item dalam array adalah dibuat dengan menggunakan indeks atau subscript di samping nama array. Misalnya, skor (3) mewakili skor ketiga dalam larik yang disebut skor.

String:

Kumpulan karakter yang bisa tetap atau variabel. Sebagai contoh, “Jenny Parker” dapat mewakili struktur data string sebagai nama siswa.

PROGRAM DATA

File

Metode penyimpanan informasi yang populer adalah memasukkan dan menyimpan data dalam file. Ada beberapa keuntungan utama menggunakan file:

- Beberapa program berbeda dapat mengakses data yang sama.
- Data dapat dimasukkan dan digunakan kembali (reuse) beberapa kali.
- Data dapat dengan mudah diperbarui (updated) dan dipelihara (maintained).
- Data lebih mudah akurat.

PROGRAM DATA

Ada dua jenis file di mana data dapat disimpan:

• file sequential (berurutan) atau teks, di mana data disimpan dan diambil secara berurutan

• direct or random-access files (file akses langsung atau acak), di mana data disimpan dan diambil secara acak, menggunakan kunci atau indeks.

Note:

File berurutan dapat dibuka untuk membaca atau menulis, tetapi tidak keduanya operasi pada file yang sama. File akses acak dapat dibuka untuk membaca dan menulis di berkas yang sama.

PROGRAM DATA

Validasi data

Data harus selalu menjalani pemeriksaan validasi sebelum diproses oleh sebuah program. Jenis data yang berbeda memerlukan pemeriksaan yang berbeda dan bisa sangat spesifik; namun, pemeriksaan validasi data yang paling umum adalah sebagai berikut:

• Jenis yang benar: data input harus sesuai dengan definisi tipe data yang dinyatakan di awal program.

• Rentang yang benar: data input harus berada dalam kumpulan/rentang nilai yang diperlukan.

• Panjang yang benar: data input – misalnya, string – harus benar panjangnya.

• Kelengkapan: semua bagian wajib harus ada.

• Tanggal yang benar: tanggal masuk harus dapat diterima (beda format penulisan tanggal dapat menyebabkan kesalahan saat validasi data).

PSEUDOCODE

Outline

2.1 How to write pseudocode

2.2 Meaningful names

2.3 The Structure Theorem

PSEUDOCODE

HOW TO WRITE PSEUDOCODE

Saat merancang algoritma, fakta yang perlu diingat bahwa komputer melakukan serangkaian instruksi yang Anda tulis. Jika Anda menggunakan kata dan frasa dalam kode semu (pseudocode) yang sesuai dengan sejumlah enam operasi dasar computer. Enam operasi komputer dasar:

1 Komputer dapat menerima (masukan) informasi

2 Komputer dapat mengeluarkan informasi

3 Komputer dapat melakukan operasi aritmatika (perhitungan)

4 Komputer dapat menetapkan/memberikan nilai ke variabel atau memori

5 Komputer dapat membandingkan dua variabel dan juga dapat memilih salah satu dari dua tindakan alternatif yang ada.

6 Komputer dapat mengulangi sekelompok perintah/tindakan

PSEUDOCODE

Komputer dapat menerima (masukan) informasi

Ketika komputer diperlukan untuk menerima informasi atau input dari sumber tertentu, apakah itu terminal, disk atau perangkat lain, kata kerja **Read** dan **Get** digunakan dalam pseudocode. **Read** biasanya digunakan ketika algoritma menerima input dari record pada file, sedangkan **Get** digunakan ketika algoritma menerima input dari keyboard. Contoh, instruksi pseudocode untuk menerima informasi adalah:

```
Read student name
Get system date
Read number_1, number_2
Get tax_code
```

Setiap contoh menggunakan kata kerja tunggal, **Read** atau **Get**, diikuti oleh satu atau lebih kata benda untuk menunjukkan data apa yang akan diperoleh.

PSEUDOCODE

A computer can put out information (Komputer dapat mengeluarkan informasi)

Ketika komputer diperlukan untuk mengirimkan informasi atau output ke perangkat, kata kerja Print, Write, Put, Output atau Display digunakan dalam pseudocode. Print biasanya digunakan ketika output akan dikirim ke printer, sedangkan Write digunakan ketika output akan ditulis ke file. Jika output akan ditulis ke layar, kata-kata Put, Output atau Display digunakan dalam pseudocode. Contoh pseudocode yang umum adalah:

```
Print 'Program Completed'
Write customer record to master file
Put out name, address and postcode
Output total_tax
Display 'End of data'
```

PSEUDOCODE

A computer can perform arithmetic (Komputer dapat melakukan aritmatika)

Sebagian besar program memerlukan bantuan komputer untuk melakukan berbagai macam perhitungan matematis, atau untuk menerapkan rumus, dan untuk ini seorang programmer dapat menggunakan simbol matematika yang sebenarnya atau kata-kata untuk simbol tersebut. Misalnya, instruksi pseudocode yang sama dapat dinyatakan sebagai salah satu dari berikut ini:

```
add number to total
total = total + number
```

Kedua ekspresi dengan jelas menginstruksikan komputer untuk menambahkan satu nilai ke nilai lainnya, sehingga keduanya dapat diterima dalam pseudocode. Simbol sama dengan '=' digunakan untuk menunjukkan penetapan nilai sebagai hasil dari beberapa pemrosesan.

PSEUDOCODE

Agar konsisten dengan bahasa pemrograman tingkat tinggi, simbol berikut dapat ditulis dalam pseudocode:

- **+** for **add**
- **–** for **subtract**
- ***** for **multiply**
- **/** for **divide**
- **( )** for **parentheses**

Kata kerja Hitung dan Hitung juga tersedia. Beberapa contoh instruksi pseudocode untuk melakukan perhitungan adalah:

```
divide total_marks by student_count
sales_tax = cost_price * 0.10
compute C = (F – 32) * 5/9
```

PSEUDOCODE

Urutan operasi

Saat menulis perhitungan matematis untuk komputer, urutan operasi matematika standar berlaku untuk kodesemu dan sebagian besar bahasa komputer. Operasi pertama yang dilakukan adalah perhitungan apa pun yang ada di dalam tanda kurung. Selanjutnya, perkalian atau pembagian apa pun, seperti yang terjadi dari kiri ke kanan, akan dilakukan. Kemudian, setiap penambahan atau pengurangan, seperti yang terjadi dari kiri ke kanan, akan dilakukan.

PSEUDOCODE

A computer can assign a value to a variable or memory location (Komputer dapat menetapkan nilai ke variabel atau lokasi memori). Ada tiga contoh di mana Anda dapat menulis kodesemu untuk menetapkan nilai ke variabel atau lokasi memori:

1. Untuk memberikan data nilai awal dalam pseudocode, kata kerja Initialise atau Set digunakan.

2. Untuk menetapkan nilai sebagai hasil dari beberapa pemrosesan, simbol '=' atau '←' ditulis.

3. Untuk menyimpan variabel untuk digunakan nanti, kata kerja Save atau Store digunakan.

PSEUDOCODE

Beberapa contoh pseudocode nya:

```
Initialise total_price to zero
Set student_count to 0
total_price = cost_price + sales_tax
total_price  cost_price + sales_tax
store customer_num in last_customer_num
```

Perhatikan bahwa simbol '=' digunakan untuk menetapkan nilai ke variabel sebagai hasil dari beberapa pemrosesan dan tidak setara dengan simbol matematika '='. Untuk alasan ini, beberapa programmer lebih suka menggunakan simbol '←' untuk mewakili operasi penetapan.

PSEUDOCODE

A computer can compare two variables and select one of two alternative actions (Komputer dapat membandingkan dua variabel dan memilih salah satu dari dua tindakan alternatif)

Operasi komputer penting yang tersedia bagi pemrogram adalah kemampuan untuk membandingkan dua variabel dan kemudian, sebagai hasil perbandingan, memilih salah satu dari dua tindakan alternatif. Untuk merepresentasikan operasi ini dalam pseudocode, kata kunci khusus digunakan: IF, THEN dan ELSE. Perbandingan data ditetapkan dalam klausa IF, dan pilihan alternatif ditentukan oleh opsi THEN atau ELSE. Hanya satu dari alternatif ini yang akan dilakukan. Contoh pseudocode yang khas untuk mengilustrasikan operasi ini adalah:

```
IF student_attendance_status is part_time THEN
add 1 to part_time_count
ELSE
add 1 to full_time_count
ENDIF
```

Dalam contoh ini, status kehadiran siswa diselidiki, dengan hasil bahwa akumulasi jumlah_waktu_paruh atau hitung_waktu_penuh bertambah. Perhatikan penggunaan indentasi untuk menekankan opsi THEN dan ELSE, dan penggunaan ENDIF pembatas untuk menutup operasi.

PSEUDOCODE

A computer can repeat a group of actions (Komputer dapat mengulangi sekelompok Tindakan)

Ketika ada urutan langkah pemrosesan yang perlu diulang, dua kata kunci khusus, DOWHILE dan ENDDO, digunakan dalam pseudocode. Kondisi pengulangan sekelompok tindakan ditetapkan dalam klausa DOWHILE, dan tindakan yang akan diulang tercantum di bawahnya. Sebagai contoh:

```
DOWHILE student_total < 50
  Read student record
  Print student name, address to report
  add 1 to student_total
ENDDO
```

Dalam contoh ini, mudah untuk melihat pernyataan yang harus diulang, karena pernyataan tersebut segera mengikuti pernyataan DOWHILE dan diberi indentasi untuk penekanan tambahan. Kondisi yang mengontrol dan akhirnya mengakhiri pengulangan ditetapkan dalam klausa DOWHILE, dan kata kunci ENDDO bertindak sebagai pembatas. Segera setelah kondisi pengulangan ditemukan salah, kontrol diteruskan ke pernyataan berikutnya setelah ENDDO

PSEUDOCODE

Meaningful names

Saat merancang algoritma solusi, seorang programmer harus memperkenalkan beberapa nama unik, yang akan digunakan untuk mewakili variabel atau objek dalam masalah. Semua nama harus bermakna. Nama yang diberikan ke variabel hanyalah metode untuk mengidentifikasi lokasi penyimpanan tertentu di komputer.

Keunikan sebuah nama akan membedakan lokasi ini dengan lokasi lainnya. Seringkali sebuah nama menggambarkan jenis data yang disimpan dalam variabel tertentu. Misalnya, variabel mungkin salah satu dari tiga tipe data sederhana: bilangan bulat, bilangan real atau karakter.

Nama itu sendiri harus cukup transparan untuk menggambarkan variabel secara memadai; misalnya, angka1, angka2 dan angka3 adalah nama yang lebih bermakna untuk tiga angka daripada A, B dan C.

Jika lebih dari satu kata digunakan dalam nama variabel, maka garis bawah berguna sebagai pemisah kata, misalnya pajak_penjualan dan jumlah_kata.

Sebagian besar bahasa pemrograman tidak mentolerir spasi dalam nama variabel, karena spasi akan menandakan akhir dari nama variabel dan dengan demikian menyiratkan bahwa ada dua variabel. Jika garis bawah tidak dapat digunakan, maka kata dapat digabungkan dengan penggunaan huruf kapital sebagai pemisah kata, misalnya Pajak Penjualan dan WordCount. Untuk keterbacaan, tidak disarankan untuk merangkai kata-kata semua dalam huruf kecil. Nama seperti 'carregistration' jauh lebih sulit dibaca daripada 'carRegistration'.

PSEUDOCODE

The Structure Theorem

Teorema Struktur merevolusi desain program dengan membangun kerangka kerja terstruktur untuk mewakili algoritma solusi. Teorema Struktur menyatakan bahwa adalah mungkin untuk menulis program komputer apa pun dengan hanya menggunakan tiga struktur kontrol dasar yang mudah direpresentasikan dalam pseudocode: urutan, seleksi dan pengulangan.

PSEUDOCODE

Urutan

Struktur kontrol urutan adalah eksekusi langsung dari satu langkah pemrosesan demi satu. Dalam pseudocode, konstruksi ini direpresentasikan sebagai urutan pernyataan pseudocode:

```
statement a
statement b
statement c
```

Struktur kontrol urutan dapat digunakan untuk mewakili empat operasi komputer dasar pertama yang terdaftar sebelumnya: menerima informasi, mengeluarkan informasi, melakukan aritmatika, dan menetapkan nilai. Misalnya, urutan pernyataan yang khas dalam suatu algoritma mungkin berbunyi:

```
add 1 to pageCount
Print heading line1
Print heading line2
Set lineCount to zero
Read customer record
```

Instruksi ini menggambarkan struktur kontrol urutan sebagai straightforward list yang ditulis satu demi satu, secara top-to-bottom. Setiap instruksi akan dieksekusi sesuai urutan kemunculannya.

Selection

Struktur kontrol seleksi adalah penyajian suatu kondisi dan pilihan antara dua tindakan, pilihan tergantung pada apakah kondisinya benar atau salah. Konstruk ini mewakili kemampuan pengambilan keputusan komputer dan digunakan untuk menggambarkan operasi komputer dasar kelima, yaitu membandingkan dua variabel dan memilih salah satu dari dua alternatif tindakan. Dalam pseudocode, seleksi diwakili oleh kata kunci IF, THEN, ELSE dan ENDIF:

```
IF condition p is true THEN
   statement(s) in true case
ELSE
   statement(s) in false case
ENDIF
```

Jika kondisi p benar, maka pernyataan atau pernyataan dalam kasus benar akan dieksekusi, dan pernyataan dalam kasus salah akan dilewati. Jika tidak (pernyataan ELSE) pernyataan dalam kasus benar akan dilewati dan pernyataan dalam kasus salah akan dieksekusi. Dalam kedua kasus, kontrol kemudian lolos ke langkah pemrosesan berikutnya setelah pembatas ENDIF. Contoh pseudocode yang khas mungkin berbunyi:

```
IF student_attendance_status is part_time THEN
   add 1 to part_time_count
ELSE
   add 1 to full_time_count
ENDIF
```


PSEUDOCODE

Repetition

Struktur kendali repetisi dapat didefinisikan sebagai penyajian sekumpulan instruksi yang akan dilakukan berulang-ulang, selama kondisinya benar. Ide dasar dari kode berulang adalah bahwa blok pernyataan dieksekusi berulang-ulang, sampai kondisi terminasi terjadi. Konstruksi ini mewakili operasi komputer dasar keenam, yaitu mengulangi sekelompok tindakan. contoh pseudocode repetition sebagai:

```
DOWHILE condition p is true
statement block
ENDDO
```

Perulangan DOWHILE adalah perulangan keputusan terdepan; yaitu, kondisi diuji sebelum pernyataan apa pun dieksekusi. Jika kondisi dalam pernyataan DOWHILE ditemukan benar, blok pernyataan setelah pernyataan itu dieksekusi satu kali. Pembatas ENDDO kemudian memicu kembalinya kontrol ke pengujian ulang kondisi. Jika kondisi masih benar, pernyataan diulangi, dan proses pengulangan berlanjut sampai kondisi ditemukan salah. Kontrol kemudian diteruskan ke pernyataan yang mengikuti pernyataan ENDDO.

PSEUDOCODE

Sangat penting bahwa setidaknya satu pernyataan dalam blok pernyataan mengubah kondisi dan akhirnya menjadikannya salah, karena jika tidak, logika dapat menghasilkan loop tanpa akhir. Berikut adalah contoh pseudocode yang mewakili struktur kontrol pengulangan:

```
Set student_total to zero
DOWHILE student_total < 50
    Read student record
    student name, address to report
    add 1 to student_total
ENDDO
```

Contoh ini menggambarkan sejumlah poin:

1 Variabel student_total diinisialisasi sebelum kondisi DOWHILE dieksekusi.

2 Selama total_siswa kurang dari 50 (yaitu, kondisi DOWHILE benar), blok pernyataan akan diulang.

3 Setiap kali blok pernyataan dieksekusi, satu instruksi di dalam blok itu akan menyebabkan variabel student_total bertambah.

4 Setelah 50 iterasi, total_siswa akan sama dengan 50, yang menyebabkan kondisi DOWHILE menjadi salah dan pengulangan berhenti. Penting untuk disadari bahwa inisialisasi dan peningkatan selanjutnya dari variabel yang diuji dalam kondisi adalah fitur penting dari konstruk DOWHILE.