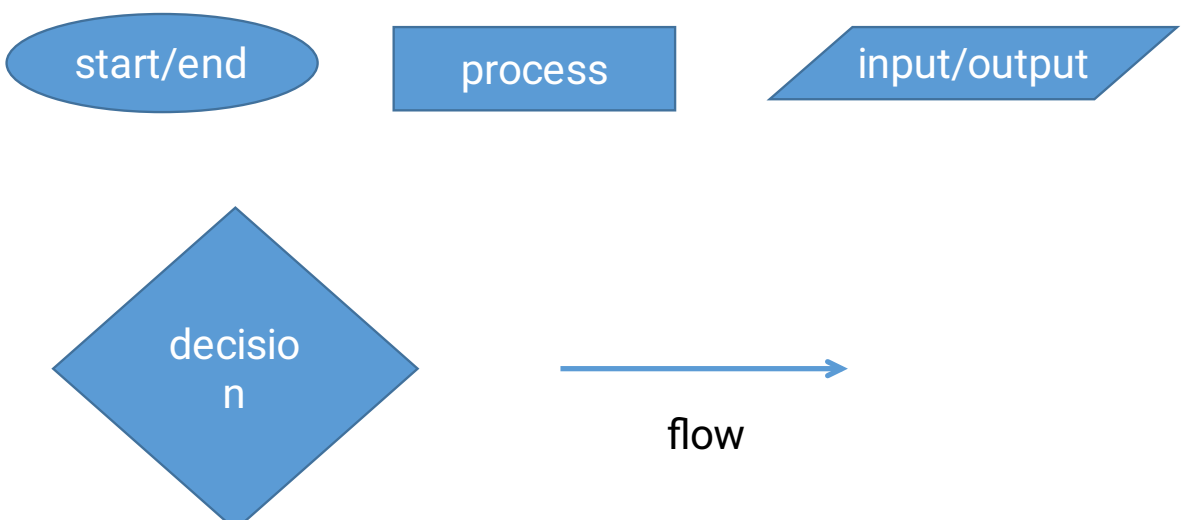
Mengimplementasikan Algoritma Pemrograman

Algoritma

- Langkah-langkah logis untuk menyelesaikan suatu masalah
- Flow Chart



Algoritma

- Sequence -> dijalankan secara terurut dari atas ke bawah
- Selection -> dijalankan bercabang sesuai dengan kondisi yang diberikan
- Repitition -> dijalankan berulang sesuai dengan kondisi yang diberikan

Control Flow/Control Program

- Condition & loops
- Return & jump
- Exception

Condition

- if -> menjalankan suatu (block) pernyataan bila suatu kondisi bernilai true
  - dengan kata kunci if
  - memasukkan kondisi di dalam () yang memiliki nilai boolean
  - membutuhkan {} jika ingin menggunakan statement selbih dari satu
  - menambahkan kata kunci else setelah pernyataan if untuk menjalankan pernyataan bila kondisi bernilai false
  - Setelah kata kunci else dapat ditambahkan kata kunci if lagi untuk mengecek kondisi yang berbeda
- when -> menjalankan (block) pernyataan bila suatu kondisi untuk variabel yang dicek sama dengan suatu nilai
  - dengan kata kunci when
  - masing-masing nilai yang digunakan untuk mengecek variabel dipisahkan dengan ->
  - untuk pernyataan yang tidak memenuhi semua nilai yang ada dapat menggunakan kata kunci else
  - untuk kondisi nilai yang memiliki pernyataan yang sama, dapat digabung dengan tanda koma

Latihan 1 Program Grade Nilai

- Buatlah program menggunakan Kotlin sehingga menghasilkan keluaran sebagai berikut:

- - Batasan skor 0 - 100.
- - Skor 0 – 60 masuk grade nilai D
- - Skor 61 – 70 masuk grade nilai C
- - Skor 71 – 80 masuk grade nilai B
- - Skor 81 – 100 masuk grade nilai A

Latihan 2

- - Ubahlah kode pada Tugas 1 menggunakan when.
- - Tambahkan keterangan:
- Bila kategori nilai D “tidak lulus”
- Bila kategori nilai C “remedial”
- Bila kategori nilai A dan B “lulus”

Latihan 3

- Pemberian diskon pembelian

- Hasil diskon = Harga beli - (Harga beli * 30 / 100)

- Harga beli di atas 0
- Harga beli di bawah 500.000 -> diskon 5%
- Harga beli 500.000 - 749.999 -> diskon 10%
- Harga beli 750.000 - 999.999 -> diskon 15%
- Harga beli di atas 1.000.000 -> diskon 30%

Latihan 4

- Pemberian diskon pembelian berdasarkan poin member
- Hasil diskon = Harga beli - (Harga beli * 30 / 100)

- Batas poin 0 - 1000
- Poin 0 - 100 -> member biasa
- Poin 101 - 300 -> silver
- Poin 301 - 500 -> gold
- Poin 501 - 1000 -> platinum

Harga Pembelian: 500.000
Poin: 1000
Status: platinum
Hasil Diskon: Harga beli - (Harga beli * diskon%)

- Member silver -> diskon 5%
- Member gold -> diskon 15%
- Member platinum -> diskon 30%

While Loop

- While -> mengulang pernyataan selama kondisi bernilai benar
  - Ada kemungkinan tidak mengeksekusi sama sekali pernyataan
  - Menggunakan kata kunci while diikuti () yang berisi kondisi untuk pengulangan
- Do - While -> mengeksekusi pernyataan terlebih dahulu setelah itu disek apakah pernyataan tersebut perlu diulang atau tidak
  - Pasti mengeksekusi pernyataan sebanyak 1 kali
  - Menggunakan kata kunci do diikuti pernyataan yang ingin diulang, lalu menggunkan kata kunci while diikuti () yang berisi kondisi untuk pengulangan
- Pastikan kondisi untuk pengulangan dapat bernilai false
- Digunakan untuk pengulangan pernyataan yang tidak diketahui jumlah pengulangannya

Latihan 5

- Buat program untuk konversi bilangan desimal ke bilangan biner dengan while loop

For Loop

- For pada kotlin digunakan untuk pengulangan terhadap suatu koleksi data
- Mengulang sebanyak jumlah item pada array
  - Variabel pada sebelah kiri kata kunci in akan berisi value array
  - Variabel pada sebelah kiri kata kunci in dapat diisi dengan nomor index array dengan menambahkan property indeces pada array
  - Variabel pada sebelah kiri kata kunci in dapat diisi dengan nomor index array beserta value array dengan menambahkan method withIndex() pada array
- Mengulang sebanyak nilai pada range expression
  - Operator .. -> 0..5 -> 0, 1, 2, 3, 4, 5
  - Kata kunci downTo -> 5 downTo 0 -> 5, 4, 3, 2, 1, 0
  - Kata kunci until -> 0 until 5 -> 0, 1, 2, 3, 4
  - Dapat ditambahkan kata kunci step untuk mengubah jumlah iterasinya (default = 1)

Latihan 6

- Buat program untuk konversi bilangan biner ke bilangan desimal dengan for loop

Tugas Control Flow

- Buatlah project baru bernama “TugasControlFlow{Nama}”
- Tulis kode program menggunakan control flow for loops sehingga menghasilkan output sebagai berikut : video

Function

- Urutan sequence dari beberapa instruksi kode program, yang menjalankan suatu tugas tertentu/spesifik
- Dideklarasikan dengan kata kunci fun, diikuti identifier dilanjutkan dengan penentuan parameter dengan simbol (), lalu menentukan type output diakhiri dengan deklarasi statements
- Function dapat memiliki satu atau lebih parameter
- Deklarasi parameter dilakukan dengan cara menentukan identifier dan menentukan type
- Parameter adalah immutable variable ber-scope local (hanya dapat digunakan di dalam function)
- Pemanggilan function dilakukan dengan menulis identifier function diikuti kurung () yang dapat diisi suatu argumen

Function

- Parameter : variabel yang dideklarsikan saat mendeklarasikan function
- Argumen : nilai yang dimasukkan ke dalam parameter saat pemanggilan function
- Argumen dapat diisi dengan suatu nilai, variabel atau expression
- Type argumen yang dimasukkan harus sama dengan type parameter

Control Flow Return

- Kata kunci untuk keluar dari function
- Dapat menghasilkan nilai dengan menambahkan nilai, variable, atau expression di sebelah kanan kata kunci

Unit Function

- Function yang tidak memiliki statement return, akan mengembalikan object Unit
- Unit adalah object yang hanya memiliki sebuah nilai, yaitu nilai Unit
- Digunakan untuk membuat function yang tidak memerlukan output

Parameter

- Mendeklarasikan lebih dari sebuah parameter dengan trailing comma, yaitu menambahkan simbol ‘,’
- Deklasari parameter dapat dibuat multiline layaknya suatu statement
- Default argument memberikan nilai inisialisasi pada parameter
- Parameter yang menggunakan default argument, tidak perlu mencantumkan lagi argument pada saat pemanggilan function
- Bila menambahkan default argument sebelum parameter yang tidak memiliki default argument, pengisian argument terhadap parameter yang tidak memiliki default argument dapat dilakukan dengan named arugument
- Named argument adalah argument yang diberikan identifier yang sama dengan identifier parameter
- Urutan argument dapat diubah dengan bebas bila menggunakan named argument

Single-expression Function

- Function dengan sebuah pernyataan
- Tidak membutuhkan simbol {}
- Menulis expression setelah simbol =

Vararg

- Kata kunci untuk membuat paramater yang berisi beberapa argumen dalam bentuk struktur array
- Dapat dipanggil menggunakan named argument
- Bila menggunakan named argument, nilai/value yang dimasukkan berupa array
- Parameter yang menggunakan kata kunci vararg dapat digunakan layaknya sebuah array di dalam function

Latihan 7

- Ubahlah kode program pada Latihan 1 - 6 menggunakan function

Tugas Function

- Buatlah project baru bernama “TugasFunction{Nama}”
- Ubahlah kode program pada Tugas Control Flow menggunakan function

Function Type

- Type yang dapat diisi dengan lambda expression atau anonymous function pada suatu variabel
- Dapat memiliki parameter dan return type
- Pemanggilan function type dilakukan dengan cara yang sama dengan pemanggilan function
- Parameter function type dapat diberi identifier
- Function type harus mencantumkan type Unit bila memiliki return type Unit

Lambda Expression

- Mengisi argumen atau variabel dengan function
- Diawali dengan {}, lalu diisi dengan identifier parameter yang digunakan bila ada, selanjutnya menggunakan simbol ->, dan terakhir badan statement
- Nilai return dari lambda expression adalah nilai dari expression terakhir
- Kata kunci it dapat digunakan untuk menunjuk parameter lambda expression bila hanya memiliki sebuah parameter
- Untuk parameter yang tidak digunakan, dapat digantikan dengan simbol _
- Dapat menggunakan variabel yang berada di luar scope lambda expression