**Topik:** Dekomposisi, Abstraksi, dan Pengenalan Pola dalam Konteks Pemrograman Prosedural

**Tujuan Praktikum:**

Mahasiswa memahami dekomposisi, abstraksi, dan pengenalan pola serta dapat menerapkannya dalam konteks pemrograman prosedural dengan memanfaatkan konsep fungsi/prosedur dan type bentukan dengan menggunakan Bahasa Pascal.

**PETUNJUK PRAKTIKUM:**

1. Tuliskan jawaban setiap soal di sebuah file program Pascal yang diberi nama P05<nim>YYMMHHXX.pas dengan:

* <nim> : NIM peserta
* YY : 2 digit terakhir tahun pengumpulan tugas,
* MM : bulan pengumpulan tugas,
* HH : tanggal pengumpulan tugas,
* XX : nomor soal (2 digit, contoh: 01).

1. Untuk setiap file yang Anda buat, buat header sebagai berikut:

(\* NIM/Nama : \*)

(\* Nama file : \*)

(\* Topik : \*)

(\* Tanggal : \*)

(\* Deskripsi : \*)

1. Untuk semua program, gunakan standar *coding* yang baik yang disampaikan di kuliah. Untuk semua fungsi/prosedur, harus dibuat definisi, spesifikasi, implementasinya.
2. *Softcopy* materi kuliah, termasuk yang terkait dengan pemrograman dengan Bahasa Pascal dapat dilihat pada situs **http://kuliah.itb.ac.id/app245** pada link **IF1210 Dasar Pemrograman**.
3. HANYA FILE YANG DAPAT DI-*COMPILE* YANG AKAN DIPERIKSA. File yang tidak dapat di-compile akan otomatis mendapatkan nilai 0.
4. Pengerjaan praktikum bersifat INDIVIDUAL. Tidak ada toleransi bagi pencontek. Jika terbukti, baik yang dicontek maupun yang mencontek akan mendapatkan nilai 0 dan terancam tidak lulus kuliah.
5. Tidak boleh ada keterlambatan dalam pengumpulan hasil praktikum.

**SELAMAT BEKERJA ☺**

**Soal 1. Program Operasi Matriks 3x3 (Bobot: 50%)**

Soal ini telah dibahas di perkuliahan IF1210 Kamis, 19 Maret 2015 sebagai studi kasus-3.

*Buatlah sebuah program yang digunakan untuk melakukan operasi matriks dengan elemen bertype integer dengan ukuran 3 x 3. Program diawali dengan membaca masukan elemen 2 buah matriks 3 x 3, yaitu matriks A dan matriks B dari pengguna.*

*Selanjutnya, diberikan pilihan menu (integer) yang bisa digunakan untuk memilih operasi matriks yang akan dilakukan, yaitu:*

* *pilihan 1 untuk operasi penjumlahan kedua matriks dan menuliskan hasilnya,*
* *pilihan 2 untuk operasi pengurangan matriks A dengan matriks B dan menuliskan hasilnya, serta*
* *pilihan 3 adalah operasi untuk memeriksa dan menuliskan apakah kedua matriks adalah matriks satuan atau tidak.*

*Matriks satuan adalah matriks yang elemen-elemennya hanya terdiri atas angka 0 dan/atau 1.*

*Jika dimasukkan pilihan selain 1, 2, 3, maka dituliskan pesan kesalahan “Bukan pilihan yang benar”. Tidak perlu ada validasi masukan pilihan dengan pengulangan.*

Telah dipaparkan 4 alternatif solusi. Buatlah implementasi solusi ke-4 yaitu dengan memanfaatkan type data matriks (lihat slides kuliah).

Program tersebut harus dibuat dengan mengubah potongan program sbb. Bagian program utama tidak boleh diubah.

Program Matriks3x3;

{ I.S.: 2 matriks of integer 3x3, mis. matriks A dan B; pilihan menu

(integer)

F.S.: Jika pilihan menu = 1, dituliskan hasil penjumlahan kedua matriks

Jika pilihan menu = 2, dituliskan hasil pengurangan matriks A dengan

matriks B

Jika pilihan menu = 3, dituliskan apakah kedua matriks adalah matriks

satuan atau bukan

Jika pilihan menu yang lain, dituliskan "Bukan pilihan yang benar" }

{ KAMUS }

var

MA, MB, MHasil : Matriks;

pilihan : integer;

{ ALGORITMA }

begin

{ Mengisi matriks }

writeln('Masukan Matriks A = ');

InputMatriks(MA);

writeln('Masukan Matriks B = ');

InputMatriks(MB);

{ Baca pilihan menu dan lakukan operasi sesuai pilihan menu }

write('Masukkan pilihan menu (1,2,3) = ');

readln(pilihan);

case (pilihan) of

1 : begin

{ Menjumlahkan kedua matriks dan mencetak hasilnya }

Operasi2Matriks(MA,MB,'+',MHasil);

writeln('Hasil penjumlahan matriks A dan B =');

CetakMatriks(MHasil);

end;

2 : begin

{ Mengurangkan matriks A dengan B dan mencetak hasilnya }

Operasi2Matriks(MA,MB,'-',MHasil);

writeln('Hasil pengurangan matriks A dengan B =');

CetakMatriks(MHasil);

end;

3 : begin

{ Cek apakah kedua matriks adalah matriks satuan/bukan }

{ Cek MA }

write('Matriks A '); CetakMatriksSatuan(MA);

{ Cek MB }

write('Matriks B '); CetakMatriksSatuan(MB);

end;

else { pilihan menu bukan 1, 2, 3 }

begin

writeln('Bukan pilihan yang benar');

end;

end;

end.

**Alternatif Solusi:**

Yang **dicetak tebal** adalah tambahan kode

Program Matriks3x3;

{ I.S.: 2 matriks of integer 3x3, mis. matriks A dan B; pilihan menu

(integer)

F.S.: Jika pilihan menu = 1, dituliskan hasil penjumlahan kedua matriks

Jika pilihan menu = 2, dituliskan hasil pengurangan matriks A dengan

matriks B

Jika pilihan menu = 3, dituliskan apakah kedua matriks adalah matriks

satuan atau bukan

Jika pilihan menu yang lain, dituliskan "Bukan pilihan yang benar" }

{ KAMUS }

**{ Type Bentukan Matriks }**

**type**

**Matriks = array [1..3] of array [1..3] of integer;**

var

MA, MB, MHasil : Matriks;

pilihan : integer;

**{ DEKLARASI DAN REALISASI FUNGSI DAN PROSEDUR }**

**procedure InputMatriks(var M : Matriks);**

**{ I.S.: M sembarang }**

**{ F.S.: Setiap elemen M terdefinisi berdasarkan pembacaan dari keyboard }**

**{ Kamus Lokal }**

**var**

**i, j : integer;**

**{ Algoritma }**

**begin**

**for i:=1 to 3 do**

**begin**

**for j:=1 to 3 do**

**begin**

**write('Elemen ke-[', i, ',', j, '] = ');**

**readln(M[i][j]);**

**end;**

**end;**

**end;**

**procedure CetakMatriks(M : Matriks);**

**{ I.S.: M terdefinisi }**

**{ F.S.: M tercetak ke layar dalam bentuk matriks 3x3 }**

**{ Kamus Lokal }**

**var**

**i, j : integer;**

**{ Algoritma }**

**begin**

**for i:=1 to 3 do**

**begin**

**for j:=1 to 3 do**

**begin**

**write(M[i][j], ' ');**

**end;**

**writeln;**

**end**

**end;**

**procedure Operasi2Matriks (M1, M2 : Matriks; op : char;**

**var MHasil : Matriks);**

**{ I.S. M1 dan M2 sudah terdefinisi dan terisi, op terdefinisi dengan nilai**

**'+' atau '-' }**

**{ F.S. MHasil berisi hasil operasi M1 dengan M2 tergantung op }**

**{ KAMUS LOKAL }**

**var**

**i, j : integer;**

**{ Algoritma }**

**begin**

**for i:=1 to 3 do**

**begin**

**for j:=1 to 3 do**

**begin**

**if (op = '+') then**

**MHasil[i][j] := M1[i][j] + M2[i][j]**

**else { op = '-' }**

**MHasil[i][j] := M1[i][j] - M2[i][j];**

**end;**

**end;**

**end;**

**function IsMatriksSatuan(M : Matriks) : boolean;**

**{ Menghasilkan true jika M adalah matriks satuan, false jika tidak }**

**{ Alternatif-2 : menggunakan counting elements }**

**{ Kamus Lokal }**

**var**

**i, j : integer;**

**count : integer; { counter nilai bukan 0 dan 1 }**

**{ Algoritma }**

**begin**

**count := 0;**

**for i:=1 to 3 do**

**begin**

**for j:=1 to 3 do**

**begin**

**if (M[i][j] <> 0) and (M[i][j] <> 1) then**

**begin**

**count := count + 1;**

**end;**

**end;**

**end;**

**IsMatriksSatuan := (count = 0); { jika count = 0, berarti tidak ada elemen bukan 0/1 }**

**end;**

**procedure CetakMatriksSatuan (M : matriks);**

**{ I.S. M sudah terdefinisi }**

**{ F.S. Tercetak ke layar apakah MA matriks satuan atau bukan }**

**{ KAMUS LOKAL }**

**{ ALGORITMA }**

**begin**

**if (IsMatriksSatuan(M)) then**

**begin**

**writeln('adalah matriks satuan');**

**end else**

**begin**

**writeln('bukan matriks satuan');**

**end;**

**end;**

{ ALGORITMA }

begin

{ Mengisi matriks }

writeln('Masukan Matriks A = ');

InputMatriks(MA);

writeln('Masukan Matriks B = ');

InputMatriks(MB);

{ Baca pilihan menu dan lakukan operasi sesuai pilihan menu }

write('Masukkan pilihan menu (1,2,3) = ');

readln(pilihan);

case (pilihan) of

1 : begin

{ Menjumlahkan kedua matriks dan mencetak hasilnya }

Operasi2Matriks(MA,MB,'+',MHasil);

writeln('Hasil penjumlahan matriks A dan B =');

CetakMatriks(MHasil);

end;

2 : begin

{ Mengurangkan matriks A dengan B dan mencetak hasilnya }

Operasi2Matriks(MA,MB,'-',MHasil);

writeln('Hasil pengurangan matriks A dengan B =');

CetakMatriks(MHasil);

end;

3 : begin

{ Cek apakah kedua matriks adalah matriks satuan/bukan }

{ Cek MA }

write('Matriks A '); CetakMatriksSatuan(MA);

{ Cek MB }

write('Matriks B '); CetakMatriksSatuan(MB);

end;

else { pilihan menu bukan 1, 2, 3 }

begin

writeln('Bukan pilihan yang benar');

end;

end;

end.

**Soal 2. Titik 2 Dimensi (Bobot: 50%)**

Buatlah sebuah program yang mengelola data yang merepresentasikan titik dalam bidang kartesius <x,y > dengan:

* x : real, merepresentasikan bagian absis dari titik
* y : real, merepresentasikan bagian ordinat dari titik

Contoh: <1,1> merepresentasikan titik <1,1>

Program menerima masukan 3 buah titik, misalnya P1 <x1,y1>, P2 <x2,y2>, dan P3 <x3,y3> dari pengguna, dan melakukan hal sbb:

* Jika ketiga titik dapat membentuk segitiga, maka hitunglah keliling segitiga dan tuliskan pesan sbb: “Keliling segitiga yang dibentuk dari titik (<x1>,<y1>); (<x2>,<y2>); (<x3>,<y3>) adalah <keliling>”

2 titik dapat membentuk sebuah segmen garis lurus. 3 buah titik membentuk 3 buah segmen garis. Jika ke-3 titik adalah P1, P2, dan P3, maka segmen garis yang terbentuk adalah P1-P2, P1-P3, dan P2-P3.

3 titik dapat membentuk segitiga, jika panjang segmen garis terpanjang lebih kecil daripada jumlah total panjang 2 segmen garis yang lain.

Contoh:

* + Jika panjang P1-P2 = 5, panjang P1-P3 = 2, panjang P2-P3 = 4, maka P1, P2, P3 dapat membentuk segitiga
  + Jika panjang P1-P2 = 5, panjang P1-P3 = 2, panjang P2-P3 = 9, maka P1, P2, P3 tidak dapat membentuk segitiga

Keliling segitiga yang terbentuk dari segmen garis P1-P2, P1-P3, dan P2-P3 adalah panjang P1-P2 + panjang P1-P3 + panjang P2-P3.

* Jika ketiga titik tidak dapat membentuk segitiga, maka tuliskan pesan: “Segitiga tidak dapat dibentuk dari titik (<x1>,<y1>); (<x2>,<y2>); (<x3>,<y3>)”.

Bagian <...> diganti dengan nilai sesuai dengan masukan dari pengguna atau berdasarkan hasil perhitungan.

Petunjuk:

* Proses di atas membutuhkan perhitungan panjang (real) segmen garis yang dibentuk oleh 2 titik. Untuk menghasilkan panjang segmen garis yang dibentuk dari titik <x1,y1> dan <x2,y2>, gunakan rumus:

sqrt((x1-x2)^2 + (y1-y2)^2)

Gunakan fungsi sqrt bawaan Pascal untuk menghitung akar kuadrat.

* Penulisan titik <x,y> selalu dalam dalam bentuk (x,y) dan dapat menjadi bagian dari suatu kalimat seperti contoh di atas. Buatlah menjadi prosedur.
* Identifikasi seoptimal mungkin dan buatlah fungsi, prosedur, serta type bentukan yang bisa dimanfaatkan untuk persoalan di atas.

**Alternatif Solusi:**

Alternatif dekomposisi yang lain sangat dimungkinkan.

Program MengelolaPoint;

{ Input : 3 buah Point, misalnya P1, P2, dan P3

Output : Jika P1, P2, dan P3 dapat membentuk segitiga, tulis keliling

segitiga yang terbentuk

Jika tidak, tuliskan pesan bahwa titik tidak dapat membentuk

segitiga }

{ KAMUS }

type Point = record

X : real; { absis }

Y : real; { ordinat }

end;

var

P1, P2, P3 : Point;

{ DEKLARASI DAN REALISASI FUNGSI DAN PROSEDUR }

function Jarak (P1, P2 : Point) : real;

{ menghasilkan jarak antara P1 dengan P2 }

{ KAMUS LOKAL }

{ ALGORITMA }

begin

Jarak := sqrt((P1.X-P2.X) \* (P1.X-P2.X) + (P1.Y-P2.Y) \* (P1.Y-P2.Y));

end;

function IsMembentukSegitiga (P1, P2, P3 : Point) : boolean;

{ true jika P1, P2, dan P3 dapat membentuk segitiga }

var

j1, j2, j3 : real;

{ ALGORITMA }

begin

j1 := Jarak(P1,P2); j2 := Jarak(P1,P3); j3 := Jarak(P2,P3);

if (j1 > j2) and (j1 > j3) then

begin

IsMembentukSegitiga := j1 < (j2 + j3);

end else if (j2 > j1) and (j2 > j3) then

begin

IsMembentukSegitiga := j2 < (j1 + j3);

end else { j3 paling besar }

begin

IsMembentukSegitiga := j3 < (j1 + j2);

end;

end;

function KelilingSegitiga (P1, P2, P3 : Point) : real;

{ menghasilkan keliling segitiga yang dibentuk dari P1, P2, P3 }

{ KAMUS LOKAL }

{ ALGORITMA }

begin

KelilingSegitiga := Jarak(P1,P2) + Jarak(P1,P3) + Jarak(P2,P3);

end;

procedure BacaPoint (var P : Point);

{ I.S. P sembarang }

{ F.S. P terdefinisi dengan elemen dibaca dari user }

{ KAMUS LOKAL }

{ ALGORITMA }

begin

write('Masukkan absis = '); readln(P.X);

write('Masukkan ordinat = '); readln(P.Y);

end;

procedure TulisPoint (P : Point);

{ I.S. P terdefinisi }

{ F.S. P tercetak di layar sebagai bagian dari teks dengan format (X,Y) }

{ KAMUS LOKAL }

{ ALGORITMA }

begin

write('(', P.X:0:2, ',', P.Y:0:2, ')');

end;

{ ALGORITMA }

begin

{ Masukan Point }

writeln('Point pertama =');

BacaPoint(P1);

writeln('Point kedua =');

BacaPoint(P2);

writeln('Point ketiga =');

BacaPoint(P3);

{ Hitung Operasi dan menampilkan ke layar }

if (IsMembentukSegitiga(P1,P2,P3)) then

begin

write('Keliling segitiga yang dibentuk dari titik ');

TulisPoint(P1);

write('; ');

TulisPoint(P2);

write('; ');

TulisPoint(P3);

writeln(' adalah ', KelilingSegitiga(P1,P2,P3):0:2);

end else { P1, P2, P3 tidak dapat membentuk segitiga }

begin

write('Segitiga tidak dapat dibentuk dari titik ');

TulisPoint(P1);

write('; ');

TulisPoint(P2);

writeln('; ');

TulisPoint(P3);

writeln;

end;

end.