**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. Pecahan Non-Negatif (Bobot: 50%)**

Buatlah sebuah program yang mengelola data yang merepresentasikan pecahan non-negatif (0 atau positif) <n,d> dengan:

* n : integer >= 0, merepresentasikan bagian pembilang (numerator) dari pecahan. Jika n = 0, maka pecahan adalah pecahan nol.
* y : integer > 0, merepresentasikan bagian penyebut (denominator) dari pecahan.

Contoh: <1,2> merepresentasikan pecahan 1/2.

Program menerima masukan 2 buah pecahan, misalnya P1 <n1,d1> dan P2 <n2,d2> dari pengguna (asumsikan masukan pecahan selalu dalam *range* yang benar sesuai dengan definisi di atas), dan melakukan hal sbb:

* Jika P1 lebih besar atau sama dari P2, maka hitunglah selisih antara kedua pecahan (dalam bentuk pecahan, misalnya P3 <n3,d3>), lalu menuliskan pesan sbb: “<n1>/<d1> - <n2>/<d2> = <n3/d3>”

Selisih pecahan <n1,d1> dengan <n2,d2> menghasilkan pecahan <n3,d3> dengan n3 = n1\*d2 ‑ n2\*d1 dan d3 = d1\*d2.

Pecahan <n1,d1> lebih besar atau sama dengan <n2,d2> jika dan hanya jika n1\*d2 >= n2\*d1.

* Jika P1 lebih kecil dari P2, maka hitunglah penjumlahan antara kedua pecahan (dalam bentuk pecahan, misalnya P3 <n3,d3>), lalu menuliskan pesan sbb: “Jumlah <n1>/<d1> + <n2>/<d2> = <n3/d3>”

Penjumlahan pecahan <n1,d1> dengan <n2,d2> menghasilkan pecahan <n3,d3> dengan n3 = n1\*d2 + n2\*d1 dan d3 = d1\*d2.

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

Petunjuk:

* Penulisan pecahan <n,d> selalu dalam dalam bentuk n/d 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 MengelolaPecahan;

{ Input : 2 buah Pecahan non-negatif, misalnya P1 dan P2

Output : Jika P1 < P2, hitung selisih antara P1 dan P2 dan menuliskan

hasilnya ke lauar

Jika P1 >= P2, hitung jumlah P1 dan P2 dan menuliskan hasil ke

layar }

{ KAMUS }

type Pecahan = record

n : integer; { integer >= 0; pembilang }

d : integer; { integer > 0; penyebut }

end;

var

P1, P2 : Pecahan;

{ DEKLARASI DAN REALISASI FUNGSI DAN PROSEDUR }

function SubPecahan (P1, P2 : Pecahan) : Pecahan;

{ menghasilkan P1-P2 (dalam bentuk pecahan) }

{ KAMUS LOKAL }

var

P : Pecahan;

{ ALGORITMA }

begin

P.n := P1.n \* P2.d - P2.n \* P1.d;

P.d := P1.d \* P2.d;

SubPecahan := P;

end;

function AddPecahan (P1, P2 : Pecahan) : Pecahan;

{ menghasilkan P1+P2 (dalam bentuk pecahan) }

{ KAMUS LOKAL }

var

P : Pecahan;

{ ALGORITMA }

begin

P.n := P1.n \* P2.d + P2.n \* P1.d;

P.d := P1.d \* P2.d;

AddPecahan := P;

end;

function IsPecahanGTEq (P1, P2 : Pecahan) : boolean;

{ true jika P1 >= P2 }

{ KAMUS LOKAL }

{ ALGORITMA }

begin

IsPecahanGTEq := P1.n \* P2.d >= P2.n \* P1.d;

end;

procedure BacaPecahan (var P : Pecahan);

{ I.S. P sembarang }

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

{ Asumsi: masukan dari user dianggap selalu benar }

{ KAMUS LOKAL }

{ ALGORITMA }

begin

write('Masukkan pembilang = '); readln(P.n);

write('Masukkan penyebut = '); readln(P.d);

end;

procedure TulisPecahan (P : Pecahan);

{ I.S. P terdefinisi }

{ F.S. P tercetak di layar sebagai bagian dari teks dengan format n/d}

{ KAMUS LOKAL }

{ ALGORITMA }

begin

write(P.n, '/', P.d);

end;

{ ALGORITMA }

begin

{ Masukan pecahan }

writeln('Pecahan pertama =');

BacaPecahan(P1);

writeln('Pecahan kedua =');

BacaPecahan(P2);

{ Hitung Operasi dan menampilkan ke layar }

if (IsPecahanGTEq(P1,P2)) then

begin

TulisPecahan(P1);

write(' - ');

TulisPecahan(P2);

writeln(' = ');

TulisPecahan(SubPecahan(P1,P2));

writeln;

end else { P1 lebih kecil dari P2 }

begin

TulisPecahan(P1);

write(' + ');

TulisPecahan(P2);

writeln(' = ');

TulisPecahan(AddPecahan(P1,P2));

writeln;

end;

end.