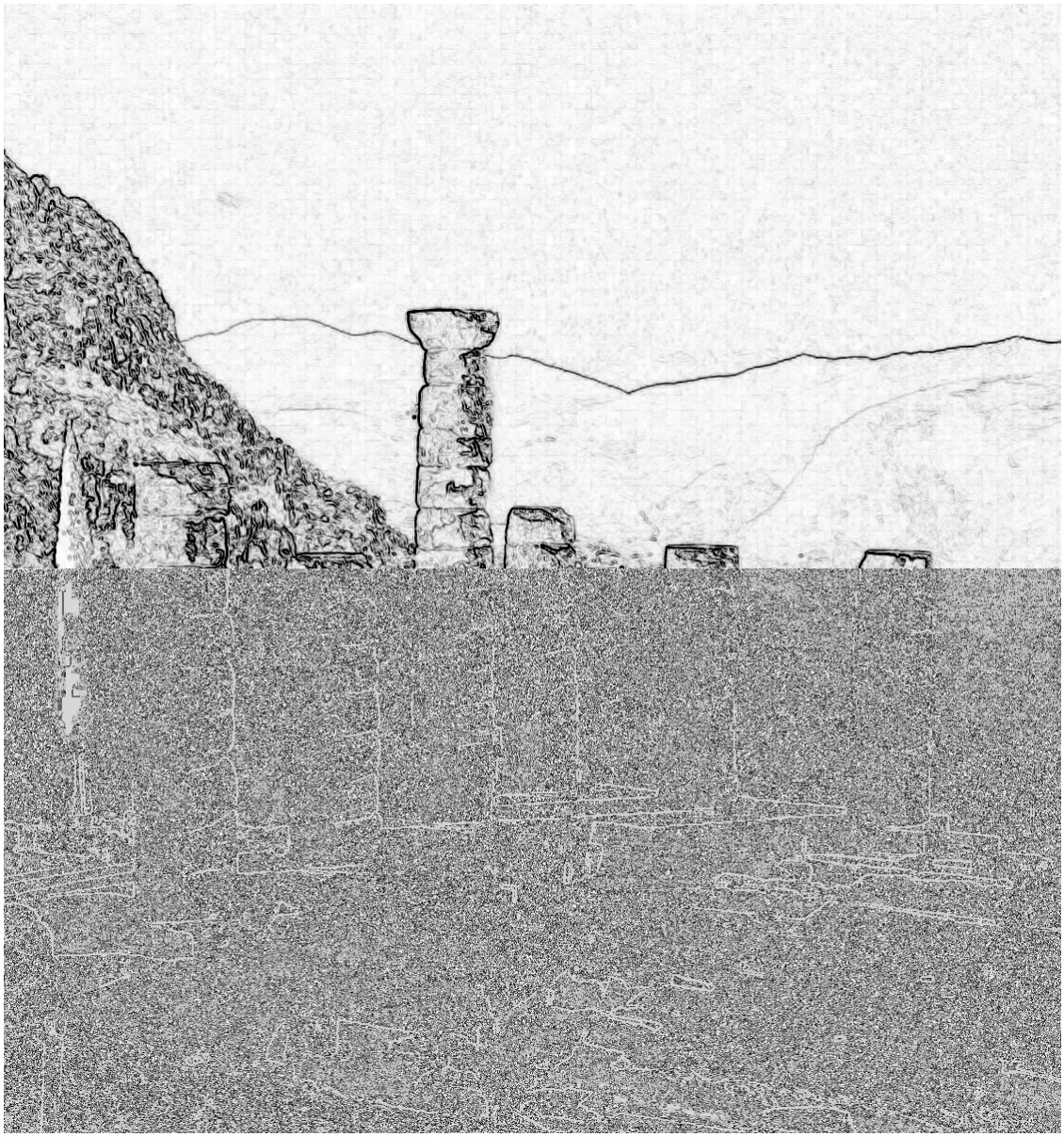
Modul Praktikum

PEMROGRAMAN PASCAL

Didasarkan pada Turbo Pascal 7.0



Pascal 2006 Team:

Kristian Trias Yulianto
Eko Agung Winarto
Adhis Mahaswi Dewi

Sekolah Tinggi Informatika dan Komputer Indonesia

2006

Winner vs. Looser

Winner is always a part of solutions

Looser is always a part of problems

Winner sees answer in every problem

Looser sees problem in every answer

Winner always has a program

Looser always has an excuse

Winner always says, “It’s difficult, but it’s possible.”

Looser always says, “It’s possible, but it’s difficult.”

Kata Pengantar

Pascal merupakan salah satu bahasa pemrograman yang terkenal dengan kekuatan strukturnya. Selain itu didukung pula dengan kemudahannya untuk digunakan, sehingga menjadi pilihan yang patut di-perhitungkan dalam dunia pemrograman. Kecepatan penjalanan program yang dihasilkannya juga menjadi salah satu daya saing mengapa Pascal terus berkembang hingga saat ini dengan berbagai bentuk, sampai terakhir pada bentuk OOP (*object oriented programming*) dan visual.

Modul praktikum Pascal I ini disusun sedemikian rupa sehingga mendekati kemudahan dalam penyerapan materi dan pemahaman serta penguasaan praktek pada praktikan. Selain itu juga telah disusun paralel dengan kelas teori, sehingga korelasi antara teori dan praktek akan lebih terintegrasi.

Saran dan kritik tetaplah diperlukan untuk meningkatkan mutu dari modul praktium ini sebagai penunjang belajar bahasa pemrograman, khususnya Pascal.

Malang, Agustus 2006

Penyusun

Daftar Isi

Praktikum

1

Struktur Dasar Bahasa Pascal

Tujuan:

- *Praktikan mampu menjelaskan tentang langkah-langkah pembuatan program hingga menghasilkan program yang bisa dieksekusi dari lingkungan MS-DOS secara langsung*
- *Praktikan mengerti pendeklarasian constanta, type, variable mampu menggunakannya dalam pembuatan program.*

Persiapan

Menjalankan program Pascal, mengikuti percobaan yang diberikan sesuai dengan urutannya. Mengerjakan Latihan-latihan yang diberikan.

Pekerjaan

Mengetikkan program sederhana dengan menggunakan *type*, *constanta*, dan *variable* seperti pada percobaan.

Teori

Sebuah program Pascal berisi kepala program, anak kalimat *uses* (tidak harus ada), serta blok pengumuman dan pernyataan. Kepala program menunjukkan nama program. Anak kalimat *uses* menggunakan unit-unit yang dipakai oleh program tersebut. Bagian terakhir merupakan blok yang berisi pengumuman san pernyataan yang akan dijalankan.

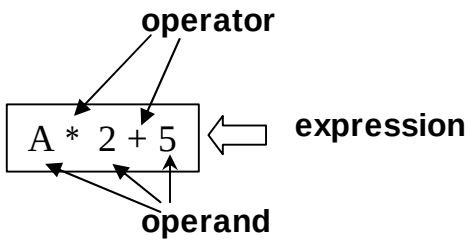
Setiap program Pascal mempunyai susunan sebagai berikut:

```
Program Nama_Program;
uses
     . . .  {Unit-unit yang dipakai} ;
label
     . . .  {label-label yang dipakai } ;
const
     . . .  {pengumuman tetapan-tetapan} ;
type
     . . .  { pengumuman tipe-tipe data };
```

```
var
     . . .  { pengumuman peubah-peubah };

procedure Nama_Prosedur;
begin
    . . .
end;

Function Nama_Fungsi;
begin
    . . .
end;

{  Program utama   }
begin
    . . .
end.
```

Unsur aturan penulisan dasar—disebut token—akan digabung membentuk ungkapan (*expressions*), pengumuman (*declarations*), dan pernyataan (*statements*). Pernyataan menggambarkan tindakan algoritma yang dapat dijalankan dalam program. Setiap ungkapan adalah sebuah aturan kesatuan aturan penulisan yang terdapat dalam pernyataan dan menunjukkan sebuah nilai. Setiap ungkapan terdiri dari *operator* dan *operand*.

Sedangkan pengumuman menegaskan sebuah pengenal (*identifier*) yang dapat digunakan pada ungkapan atau pernyataan, dan bila diperlukan memesan tempat pada pengingat untuk pengenal tersebut.



Percobaan

Salinlah program berikut ini :

```
program P0101;
   {  program untuk menampilkan data diri  }
uses Crt;
var
  nama, alamat,hobby: string;
     {  program utama  }
begin
  Writeln('   Program Penampil Data Diri ');
```

```
  Writeln('********************************');
  Write('Masukkan Nama   : ');
  Readln(nama);
  Write('Masukkan Alamat : ');
  Readln(alamat);
  Write('Masukkan Hobby  : ');
  Readln(hobby);
  Writeln;
  Writeln('Hallo ',nama,' !!! kamu tinggal di ',
          alamat, ' dan Hobbymu adalah ',hobby);
  Readln;
end.
```

Kompilasi program tersebut dengan menekan *Alt+F9* dan jalankan program tersebut dengan menekan *Ctrl+F9*, kemudian amati hasilnya bila dimasukkan masukan tertentu.

Sekarang simpan program tersebut dengan memilih menu *File* lalu pilih *Save*. Simpan dengan nama **Prak1_01.PAS**.

Kemudian coba lakukan kompilasi ke pengingat bantu. Hasil dari kompilasi ke pengingat bantu ini adalah sebuah berkas bernama **Prak1_01.EXE.**

Berikut ini akan diperkenalkan penggunaan konstanta dalam program. Salinlah program berikut ini :

```
program P0102;
{  program untuk Menghitung Luas  Lingkaran }
uses Crt;
const
  phi=3.14159;
var
  luas, jejari: Real;
begin
  Writeln('Program Menghitung Luas Lingkaran ');
  Writeln('****************************************');
  Write('Masukkan Jari-jari Lingkaran : ');
  Readln(jejari);
  luas := phi * sqr(jejari);
  Writeln('Luas Lingkaran adalah : ', round(luas));
  Readln;
end.
```

Kompilasi dan jalankan program tersebut, kemudian amati hasilnya bila dimasukkan masukan tertentu. Apa yang terjadi bila nilai jejari sama dengan nol? Apa pula yang terjadi jika nilai jejari diisi dengan huruf?

Untuk lebih memahami pendeklarasian dari *type, konstanta* dan *variabel* serta penggunaannya dalam pembuatan *statement*, salinlah program di bawah ini

```
program P0103;
```

```
{  program untuk Menampilkan Biodata Diri }
uses Crt;
type
  str10 = string[10];
const
  koma=',';
var
  nama1,nama2 : str10;
  alamat : string;
  usia ,thn_lhr,thn_skr: Integer;
begin
  Write('Masukkan Nama Depan  : ');
  Readln(nama1);
  Write('Masukkan Nama Belakang  : ');
  Readln(nama2);
  Write('Masukkan Alamat : ');
  Readln(alamat);
  Write('Masukkan Tahun Lahir : ');
  Readln(thn_lhr);
  Write('Masukkan Tahun Sekarang : ');
  Readln(thn_skr);
  usia := thn_skr - thn_lhr;
  Writeln;
  Writeln('=== BIODATA DIRI ===');
  Writeln(nama2,koma,nama1);
  Writeln(alamat);
  Writeln(usia,' tahun');
  Readln;
end.
```

Kompilasi dan jalankan program di atas. Amati keluaran yang dihasilkan dan pahami logikanya. Pada penghitungan usia jelas bahwa peubah yang berada pada ruas kiri suatu pernyataan pemberian akan berubah nilainya sesuai dengan nilai ungkapan yang berada di ruas kanannya.

Latihan

Latihan 1.1

Lengkapilah program **P0102.PAS** di atas untuk menghitung keliling lingkaran.

Latihan 1.2

Buatlah program untuk menentukan hasil penjumlahan atau pengurangan dari 2 bilangan bulat.

Latihan 1.3

Buatlah Program untuk menentukan hasil dari pembagian bulat dan sisa pembagian bulat dari 2 bilangan bulat.

Laporan

Laporan 1.1

Buatlah kesimpulan dari kegiatan praktikum 1 ini.

Laporan 1.2

Jelaskan perbedaan antara

a. Write dan *Writeln*

b. Read dan *Readln*

Laporan 1.3

Buatlah program yang menampilkan data mahasiswa berupa NRP, Nama, Nilai Tugas, Nilai UTS, Nilai UAS dan Total Nilai.

Praktikum

2

Struktur Kendali Aliran

Tujuan:

- *Bila diberikan kasus sederhana yang memerlukan alur terkondisi, mahasiswa mampu membuat program untuk memecahkan kasus tersebut dalam Bahasa Pascal.*

Persiapan

Menjalankan Program Turbo Pascal, mengikuti percobaan yang diberikan sesuai dengan urutannya. Mengerjakan latihan-latihan yang diberikan.

Pekerjaan

Mengetikkan *coding* dengan menggunakan struktur kendali *if* dan *case*, menjalankannya, dan menganalisa hasil.

Pengenalan Pernyataan Kendali (*Conditional Statement*)

Struktur kendali aliran adalah suatu bentuk/struktur yang memiliki peranan khusus untuk mengatur aliran urutan pengerjaan operasi atau beberapa operasi tertentu.

Pernyataan kendali terbagi menjadi dua, yaitu pernyataan *if* dan pernyataan *case*.

Pernyataan *If* (*If Statement*)

Pernyataan *if* (*if statement*) akan memeriksa suatu persyaratan dan menentukan apakah syarat tersebut benar atau salah, kemudian melakukan pekerjaan sesuai dengan nilai pernyataan tersebut.

Berikut adalah bentuk-bentuk dari pernyataan *if* yang sering digunakan :

1. *If* dengan satu pernyataan (*statement*)

```
If (kondisi) then pernyataan ;
```

2. *If* dengan dua atau lebih pernyataan (*statement*)

```
If (kondisi) then
begin
   pernyataan1 ;
   pernyataan2 ;
   ….
end;
```

3. *If* dan *else*

```
If (kondisi) then
begin
   pernyataan1 ;
   pernyataan2 ;
   ….
end
else
begin
   pernyataan1 ;
   pernyataan2 ;
   ….
end;
```

Dari bentuk bentuk pernyataan *if* di atas yang harus diperhatikan adalah untuk pernyataan *if* dan *else*, pernyataan-pernyataan setelah *then* tanpa menggunakan “;”. Dengan kata lain jika pernyataan setelah then hanya terdiri dari satu pernyataan saja makan pernyataan tersebut tanpa menggunakan “;”, namun jika pernyataan setelah then terdiri dari lebih dari satu pernyataan makan setelah *end* tanpa menggunakan “;”.

Program Sederhana dengan Pernyataan Kendali *If.*

Berikut contoh program sederhana untuk menghitung luas segitiga dan lingkaran. Yang mana perhitungan luas dua buah bangun tersebut digabung menjadi satu program, dan pemakai dapat memilih masalah mana yang akan dipecahkan.

Sebagaimana diketahui untuk menghitung luas segitiga adalah dengan :

$L=1/2 \cdot a \cdot t$

Dan luas lingkaran adalah dengan :

$L=? \cdot r. r$

Untuk lebih jelasnya salin program berikut :

```
program P0201;
var
```

```
  alas, tinggi, jejari, luas: Real;
  pilih: Integer;
begin
  Writeln('1. Segitiga   2. Lingkaran');
  Readln(pilih);
  if pilih = 1 then
  begin
    Readln(alas, tinggi);
    luas := 1 / 2 * alas * tinggi;
    Writeln('Luas Segitiga : ', luas);
  end
  else
  begin
    Readln(jejari);
    luas := pi * Sqr(jejari);
    Writeln('Luas Segitiga : ', luas);
  end;
  Readln;
end.
```

Jalankan program tersebut. Pada program tersebut disajikan dua macam pilihan kepada pemakai untuk menghitung luas segitiga atau luas lingkaran. Peubah pilih digunakan untuk pemakai guna menentukan pilihannya.

Pernyataan *Case* (*Case Statement*)

Pernyataan *case* (*case statement*) berisi ungkapan pemilih (*selector*) dan sederetan pernyataan yang masing-masing diawali dengan satu atau lebih tetapan (*case constant*) atau dengan kata kunci *else*.

Semua tetapan *case* harus khas dan tipe berurutan yang digunakan harus sesuai dengan tipe pemilih.

Sama halnya dengan pernyataan *if*, pada pernyataan *case* jika pernyataan yang akan dijalankan lebih dari satu pernyataan maka sebelum pernyataan-pernyataan tersebut harus diawali dengan *begin* dan diakhiri dengan *end*.

Berikut bentuk pernyataan *case* :

```
case <peubah> of
     tetapan1: pernyataan1;
        tetapan2: begin
     pernyataan1;
     pernyataan2;
     …
   end;
        tetapan3, tetapan4: pernyataan1;
        tetapan5 .. tetapan8: pernyataan1;
  else  pernyataan1;
```

```
  end;
```

Program Sederhana dengan Pernyataan Kendali *Case*.

Dengan contoh program yang sama untuk versi penyataan *case* dapat dilihat pada program di bawah ini :

```
program P0202;
var
  alas, tinggi, jejari, luas: Real;
  pilih: Integer;
begin
  Writeln('1. Segitiga   2. Lingkaran');
  Readln(pilih);
  case pilih of
    1: begin
         Readln(alas, tinggi);
         luas := 1 / 2 * alas * tinggi;
         Writeln('Luas Segitiga : ', luas);
       End;
    2: begin
      Readln(jejari);
         luas := pi * Sqr(jejari);
         Writeln('Luas Segitiga : ', luas);
       end;
  end;
  Readln;
end.
```

Dari bentuk-bentuk struktur kendali yang telah dijelaskan di atas dapat terjadi dimana struktur kendali tertentu berada pada struktur kendali yang lain, yang dikenal dengan struktur kendali bersarang. Misalnya didalam pernyataan *if* terdapat pernyataan *if* lagi atau dalam pernyataan *case* terdapat pernyataan *if* dan seterusnya.

Untuk lebih jelasnya salin program berikut :

```
program P0203;
var
  nama, alamat: string;
  nrp: string[8];
  keterangan: string[2];
  grade, pilih: Char;
  nilai: Integer;
begin
  Writeln('1. Memasukkan data mahasiswa  2. Keluar');
  Write('Pilihan Anda : ');
  Readln(pilih);
  case pilih of
    '1': begin
```

```
            Write('Masukkan Nama Anda : ');
            Readln(nama);
            Write('Masukkan NRP Anda : ');
            Readln(nrp);
            Write('Masukkan Alamat Anda : ');
            Readln(alamat);
            Write('Masukkan Nilai Anda : ');
            Readln(nilai);
            {menentukan program studi}
            if (copy(nrp, 3, 2)='11') then
              keterangan := 'S1'
            else
              if (copy(nrp, 3, 2)='31') then
                keterangan := 'D3';
            {menentukan grade}
            if (nilai >= 80) then
              grade := 'A'
            else
              if (nilai >= 70) then
                grade := 'B'
              else
                if (nilai >= 60) then
                  grade := 'C'
                else
                  grade := 'D';
            Writeln('Data mahasiswa yang diinputkan');
            Writeln('Nama : ',nama);
            Writeln('NRP : ',nrp);
            Writeln('Program Studi : ',keterangan);
            Writeln('Alamat : ',alamat);
            Writeln('Nilai : ',nilai);
            Writeln('Grade : ',grade);
          end;
      '2': begin
             Writeln('Anda mengakhiri program !');
           Exit;
          end;
      else
      begin
        Writeln('Pilihan Anda salah !');
        Exit;
      end;
    end;
    Readln;
end.
```

Latihan

Latihan 2.1

Ubah program **P0203**, tambahkan sehingga :

- nilai yang diinputkan ada tiga macam yaitu : nilai Tugas, nilai UTS dan Nilai UAS.
- Untuk perhitungan grade didapatkan dari nilai akhir dengan perhitungan :
- Nilai akhir = (2 * nilai UAS + nilai Tugas + nilai UTS) / 4
- Data mahasiswa yang ditampilkan dalam huruf besar, meskipun dalam pengisian awal menggunakan huruf kecil.
- Tampilkan juga apakah mahasiswa tersebut LULUS atau TIDAK LULUS dengan ketentuan jika grade dari mahasiswa "D" maka ia TIDAK LULUS selain itu maka ia LULUS.

Latihan 2.2

Tambahkan dari program no. 1 *username* dan *password*, sehingga ketika pengguna mengisi data *username* dan *password*, dan jika tidak sesuai dengan yang ada pada listing maka pemakai tidak dapat melakukan proses selanjutnya, dan munculkan teks "username atau password yang Anda inputkan salah !!"

Latihan 2.3

Buatlah program untuk menentukan apakah nilai yang diinputkan genap atau ganjil.

Laporan

Laporan 2.1

Buat kesimpulan untuk praktikum hari ini.

Laporan 2.2

Buatlah program untuk menentukan bilangan terbesar dan bilangan terkecil dari tiga buah masukan angka.

Laporan 2.3

Buatlah program kalkulator sederhana dengan memberikan pilihan bagi pengguna. Operasi-operasinya meliputi : penjumlahan, pengurangan, perkalian, pembagian, sisa hasil bagi dan pangkat.

Praktikum

3

Struktur Perulangan

Tujuan:

- *Bila diberikan kasus sederhana yang memerlukan alur berulang, mahasiswa akan mampu membuat program untuk memecahkan kasus tersebut dalam Bahasa Pascal, untuk setiap perulangan.*

Persiapan

Menjalankan Program Turbo Pascal, mengikuti percobaan yang diberikan sesuai dengan urutannya. Mengerjakan latihan-latihan yang diberikan.

Pekerjaan

Mengetikkan *coding* dengan menggunakan struktur perulangan *repeat, while,* dan *for*, menjalankannya, dan menganalisa hasil.

Pengenalan Pernyataan Perulangan (*Loop Statement*)

Pernyataan Perulangan memiliki tiga pernyataan, *repeat*, *while*, dan *for*. Pernyataan perulangan dipakai untuk melakukan proses berulang terhadap pernyataan sederhana atau pernyataan terstruktur.

Pernyataan *Repeat* (*Repeat Statements*)

Pernyataan *repeat* (*repeat statements*) digunakan untuk melakukan perulangan terhadap suatu pernyataan, dimana proses pemeriksaan syaratnya berada pada akhir pernyataan *repeat* tersebut. Pernyataan-pernyataan yang ada pada *repeat* akan dijalankan (diulang terus) sampai kondisi yang diseleksi di *until* tidak terpenuhi**.**

Pernyataan *While* (*While Statements*)

Pernyataan *while* (*while statements*) hampir sama dengan pernyataan *repeat*, dengan sedikit perbedaan bahwa pernyataan *while* melakukan pengujian syarat pada awal proses berulang (pernyataan *repeat* melakukannya di akhir proses).

Pengujian awal digunakan untuk agar program dapat menyeleksi kondisi, sehingga program dapat menentukan tindakan apa yang harus dikerjakan, tergantung dari kondisi yang diseleksi tersebut. Pada pernyataan-pernyataan yang ada pada *while* tidak akan dijalankan jika kondisi tidak terpenuhi.

Pernyataan *For* (*For Statements*)

Pernyataan *for* (*for statements*) juga digunakan untuk melakukan proses perulangan. Hanya saja proses perulangan pada pernyataan for langsung dikendalikan oleh suatu peubah yang disebut peubah kendali (*control variables*) yang harus bertipe berurutan. Jadi pada pernyataan for pada dasar sudah diketahui jumlah perulangannya. Perulangan dengan pernyataan *for* dapat berupa perulangan positif (*'to'*) dan perulangan negatif (*'downto'*).

Jika pernyataan yang akan mengalami perulangan lebih dari satu pernyataan, maka harus diawali dengan *begin* dan diakhiri dengan *end*;.

Pengendalian Perulangan

Adakalanya pemrogram menginginkan suatu keadaan dimana dalam proses perulangan, perulangan dapat dihentikan atau dilanjutkan tanpa selalu harus melalui pengujian syaratnya. Untuk keperluan itu, Pascal telah menyediakan dua prosedur bawaan, yaitu *Break* dan *Continue*.

Untuk menghentikan proses perulangan ketika proses belum mencapai pengujian syaratnya, dapat digunakan prosedur *Break*. Sedangkan untuk meneruskan proses perulangan ke proses selanjutnya ketika proses belum mencapai pengujian syaratnya, dapat digunakan prosedur *Continue*.

Program Sederhana dengan Pernyataan Perulangan

Berikut ini akan disajikan pemecahan masalah menggunakan struktur perulangan.

Contoh yang disajikan adalah membalik sebuah untai karakter yang dimasukkan pemakai dan menampilkan hasilnya. Algoritma yang

digunakan adalah dengan memasukkan satu demi satu karakter ke dalam suatu peubah hasil dengan urutan dari depan ke belakang, yang diambil dari peubah masukan dengan urutan yang berlawanan. Peubah hasil adalah yang akan ditampilkan.

Salin program berikut ini :

```
program P0301;
var
  I, Panjang: Integer;
  Kata, Balik: string;
begin
  Writeln('Balik Kalimat');
  Writeln('=============');
  Write('Masukkan kalimat : ');
  Readln(Kata);
  Panjang := Length(Kata);
  Balik := '';
  for I := Panjang downto 1 do
    Balik := Balik + Kata[I];
  Writeln('Hasil pembalikannya : ', Balik);
  Readln;
end.
```

Berikut contoh program untuk menampilkan bilangan genap dari 2 sampai dengan 100. Salin program berikut :

```
program P0302;
var
  i: Integer;
begin
  Writeln('Bilangan genap');
  i := 1;
  repeat
    Inc(i);
    if i mod 2 = 0 then Write(i:4);
  until (i=100);
  Readln;
end.
```

Sama halnya dengan struktur kendali yang telah dibahas sebelumnya, pada struktur perulangan dapat terjadi kasus dimana di dalam struktur perulangan tertentu terdapat struktur perulangan yang lain dan seterusnya. Program berikut akan menerapkan struktur perulangan dengan menggunakan *while*. Salin program berikut :

```
program P0303;
var
  i, j: Integer;
  M: Char;
begin
```

```
  Write('Masukkan sembarang karakter : ');
  Readln(M);
  i := 1;
  while i <= 10 do
  begin
    j := 1;
    while j <= i do
    begin
      Write(M);
      Inc(j);
    end;
    Writeln;
    Inc(i);
  end;
  Readln;
end.
```

Salin program berikut :

```
program P0304;
uses Crt;
var
  Hasil: Real;
  I, pilih, A, X, N: Integer;
  Status: Boolean;
begin
  Repeat
    Clrscr;
    Writeln('1. Pemangkatan  2. Faktorial  3. Keluar');
    Write('Masukkan Pilihan Anda : ');
    Readln(pilih);
    case pilih of
      1 : begin
         Write('Masukkan Sembarang angka : ');
          Readln(X);
          Write('Akan dipangkatkan berapa : ');
          Readln(A);
          Hasil := 1;
          for I := 1 to A do Hasil := X * Hasil;
            Writeln('Jadi ', X,' dipangkatkan ', A,' : ',
Hasil:5);
          Status := False;
        end;
   2 : begin
          Write('Masukkan Sembarang angka : ');
          Readln(N);
          if N <= 1 then Hasil := 1
          else
          begin
            Hasil := 1;
            for I := 2 to N do Hasil := Hasil * I;
```

```
            end;
              Writeln('Jadi factorial ', N,' (', N,'!) : ',
Hasil:5);
            Status := False;
          end;
     3 : Status := True;
     else
       begin
   Writeln('Pilihan Anda salah !');
   Status := True;
end;
     end;
     Readln;
   until Status;
end.
```

Latihan

Latihan 3.1

Buat program untuk menampilkan deret fibbonaci 1, 1, 2, 3, 5, 8, .. !

Latihan 3.2

Buat program untuk menampilkan deret kuadrat 1, 4, 9, 16, .. !

Latihan 3.3

Buat program untuk menampilkan angka sebagai berikut :

```
1
22
333
4444
55555
```

Laporan

Laporan 3.1

Buat kesimpulan untuk praktikum hari ini.

Laporan 3.2

Buat program untuk menampilkan bilangan Prima antara 1 sampai 100.

Laporan 3.3

Buatkan program untuk menampilkan segitiga pascal !

```
1
11
121
1331
14641
```

Laporan 3.4

Buat program untuk menghitung gaji pegawai dengan ketentuan sebagai berikut :

- Terdapat dua buah golongan yaitu : A dengan gaji pokok 500000 dan B dengan gaji pokok 1000000
- Pegawai akan mendapatkan tunjangan jika sudah menikah sebesar 10% dari gaji pokok. Untuk status tidak menikah, janda, dan duda tunjangan=0.
- Jika pemakai memasukkan inputan baik itu untuk status dan golongan tidak sesuai dengan yang diminta ulang terus sampai yang diinputkan benar.
- Tampilkan nama, alamat, status, gaji pokok dan total gajinya

Praktikum

4

Prosedur dan Fungsi

Tujuan:

- *Praktikan mengerti tentang prosedur dengan parameter dan prosedur tanpa parameter.*
- *Praktikan mengerti tentang fungsi dengan parameter dan fungsi tanpa parameter.*
- *Praktikan mengerti tentang penggunaan variable global dan variable lokal.*
- *Praktikan mampu membuat program yang berisi prosedur dan fungsi*

Persiapan

Menjalankan program Pascal, mengikuti percobaan yang diberikan sesuai dengan urutannya. Mengerjakan Latihan-latihan yang diberikan.

Pekerjaan

Mengetikkan program sederhana dengan menggunakan prosedur dan fungsi seperti pada percobaan.

TEORI

Program Pascal akan menjadi mudah dibuat jika ditulis dalam bentuk modul-modul. Sistem modul ini memiliki beberapa keuntungan, diantaranya :

- untuk langkah-langkah yang sering dilakukan (bukan perulangan), akan terhindar dari pembuatan pernyataan-pernyataan yang sama.
- Suatu modul program hanya sekali ditetapkan dan dapat dipanggil dari beberapa tempat dalam program. Sekumpulan data yang berbeda juga dapat diproses setiap kali modul tersebut dijalankan.

Dengan menggunakan sistem modul ini panjang program akan dapat dikurangi. Dalam pascal terdapat dua tipe modul, yaitu prosedur dan fungsi (*procedures and functions*). Dua tipe modul program ini sama,

hanya cara pemanggilannya berbeda, dan memberikan informasi dengan cara yang berbeda. Pada praktikum-praktikum sebelumnya telah digunakan beberapa *prosedur bawaan*, misalya *Write, Writeln, Read* dan *Readln, serta Fungsi bawaan*, misalnya, *Odd, Sqr, dll.*

Prosedur (*Procedures*) memiliki struktur yang sama dengan struktur program, yaitu terdiri dari nama prosedur, pengumuman-pengumuman atau deklarasi (kecuali pengumuman *uses*), dan bagian utama (pernyataan) dari prosedur tersebut. Di dalam prosedur juga dimungkinkan terdapat prosedur (atau fungsi) lainnya, sehingga dapat disebut dengan prosedur bersarang (*nested procedures*).

PROCEDURE *nama(daftar_parameter)*
Bagian deklarasi / pengumuman;
Bagian pernyataan;

Fungsi (*functions*) hampir sama dengan prosedur, dengan sedikit perbedaan bahwa nama fungsi sekaligus berfungsi sebagai suatu ungkapan (pada blok pemanggil fungsi tersebut). Sehingga setiap fungsi harus diumumkan tipe datanya.

FUNCTION *nama_fungsi(daftar_parameter)*: *tipe;*
Bagian deklarasi / pengumuman;
Bagian pernyataan;

Percobaan

Perhatikan dan salinlah program di bawah ini :

```
program P0401;
  {  Contoh Program pembuatan Prosedur  }
uses Crt;
var
  A,B: Integer;        {  variable global  }

      { Prosedur mencari nilai maximum }
procedure Maximum;
var
  max : Integer;        {  variable lokal  }
begin
  if A > B then
    max := A
  else
    max := B;
```

```
  Writeln('Angka Terbesar adalah : ',max);
end;      {  Akhir Prosedur  }

    { Program Utama }
begin
  ClrScr;
  Write('Masukkan angka Pertama : ');
  Readln(A);
  Write('Masukkan angka Kedua : ');
  Readln(B);
  while A <> 0 do
  begin
    Maximum;              {  pemanggilan prosedur  }
    Readln(A,B);
  end;
end.
```

Kompilasi dan jalankan program tersebut. Prosedur Maximum berfungsi untuk menentukan nilai maksimum dari 2 buah bilangan bulat. Program tersebut merupakan contoh penerapan pembuatan prosedur tanpa parameter. Amati nilai yang dihasilkan jika dimasukkan data tertentu.

Untuk lebih memahami tentang fungsi, salinlah program di bawah ini :

```
program P0402;
   {  Contoh Program pembuatan Fungsi  }
uses crt;
var
  A,B : integer;            {  variable global  }

      { Fungsi menentukan nilai Faktorial }
function Faktorial(N: Integer) : LongInt;
var
  faktor, Hasil : integer;    {  variable lokal  }
begin
  if  N <=1  then
    faktorial :=1             {Pengembalian nilai }
  else
  begin
    Hasil :=1;
    for faktor := 2 to N do Hasil := Hasil * Faktor;
    Faktorial := Hasil;        { Pengembalian nilai }
end;

      {  Program Utama   }
begin
  ClrScr;
  Write('Masukkan sembarang bilangan : ');
  Readln(A);
```

```
  B:= faktorial(A);            { pemanggilan fungsi }
  Writeln('Faktorial ',A,' : ',B);
  Readln;
end.
```

Kompilasi dan jalankan Program tersebut. Program tersebut menerapkan penggunaan fungsi. Fungsi Faktorial hanya memakai parameter nilai, dan karena sebuah fungsi, maka memiliki nilai kembalian. Sehingga pemanggilan nama fungsi tersebut dapat berperan seperti ungkapan.

Untuk lebih memahami prosedur, salinlah program di bawah ini :

```
program P0403;
uses Crt;

var
  A: integer;

Procedure Flip(N:Integer); forward;

Procedure Flop(N:Integer);
begin
  Writeln('Flop');
  If N > 0 then Flip(N - 1);
end;

Procedure Flip(N:Integer);
begin
  Writeln('Flip');
  If N > 0 then Flop(N - 1);
end;

begin
  ClrScr;
  Write('Masukkan Banyak kata : ');
  Readln(A);
  A := A - 1;
  Flip(A);
  Readln;
end.
```

Kompilasi dan jalankan program tersebut lalu amatilah hasilnya. Prosedur Flip dan Prosedur Flop di atas menampilkan kata ‘Flip’ dan ‘Flop’ secara bergantian sesuai dengan inputan banyak kata.

Deklarasi *forward* digunakan untuk mendeklarasikan Prosedur Flip di atas prosedur Flop, sehingga pada saat prosedur Flop dijalankan dan prosedur Flip dipanggil, prosedur Flip dapat dikenali.

Latihan

Latihan 4.1

Rubahlah Prosedur *Maximum* pada program **P0401.PAS** menjadi prosedur dengan parameter.

Latihan 4.2

Buatlah Program yang terdiri dari 2 buah prosedur :

- Prosedur pertama digunakan untuk memasukkan 2 buah bilangan sebagai inputan
- Prosedur kedua menampilkan hasil dari bilangan tersebut setelah nilainya ditukar.

Latihan 4.3

Buatlah program yang berisi sebuah fungsi yang dapat menampilkan bilangan ganjil antara 1 sampai dengan 15.

Laporan

Laporan 4.1

Jelaskan perbedaan antara

a. *Prosedur tanpa parameter* dan *prosedur dengan parameter*

b. *Fungsi tanpa parameter* dan *fungsi dengan parameter*

Laporan 4.2

Apa Yang dimaksud dengan

a. Variable Global

b. Variable Lokal

Laporan 4.3

Buatlah *Fungsi* untuk menentukan nilai *Pangkat* dari suatu bilangan bulat. Inputannya berupa suatu bilangan yang akan dipangkatkan dan bilangan pangkatnya.

Misalnya : 5 pangkat 2 = 25.

Laporan 4.4

Buatlah sebuah program untuk menghitung :

- Luas Persegi Panjang, bila diinputkan panjang dan lebarnya.
- Luas Lingkaran, bila diinputkan jari-jarinya.
- Luas Segitiga, bila diinputkan alas dan tingginya.

Masing-masing perhitungan di atas harus dimasukkan fungsi atau prosedur dan memakai parameter. Begitu program dijalankan, pemakai harus diberi pilihan untuk menghitung salah satu dari perhitungan di atas atau mengakhiri program.

Praktikum

5

Array

Tujuan:

- *Praktikan mengerti struktur data dengan menggunakan banyak variable dengan tipe data yang sama.*

Persiapan

Menjalankan program Pascal, mengikuti percobaan yang diberikan sesuai dengan urutannya. Mengerjakan Latihan-latihan yang diberikan dan pelajari bab Procedure dan Function.

Pekerjaan

Mengetikkan program sederhana dengan menggunakan *type*, *constanta* dan *variable* yang dikombinasikan dengan *array* seperti pada percobaan.

TEORI

Array merupakan struktur data yang *statis*, yaitu jumlah elemen yang ada harus ditentukan terlebih dahulu dan tak bisa di ubah saat program berjalan. Untuk menyatakan *array* dalam PASCAL kita harus terlebih dahulu:

- Mendefinisikan jumlah elemen *array*,
- Mendefinisikan tipe data dari elemen *array*

Contoh :

```
const
  N=10;
type
  A= array [1..N] of integer;
```

Array Satu Dimensi

Pendefinisian *array* secara umum adalah sebagai berikut: jika kita ingin membuat beberapa *array* dengan tipe/jenis yang sama, kita lebih baik jika mendeklarasikan dengan *type* selanjutnya dengan deklarasi *var*.

```
type
  nama_array = ARRAY[bawah..atas] of tipe_data;
var
  variabel_array : nama_array;
```

atau dengan menggunakan *statement var* :

```
var
  variabel_array : ARRAY[bawah..atas] of tipe_data;
```

Penjelasan: Bawah dan Atas menyatakan batas untuk *array*. tipe_data adalah merupakan tipe *variabel* yang dipunyai *array* (mis. *Integer*, *char*, *real*, dsb)

Array Multidimensi

Dalam *array multidimensi* terdiri atas baris (*row*) dan kolom (*column*). Index pertama adalah baris dan yang kedua adalah kolom .

```
Type nama_array =ARRAY[bawah..atas, bawah..atas] of tipe_data;
var variabel_array : nama_array;
```

atau dengan menggunakan *statement var* :

```
var variabel_array : ARRAY[bawah..atas, bawah..atas] of
tipe_data;
```

Percobaan

Salinlah program berikut ini :

```
program P0501;
  {Program Array menggunakan 1 dimensi}
uses Crt;
var
  a: array[1..10] of byte;{maksimum jumlah elemen=10}
begin
  a[1]:=10;
  a[2]:=15;
  a[3]:=a[1]+a[2];
  Writeln(a[1]);
  Writeln(a[2]);
```

```
  Writeln(a[3]);
end.
```

Kompilasi program tersebut dengan menekan *Alt+F9* dan jalankan program tersebut dengan menekan *Ctrl+F9*, kemudian amati hasilnya bila dimasukkan masukan tertentu.

Sekarang simpan program tersebut dengan memilih menu *File* lalu pilih *Save*. Simpan dengan nama **Prak5_01.PAS**.

Berikut ini akan diperkenalkan *array* 1 dimensi dengan menggunakan *for..do*. Salinlah program berikut ini :

```
program P0502;
uses crt;
const
  N=10;
type
  int_array= ARRAY [1..N] of integer;
var
  bil : int_array;
  indeks : integer;
begin
  writeln('masukkan sepuluh bilangan integer.');
  for indeks := 1 to 10 do
  begin
   readln(bil[indeks]);
  end;
  writeln('Isi dari array ini adalah');
  for indeks := 1 to 10 do
  begin
   writeln('bil[', indeks:2,'] adalah ',bil[indeks] );
  end
end.
```

Kompilasi dan jalankan program tersebut, kemudian amati hasilnya bila dimasukkan 10 nilai tertentu.

Untuk lebih memahami pendeklarasian dari *type*, *konstanta* dan *variabel* dalam *array* lebih lanjut serta tingkatan *array* yang lebih banyak lagi, salinlah program *array multidimensi* di bawah ini

```
program P0503;
uses crt;
const
  kolom = 3;
  baris = 3;
type
  matriks = ARRAY [1..baris, 1..kolom] of integer;
var
  AKU: matriks;
```

```
procedure ISI_MATRIK(m,n:integer);
var
  i,j: integer;
begin
  for i:=1 to m do
  begin
    for j:=1 to n do
    begin
      read(AKU[i,j]);
    end;
    readln ;
  end;
end;

procedure TULIS_MATRIK(m,n:integer);
  var i,j: integer;
begin
  for i:=1 to m do
  begin
    for j:=1 to n do
    begin
      write(AKU[i,j]:6);
    end;
    writeln ;
  end;
end;

begin
  clrscr;
  isi_matrik(kolom,baris);
  tulis_matrik(kolom,baris);
end.
```

Kompilasi dan jalankan program di atas. Amati hasil dari program tersebut, program diatas digabung dengan 2 *procedure*. *Procedure* pertama untuk mengisi nilai matrik, dan *procedure* yang kedua untuk menulis hasil dari nilai matrik yang sudah diisikan.

Latihan

Latihan 5.1

Tambahkan procedure untuk menghitung nilai maksimal

Latihan 5.2

Tambahkan procedure untuk menghitung nilai minimum

Laporan

Laporan 5.1

Buatlah kesimpulan dari kegiatan praktikum 1 ini.

Laporan 5.2

Jelaskan perbedaan antara *array* 1 *dimensi* dengan *array multidimensi*.

Laporan 5.3

Buatlah program penjumlahan matrik.

Praktikum

6

Record

Tujuan:

- *Praktikan mengerti struktur data dengan menggunakan record untuk menyimpan data.*

Persiapan

Menjalankan program Pascal, mengikuti percobaan yang diberikan sesuai dengan urutannya. Mengerjakan Latihan-latihan yang diberikan dan pelajari bab *Procedure* dan *Function*. Dan pahami array di bab sebelumnya.

Pekerjaan

Mengetikkan program sederhana dengan menggunakan *type, constanta* dan *variable* yang dikombinasikan dengan *array* ditambahkan dengan *record* seperti pada percobaan.

TEORI

Sebuah *record* rekaman disusun oleh beberapa *field*. Tiap *field* berisi data dari tipe dasar / bentukan tertentu. *Record* mempunyai kelebihan untuk menyimpan suatu sekumpulan elemen data yang berbeda-beda tipenya (di banding *array*).

Cara pendeklarasian dari *record* adalah sbb:

- Mendefinisikan tipe dari *record* (jumlah *field*, jenis tipe data yang dipakai),
- Mendefinisikan variabel untuk dilakukan operasi.

Syntax

```
type
  nama_record = record
    identifier_1 : tipe_data_1;
    :
    :
```

```
      identifier_n : tipe_data_n;
    end;
  var
    variabel : nama_record;
```

Contoh :

```
  type
    Data_mahasiswa = record
      Nama : string;
      Usia : integer;
      Kota : string;
      Kodepos : integer;
    end;
  Var
    x: Data_mahasiswa;
```

Percobaan

Salinlah program berikut ini :

Cara mengacu pada tiap *field* pada *record* pada contoh di atas adalah sbb:

```
x.Nama
x.Usia
x.Kota
x.Kodepos
```

```
program P0601;
type
  tanggal = record
  bulan, hari, tahun: integer;
  end;
var
  waktu: tanggal;
begin
  waktu.hari :=25;
  waktu.bulan:=09;
  waktu.tahun:= 1983;
  Writeln('hari ini adalah
',waktu.hari,':',waktu.bulan,':',waktu.tahun)
end.
```

Berikut ini akan diperkenalkan record dengan menggunakan array. Salinlah program berikut ini :

```
program P0602;
const
```

```
  N = 2;
type
  data = record
  nrp : string[8];
  nama : string[20];
  end;
var
  mhs : array [1..N] of data;
begin
  mhs[1].nrp := '00112398';
  mhs[1].nama := 'Eko Agung W';
  mhs[2].nrp := '03113551';
  mhs[2].nama := 'Ni Luh';
  Writeln(mhs[1].nrp);
  Writeln(mhs[1].nama);
  Writeln(mhs[2].nrp);
  Writeln(mhs[2].nama);
end.
```

Program diatas telah menggunakan *record* dikombinasi dengan *array*, bertujuan untuk mengisi data lebih dari satu. Kemudian salinlah program yang dibawah ini yang sudah dikombinasikan dengan perulangan untuk mempermudahkan menginput dan menampilkan data.

```
program P0603;
const
  N = 2;
type
  tmhs = record
  nim : string[11];
  nama : string[30];
  alamat : string;
end;
var
  datamhs: array[1..N] of tmhs;
  i, j: integer;
begin
  for i:= 1 to N do
  begin
    with datamhs[i] do
    begin
      Write('NIM    : ');
      Readln(nim);
      Write('NAMA   : ');
      Readln(nama);
      Write('ALAMAT : ');
      Readln(alamat);
    end;
  end;
  for j:= 1 to N do
```

```
    begin
      Writeln('NIM    : ',datamhs[j].nim);
      Writeln('NAMA   : ',datamhs[j].nama);
      Writeln('ALAMAT : ',datamhs[j].alamat);
    end;
    readln;
  end.
```

Kompilasi dan jalankan program di atas. Amati hasil dari program tersebut. Dengan dikombinasikan perulangan program tersebut akan lebih baik lagi. Biasanya perulangan disini untuk pencarian data.

Latihan

Latihan 6.1

Buatlah 2 *procedure* untuk menginput data dan menampilkan data pada program **P0603** diatas.

Laporan

Laporan 6.1

Buatlah kesimpulan dari kegiatan praktikum 6 ini.

Laporan 6.1

Jelaskan tentang *field* dan *record*.

Laporan 6.1

Buatlah program penyusunan data dengan teknik *bubble sort*.

Praktikum

7

Unit

Tujuan:

- *Bila terdapat prosedur, fungsi, tipe data, peubah, atau tetapan yang dipakai pada lebih dari satu program, mahasiswa akan mampu memilih, menuliskan ke dalam unit, mengkompilasi unit, serta menggunakan unit tersebut..*

Persiapan

Menjalankan Program Turbo Pascal, mengikuti percobaan yang diberikan sesuai dengan urutannya. Mengerjakan latihan-latihan yang diberikan.

Pekerjaan

Mengetikkan *coding*, menjalankannya, dan menganalisa hasil.

Pengenalan Unit

Unit adalah sebuah penerapan pemrograman *modularitas* pada Turbo Pascal. Setiap *unit* mirip seperti program Pascal yang terpisah, dan merupakan kumpulan tetapan, tipe data, peubah, prosedur dan fungsi.

Dengan memakai unit-unit yang diperlukan saja, maka akan sangat menghemat pengingat pada waktu program itu dijalankan.

Secara singkat dapat dikatakan bahwa unit merupakan suatu kepustakaan pengumuman (*declaration library*) yang dapat digunakan oleh program (atau unit yang lain) yang menggunakannya.

Turbo Pascal 7.0 menyediakan delapan unit bawaan yang dapat langsung digunakan. Enam diantaranya adalah unit *System*, *Overlay*, *Graph*, *DOS*, *CRT*, dan *Printer* dapat digunakan secara langsung. Dua yang lainnya, *Turbo3* dan *Graph3* disediakan untuk kesesuaian program dengan Turbo Pascal versi 3. Semua unit bawaan tersebut telah dijadikan satu di dalam berkas **TURBO.TPL**-kecuali *Graph*, *Graph3* dan *Turbo3*-

dan dapat digunakan secara langsung tanpa membutuhkan berkas unitnya.

Pembuatan Unit Sederhana

Berikut ini akan disajikan contoh-contoh pembuatan unit-unit sederhana, yang nantinya akan dapat digunakan oleh program-program lainnya.

Salin berikut ini :

```
unit U0701;

interface
function CekSandi(Kata, Password: string): Boolean;

implementation
function CekSandi(Kata, Password: string): Boolean;
begin
  if Kata=Password then
    CekSandi := True
  Else
    CekSandi := False;
end;

end.
```

Kemudian simpan program tersebut di atas dengan nama yang sama dengan nama unitnya. Untuk pemanggilan unit di ketikkan setelah kata kunci *Uses*.

Kompilasi unit tersebut di atas (ingat, selalu kompilasi unit ke pengingat tambahan (*compile to disk*)). Kemudian setelah dikompilasi, salin program berikut ini :

```
program P0701;
uses
  Crt, U0701;
var
  Kata_sandi : string;
begin
  Clrscr;
  repeat
    Write('Masukkan Password Anda !! : ');
    Readln(Kata_sandi);
  until CekSandi (Kata_sandi, 'praktikan');
  Writeln;
  Write('Password Anda Benar !!');
  Readln;
end.
```

Jalankan program tersebut dan amati hasilnya. Program tersebut menggunakan unit **U0701** dan unit **U0701** itu harus sudah terkompilasi pada berkas. Bila dijalankan, pernyataan-pernyataan antara *repeat* dengan *until* pada program tersebut akan diulang terus hingga kata sandi yang diketikkan adalah 'praktikan' dan CekSandi bernilai *True*.

Latihan

Latihan 7.1

Buatlah sebuah unit yang mana dalam unit tersebut terdapat prosedur atau fungsi untuk merubah satuan suhu tertentu (*Celcius*) ke satuan yang lainnya contohnya ke satuan *Kelvin*, *Fahrenheit*, dan *Reamur*. Kemudian buatlah suatu program sederhana yang mana menggunakan unit yang telah dibuat tersebut, ketika program dijalankan dengan memasukkan sebuah masukan berupa suhu dalam satuan *Celcius* maka akan ditampilkan suhu-suhu dengan satuan yang lainnya.

Adapun Rumusnya adalah sebagai berikut :

- **t °C = 4/5 t °R = (9/5t + 32) °F = (t + 273) °K**

Latihan 7.1

Buatlah unit untuk menghitung nilai *sinus*, *kosinus*, dan *tangen*, Ingat bahwa fungsi `sin` dan `cos` menggunakan parameter bersatuan *radian*. Buatlah pula sebuah program yang nantinya menggunakan unit tersebut, ketika program dijalankan dengan memasukkan sebuah masukan dalam satuan derajat, secara otomatis program merubah masukan tersebut ke dalam satuan *radian* dan menampilkan nilai *sinus*, *kosinus*, dan tangen dari masukan tersebut.

Adapun **tan x = sin x / cos x**

Laporan

Laporan 7.1

Buat kesimpulan untuk praktikum hari ini.

Laporan 7.1

Buatlah unit yang di dalamnya terdapat *function* untuk menghitung perhitungan matematik, yang meliputi penjumlahan, pengurangan, perkalian, pembagian, sisa hasil bagi, pemangkatan, factorial, sinus, cosinus, dan tangent.

Praktikum

8

File

Tujuan:

- *Praktikan mengerti struktur data dengan menggunakan file (berkas).*

Persiapan

Menjalankan program Pascal, mengikuti percobaan yang diberikan sesuai dengan urutannya. Mengerjakan Pahami array dan record di bab sebelumnya.

Pekerjaan

Mengetikkan program sederhana dengan menggunakan type, constanta dan variable yang dikombinasikan dengan array ditambahkan dengan record seperti pada percobaan.

Teori

File Teks disusun sebagai runtunan beberapa baris .

- Tiap baris terdiri dari runtunan karakter.
- Tiap baris diakhiri oleh karakter khusus, yaitu END-OF-LINE (EOLN) .
- Karakter yang terakhir dari file teks adalah END-OF-FILE (EOF)

EOF adalah menerima argumen nama file dan menghasilkan nilai true jika sudah tidak ada data yang bisa dibaca lagi. Dan EOLN adalah menerima argumen nama file dan menghasilkan nilai true jika sudah tidak ada lagi data yang bisa lagi dalam satu baris.

Untuk melakukan operasi pada file teks, kita perlu mendeklarasikan suatu variabel dengan tipe teks seperti berikut:

```
var
F : text;
```

F adalah sembarang variabel *file* teks dan *readme* adalah nama *file* teks yang akan dibaca, pertama kita harus memanggilnya dengan fungsi:

```
assign(F,'README');
```

Sebelum kita bisa membaca kita harus membuka file tersebut.

```
reset(F);
```

Kita bisa membaca file baris demi baris, misalkan dengan menyatakan suatu variabel s sebagai *string*:

```
readln(F, s);
```

Setelah selesai membaca keseluruhan teks kita harus menutupnya dengan:

```
close(F);
```

Kita membuat suatu *file* teks dengan mengubah kata kunci :

`reset(F);` menjadi `rewrite(F);`

Kemudian: Gunakan *readln(F,s)* untuk membaca file yang akan dikopi dan *writeln(F,s)* untuk menulis ke *file* tujuan (misal kita punya suatu file dengan path '**D:\file1.txt'**) dan kita akan memgkopinya ke *file* dengan path '**D:\file1.txt'**)

Untuk mengatasi *error* pada pembacaan *file* teks digunakan *error-handling*. yaitu :

{$I-} --> membuat Pascal stabil
: --> proses pada file
{$I+} --> deteksi terhadap error

Error dapat di deteksi dengan memanggil fungsi *IOResult*. jika *IOresult* adalah 0, maka tidak terjadi *error*.

Percobaan

Salinlah program berikut ini :

Membuat program file yang sederhana , tetapi sebelumnya anda harus membuat file terlebih dahulu di "**c:\test.txt**"

```
program P0801;
type
  coba = file of byte;
var
  fcoba : coba;
  i : byte;
begin
  Assign(fcoba,'c:\test.txt');
  Reset(fcoba);
  i:=1;
```

```
  Write(fcoba,i);
  Close(fcoba);
end.
```

Compile program tersebut, dan lihat hasil nya di “c:\test.txt”.

Berikut ini akan diperkenalkan program file dengan kompleks dengan menggabungkan record:

```
program P0802;
type
  mhs = record
  nrp : string[8];
  nama : string[20];
  end;
var
  fmhs : file of mhs;
  tmhs : mhs;
begin
  tmhs.nrp := '00112398';
  tmhs.nama := 'Wiwien';
  Assign(fmhs,'c:\test.txt');
  Reset(fmhs);
  Write(fmhs,tmhs);
  Close(fmhs);
end.
```

Compile program tersebut dan lihat hasil nya di “c:\text.txt”

Kemudian salinlah program yang dibawah yang telah dikombinasikan dengan record, array dan disimpan kedalam file.

```
program P0803;
const
  N = 2;
type
  tmhs = record
  nrp : string[8];
  nama : string[20];
  alamat : string;
  end;
  dtmhs = array[1..N] of tmhs;
var
  F : File of dtmhs;
  datamhs : dtmhs;
  i : integer;
begin
  for i:= 1 to N do
  begin
    with datamhs[i] do
    begin
      Write('Masukkan NRP    : ');
```

```
        Readln(nrp);
        Write('Masukkan Nama   : ');
        Readln(nama);
        Write('Masukkan Alamat : ');
        Readln(alamat);
      end;
    end;
    Assign(F,'c:\test.txt');
    Rewrite(F);
    Write(F,datamhs);
    Close(F);

    Assign(F,'c:\test.txt');
    Reset(F);
    Read(F,datamhs);
    Close(F);
    for i : = 1 to N do
    begin
      Writeln(datamhs[i].nrp);
      Writeln(datamhs[i].nama);
      Writeln(datamhs[i].alamat);
    end;
    Readln;
  end.
```

Kompilasi dan jalankan program di atas. Amati hasil dari program tersebut. Program tersebut akan menghasilkan data lebih banyak lagi didalam 1 *file* (berkas).

Latihan

Latihan 8.1

Buatlah 2 procedure untuk menginput data dan menampilkan data pada program **P0803** di atas.

Laporan

Laporan 8.1

Buatlah kesimpulan dari kegiatan praktikum 8 ini.