

Praktikum

Pengantar Algoritma

[MODUL]



Ganjil
2012/2013

Daftar Isi

Daftar Isi	2
KONTRAK PRAKTIKUM	3
Modul 1 Struktur Program Pascal, Tipe Data, Variabel dan Konstanta	7
Modul 2 Alur Percabangan	11
Modul 3 Alur Perulangan.....	16
Modul 4 Prosedur dan Fungsi	20
Modul 5 Array.....	26
Modul 6 Record	29
Modul 7 Project Praktikum.....	32

KONTRAK PRAKTIKUM

Nama Mata Kuliah	: Praktikum Pengantar Algoritma
Kode Mata Praktikum	: TKC 111
SKS	: 1
Mata Kuliah Prasyarat	: -
Dosen Penanggung Jawab	: M. Kautsar Sophan
Dosen Penyusun Modul	: Mochammad Kautsar Sophan Arik Kurniawati Mohammad Mastur
Semester / Th Ajaran	: Ganjil / 2012-2013
Hari Pertemuan / Jam	: Sesuai Jadwal Praktikum
Tempat Pertemuan	: Lab. Common Computing

Gambaran Umum :

Praktikum ini merupakan bagian dari kuliah Pengantar Algoritma Pemrograman. Dalam praktikum ini, praktikan di kenalkan dengan beberapa aspek yang berkaitan dalam pemrograman dasar dan dalam mendesain dan menerapkan sebuah algoritma.

Praktikum ini memuat beberapa modul yang berisi tentang Struktur Program Pascal : Tipe Data, Variabel dan Konstanta; Penyeleksian Kondisi; Pengulangan (*Looping*); Prosedur dan Fungsi; Lariks dan Matriks; serta *Record*. Modul-modul ini harus dapat dikuasai oleh mahasiswa sebagai dasar penguasaan Algoritma Pemrograman.

Mahasiswa diharapkan dapat:

- Mengenal berbagai macam type data dalam bahasa pemrograman dan menggunakan type data yang sesuai dalam menyimpan variabel
- Mengenal, memahami dan dapat menggunakan alur pemrograman
- Mampu membuat dan memanfaatkan fungsi dan prosedur
- Mampu membuat dan menafaatkan array dan record

Tujuan Pembelajaran Praktikum

Mahasiswa mampu memahami dan menerapkan algoritma untuk menyelesaikan sebuah masalah dan dapat menerapkannya dalam bahasa pemrograman Pascal.

Rumusan kompetensi Dasar

1. Mahasiswa mampu membuat flowchart dengan benar
2. mahasiswa memahami type data dan variabel
3. mahasiswa memahami alur pemrograman, dan penggunaannya
4. mahasiswa mampu membuat fungsi dan prosedur
5. mahasiswa mampu menggunakan record
6. mahasiswa mampu menggunakan array

Referensi / Bahan Bacaan

1. Struktur Data Menggunakan Turbo Pascal 6.0, Ir.P.Insap Santosa, MSc, Andi Offset Yogyakarta, 2001.
2. Beginning Algorithms, Simon Haris, James Ross, Wiley Publishing, 2006
3. Algoritma dan Struktur Data dalam Bahasa Java, Adi Nugroho, Penerbit Andi Offset, Yogyakarta, 2008

Alur bagi peserta praktikum :

1. Pada Tiap Sesi Pelaksanaan Praktikum:
 - a. Peserta praktikum menerima dan kemudian mempelajari modul praktikum.
 - b. Peserta praktikum mengerjakan tugas pendahuluan yang diberikan.
 - c. Peserta praktikum melakukan asistensi tugas pendahuluan tersebut. Asistensi ini digunakan sebagai bagi asisten untuk menilai kesiapan peserta juga berfungsi sebagai ajang diskusi peserta praktikum atas kesulitan yang dialaminya.
 - d. Peserta Praktikum melakukan implementasi Tugas Praktikum di kelas/
 - e. Peserta praktikum mendemokan implementasi pada asisten. Penilaian yang dilakukan oleh asisten bersifat WISIWYG (What I see Is What You Get)
2. Pada akhir pelaksanaan seluruh sesi praktikum, ada Posttest, berupa test program.

Tugas

NO	KRITERIA	INDIVIDU	KELOMPOK
1	Jenis Tugas	Post Test	1. Tugas Pendahuluan 2. Tugas Sesi Praktikum
2	Tujuan	1. Mahasiswa memahami Konsep Strukt Data	Mampu menyelesaikan masalah yang diberikan dengan program
3	Jadwal Pengumpulan	Diakhir Praktikum	1. Di Awal Sesi Praktikum 2. Di akhir sesi praktikum
4	Keluaran Tugas	1. Program + Penjelasan	1. Program 2. Listing Program beserta penjelasan (soft copy – Hardcopy di kumpulkan di akhir kegiatan praktikum)

Bobot Prosentase Penilaian Praktikum :

1. Pelaksanaan Praktikum – 7 Sesi / Modul
 - a. Tugas pendahuluan 15%

b. Kehadiran dan tugas praktikum 35%

c. Asistensi dan laporan praktikum 35%

2. Posttest + Laporan resmi 15%

Form Nilai

Kelas : A	Asdos :
-----------	---------

No	NRP	NAMA	Modul 1			Modul 2			Modul 3			Modul 4			Modul 5			Modul 6			Rata-rata			Posttest + Lapres Akhir	Nilai Akhir	Nilai Huruf
			n1	n2	n3	n1	n2	n3																		
1	010041110000		85	80	78	85	83	84	83	82	78	85	80	76	83	79	76	85	78	74	84,33	80,33	77,67	78	77,45	B+
2	008041110010		83	78	79	85	79	72	81	75	75	84	77	74	85	70	71	84	72	71	83,67	75,17	73,67	65	72,49	B
3	010041110005		87	85	88	86	85	87	85	87	86	86	88	87	87	78	79	86	78	77	86,17	83,50	84,00	83	83,10	A
4	010041110009		85	80	78	84	82	83	84	78	80	85	80	79	84	72	73	85	74	73	84,50	77,67	77,67	73	77,19	B+
5	010041110007		85	79	78	85	83	85	85	80	78	84	78	80	83	74	76	84	77	75	84,33	78,50	78,67	78	77,76	B+
6	010041110009		85	79	78	85	83	83	85	78	79	83	79	78	85	75	75	83	76	75	84,33	78,33	78,00	78	77,42	B+
7																										

Presentase Nilai

Kegiatan	%	KET
Tugas Pendahuluan	15%	n1
Kehadiran + Tugas	35%	n2
Asistensi + Laporan	35%	n3
Posttest + Lapres	15%	post test

Grade Nilai

Kegiatan	%
A	> 80
B+	> 75 - 80
B	> 70 - 75
C+	> 65 - 70
C	> 60 - 65
D	50 - 60
E	< 50

Nilai	Point	Range
A	4	> 80
B+	3,5	>75 – 80
B	3	>70 – 75
C+	2,5	> 65 – 70
C	2	>60 – 65
D	1	50 - 60
E	0	< 50

Catatan : Range Nilai bisa berubah sesuai dengan kondisi kelas

Peraturan Praktikum :

1. Praktikan harus menghadiri setiap sesi praktiku, tidak hadir tanpa keterangan akan menyebabkan hak menjadi praktikan gugur (nilai **E**).
2. Tugas pendahuluan diserahkan sebelum praktikum dimulai.
3. Laporan praktikum dianggap sah apabila praktikan telah melakukan asistensi sesuai dengan materi praktikum.
4. Laporan resmi adalah gabungan dari semua laporan praktikum tiap sesi.
5. Perbuatan Plagiat / Copy –Paste / Mencontoh sangat dilarang. Jika ada indikasi, akan di beri nilai E.

Lain – Lain

- Pengumuman : kautsarsophan.wordpress.com
- Pertanyaan : kautsar@if.trunojoyo.ac.id
- Ketua Koordinator Asisten Praktikum / HP :
- Bahasa Pemrograman : Pascal
- Tool
 - Turbo Pascal 5.5
 - Turbo Pascal 7
 - Free Pascal
 - GNU Pascal, dll

Modul 1

Struktur Program Pascal, Tipe Data, Variabel dan Konstanta

I. Teori

Struktur program Pascal terdiri dari tiga bagian (Gambar 1.1), yaitu:

- **Judul Program.** Judul ini sifatnya adalah optional dan tidak signifikan.
- **Bagian Deklarasi.** Bagian ini digunakan bila program akan untuk mendefinisikan berbagai macam *identifiser*, seperti label, konstanta, tipe, variabel, prosedur dan fungsi.
- **Bagian Deskripsi.** bagian ini digunakan untuk meletakkan seluruh instruksi program. Instruksi-instruksi yang akan diberikan untuk dikerjakan ditulis di antara “Begin” dan “End”. Penulisan “End” diakhiri dengan tanda baca titik (‘.’). Yang perlu diperhatikan bahwa setiap instruksi diakhiri dengan tanda baca titik koma (;’).

```

program NAMA_PROGRAM;
{ Penjelasan mengenai program, yang berisi uraian singkat
mengenai apa yang dilakukan oleh program }
(* DEKLARASI *)
[const]
{ semua nama tetapan dan harga tetapannya didefinisikan di
sini}
[type]
{ semua nama tipe bentukan didefinisikan di sini }
[var]
{ semua nama variabel global didefinisikan di sini }

{ deklarasi prosedur dan fungsi didefinisikan di sini }

(* DESKRIPSI *)
begin
  { semua instruksi program dituliskan di sini }
end.
```

Gambar 1.1. Struktur Program Bahasa Pascal

Tabel 1.1. Notasi Bahasa Pascal

Kelompok	Notasi Pascal	Ket.
1. Tipe Dasar	boolean byte shortint word integer longint real double	 1 byte 1 byte 2 byte 2 byte 4 byte

	extended char string string[N] record field1: type ; field2: type ; ... fieldN: type ; end;	
2. Operator a. Aritmatika b. Perbandingan c. Logika	+ - * / div mod < <= > >= = <> not and or xor	Penjumlahan Pengurangan Perkalian Pembagian Hasil bagi Sisa bagi Lebih kecil dari Lebih kecil atau sama dengan Lebih besar dari Lebih besar atau sama dengan Sama dengan Tidak sama dengan
3. Komentar	{ komentar } (* komentar *)	
4. Lain-lain	const type true false	

II. Prepraktikum

Kerjakan tugas-tugas di bawah ini sebelum praktikum dimulai.

1. Buatlah contoh deklarasi masing-masing tipe dasar dalam Bahasa Pascal!
2. Isilah tabel berikut ini untuk melakukan penelusuran kebenaran penulisan nama variabel-variabel ini.

Tabel 1.2. Penelusuran Kebenaran Nama Variabel

Nama Variabel	Benar/Salah	Alasan	Pembetulan
_data2			
Huruf5			
\$kondisi			

Alamat rumah			
4_alasan			
Data_nilai			
Max			

3. Jelaskan aturan-aturan yang digunakan dalam penulisan nama variable.
4. Buatlah sebuah Algoritma untuk mencari keliling dan luas sebuah lingkaran, gunakan deklarasi konstanta untuk menyimpan nilai Phi.

III. Kegiatan Praktikum

1. Buatlah sebuah program sederhana yang berisi deklarasi masing-masing tipe dasar sesuai tugas prepraktikum no.1, kemudian pada bagian deskripsi buatlah contoh instruksi masukan untuk mengisi data dan intruksi keluaran untuk menampilkan data dari masing-masing tipe dasar tersebut!
2. Buat program untuk tugas prepratikum no.4!
3. Tugas tambahan (ditentukan oleh asisten praktikum)!

IV. Tugas Akhir

Buatlah algoritma dan terjemahkan dalam Bahasa Pascal untuk menghitung akar persamaan kuadrat : $ax^2 + bx + c = 0$

Masukan:

Masukkan nilai a : ___

Masukkan nilai b : ___

Masukkan nilai c : ___

Keluaran

Akar persamaan: x1 = ___

x2 = ___

V. Catatan Praktikum

Sumber Pustaka :

1. Noor Ifada, 2005. **Diktat Mata Kuliah Algoritma Pemrograman**, Hibah Kompetisi A1 Jurusan Teknik Informatika Universitas Trunojoyo.
2. Rinaldi Munir, 2003. **Algoritma dan Pemrograman dengan Pascal dan C edisi Kedua**, Penerbit Informatika, Bandung.

Modul 2

Alur Percabangan

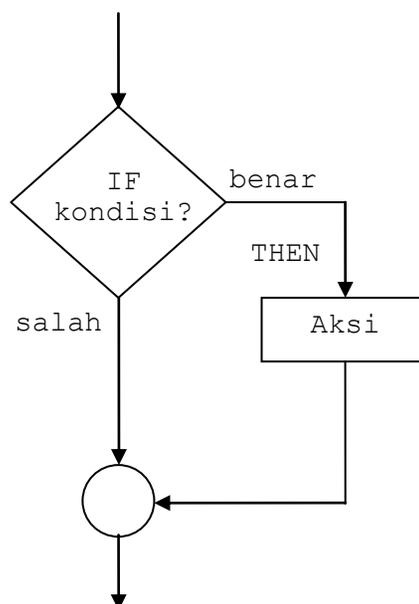
I. Teori

Pada umumnya satu permasalahan yang kompleks memerlukan suatu penyelesaian kondisi. Dengan menyeleksi suatu kondisi, maka selanjutnya dapat ditentukan tindakan apa yang harus dilakukan, tergantung pada hasil kondisi penyeleksian. Jadi suatu aksi hanya dikerjakan apabila persyaratan atau kondisi tertentu terpenuhi.

a. Penyeleksian Satu Kasus, dengan menggunakan struktur IF-THEN:

Pada penyeleksian satu kasus, *kondisi* akan diseleksi oleh statemen *if*. Bila *kondisi* bernilai benar (*true*), maka *aksi* sesudah kata *then* (dapat berupa satu atau lebih *aksi*) akan diproses. Bila *kondisi* bernilai salah (*false*), maka tidak ada *aksi* yang akan dikerjakan. Statemen *endif* adalah untuk mempertegas bagian awal dan bagian akhir struktur IF-THEN.

Gambar diagram alir penyelesaian satu kasus untuk struktur IF-THEN ditunjukkan dalam Gambar 2.1 dan struktur penulisan dalam bahasa pascal dapat dilihat dari Gambar 2.2.



Gambar 2.1. Diagram Alir Struktur Penyeleksian Satu Kasus (IF-THEN)

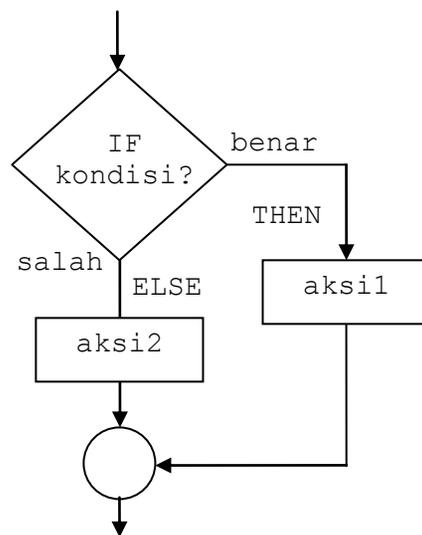
```
if kondisi then
aksi
```

Gambar 2.2. Struktur Bahasa Pascal Untuk Penyeleksian Satu Kasus (IF-THEN)

b. Penyeleksian Dua Kasus, menggunakan struktur IF-THEN-ELSE:

Dalam struktur IF-THEN-ELSE, *aksi1* akan dilaksanakan jika *kondisi* bernilai benar (*true*), dan jika *kondisi* bernilai salah (*false*) maka *aksi2* yang akan dilaksanakan. Statemen *else* menyatakan ingkaran (*negation*) dari *kondisi*.

Gambar diagram alir penyelesaian dua kasus untuk struktur IF-THEN-ELSE ditunjukkan dalam Gambar 2.3 dan struktur penulisan dalam bahasa pascal dapat dilihat dari Gambar 2.4.



Gambar 2.3. Diagram Alir Struktur IF-THEN-ELSE

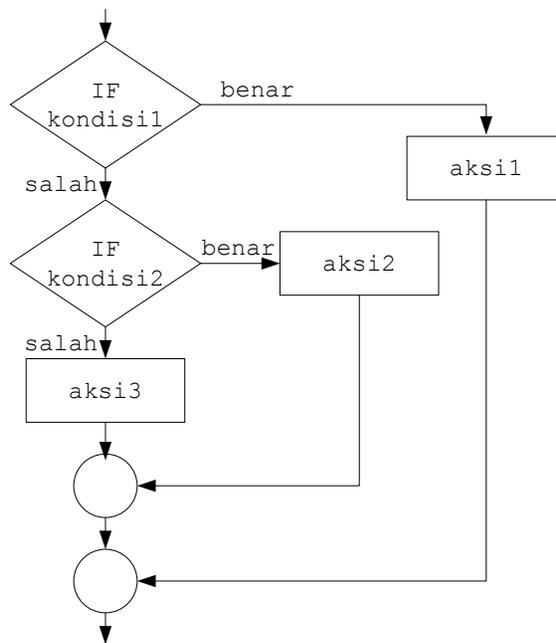
```
if kondisi then
    aksi1
else
    aksi2
```

Gambar 2.4. Struktur Bahasa Pascal Untuk Penyeleksian Dua Kasus (IF-THEN-ELSE)

c. Penyeleksian Tiga Kasus atau Lebih (Penyeleksian Tersarang)

Untuk penyeleksian tiga kasus atau lebih juga menggunakan struktur IF-THEN-ELSE sebagaimana halnya permasalahan dua kasus.

Gambar diagram alir penyelesaian tiga kasus untuk struktur IF-THEN-ELSE ditunjukkan dalam Gambar 2.5 dan struktur penulisan dalam bahasa pascal dapat dilihat dari Gambar 2.6.



Gambar 2.5. Diagram alir dari struktur tiga kasus IF-THEN-ELSE (tersarang)

```

if kondisi1 then
    aksi1
else
    if kondisi2 then
        aksi2
    else
        aksi3
  
```

Gambar 2.6. Struktur Bahasa Pascal untuk Penyeleksian Tiga Kasus IF-THEN-ELSE (tersarang)

d. Struktur CASE. Struktur ini dapat digunakan untuk menganalisa penyeleksian dua kasus atau lebih dan bentuknya adalah lebih sederhana daripada struktur IF-THEN-ELSE yang memiliki bentuk bertingkat-tingkat. Gambar 2.7 memperlihatkan Struktur CASE.

```

case (nama)
    kondisi1 : aksi1
    kondisi2 : aksi2
    ...
    kondisiN : aksiN
    [otherwise aksiX]
  
```

Gambar 2.7. Struktur Bahasa Pascal Untuk Seleksi CASE

Kondisi1, kondisi2, ... kondisiN dapat bernilai benar atau salah. Tiap *kondisi* diperiksa nilai kebenarannya mulai dari *kondisi* pertama sampai ditemukan *kondisi* yang benar. Jika *kondisi* ke-*k* benar, maka *aksi* ke-*k* dilaksanakan, selanjutnya keluar dari struktur CASE. *Aksi* yang dipasangkan dengan *kondisi* ke-

k dapat lebih dari satu, karena itu ia berupa runtunan. Jika tidak ada satupun kondisi yang benar, maka aksi sesudah *otherwise* (optional) dikerjakan.

II. Prepraktikum

Kerjakan tugas-tugas di bawah ini sebelum praktikum dimulai.

1. Jelaskan pada kondisi apa sebaiknya struktur seleksi if dan case digunakan! Apa perbedaan dari kedua struktur tersebut?
2. Buatlah program untuk menghitung total 3 buah hambatan dengan menu pilihan: Jika dimasukkan angka 1, maka total hambatan dihitung secara seri dan jika dimasukkan angka 2, maka total hambatan dihitung secara paralel.

Input : R1, R2, R3 dan pilihan

Output : Total Hambatan

Contoh tampilan:

Masukkan R1 = 1

Masukkan R2 = 1

Masukkan R3 = 1

Tekan 1 untuk seri atau tekan 2 untuk paralel ? (1)

Total Hambatan = 3

Alternatif tampilan lain:

Masukkan R1 = 1

Masukkan R2 = 2

Masukkan R3 = 3

Tekan 1 untuk seri atau tekan 2 untuk paralel ? (2)

Total Hambatan = 0.55

Rumus:

Seri : $R_{total} = R1 + R2 + R3$

Paralel :

$R_{total} = (R1 \cdot R2 \cdot R3) / (R1 \cdot R2 + R1 \cdot R3 + R2 \cdot R3)$

3. Buatlah algoritma konversi dari nilai angka menjadi nama hari. (Asumsi: 1=Senin, 2=Selasa, 3=Rabu, 4=Kamis, 5=Jumat, 6=Sabtu, 7=Minggu)

Contoh keluaran yang diinginkan:

Masukkan Angka yang menunjukkan hari : 1

Hari yang anda pilih : Senin

III. Kegiatan Praktikum

Pada saat praktikum, kerjakan tugas-tugas berikut ini:

1. Buat program untuk prepraktikum 2!
2. Buat program untuk prepraktikum 3!
3. Tugas tambahan (ditentukan oleh asisten praktikum)!

IV. Tugas Akhir

Buatlah algoritma dan terjemahkan dalam Bahasa Pascal untuk mengurutkan tiga buah bilangan bulat (dari yang terkecil ke yang terbesar) yang dimasukkan melalui piranti masukan! (Asumsi: ketiga bilangan yang dimasukkan adalah bilangan yang berbeda)

Contoh:

Masukkan bilangan pertama : 9

Masukkan bilangan kedua : 1

Masukkan bilangan ketiga : -7

Urutan bilangan: -7 1 9

V. Catatan Praktikum

Referensi :

1. Ifada N, **Diktat Mata Kuliah Algoritma Pemrograman**, Bangkalan: Jurusan Teknik Informatika Universitas Trunojoyo, 2005.
2. Munir R, **Algoritma dan Pemrograman dengan Pascal dan C edisi Kedua**, Bandung: Informatika, 2003.

Modul 3

Alur Perulangan

I. Teori

Terdapat dua model struktur pengulangan, yaitu:

1. Struktur Pengulangan Tanpa Kondisi (*unconditional looping*). Di dalam struktur ini, instruksi-instruksi di dalam badan pengulangan diulangi sejumlah kali yang dispesifikasikan (jumlah pengulangan sudah diketahui sebelum eksekusi). Contoh: Struktur FOR.
2. Struktur Pengulangan Dengan Kondisi (*conditional looping*). Di dalam struktur ini, jumlah pengulangan tidak diketahui sebelum eksekusi program. Yang dapat ditentukan hanya kondisi berhenti pengulangan, artinya instruksi-instruksi di dalam badan pengulangan diulangi sampai kondisi berhenti terpenuhi. Contoh: Struktur WHILE dan Struktur REPEAT.

Macam-macam struktur pengulangan:

a. Struktur FOR

Struktur pengulangan FOR digunakan untuk mengulang statemen atau satu blok statemen berulang kali. Jumlah pengulangan diketahui atau dapat ditentukan sebelum eksekusi. Untuk mencacah sudah jumlah pengulangan diperlukan sebuah variabel pencacah (*counter*). Variabel ini nilainya selalu bertambah satu setiap kali pengulangan dilakukan. Jika cacah pengulangan sudah mencapai jumlah yang dispesifikasikan, maka proses pengulangan berhenti. Pada struktur FOR, *pencacah* haruslah dari tipe data yang memiliki predecessor dan successor, yaitu integer atau karakter. Tipe riil tidak dapat digunakan sebagai *pencacah*. Aksi adalah satu atau lebih instruksi yang diulang.

Bentuk struktur FOR ada dua macam:

- Menaik (*ascending*)

Pada struktur FOR menaik, *nilai_awal* harus lebih kecil atau sama dengan *nilai_akhir*. Jika *nilai_awal* lebih besar dari *nilai_akhir*, maka badan pengulangan tidak dimasuki. Pada awalnya, *pencacah* diinisialisasikan dengan *nilai_awal*. Nilai *pencacah* secara otomatis bertambah satu setiap kali aksi pengulangan dimasuki, sampai akhirnya nilai *pencacah* sama dengan *nilai_akhir*. Jumlah pengulangan yang terjadi = $nilai_akhir - nilai_awal + 1$. Struktur Bahasa Pascal untuk pengulangan FOR menaik ditunjukkan pada Gambar 3.1.

```
for pencacah:=nilai_awal to nilai_akhir do
aksi
```

Gambar 3.1. Struktur Bahasa Pascal Untuk Pengulangan FOR Menaik

- Menurun (*descending*)

Pada struktur FOR menurun, *nilai_akhir* harus lebih besar atau sama dengan *nilai_awal*. Jika *nilai_akhir* lebih kecil dari *nilai_awal*, maka badan pengulangan tidak dimasuki. Pada awalnya, *pencacah* diinisialisasikan dengan *nilai_akhir*. Nilai *pencacah* secara otomatis berkurang satu setiap kali *aksi* diulangi, sampai akhirnya nilai *pencacah* sama dengan *nilai_awal*. Jumlah pengulangan yang terjadi = $nilai_akhir - nilai_awal + 1$. Struktur Bahasa Pascal untuk pengulangan FOR menurun ditunjukkan pada Gambar 3.2.

```
for pencacah:=nilai_akhir downto nilai_awal do
aksi
```

Gambar 3.2. Struktur Bahasa Pascal Untuk Pengulangan FOR Menurun

b. Struktur WHILE

Pada struktur WHILE, *aksi* (atau runtunan *aksi*) akan dilaksanakan berulang kali selama *kondisi* bernilai *true*. Jika kondisi bernilai *false*, badan pengulangan tidak akan dilaksanakan, yang berarti pengulangan selesai. Yang harus diperhatikan adalah pengulangan harus berhenti. Pengulangan yang tidak pernah berhenti menandakan bahwa logika pemrograman tersebut salah. Pengulangan berhenti apabila *kondisi* bernilai *false*. Agar *kondisi* suatu saat bernilai *false*, maka di dalam badan pengulangan harus ada instruksi yang mengubah nilai variabel *kondisi*.

```
while kondisi do
aksi
```

Gambar 3.3. Struktur Bahasa Pascal Untuk Pengulangan WHILE

c. Struktur REPEAT

Struktur ini mendasarkan pengulangan pada kondisi *boolean*. Aksi di dalam badan pengulangan diulang sampai *kondisi boolean* bernilai *true*. Dengan kata lain, jika kondisi *boolean* masih *false*, pengulangan masih terus dilakukan. Karena proses pengulangan suatu saat harus berhenti, maka di dalam badan pengulangan harus ada *aksi* yang mengubah nilai variabel *kondisi*. Struktur REPEAT mempunyai makna yang sama dengan WHILE, dan dalam beberapa masalah kedua struktur tersebut komplemen satu sama lain.

```
repeat
aksi
```

II. Prepraktikum

Kerjakan tugas-tugas di bawah ini sebelum praktikum dimulai.

1. Jelaskan perbedaan antara struktur pengulangan FOR, WHILE dan REPEAT dan bagaimana hasilnya untuk implementasi masing-masing!
2. Buatlah program untuk menampilkan bilangan ganjil untuk mahasiswa ber-NIM ganjil, sedangkan bilangan genap untuk mahasiswa ber-NIM genap. Deretan bilangan itu berhenti sampai dengan angka 100
3. Buatlah algoritma untuk menampilkan deret bilangan 1, -2, 4, -8 sampai dengan 20 suku.
4. Buatlah algoritma untuk menampilkan deret bilangan hasil faktorial 1, 2,6,24, ... (berhenti jika sama dengan banyaknya suku tertentu yang dimasukkan melalui piranti masukan, nilai awal deret dimasukkan melalui piranti masukan)!

Contoh:

Masukan suku pertama : 1

Masukan batas suku: 4

Deret Keluaran: 1, 2,6,24

III. Kegiatan Praktikum

Pada saat praktikum, kerjakan tugas-tugas berikut ini:

1. Buat program untuk tugas prepraktikum 2!
2. Buat program untuk tugas prepraktikum 3!
3. Tugas tambahan (ditentukan oleh asisten praktikum)!

IV. Tugas Akhir

Buatlah algoritma dan terjemahkan dalam Bahasa Pascal untuk membuat bentuk “X” dari deretan angka berdasarkan batas nilai yang dimasukkan melalui piranti masukan!

Contoh:

Masukkan batas nilai: 5

Hasil X deretan angka:

1 1

2 2

3 3

4 4

5 5

V. Catatan Praktikum

Referensi :

1. Ifada N, **Diktat Mata Kuliah Algoritma Pemrograman**, Bangkalan: Jurusan Teknik Informatika Universitas Trunojoyo, 2005.
2. Munir R, **Algoritma dan Pemrograman dengan Pascal dan C edisi Kedua**, Bandung: Informatika, 2003.

Modul 4

Prosedur dan Fungsi

I. Teori

Seringkali dalam pembuatan program yang berukuran besar, program tersebut dapat dipecah menjadi beberapa subprogram/modul yang lebih kecil. Tiap modul dapat dipasangkan ke program lain yang membutuhkannya. Jenis modul program ada dua macam, yaitu Prosedur dan Fungsi.

a. Prosedur

Prosedur adalah modul program yang mengerjakan aktivitas yang spesifik dan hasilnya diperoleh dengan membandingkan keadaan awal dan keadaan akhir pada pelaksanaan sebuah prosedur. Sehingga pada setiap prosedur **keadaan awal** harus didefinisikan sebelum rangkaian instruksi di dalam prosedur dilaksanakan dan **keadaan akhir** yang diharapkan setelah rangkaian instruksi dilaksanakan.

Struktur prosedur terdiri dari tiga bagian (Gambar 4.1), yaitu:

- Judul
- Deklarasi
- Deskripsi/Badan Prosedur

```
procedure NAMA_PROSEDUR;  
{ Spesifikasi Prosedur, menyebutkan nama, parameter, kondisi  
awal, kondisi akhir dan proses }  
{ Kondisi awal : keadaan sebelum prosedur dilaksanakan }  
{ Kondisi akhir : keadaan setelah prosedur dilaksanakan }  
  
(* DEKLARASI *)  
  { Semua nama yang dipakai prosedur didefinisikan di sini  
  dan hanya berlaku lokal }  
  
(* DESKRIPSI *)  
begin  
  { badan prosedur, berisi semua langkah atau aksi }  
end;
```

Gambar 4.1. Struktur Bahasa Pascal Untuk Prosedur

Cara memanggil prosedur: NAMA_PROSEDUR;

Ketika NAMA_PROSEDUR dipanggil, kendali program berpindah secara otomatis ke prosedur tersebut. Instruksi di dalam badan prosedur dilaksanakan. Setelah semua instruksi selesai dilaksanakan, kendali program berpindah secara otomatis ke instruksi sesudah pemanggilan prosedur.

Kebanyakan program memerlukan pertukaran informasi antara prosedur (atau fungsi) dan titik dimana ia dipanggil. Penggunaan parameter menawarkan

mekanisme penukaran informasi tersebut. Tiap item data ditransfer antara **parameter aktual** dan **parameter formal** yang bersesuaian. Parameter aktual adalah parameter yang disertakan ketika pemanggilan, sedangkan parameter formal adalah parameter yang dideklarasikan di bagian *header* prosedur itu sendiri. Ketika prosedur dipanggil, parameter aktual menggantikan parameter formal.

Berdasarkan maksud penggunaannya, terdapat tiga jenis parameter formal yang disertakan di dalam prosedur, yaitu:

- **Prosedur dengan Parameter Masukan**

Parameter masukan adalah parameter yang nilainya berlaku sebagai masukan untuk prosedur. Pada parameter masukan, nilai parameter aktual diisikan ke dalam parameter formal yang bersesuaian. Nilai ini digunakan di dalam badan prosedur yang bersangkutan. Nilai yang dinyatakan oleh parameter masukan tidak dapat dikirim dalam arah sebaliknya. Itulah alasan mengapa parameter jenis ini diacu sebagai parameter masukan. Perubahan nilai parameter di dalam badan prosedur tidak mengubah nilai parameter aktual. Karena yang dipentingkan adalah nilainya, maka nama parameter aktual boleh berbeda dengan nama parameter formal yang bersesuaian.

Struktur Bahasa Pascal untuk prosedur dengan parameter masukan ditunjukkan oleh Gambar 4.2.

```
procedure NAMA_PROSEDUR(parameter : tipe_param);
{ Spesifikasi Prosedur, menyebutkan nama, parameter, kondisi
awal, kondisi akhir dan proses }
{ Kondisi awal : keadaan sebelum prosedur dilaksanakan }
{ Kondisi akhir : keadaan setelah prosedur dilaksanakan }

(* DEKLARASI *)
  { Semua nama yang dipakai prosedur didefinisikan di sini
  dan hanya berlaku lokal }

(* DESKRIPSI *)
begin
  { badan prosedur, berisi semua langkah atau aksi }
end;
```

Gambar 4.2. Struktur Bahasa Pascal Untuk Prosedur Dengan Parameter Masukan

Cara memanggil prosedur: NAMA_PROSEDUR(parameter_aktual);

- **Prosedur dengan Parameter Masukan/Keluaran**

Parameter masukan/keluaran adalah parameter yang berfungsi sebagai masukan sekaligus keluaran bagi prosedur tersebut. Bila prosedur yang mengandung parameter keluaran dipanggil, nama parameter aktual di dalam program

pemanggil menggantikan nama parameter formal yang bersesuaian di dalam prosedur. Jadi, nama parameter aktual akan digunakan selama pelaksanaan prosedur. Akibat penggunaan parameter masukan/keluaran, bila parameter actual diubah nilainya di dalam badan prosedur, maka sesudah pemanggilan prosedur nilai parameter aktual di titik pemanggilan juga berubah. Ini berbeda dengan parameter masukan, yang dalam hal ini meskipun nilai parameter aktual di dalam badan prosedur diubah, nilai parameter aktual tersebut tidak berubah di titik pemanggilan.

Struktur Bahasa Pascal untuk prosedur dengan parameter masukan/keluaran ditunjukkan oleh Gambar 4.3.

```
procedure NAMA_PROSEDUR(var parameter : tipe_param);  
{ Spesifikasi Prosedur, menyebutkan nama, parameter, kondisi  
awal, kondisi akhir dan proses }  
{ Kondisi awal : keadaan sebelum prosedur dilaksanakan }  
{ Kondisi akhir : keadaan setelah prosedur dilaksanakan }  
  
(* DEKLARASI *)  
  { Semua nama yang dipakai prosedur didefinisikan di sini  
  dan hanya berlaku lokal }  
  
(* DESKRIPSI *)  
begin  
  { badan prosedur, berisi semua langkah atau aksi }  
end;
```

Gambar 4.3. Struktur Bahasa Pascal Untuk Prosedur Dengan Parameter Masukan/Keluaran

Cara memanggil prosedur: NAMA_PROSEDUR(parameter_aktual);

b. Fungsi

Fungsi juga merupakan modul program yang mempunyai tujuan spesifik. Fungsi memberikan/mengembalikan (*return*) sebuah nilai dan tipe tertentu (tipe dasar atau tipe bentukan). Fungsi diakses dengan memanggil namanya. Selain itu, fungsi juga dapat mengandung daftar parameter formal. Parameter pada fungsi selalu merupakan parameter masukan. Jenis parameter masukan pada fungsi disebabkan oleh kenyataan bahwa parameter pada fungsi merupakan masukan yang digunakan oleh fungsi tersebut untuk menghasilkan nilai.

Struktur Bahasa Pascal untuk fungsi ditunjukkan oleh Gambar 4.4.

```

function NAMA_FUNGSI(daftar parameter formal) : tipe_hasil;
{ Spesifikasi Fungsi, menjelaskan apa yang dilakukan dan
dikembalikan oleh fungsi }

(* DEKLARASI *)
{ Semua nama yang dipakai fungsi didefinisikan di sini dan
hanya berlaku lokal }

(* DESKRIPSI *)
begin
{ badan fungsi, berisi semua langkah atau aksi algoritma
untuk menghasilkan nilai yang akan dikembalikan }
NAMA_FUNGSI:=hasil; { pengembalian nilai yang dihasilkan
fungsi }
end;

```

Gambar 4.4. Struktur Bahasa Pascal Untuk Fungsi

c. Variabel Global dan Lokal

Variabel-variabel yang dideklarasikan di dalam bagian Deklarasi Prosedur ataupun Fungsi bersifat “lokal” dan hanya dapat digunakan di dalam Prosedur/Fungsi yang melingkupinya saja. Sedangkan variabel-variabel yang dideklarasikan di dalam program utama bersifat “global” dan dapat digunakan di bagian manapun di dalam program, baik di dalam program utama maupun di dalam Prosedur/Fungsi.

II. Prepraktikum

Kerjakan tugas-tugas di bawah ini sebelum praktikum dimulai.

1. Prosedur dan Fungsi merupakan pemrograman modular. Apakah perbedaan antara keduanya? Kapankah suatu fungsi dibuat dan digunakan? Dan kapankah suatu prosedur digunakan? Bisakah suatu kasus diselesaikan menggunakan prosedur dan fungsi? Sertai alasannya
2. Sebuah truk bergerak dari Kota Bangkalan Menuju kedalaman Pulau Madura. Pada Km pertama ia menghabiskan 10lt bahanbakar. Pada Km ke 2 ia menghabiskan 8lt bahan bakar(atau 80% dibanding Km pertama). Demikian seterusnya setiap Km sebelumnya. Susun algoritma untuk menghitung dan menampilkan total bahan bakar yang dihabiskan setelah mencapai 100km
Batas nilai yang bisa dimasukkan melalui piranti masukan dengan menggunakan Prosedur!

Contoh:

Masukkan batas nilai: 100

Bahan Bakar yang telah dihabiskan (lt)

3. Buatlah algoritma untuk menghitung rumus-rumus persamaan kuadrat ($ax^2 + bx + c = 0$) berikut ini:

- Rumus diskriminan ($D = \sqrt{b^2 - 4 * a * c}$)
- Rumus mencari akar-akar persamaan

$$(X_1 = \frac{-b + \sqrt{b^2 - 4 * a * c}}{2 * a}; X_2 = \frac{-b - \sqrt{b^2 - 4 * a * c}}{2 * a}).$$

Setiap rumus dijadikan sebuah fungsi tersendiri dengan parameter-parameter yang dimasukkan dari luar fungsi.

Contoh:

Masukkan nilai a : 1
Masukkan nilai b : 4
Masukkan nilai c : 3

Hasil diskriminan (D) : 2
Nilai X_1 : -1
Nilai X_2 : -3

4. PLN menerapkan pembayaran listrik perumahan dengan cara perhitungan sebagai berikut :-Untuk golongan 1 tarif : Rp 1000/kWh
-Untuk golongan 2 tarif : Rp 2000/kWh

Dengan syarat penghitungan:Minimum pembayaran adalah 100 kWh sedangkan untuk pemakaian 1000kWh dan seterusnya tarifnya ditambah 10% dari total pembayaran.

Tampilan

Golongan Pemakaian Pembayaran

1	50 kwh	100.000
2	150kwh	200.000
1	1000kwh	1.100.000

III. Kegiatan Praktikum

Pada saat praktikum, kerjakan tugas-tugas berikut ini:

1. Buat program untuk tugas prepraktikum 2!
2. Buat program untuk tugas prepraktikum 3!
3. Buat program untuk tugas prepraktikum 3!
4. Tugas tambahan (ditentukan oleh asisten praktikum)!

IV. Tugas Akhir

Buatlah algoritma dan terjemahkan dalam Bahasa Pascal untuk menghitung bilangan fibonacci dengan menggunakan teknik rekursif dan non-rekursif!

Contoh:

Masukkan fibonaci deret bilangan fibbonaci 1, 1, 2, 3, 5, 8, 13, ...

Masukan suku pertama : 1

Masukan batas suku: 6

Deret Keluaran: 1, 1, 2, 3, 5, 8

V. Catatan Praktikum

Sumber Pustaka :

1. Ifada N, **Diktat Mata Kuliah Algoritma Pemrograman**, Bangkalan: Jurusan Teknik Informatika Universitas Trunojoyo, 2005.
2. Munir R, **Algoritma dan Pemrograman dengan Pascal dan C edisi Kedua**, Bandung: Informatika, 2003.

Modul 5

Array

I. Teori

Larik adalah tipe terstruktur yang terdiri dari sejumlah elemen-elemen yang bertipe sama. Banyaknya elemen dalam suatu larik ditunjukkan oleh suatu indeks yang harus merupakan tipe data yang menyatakan keterurutan, misalnya integer atau karakter (karena ia menyatakan posisi data). Tiap-tiap elemen di larik dapat diakses langsung melalui indeksnya. Suatu larik memiliki jumlah elemen yang jumlahnya tetap, sehingga jumlah elemen larik tidak dapat diubah selama pelaksanaan program. Gambar 5.1. memperlihatkan struktur deklarasi Larik di dalam Bahasa Pascal.

```
(* DEKLARASI *)  
var  
  NamaLarik : array[1..JumlahElemen] of TipeElemen;
```

Gambar 5.1. Struktur Deklarasi Bahasa Pascal Untuk Larik

Cara mengacu elemen larik : NamaLarik[indeks]

Larik dapat berupa larik satu dimensi ataupun lebih dari satu dimensi. Matriks merupakan contoh larik yang memiliki dimensi lebih dari satu. Struktur penyimpanan matriks di dalam memori utama yang setiap individu elemennya diakses dengan menggunakan dua buah indeks (yang biasanya dikonotasikan dengan baris dan kolom). Karena matriks sebenarnya adalah larik, konsep umum dari larik juga berlaku untuk matriks, yaitu:

- Kumpulan elemen yang bertipe sama, dapat dapat berupa tipe dasar (*integer, real, boolean, char, dan string*), atau tipe terstruktur seperti *record*.
- Setiap elemen data dapat diakses secara langsung jika indeksnya (baris dan kolom) diketahui.
- Merupakan struktur data yang statik, artinya jumlah elemennya sudah dideklarasikan terlebih dahulu di dalam bagian DEKLARASI dan tidak bisa diubah selama pelaksanaan program

Elemen matriks diakses melalui indeks baris dan indeks kolomnya. Jika indeks baris dinyatakan dengan *i* dan indeks kolom dinyatakan dengan *j*, maka notasi algoritmik untuk mengakses elemen pada baris *i* dan kolom *j* adalah

nama_matriks[i,j]

II. Prepraktikum

Kerjakan tugas-tugas di bawah ini sebelum praktikum dimulai.

1. Sebutkan dan jelaskan keuntungan-keuntungan penggunaan larik dan matriks!
2. Kapan sebaiknya larik dan matriks digunakan? Jelaskan!
3. Buat algoritma untuk menentukan nilai min, kuartil bawah, kuartil tengah, kuartil atas dan nilai max dari sejumlah bilangan bulat yang dimasukkan melalui piranti masukan.

Contoh:

Jumlah bilangan yang dimasukkan: 11

Bilangan ke-1: 75

Bilangan ke-2: 60

Bilangan ke-3: 70

Bilangan ke-4: 55

Bilangan ke-5: 85

Bilangan ke-6: 65

Bilangan ke-7: 80

Bilangan ke-8: 90

Bilangan ke-9: 50

Bilangan ke-10: 95

Bilangan ke-11: 100

Nilai Min : 50

Kuartil Bawah : 60

Kuartil Tengah: 75

Kuartil Atas : 90

Nilai Max :100

4. Buat algoritma untuk menghitung hasil perkalian antara dua buah matriks

Masukan : nilai-nilai dalam dua buah matriks

Keluaran : hasil perkalian dua buah matriks

III. Kegiatan Praktikum

Pada saat praktikum, kerjakan tugas-tugas berikut ini:

1. Buat program untuk tugas prepraktikum 3!
2. Buat program untuk tugas prepraktikum 4!
3. Tugas tambahan (ditentukan oleh asisten praktikum)!

IV. Tugas Akhir

Buatlah algoritma dan terjemahkan dalam Bahasa Pascal untuk operasi invers matriks ordo N!

V. Catatan Praktikum

Sumber Pustaka:

1. Ifada N, **Diktat Mata Kuliah Algoritma Pemrograman**, Bangkalan: Jurusan Teknik Informatika Universitas Trunojoyo, 2005.
2. Munir R, **Algoritma dan Pemrograman dengan Pascal dan C edisi Kedua**, Bandung: Informatika, 2003.

Modul 6

Record

I. Teori

Record adalah elemen larik yang bertipe terstruktur. Dengan menggunakan tipe data *record*, beberapa item data yang masing-masing dapat mempunyai tipe data berbeda-beda dapat dikumpulkan. Masing-masing item data disebut dengan *field*. Jadi *record* terdiri dari kumpulan *field* yang dapat berbeda tipe. Biasanya suatu *record* berisi beberapa *field* untuk sebuah subyek tertentu.

Misalnya *record* TabelMhs dalam Gambar 6.1. adalah sebuah larik (sejumlah 100 elemen) yang elemennya menyatakan nilai ujian mata kuliah (MK) yang diambil seorang mahasiswa. Data (*field*) setiap mahasiswa adalah NIM (Nomor Induk Mahasiswa), nama mahasiswa, mata kuliah yang diambil, dan nilai mata kuliah tersebut.

Tabel Mhs

	NIM	NamaMhs	KodeMK	Nilai
1	0604100001	M. Khairuddin	FTC155	A
2	0604100002	Agastya	FTC165	B
3	0604100003	Budi Kusuma	FTC177	D
4	0604100004	Marina	FTC178	B
5	0604100005	Eliawati	FTC180	A
.				
.				
100	0604100100	Prasetyo	FTC185	C

Gambar 6.1. Struktur Record TabelMhs

II. Prepraktikum

Kerjakan tugas-tugas di bawah ini sebelum praktikum dimulai.

1. Sebutkan dan jelaskan keuntungan-keuntungan penggunaan *record*!
2. Kapan sebaiknya *record* digunakan? Jelaskan!
3. Buatlah algoritma untuk mengurutkan data berdasarkan nomor mahasiswa (NPM) dengan urutan dari yang terkecil ke yang terbesar atau Nama dengan urutan nama secara alfabet serta mempunyai kemampuan untuk menampilkan jumlah total mahasiswa sesuai dengan kriteria yang dimasukkan oleh *user*.

Misalnya: *user* ingin mengetahui jumlah mahasiswa yang mempunyai nilai A untuk mata kuliah X, maka algoritma akan menampilkan jumlah mahasiswa yang mempunyai nilai A untuk mata kuliah X .

Masukan : jumlah mahasiswa.

Selanjutnya memasukkan record data mahasiswa. Data (*field*) setiap mahasiswa adalah NPM (Nomor Pokok Mahasiswa), nama mahasiswa, mata kuliah yang diambil, dan nilai (Angka) mata kuliah tersebut. Untuk nilai huruf akan terisi secara otomatis sesuai range nilai pada grade nilai berikut nilai rata-ratanya.

Keluaran : hasil pengurutan data berdasarkan NPM atau Nama

Hasil ditampilkan dalam bentuk tabel seperti di bawah ini :

NPM	Nama	Kode	Nilai Angka	Nilai Huruf
Rata-rata				

III. Kegiatan Praktikum

Pada saat praktikum, kerjakan tugas-tugas berikut ini:

1. Buat program untuk tugas prepraktikum 3!
2. Tugas tambahan (ditentukan oleh asisten praktikum)!

IV. Tugas Akhir

Buatlah algoritma dan terjemahkan dalam bahasa pascal untuk memasukkan data belanja dan menampilkannya dalam bentuk nota Belanja. Total Harga, Total belanja, Total Bayar dan Total Kembali atau sisa didapat dengan melakukan proses perhitungan kemudian hasilnya ditampilkan dalam bentuk seperti contoh dibawah ini.

NOTA BELANJA
Toko Makmur
Jl. Raya Trunojoyo Telang Kamal Bangkalan

Tanggal Pembelian : 08-10-2012
Nama Pembeli : Ricky Nitinegoro
Alamat : Jl. Mawar No. 23 Bangkalan

No	Kode Barang	Nama Barang	Harga Satuan (Rp)	Jumlah	Harga x Jumlah (Rp)
1	A123	Barang A	15000	2	30000
2	B234	Barang B	20000	1	20000
3	C345	Barang C	25000	2	50000
4	D456	Barang D	30000	3	90000
5	E567	Barang E	35000	4	140000
TOTAL PEMBELIAN (Rp)					330000
TOTAL BAYAR (Rp)					350000
TOTAL KEMBALI (Rp)					20000

V. Catatan Praktikum

Sumber Pustaka :

- Noor Ifada, 2005. **Diktat Mata Kuliah Algoritma Pemrograman**, Hibah Kompetensi A1 Jurusan Teknik Informatika Universitas Trunojoyo.
- Rinaldi Munir, 2003. **Algoritma dan Pemrograman dengan Pascal dan C edisi Kedua**, Penerbit Informatika, Bandung.

Modul 7

Project Praktikum

I. Teori

A file contains data which is saved in the hard disk. You can only view a file from the hard disk by using an operating system. A File can contain text data which is used by word processors. Many text files are saved in the hard disk with the extensions: *.txt and *.doc.

The file with extension of *.txt, means that it is created by the Microsoft Notepad.

Whenever you use the Microsoft Notepad and save a text file, this saved file is created and written on to the hard disk. However, some programs have various formats on how to maintain the text - such as justification, font, font colour. So many files contain more data other than text data! The program itself uses various techniques to create a file in such a way that, when it is being read again it can read (using programming skills, etc..) if it is justified, its font style etc.

The program that is able to read the file of extension *.xxx, is called the viewer of that file.

Read from a File (file input)

Reading a file in pascal is very easy. Note that there are no reserved words in Pascal that are used to to read or write to a file. We used the 'usual words':
readln() and writeln();

Here's the technique of how to write / rewrite a text file

```
uses wincrt;

var
  namafile : string;
  vfile: text;
  input1:string;
begin
  namafile := 'd:\test1.txt';
  assign(vfile,namafile);
```

```

rewrite(vfile);
writeln('inputkan data. tekan x utk selesai');
repeat
  readln(input1);
  if input1 <> 'x' then
  begin
    writeln(vfile,input1);
  end;
until input1 = 'x';
close(vfile);
end.

```

If you run the script, at the end of program, you will found file created at **d:\test1.txt**. The file contains all data that have you entered before.

Here's the technique of how to read a text file

```

uses wincrt;

var
  namafile : string;
  vfile: text;
  output1:string;
begin
  namafile := 'd:\test1.txt';
  assign(vfile,namafile);
  reset(vfile);
  writeln('isi File:');
  repeat
    readln(vfile,output1);
    writeln(output1);
  until eof(vfile);
  close(vfile);
end.

```

II. Prepraktikum

Kerjakan tugas-tugas di bawah ini sebelum praktikum dimulai.

1. Buat Program yang menyimpan beberapa data dalam sebuah array
2. Buat Program untuk menyimpan isi Array ke File text
3. Buat Program Untuk membaca isi File text, dan simpan ke dalam array
4. Buat Program untuk menyimpan N data mahasiswa dengan field Nama, NIM, Nilai. Gunakan Record.

III. Kegiatan Praktikum

Pada saat praktikum, kerjakan tugas-tugas berikut ini:

1. Buat program untuk tugas prepraktikum no 1 s/d 4

IV. Tugas Akhir

Buat Program Prepraktikum no 4, dengan tambahan

1. Bisa melakukan pencarian data dari data yang telah diinputkan, pencarian berdasarkan nama
2. Data yang diinputkan bisa disimpan ke dalam file text
3. Data yang telah di simpan ke dalam file text bisa di Load ulang dan di masukkan ke dalam array record untuk di tampilkan

V. Catatan Praktikum

Sumber Pustaka :

1. Noor Ifada, 2005. **Diktat Mata Kuliah Algoritma Pemrograman**, Hibah Kompetisi A1 Jurusan Teknik Informatika Universitas Trunojoyo.
2. Rinaldi Munir, 2003. **Algoritma dan Pemrograman dengan Pascal dan C edisi Kedua**, Penerbit Informatika, Bandung.