

Materi I

TIPE ENUMERASI, TEKS DAN FILE

1. Tipe enumerasi dan sub jangkauan (sub range)
2. Pemrosesan Teks
3. File teks, Bertipe dan Tak Bertipe

Tipe Enumerasi

Enumerasi adalah salah satu tipe yang dibuat sendiri oleh pemrogram. Tipe ini digunakan pada variable yang nilai kemungkinannya terbatas (tidak begitu banyak). Nilainya dinyatakan dengan nama-nama pengenalan, sehingga diharapkan lebih memberikan kejelasan, daripada kalau memakai kode seperti 0,1,2.

Contoh :

```
TYPE
    SistemOperasi = (UNIX, MS_DOS, OS2, WindowsNT, WindowsXP)
```

Statement di atas sama dengan deklarasi berikut :

```
CONST
    UNIX=0;
    MS_DOS=1;
    :
    WindowsNT=4;
```

Fungsi-fungsi yang digunakan :

Ord (X) : memperoleh nilai ordinal dari X
Pred (X) : memperoleh nilai pendahulu dari (predecessor) dari X
Succ(X) : memperoleh nilai penerus (successor) dari X

Dari tipe SistemOperasi di atas :

- nilai ordinal UNIX adalah 0
- nilai ordinal WindowsXP adalah 4

Fungsi ord, pred, succ dapat dikenakan pada semua tipe ordinal (mis : Boolean, char, integer).

Contoh :

```
Program Enum1;
TYPE
    SistemOperasi = (UNIX, MS_DOS, OS2, WindowsNT, WindowsXP);
Var
    SisOp : SistemOperasi;
```

```

{deskripsi}
begin
  for SisOp := UNIX to WindowsXP do
    writeln (Ord (SisOp));
  end;
end.

Hasil eksekusi :
0
1
2
3
4

```

Contoh penggunaan succ :

Program Enum2;

```

TYPE
  SistemOperasi = (UNIX, MS_DOS, OS2, WindowsNT, WindowsXP);
Var
  SisOp : SistemOperasi;
  Selesai : Boolean;

{deskripsi}
begin
  SisOp:=UNIX;
  Selesai := false;
  Repeat
    Writeln (ord(SisOp));
    If SisOp <> WindowsXP then
      SisOp := succ (SisOp)
    Else
      Selesai := true;
  Until selesai;
End.

```

Hasil eksekusi : sama dengan contoh pertama.

Bagaimana jika ingin menggunakan fungsi Pred ?

Berikut contoh dengan menggunakan case of :

Program Enum2;

```

TYPE
  SistemOperasi = (UNIX, MS_DOS, OS2, WindowsNT, WindowsXP);
Var
  SisOp : SistemOperasi;

{deskripsi}
begin
  for SisOp := UNIX to WindowsXP do
    begin

```

```

        case SisOp of
            UNIX : writeln ('UNIX');
                :
            windowsXP: writeln ('WindowsXP');
        end;
    end;
End.

```

Contoh lain penggunaan enumerasi :

```

Program contoh_ORD_X;
{mencoba fungsi ord pada tipe enumerasi}

Uses wincrt;

{deklarasi}
Type
    jurusan = (ti_s1, si_s1, mi_d3, tk_d3); {tipe enumerasi}

Var
    kodejur : jurusan;

{deskripsi}
Begin
    clrscr;
    writeln('tipe jurusan = (ti_s1, si_s1, mi_d3, tk_d3)');
    for kodejur := ti_s1 to tk_d3 do
        writeln ('ord(jurusan) adalah : ',ord(kodejur));
    End.

```

Sub Jangkauan (sub range)

Sintaks :

```

TYPE nama_tipe = nilai_terkecil .. nilai_terbesar

```

Contoh :

```

TYPE
    NilaiBulan : 1 .. 12;
    HurufKapital: 'A' .. 'Z';

TYPE
    NamaBulan : (Jan,Feb, ... , Des)      {enumerasi}
    MusimDingin : Jan .. Mar              {subrange}

```

Contoh penggunaan tipe subrange :

```

program aplikasi_tipe enumerasi_dan_tipe_subrange;
{Menuliskan nama bulan yang ingin ditampilkan}

Uses wincrt;

{deklarasi}

```

```

Type
    tipebulan = (jan, feb, mar, apr, mei, jun, jul, agu, sep, okt, nop, des);
    {tipe enumerasi}

Var
    n : 1 .. 12; {tipe subrange}
    awalbulan, namabulan : tipebulan;

{deskripsi}
Begin
    clrscr;
    writeln('program mencetak nama bulan dan mulan bulan ke-n');
    writeln('-----');
    writeln;
    write('bulan yang ingin ditampilkan dari bulan ke (1..12) : ');
    readln(n);
    writeln;
    awalbulan := tipebulan(n-1);

    for namabulan := awalbulan to des do
    case namabulan of
        jan : writeln ('januari');
        feb : writeln ('februari');
        mar : writeln ('maret');
        apr : writeln ('april');
        mei : writeln ('mei');
        jun : writeln ('juni');
        jul : writeln ('juli');
        agu : writeln ('agustus');
        sep : writeln ('september');
        okt : writeln ('oktober');
        nop : writeln ('nopember');
        des : writeln ('desember');
    end;
End.

```

Pemrosesan Teks

Mengakses data yang tersusun secara beruntun sering dijumpai dalam pemrograman. Contoh pengaksesan beruntun adalah seperti gambar slide yang diproyeksikan oleh slide projector, dimana gambar diakses satu persatu secara beruntun sehingga terlihat menjadi suatu gambar bergerak yang teratur. Kumpulan gambar slide tersebut dianggap sebagai sekumpulan data yang tersusun secara beruntun.

Di dalam pemrograman, contoh data yang tersusun secara beruntun antara lain :

- array (larik)
- arsip beruntun (sequential file)
- arsip teks (text file)
- Senarai berkait (linked list)

Pada Pascal ada 3 jenis file, yaitu :

- File bertipe
- File teks

- File tidak Bertipe

Yang akan dipakai pada pemrosesan teks adalah file teks.

Prosedur standar pada operasi file teks (dan sebagian besar berlaku juga untuk file bertipe):

Assign : inisialisasi nama file

Rewrite : menciptakan file untuk ditulis

Reset : membuka file untuk dibaca

Close : menutup file

Write, writeln : tulis file

Read, readln : baca file

EOF (end of file) : fungsi untuk mengetahui akhir dari file, mengembalikan nilai Boolean :

Function EOF(NamaVarFile) : Boolean

Append : menambah data

Di bagian ini, akan dibahas mengenai file / arsip teks.

File teks adalah – selanjutnya disebut **teks** - adalah :

- file yang terdiri atas deretan karakter. Deretan tersebut dapat membentuk sebuah “kata”. Yang dimaksud dengan kata adalah kelompok karakter yang dipisahkan dengan kelompok lain dengan satu atau lebih spasi.
- File teks hanya bisa diakses secara sequensial, tidak bisa secara random / acak. (bedanya dengan file bertipe).
- Pada implementasinya, di dalam teks ditambahkan akhir-baris (*end-of-line* - terdiri dari karakter *line-feed*) yang membedakan sebuah baris teks dengan baris lainnya
- Isi file teks, biasa diedit secara langsung dengan menggunakan editor teks.

Contoh teks adalah sumber program Pascal (*.pas), atau C (*.C), file README.TXT, AUTOEXEC.BAT pada sistem operasi DOS, dan lain-lain. Teks dapat dibuat dengan :

- pengolah kata teks (notepad, wordpad)
- kompilator bahasa pemrograman (Turbo Pascal, Turbo C)
- pengolah kata *Microsoft Word*

Cakupan pembahasan file teks :

- Tulis dan baca file teks
- Hitung karakter
- Hitung 2 karakter
- Hitung kata
- Cari kata
- Penggunaan end of file (eof)

Suatu teks P didefinisikan di dalam bagian DEKLARASI sebagai berikut :

DEKLARASI

P : text {P adalah variabel teks}

Tanda akhir Teks

Pemrosesan Teks yang utama adalah proses pembacaan. Akhir teks ditandai dengan sebuah karakter khusus. Bila pembacaan teks bertemu dengan karakter khusus tersebut, maka proses pembacaan teks dihentikan.

Disini akan didefinisikan karakter titik (‘.’) sebagai akhir teks. Pada aplikasinya dapat digunakan karakter lain, seperti ‘#’, ‘\$’ dan sebagainya.

Definisi Teks Kosong

Sebuah teks mungkin saja kosong. Didefinisikan, teks kosong adalah teks yang hanya berisi karakter titik dan jumlah unsurnya nol.

Operasi atau proses yang dilakukan pada file teks adalah pembacaan dan penulisan teks.

Pembacaan Teks.

Pembacaan teks dimulai dari awal teks (dari kiri ke kanan). Posisi saat tertentu pada pembacaan teks ditunjukkan oleh *pointer* dan karakter yang dibaca pada posisi tersebut dinyatakan di dalam variabel C. Nama variabel didefinisikan di dalam bagian DEKLARASI program utama, sehingga dikenal di seluruh bagian algoritma termasuk prosedur dan fungsi yang dipakai oleh algoritma.

DEKLARASI

C: char {karakter yang sedang ditunjuk oleh pointer baca}

Didefinisikan procedure RESET_TEKS, yaitu prosedur universal untuk teks. Procedure ini menyebabkan *pointer* akan menunjuk pada karakter pertama di teks. Jika kosong, pointer akan menunjuk karakter titik. Dapat dilihat penggunaannya pada contoh di bawah.

Berikut contoh algoritma dan program Pembacaan teks (= pita karakter) sambil menghitung jumlah karakter, tidak termasuk titik :

Algoritma	Program
<p>Algoritma Program hitung_karakter {menghitung karakter pada file teks}</p> <p>Deklarasi :</p> <p> P : text { TANDA = '.' }</p> <p> C : char</p> <p>procedure reset_teks {menyiapkan teks pada posisi awal K.Awal : sembarang K.Akhir : pointer menunjuk pada karakter pertama di dalam teks. Akibat pemanggilan prosedur ini, pointer menunjuk ke karakter pertama teks. Karakter yang ditunjuk</p>	<p>Program hitung_karakter; {menghitung karakter pada file teks} uses wincrt;</p> <p>{Deklarasi} var P : text; { TANDA = '.' } C : char; namafile : string;</p> <p>procedure reset_teks;</p> <p>begin Write(' ketikkan nama file (*.txt) : '); readln(namafile); assign (P,namafile); reset (P);{membuka file untuk pembacaan}</p>

<pre> mungkin '.'} assign (P) reset (P) procedure hitung_kar {menghitung banyaknya karakter di dalam teks K.Awal : sembarang K.Akhir : banyaknya karakter di dalam teks diketahui} Deklarasi : jkar : integer Deskripsi : reset_teks jkar ← 0 read(P,C) while C <> '.' do jkar ← jkar + 1 read(P,C) end while Close (P) Write (jkar) DESKIPSI write('MENGHITUNG KARAKTER PADA FILE TEXT') hitung_kar </pre>	<pre> end; procedure hitung_kar; {menghitung banyaknya karakter di dalam teks K.Awal : sembarang K.Akhir : banyaknya karakter di dalam teks diketahui} var jkar : integer; begin reset_teks; jkar := 0; read(P,C); while C <> '.' do begin jkar := jkar + 1; read(fKar,C); end; Close (P); {menutup file} writeln('jumlah karakter pada ',namafile,' : ',jkar:5); end; {program utama} Begin writeln('MENGHITUNG KARAKTER PADA FILE TEXT'); hitung_kar; end. </pre>
--	---

Penulisan Teks.

Selain dibaca, file teks juga bisa ditulis. Perbedaannya hanya pada Procedure RESET_TEKS, digunakan :

```

Assign (P)
Rewrite (P).

```

Fungsi Rewrite menandai bahwa file teks yang didefinisikan adalah untuk ditulis. Berikut adalah contoh program penulisan teks, dimana diterima satu karakter dari papan kunci, kemudian ditulis ke file, dan berulang samapai dimasukkan karakter titik.

Algoritma	Program
<pre> Algoritma tulis_kar; {merekam / menulis karakter ke file teks} {Deklarasi} var P : text { TANDA = '.' } C : char </pre>	<pre> Program tulis_kar; {merekam / menulis karakter ke file teks} uses wincrt; {Deklarasi} var fKar : text; { TANDA = '.' } C : char; </pre>

<pre> procedure reset_teks2; {deskripsi} read(namafile) assign (P,namafile) rewrite (P){membuka file untuk penulisan} Procedure tulis_karakter {deklarasi} {tidak ada } {deskripsi} reset_teks2; write ('jika selesai ketik "." :') read(C) while C <> '.' do write(P,C) read(C) end while write(P,C) {end data} Close (P) {menutup file} {deskripsi } Begin write('MENULIS KARAKTER KE FILE TEXT') tulis_karakter; end. </pre>	<pre> namafile : string; procedure reset_teks2; begin Write(' ketikkan nama file (*.txt) : '); readln(namafile); assign (P,namafile); rewrite (P){membuka file untuk penulisan} end; Procedure tulis_karakter; Var {tidak ada } begin reset_teks2; writeln('masukkan data karakter, jika selesai ketik "." :'); write('ketik sebuah karakter : '); readln(C); while C <> '.' do begin write(P,C); write('ketik sebuah karakter : '); end; readln(C); write(P,C); {end data} Close (P); {menutup file} writeln('penulisan selesai'); end; {deskripsi-program utama} Begin writeln('MENULIS KARAKTER KE FILE TEXT'); tulis_karakter; end. </pre>
--	--

Pengembangan Pembacaan Teks

Berikut adalah program menghitung banyak karakter ‘G’, dengan mengacu algoritma menghitung karakter pada contoh sebelumnya. Perhatikan prosedur hitungnya.

```

Program hitung_banyak_karakter_G;
{menghitung karakter G pada file teks}
uses wincrt;

{Deklarasi}
var
  fKar      : text;      { TANDA = '.' }
  C         : char;
  namafile  : string;
  jkar      : integer;

procedure reset_teks;
{menyiapkan teks pada posisi awal}
K.Awal    : sembarang
K.Akhir   : pointer menunjuk pada karakter pertama di dalam teks.
           Akibat pemanggilan prosedur ini,pointer menunjuk ke karakter

```



```

        pertama teks. Karakter yang ditunjuk mungkin '.'}

begin
    Write(' ketikkan nama file (*.txt) : '); readln(namafile);
    assign (fKar,namafile);
    reset (fKar); {membuka file untuk pembacaan}
end;

procedure hitung_kar_G(var jkar : integer);
{menghitung banyaknya karakter G di dalam teks
K.Awal  : sembarang
K.Akhir : banyaknya karakter G di dalam teks diketahui}

begin
    reset_teks;
    read(fKar,C);
    if C = '.' then
        jkar := 999
    else
        begin
            jkar := 0;
            repeat
                begin
                    if C = 'G' then
                        jkar := jkar + 1;
                    {end if}
                    read(fKar,C);
                end
            until (C = '.');
        end;
    Close (fKar); {menutup file}
end;

{deskripsi}
Begin
    writeln('MENGHITUNG KARAKTER "G" PADA FILE TEXT');
    hitung_kar_G (jkar);
    if jkar = 999 then
        writeln (namafile,' adalah teks kosong')
    else
        writeln('jumlah karakter G pada ',namafile,' : ',jkar);
end.

```

Berikut adalah program menghitung banyak suku kata ‘AR’, dengan mengacu algoritma menghitung karakter pada contoh sebelumnya. Perhatikan prosedur hitungnya.

```

Program hitung_banyak_AR;
{menghitung "AR pada file teks}
uses wincrt;

{Deklarasi}
var
    fKar      : text;      { TANDA = '.'}
    C         : char;
    namafile  : string;
    jAR       : integer;

procedure reset_teks;
{menyiapkan teks pada posisi awal
K.Awal  : sembarang
K.Akhir : pointer menunjuk pada karakter pertama di dalam teks.
        Akibat pemanggilan prosedur ini,pointer menunjuk ke karakter

```

```

        pertama teks. Karakter yang ditunjuk mungkin '.'}

begin
    Write(' ketikkan nama file (*.txt) : '); readln(namafile);
    assign (fKar,namafile);
    reset (fKar); {membuka file untuk pembacaan}
end;

procedure hitung_teks_AR(var jAR : integer);
{menghitung banyaknya 'AR' di dalam teks
K.Awal  : sembarang
K.Akhir : banyaknya 'AR' di dalam teks diketahui}

begin
    reset_teks;
    read(fkar,C);
    if C = '.' then    {teks kosong}
        jAR := 999
    else
        begin
            jAR := 0;
            repeat
                begin
                    while ( C<>'A') and (C<>'.') do
                        read(fkar,C);
                    {end while}

                    if C='A' then
                        begin
                            read(fKar,C);
                            if C='R' then
                                jAR := jAR + 1;
                            end;
                        {end if}

                    end
                until (C = '.');
            end;
            Close (fKar);    {menutup file}
        end;
end;

{deskripsi}
Begin
    writeln('MENGHITUNG "AR" PADA FILE TEXT');
    hitung_teks_AR (jAR);
    if jAR = 999 then
        writeln (namafile,' adalah teks kosong')
    else
        writeln('jumlah teks "AR" pada ',namafile,' : ',jAR);
end.

```

Berikut adalah cara lain untuk menghitung suku kata ‘AR’ :

```

procedure hitung_teks_AR2(var jAR : integer);
{menghitung banyaknya 'AR' di dalam teks
K.Awal  : sembarang
K.Akhir : banyaknya 'AR' di dalam teks diketahui}

var
    C_before : char;

begin

```

```

reset_teks;
read(fkar,C);
if C = '.' then {teks kosong}
    jAR := 999
else {C <> '.'}
begin
    jAR := 0;
    C_before := ' ';
    repeat
    begin
        if (C_before = 'A') and (C='R') then
            jAR := jAR + 1;
        {end if}
        C_before := C;
        read(fkar,C);
    end
    until (C = '.');
    Close (fKar); {menutup file}
end; {end if}
end;

```

Coba buat program lengkap nya !

Berikut adalah contoh program menghitung jumlah kata yang terdapat dalam suatu teks.

Perhatikan procedure nya :

Program hitung_kata;

{menghitung jumlah kata pada teks}

uses wincrt;

{Deklarasi}

var

```

fKar      : text;      { TANDA = '.' }
C         : char;
namafile  : string;
jKata     : integer;

```

procedure reset_teks;

{menyiapkan teks pada posisi awal}

K.Awal : sembarang

K.Akhir : pointer menunjuk pada karakter pertama di dalam teks.

Akibat pemanggilan prosedur ini, pointer menunjuk ke karakter pertama teks. Karakter yang ditunjuk mungkin '.'}

begin

Write(' ketikkan nama file (*.txt) : '); readln(namafile);

assign (fKar,namafile);

reset (fKar); *{membuka file untuk pembacaan}*

end;

procedure hitung_banyak_kata(var nKata : integer);

{menghitung banyak kata di dalam teks}

K.Awal : nKata belum terdefinisi nilainya

K.Akhir : banyaknya kata di dalam teks diketahui}

var

C_before : char;

begin

nKata := 0;

C_before := ' ';

reset_teks;

read(fkar,C);

```

while (C<>'.' ) do
begin
  if (C_before<>' ') and (C=' ') then
    nKata:= nKata + 1;
  {end if}
  C_before := C;
  read(fkar,C);
end;

{kata terakhir diproses khusus}
if (C_before<>' ') and (C='.') then
  nKata:= nKata + 1;
{end if}
Close (fKar); {menutup file}
end;

{deskripsi}
Begin
  writeln('MENGHITUNG JUMLAH KATA PADA TEKS');
  hitung_banyak_kata (jKata);
  if jKata = 0 then
    writeln (namafile,' adalah teks kosong')
  else
    writeln('jumlah kata pada ',namafile,' : ',jKata);
  end.
end.

```

Berikut adalah contoh program menghitung jumlah kata dengan cara lain, perhatikan procedure menghitung kata dan procedure abaikan spasi :

```

Program hitung_kata2;
{menghitung jumlah kata pada teks}
uses wincrt;

{Deklarasi}
var
  fKar      : text;      { TANDA = '.' }
  C         : char;
  namafile  : string;
  jKata     : integer;

procedure reset_teks;
{menyiapkan teks pada posisi awal}
K.Awal      : sembarang
K.Akhir     : pointer menunjuk pada karakter pertama di dalam teks.
              Akibat pemanggilan prosedur ini,pointer menunjuk ke karakter
              pertama teks. Karakter yang ditunjuk mungkin '.'}

begin
  Write(' ketikkan nama file (*.txt) : '); readln(namafile);
  assign (fKar,namafile);
  reset (fKar); {membuka file untuk pembacaan}
end;

procedure abaikan_spasi;
{membaca deretan spasi dan mengabaikannya}
K. Awal     : sembarang
K. Akhir    : karakter bukan spasi ditemukan}

{deklarasi}
{tidak ada}

{deskripsi}

```

```

begin
  {selama ada spasi dan belum titik, baca terus karakter di teks}
  while (C=' ') and (C<>'.') do
    begin
      read(fKar,C);
    end;
  {C<>' ' or C='.'}
end;

```

```

procedure hitung_banyak_kata2(var nKata : integer);
{menghitung banyak kata di dalam teks
K.Awal  : nKata belum terdefinisi nilainya
K.Akhir : banyaknya kata di dalam teks diketahui}

```

```

{deskripsi}

```

```

begin
  reset_teks;
  read(fkar,C);
  nKata := 0;
  while (C<>'.') do
    begin
      abaikan_spasi;
      {baca terus pita selama belum spasi atau belum '.'}
      while (C<>' ') and (C<>'.') do
        read(fkar,C);
      {end while}
      nKata:= nKata + 1;

      abaikan_spasi;
    end; {C='.'}
  end;

```

```

{deskripsi}

```

```

Begin
  writeln('MENGHITUNG JUMLAH KATA PADA TEKS, CARA 2');
  hitung_banyak_kata2 (jKata);
  if jKata = 0 then
    writeln (namafile,' adalah teks kosong')
  else
    writeln('jumlah kata pada ',namafile,' : ',jKata);
end.

```

Berikut adalah contoh algoritma procedure cari kata :

```

procedure cari_kata (s:string; var ketemu:boolean)
{mencari kata tertentu di dalam teks
K.Awal  : s sudah berisi kata yang akan dicari
K.Akhir : ketemu bernilai true jika kata di dalam parameter s ditemukan
         false jika tidak ditemukan}

```

```

{deklarasi}

```

```

var
  kata : string
  pjkt : integer
  i     : integer

```

```

procedure abaikan_spasi
{membaca deretan spasi dan mengabaikannya

```

```

{deskripsi cari kata}

```

```

begin
  reset_teks
  read(fkar,C)
  ketemu ← false
  kata ← ''
  while (C<>'.' ) and (not ketemu) do begin
    abaikan_spasi
    {baca terus teks selama belum mencapai spasi atau belum '.'}
    pjkt ← 0
    while (C<>' ') and (C<>'.' ) do
      kata ← kata + C
      pjkt ← pjkt+1
      read(fkar,C)
    end while
    {C<>' ' or C='.', satu kata ditemukan}

    {bandingkan kata}
    if kata = s then
      ketemu ← true
    else
      ketemu ← false
    end if
    abaikan_spasi
    kata ← ''
  end while {C='.' or ketemu}

```

Algoritma di atas diaplikasikan pada program mencari kata tertentu yang terdapat dalam suatu teks. Perhatikan procedure cari_kata nya:

```

Program cari_kata_dalam_teks;
{program pencarian kata tertentu dalam teks}
uses wincrt;

{Deklarasi}
var
  fKar      : text;      { TANDA = '.' }
  C         : char;
  namafile  : string;
  word      : string;
  ketemu    : boolean;

procedure reset_teks;
{menyiapkan teks pada posisi awal}
K.Awal    : sembarang
K.Akhir   : pointer menunjuk pada karakter pertama di dalam teks.
           Akibat pemanggilan prosedur ini,pointer menunjuk ke karakter
           pertama teks. Karakter yang ditunjuk mungkin '.'}

begin
  Write(' ketikkan nama file (*.txt) : '); readln(namafile);
  assign (fKar,namafile);
  reset (fKar); {membuka file untuk pembacaan}
end;

procedure cari_kata (s:string; var ketemu:boolean);
{mencari kata tertentu di dalam teks}
K.Awal    : s sudah berisi kata yang akan dicari
K.Akhir   : ketemu bernilai true jika kata di dalam parameter s ditemukan
           false jika tidak ditemukan}

{deklarasi}

```

```

var
    kata : string;
    pjkt : integer;
    i     : integer;

procedure abaikan_spasi;
{membaca deretan spasi dan mengabaikannya
K. Awal  : sembarang
K. Akhir : karakter bukan spasi ditemukan}

{deklarasi}

{deskripsi}
begin
    {selama ada spasi dan belum titik, baca terus karakter di teks}
    while (C=' ') and (C<>'.' ) do
    begin
        read(fKar,C);
    end;
    {C<>' ' or C='.'}
end;

{deskripsi cari kata}
begin
    reset_teks;
    read(fkar,C);
    ketemu := false;
    kata := '';
    while (C<>'.' ) and (not ketemu) do
    begin
        abaikan_spasi;
        {baca terus teks selama belum mencapai spasi atau belum '.'}
        pjkt :=0;
        while (C<>' ') and (C<>'.' ) do
        begin
            kata := kata + C;
            pjkt := pjkt+1;
            read(fkar,C);
        end;
        {C<>' ' or C='.' , satu kata ditemukan}

        {bandingkan kata}
        if pjkt = length(kata) then
        begin
            i := 1;
            while (s[i] = kata[i]) and (i <= pjkt) do
            begin
                i:= i+1;
            end; {while}
            i:=i-1;
            if i < pjkt then
                ketemu := false
            else {i=pjkt}
                ketemu := true;
            {end if}
        end
        else
            ketemu := false;

        abaikan_spasi;
        kata := '';
    end; {C='.' or ketemu}
end;{cari kata}

```

```

{program utama}
Begin
  writeln('MENCARI KATA PADA TEKS');
  write('ketik kata yang dicari : '); readln(word);

  cari_kata (word,ketemu);
  if ketemu then
    writeln ('kata ',word,' ditemukan')
  else
    writeln ('kata ',word,' tidak ditemukan');
end.

```

Proses file teks menggunakan end of file (eof).

Perhatikan contoh-contoh program di bawah yang menggunakan file teks. Operasi / proses yang dilakukan adalah : tulis – baca – tambah. Contoh pertama adalah menulis ke file dan menandai akhir file tidak dengan karakter tertentu. Kemudian file dibaca dan pendeteksian akhir file menggunakan perintah eof.

tulis	baca
<pre> Program tulis_file_teks; {menulis data pada file teks } uses wincrt; {Deklarasi} var fhuruf : text; {deskripsi} Begin clrscr; writeln('MENULIS DATA KE FILE TEKS'); assign (fhuruf,'HURUF.TXT'); rewrite (fhuruf); {menciptakan file} {merekam tiga buah huruf} writeln (fhuruf,'A'); writeln (fhuruf,'B'); writeln (fhuruf,'C'); Close (fhuruf); {menutup file} writeln('buka file "HURUF.TXT" pada windows explorer di subdirektori kerja'); end. </pre>	<pre> Program baca_file_teks_dgn_EOF; {membaca data pada file teks} uses wincrt; {Deklarasi} var fhuruf : text; baris : string; {deskripsi} Begin clrscr; writeln('MEMBACA DATA DARI FILE TEKS'); writeln('isi dari file yang dibaca :'); assign (fhuruf,'HURUF.TXT'); reset (fhuruf); {membuka file} while not eof(fhuruf) do begin readln(fhuruf, baris); writeln(baris); end; Close (fhuruf); {menutup file} end. </pre>

File teks juga dapat ditambah isinya dengan menggunakan perintah ‘Append’, seperti contoh berikut :

```

Program tambah_file_teks;
{menambah data pada file teks}

uses wincrt;

{Deklarasi}

```



```

var
    fhuruf : text;
    baris   : string;

{deskripsi}
Begin
    clrscr;
    writeln('MENAMBAH DATA KE FILE TEKS');
    assign (fhuruf,'HURUF.TXT');
    append (fhuruf); {menambah data ke file teks}

    {merekam tiga buah huruf}
    writeln (fhuruf,'DEFGHIJKL');

    Close (fhuruf); {menutup file}
    writeln('buka file "HURUF.TXT" pada windows explorer di subdirektori kerja');
end.

```

File Bertipe

File bertipe, terdiri dari 2 :

- Urut (sequential access)
- Acak (Random access)
- Urutan proses :
 1. Menyebutkan variable file dan nama file
 2. Membuka file
 3. Melaksanakan pengaksesan file
 4. Menutup file
- Prosedur standar pada operasi file :
 - Assign
 - Rewrite
 - Reset
 - Close
 - EOF (end of file)
 - Write
 - Read
 - FileSize
 - FilePos
- Operasi-operasi yang digunakan :
 - Tambah / rekam / tulis data
 - Lihat / tampil / baca data
 - Hapus data
 - Koreksi / ubah data

File dapat bertipe : integer, real, karakter atau record.

Contoh Deklarasi tipe dasar :

DEKLARASI

NamaFile: SeqFile of TipeRekaman

Dimana TipeRekaman bisa integer, real, karakter, record.

Jika menggunakan tipe baru :

DEKLARASI

Type FileBeruntun : SeqFile of TipeRekaman

NamaFile : FileBeruntun

Dimana TipeRekaman bisa integer, real, karakter, record.

Perintah-perintah pada file yang umum digunakan :

Membuka file untuk dibaca atau ditulis :

OPEN (NamaFile, Kode)

Contoh :

Open (mhs,1) → buka file MHS untuk pembacaan

Open (mhs,2) → buka file MHS untuk penulisan

Menulis data ke file :

Fwrite (NamaFile, NamaRek)

Contoh :

Fwrite (MHS,MhsDat) → menulis data di variabel MhsDat ke file MHS

Membaca data dari file :

Fread (NamaFile, NamaRek)

Contoh :

Fread (MHS,MhsDat) → membaca data dari file MHS dan ditampung di variabel MhsDat

Menutup file :

Close (NamaFile)

Contoh :

Close (MHS) → menutup file MHS

Akhir suatu file dapat digunakan tanda yang didefinisikan sendiri, misal :

Karakter : \. '

Real : 999.99

Integer : 9999

String : 'xxxxxx'

Dapat digunakan juga : perintah end of file (**eof**)

File Integer

Berikut adalah contoh algoritma dan program dari file bertipe **integer**. Contoh pertama adalah proses menulis file, kemudian membaca file.

Tulis File (perhatian tanda yang dipakai untuk menandai akhir file) :

Algoritma	Program
<pre>Algoritma tulis_file_bil_bulat {menulis data pada file bertipe integer} {Deklarasi} type ArsipBil = file of integer {TANDA = 9999} fBil : ArsipBil bil,I,jumdat : integer {deskripsi} write('MENULIS DATA BIL. INTEGER KE FILE RUNTUNAN') read(jumdat) Open (fBil,1) {membuka file utk penulisan} for i ← 1 to jumdat do read(bil) fwrite(fBil, bil) end for bil ← 9999 {end data} fwrite(fBil,bil) Close (fBil) {menutup file}</pre>	<pre>Program tulis_file_bil_bulat; {menulis data pada file bertipe integer} uses wincrt; {Deklarasi} type ArsipBil = file of integer;{TANDA = 9999} var fBil : ArsipBil; bil,I,jumdat : integer; namafile : string; {deskripsi} Begin writeln('MENULIS DATA BIL. INTEGER KE FILE RUNTUNAN'); Write('ketikkan nama file (*.txt): '); readln(namafile); Write(' jumlah data : '); readln(jumdat); assign (fBil,namafile); rewrite(fBil);{membuka file utk penulisan} writeln('masukkan data :'); for i := 1 to jumdat do begin write('data ',i:2,' : '); readln(bil); write(fBil, bil); end; bil := 9999; {end data} write(fBil,bil); Close (fBil); {menutup file} end.</pre>

Berikut adalah penulisan file cara ke 2 :

Algoritma	Program
<pre>Algoritma tulis_file_bil_bulat2 {menulis data pada file bertipe integer} {Deklarasi}</pre>	<p>Bagaimana Programnya ?</p>

<pre> type ArsipBil = file of integer{TANDA = 9999} fBil : ArsipBil bil,i : integer {deskripsi} writeln('MENULIS DATA BIL. INTEGER KE FILE RUNTUNAN') open(fBil,1){membuka file utk penulisan} write('masukkan data, jika selesai ketik 9999 :') i ← 1 write('data ',i) read(bil) while bil <> 9999 do fwrite(fBil, bil) i ← i+1 write('data ',i) read(bil) end while fwrite(fBil, bil) {tuliskan / rekam TANDA} Close (fBil) {menutup file} </pre>	
--	--

Membaca File (perhatian tanda yang dipakai untuk menandai akhir file) :

Algoritma	Program
<pre> Algoritma baca_file_bil_bulat {membaca data pada file bertipe integer} {Deklarasi} type ArsipBil = file of integer{TANDA = 9999} fBil : ArsipBil bil,i,jumdat : integer {deskripsi} write('MEMBACA DATA BIL. INTEGER DARI FILE RUNTUNAN') Open(fBil,2){membuka file utk pembacaan} fread(fBil, bil) while bil <> 9999 do fread(fBil, bil) end while Close (fBil) {menutup file} </pre>	<pre> Program baca_file_bil_bulat; {membaca data pada file bertipe integer} uses wincrt; {Deklarasi} type ArsipBil = file of integer;{TANDA = 9999} var fBil : ArsipBil; bil,I,jumdat : integer; namafile : string; {deskripsi} Begin writeln('MEMBACA DATA BIL. INTEGER DARI FILE RUNTUNAN'); Write(' ketikkan nama file (*.txt) : '); readln(namafile); assign (fBil,namafile); reset (fBil);{membuka file utk pembacaan} read(fBil, bil); while bil <> 9999 do begin writeln (bil:2,' '); read(fBil, bil); end; Close (fBil); {menutup file} end. </pre>

Contoh penggunaan akhir file menggunakan **eof** :

Tulis	Baca
<pre> Program tulis_file; uses wincrt; {Deklarasi} type ArsipInt = file of integer; var fInt : ArsipInt; namafile : string; bil : integer; {deskripsi} begin writeln ('MENULIS DATA BIL. INTEGER KE FILE BERUNTUN'); write('ketik nama file:'); readln(namafile); assign(fInt,namafile); rewrite(fInt); {ciptakan file} bil := 10; write(fInt,bil); {rekam data} bil := 20; write(fInt,bil); {rekam data} bil := 30; write(fInt,bil); {rekam data} close(fInt); writeln('selesai'); end. </pre>	<pre> Program baca_file; uses wincrt; {Deklarasi} type ArsipInt = file of integer; var fInt : ArsipInt; namafile : string; bil : integer; {Deskripsi} begin writeln ('MEMBACA DATA BIL. INTEGER DARI FILE BERUNTUN'); write('ketik nama file : '); readln(namafile); assign(fInt,namafile); reset(fInt); {membuka file untuk dibaca} {baca dari komponen 0 hingga akhir} while not Eof(fInt) do begin read(fInt,bil); write(bil:8); end; {while} close(fInt); end. </pre>

Berikut adalah contoh proses lainnya : ubah, hapus

Ubah	Hapus
<pre> Program ubah_data_file; {mengubah data pada file bertipe} uses wincrt; {Deklarasi} type ArsipInt = file of integer; var fInt : ArsipInt; namafile : string; bil : integer; posisi : integer; {Deskripsi} begin writeln ('MENAMBAH DATA BIL. INTEGER KE FILE BERUNTUN'); write('ketik nama file : '); readln(namafile); assign(fInt,namafile); reset(fInt); {membuka file tanpa menghapus} </pre>	<pre> Program hapus_data_file; {menghapus data pada file bertipe} uses wincrt; {Deklarasi} type ArsipInt = file of integer; var fInt : ArsipInt; namafile : string; bil : integer; posisi : integer; procedure tampilkan_data(ket:string; var nmfile:ArsipInt); var bil : integer; begin writeln(ket); seek(fInt,0); { ke komponen 0} while not Eof(fInt) do </pre>

<pre> randomize; {mengatur pembangkit bilangan acak} posisi := random(FileSize(fInt)); bil := random(MaxInt); {ke posisi yang akan diubah secara random} seek(fInt,posisi); {rekam data pengganti} write(fInt,bil); writeln('komponen ke', posisi:4,' diganti menjadi ', bil:5); close(fInt); {tutup file} end.</pre>	<pre> begin read(fInt,bil); writeln(bil); end; writeln; end; {deskripsi} begin writeln ('MEMOTONG DATA BIL. INTEGER DARI FILE BERUNTUN'); write('ketik nama file : '); readln(namafile); assign(fInt,namafile); reset(fInt);{membuka file tanpa menghapus} tampilkan_data('Isi semula : ',fInt); seek(fInt,2); {ke komponen ke 2} truncate(fInt); tampilkan_data('Isi sekarang : ',fInt); close(fInt); {tutup file} end.</pre>
--	---

Perhatikan, ada perintah lain yang digunakan : FileSize, seek, truncate.

File Karakter

Berikut contoh Menulis File :

Algoritma	Program
<pre> Algoritma tulis_file_karakter {menulis data pada file bertipe karakter} {Deklarasi} type ArsipKar = file of char { TANDA = '.' } var fKar : ArsipKar; kar : char; i : integer; jumdat : integer; {deskripsi} writeln('MENULIS DATA KARAKTER KE FILE RUNTUNAN') read(jumdat) Open(fKar,1){membuka file untuk penulisan} write('masukkan data :')</pre>	<pre> Program tulis_file_karakter; {menulis data pada file bertipe karakter} uses winCRT; {Deklarasi} type ArsipKar = file of char; { TANDA = '.' } var fKar : ArsipKar; kar : char; i : integer; jumdat : integer; namafile : string; {deskripsi} Begin writeln('MENULIS DATA KARAKTER KE FILE RUNTUNAN'); Write(' ketikkan nama file (*.txt) : '); readln(namafile); Write(' jumlah data : '); readln(jumdat); assign (fKar,namafile); rewrite (fKar); {membuka file untuk</pre>

<pre> for i ← 1 to jumdat do write('data ',i) read(kar) fwrite(fKar, kar) end for kar ← '.' {end data} fwrite(fKar,kar) Close (fKar) {menutup file} </pre>	<pre> penulisan} writeln('masukkan data :'); for i := 1 to jumdat do begin write('data ',i:2,' '); readln(kar); write(fKar, kar); end; kar := '.'; {end data} write(fKar,kar); Close (fKar); {menutup file} end. </pre>
---	---

Penulisan ke File karakter Cara 2 :

Algoritma	Program
<pre> Algoritma tulis_file_karakter2 {menulis data pada file bertipe karakter} {Deklarasi} type ArsipKar = file of char { TANDA = '.' } fKar : ArsipKar kar : char i : integer {deskripsi} Open (fKar,1){membuka file untuk penulisan} write('masukkan data,jika selesai ketik ".") i ← 1 write('data ',i) readln(kar) while kar <> '.' do fwrite(fKar, kar) i ← i+1 write('data ',i) readln(kar) end while fwrite(fKar,kar) {rekam / tulis TANDA} Close (fKar) {menutup file} </pre>	<p>Bagaimana Programnya ?</p>

Membaca File Karakter :

Algoritma	Program
<pre> Algoritma baca_file_karakter; {membaca data pada file bertipe karakter} {Deklarasi} type ArsipKar = file of char; { TANDA = '.' } fKar : ArsipKar kar : char i : integer </pre>	<pre> Program baca_file_karakter; {membaca data pada file bertipe karakter} uses wincrt; {Deklarasi} type ArsipKar = file of char; { TANDA = '.' } var fKar : ArsipKar; kar : char; </pre>

<pre> jumdat : integer {deskripsi} write('MEMBACA DATA KARAKTER DARI FILE RUNTUNAN') open (fKar,2) {membuka file untuk pembacaan} fread(fKar, kar) while kar <> '.' do write (kar,' ') fread(fKar, kar) end while Close (fKar); {menutup file} </pre>	<pre> i : integer; jumdat : integer; namafile : string; {deskripsi} Begin writeln('MEMBACA DATA KARAKTER DARI FILE RUNTUNAN'); Write(' ketikkan nama file (*.txt) : '); readln(namafile); assign (fKar,namafile); reset (fKar); {membuka file untuk pembacaan} read(fKar, kar); while kar <> '.' do begin writeln (kar,' '); read(fKar, kar); end; Close (fKar); {menutup file} end. </pre>
--	---

File bertipe Record

Contoh Program file berikut ini menggunakan tipe record 'DataMhs'.

Menulis File DataMhs :

```

Program tulis_file_record;
{menulis data pada file bertipe record}

uses wincrt;

{Deklarasi}
type
    DataMhs = record
        NIM      : longint;
        Nama     : string[25];
        KodeMK   : string[6];
        SKS      : integer;
        indeks   : char;
    end;

    ArsipMhs = file of DataMhs; {TANDA = <9999999,'XXXXX','XXXXX',99,'X' }

var
    fMhs      : ArsipMhs;
    i          : integer;
    namafile   : string;

function MARK(RekMhs : DataMhs) : boolean;
{mengembalikan nilai true jika RekMhs adalah rekaman TANDA;
atau false jika tidak }

begin
    MARK :=(RekMhs.NIM = 999999);
end;

procedure TulisDataMahasiswa;
{merekam data ujian ke dalam arsip fMhs, data dibaca dari piranti masukan,

```



```

proses perekaman berhenti jika NIM yang dimasukan = 999999
K.Awal   : sembarang;
K.Akhir  : arsip fMhs berisi data ujian mahasiswa }

var
    KuliahMhs : DataMhs;
    i : integer;

begin
    writeln('MENULIS DATA MAHASISWA KE FILE RUNTUNAN BERTIPE RECORD');
    Write(' ketikkan nama file (*.txt) : '); readln(namafile);
    assign (fMhs,namafile);
    rewrite (fMhs); {membuka file untuk penulisan}
    writeln('masukkan data, jika selesai ketik NIM = 999999 :');
    i := 1;
    writeln('mahasiswa ',i:2,' : ');
    write('NIM (7 digit)  : '); readln (KuliahMhs.NIM);

    while (not MARK(KuliahMhs)) do
    begin
        write('Nama           : '); readln (KuliahMhs>Nama);
        write('KodeMK(6 digit): '); readln (KuliahMhs.KodeMK);
        write('SKS           : '); readln (KuliahMhs.SKs);
        write('indeks        : '); readln (KuliahMhs.indeks);
        write(fMhs, KuliahMhs);      {rekam / tulis data ke file}
        i := i+1;
        writeln('mahasiswa ',i:2,' : ');
        write('NIM (7 digit)  : '); readln (KuliahMhs.NIM);
    end;

    KuliahMhs.NIM      := 999999;
    KuliahMhs>Nama     := 'XXXXX';
    KuliahMhs.KodeMK   := 'XXXXX';
    KuliahMhs.SKs      := 99;
    KuliahMhs.indeks   := 'X';
    write(fMhs, KuliahMhs);      {rekam / tulis TANDA}
    Close (fMhs);      {menutup file}

end;

{program utama}
begin
    TulisDataMahasiswa;
end.

```

Membaca File DataMhs :

```

Program baca_file_record;
{membaca data pada file bertipe record}

uses wincrt;

{Deklarasi}
type
    DataMhs = record
        NIM      : longint;
        Nama     : string[25];
        KodeMK   : string[6];
        SKs      : integer;
        indeks   : char;
    end;
end;

```

```

    ArsipMhs = file of DataMhs; {TANDA = <999999,'XXXXX','XXXXX',99,'X' }

var
    fMhs      : ArsipMhs;
    i         : integer;
    namafile  : string;

function MARK(RekMhs : DataMhs) : boolean;
{mengembalikan nilai true jika RekMhs adalah rekaman TANDA;
atau false jika tidak }

begin
    MARK :=(RekMhs.NIM = 999999);
end;

procedure BacaDataMahasiswa;
{membaca data ujian dari arsip fMhs
proses pembacaan berhenti jika NIM yang dibaca = 999999
K.Awal   : arsip fMhs terdefinisi;
K.Akhir  : arsip fMhs berisi data ujian mahasiswa ditampilkan pada layar}

var
    KuliahMhs : DataMhs;
    i         : integer;

begin
    writeln('MEMBACA DATA MAHASISWA DARI FILE RUNTUNAN BERTIPE RECORD');
    Write(' ketikkan nama file (*.txt) : '); readln(namafile);
    assign (fMhs,namafile);
    reset (fMhs); {membuka file untuk pembacaan}
    writeln('-----');
    writeln(' No.    NIM        Nama        KodeMK SKS Indeks ');
    writeln('-----');
    i := 1;
    read(fMhs,KuliahMhs); {baca data dari file}
    while (not MARK(KuliahMhs)) do
        begin
            writeln(i:3,' ',KuliahMhs.NIM:7,' ',KuliahMhs>Nama:10,' ',
                    KuliahMhs.KodeMK:6,' ',KuliahMhs.SKS:3,' ',KuliahMhs.indeks);
            i := i+1;
            read(fMhs,KuliahMhs); {baca data dari file}
        end;
    writeln('-----');
    Close (fMhs); {menutup file}

end;

{program utama}
begin
    BacaDataMahasiswa;
end.

```

Perhatikan bahwa tanda untuk mengakhiri file dijadikan fungsi, dan proses pembacaan serta penulisan dijadikan prosedur. Semua proses yang terjadi pada file akhirnya dijadikan prosedur, sehingga program utama menjadi sederhana. Hanya di perlukan ketelitian dalam penentuan passing parameter antara prosedur dan program utama.

File Tak Bertipe: cara pengaksesan disk (disket) secara low level

- Pendeklarasian file tak bertipe dilakukan dengan menggunakan kata tercadang file

Contoh :

```
Var  
    FileProses : FILE
```

- Yang dapat dilakukan dengan file tak bertipe :
 1. Menyalin sebarang file (termasuk file berekstensi COM atau EXE)
 2. Mengganti nama file
 3. Menghapus nama file
 4. Memeriksa keberadaan suatu file
 5. Menampilkan isi sebarang file
 6. Mengganti data yang ada pada sebarang file dengan basis karakter
 7. Mencari kata-kata tertentu pada sebarang file
- File tak bertipe bekerja dengan **basis record** (suatu istilah yang harus dibedakan dengan tipe record)
- Operasi File Tak Bertipe = file bertipe dan file teks
Diawali dengan assign dan diikuti rewrite atau reset
- Selain itu fungsi dan prosedur yang dipakai :

1. *Membaca record*

Untuk membaca satu atau sejumlah record file tak bertipe, menggunakan prosedur blockread :

```
Procedure BlockRead(var f:FILE; var Buf; cacah; cacah:word; var  
    hasil:word)
```

f : variable file tak bertipe

Buf : sebarang variable; digunakan sebagai penyangga file

Cacah : ekspresi bertipe word

Hasil : variable bertipe word

BlockRead akan membaca record sebanyak cacah atau lebih kecil dari cacah dari file f. Jumlah actual yang dibaca (kurang atau sama dengan cacah) dikembalikan ke hasil (kalau hasil disertakan).

Jika hasil kurang daripada cacah, menyatakan bahwa akhir file sudah tercapai.

Ukuran variable buf adalah :

Ukuran record * cacah

2. *Merekam file*

Untuk merekam / menulis sejumlah record ke file tak bertipe, menggunakan prosedur blockwrite :

```
Procedure BlockWrite(var f:FILE; var Buf; cacah; cacah:word; var  
    hasil:word)
```

f : variable file tak bertipe

Buf : sebarang variable; digunakan sebagai penyangga file

Cacah : ekspresi bertipe word
Hasil : variable bertipe word

BlockWrite akan merekam record sebanyak cacah atau lebih kecil dari cacah dari file f. Jumlah actual yang direkam (kurang atau sama dengan cacah) dikembalikan ke parameter yang bersifat opsional, hasil
Jika hasil kurang daripada cacah, menyatakan bahwa akhir file sudah tercapai.

3. Perintah-perintah lain

File tak-bertipe juga mengenal fungsi dan prosedur yang digunakan pada file bertipe. Fungsi seperti **Eof**, **FileSize**, **FilPos**, **Truncate** dan **Seek** juga bisa dipakai. Namun perlu diingat bahwa kalau sejumlah perintah pada file bertipe bekerja dengan basis komponen, basis pada file tak bertipe adalah *record*.

- Contoh program

Menulis File Tak Bertipe :

```
Program file_tak_bertipe;
{demo file tak bertipe}

uses wincrt;

{Deklarasi}
const
    ukuran = 128;

type
    arr = array[1..ukuran] of byte;

var
    fbiner : FILE;
    buffer : arr;
    Jumdata : word;
    namafile : string;

procedure tampilkan_data(var f:FILE);

var
    penyangga : arr;
    JumData,i : word;

begin
    seek(f,0); { ke awal file}
    writeln('isi file :');
    repeat
        begin
            BlockRead(f,penyangga,ukuran,JumData);
            for i := 1 to JumData do
                write(chr(penyangga[i]));
            end
        until JumData < ukuran;
        writeln;
    end;

{deskripsi}
begin
```

```

writeln ('DEMO FILE TAK BERTIPE');
write('ketik nama file : '); readln(namafile);
assign(fbiner,namafile);
rewrite(fbiner,1);

writeln('posisi setelah rewrite = ', filepos(fbiner));

buffer[1] := ord('T');
buffer[2] := ord('E');
buffer[3] := ord('S');

{rekam 3 karakter}
BlockWrite(fbiner,buffer,3,JumData);

writeln('posisi setelah BlockWrite = ', filepos(fbiner));

tampilkan_data(fbiner);

seek(fbiner,2); {ke komponen ke 2}
buffer[1] := ord('1');
buffer[2] := ord('2');
buffer[3] := ord('3');

{rekam 3 karakter}
BlockWrite(fbiner,buffer,3,JumData);

tampilkan_data(fbiner);

close(fbiner); {tutup file}
end.

```

Menyalin / mengcopy file Tak Bertipe :

```

Program salin_file_tak_bertipe;
{salin/copy file tak bertipe ke file lain}

uses wincrt;

{Deklarasi}
const
    ukuran = 1040; {ukuran penyangga file}

type
    arr = array[1..ukuran] of byte;

var
    fIn,fOut                : FILE;
    penyangga               : arr;
    JumBaca,JumTulis        : word;
    namafileIn,namafileOut  : string;

procedure tampilkan_data(var f:FILE);

var
    penyangga : arr;
    JumData,i : word;

begin
    seek(f,0); { ke awal file}
    writeln('isi file :');
    repeat
        begin
            BlockRead(f,penyangga,ukuran,JumData);

```

```

        for i := 1 to JumData do
            write(chr(penyangga[i]));
        end
    until JumData < ukuran;
    writeln;
end;

{deskripsi}
begin
    writeln ('SALIN FILE TAK BERTIPE');
    write('nama file yang akan disalin: '); readln(namafileIn);
    write('nama file hasil salinan      : '); readln(namafileOut);

    assign(fIn,namafileIn);
    reset(fIn,1);      {ukuran record = 1 byte}

    assign(fOut,namafileOut);
    rewrite(fOut,1);
    repeat
        BlockRead(fIn,penyangga,ukuran,JumBaca);
        if JumBaca <> 0 then
            BlockWrite(fOut,penyangga,JumBaca,JumTulis);
        until (JumBaca < ukuran) or (JumBaca <> JumTulis);
        writeln('sudah salin file');

        writeln('file ',namafileIn, ' : ');
        tampilkan_data(fIn);

        writeln('file ',namafileOut, ' : ');
        tampilkan_data(fOut);

        close(fIn);
        close(fOut);
    end.

```

DAFTAR PUSTAKA

- [KAD02] Kadir, Abdul,Pemrograman Pascal, Andi Yogyakarta, 2002
[MUN02] Munir, Rinaldi, Algoritma dan Pemrograman – Buku 2, Penerbit Informatika Bandung, 2002.