



Pemrograman Dasar



Dhidhi Pambudi
Dwi Maryono



Hak Cipta pada Kementerian Pendidikan dan Kebudayaan
Dilindungi Undang-Undang

Milik Negara
Tidak Diperdagangkan

Kontributor :
Penyunting materi : Rosihan Ariyuana, S.Si., M.Kom
Penyunting bahasa : Rosihan Ariyuana, S.Si., M.Kom
Desain Sampul : Punto Wicaksono, Bayu Adi Prasetyo Wibowo

Katalog dalam terbitan

Cetakan ke 1, 2014
Disusun dengan huruf Arial 11pt

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan yang Maha Esa atas tersusunnya buku teks ini, dengan harapan dapat digunakan sebagai buku teks Pemrograman Dasar untuk siswa Sekolah Menengah Kejuruan (SMK) Bidang Keahlian Teknologi Informasi.

Penerapan kurikulum 2013 mengacu pada paradigma belajar kurikulum abad 21 menyebabkan terjadinya perubahan, yakni dari pengajaran (*teaching*) menjadi BELAJAR (*learning*), dari pembelajaran yang berpusat kepada guru (*teachers-centered*) menjadi pembelajaran yang berpusat kepada peserta didik (*student-centered*), dari pembelajaran pasif (*pasive learning*) ke cara belajar peserta didik aktif (*active learning-CBSA*) atau *Student Active Learning-SAL*.

Buku teks "Pemrograman Dasar" ini disusun berdasarkan tuntutan paradigma pengajaran dan pembelajaran kurikulum 2013 diselaraskan berdasarkan pendekatan model pembelajaran yang sesuai dengan kebutuhan belajar kurikulum abad 21, yaitu pendekatan model pembelajaran berbasis peningkatan keterampilan proses sains.

Penyajian buku teks untuk Mata Pelajaran Pemrograman Komputer ini disusun dengan tujuan agar peserta didik dapat melakukan proses pencarian pengetahuan berkenaan dengan materi pelajaran melalui berbagai aktivitas proses sains sebagaimana dilakukan oleh para ilmuwan dalam melakukan eksperimen ilmiah (penerapan pendekatan saintifik), dengan demikian peserta didik diarahkan untuk menemukan sendiri berbagai fakta, membangun konsep, dan nilai-nilai baru secara mandiri. Pendekatan saintifik ini meliputi proses 5 M,yaitu mengamati, menanya, mencoba/mengumpulkan informasi, mengassosiasikan, dan mengkomunikasikan.

Kementerian Pendidikan dan Kebudayaan, Direktorat Pembinaan Sekolah Menengah Kejuruan, dan Direktorat Jenderal Peningkatan Mutu Pendidik dan Tenaga Kependidikan menyampaikan terima kasih, sekaligus saran kritik demi kesempurnaan buku teks ini dan penghargaan kepada semua pihak yang telah berperan serta dalam membantu terselesaikannya buku teks siswa untuk Mata Pelajaran "Pemrograman Komputer Kelas XI Semester 1 Sekolah Menengah Kejuruan (SMK).

Jakarta, Desember 2014

Menteri Pendidikan dan Kebudayaan

DAFTAR ISI

Kata Pengantar	iii
Daftar Isi	iv
Pendahuluan	v
Bab I. Operator Aritmatika dan Logika	1
KB 1. Operator Aritmatika	2
KB 2. Pemanfaatan Operator Aritmatika	13
KB 3. Fungsi Aritmatika	21
KB 4. Pemanfaatan Fungsi Aritmatika	28
KB 5. Operasi Logika	35
Bab II. Array	45
KB 1. Array Satu Dimensi	46
KB 2. Pemanfaatan Array Satu Dimensi	75
KB 3. Array Dua Dimensi	85
KB 4. Pemanfaatan Array Dua Dimensi	96
KB 5. Array Multi Dimensi	104
KB 6. Pemanfaatan Array Multi Dimensi	111
Bab III. Operasi String dan Konversi Data	117
KB 1. Pengertian String	118
KB 2. Operasi Pada String	127
KB 3. Pemanfaatan String dalam Aplikasi	137
KB 4. Konversi Data	134
KB 5. Pemanfaatan Konversi Data dalam Aplikasi	150
Bab IV. Pointer	157
KB 1. Pointer ke Variabel	158
KB 2. Pointer ke Array	183
KB 3. Pemanfaatan Pointer	195
Daftar Pustaka	203

PENDAHULUAN

Buku Pemrograman Dasar ini ditujukan untuk SMK/MAK bidang keahlian Teknologi Informasi dan Komunikasi kelas XI semester 1. Buku ini dirancang untuk menjadi buku pegangan siswa agar bisa digunakan untuk membantu pembelajaran yang menerapkan Kurikulum 2013 (K13). Buku ini bukanlah pedoman utama dalam pembelajaran berbasis Kurikulum 2013 tetapi buku ini hanya sebagai salah satu sumber belajar yang mengarah pada penerapan Kurikulum 2013. Keberhasilan proses pembelajaran bukan ditentukan semata-mata oleh buku ini tetapi lebih ditekankan pada bagaimana siswa bisa lebih berkembang dengan menerapkan cara belajar dan berpikir *scientific*.

Buku ini sudah dirancang sedemikian rupa agar mudah dipelajari. Supaya siswa lebih mudah menguasai setiap materi yang disajikan maka siswa diharapkan sudah memenuhi standar ketuntasan belajar dalam pelajaran Pemrograman Dasar kelas X. Siswa diharapkan banyak mencoba dan mengembangkan kreativitasnya dalam pembuatan program. Buku ini menggunakan bahasa pemrograman Pascal dan *compiler* yang digunakan dalam penyusunan buku adalah Free Pascal. Diharapkan siswa sudah terbiasa menggunakan *compiler* dan *Integrated Development Environment* (IDE) Free Pascal.

Penguasaan materi buku ini bisa dilakukan dengan cara mempelajarinya secara terurut mulai dari Bab I sampai Bab IV dan mempelajari setiap Kegiatan Belajar (KB) secara terurut pula. Pada setiap Kegiatan Belajar siswa diminta untuk benar-benar membaca isi pada bagian kegiatan Mengamati dan selanjutnya siswa diharapkan memunculkan pertanyaan dari dirinya sendiri, meskipun dalam buku ini sudah disediakan beberapa pertanyaan yang berkaitan dengan materi. Setiap pertanyaan yang sudah disediakan dalam buku ini bisa dijawab oleh siswa setelah melakukan beberapa percobaan yang sudah disediakan pada kegiatan selanjutnya. Setiap nomor pertanyaan akan bersesuaian dengan nomor eksperimen, yaitu: pertanyaan nomor satu bersesuaian dengan eksperimen nomor satu, pertanyaan nomor dua bersesuaian dengan eksperimen nomor dua, dan seterusnya.

Kompetensi Inti:

- K1. Menghayati dan mengamalkan ajaran agama yang dianutnya
- K2. Menghayati dan mengamalkan perilaku jujur, disiplin, tanggungjawab, peduli (gotong royong, kerjasama, toleran, damai), santun, responsif dan pro-aktif dan menunjukkan sikap sebagai bagian dari solusi atas berbagai permasalahan dalam berinteraksi secara efektif dengan lingkungan sosial dan alam serta dalam menempatkan diri sebagai cerminan bangsa dalam pergaulan dunia.
- K3. Memahami, menerapkan, dan menganalisis pengetahuan faktual, konseptual, prosedural, dan metakognitif berdasarkan rasa ingin tahu tentang ilmu pengetahuan, teknologi, seni, budaya, dan humaniora dalam wawasan kemanusiaan, kebangsaan, kenegaraan, dan peradaban terkait penyebab fenomena dan kejadian dalam bidang kerja yang spesifik untuk memecahkan masalah.
- K4. Mengolah, menalar, dan menyaji dalam ranah konkret dan ranah abstrak terkait dengan pengembangan dari yang dipelajarinya di sekolah secara mandiri, bertindak secara efektif dan kreatif, dan mampu melaksanakan tugas spesifik di bawah pengawasan langsung.

Kompetensi Dasar:

- 1.1. Memahami nilai-nilai keimanan dengan menyadari hubungan keteraturan dan kompleksitas alam dan jagad raya terhadap kebesaran Tuhan yang menciptakannya
- 1.2. Mendeskripsikan kebesaran Tuhan yang menciptakan berbagai sumber energi di alam
- 1.3. Mengamalkan nilai-nilai keimanan sesuai dengan ajaran agama dalam kehidupan sehari-hari
- 1.4. Menunjukkan perilaku ilmiah (memiliki rasa ingin tahu; objektif; jujur; teliti; cermat; tekun; hati-hati; bertanggung jawab; terbuka; kritis; kreatif; inovatif dan peduli lingkungan) dalam aktivitas sehari-hari sebagai wujud implementasi sikap dalam melakukan percobaan dan berdiskusi
- 1.5. Menghargai kerja individu dan kelompok dalam aktivitas sehari-hari sebagai wujud implementasi melaksanakan percobaan dan melaporkan hasil percobaan.

PEMBELAJARAN

Materi:

- Bab I. Aritmatika dan Logika
- Bab II. Array
- Bab III. Operasi String dan Konversi Data
- Bab IV. Pointer

BAB

I

OPERATOR ARITMATIKA DAN LOGIKA

Kompetensi Dasar:

- Mendeskripsikan operasi aritmatika
- Mendeskripsikan operasi logika
- Memecahkan permasalahan perhitungan dengan operasi aritmatika
- Memecahkan permasalahan logika dengan operasi logika

BAB I OPERASI ARITMATIKA DAN LOGIKA

Dewasa ini seiring dengan perkembangan teknologi informasi yang pesat, perkembangan software-software yang digunakan untuk membantu setiap kegiatan manusia juga semakin berkembang ke arah yang lebih baik. Sebagai dampak perkembangan software yang begitu pesat maka kebutuhan akan seorang programer software yang handal pun juga meningkat. Namun kebutuhan akan programer yang handal tidak diikuti dengan upaya peningkatan sumber daya manusia yang memadai.

Siswa SMK yang nantinya diharapkan bisa memenuhi kebutuhan pasar software yang ada, perlu adanya pemahaman mengenai dasar-dasar pemrograman yang baik. Dengan pemahaman dasar-dasar pemrograman yang baik diharapkan ketika mengembangkan suatu program dapat diminimalisir kesalahan-kesalahan dasar yang dapat mengakibatkan program tidak dapat berjalan sebagaimana mestinya. Ada berbagai macam dasar-dasar pemrograman yang harus dikuasai antara lain struktur program, variabel, operator, array dll.

Maka dalam bab ini, kita akan membahas dasar pemrograman yang berkaitan dengan operator. Operator memiliki peran yang vital dalam suatu struktur program. Operator bertugas untuk melakukan manipulasi-manipulasi terhadap suatu nilai variabel yang ada menjadi nilai variabel yang diinginkan programmer.

Pada bab ini, kita akan mempelajari berbagai jenis operator yang ada dalam bahasa pemrograman *Pascal*. Ada beberapa macam operator yang dikenal antara lain operator aritmatika (*arithmetic operator*), operator relasi (*relational operator*), operator logika (*logical operator*), operator *Bitwise* (*Bitwise operator*) dan operator penugasan (*assignment operator*).

1.1. Kegiatan Belajar 1. Operator Aritmatika

Alokasi Waktu : 2 x 45 menit

1.1.1. Tujuan Pembelajaran

Tujuan pembelajaran pada Kegiatan Belajar 1 tentang Operator Aritmatika adalah:

1. Siswa mengetahui berbagai jenis operator aritmatika yang digunakan dalam pemrograman.
2. Siswa dapat menjelaskan pengertian operand dan operator.
3. Siswa dapat menjelaskan cara kerja dari operator aritmatika dalam suatu program.
4. Siswa dapat menggunakan operator aritmatika dalam program yang sederhana.

1.1.2. Aktivitas belajar siswa

1.1.2.1. Mengamati/ observasi

Ketika masih duduk di bangku sekolah dasar, pernahkah kamu menemui permasalahan di bawah ini?. Permasalahan di bawah ini merupakan contoh masalah operasi matematika dasar yang dapat dijumpai dalam kehidupan sehari-hari.

Satu bulan yang lalu, Pak ketut membeli 3 ekor kambing. Kemudian, Kemarin Pak Ketut membeli lagi 4 ekor kambing. Berapakah jumlah kambing Pak Ketut sekarang?

Kemudian muncul pertanyaan, bagaimanakah cara untuk menyelesaikan permasalahan di atas?. Coba kalian cermati cara yang digunakan untuk menyelesaikan permasalahan di atas :

Penyelesaian :

Pertama, Bulan lalu Pak Ketut membeli 3 ekor kambing.

Kedua, Kemarin Pak ketut membeli lagi 4 ekor kambing.

Maka, jumlah kambing pak ketut :

$$3 + 4 = 7$$

Jadi, jumlah kambing Pak Ketut adalah 7 ekor.

Coba kalian cermati bagian ekspresi matematika yang digunakan untuk menyelesaikan permasalahan di atas.

$$3 + 4 = 7$$

Bilangan 3 dan 4 dalam ekspresi matematika di atas bertindak sebagai Operand sedangkan simbol “+” berperan sebagai operator. Lalu, Apa definisi dari operand dan operator?. Untuk itulah pada bab ini kita akan mempelajari definisi operand dan operator, jenis-jenis operator dan bagaimana cara menggunakan operator dalam bahasa pemrograman Pascal.

Definisi Operator dan Operand

Operand adalah nilai atau data asal yang digunakan dalam suatu operasi. Operator dapat didefinisikan sebagai simbol atau instruksi khusus yang digunakan untuk memanipulasi operand.

Operator Relasional

Operator Relasional adalah operator yang digunakan untuk menentukan relasi atau hubungan dari dua buah operand. Misalkan variabel $a = 10$ dan $b = 12$ maka :

Operator	Simbol	Deskripsi	Contoh
Sama dengan	=	Untuk mengecek nilai dari kedua operand, jika nilainya sama maka pernyataan tersebut bernilai benar.	$(a = b)$ salah
Tidak sama dengan	\neq	Untuk mengecek nilai dari kedua operand, jika nilainya tidak sama maka pernyataan tersebut bernilai benar.	$(a \neq b)$ benar
Lebih dari	>	Untuk mengecek nilai dari operand kiri lebih besar daripada operand kanan. Jika nilainya lebih besar maka pernyataan tersebut bernilai benar.	$(a > b)$ salah
Lebih dari sama dengan	\geq	Untuk mengecek nilai dari operand kiri lebih besar atau sama dengan operand kanan. Jika nilainya lebih besar atau sama dengan maka pernyataan tersebut bernilai benar.	$(a \geq b)$ salah
Kurang dari	<	Untuk mengecek nilai dari operand kanan lebih besar daripada operand kiri. Jika nilainya lebih besar maka pernyataan tersebut bernilai benar.	$(a < b)$ benar
Kurang dari sama dengan	\leq	Untuk mengecek nilai dari operand kanan lebih besar atau sama dengan operand kiri. Jika nilainya lebih besar atau sama dengan maka pernyataan tersebut bernilai benar.	$(a \leq b)$ benar

Operator Aritmatika

Operator aritmatika adalah operator yang digunakan untuk melakukan operasi-operasi perhitungan aritmatika seperti penjumlahan, pengurangan, pembagian, perkalian dll. Dalam bahasa pemrograman Pascal operator aritmatika dapat dibagi menjadi 2, yaitu :

- Operator Aritmatika Unair

Operator aritmatika unair adalah operator aritmatika yang hanya melibatkan 1 operand saja.

- Operator Aritmatika Biner

Operator aritmatika biner adalah operator aritmatika yang melibatkan 2 buah operand.

Berikut ini jenis operator dan kegunaannya dalam bahasa pemrograman pascal :

Operator	Simbol	Deskripsi	Contoh
Penjumlahan	+	menambahkan suatu operand dengan operand yang lain	$5 + 7 = 11$
Pengurangan	-	mengurangkan suatu operand dengan operand yang lain	$5 - 7 = -2$
Perkalian	*	melakukan penjumlahan berulang pada suatu operand	$5 * 7 = 35$

Pembagian	/	melakukan pembagian pada operand	10 / 2 = 5
Pembagian Bulat	Div	membagi 2 bilangan bulat (integer) kemudian mengambil hasil bagi dan sisanya dibuang	7 div 2 = 3
Modulus	Mod	membagi 2 bilangan bulat (integer) kemudian mengambil sisanya dan hasil baginya dibuang.	7 mod 2 = 1

Setelah mengetahui definisi dan kegunaan dari berbagai macam operator aritmatika, berikut ini adalah contoh program sederhana yang memanfaatkan operasi aritmatika pada FreePascal :

Contoh :

Jika diketahui $a = 1$ dan $b = 2$. Buatlah program pada FreePascal untuk memperoleh hasil operasi $a + b$!

```
program Penjumlahan;
var a,b:integer;
begin
    a:=1;
    b:=2;
    Writeln('a + b = ', a+b);
    Write('Press any key to continue... ');
    Readln;
end.
```

Contoh :

Diketahui sebuah persegi panjang dengan panjang 10 cm dan lebar 5 cm. Buatlah program menggunakan FreePascal untuk menghitung luas dari persegi panjang tersebut!

```
program LuasPersegi;
var p,l:integer;
begin
    p:=10;
    l:=5;
    Writeln('Luas Persegi = ', p*l , 'cm2');
    Write('Press any key to continue... ');
    Readln;
end.
```

Contoh :

Willy mempunyai 7 buah permen. Dia berencana akan membagi permen yang dimiliki kepada 3 orang adiknya dengan jumlah yang sama banyak. Dengan menggunakan FreePascal hitunglah jumlah permen yang diterima masing-masing adik willy dan apakah ada sisanya?

```
program permeh;  
  
begin  
  
    writeln('permen yang diperoleh = ', 7div3);  
  
    writeln('sisa permen willy = ', 7mod3)  
  
    write('press any key to continue...');  
  
    readln;  
  
end.
```

1.1.2.2. Menanya

Berdasarkan kegiatan mengamati, ada beberapa hal yang bisa diperhatikan dan dibahas lebih lanjut. Beberapa pertanyaan yang berkaitan dengan operator aritmatika adalah:

1. Bagaimana jika operator minus (-) hanya melibatkan satu operand saja?
2. Bagaimana jika operator plus (+) hanya melibatkan satu operand saja?
3. Bagaimana jika kita salah meletakkan urutan operand dengan operator ? Misalnya a dan b kemudian ditulis ab+, apa yang akan terjadi?
4. Bagaimana urutan proses penghitungan jika kita melibatkan lebih dari 1 operator matematika, misalkan $a+b*c-d+e/f$?

Untuk menjawab pertanyaan-pertanyaan tersebut, siswa diharapkan bisa melakukan kegiatan berikut dengan bimbingan guru.

1.1.2.3. Mencoba

Percobaan 1

Untuk mengetahui jawaban dari pertanyaan pertama yaitu bagaimana jika operator minus (-) hanya melibatkan satu operand saja?, salin program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Aritmatika1;  
  
uses crt;  
  
var a:integer; //deklarasi variabel a  
  
begin  
    clrscr;  
  
    writeln('Masukkan nilai a');  
    readln(a); //input variabel a  
  
    writeln('b =',-a); //menampilkan hasil dari -a  
    readln;  
end.
```

Hasil Percobaan Program Aritmatika1:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 2

Untuk mengetahui jawaban dari pertanyaan kedua yaitu bagaimana jika operator plus (+) hanya melibatkan stu operand saja?, salin program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Aritmatika2;
uses crt;
var a:integer; //deklarasi variabel a
begin
    clrscr;
    writeln('Masukkan nilai a');
    readln(a); //input variabel a
    writeln('b =',+a); //menampilkan hasil dari +a
    readln;
end.
```

Hasil Percobaan Program Aritmatika2:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 3

Untuk mengetahui jawaban dari pertanyaan ketiga yaitu bagaimana jika kita salah meletakkan urutan operand dengan operator ? Misalnya a dan b kemudian ditulis ab+, apa yang akan terjadi?, salin program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

Program 3. Aritmatika3

```
program Aritmatika3;
uses crt;
var a,b:integer;
begin
    clrscr;
```

```
writeln('Masukkan nilai a '); //input variabel a
readln(a);

writeln('=====');
writeln('Masukkan nilai b '); //input variabel b
readln(b);
writeln('=====');

//menampilkan hasil operasi ab+
writeln('ab+ =', ab+);
readln;
end.
```

Hasil Percobaan Program Aritmatika3:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

Percobaan 4

Untuk mengetahui jawaban dari pertanyaan keempat yaitu bagaimana urutan proses penghitungan jika kita melibatkan lebih dari 1 operator matematika, misalkan $a+b*c-d+e/f$?, salin program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Aritmatika4;

uses crt;

var a,b,c,d,e,f:integer; //deklarasi variabel a,b,c,d,e,f
```

```
begin
    clrscr;

    writeln('Masukkan nilai a'); //input nilai a
    readln(a);

    writeln('Masukkan nilai b'); //input nilai b
    readln(b);

    writeln('Masukkan nilai c'); //input nilai c
    readln(c);

    writeln('Masukkan nilai d'); //input nilai d
    readln(d);

    writeln('Masukkan nilai e'); //input nilai e
    readln(e);

    writeln('Masukkan nilai f'); //input nilai f
    readln(f);

    //mengoperasikan a+b
    writeln('=====');
    writeln('Hasil dari a+b= ',a+b);

    //mengoperasikan a+b*c
    writeln('=====');
    writeln('Hasil dari a+b*c= ',a+b*c);

    //mengoperasikan a+b*c-d
    writeln('=====');
    writeln('Hasil dari a+b*c-d= ',a+b*c-d);

    //mengoperasikan a+b*c-d+e
    writeln('=====');
    writeln('Hasil dari a+b*c-d+e= ',a+b*c-d+e);

    //mengoperasikan a+b*c-d+e/f
    writeln('=====');
    writeln('Hasil dari a+b*c-d+e/f= ',a+b*c-d+e/e:4:2);

    writeln('=====');
    readln;

end.
```

Hasil Percobaan Program Aritmatika4:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

1.1.2.4. Mengasosiasi/ menalar

Setelah siswa melakukan percobaan yang sesuai dengan pertanyaan yang ada, siswa diharapkan melakukan kegiatan menalar yang bisa dilakukan secara individu oleh masing-masing siswa atau bisa juga berkelompok. Beberapa hal yang bisa dijadikan arahan untuk menalar :

1. Perhatikan hasil kompilasi, apakah berhasil atau justru terjadi kesalahan.
2. Perhatikan urutan proses dari setiap program yang sudah dicoba.
3. Coba bandingkan beberapa program yang memiliki kemiripan, program mana yang berhasil dikompilasi dan program mana yang tidak berhasil dikompilasi.
4. Perhatikan letak/posisi kesalahan, coba pikirkan mengapa terjadi kesalahan pada posisi tersebut.
5. Cobalah membuat kesimpulan dari percobaan yang telah dilakukan.

1.1.3. Rangkuman

Dari percobaan-percobaan yang telah dilakukan, bisa ambil kesimpulan bahwa:

1. Operand adalah nilai atau data asal yang digunakan dalam suatu operasi.
2. Operator adalah simbol atau instruksi khusus yang digunakan untuk memanipulasi operand.
3. Operator aritmatika dapat dibedakan menjadi 2, yaitu :
 - Operator Aritmatika Unair
 - Operator Aritmatika Biner
4. Operator aritmatika unair adalah operator aritmatika yang hanya melibatkan 1 operand saja.
Operator aritmatika unair terdiri dari :

Operator	Simbol	Deskripsi	Contoh
Plus	+	menunjukkan bilangan positif baik pada operand numerik integer maupun numerik real	<ul style="list-style-type: none"> Misal $a = -1 ; b = +a = -1$ $c = 1; d = +a = 1$
Minus	-	menunjukkan bilangan negatif baik pada operand numerik integer maupun numerik real	<ul style="list-style-type: none"> Misal $a = -1 ; b = -a = -(-1) = 1$ $c = 1; d = +a = +(-1) = 1$

5. Operator aritmatika biner adalah operator aritmatika yang melibatkan 2 buah operand. Berikut ini jenis operator dan kegunaannya dalam bahasa pemrograman pascal :

Operator	Simbol	Deskripsi	Contoh
Penjumlahan	+	menambahkan suatu operand dengan operand yang lain	$5 + 7 = 11$
Pengurangan	-	mengurangkan suatu operand dengan operand yang lain	$5 - 7 = -2$
Perkalian	*	melakukan penjumlahan berulang pada suatu operand	$5 * 7 = 35$
Pembagian	/	melakukan pembagian pada suatu bilangan terhadap operand	$10 / 2 = 5$
Pembagian Bulat	div	membagi 2 bilangan bulat (integer) kemudian mengambil hasil bagi dan sisanya dibuang	$7 \text{ div } 2 = 3$
Modulus	mod	membagi 2 bilangan bulat (integer) kemudian mengambil sisanya dan hasil baginya dibuang.	$7 \text{ mod } 2 = 1$

6. Untuk operasi aritmatika biner, kita harus meletakkan operator diantara dua operand yang akan dioperasikan. Apabila operator diletakkan setelah dua operand (misal : ab+) maka pada saat proses compile akan terjadi eror yang diakibatkan oleh kesalahan peletakan operator.
7. Operator aritmatika dalam bahasa pemrograman Pascal juga menerapkan derajat operasi seperti di matematika. Jika di urutkan berdasarkan derajat operasinya maka operator aritmatika dapat dirutkan sebagai berikut :
- Operator perkalian (*), Operator Pembagian (/), Operator Pembagian bulat (div), Operator Modulus (mod)
 - Operator penjumlahan (+) dan Operator Pengurangan (-)

1.1.4. Tugas

Setelah mengikuti kegiatan belajar di atas, berikutnya siswa bisa memperdalam pengetahuannya dan berlatih membuat program sendiri untuk mengasah kemampuan pembuatan program yang memanfaatkan operator aritmatika. Contoh program sederhana misalnya membuat kalkulator sederhana atas kreativitas siswa sendiri.

1.1.5. Uji Kompetensi

Kerjakan soal-soal berikut.

1. Apakah yang dimaksud dengan Operand dan Operator ?
2. Jelaskan perbedaan antara operator aritmatika biner dan operator aritmatika unair!
3. Sebutkan jenis-jenis dari operator aritmatika biner? jelaskan fungsinya!
4. Jelaskan perbedaan operator relasional “<” dengan operator relasional “<=”. Berikan contoh penggunaannya dalam FreePascal!
5. Jelaskan perbedaan antara operator aritmatika “/” dengan operator aritmatika “mod”!

1.2. Kegiatan Belajar 2. Pemanfaatan Operator Aritmatika

Alokasi Waktu : 2 x 45 menit

1.2.1. Tujuan Pembelajaran

Tujuan pembelajaran pada Kegiatan Belajar 2 tentang Pemanfaatan Operator Aritmatika adalah :

1. Menggunakan definisi operator aritmatika dalam penyelesaian masalah sehari-hari, seperti aplikasi bisnis atau permasalahan lainnya.

1.2.2. Aktivitas Belajar Siswa

1.2.2.1. Mengamati

Pada kegiatan belajar 1, kita telah mempelajari berbagai jenis operator relasi dan operator aritmatika. Kita sering menjumpai berbagai macam masalah yang berkaitan dengan operasi aritmatika sederhana seperti penjumlahan, pengurangan, pembagian dan perkalian. Namun kebanyakan dari kita menyelesaikan masalah tersebut dengan pengoperasian secara manual saja. Proses pengoperasian manual cenderung tidak efektif dan efisien. Untuk menghemat waktu dan mendapatkan hasil yang akurat diperlukan suatu program yang dapat memudahkan dalam menyelesaikan masalah tersebut.

Setelah kita memahami penggunaan operator aritmatika maka pada kegiatan belajar kali ini akan membahas mengenai pemanfaatan operator aritmatika untuk menyelesaikan berbagai masalah sehari-hari dengan memanfaatkan Freepascal.

Misalnya kita diminta menyelesaikan permasalahan dibawah ini menggunakan program pascal :

Sebuah perusahaan komputer memiliki 5 unit penjualan dengan setiap unit memiliki laba per bulannya adalah sebagai berikut :

unit 1 = Rp 2.456.750,00

unit 2 = Rp 2.738.900,00

unit 3 = Rp 1.578.000,00

unit 4 = Rp 1.875.000,00

unit 5 = Rp 2.123.000,00

akan dicari laba total dan rata-rata laba dari penjualan produk perusahaan komputer tersebut.

Tujuan dari penyelesaian masalah di atas adalah untuk mencari jumlah dan rata-rata laba dari perusahaan komputer. Untuk menyelesaikan masalah tersebut dengan bahasa pemrograman Pascal, kita perlu merancang program yang dapat menghitung jumlah dan rata-rata laba dari kelima unit penjualan perusahaan komputer tersebut.

Langkah pertama yang harus dilakukan adalah menentukan input dan tipe data yang dibutuhkan program yang akan dibuat. Apabila kita cermati, input dari masalah di atas adalah laba dari setiap unit penjualan. Dari masalah tersebut kita dapat memisalkan :

Variabel a = laba dari unit penjualan 1

Variabel b = laba dari unit penjualan 2

Variabel c = laba dari unit penjualan 3

Variabel d = laba dari unit penjualan 4

Variabel e = laba dari unit penjualan 5

Dan variabel variabel tersebut bertipe integer. Kemudian kita harus mentranslasikan input di atas menjadi kode yang dikenali oleh FreePascal, seperti di bawah ini :

```
//deklarasi tipe variabel  
var a,b,c,d,e:integer;  
rata,jumlah:real;
```

```
//input data  
  
write('Masukkan laba dari toko a = Rp.');//  
readln(a);  
  
write('Masukkan laba dari toko b = Rp.');//  
readln(b);  
  
write('Masukkan laba dari toko c = Rp.');//  
readln(c);  
  
write('Masukkan laba dari toko d = Rp.');//  
readln(d);  
  
write('Masukkan laba dari toko e = Rp.');//  
readln(e);
```

Setelah menentukan input dari program yang akan dibuat maka langkah selanjutnya kita harus menentukan cara atau metode untuk mengolah data input menjadi data yang dikehendaki. Dalam masalah ini, data yang ingin dicari / ditampilkan adalah rata-rata dan jumlah laba dari semua unit penjualan. Untuk memperoleh rata-rata dan jumlah laba maka kita harus melakukan perhitungan sebagai berikut :

$$\sum X = x_1 + x_2 + \cdots + x_n$$

$$\bar{X} = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

Maka dalam konteks masalah di atas, untuk mencari rata-rata dan total laba dari semua unit penjualan adalah

$$\text{Total laba} = \text{Laba Unit 1} + \text{Laba Unit 2} + \text{Laba Unit 3} + \text{Laba Unit 4} + \text{Laba Unit 5}$$

$$\text{Rata-rata laba} = \text{Total laba} : 5$$

Kemudian kita translasikan perhitungan di atas menjadi kode pascal :

```
//Menhitung Total Laba
jumlah :=(a+b+c+d+e);

//Menampilkan Total laba
writeln('Jumlah laba dari kelima toko tersebut adalah
Rp',jumlah:6:0);

// Menghitung Rata-rata Laba
rata :=((a+b+c+d+e)/5);

// Menampilkan Rata-rata Laba
writeln('laba rata-rata dari kelima toko tersebut adalah
Rp',rata:6:0);
```

Setelah komponen deklarasi variabel, input dan proses pengolahan variabel telah dibuat maka langkah selanjutnya adalah merangkai semua komponen menjadi satu dan melengkapinya dengan komponen tambahan agar program dapat berjalan dan ditampilkan dengan baik. Di bawah ini adalah sourcecode yang lengkap dari program menghitung rata-rata serta total laba dari perusahaan komputer tersebut :

```
program rata2;
uses crt;
```

```
//deklarasi variabel yang digunakan
var a,b,c,d,e,rata,jumlah:real;

begin
    {Program untuk menghitung rata-rata dan jumlah laba}
    //menghapus tampilan layar awal
    clrscr;

    writeln('=====');
    writeln('Program Perhitungan Rata-rata dan Jumlah Laba');
    writeln('=====');
    writeln('');
    //input laba toko pertama
    write('Masukkan laba dari dealer a = Rp'); readln(a);
    writeln('');

    //input laba toko kedua
    write('Masukkan laba dari dealer b = Rp'); readln(b);
    writeln('');

    //input laba toko ketiga
    write('Masukkan laba dari dealer c = Rp'); readln(c);
    writeln('');

    //input laba toko keempat
    write('Masukkan laba dari dealer d = Rp'); readln(d);
    writeln('');

    //input laba toko kelima
    write('Masukkan laba dari dealer e = Rp'); readln(e);
    writeln('');

    writeln('=====');
    writeln('');
    jumlah :=(a+b+c+d+e+f+g+h+i+j);
    writeln('Jumlah laba dari perusahaan motor X tersebut adalah
Rp',jumlah:6:0);
    writeln('');

    rata :=((a+b+c+d+e)/5);
    writeln('laba rata-rata dari semua dealer penjualan adalah
Rp',rata:6:0);
    writeln('');
    writeln('=====');
    readln;
end.
```

1.2.2.2. Menanya

Berdasarkan kegiatan mengamati, ada beberapa hal yang bisa diperhatikan dan dibahas lebih lanjut. Beberapa pertanyaan yang berkaitan dengan operator aritmatika adalah:

1. Dari program mencari rata-rata dan total laba yang telah di amati sebelumnya, bagaimanakah jika jumlah unit penjualan diperbanyak? Apakah merubah susunan sourcecode yang dibuat?

2. Apakah kita juga dapat membuat program untuk menyelesaikan permasalahan bisnis lainnya dengan memanfaatkan operator aritmatika, misalnya : program menghitung diskon harga jual dari suatu barang?

1.2.2.3. Mencoba

Percobaan 1

Sebuah Perusahaan Motor "X" memiliki 8 dealer pemasaran di Provinsi Jawa Tengah. Setiap dealer penjualan menghasilkan laba penjualan yang berbeda-beda setiap bulannya. Buatlah sebuah program yang digunakan untuk menghitung total laba dan laba rata-rata dari setiap dealer dalam kurun waktu 1 bulan?

```
{Program u/ hitung rata-rata dan jumlah laba}
program Laba;

uses crt;

//deklarasi variabel yang digunakan
var a,b,c,d,e,f,g,h,rata,jumlah:real;

begin
    //menghapus tampilan layar awal
    clrscr;

    writeln('=====');
    writeln('Program Perhitungan Rata-rata dan Jumlah Laba');

    writeln('=====');
    writeln('');

    //input laba dealer pertama
    write('Masukkan laba dari dealer a = Rp'); readln(a);
    writeln('');

    //input laba dealer kedua
    write('Masukkan laba dari dealer b = Rp'); readln(b);
    writeln('');

    //input laba dealer ketiga
    write('Masukkan laba dari dealer c = Rp'); readln(c);
    writeln('');

    //input laba dealer keempat
    write('Masukkan laba dari dealer d = Rp'); readln(d);
    writeln('');

    //input laba dealer kelima
    write('Masukkan laba dari dealer e = Rp'); readln(e);
    writeln('');

    //input laba dealer keenam
    write('Masukkan laba dari dealer f = Rp'); readln(f);
    writeln('');

    //input laba dealer ketujuh
    write('Masukkan laba dari dealer g = Rp'); readln(g);
```

```
writeln('');

//input laba dealer kedelapan
write('Masukkan laba dari dealer h = Rp'); readln(h);
writeln('');

jumlah :=(a+b+c+d+e+f+g+h+i+j);
rata :=((a+b+c+d+e+f+g+h)/8);

writeln('=====');
writeln('');
write('Jumlah laba perusahaan motor X : Rp');
writeln(jumlah:6:0);
writeln('');
write('Laba rata-rata dari semua dealer penjualan : Rp');
writeln(rata:6:0);
writeln('');
writeln('=====');
readln;
end.
```

Hasil Percobaan Program Laba:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 2

Toko sepatu “Italy” akan memberikan diskon harga sebesar 10% untuk semua jenis sepatu yang dijual di toko tersebut. Misalnya, Ali membeli sepasang sepatu ditoko tersebut dengan harga sebelum didiskon adalah Rp 250.000,00. Berapakah harga sepatu Ali setelah dikenai diskon sebesar 10%?

```
{Program Untuk menghitung diskon dan harga jual sepatu}
program Diskon;

uses crt;

//deklarasi variabel
var a,b,c:real;

begin
    clrscr;

    //judul program
    writeln('Program Perhitungan Diskon Harga Sepatu');
    writeln('');

    //input harga sepatu
    write('Masukkan Harga Awal Sepatu = '); readln(a);

    //menghitung besaran diskon
    b:=(10/100)*a;
    writeln('           Besaran diskon = ',b:6:0);

    //menghitung harga jual sepatu setelah didiskon
    c:=a-b;
    writeln('           Harga jual = ',c:6:0);
    readln;
end.
```

Hasil Percobaan Program Diskon:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

1.2.2.4. Mengasosiasi

Setelah siswa melakukan percobaan yang sesuai dengan pertanyaan yang ada, siswa diharapkan melakukan kegiatan menalar yang bisa dilakukan secara individu oleh masing-masing siswa atau bisa juga berkelompok. Beberapa hal yang bisa dijadikan arahan untuk menalar:

1. Perhatikan hasil kompilasi, apakah berhasil atau justru terjadi kesalahan.
2. Perhatikan urutan proses dari setiap program yang sudah dicoba.
3. Cobalah membuat kesimpulan dari percobaan yang telah dilakukan.

1.2.3. Rangkuman

Dari percobaan-percobaan yang telah dilakukan, bisa ambil kesimpulan bahwa:

1. Operator Aritmatika dapat digunakan untuk menyelesaikan berbagai masalah yang berkaitan dengan kehidupan sehari-hari.
2. Urutan langkah dalam membuat program yang digunakan untuk menghitung rata-rata maupun total laba dari suatu perusahaan antara lain :
 - a. Menentukan jenis variabel yang digunakan
 - b. Menentukan input data yang akan diproses
 - c. Menentukan ekspresi matematika yang dapat dikaitkan dengan masalah yang dihadapi
 - d. Mengubah ekspresi matematika menjadi *code* yang dapat dipahami oleh *pascal*
 - e. Merangkai semua kode menjadi satu kesatuan program.
3. Selain untuk menghitung rata-rata dan total laba, Operator aritmatika juga dapat digunakan untuk menghitung besaran diskon dari suatu barang yang dijual, dll.

1.2.4. Tugas

Setelah mengikuti kegiatan belajar di atas, berikutnya siswa bisa memperdalam pengetahuannya dan berlatih membuat program sendiri untuk mengasah kemampuan pembuatan program yang memanfaatkan operator aritmatika. Contoh program sederhana misalnya membuat program untuk menghitung tambahan gaji pada sebuah perusahaan.

1.2.5. Uji Kompetensi

1. Buatlah program untuk menghitung besaran diskon dari sebuah benda yang dijual jika diketahui harga sebelum dan sesudah didiskon.
2. Buatlah program untuk mencari total laba dari sebuah perusahaan yang memiliki banyak cabang jika diketahui rata-rata laba dan jumlah cabang.

1.3. Kegiatan Belajar 3. Fungsi Aritmatika

Alokasi Waktu : 3 x 45 menit

1.3.1. Tujuan Pembelajaran

Tujuan pembelajaran pada Kegiatan Belajar 3 tentang Fungsi Aritmatika adalah :

1. Siswa dapat menjelaskan pengertian fungsi aritmatika.
2. Siswa dapat menjelaskan cara kerja dari fungsi aritmatika dalam suatu program.
3. Siswa dapat menggunakan fungsi aritmatika dalam program yang sederhana.

1.3.2. Aktivitas Belajar Siswa

1.3.2.1. Mengamati

Pada Kegiatan Belajar 1 kita telah belajar tentang berbagai jenis operator aritmatika beserta penggunaannya dalam sebuah program. Kita dapat melakukan berbagai operasi dasar dengan menggunakan operator aritmatika seperti penjumlahan, pengurangan, pembagian dan perkalian. Namun bagaimana jika kita dihadapkan dengan ekspresi matematika di bawah ini?.

- $\sin \theta$
- $\cos \theta$
- $a^2 + b^2 = c^2$
- $A = \pi r^2$

Kita akan kesulitan mengakomodasi permasalahan di atas dalam FreePascal jika hanya menggunakan operator aritmatika yang telah dipelajari sebelumnya. Maka FreePascal mengakomodasi permasalahan tersebut dengan menyediakan berbagai bentuk fungsi matematika yang lazim digunakan dalam kehidupan sehari-hari. Berikut contoh-contoh fungsi aritmatika pada freepascal yang sering digunakan :

ABS

ABS merupakan fungsi matematika dalam pascal yang berguna untuk menghasilkan nilai mutlak dari sebuah bilangan negatif . Sintax : ABS (x)

FRAC

Merupakan fungsi matematika dalam Pascal yang berguna untuk menghasilkan nilai desimal dari bilangan real. Sintax : FRAC (x)

INT

Berguna untuk membulatkan angka pecahan menjadi sebuah bilangan bulat positif. Sintax : INT (x) .

LN

Berfungsi untuk menghasilkan nilai logaritma dari sebuah bilangan. Sintax : `Ln (x)`

SIN

Sintaks sin berfungsi untuk menghitung nilai sinus dari sebuah bilangan yang biasanya digunakan untuk menghitung sudut tertentu dari 0 derajat samapi 360 derajat pada operasi matematika. Sintax : `sin (x)`.

SQR

Berfungsi untuk melakukan perhitungan kuadrat dari sebuah bilangan. Sintax : `SQR (x)`

SQRT

Berfungsi untuk melakukan penghitungan akar kuadrat dari sebuah bilangan. Sintax : `Sqrt (x)`

PRED

Perintah PRED digunakan untuk menghasilkan nilai yang lebih kecil darinya. Perintah PRED hanya dapat digunakan pada tipe data numerik, character dan *boolean*. Sintax : `PRED (x)`

SUCC

Perintah SUCC digunakan untuk menghasilkan nilai yang lebih besar darinya. Sama seperti perintah PRED, SUCC hanya dapat digunakan pada tipe data numerik, karakter dan *boolean*. Sintax : `SUCC (x)`

ODD

Digunakan untuk menentukan sebuah bilangan apakah termasuk bilangan ganjil atau bukan. Output dari operasi ini adalah *boolean*. Sintax : `ODD (x)`

INC

Digunakan untuk menambah nilai bilangan numerik sebesar 1. Sintax : `INC (x)`

DEC

Digunakan untuk mengurangi nilai bilangan numerik sebesar 1. Sintax : `DEC (x)`

1.3.2.2. Menanya

Berdasarkan kegiatan mengamati, ada beberapa hal yang bisa diperhatikan dan dibahas lebih lanjut. Beberapa pertanyaan yang berkaitan dengan operator aritmatika adalah:

- Apakah fungsi aritmatika pada Freepascal dapat digunakan secara bersama-sama dengan operator aritmatika yang telah kita ketahui sebelumnya?
- Bagaimakah cara menggunakan fungsi aritmatika pada Freepascal untuk menghitung sisi miring dari suatu segitiga siku-siku?
- Bagaimakah cara kerja dari fungsi aritmatika increment dan decrement pada suatu program?

1.3.2.3. Mencoba

Percobaan 1

Untuk mengetahui jawaban dari pertanyaan pertama yaitu apakah fungsi aritmatika pada Freepascal dapat digunakan secara bersama-sama dengan operator aritmatika yang telah kita ketahui sebelumnya?, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program AritFunct1;
uses crt;
var x,y:integer;
    z:integer;
begin
    clrscr;
    writeln('Program Menghitung Nilai abs(y) + sqr(x) ');
    x:=2;
    writeln('x = ',x);
    y:=-3;
    writeln('y = ',y);
    z:=ABS(y)+SQR(x); //definisi variabel z
    writeln('=====');
    writeln('Hasil dari ABS(y) + SQR(x) = ',z);
    readln;
end.
```

Hasil Percobaan Program AritFunct1:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 2

Untuk mengetahui jawaban dari pertanyaan kedua yaitu bagaimakah cara menggunakan fungsi aritmatika pada Freepascal untuk menghitung sisi miring dari suatu segitiga siku-siku?, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program AritFunct2;
uses crt;
var a,b,c:real;
begin
  clrscr;
  writeln('=====');
  writeln('Program Hitung Sisi Miring Segitiga Siku-Siku');

  writeln;
  writeln('misalkan :');
  writeln('a = sisi tegak pertama');
  writeln('b = sisi tegak kedua');
  writeln('c = sisi miring segitiga');
  writeln('=====');
  writeln;
  writeln('Masukkan nilai a = ');readln(a);
  writeln('Masukkan nilai b = ');readln(b);
  writeln;
  writeln('=====');

  c:=sqrt(sqr(a)+sqr(b));

  writeln('c= ',c:4:2);
  writeln;
  writeln('Maka panjang sisi miring segitiga : ',c:4:2);

  readln;
end.
```

Hasil Percobaan AritFunct2:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 3

Untuk mengetahui jawaban dari pertanyaan pertama yaitu Bagaimanakah cara kerja dari fungsi aritmatika increment dan decrement pada suatu program?, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

Program 1. AritFunct3

```
program AritFunct3;
uses crt;
var x:integer; //deklarasi 25ariable x

begin
    clrscr;
    x:=1;

    writeln('Perintah Increment dengan while');

    while x<=10 do //25selama x ≤ 10
    begin
        write(x,' ');
        INC(x); //pengenaan fungsi increment pada x
    end;

    writeln;
    writeln('Perintah Decrement dengan while');

    while x>=1 do //25selama x ≥ 1
    begin
        write(x,' ');
        DEC(x); //pengenaan fungsi increment pada x
    end;
    readln;
end.
```

Hasil Percobaan Program AritFunct:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

1.3.2.4. Mengasosiasi

Setelah siswa melakukan percobaan yang sesuai dengan pertanyaan yang ada, siswa diharapkan melakukan kegiatan menalar yang bisa dilakukan secara individu oleh masing-masing siswa atau bisa juga berkelompok. Beberapa hal yang bisa dijadikan arahan untuk menalar:

1. Perhatikan hasil kompilasi, apakah berhasil atau justru terjadi kesalahan.
2. Perhatikan urutan proses dari setiap program yang sudah dicoba.
3. Coba bandingkan beberapa program yang memiliki kemiripan, program mana yang berhasil dikompilasi dan program mana yang tidak berhasil dikompilasi.
4. Perhatikan letak/posisi kesalahan, coba pikirkan mengapa terjadi kesalahan pada posisi tersebut.
5. Cobalah membuat kesimpulan dari percobaan yang telah dilakukan.

1.3.3. Rangkuman

Dari percobaan-percobaan yang telah dilakukan, bisa ambil kesimpulan bahwa:

1. Fungsi aritmatika adalah suatu fungsi matematika sederhana yang merupakan pengembangan dari beberapa operator aritmatika yang telah disediakan oleh FreePascal.
2. Jenis jenis dari fungsi aritmatika yang termuat di dalam FreePascal :
 - ABS
ABS merupakan fungsi matematika dalam pascal yang berguna untuk menghasilkan nilai mutlak dari sebuah bilangan negatif . Sintax : ABS(x)
 - FRAC
Merupakan fungsi matematika dalam Pascal yang berguna untuk menghasilkan nilai desimal dari bilangan real. Sintax : FRAC(x)
 - INT
Berguna untuk membulatkan angka pecahan menjadi sebuah bilangan bulat positif. Sintax : INT(x).

- LN
Berfungsi untuk menghasilkan nilai logaritma dari sebuah bilangan. Sintax : Ln(x)
- SIN
Sintaks sin berfungsi untuk menghitung nilai sinus dari sebuah bilangan yang biasanya digunakan untuk menghitung sudut tertentu dari 0 derajat samapi 360 derajat pada operasi matematika. Sintax : Sin(x).
- SQR
Berfungsi untuk melakukan perhitungan kuadrat dari sebuah bilangan. Sintax : SQR(x)
- SQRT
Berfungsi untuk melakukan penghitungan akar kuadrat dari sebuah bilangan. Sintax : Sqrt(x)
- PRED
Fungsi PRED digunakan untuk menghasilkan nilai yang lebih kecil darinya. Perintah PRED hanya dapat digunakan pada tipe data numerik, character dan boolean. Sintax : PRED(x)
- SUCC
Perintah SUCC digunakan untuk menghasilkan nilai yang lebih besar darinya. Sama seperti perintah PRED, SUCC hanya dapat digunakan pada tipe data numerik, karakter dan boolean. Sintax : SUCC(x)
- ODD
Digunakan untuk menentukan sebuah bilangan apakah termasuk bilangan ganjil atau bukan. Output dari operasi out adalah boolean. Sintax : ODD(x)

1. Penggunaan fungsi aritmatika dapat digabungkan dengan berbagai operator aritmatika yang tersedia di dalam FreePascal. Contohnya : SQRT (x) + SQR (y)
2. Cara kerja dari fungsi INC adalah menambahkan nilai sebesar 1 pada variabel awal sampai batas variabel yang dikehendaki programmer.
3. Cara kerja dari fungsi DEC adalah mengurangi variabel awal sebesar 1 sesuai dengan batas-batas yang dikehendaki.

1.3.4. Tugas

Setelah mengikuti kegiatan belajar di atas, berikutnya siswa bisa memperdalam pengetahuannya dan berlatih membuat program sendiri untuk mengasah kemampuan pembuatan program yang memanfaatkan fungsi aritmatika. Contoh program sederhana untuk menghitung luas suatu lingkaran.

1.3.5. Uji Kompetensi

1. Sebut dan jelaskan jenis-jenis fungsi aritmatika dalam FreePascal!
2. Pada suatu program, Apakah kita dapat menggunakan lebih dari satu fungsi aritmatika secara bersamaan?
3. Apakah perbedaan dari fungsi INC dan DEC?

1.4. Kegiatan Belajar 4. Pemanfaatan Fungsi Aritmatika

Alokasi Waktu : 2 x 45 menit

1.4.1. Tujuan Pembelajaran

Tujuan pembelajaran pada Kegiatan Belajar 4 tentang Pemanfaatan Fungsi Aritmatika adalah :

1. Terampil membuat program yang menggunakan fungsi aritmatika.
2. Menggunakan definisi fungsi aritmatika dalam penyelesaian masalah sehari-hari, seperti aplikasi bisnis atau permasalahan lainnya.

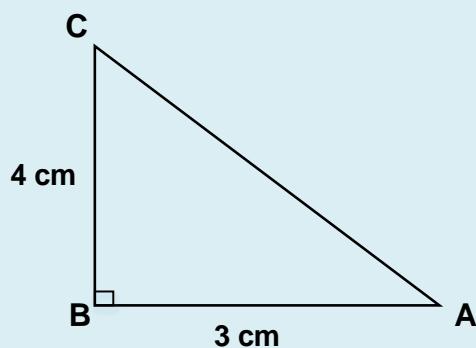
1.4.2. Aktivitas Belajar Siswa

1.4.2.1. Mengamati

Pada kegiatan belajar 3, kita telah mempelajari berbagai jenis fungsi aritmatika. Seperti halnya operator aritmatika, fungsi aritmatika juga dapat kita manfaatkan untuk mnyelesaikan permasalahan sehari-hari.

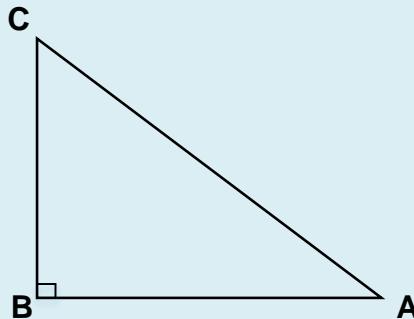
Di bawah ini merupakan contoh masalah yang dapat kita selesaikan dengan memanfaatkan fungsi aritmatika pada FreePascal :

Masalah 1 :



Diketahui : Segitiga ABC dengan sudut siku-siku $\angle ABC$. Panjang AB = 3 cm dan panjang BC = 4 cm. Tentukan panjang AC.

Permasalahan di atas meminta kita untuk menentukan panjang AC. Untuk mencari panjang AC dapat memanfaatkan teorema Phytagoras yang telah dipelajari di bangku sekolah dasar. Coba kalian ingat, bagaimanakah bentuk umum dari teorema *Phytagoras*?



$$AC^2 = AB^2 + BC^2$$

$$AC = \sqrt{AB^2 + BC^2}$$

Ilustrasi di atas merupakan bentuk umum dari teorema phytagoras yang akan digunakan untuk mencari panjang sisi miring dari masalah 1. Setelah itu kita harus mengaitkan bentuk umum dari teorema *Phytagoras* dengan fungsi-fungsi aritmatika yang telah dipelajarari.

Dari bentuk umum *Phytagoras*, apabila diamati bentuk tersebut secara garis besar memanfaatkan bentuk kuadrat dan bentuk akar. Dan dalam bahasa pemrograman *Pascal* juga dulu tentang fungsi SQR (x) dan SQRT (x) yang mana kedua fungsi itu mengacu pada operasi kuadrat dan operasi akar dari suatu bilangan.

Setelah itu, kita harus menyusun *sourcecode* yang sesuai dengan masalah 1. Berikut ini alternatif code yang bisa digunakan untuk menyelesaikan masalah 1 :

```
program Segitiga;
uses crt;

//deklarasi variabel
var      a,b:integer;
        c:real;

begin
  clrscr;
  //judul program
  writeln('=====');
  writeln('Hitung sisi miring segitiga siku-siku');
  writeln('=====');
  writeln;

  //pemberian nilai variabel a
  a:=3;
  //menampilkan isi variabel a
  writeln('AB = ',a);
```

```

//pemberian nilai variabel b
b:=4;
//menampilkan isi variabel b
writeln('BC = ',b);
writeln('');
writeln('=====');

//operasi phytagoras
c:=sqrt(sqr(a)+sqr(b));

//menampilkan AC
writeln('AC = ',c:1:0);
writeln('=====');
readln;
end.

```

1.4.2.2. Menanya

Berdasarkan kegiatan mengamati, ada beberapa hal yang bisa diperhatikan dan dibahas lebih lanjut. Beberapa pertanyaan yang berkaitan dengan operator aritmatika adalah:

1. Apakah fungsi aritmatika dapat dimanfaatkan untuk menyelesaikan permasalahan yang terkait jarak antara dua buah titik?
2. Apakah fungsi aritmatika dapat digunakan untuk menyelesaikan permasalahan bisnis yang ada dalam kehidupan sehari-hari, misalnya tentang penentuan biaya pembangunan jalan tol?

1.4.2.3. Mencoba

Percobaan 1

Dua titik A (x_1, y_1) dan B (x_2, y_2). Kedua titik A dan B akan dihubungkan dengan sebuah garis sehingga membentuk ruas garis AB. Tentukan panjang ruas garis AB.

Untuk mencari jarak antara dua titik dapat digunakan rumus

$$\text{Jarak} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Keterangan:

Jarak = jarak dua titik

x_1 = nilai x pada titik pertama

x_2 = nilai x pada titik kedua

y_1 = nilai y pada titik pertama

y_2 = nilai y pada titik kedua

Untuk mengetahui jawaban dari pertanyaan Apakah fungsi aritmatika dapat memecahkan masalah terkait dengan jarak antara dua titik. Salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Jarak;

uses crt;

var      x1,x2,y1,y2:integer;
         a:real;

begin

  clrscr;

  writeln('=====');
  writeln('Menghitung Jarak Antara 2 Titik');
  writeln('=====');
  writeln;

  //input variabel yang digunakan untuk titik A
  writeln('Masukkan koordinat titik A(x1,y1)');
  write('x1 = '); readln(x1);
  write('y1 = '); readln(y1);
  writeln;

  //input variabel yang digunakan untuk titik B
  writeln('Masukkan koordinat titik B(x2,y2)');
  write('x2 = '); readln(x2);
  write('y2 = '); readln(y2);
  writeln;

  //Menghitung jarak titik A dan B
  a:= SQRT(SQR(x2-x1)+SQR(y2-y1));

  //Menampilkan hasil perhitungan jarak A dan B
  writeln('=====');
  writeln('Jarak Titik A dengan B : ',a:4:2,' satuan');
  writeln('=====');

  readln();

end.
```

Hasil Percobaan Jarak :

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 2

Untuk mengetahui jawaban dari pertanyaan Apakah fungsi aritmatika dapat memecahkan masalah terkait dengan masalah bisnis. Salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

Dua kota A berlokasi di titik (x_1, y_1) dan kota B berlokasi di titik (x_2, y_2) akan dihubungkan dengan jalan tol. Biaya pembangunan jalan per kilometer sebesar Rp 100 juta. Akan dicari biaya minimal yg harus disiapkan untuk membangun jalan tol tersebut.

```
program Tol;
uses crt;
var      x1,x2,y1,y2:integer;
         a,b:real;
begin
  clrscr;
  writeln('=====');
  writeln('Biaya Minimum Pembuatan Jalan Tol');
  writeln('Biaya dasar Rp 100juta/km');
  writeln('=====');
  writeln('');

  //input variabel yang digunakan untuk Kota A
  writeln('Masukkan koordinat Kota A(x1,y1)');
  write('x1 = '); readln(x1);
  write('y1 = '); readln(y1);
  writeln;
```

```
//input variabel yang digunakan untuk Kota B  
writeln('Masukkan koordinat Kota B(x2,y2)');  
write('x2 = '); readln(x2);  
write('y2 = '); readln(y2);  
writeln;  
  
//Menghitung jarak titik A dan B  
a:= SQRT(SQR(x2-x1)+SQR(y2-y1));  
//Menghitung biaya minimum  
b:= (1000*a)*100;  
  
writeln('=====');  
//Menampilkan hasil perhitungan jarak A dan B  
writeln('Jarak Kota A dengan B : ',a:4:2,' km');  
//menampilkan Hasil perhitungan biaya  
writeln('Biaya minimum : ',b:6:0,' juta rupiah');  
writeln('=====');  
readln();  
end.
```

Hasil Percobaan Program Tol :

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

1.4.2.4. Mengasosiasi

Setelah siswa melakukan percobaan yang sesuai dengan pertanyaan yang ada, siswa diharapkan melakukan kegiatan menalar yang bisa dilakukan secara individu oleh masing-masing siswa atau bisa juga berkelompok. Beberapa hal yang bisa dijadikan arahan untuk menalar:

1. Perhatikan hasil kompilasi, apakah berhasil atau justru terjadi kesalahan.

2. Perhatikan urutan proses dari setiap program yang sudah dicoba.
3. Coba bandingkan beberapa program yang memiliki kemiripan, program mana yang berhasil dikompilasi dan program mana yang tidak berhasil dikompilasi.
4. Perhatikan letak/posisi kesalahan, coba pikirkan mengapa terjadi kesalahan pada posisi tersebut.
5. Cobalah membuat kesimpulan dari percobaan yang telah dilakukan.

1.4.3. Rangkuman

Dari percobaan-percobaan yang telah dilakukan, bisa ambil kesimpulan bahwa:

1. Fungsi aritmatika dapat digunakan untuk menyelesaikan permasalahan yang terkait dengan penentuan jarak dua titik, panjang sisi miring segitiga siku-siku, dll.
2. Fungsi aritmatika juga dapat digunakan untuk menyelesaikan permasalahan bisnis misalnya menentukan biaya minimum pembuatan jalan tol, menghitung luas bangunan,dll.
3. Langkah- langkah membuat program *FreePascal* untuk menghitung biaya minimum pembuatan jalan tol :
 - a. Menentukan variabel yang akan digunakan.
 - b. Menghitung jarak antara dua kota yang dihubungkan dengan jalan tol tersebut. Perhitungan jarak dapat memanfaatkan fungsi aritmatika SQR dan SQRT
 - c. Menghitung biaya minimal yang dibutuhkan. Perhitungan dengan memanfaatkan jarak dua kota yang telah dicari sebelumnya dikalikan dengan biaya pembuatan jalan tol per meter persegi.

1.4.4. Tugas

Setelah mengikuti kegiatan belajar di atas, berikutnya siswa bisa memperdalam pengetahuannya dan berlatih membuat program sendiri untuk mengasah kemampuan pembuatan program yang memanfaatkan fungsi aritmatika dalam penyelesaian masalah bisnis. Contoh program untuk menghitung biaya minimum pembuatan jalan tol yang menghubungkan 3 kota dengan biaya pembuatan jalan tol 100 juta rupiah untuk 1 meter.

1.4.5. Uji Kompetensi

1. Buatlah program menggunakan fungsi aritmatika untuk menyelesaikan masalah berikut :

Diketahui kota P terletak pada P(3,6) dan kota Q terletak pada Q(6,8) akan dibuat jalan tol yang menghubungkan antara kota P dan kota Q, jika biaya per satuan jarak jalan tol adalah Rp 100.000.000,00 maka berapakah dana minimal yang harus disiapkan?

Susun laporan yang terdiri atas kode program, penjelasan program, dan output dari program.

2. Buatlah program menggunakan fungsi aritmatika untuk menyelesaikan masalah berikut :

Pak Pambudi memiliki sebidang tanah berbentuk persegi panjang dengan panjang 20 meter dan lebar 38 meter. Beliau berencana menjual tanah tersebut dengan harga Rp. 500.000,00 / m². Tentukan harga jual dari tanah tersebut.

Susun laporan yang terdiri atas kode program, penjelasan program, dan output dari program.

- Buatlah program menggunakan fungsi aritmatika untuk menyelesaikan masalah berikut :

“Intan Pari” swimming pool memiliki sebuah kolam renang berbentuk balok dengan panjang 100 meter, lebar 50 meter dan kedalaman 2 meter. Setiap hari kamis kolam renang tersebut akan dikuras dan diisi kembali dengan air bersih. Biaya pengisian air kolam renang tersebut adalah Rp. 50.000,00 untuk setiap m^3 air. Berapakah biaya yang dibutuhkan untuk satu kali pengisian penuh kolam renang tersebut?.

Susun laporan yang terdiri atas kode program, penjelasan program, dan output dari program.

1.5. Kegiatan Belajar 5. Operasi Logika

Alokasi Waktu : 3 x 45 menit

1.5.1. Tujuan Pembelajaran

Tujuan pembelajaran pada Kegiatan Belajar 5 tentang Operator Logika adalah :

- Siswa dapat menjelaskan pengertian Operator Logika.
- Siswa dapat menjelaskan cara kerja dari Operator Logika dalam suatu program.
- Siswa dapat menggunakan Operator Logika dalam program yang sederhana.

1.5.2. Aktivitas Belajar Siswa

1.5.2.1. Mengamati

Operator Logika

Operator logika digunakan untuk menentukan nilai boolean dari satu atau beberapa operand. Operator logika bekerja sebagaimana hukum-hukum logika yang ada dalam hukum matematika. Di dalam Free Pascal ada 4 buah operator logika yaitu : NOT, AND, OR dan XOR.

Operator	Operasi	Tipe Operand	Tipe Hasil
not	Negasi	Boolean	Boolean
and	Konjungsi	Boolean	Boolean
or	Disjungsi	Boolean	Boolean
xor	Disjungsi Eksklusif	Boolean	Boolean

Operator Bitwise

Operasi bit digunakan untuk melakukan manipulasi bit pada bilangan bertipe data Byte dan word. Operator bitwise mirip dengan operator aljabar yang mengolah data bertipe byte dan word sehingga menghasilkan nilai output berupa bilangan juga. Terdapat lima jenis operator bitwise yang dikenal dalam Pascal yaitu AND, OR, XOR, SHR, SHL.

Operator	Simbol	Operasi
Not	~	Bit not

and	&	Bit Konjungsi
or		Bit Disjungsi
Xor	!	Bit Eksklusif Disjungsi
Shl	<<	Bit <i>shift left</i>
Shr	>>	Bit <i>shift right</i>

Konversi Bilangan Biner

Bilangan biner terdiri atas dua jenis, yaitu true (1) dan false (0). Nilai true mewakili bilangan 1 atau kondisi saklar listrik dalam kondisi ON, sedangkan false mewakili bilangan 0 atau kondisi saklar listrik dalam kondisi OFF.

Mengubah bilangan desimal menjadi bilangan biner. Contoh : $100_{10} = (\dots \dots \dots)_2$. Berikut ini cara mengubah bilangan desimal menjadi bilangan biner :

$$\begin{array}{r}
 & 100 \\
 2 & \overline{)50} & 0 \\
 2 & \overline{)25} & 0 \\
 2 & \overline{)12} & 1 \\
 2 & \overline{)6} & 0 \\
 2 & \overline{)3} & 0 \\
 2 & \overline{)1} & 1
 \end{array}$$

Maka hasil konversinya adalah 1100100_2

Mengubah bilangan biner menjadi bilangan desimal. Contoh : $1100100_2 = (\dots \dots \dots)_{10}$. Berikut cara mengubah bilangan tersebut menjadi bilangan desimal :

$$\begin{aligned}
 1100100_2 &= (0 \times 2^0) + (0 \times 2^1) + (1 \times 2^2) + (0 \times 2^3) + (0 \times 2^4) + (1 \times 2^5) + (1 \times 2^6) \\
 &= 0 + 0 + 4 + 0 + 0 + 32 + 64 \\
 &= 100
 \end{aligned}$$

Maka hasil konversinya adalah 100.

1.5.2.2. Menanya

Berdasarkan kegiatan mengamati, ada beberapa hal yang bisa diperhatikan dan dibahas lebih lanjut. Beberapa pertanyaan yang berkaitan dengan operator aritmatika adalah:

1. Apakah Operator logika dapat digunakan untuk data yang bertipe karakter?
2. Bagaimanakah hasil operasi logika NOT dari variabel a yang bernilai boolean true?
3. Bagaimanakah hasil operasi logika AND, OR dan XOR antara variabel a yang bernilai true dan variabel b yang bernilai false?

4. Bagaimanakah hasil operasi Bitwise AND, OR, XOR, SHR dan SHL antara $a = 9$ dan $b = 10$?

1.5.2.3. Mencoba

Percobaan 1

Untuk mengetahui jawaban dari pertanyaan pertama yaitu apakah operator logika dapat digunakan untuk data yang bertipe karakter?, salin dan lengkapilah program berikut ke dalam Free Pascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Logikal;

uses crt;
var a,b:char;

begin
    clrscr;
    a:='a';
    b:='b';

    //Pengandaian If untuk menampilkan hasil operasi logika a
    and b
    if(a and b) then
        writeln('Line 1 - Condition is true')
    else
        writeln('Line 1 - Condition is not true');

    //Pengandaian If untuk menampilkan hasil operasi logika a
    OR b
    if(a or b) then
        writeln('Line 2 - Condition is true')
    else
        writeln('Line 2 - Condition is not true');

    readln;
end.
```

Hasil Percobaan Program Logika1:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....

Letak kesalahan (baris, kolom)	Terjadi karena

Percobaan 2

Untuk mengetahui jawaban dari pertanyaan kedua yaitu bagaimakah hasil operasi logika NOT dari variabel a yang bernilai boolean true, salin dan lengkapilah program berikut ke dalam Free Pascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Logika2;
uses crt;
var a,b:boolean;
begin
    clrscr;
    a:=true; //nilai variabel a
    b:=false; //nilai variabel b

    //Pengandaian if untuk menampilkan nilai dari not(a)
    if NOT(a) then
        writeln('nilai dari not(a) adalah',not(a));

    //Pengandaian if untuk menampilkan
    if NOT(b) then
        writeln('nilai dari not(b) adalah',not(b));

    readln;
end.
```

Hasil Percobaan Program Logika2:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 3

Untuk mengetahui jawaban dari pertanyaan kedua yaitu bagaimanakah hasil operasi logika AND, OR dan XOR antara variabel a yang bernilai true dan variabel b yang bernilai false,salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Logika3;

uses crt;
var a,b:boolean;

begin
    clrscr;
    a:=true;
    b:=false;

    if(a and b) then
        writeln('Line 1 - Condition is true')
    else
        writeln('Line 1 - Condition is not true');

    if(a or b) then
        writeln('Line 2 - Condition is true')
    else
        writeln('Line 2 - Condition is not true');

    if(a xor b) then
        writeln('Line 3 - Condition is true')
    else
        writeln('line 3 - Condition is not true');

    readln;
end.
```

Keterangan :

Cobalah mengganti nilai dari
a. var a = false dan b = true

- b. var a = true dan b = true
- c. var a = false dan b = false

Kemudian bandingkan hasil dari masing-masing program yang kalian buat.

Hasil Percobaan Program Logika3:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 4

Untuk mengetahui jawaban dari pertanyaan kedua yaitu Bagaimanakah hasil operasi Bitwise AND, OR, XOR, SHR dan SHL antara $a = 9$ dan $b=10$?, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Logika4;
uses crt;
var a,b,c,d,e,f,g:integer;
begin
    clrscr;
    a:=9;
    b:=10;
    writeln('a = ',a);
    writeln('b = ',b);
    writeln('=====');
    //melakukan operasi bitwise AND
    c:=a And b;
    writeln('a and b = ',c);
    writeln('=====');
```

```
//melakukan operasi bitwise OR  
d:=a or b;  
writeln('a or b = ',d);  
writeln('=====');  
  
//melakukan operasi bitwise XOR  
e:=a xor b;  
writeln('a xor b = ',e);  
writeln('=====');  
  
//melakukan operasi bitwise shr  
f:=a shr 2;  
writeln('a shr 2 = ',f);  
writeln('=====');  
  
//melakukan operasi bitwise shl  
g:=b shl 2;  
writeln('b shl 2 = ',g);  
writeln('=====');  
  
readln;  
end.
```

Hasil Percobaan Program 4:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

1.5.2.4. Mengasosiasi

Setelah siswa melakukan percobaan yang sesuai dengan pertanyaan yang ada, siswa diharapkan melakukan kegiatan menalar yang bisa dilakukan secara individu oleh masing-masing siswa atau bisa juga berkelompok. Beberapa hal yang bisa dijadikan arahan untuk menalar:

1. Perhatikan hasil kompilasi, apakah berhasil atau justru terjadi kesalahan.
2. Perhatikan urutan proses dari setiap program yang sudah dicoba.

3. Coba bandingkan beberapa program yang memiliki kemiripan, program mana yang berhasil dikompilasi dan program mana yang tidak berhasil dikompilasi.
4. Perhatikan letak/posisi kesalahan, coba pikirkan mengapa terjadi kesalahan pada posisi tersebut.
5. Cobalah membuat kesimpulan dari percobaan yang telah dilakukan.

1.5.3. Rangkuman

Dari percobaan-percobaan yang telah dilakukan, bisa ambil kesimpulan bahwa:

1. Operator logika digunakan untuk menentukan nilai boolean dari satu atau beberapa operand. Operator logika bekerja sebagaimana hukum-hukum logika yang ada dalam hukum matematika.
2. Jenis-Jenis Operator Logika :

Operator	Operasi	Tipe Operand	Tipe Hasil
Not	Negasi	Boolean	Boolean
And	Konjungsi	Boolean	Boolean
Or	Disjungsi	Boolean	Boolean
Xor	Eksklusif Disjungsi	Boolean	Boolean

3. Operasi bit digunakan untuk melakukan manipulasi bit pada bilangan bertipe data Byte dan word. Operator bitwise mirip dengan operator aljabar yang mengolah data bertipe byte dan word sehingga menghasilkan nilai output berupa bilangan juga.

Operator	Simbol	Operasi
Not	\sim	Bit not
and	$\&$	Bit Konjungsi
or	$ $	Bit Disjungsi
Xor	$!$	Bit Eksklusif Disjungsi
Shl	$<<$	Bit shift left
Shr	$>>$	Bit shift right

4. Operator logika tidak dapat mengoperasikan data yang bertipe karakter dan string.
5. Tabel kebenaran Operator Logika pada FreePascal :

a	b	NOT(a)	a AND b	a OR b	a XOR b
True	True	False	True	True	False
True	False	False	False	True	True
False	True	True	False	True	True
False	False	True	False	False	False

6. Tabel Kebenaran Operator Bitwise :

a	b	NOT(a)	a AND b	a OR b	a XOR b
1	1	0	1	1	0
1	0	0	0	1	1
0	1	1	0	1	1

0	0	0	0	0	0
---	---	---	---	---	---

7. Misalkan $a = 9$ dan $b = 10$. Apabila kedua variabel a dan b dikenakan operasi SHL dan SHR maka :

Konversi bilangan desimal menjadi biner

$$a = 9 \rightarrow 1001$$

$$b = 10 \rightarrow 1010$$

9 SHR 2

$$1001 \xrightarrow{\text{menggeser nilai bit sebanyak 2 langkah ke kanan}} 0010 = 2$$

10 SHL 2

$$1010 \xrightarrow{\text{menggeser nilai bit sebanyak 2 langkah ke kiri}} 1000 = 40$$

1.5.4. Tugas

Setelah mengikuti kegiatan belajar di atas, berikutnya siswa bisa memperdalam pengetahuannya dan berlatih membuat program sendiri untuk mengasah kemampuan pembuatan program yang memanfaatkan operator Logika. Contoh : membuat program untuk menghitung nilai-nilai di bawah ini :

- 225 or 172
- 10 and 90
- 133 xor 99
- 67 shl 3
- 142 shr 2

1.5.5. Uji Kompetensi

Kerjakan soal-soal berikut ini.

- Konversikan bilangan berikut menjadi bilangan biner :
 - 6
 - 13
 - 87

- d. 111
 - e. 324
2. Tulis dan jelaskan operator logika yang kamu ketahui!
 3. Apakah perbedaan dari operator logika OR dengan Operator logika XOR?
 4. Jelaskan perbedaan antara operator logika dengan operator bitwise!
 5. Tuliskan dan jelaskan jenis-jenis operator bitwise!

Lembar kreativitas siswa pembuatan program tentang operasi aritmatika dan logika:

Program

Uses

Type

Var

Begin

End.

Output program:

BAB II

ARRAY

Kompetensi Dasar:

- Memahami penggunaan array 1 dimensi untuk penyimpanan data di memori
- Memahami penggunaan array 2 dimensi untuk penyimpanan data di memori
- Memahami penggunaan array multidimensi untuk penyimpanan data di memori
- Menyajikan kumpulan data berdimensi 1 dalam bentuk array
- Menyajikan kumpulan data berdimensi 2 dalam bentuk array

BAB II. ARRAY

Setelah mengenal data tunggal, yaitu sebuah variabel yang hanya bisa menyimpan satu data saja, kita akan belajar suatu struktur data yang mampu menyimpan beberapa data sekaligus. Dalam kehidupan sehari-hari kita mungkin pernah melihat sebuah loker yang memiliki banyak tempat penyimpanan, misalnya di tempat penitipan barang di supermarket. Untuk mempermudah mengenali setiap tempat penyimpanan pada loker tersebut biasanya diberi nomor urut. Setiap tempat pada loker memiliki kegunaan yang sama, yaitu sama-sama digunakan untuk menyimpan barang pengunjung. Struktur data yang memiliki konsep serupa dengan ini dalam pemrograman disebut dengan array. Array adalah sekumpulan data yang bertipe sama. Variabel bertipe array memiliki kemampuan untuk menyimpan sekumpulan data yang mempunyai tipe sama.



Gambar 2.1.Loker (*sumber www.smpia23.wordpress.com*)

2.1. Kegiatan Belajar 1. Array Satu Dimensi

Alokasi Waktu : 2 x 45 menit

2.1.1. Tujuan Pembelajaran

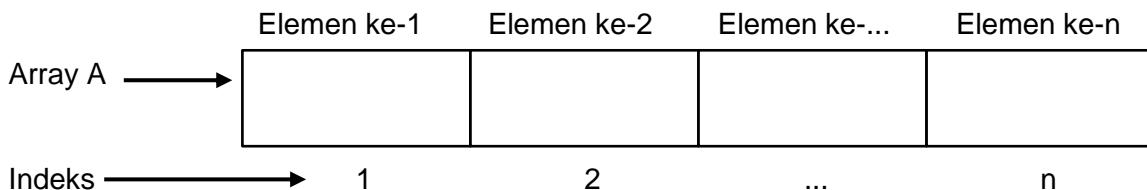
Setelah mempelajari materi pada kegiatan belajar ini, siswa diharapkan dapat memahami penggunaan array 1 dimensi untuk penyimpanan data di memori

2.1.2. Aktivitas belajar siswa

2.1.2.1. Mengamati/ observasi

Array dapat dianalogikan sebagai sebuah lemari atau loker yang memiliki sederetan kotak penyimpanan yang diberi nomor berurutan, misal loker tersebut diberi nama dengan loker A. Loker A berfungsi sebagai media penyimpanan, sehingga loker A di sini bisa dianggap sebagai variabel A yang berupa array. Setiap ruang penyimpanan pada loker memiliki nomor yang menyatakan posisinya dan menjadi pembeda dengan ruang penyimpanan lainnya, sehingga nomor bertindak sebagai indeks. Setiap ruang penyimpanan bertindak sebagai elemen array. Jadi, array adalah kumpulan data yang mempunyai nama variabel yang sama dan setiap elemen tersebut dibedakan berdasar nomor indeksnya. Pada variabel array, kita tidak hanya menentukan tipe datanya saja, tetapi juga jumlah elemen dari array tersebut atau dalam hal ini adalah batas atas indeksnya.

Agar lebih jelas mengenai array, Loker A dapat dapat digambarkan melalui skema berikut:



Gambar 2.2. Ilustrasi Array

Berdasar pada skema array di atas, array satu dimensi bisa dipandang sebagai tabel satu baris dengan banyak kolom. Array bisa dimanfaatkan saat kita membutuhkan untuk menyimpan banyak data dengan semua data tersebut memiliki tipe data yang sama. Daripada menggunakan banyak data tunggal akan menjadi lebih efisien dan lebih mudah dikelola jika menggunakan array.

Contoh :

Andaikata kita memerlukan untuk menyimpan 10 nama. Kita dapat membuat array dengan 10 elemen yang dapat digambarkan sebagai berikut:

Nama[1]	Nama[2]	Nama[3]	Nama[4]	Nama[10]
Annisa	Tio	Dhea	Dewi	Lisa

Ditinjau dari sisi memori, semakin banyak variabel berbeda yang kita gunakan berarti ada kemungkinan stack memori yang dialokasikan untuk program tersebut bisa membesar, sehingga ukuran program saat dijalankan bisa membutuhkan ruang memori RAM komputer yang cukup besar. Hal ini dapat mengakibatkan kurangnya kecepatan ketika program dijalankan. Mungkin untuk program-program kecil tidak begitu kelihatan, akan tetapi untuk program yang besar hal itu akan sangat kelihatan sekali.

Array merupakan struktur data yang statis, yaitu jumlah elemen yang ada harus ditentukan terlebih dahulu dan tak bisa diubah saat program berjalan. Untuk menyatakan array dalam bahasa Pascal harus terlebih dahulu mendefinisikan jumlah elemen array, dan juga mendefinisikan tipe data dari elemen array.

Dalam bahasa Pascal mengijinkan kita untuk menggunakan indeks dengan tipe data skalar, meliputi tipe data byte, char, integer, boolean, terbilang (numerated), dan tipe data subjangkauan dari tipe-tipe data tersebut.

Untuk membuat array bisa dengan dua cara, yaitu:

1. Deklarasi type diikuti dengan deklarasi variabel, dan
2. Langsung deklarasi variabel.

Cara 1. Deklarasi type diikuti dengan deklarasi variabel

```
Type  
  
  nama_array = array[iAwal..iAkhir] of tipe_data;  
  
var  
  
  variabel_array : nama_array;
```

Keterangan:

Bagian yang tercetak tebal adalah bagian yang harus dituliskan dalam menuliskan array.

Bagian yang miring adalah bagian yang dapat diganti sesuai dengan kebutuhan kita.

nama_array adalah definisi array pada type

iAwal adalah indeks awal dari array

iAkhir adalah indeks akhir dari array

tipe_data adalah tipe data yang digunakan pada array

variabel_array adalah deklarasi untuk variabel pada array

Contoh :

Untuk menampung nilai mata pelajaran dari 10 siswa bisa dibuat deklarasi sebagai berikut:

```
type
    nilai = array[1..10] of integer;

var
    nilai_siswa : nilai;
```

Cara 1. Langsung deklarasi variabel.

```
Var
    nama_array : array [iAwal..iAkhir] of
    tipe_data;
```

Keterangan:

Bagian yang tercetak tebal adalah bagian yang harus dituliskan dalam mendeklarasikan array.

Bagian yang miring adalah bagian yang dapat diganti sesuai dengan kebutuhan kita.

nama_array nama variabel array

iAwal adalah indeks awal dari array

iAkhir adalah indeks akhir dari array

tipe_data adalah tipe data yang digunakan pada array

Banyaknya elemen yang dimiliki array adalah (iAkhir - iAwal)

Contoh:

```
var nama : array[1..10] of string;
    nilai : array[1..10] of integer;
```

Keterangan:

- variabel nama adalah array atas string (kumpulan character) yang memiliki 10 elemen, digunakan untuk menyimpan data nama siswa
- variabel nilai adalah array atas integer (bilangan bulat) yang memiliki 10 elemen, digunakan untuk menyimpan data nilai siswa

Operasi pada Array

1. Penetapan Nilai Awal (inisialisasi)

Mengenai proses dalam pemberian nilai pada suatu variabel/objek data, beberapa compiler (termasuk Pascal) memberikan suatu nilai awal pada setiap variabel. Variabel bertipe numerik biasanya diberi nilai awal 0, sedangkan variabel yang bertipe string diberi nilai awal kosong(' '). Hal yang sama juga terjadi pada array, misalnya array nilai, maka setelah dideklarasikan, semua elemen-nya, yaitu `nilai[1]` sampai `nilai[10]` sudah mempunyai nilai yaitu kosong(' '). Untuk nilai [1] berarti array dengan nama nilai pada indeks ke-1 Sehingga array dapat diakses dengan menuliskan

```
nama_array[indeks_array]
```

Karena tidak semua compiler menyediakan fasilitas semacam itu, dan juga tidak semua nilai awal yang diberikan compiler sesuai dengan yang diharapkan, maka kita harus terbiasa untuk memberikan nilai awal pada setiap objek data. Langkah ini dikenal sebagai *proses inisialisasi*.

Berikut ini cara mendeklarasikan kelompok data kedalam array berdasarkan *proses inisialisasi*:

```
Var
  nama_array : array[iAwal..iAkhir] of tipe_data =
    (data ke-iAwal, data ke-iAwal+1, data ke-...,data
     ke iAkhir) ;
```

Keterangan:

Bagian yang tercetak tebal adalah bagian yang harus dituliskan dalam menuliskan inisialisasi sekelompok data ke dalam array.

Bagian yang miring adalah bagian yang dapat diganti sesuai dengan kebutuhan kita.

nama_array adalah nama variabel array

iAwal adalah indeks awal dari array

iAkhir adalah indeks akhir dari array

tipe_data adalah tipe data yang digunakan pada array

data ke-iAwal, data ke-iAwal+1, data ke akhir adalah data yang berada di dalam array sebanyak indeks array.

Contoh:

```
Var
  nilai : array[1..10] of integer = (8,7,8,9,7,6,10,
   9,7,8) ;
```

Keterangan:

Nilai adalah array dengan 10 elemen yang berisi bilangan bulat diantaranya 8,7,8,9,7,6,10,9,7,8.

2. Penetapan Nilai Awal (Assignment)

Bila array sudah dideklarasikan dan sudah diberi suatu nama, maka dapat dimanfaatkan sesuai fungsinya sebagai objek data. Operasi memasukkan nilai adalah operasi untuk memasukkan nilai data ke dalam elemen-elemen array. Biasanya hal ini dilakukan dengan operasi *assignment* dengan objek array terletak sebagai operan di sebelah kiri tanda ‘:=’.

Berikut ini cara memasukkan kelompok data kedalam array berdasarkan operasi *assignment*:

```
nama_array[indeks_array]:=data
```

Keterangan:

Bagian yang tercetak tebal adalah bagian yang harus dituliskan dalam memasukkan data ke dalam array.

Bagian yang miring adalah bagian yang dapat diganti sesuai dengan kebutuhan kita.

nama_array adalah nama variabel array

indeks_array adalah nomor indeks dari array

data adalah isi data yang dimasukkan ke dalam array

Contoh :

```
nilai[1]:=8
nilai[2]:=7
nilai[3]:=8
nilai[4]:=9
nilai[5]:=7
nilai[6]:=6
nilai[7]:=10
nilai[8]:=9
nilai[9]:=7
nilai[10]:=8
```

Keterangan:

Array yang mempunyai nama nilai dengan indeks 1 berisi data 8, nilai dengan indeks 2 berisi data 7, nilai dengan indeks 3 berisi data 8, nilai dengan indeks 4 berisi data 9, nilai dengan indeks 5 berisi data 7, nilai dengan indeks 6 berisi data 6, nilai dengan indeks 7 berisi data 10, nilai dengan indeks 8 berisi data 9, nilai dengan indeks 9 berisi data 7, nilai dengan indeks 10 berisi data 8.

Sebagai struktur data statis, jumlah elemen array sudah tertentu dan tidak berubah. Pada array tidak ada operasi memasukkan dan menghapus elemen. Operasi pemasukan nilai pada array bukanlah operasi untuk memasukkan elemen melainkan operasi penggantian nilai.

Sedangkan operasi mengambil nilai adalah operasi untuk mendapatkan / membaca nilai dari suatu array. Hal ini dilakukan ketika menggunakan array sebagai operan pada suatu operasi atau sebagai parameter sebuah fungsi / prosedur.

Untuk dapat menampilkan array kita dapat menggunakan perintah write seperti berikut.

```
write(nama_array[indeks_array]);
```

Keterangan:

nama_array adalah nama variabel dari array

indeks_array adalah nomor indeks dari array

contoh :

```
write(nilai[3]);
```

Keterangan:

Menampilkan data pada array dengan nama nilai pada ruang array 3.

Sedangkan untuk menampilkan seluruh array yang kita masukkan dengan menggunakan bantuan perulangan.

```
for i := iAwal to iAkhir do  
begin  
  writeln(nama_array[i]);  
end;
```

Keterangan:

Bagian yang tercetak tebal adalah bagian yang harus dituliskan dalam program, untuk **writeln** dapat juga ditulis dengan **write**.

Bagian yang miring adalah bagian yang dapat diganti sesuai dengan kebutuhan kita, untuk **i** harus dideklarasikan terlebih dahulu sebagai suatu variabel dengan tipe data integer.

nama_array adalah nama variabel array

iAwal adalah indeks awal dari array

iAkhir adalah indeks akhir dari array

Contoh :

```
for i := 1 to 10 do
begin
  writeln(nilai[i]);
end;
```

Keterangan:

Menampilkan data pada array dengan nama nilai pada ruang array **i**, untuk **i** dari 1 sampai dengan 10.

Contoh program lengkap array satu dimensi yang digunakan untuk mendata 5 orang siswa didalam suatu kelas.

Contoh :

```
program nama_siswa;
uses crt;
var
  namasiswa: array[1..5]of string;
begin
  clrscr;
  write('masukkan nama siswa pertama :');
  readln(namasiswa[1]);
  write('masukkan nama siswa kedua :');
  readln(namasiswa[2]);
  write('masukkan nama siswa ketiga :');
  readln(namasiswa[3]);
  write('masukkan nama siswa keempat :');
  readln(namasiswa[4]);
  write('masukkan nama siswa kelima :');
  readln(namasiswa[5]);
```

```
writeln('nama siswa yang telah dimasukkan
',namasiswa[1],
',namasiswa[2],',',namasiswa[3],',',namasiswa[4],',',na
masiswa[5]);
readln;
end.
```

Keterangan:

Program diatas bernama `nama_siswa`. Array didefinisikan dengan variabel `namasiswa` dan jumlah variabel array adalah 5 dengan type data yang akan diinputkan adalah string. Nama siswa pertama dimasukkan ke dalam array dengan indeks 1, nama siswa kedua dimasukkan ke dalam array dengan indeks 2, nama siswa ketiga dimasukkan ke dalam array dengan indeks 3, nama siswa keempat dimasukkan ke dalam array dengan indeks 4, nama siswa kelima dimasukkan ke dalam array dengan indeks 5. Kemudian seluruh nama siswa yang telah dimasukkan ke dalam array ditampilkan.

Untuk ruang array yang sedikit cara penulisan program diatas masih dapat dilakukan, tetapi jika ruang array berjumlah ratusan atau ribuan kita tidak mungkin menuliskan satu per satu deklarasi untuk memasukkan data ke dalam array. Maka dapat kita gunakan deklarasi perulangan `for ... do ...` seperti berikut.

```
for i:=1 to 5 do
Begin
    write('masukkan nama siswa ke-',i,' :');
    readln(namasiswa[i]);
end;
```

Keterangan:

`i` merupakan variabel untuk menentukan indeks pada array. Nama siswa ke-`i` dimasukkan ke dalam indeks ke-`i`

2.1.2.2. Menanya

Berdasarkan kegiatan mengamati, ada beberapa hal yang bisa diperhatikan dan dibahas lebih lanjut. Beberapa pertanyaan yang berkaitan dengan array satu dimensi adalah:

1. Bagaimana jika kita membuat array dengan deklarasi type tanpa spasi? Apa yang akan terjadi?

2. Apakah boleh jika kita melakukan proses inisialisasi pada saat deklarasi type?
3. Bagaimana jika kita menuliskan deklarasi variabel tanpa menggunakan spasi? Apa yang akan terjadi?
4. Apabila kita mempunyai array sebanyak n elemen, apakah boleh kita memasukkan data ke dalam array berdasarkan proses inisialisasi hanya sebagian saja (tidak sebanyak n elemen)?
5. Index awal apakah harus selalu 1? Bagaimana jika kita menuliskan indeks awal 0, bilangan negatif, atau bilangan yang lebih besar dari 1?
6. Apa yang akan terjadi apabila tanda titik pada indeks array dikurangi atau ditambah, misal [1..10] atau [1...10]?
7. Apa yang akan terjadi apabila indeks awal sama dengan indeks akhir, indeks awal lebih besar dari indeks akhir, dan indeks awal lebih kecil dari indeks akhir?
8. Apabila kita mempunyai array yang sudah diisi data secara inisialisasi, kemudian kita memasukkan data yang berbeda dengan inisialisasi secara assignment ke dalam array tersebut ? Apa yang akan terjadi?

Untuk menjawab pertanyaan-pertanyaan tersebut, siswa diharapkan bisa melakukan kegiatan mencoba berikut dengan bimbingan guru.

2.1.2.3. Mencoba

Percobaan 1

Untuk mengetahui jawaban dari pertanyaan pertama yaitu bagaimana jika kita membuat array dengan deklarasi type tanpa spasi, salin dan lengkapilah program berikut ke dalam Free Pascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Spasitype1;
uses crt;

{penulisan deklarasi type array tanpa menggunakan spasi}
type nilai=array[1..10]of integer;

var nilai_siswa : nilai;
begin

end.
```

Hasil Percobaan Spasitype1:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

```
program Spasitype2;
uses crt;

{penulisan pada deklarasi type array, spasi diletakkan
antara nilai dengan tanda '=' (sama dengan) }
type nilai =array[1..10] of integer;

var nilai_siswa : nilai;
begin

end.
```

Hasil Percobaan Program Spasitype2:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....
.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

```
program Spasitype3;
uses crt;

{penulisan deklarasi type array, spasi diletakkan antara '=' (sama dengan) dengan array}
type nilai= array[1..10]ofinteger;

var nilai_siswa : nilai;
begin

end.
```

Hasil Percobaan Program Spasitype3:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

```
program Spasitype4;
uses crt;

{penulisan deklarasi type array, spasi diletakkan antara
array dengan tanda '['}
type nilai=array [1..10] ofinteger;

var nilai_siswa : nilai;
begin

end.
```

Hasil Percobaan Program Spasitype4:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

```
program Spasitype5;
uses crt;

{penulisan deklarasi type array, spasi diletakkan antara ']' dengan of}
type nilai=array[1..10] ofinteger;
var nilai_siswa : nilai;
begin

end.
```

Hasil Percobaan Program Spasitype5:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

```
program Spasitype6;
uses crt;

{penulisan deklarasi type array, spasi diletakkan antara of dengan integer}
type nilai=array[1..10]of integer;

var nilai_siswa : nilai;
begin

end.
```

Hasil Percobaan Program Spasitype6:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

Percobaan 2

Untuk mengetahui jawaban dari pertanyaan kedua yaitu apakah boleh jika kita melakukan proses inisialisasi pada saat deklarasi type, salin dan lengkapilah program berikut ke dalam Free Pascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program InisialisasiType;
uses crt;

{inisialisasi dituliskan pada deklarasi type}
type
    nilai = array [1..10] of integer =(8,7,8,9,7,6,10,
    9,7,8);
var
    nilai_siswa : nilai;
begin
end.
```

Hasil Percobaan Program InisialisasiType:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

Percobaan 3

Untuk mengetahui jawaban dari pertanyaan ketiga yaitu bagaimana jika kita menuliskan deklarasi variabel tanpa menggunakan spasi, salin dan lengkapilah program berikut ke dalam Free Pascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Spasivar1;
uses crt;

{penulisan deklarasi variabel array tanpa menggunakan spasi}
var
    nilai:array[1..10]ofinteger;

begin
end.
```

Hasil Percobaan Program Spasivar1:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....

Letak kesalahan (baris, kolom)	Terjadi karena

```
program Spasivar2;
uses crt;

{penulisan pada deklarasi variabel array, spasi diletakkan
antara nilai dengan tanda ':' (titik dua) }

var
    nilai :array[1..10]ofinteger;

begin
end.
```

Hasil Percobaan Program Spasivar2:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

```

program Spasivar3;
uses crt;

{penulisan deklarasi variabel array, spasi diletakkan antara
'::' (titik dua) dengan array}

var
    nilai: array[1..10] of integer;

begin
end.

```

Hasil Percobaan Program Spasivar3:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

```

program Spasivar4;
uses crt;

{penulisan deklarasi variabel array, spasi diletakkan antara
array dengan tanda '['}

var
    nilai:array [1..10] of integer;
begin
end.

```

Hasil Percobaan Program 4:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

```
program Spasivar5;
uses crt;

{penulisan deklarasi variabel array, spasi diletakkan antara
' ] ' dengan of}

var
    nilai:array[1..10] ofinteger;

begin
end.
```

Hasil Percobaan Program Spasivar5:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

```
program Spasivar6;
uses crt;

{penulisan deklarasi variabel array, spasi diletakkan antara
of dengan integer}

var
    nilai:array[1..10] of integer;
begin
end.
```

Hasil Percobaan Program Spasivar6:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

Percobaan 4

Untuk mengetahui jawaban dari pertanyaan keempat yaitu apabila kita mempunyai array sebanyak n elemen, apakah boleh kita masukkan data ke dalam array berdasarkan proses inisialisasi hanya sebagian saja (tidak sebanyak n elemen), salin dan lengkapilah program berikut ke dalam Free Pascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Inisialisasisebagian;
uses crt;

var
{inisialisasi kurang dari indeks array}
    nilai : array [1..10] of integer = (8,7,8,9,7);

begin
end.
```

Hasil Percobaan Program Inisialisasisebagian:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

Percobaan 5

Untuk mengetahui jawaban dari pertanyaan kelima yaitu bagaimana jika kita menuliskan indeks awal 0, bilangan negatif, atau bilangan yang lebih besar dari 1, salin dan lengkapilah program berikut ke dalam Free Pascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Indeksnol;
uses crt;

var
{indeks array diawali dengan angka 0 (nol)}
    nilai : array [0..10] of integer;

begin
end.
```

Hasil Percobaan Program Indeksnol:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

```
program Indeksnegatif;
uses crt;
```

```

var
{indeks array diawali dengan bilangan negatif (-1)}
    nilai : array [-1..10] of integer;

begin
end.

```

Hasil Percobaan Program Indeksnegatif:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

Percobaan 6

Untuk mengetahui jawaban dari pertanyaan keenam yaitu apa yang akan terjadi apabila tanda titik pada indeks array dikurangi atau ditambah, misal [1.10] atau [1...10], salin dan lengkapilah program berikut ke dalam Free Pascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```

program Satutitik;
uses crt;

var
{penulisan indeks array dengan satu titik}
    nilai : array [1..10] of integer;

```

```
begin
end.
```

Hasil Percobaan Program Satutitik:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

Letak kesalahan (baris, kolom)	Terjadi karena

```
program Tigatitik;

uses crt;
var
  {penulisan indeks array dengan tiga titik}
  nilai : array [1...10] of integer;

begin
end.
```

Hasil Percobaan Program Tigatitik:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....

.....

Letak kesalahan (baris, kolom)	Terjadi karena

Percobaan 7

Untuk mengetahui jawaban dari pertanyaan ketujuh yaitu apa yang akan terjadi apabila indeks awal sama dengan indeks akhir, indeks awal lebih besar dari indeks akhir, dan indeks awal lebih kecil dari indeks akhir, salin dan lengkapilah program berikut ke dalam Free Pascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program AwalAkhirSama;
uses crt;

var nilai : array [10..10] of integer;

begin

end.
```

Hasil Percobaan Program AwalAkhirSama:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

```
program Awalbesar;

uses crt;

var nilai : array [10..1] of integer;

begin

end.
```

Hasil Percobaan Program Awalbesar:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

```
program Awalkecil;

uses crt;

var nilai : array [5..10] of integer;

begin

end.
```

Hasil Percobaan Program Awalkecil:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

Percobaan 8

Untuk mengetahui jawaban dari pertanyaan pertama yaitu apabila kita mempunyai array yang sudah diisi data secara inisialisasi, kemudian kita memasukkan data yang berbeda dengan inisialisasi secara assignment ke dalam array tersebut, salin dan lengkapilah program berikut ke dalam Free Pascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program CekArray;
uses crt;
var nilai : array[1..10] of integer = (8,7,8,9,7,6,10,9,
7,8);
begin
  clrscr;
  //menampilkan data yang tersimpan pada elemen pertama
  writeln(nilai[1]);
  //assignment
  nilai[1]:=10;
  //menampilkan data yang tersimpan pada elemen pertama
  writeln(nilai[1]);
  readln;
end.
```

Hasil Percobaan Program CekArray:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

2.1.2.4. Mengasosiasi/ menalar

Setelah siswa melakukan percobaan yang sesuai dengan pertanyaan yang ada, siswa diharapkan melakukan kegiatan menalar yang bisa dilakukan secara individu oleh masing-masing siswa atau bisa juga berkelompok. Beberapa hal yang bisa dijadikan arahan untuk menalar:

1. Perhatikan hasil kompilasi, apakah berhasil atau justru terjadi kesalahan.
2. Perhatikan urutan proses dari setiap program yang sudah dicoba.
3. Coba bandingkan beberapa program yang memiliki kemiripan, program mana yang berhasil dikompilasi dan program mana yang tidak berhasil dikompilasi.
4. Perhatikan letak/posisi kesalahan, coba pikirkan mengapa terjadi kesalahan pada posisi tersebut.
5. Cobalah membuat kesimpulan dari percobaan yang telah dilakukan.

2.1.3. Rangkuman

Dari percobaan-percobaan yang telah dilakukan, bisa ambil kesimpulan bahwa:

1. Array adalah sekumpulan data yang bertipe sama dan antara data satu dengan data yang lain dibedakan dengan indeksnya.
2. Dalam membuat array dengan deklarasi type dapat dituliskan tanpa spasi, tetapi setelah kata “of” dan sebelum type data harus dipisah dengan tanda spasi.
3. Dalam memasukkan data berdasarkan proses inisialisasi tidak dapat dilakukan dalam deklarasi type.
4. Dalam mendeklarasikan array pada deklarasi variabel dapat dituliskan tanpa spasi, tetapi untuk “of type_data” harus dipisah dengan spasi.
5. Jika terdapat array dengan n elemen, maka proses inisialisasi juga harus sebanyak n elemen.

6. Indeks awal pada array dapat berupa angka 1, bilangan negatif, maupun bilangan yang lebih besar dari 1
7. Jumlah titik pada indeks array harus berjumlah dua titik, tidak boleh kurang, juga tidak boleh lebih.
8. Indeks pada array dapat berupa angka dengan ketentuan indeks awal kurang dari indeks akhir, indeks awal sama dengan indeks akhir, dan juga indeks awal lebih besar dari indeks akhir.
9. Data dapat dimasukkan melalui proses assignment, meskipun data tersebut berbeda dengan data yang dimasukkan melalui proses inisialisasi, tetapi data yang akan ditampilkan pada saat program running tergantung dimana meletakkan perintah write adalah data yang dimasukkan melalui proses assignment.

2.1.4. Tugas

Perhatikan inisialisasi array berikut:

```
nilai : array [2..6] of integer = (78, 90, 65, 75, 83);
```

Tentukan nilai dalam elemen array berikut;

- a. nilai[3]
- b. nilai[5]
- c. nilai[4]

2.1.5. Uji Kompetensi

Kerjakan soal-soal berikut.

1. Pengertian dari Array adalah...
 - a. Sekumpulan data yang memiliki tipe sama
 - b. Sekumpulan data yang memiliki tipe data berbeda
 - c. Sekumpulan data yang memiliki tipe data heterogen
 - d. Sekumpulan data yang tidak memiliki tipe data
2. Jumlah elemen data dalam array bersifat...
 - a. Dinamis
 - b. Squensial
 - c. Statis
 - d. Heterogen
3. Perhatikan pendeklarasian array berikut :
unit_produksi : array[1..10] of string;
Maksud dari ‘unit_produksi’ dan ‘string’ adalah...
 - a. Nama array dan panjang array
 - b. Nama array dan indeks array

- c. Nama array dan tipe data
 - d. Panjang array dan tipe data
4. Pendeklarasian array satu dimensi yang benar adalah...
- a. nama array : array[1..10];
 - b. nama array : array[1..10] ofinteger;
 - c. nama_array: array[1..10] of integer
 - d. nama_array : array[1..10] of Integer;
5. Indeks array secara default dimulai dari ...
- a. -1
 - b. 0
 - c. 1
 - d. Tergantung indeks awal

Penggunaan Array sangat dibutuhkan dalam pemrograman. Pada Kegiatan Belajar sebelumnya kita telah mempelajari tentang seluk beluk array satu dimensi. Pada kegiatan belajar kali ini kita akan menggunakan array satu dimensi untuk menyelesaikan suatu permasalahan.

2.2. Kegiatan Belajar 2. Pemanfaatan Array Satu Dimensi

Alokasi Waktu : 2 x 45 menit

2.2.1. Tujuan Pembelajaran

Setelah mempelajari materi pada kegiatan belajar ini, siswa diharapkan dapat menyajikan kumpulan data berdimensi 1 dalam bentuk array.

2.2.2. Aktivitas belajar siswa

2.2.2.1. Mengamati

Sum atau Jumlah

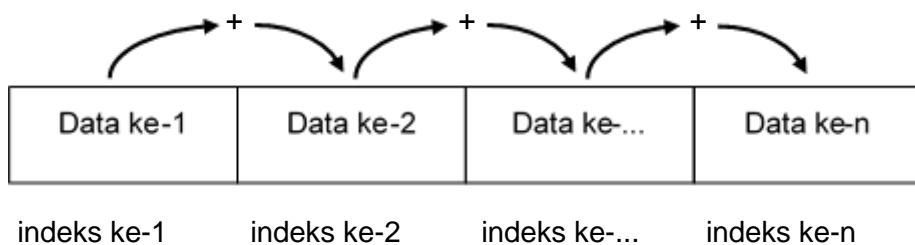
Perjumlahan adalah salah satu operasi aritmetika dasar. Perjumlahan merupakan penambahan dua buah bilangan atau lebih menjadi suatu bilangan yang merupakan jumlah atau sum.

Penjumlahan ditulis dengan menggunakan tanda tambah atau plus " + " di antara kedua bilangan. Hasil dari penjumlahan dinyatakan dengan tanda sama dengan " = "

Contoh :

$$5 + 6 + 4 + 2 + 3 = 20$$

Untuk penjumlahan data yang ada dalam indeks array dari indeks awal hingga indeks akhir dapat digambarkan sebagai berikut:



Gambar 2.3. Penjumlahan dari indeks awal hingga indeks akhir

Untuk menjumlahkan data dalam array dari indeks awal hingga indeks akhir dapat dituliskan sebagai berikut:

```
sum:=nama_array[iAwal] + nama_array[iAwal+1] +
nama_array[iAwal+...] + nama_array[iAkhir];
```

Keterangan:

sum adalah jumlah dari seluruh data pada array, sum harus dideklarasikan sebagai variabel dengan type data angka terlebih dahulu.

nama_array adalah nama variabel dari array

iAwal adalah indeks awal dari array

iAkhir adalah indeks akhir dari array

Array dijumlahkan mulai dari indeks awal sampai dengan indeks akhir

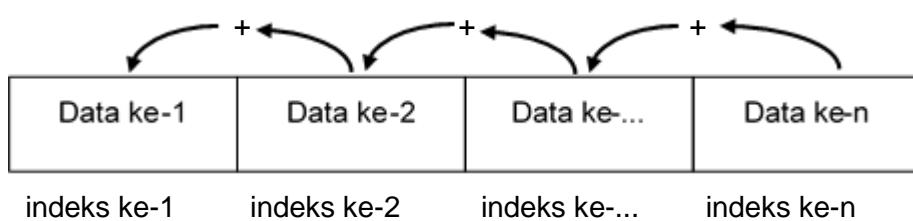
Contoh :

```
sum:= nilai[1] + nilai[2] + nilai[3] + nilai[4] +
nilai[5];
```

Keterangan:

Sum adalah hasil dari penjumlahan dalam array dengan nama nilai pada indeks 1 sampai dengan 5.

Untuk penjumlahan data yang ada dalam indeks array dari indeks akhir hingga indeks awal dapat digambarkan sebagai berikut:



Gambar 2.4. Penjumlahan dari indeks akhir hingga indeks awal

Untuk menjumlahkan data dalam array dari indeks akhir hingga indeks awal dapat dituliskan sebagai berikut:

```
sum:=nama_array[iAkhir] + nama_array[iAkhir-1] +
nama_array[iAkhir-...] + nama_array[iAwal];
```

Keterangan :

sum adalah jumlah dari seluruh data pada array, **sum** harus dideklarasikan sebagai variabel dengan type data angka terlebih dahulu.

nama_array adalah nama variabel dari array

iAwal adalah indeks awal dari array

iAkhir adalah indeks akhir dari array

Array dijumlahkan mulai dari indeks akhir sampai dengan indeks awal

Contoh :

```
sum:= nilai[5] + nilai[4] + nilai[3] + nilai[2] +
nilai[1];
```

Keterangan:

sum adalah hasil dari penjumlahan dalam array dengan nama nilai pada indeks 5 sampai dengan 1.

Mean atau rata-rata

Rata-rata atau Mean merupakan ukuran statistik yang paling sering digunakan. Rata-rata ada beberapa macam, yaitu rata-rata hitung (aritmatik), rata-rata geometrik, rata-rata harmonik dan lain-lain. Tetapi jika hanya disebut dengan kata "rata-rata" saja, maka rata-rata yang dimaksud adalah rata-rata hitung (aritmatik).

Penghitungan rata-rata dilakukan dengan menjumlahkan seluruh nilai dari suatu kelompok data, kemudian dibagi dengan jumlah data tersebut. Jadi jika suatu kelompok data dengan jumlah data n , maka bisa dihitung rata-rata dari data tersebut dengan rumus sebagai berikut:

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n} \quad \text{atau} \quad \bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

Keterangan:

\bar{x} = rata-rata hitung

x_i = nilai data ke- i

n = jumlah data

Untuk mencari mean atau rata-rata tidak lepas dari sum atau penjumlahan, begitu juga pada bahasa Pascal. Dalam menentukan mean dapat dituliskan seperti berikut:

.....

```
mean:=sum/banyak_indeks;
```

Keterangan :

mean adalah nilai rata-rata dari seluruh indeks array, **mean** harus dideklarasikan sebagai variabel dengan type data angka terlebih dahulu.

sum adalah hasil dari penjumlahan seluruh data dalam array

banyak_indeks adalah banyaknya indeks yang ada di dalam array

Contoh :

Begin

.....

```
mean:=sum/5;
```

Keterangan :

mean adalah hasil dari rata-rata data yang ada di dalam array dengan indeks 5

2.2.2.2. Menanya

Berdasarkan kegiatan mengamati, ada beberapa hal yang bisa diperhatikan dan dibahas lebih lanjut. Beberapa pertanyaan yang berkaitan dengan Pemanfaatan Array Satu Dimensi adalah:

1. Carilah sepuluh kata, kemudian tentukan apakah palindrom atau tidak dari kata-kata tersebut.
2. Jika seorang pengusaha memiliki 10 buah perusahaan, dimana perusahaan A memperoleh laba 5.000.000, perusahaan B memperoleh laba 1.000.000, perusahaan C memperoleh laba 4.000.000, perusahaan D memperoleh laba 6.000.000, perusahaan E memperoleh laba 4.000.000, perusahaan F memperoleh laba 5.000.000, perusahaan G memperoleh laba 10.000.000, perusahaan H memperoleh laba 7.000.000, perusahaan I memperoleh laba 3.000.000, perusahaan J memperoleh laba 8.000.000, berapakah laba total yang diperoleh pengusaha tersebut? Berapakah rata-rata laba pengusaha tersebut? Apakah ada perusahaan yang mempunyai laba sama dengan rata-rata, lebih dari rata-rata, dan lebih dari rata-rata?

Untuk menjawab pertanyaan-pertanyaan tersebut, siswa diharapkan bisa melakukan kegiatan mencoba berikut dengan bimbingan guru.

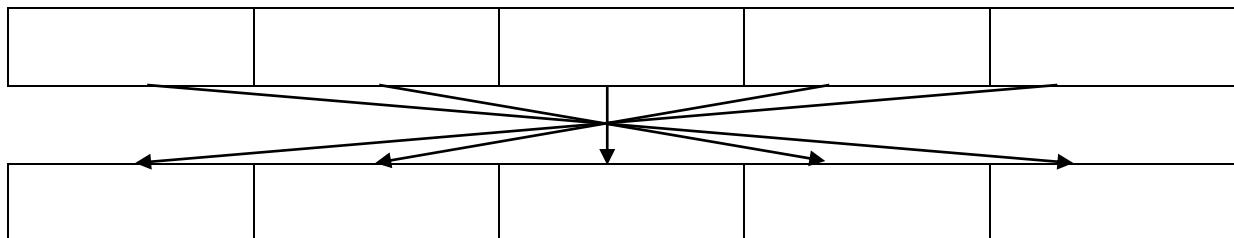
2.2.2.3. Mencoba

Percobaan 1

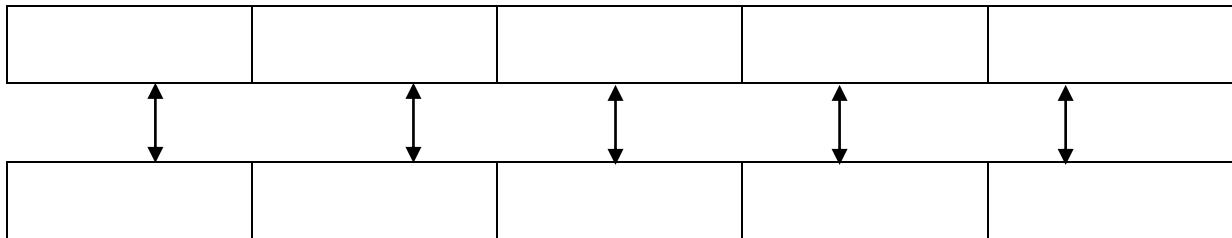
Untuk mengetahui jawaban dari pertanyaan pertama yaitu apakah palindrom atau tidak dari kata-kata yang kalian cari, salin dan lengkapilah program berikut ke dalam Free Pascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

Apa itu palindrom? Palindrom adalah kata atau kalimat yang dibaca dari depan atau dari belakang sama! Contoh kata di atas adalah **katak**. Katak kalau dibaca susunan setiap hurufnya dari depan atau dari belakang akan tetap terbaca kata!

Berikut adalah skema dalam menentukan palindrom dengan menggunakan array



Gambar 2.5 Membalikkan kata dalam array



Gambar 2.6 Memeriksa kesamaan dua array

Program 1. palindrom

```
program CekPalindrom;
uses crt;
var
    arra: array [1..10] of char;
    arrb: array [1..10] of char;
    i,j,n: integer;
    palindrom : boolean;
begin
    clrscr;
    write('masukkan jumlah huruf : ');
    readln(n);
    for i:=1 to n do
    begin
        write('masukkan huruf ke-',i,' : ');
        readln(arra[i]);
    end;
    //tampilkan kata yg dimasukkan
    write('kata yang dimasukkan : ');
    for i:=1 to n do
    begin
        write(arra[i]);
    end;

    //proses balik kata
    j:=1;
    for i:=n downto 1 do
    begin
        arrb[j]:=arra[i];
        j:=j+1;
    end;
```

```
writeln;
//tampilkan kata yang sudah dibalik
write('kata yang dibalik : ');
for j:=1 to n do
begin
    write(arrb[j]);
end;
writeln;
//periksa palindrom
palindrom:=TRUE;
for i:=1 to n do begin
    if arra[i]=arrb[i] then
        palindrom := palindrom AND true
    else
        palindrom := palindrom AND false;
end;
if palindrom=TRUE then
    writeln('Kata tersebut Palindrom')
else
    writeln('Kata tersebut BUKAN Palindrom');
readln;
end.
```

Hasil Percobaan Program CekPalindrom:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 2

Untuk mengetahui jawaban dari pertanyaan pertama yaitu berapakah laba total yang diperoleh pengusaha tersebut, berapakah rata-rata laba pengusaha tersebut dan apakah ada perusahaan yang mempunyai laba sama dengan rata-rata, lebih dari rata-rata, dan lebih dari rata-rata, salin dan lengkapilah program berikut ke dalam Free Pascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program LabaPerusahaan;
uses crt;
var
    p : array [1..10] of real;
    q : array [1..10] of string;
    i : integer;
    jum,rata : real;
begin
    clrscr;
    for i:=1 to 10 do
    begin
        write('masukkan nama perusahaan : ');
        readln(q[i]);
        write('laba perusahaan ',q[i],':');
        readln(p[i]);
    end;
    for i:=1 to 10 do
    begin
        jum:=jum+p[i];
    end;
    write('jumlah laba dari pengusaha: ',jum:0:2);
    writeln;
    rata:=jum/10;
    write('rata-rata laba dari pengusaha: ',rata:0:2);
    writeln;
    for i:=1 to 10 do
    begin
        if p[i] = rata then
            begin
```

```
        write('laba perusahaan ',q[i],' sama dengan
              rata-rata');
        readln;
    end
else if p[i] > rata then
begin
    write('laba perusahaan ',q[i],' lebih dari
          rata-rata');
    readln;
end
else if p[i] < rata then
begin
    write('laba perusahaan ',q[i],' kurang dari
          rata-rata');
    readln;
end;
readln;
end.
```

Hasil Percobaan Program LabaPerusahaan:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Pada program tersebut, untuk mencari jumlah dengan menjumlahkan p[1] sampai dengan p[10]. Apabila untuk mencari jumlah dirubah dengan menjumlahkan p[10] sampai dengan p[1] apakah hasilnya sama? Tentukan hasilnya dengan memodifikasi program tersebut.

2.2.2.4. Mengasosiasi

Setelah siswa melakukan percobaan yang sesuai dengan pertanyaan yang ada, siswa diharapkan melakukan kegiatan menalar yang bisa dilakukan secara individu oleh masing-masing siswa atau bisa juga berkelompok. Beberapa hal yang bisa dijadikan arahan untuk menalar:

1. Perhatikan hasil kompilasi, apakah berhasil atau justru terjadi kesalahan.
2. Perhatikan urutan proses dari setiap program yang sudah dicoba.
3. Bagaimana proses untuk menentukan suatu kata palindrom atau tidak?
4. Coba bandingkan program pada percobaan kedua, apakah terdapat perbedaan dengan jumlah array dari indeks akhir hingga indeks awal? Bagaimana dengan rata-ratanya?
5. Apakah untuk menentukan jumlah dan rata-rata bisa / harus dibuat dalam program terpisah? sebaiknya dibuat dalam satu program atau tidak?
6. Cobalah membuat kesimpulan dari percobaan yang telah dilakukan.

2.2.3. Rangkuman

Dari percobaan-percobaan yang telah dilakukan, bisa ambil kesimpulan bahwa:

1. Untuk membuat Palindrom bisa ditentukan melalui array dengan indeks pertama sama dengan indeks kelima dan array dengan indeks kedua sama dengan indeks keempat.
2. Untuk mendapatkan laba total, laba rata-rata, dan perusahaan mana yang sama dengan, kurang dari, atau lebih dari rata-rata bisa dengan cara melakukan perhitungan mulai dari perusahaan pertama sampai dengan perusahaan terakhir, maupun sebaliknya.
3. Urutan proses dalam program penentuan laba adalah dengan cara menghitung total laba, kemudian dicari rata-rata laba, dan yang terakhir adalah membandingkan setiap elemen array dengan rata-rata untuk menentukan perusahaan mana yang sama dengan, kurang dari, atau lebih dari rata-rata.
4. Pada program penentuan laba sebaiknya program untuk menentukan laba total dan rata-rata laba berada pada satu program, tidak dipisah-pisah untuk setiap kegunaan yang diinginkan.

2.2.4. Tugas

Diberikan suatu program sebagai berikut:

```
for i := 1 to n do
begin
    data[i]:=i;
end;
```

Tentukan nilai elemen array untuk

- a. Indeks ke-5
- b. Indeks ke-21
- c. Indeks ke-102

2.2.5. Uji Kompetensi

Kerjakan soal-soal berikut.

1. Buatlah program dengan ketentuan:

Input berupa nilai Matematika, Bahasa Indonesia, Bahasa Inggris, Agama

Output berupa total nilai

2. Modifikasi program pada soal no.1, tambahkan proses untuk mendapatkan kelulusan nilai batas tuntas adalah 70 dan proses untuk mendapatkan grade (A jika nilai ≥ 90 , B jika $90 < \text{nilai} \leq 80$, C jika $80 < \text{nilai} \leq 70$, dan D jika $\text{nilai} < 70$).

Sejauh ini struktur yang telah dibahas merupakan array yang bekerja dengan daftar linear yang diakses dengan satu indeks, yang disebut array satu dimensi. Array satu dimensi sebagai data dalam kolom tunggal, yang semua datanya bertipe sama. Kadang-kadang kita perlu membuat struktur yang lebih kompleks yang mempunyai dua dimensi yaitu berupa baris dan kolom.

Dalam Kegiatan Belajar 3 kita akan mempelajari Array Dua Dimensi

2.3. Kegiatan Belajar 3. Array Dua Dimensi

Alokasi Waktu : 2 x 45 menit

2.3.1. Tujuan Pembelajaran

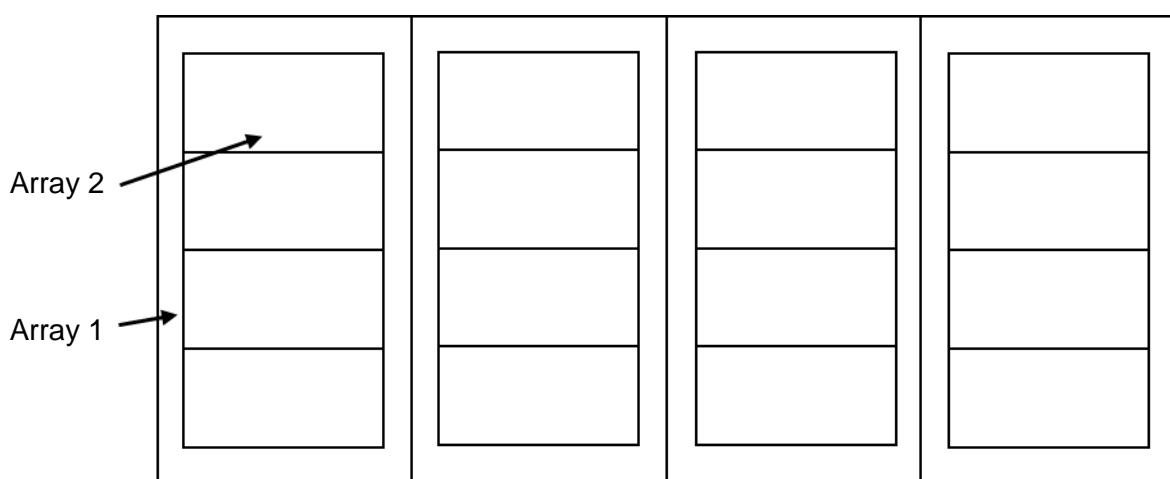
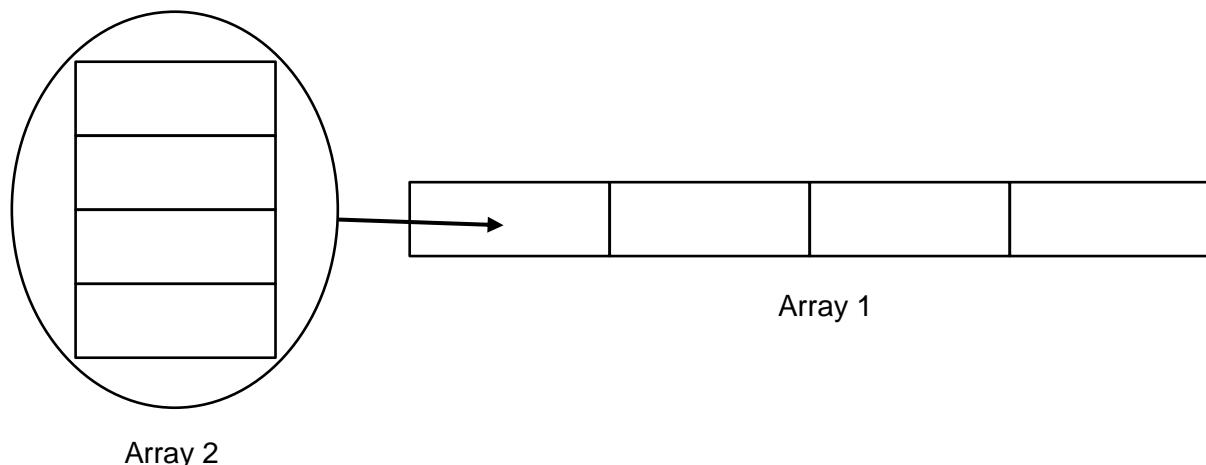
Setelah mempelajari materi pada kegiatan belajar ini, siswa diharapkan dapat Memahami penggunaan array 2 dimensi untuk penyimpanan data di memori.

2.3.2. Aktivitas belajar siswa

2.3.2.1. Mengamati/ observasi

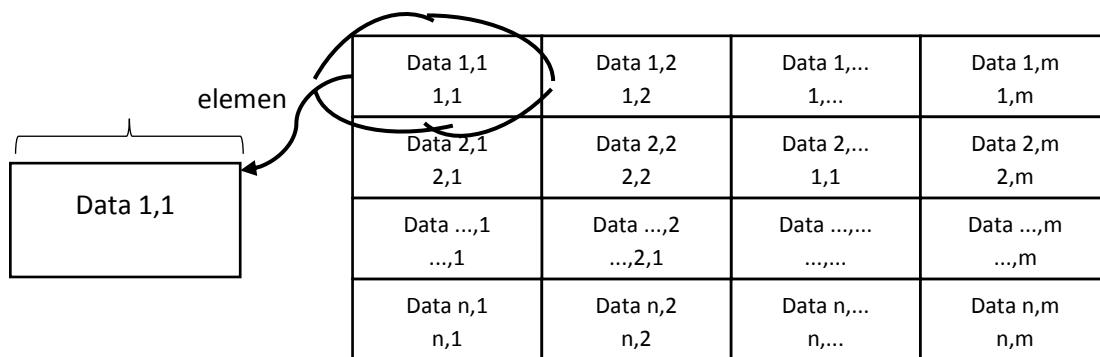
Array dua dimensi merupakan array satu dimensi yang berada pada array satu dimensi dimana variabel dan juga type datanya sama.

Perhatikan gambar berikut:



Gambar 2.7. Array dalam array

Array dua dimensi pada dasarnya sama dengan array satu dimensi, hanya saja array dua dimensi sering digambarkan sebagai sebuah matriks, merupakan perluasan dari array satu dimensi. Jika array satu dimensi hanya terdiri dari sebuah baris dan beberapa kolom elemen, maka array dua dimensi terdiri dari beberapa baris dan beberapa kolom elemen bertipe sama sehingga dapat digambarkan sebagai berikut :

**Gambar 2.8.** Array dua dimensi

Array dua dimensi merupakan array yang terdiri dari m buah baris (row) dan n buah kolom (column). Bentuk array semacam ini menggunakan 2 (dua) buah kelompok indeks yang masing-masing bisa direpresentasikan sebagai indeks baris dan kolom.

Contoh :

Andaikata kita akan menyimpan 3 tes untuk 30 siswa, kita dapat membuat tabel sebagai berikut:

Tabel 1.1 Nilai Siswa

	Tes 1	Tes 2	Tes 3
Siswa 1			
Siswa 2			
Siswa 3			
.....			
Siswa 30			

indeks pertama mewakili jumlah siswa dan indeks kedua mewakili jumlah tes. Jika nama arraynya tes_siswa, maka tes_siswa [1,2] berarti mengandung nilai untuk siswa pertama tes yang kedua. tes_siswa [3,1] mengandung tes pertama untuk siswa ke-3.

Pendeklarasian array dua dimensi hampir sama dengan pendeklarasian array satu dimensi, kecuali bahwa array dua dimensi terdapat dua jumlah elemen yang terdapat dikurung kurung siku dan keduanya boleh tidak sama. Elemen array dua dimensi diakses dengan menuliskan kedua indeks elemennya dalam kurung.

Bentuk umum deklarasi variabel dalam bentuk array dua dimensi

Var

```
nama_array : array[iAwal1..iAkhir1] of array
[iAwal2..iAkhir2] of tipe_data;
```

Keterangan:

Bagian yang tercetak tebal adalah bagian yang harus dituliskan dalam mendeklarasikan array.

Bagian yang miring adalah bagian yang dapat diganti sesuai dengan kebutuhan kita.

nama_array adalah definisi array pada type

iAwal1 adalah indeks awal dari array dimensi pertama

iAkhir1 adalah indeks akhir dari array dimensi pertama

iAwal2 adalah indeks awal dari array dimensi kedua

iAkhir2 adalah indeks akhir dari array dimensi kedua

tipe_data adalah tipe data yang digunakan pada array

Contoh :

Var

```
tes_siswa : array[1..30] of array [1..3] of real;
```

Keterangan:

tes_siswa merupakan array dua dimensi dengan indeks pertama berjumlah 30 dan indeks kedua berjumlah jumlah 3.

Cara mengakses elemen array dua dimensi, sama dengan cara mengakses elemen pada array satu dimensi yaitu cara pertama bisa menggunakan dua indeks yang dipisahkan dengan koma dan di dalam tanda kurung siku (brackets), misal tes_siswa[3,2] atau tes_siswa[indeks, tes]. Cara kedua adalah mendaftar indeks secara terpisah dan masing-masing di dalam tanda kurung siku, misal tes_siswa[3] [2] equivalen dengan tes_siswa[3,2], tes_siswa[indeks][tes] equivalen dengan tes_siswa[indek,tes].

Array dua dimensi biasa digunakan untuk menyelesaikan suatu permasalahan yang berdasarkan tabel atau koordinat cartesius dua dimensi.

Operasi Memasukkan dan Mengambil Nilai

Seperti array satu dimensi, array dua dimensi juga diakses melalui indeksnya. Contoh: nilai[1,2], menunjuk pada posisi nilai array pada baris 1, kolom 2.

Seperi pada array satu dimensi berikut ini cara mendeklarasikan kelompok data kedalam array dua dimensi berdasarkan operasi *assignment*:

```
nama_array[indeks_array1,indeks_array2]:=data
```

Keterangan:

Bagian yang tercetak tebal adalah bagian yang harus dituliskan dalam memasukkan data ke dalam array.

Bagian yang miring adalah bagian yang dapat diganti sesuai dengan kebutuhan kita.

nama_array adalah nama variabel array

indeks_array1 adalah nomor indeks dari array dimensi pertama yang menyatakan baris

indeks_array2 adalah nomor indeks dari array dimensi kedua yang menyatakan kolom

data adalah data yang dimasukkan ke dalam array

Contoh :

```
tes_siswa[3,2]:=95
```

Keterangan:

tes_siswa array yang berisi nilai dari siswa ketiga untuk tes kedua yaitu 95.

Untuk dapat menampilkan array kita dapat menggunakan perintah write seperti berikut.

```
write(nama_array[indeks_array1,indeks_array2]);
```

Keterangan:

nama_array adalah nama variabel dari array

indeks_array1 adalah nomor indeks dari array dimensi pertama

indeks_array2 adalah nomor indeks dari array dimensi kedua

Contoh :

```
write(tes_siswa[3,2]);
```

Keterangan:

Menampilkan data pada array dua dimensi dengan nama tes_siswa untuk baris ketiga dan kolom kedua.

2.3.2.2. Menanya

Berdasarkan kegiatan mengamati, ada beberapa hal yang bisa diperhatikan dan dibahas lebih lanjut. Beberapa pertanyaan yang berkaitan dengan array dua dimensi adalah:

1. Bagaimana cara deklarasi type array dua dimensi dengan singkat?
2. Bagaimana cara deklarasi variabel array dua dimensi dengan singkat?
3. Bagaimana cara inisialisasi array dua dimensi?

Untuk menjawab pertanyaan-pertanyaan tersebut, siswa diharapkan bisa melakukan kegiatan mencoba berikut dengan bimbingan guru.

2.3.2.3. Mencoba

Percobaan 1

Untuk mengetahui jawaban dari pertanyaan pertama yaitu bagaimana cara deklarasi type array dua dimensi dengan singkat, salin dan lengkapilah program berikut ke dalam Free Pascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Gajitype;
uses crt;

type gaji_pegawai = array[1..10,1..7] of real;
var gaji : gaji_pegawai;
begin
end.
```

Hasil Percobaan Program Gajitype:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

Percobaan 2

Untuk mengetahui jawaban dari pertanyaan kedua yaitu bagaimana cara deklarasi variabel array dua dimensi dengan singkat, salin dan lengkapilah program berikut ke dalam Free Pascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Gajivar;
uses crt;
var
  gaji_pegawai : array[1..10,1..7] of real;
begin
end.
```

Hasil Percobaan Program Gajivar:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

Percobaan 3

Untuk mengetahui jawaban dari pertanyaan ketiga yaitu bagaimana cara inisialisasi array dua dimensi dengan singkat, salin dan lengkapilah program berikut ke dalam Free Pascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program InitArray2d;
uses crt;

var tes_siswa : array[1..3,1..4] of real =
((60,75,70,75), (70,80,75,70), (80,80,85,90));
begin
end.
```

Hasil Percobaan Program InitArray2d:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....

.....

Letak kesalahan (baris, kolom)	Terjadi karena

2.3.2.4. Mengasosiasi

Setelah siswa melakukan percobaan yang sesuai dengan pertanyaan yang ada, siswa diharapkan melakukan kegiatan menalar yang bisa dilakukan secara individu oleh masing-masing siswa atau bisa juga berkelompok. Beberapa hal yang bisa dijadikan arahan untuk menalar:

1. Perhatikan hasil kompilasi, apakah berhasil atau justru terjadi kesalahan.
2. Perhatikan urutan proses dari setiap program yang sudah dicoba.
3. Perhatikan deklarasi type array dua dimensi pada percobaan pertama.
4. Perhatikan deklarasi variabel array dua dimensi pada percobaan kedua.
5. Perhatikan inisialisasi array dua dimensi pada percobaan ketiga.
6. Cobalah membuat kesimpulan dari percobaan yang telah dilakukan.

2.3.3. Rangkuman

Dari percobaan-percobaan yang telah dilakukan, bisa ambil kesimpulan bahwa:

1. Bentuk deklarasi type array dua dimensi

```
Type
    nama_array = array [iAwal1..iAkhir1,iAwal2..iAkhir2]
        of type_data;

var
    variabel_array : nama_array;
```

Keterangan:

Bagian yang tercetak tebal adalah bagian yang harus dituliskan dalam menuliskan deklarasi type array dua dimensi.

Bagian yang miring adalah bagian yang dapat diganti sesuai dengan kebutuhan kita.

nama_array adalah definisi array dua dimensi pada type
iAwal1 adalah indeks awal dari array dimensi pertama
iAkhir1 adalah indeks akhir dari array dimensi pertama
iAwal2 adalah indeks awal dari array dimensi kedua
iAkhir2 adalah indeks akhir dari array dimensi kedua
tipe_data adalah tipe data yang digunakan pada array dua dimensi
variabel_array adalah deklarasi untuk variabel pada array dua dimensi

2. Bentuk deklarasi variabel array dua dimensi

Var

```
nama_array : array [iAwal1..iAkhir1,iAwal2..iAkhir2]  
of tipe_data;
```

Keterangan:

Bagian yang tercetak tebal adalah bagian yang harus dituliskan dalam menuliskan deklarasi var array dua dimensi.

Bagian yang miring adalah bagian yang dapat diganti sesuai dengan kebutuhan kita.

nama_array adalah deklarasi variabel array dua dimensi

iAwal1 adalah indeks awal dari array dimensi pertama

iAkhir1 adalah indeks akhir dari array dimensi pertama

iAwal2 adalah indeks awal dari array dimensi kedua

iAkhir2 adalah indeks akhir dari array dimensi kedua

tipe_data adalah tipe data yang digunakan pada array dua dimensi

3. Inisialisasi array dua dimensi

Var

```
nama_array : array[iAwal1..iAkhir1,iAwal2..iAkhir2]  
of tipe_data = ((data {iAwal1,iAwal2}, data  
{iAwal1,iAwal2+1}, data {iAwal1,iAwal2...},  
data {iAwal1,iAkhir2}), (data {iAwal1+1,iAwal2},  
data {iAwal1+1,iAwal2+1}, data {iAwal1+1,iAwal2...},  
data {iAwal1+1,iAkhir2}), (data {iAwal1...,iAwal2},  
data {iAwal1...,iAwal2+1},  
data {iAwal1...,iAwal2...},  
data {iAwal1...,iAkhir2}), (data {iAkhir1,iAwal2},  
data {iAkhir1,iAwal2+1}, data {iAkhir1,iAwal2...},  
data {iAkhir1,iAkhir2});
```

Keterangan:

Bagian yang tercetak tebal adalah bagian yang harus dituliskan dalam menuliskan inisialisasi sekelompok data ke dalam array dua dimensi.

Bagian yang miring adalah bagian yang dapat diganti sesuai dengan kebutuhan kita.

nama_array adalah nama variabel array dua dimensi

iAwal1 adalah indeks awal dari array dimensi pertama

iAkhir1 adalah indeks akhir dari array dimensi pertama

iAwal2 adalah indeks awal dari array dimensi kedua

iAkhir2 adalah indeks akhir dari array dimensi kedua

tipe_data adalah tipe data yang digunakan pada array dua dimensi

data adalah data yang berada di dalam array dua dimensi sebanyak indeks array.

2.3.4. Tugas

Perhatikan inisialisasi array berikut:

nilai : array [2..4,5..7] of integer = ((90,65,70), (85,70,75), (80,75,70));

Tentukan nilai dalam elemen array berikut;

- nilai[3,5]
- nilai[2,6]
- nilai[4,5]

2.3.5. Uji Kompetensi/ Ulangan

Kerjakan soal-soal berikut.

- Jika diketahui data seperti berikut:

Nama	Gaji
Nining Kania	1000000
Timbul Harahap	2000000
Setyoningsih	1500000
Rendy Alvaro	2000000

Buatlah program untuk menampilkan data diatas.

Keterangan : data dimasukkan melalui proses assignment.

- Jika diketahui data seperti berikut:

Menu makanan	Harga
Sate	10000
Tongseng	8000
Rames	5000
Bakso	6000

Buatlah program untuk menampilkan data diatas.

Keterangan : data dimasukkan melalui proses inisialisasi.

2.4. Kegiatan Belajar 4. Pemanfaatan Array Dua Dimensi

Alokasi Waktu : 2 x 45 menit

2.4.1. Tujuan Pembelajaran

Setelah mempelajari materi pada kegiatan belajar ini, siswa diharapkan dapat menyajikan kumpulan data berdimensi 2 dalam bentuk array.

2.4.2. Aktivitas belajar siswa

2.4.2.1. Mengamati/ observasi

Matriks

Di dalam buku kita tidak hanya melihat kata-kata atau kalimat saja, tetapi kita juga sering melihat suatu tabel yang berisi angka-angka. Dari tabel tersebut angka-angkanya dapat disajikan dalam bentuk matriks.

Matriks adalah kumpulan bilangan, simbol, atau ekspresi, berbentuk persegi panjang yang disusun menurut *baris* dan *kolom*. Bilangan-bilangan yang terdapat di suatu matriks disebut dengan *elemen* atau *anggota* matriks. Contoh matriks dengan 2 baris dan 3 kolom yaitu:

Contoh :

$$A = \begin{bmatrix} 2 & 3 & 5 \\ 1 & 4 & -7 \end{bmatrix}$$

Dalam matriks ada yang namanya ordo. Ordo matriks berarti ukuran banyaknya kolom dan baris suatu matriks. Misal matriks yang berordo 2x3 merupakan matriks dengan 2 baris dan 3 kolom. Secara umum matriks dituliskan seperti berikut

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{1...} & a_{1m} \\ a_{21} & a_{22} & a_{2...} & a_{2m} \\ a_{...1} & a_{...2} & a_{....} & a_{...m} \\ a_{n1} & a_{n2} & a_{n...} & a_{nm} \end{bmatrix}$$

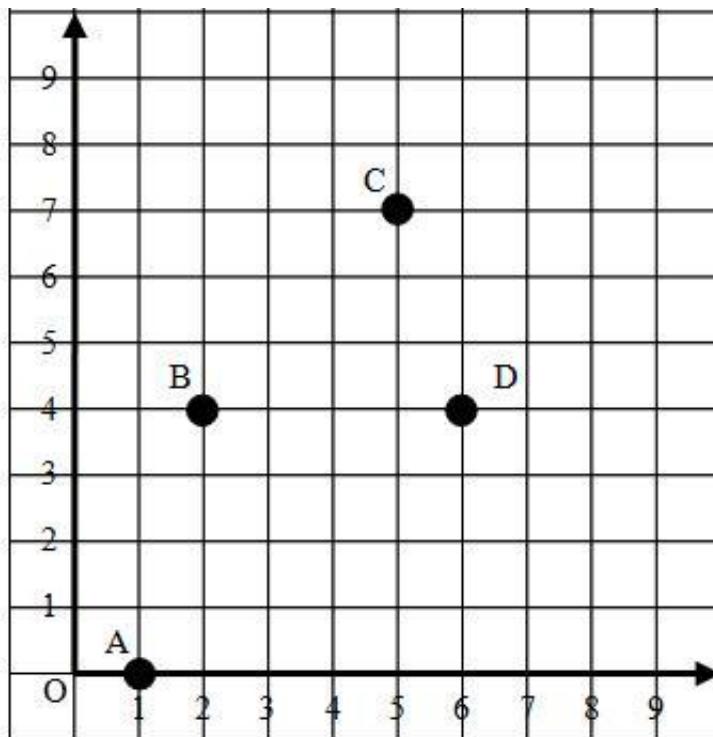
Matriks transpose adalah matriks yang mengalami pertukaran elemen dari baris menjadi kolom dan sebaliknya.

Contoh :

$$A = \begin{bmatrix} 2 & 3 & 5 \\ 1 & 4 & -7 \end{bmatrix} \quad A^t = \begin{bmatrix} 2 & 1 \\ 3 & 4 \\ 5 & -7 \end{bmatrix}$$

Kedudukan titik terhadap bidang XoY

Perhatikan gambar berikut:



Gambar 2.9. Titik pada Bidang

Bidang datar diatas disebut bidang koordinat Cartesius atau bidang XoY yang dibentuk oleh garis tegak Y (sumbu Y) dan garis mendatar X (sumbu X). Setiap sumbu juga mempunyai besaran panjang unit. Titik perpotongan antara garis Y dan garis X disebut pusat Koordinat (titik O). Bidang XoY biasa digunakan untuk menentukan letak sebuah titik yang dinyatakan dalam pasangan bilangan. Perhatikan titik A, B, C, dan D pada bidang tersebut. Untuk menentukan letaknya, mulailah dari titik O. Kemudian, bergerak mendatar ke arah kanan (sumbu X), lalu bergerak ke atas (sumbu Y).

Letak titik pada bidang XoY ditulis dalam bentuk pasangan bilangan (x,y) dimana x disebut abisis dan y disebut ordinat. Format yang dipakai selalu (x,y) dengan urutan yang tidak dapat dibalik. Pada bidang koordinat tersebut, titik A terletak pada koordinat $(1,0)$, ditulis $A(1,0)$, titik B terletak pada koordinat $(2,4)$, ditulis $B(2,4)$, titik C terletak pada koordinat $(5,7)$, ditulis dengan $C(5,7)$, dan titik D terletak pada koordinat $(6,4)$ ditulis $D(6,4)$.

Untuk mencari jarak antara dua titik dapat digunakan rumus

$$\text{Jarak} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Keterangan:

Jarak = jarak dua titik

x_1 = nilai x pada titik pertama

- x_2 = nilai x pada titik kedua
 y_1 = nilai y pada titik pertama
 y_2 = nilai y pada titik kedua

Untuk mencari koordinat titik tengah antara dua titik dapat digunakan rumus

$$x_t = (x_1 + x_2)/2 \quad y_t = (y_1 + y_2)/2$$

Keterangan:

- x_t = nilai x pada titik tengah
 y_t = nilai y pada titik tengah
 x_1 = nilai x pada titik pertama
 x_2 = nilai x pada titik kedua
 y_1 = nilai y pada titik pertama
 y_2 = nilai y pada titik kedua

2.4.2.2. Menanya

Berdasarkan kegiatan mengamati, ada beberapa hal yang bisa diperhatikan dan dibahas lebih lanjut. Beberapa pertanyaan yang berkaitan dengan pemanfaatan array dua dimensi adalah:

1. Jika diketahui matriks

$$A = \begin{bmatrix} 4 & -3 \\ -2 & 3 \\ 1 & 4 \end{bmatrix}$$

Tentukan transpose dari matriks A!

2. Diketahui kota P terletak pada P(2,7) dan kota Q terletak pada Q(10,3) akan dibuat jalan tol yang menghubungkan antara kota P dan kota Q, jika biaya per satuan jarak jalan tol adalah Rp 100.000.000,00 maka berapakah dana minimal yang harus disiapkan?
3. Diketahui pabrik A terletak pada A(5,1) dan pabrik B terletak pada B(1,8). Jika hendak dibangun gudang induk untuk menampung barang hasil produksi pabrik A dan pabrik B dan diinginkan lokasi tersebut sama-sama dekat dari kedua lokasi pabrik, maka tentukan lokasi gudang tersebut!

Untuk menjawab pertanyaan-pertanyaan tersebut, siswa diharapkan bisa melakukan kegiatan mencoba berikut dengan bimbingan guru.

2.4.2.3. Mencoba

Percobaan 1

Untuk mengetahui jawaban dari pertanyaan pertama yaitu menentukan tranpose suatu matriks ,salin dan lengkapilah program berikut ke dalam Free Pascal, kemudian lakukan kompilasi

(Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Transpose;
var
    a : array[1..10,1..10] of integer;
    i, j, bar, kol : integer;
begin
    write('masukkan jumlah baris : ');
    readln(bar);
    write('masukkan jumlah kolom : ');
    readln(kol);
    for i:=1 to bar do
begin
    for j:=1 to kol do
begin
        write('data ke-(',i,',',',',j,',')=');
        readln(a[i,j]);
    end;
end;
    writeln('Matriks yang dimasukkan : ');
    for i:=1 to bar do
begin
    for j:=1 to kol do
        write (a[i,j],' ');
    writeln;
end;
    writeln('Matriks tranposenya adalah : ');
    for i:=1 to kol do
begin
    for j:=1 to bar do
        write (a[j,i],' ');
    writeln;
end;
    readln;
end.
```

Hasil Percobaan Program Tranpose:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 2

Untuk mengetahui jawaban dari pertanyaan kedua yaitu menentukan jarak dua kota dalam titik pada bidang dua dimensi dan menentukan biaya per satuan jarak, salin dan lengkapilah program berikut ke dalam Free Pascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program TolArray;
var
    a: array[1..2,1..2] of real;
    b,harga,jar:real;
begin
    write('masukkan nilai x (absis) kota pertama : ');
    readln(a[1,1]);
    write('masukkan nilai y (ordinat) kota pertama : ');
    readln(a[1,2]);
    write('masukkan nilai x (absis) kota kedua : ');
    readln(a[2,1]);
    write('masukkan nilai y (ordinat) kota kedua : ');
    readln(a[2,2]);
    jar:=sqrt((sqr(a[1,1]-a[2,1]))+(sqr(a[1,2]-a[2,2])));
    writeln('jarak kedua kota adalah : ',jar:0:2);
    write('biaya pembuatan jalan per satuan jarak : '');
```

```

readln(b);
harga:=jar*b;
write('biaya minimal yang harus disiapkan :',
      ',harga:0:2);
readln;
end.

```

Hasil Percobaan Program TolArray:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 3

Untuk mengetahui jawaban dari pertanyaan ketiga yaitu menentukan lokasi gudang yang terdekat dari dua buah pabrik dalam titik pada bidang dua dimensi, salin dan lengkapilah program berikut ke dalam Free Pascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

Program 1. tengah

```

program LokasiGudang;
uses crt;
var
  a: array[1..2,1..2] of real;
  Lokx,Loky:real;
begin
  clrscr;
  write('masukkan nilai x (absis) pabrik pertama : ');
  readln(a[1,1]);

```

```
write('masukkan nilai y (ordinat) pabrik pertama : ');
readln(a[1,2]);
write('masukkan nilai x (absis) pabrik kedua : ');
readln(a[2,1]);
write('masukkan nilai y (ordinat) pabrik kedua : ');
readln(a[2,2]);
Lokx:=((a[1,1]+a[2,1])/2);
Loky:=((a[1,2]+a[2,2])/2);
writeln('gudang yang dibangun sebaiknya pada titik:
(',Lokx:0:2,',',Loky:0:2,')');
readln;
end.
```

Hasil Percobaan Program LokasiGudang:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

2.4.2.4. Mengasosiasi

Setelah siswa melakukan percobaan yang sesuai dengan pertanyaan yang ada, siswa diharapkan melakukan kegiatan menalar yang bisa dilakukan secara individu oleh masing-masing siswa atau bisa juga berkelompok. Beberapa hal yang bisa dijadikan arahan untuk menalar:

1. Perhatikan hasil kompilasi, apakah berhasil atau justru terjadi kesalahan.
2. Perhatikan urutan proses dari setiap program yang sudah dicoba.
3. Perhatikan transpose matriks pada percobaan pertama.
4. Perhatikan biaya minimal untuk membuat jalan tol untuk menghubungkan dua perusahaan pada percobaan kedua.

5. Perhatikan lokasi gudang yang terdekat dengan kedua pabrik tersebut pada percobaan ketiga.
6. Cobalah membuat kesimpulan dari percobaan yang telah dilakukan.

2.4.3. Rangkuman

Dari percobaan-percobaan yang telah dilakukan, bisa ambil kesimpulan bahwa:

1. Kita dapat menentukan transpose suatu matriks dengan cara menukar elemen baris menjadi kolom dan juga menukar elemen kolom menjadi baris.
2. Array dua dimensi dapat digunakan untuk menentukan biaya minimal untuk membuat jalan tol untuk menghubungkan dua perusahaan.

Array dua dimensi juga dapat digunakan untuk menentukan lokasi tempat pembangunan gudang yang terdekat dengan dua buah pabrik, dimana gudang tersebut digunakan untuk menyimpan hasil produksi dari dua buah pabrik tersebut.

2.4.4. Tugas

Diberikan program matriks,

```

begin
    clrscr;
    {matriks M}
    m[1,1]:=85;   m[1,2]:=70;
    m[2,1]:=90;   m[2,2]:=75;
    for i:=1 to 2 do {baris matriks M}
    begin
        for j:=1 to 2 do {kolom matriks M}
        begin
            write(m[i,j], '');
        end;
        writeln;
    end;
    readln;
end.

```

Tentukan nilai elemen array pada matriks M untuk:

- a. i ke-1 dan j ke 2
- b. i ke-2 dan j ke 2
- c. i ke-2 dan j ke 1

2.4.5. Uji Kompetensi

Kerjakan soal-soal berikut.

1. Jika diketahui dua buah matriks, buatlah program untuk menjumlahkan kedua matriks tersebut!
2. Untuk matriks seperti pada no.1, buatlah program untuk mengalikan matriks tersebut!

Array dua dimensi juga dapat disebut array multidimensi. Array Multi dimensi juga dapat memiliki ukuran yang lebih besar. Sebenarnya array Multi dimensi ini tidak terlalu sering digunakan, tetapi jika suatu saat kalau dimensi yang dibutuhkan banyak, maka array ini sangat memegang peranan yang penting.

2.5. Kegiatan Belajar 5. Array Multidimensi

Alokasi Waktu : 2 x 45 menit

2.5.1. Tujuan Pembelajaran

Setelah mempelajari materi pada kegiatan belajar ini, siswa diharapkan dapat memahami penggunaan array multidimensi untuk penyimpanan data di memori.

2.5.2. Aktivitas belajar siswa

2.5.2.1. Mengamati/ observasi

Dalam bahasa Pascal kita dapat membuat array hingga beberapa dimensi, misalnya array dimensi dua yang telah kita bahas sebelumnya, array dimensi tiga, array dimensi empat, dan seterusnya. Pada dasarnya array multidimensi adalah sebuah daftar yang terdiri array-array satu dimensi.

Array multidimensi adalah array yang berisi beberapa array lain. Array multidimensi bisa memudahkan pembuatan suatu program karena beberapa array dapat dibuat sekaligus dan beberapa perintah operasi bisa diringkas. Bentuk umum pendeklarasian variabel array multidimensi:

```
var
  nama_array : array[iAwal1..iAkhir1] of array
  [iAwal2..iAkhir2] of array [iAwal... ..iAkhir...] of
  array [iAwaln..iAkhirn] of tipe_data;
```

Keterangan:

Bagian yang tercetak tebal adalah bagian yang harus dituliskan dalam mendeklarasikan array.

Bagian yang miring adalah bagian yang dapat diganti sesuai dengan kebutuhan kita.

nama_array adalah definisi array pada type

iAwal adalah indeks awal dari array

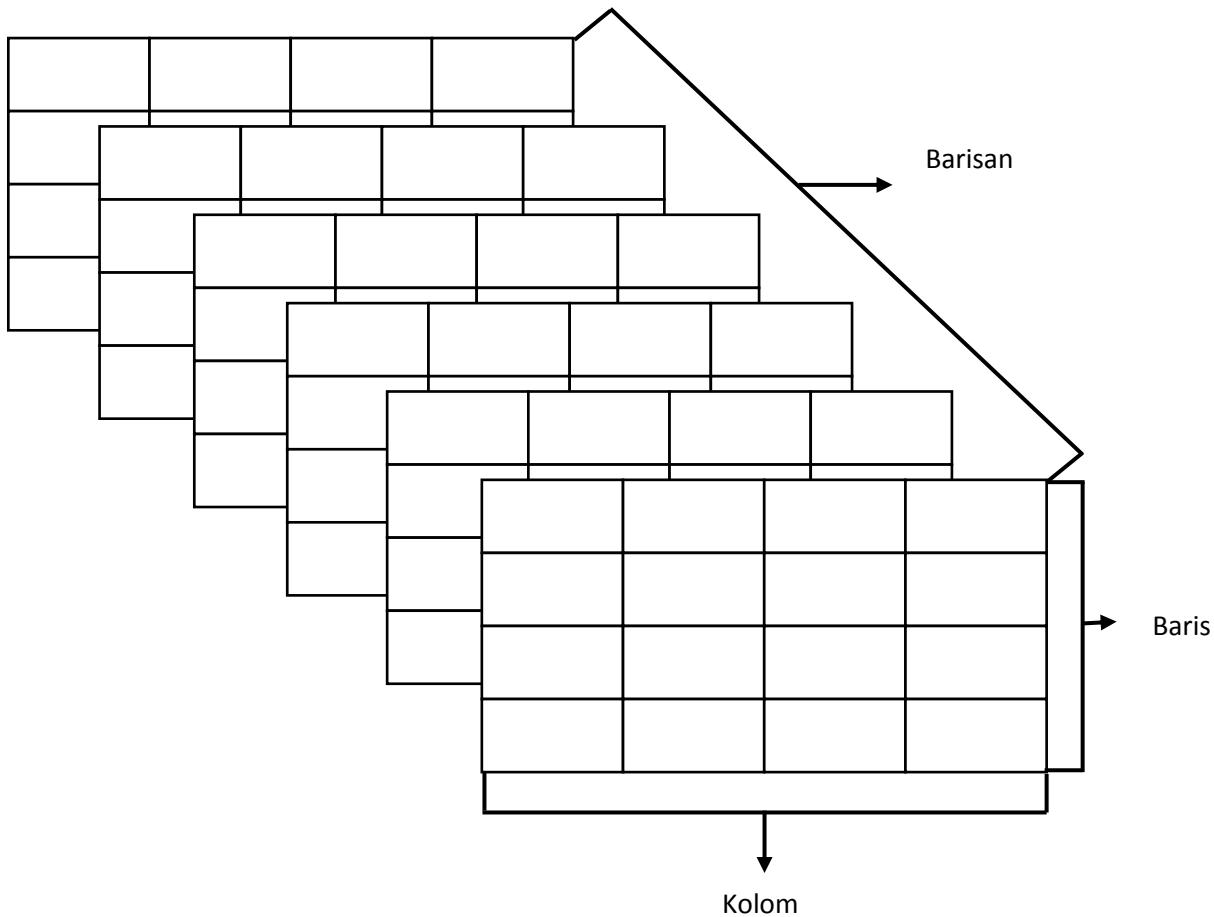
iAkhir adalah indeks akhir dari array

tipe_data adalah tipe data yang digunakan pada array

n adalah banyak dimensi array

Untuk menjelaskan array multidimensi kita akan memberikan contoh dalam array tiga dimensi.

Array tiga dimensi merupakan array yang memiliki tiga buah indeks dan lebih kompleks apabila dibandingkan dengan array dua dimensi. Di sini, indeks yang ada akan dipresentasikan dengan baris, kolom dan barisan seperti yang ditunjukkan oleh gambar berikut.



Gambar 2.10. Array Tiga Dimensi

Kita dapat melihat berdasarkan gambar tersebut, kita dapat melihat bahwa array tiga dimensi sebagai urutan 5 array dua dimensi, setiap elemennya mempunyai 3 baris dan 4 kolom. indeks pertama digunakan untuk mendesain barisan (rank), indeks kedua digunakan untuk mendesain baris dan indeks ketiga digunakan untuk mendesain kolom.

Berikut ini adalah contoh deklarasi variabel array tiga dimensi

```
var rumah_makan : array [1..2] of array [1..2] of array  
[1..4] of char;
```

Keterangan :

Array dengan nama **rumah_makan** memiliki tiga indeks, misalkan indeks pertama mewakili jenis menu, indeks kedua mewakili nama menu, indeks ketiga mewakili harga

Bentuk lain contoh deklarasi var pada array tiga dimensi:

```
Var rumah_makan : array [1..2,1..4,1..8] of char;
```

Untuk bentuk deklarasi type array tiga dimensi dapat dilihat pada contoh berikut:

```
type menu = array [1..2] of array [1..4] of array [1..6] of  
char;  
var rumah_makan : menu;
```

Bentuk lain contoh deklarasi type pada array tiga dimensi:

```
type menu = array [1..2,1..4,1..6] of char;  
var rumah_makan : menu;
```

Untuk memasukkan dan juga mengambil nilai dari array tiga dimensi pada dasarnya sama dengan pada array satu dimensi maupun array dua dimensi. Kita hanya perlu menambahkan indeksnya saja.

2.5.2.2. Menanya

Berdasarkan kegiatan mengamati, ada beberapa hal yang bisa diperhatikan dan dibahas lebih lanjut. Beberapa pertanyaan yang berkaitan dengan operator aritmatika adalah:

1. Bagaimana proses inisialisasi pada array tiga dimensi?

Untuk menjawab pertanyaan tersebut, siswa diharapkan bisa melakukan kegiatan mencoba berikut dengan bimbingan guru.

2.5.2.3. Mencoba

Percobaan 1

Untuk mengetahui jawaban dari pertanyaan pertama yaitu bagaimana proses inisialisasi pada array tiga dimensi, salin dan lengkapilah program berikut ke dalam Free Pascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program InitArray3d;

uses crt;

var
    tes_siswa : array[1..2,1..3,1..4] of real =
    (((60,75,70,75),(70,80,75,70),(80,80,85,90)),((60,75,
    70,75),(70,80,75,70),(80,80,85,90)));
begin
end.
```

Hasil Percobaan Program InitArray3d:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

2.5.2.4. Mengasosiasi/ menalar

Setelah siswa melakukan percobaan yang sesuai dengan pertanyaan yang ada, siswa diharapkan melakukan kegiatan menalar yang bisa dilakukan secara individu oleh masing-masing siswa atau bisa juga berkelompok. Beberapa hal yang bisa dijadikan arahan untuk menalar:

1. Perhatikan hasil kompilasi, apakah berhasil atau justru terjadi kesalahan.
2. Perhatikan urutan proses dari setiap program yang sudah dicoba.
3. Perhatikan inisialisasi array dua dimensi pada percobaan.
4. Cobalah membuat kesimpulan dari percobaan yang telah dilakukan.

2.5.3. Rangkuman

Dari percobaan-percobaan yang telah dilakukan, bisa ambil kesimpulan bahwa cara inisialisasi array tiga dimensi adalah

```
Var
    nama_array : array[iAwall..iAkhir1,iAwal2..iAkhir2] of
        tipe_data = (((data{iAwall,iAwal2,iAwal3},
        data{iAwall,iAwal2,iAwal3+1},data{iAwall,iAwal2,iAwal3+...},
        data{iAwall,iAwal2,iAkhir3}),(data{iAwall,iAwal2+1,iAwal3},
        data{iAwall,iAwal2+1,iAwal3+1},data{iAwall,iAwal2+1,iAwal3+...},
        data{iAwall,iAwal2+1,iAkhir3}),(data{iAwall,iAwal2+...,iAwal3},
        data{iAwall,iAwal2+...,iAwal3+1},data{iAwall,iAwal2+...,iAwal3+...},
        data{iAwall,iAwal2+...,iAkhir3}),(data{iAwall,iAkhir2,iAwal3},
        data{iAwall,iAkhir2,iAwal3+1},data{iAwall,iAkhir2,iAwal3+...},
        data{iAwall,iAkhir2,iAkhir3})),
        ((data{iAwall+1,iAwal2,iAwal3},data{iAwall+1,iAwal2,iAwal3+1},
        data{iAwall+1,iAwal2,iAwal3+...},data{iAwall+1,iAwal2,iAkhir3}),
        (data{iAwall+1,iAwal2+1,iAwal3},data{iAwall+1,iAwal2+1,iAwal3+1},
        data{iAwall+1,iAwal2+1,iAwal3+...},data{iAwall+1,iAwal2+1,iAkhir3}),
        (data{iAwall+1,iAwal2+...,iAwal3},data{iAwall+1,iAwal2+...,iAwal3+1},
        data{iAwall+1,iAwal2+...,iAwal3+...},
        data{iAwall+1,iAwal2+...,iAkhir3}),
        (data{iAwall+1,iAkhir2,iAwal3},data{iAwall+1,iAkhir2,iAwal3+1},
        data{iAwall+1,iAkhir2,iAwal3+...},data{iAwall+1,iAkhir2,iAkhir3})),
        ((data{iAwall+...,iAwal2,iAwal3},data{iAwall+...,iAwal2,iAwal3+1},
        data{iAwall+...,iAwal2,iAwal3+...},data{iAwall+...,iAwal2,iAkhir3}),
        data{iAwall+...,iAwal2+1,iAwal3},data{iAwall+...,iAwal2+1,iAwal3+1},
        data{iAwall+...,iAwal2+1,iAwal3+...},
        data{iAwall+...,iAwal2+1,iAkhir3}),
        (data{iAwall+...,iAwal2+...,iAwal3},
        data{iAwall+...,iAwal2+...,iAwal3+1},
        data{iAwall+...,iAwal2+...,iAwal3+...}),
```

```

data{iAwall+...,iAwall+...,iAkhir3}),
(data{iAwall+...,iAkhir2,iAwal3},
data{iAwall+...,iAkhir2,iAwal3+1},
data{iAwall+...,iAkhir2,iAwal3+...},
data{iAwall+...,iAkhir2,iAkhir3}),
((data{iAkhir1,iAwal2,iAwal3},data{iAkhir1,iAwal2,iAwal3+1},
data{iAkhir1,iAwal2,iAwal3+...},data{iAkhir1,iAwal2,iAkhir3}),
(data{iAkhir1,iAwal2+1,iAwal3},data{iAkhir1,iAwal2+1,iAwal3+1},
data{iAkhir1,iAwal2+1,iAwal3+...},data{iAkhir1,iAwal2+1,iAkhir3}),
(data{iAkhir1,iAwal2+...,iAwal3},data{iAkhir1,iAwal2+...,iAwal3+1},
data{iAkhir1,iAwal2+...,iAwal3+...},
data{iAkhir1,iAwal2+...,iAkhir3}),
(data{iAkhir1,iAkhir2,iAwal3},data{iAkhir1,iAkhir2,iAwal3+1},
data{iAkhir1,iAkhir2,iAwal3+...},data{iAkhir1,iAkhir2,iAkhir3}));
```

Keterangan :

nama_array adalah nama variabel array tiga dimensi
iAwall adalah indeks awal dari array dimensi pertama
iAkhir1 adalah indeks akhir dari array dimensi pertama
iAwal2 adalah indeks awal dari array dimensi kedua
iAkhir2 adalah indeks akhir dari array dimensi kedua
iAwal3 adalah indeks awal dari array dimensi ketiga
iAkhir3 adalah indeks akhir dari array dimensi ketiga
tipe_data adalah tipe data yang digunakan pada array tiga dimensi
data adalah data yang berada di dalam array tiga dimensi sebanyak indeks array.

2.5.4. Tugas

Perhatikan inisialisasi array berikut:

```
nilai : array [2..4,5..6,3..5] of integer = (((90,65,70),(85,70,75)),((90,65,70),(80,75,70)),
((85,70,75), (80,75,70)));
```

Tentukan nilai dalam elemen array berikut;

- nilai[3,5,3]
- nilai[2,6,4]
- nilai[4,5,5]

2.5.2. Uji Kompetensi/ Ulangan

Kerjakan soal berikut.

- Seorang pengusaha mempunyai 3 perusahaan di kota A dan 2 perusahaan di kota B, dimana masing-masing perusahaan mempunyai 5 orang karyawan. Buatlah program Pascal yang menampilkan semua karyawan yang dimiliki pengusaha tersebut.

Lembar kreativitas program

Program

Uses

Type

Var

Begin

End.

Output program:

2.6. Kegiatan Belajar 6. Pemanfaatan Array Multidimensi

Alokasi Waktu : 2 x 45 menit

2.6.1. Tujuan Pembelajaran

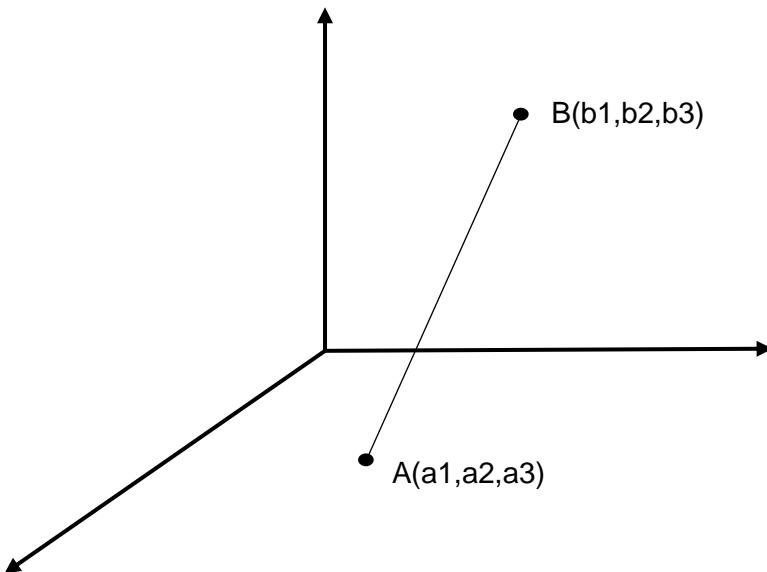
Setelah mempelajari materi pada kegiatan belajar ini, siswa diharapkan dapat menyajikan kumpulan data berdimensi banyak dalam bentuk array.

2.6.2. Aktivitas belajar siswa

2.6.2.1. Mengamati

Jarak dua buah titik pada ruang

Jarak antara dua buah titik merupakan panjang garis yang menghubungkan kedua titik, sehingga jarak antara dua titik adalah dengan menarik garis hubung terpendek antara kedua titik tersebut. Misalkan terdapat titik A dan titik B, maka jarak antara titik A dan B adalah panjang garis AB.



Gambar 2.11. Jarak Dua Titik Pada Ruang

Jika titik dalam koordinat cartesius maka jarak kedua titik adalah Rumus untuk menentukan jarak antara dua buah titik adalah sebagai berikut.

$$\text{Jarak} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

Keterangan:

Jarak = jarak dua titik

x_1 = nilai x pada titik pertama

x_2 = nilai x pada titik kedua

- y_1 = nilai y pada titik pertama
 y_2 = nilai y pada titik kedua
 z_1 = nilai z pada titik pertama
 z_2 = nilai z pada titik kedua

2.6.2.2. Menanya

Berdasarkan kegiatan mengamati, ada beberapa hal yang bisa diperhatikan dan dibahas lebih lanjut. Beberapa pertanyaan yang berkaitan dengan pemanfaatan array multidimensi adalah:

1. Jika diketahui titik P(3,2,3) dengan titik Q(8,1,7), berapakah jarak kedua titik pada ruang tersebut?

Untuk menjawab pertanyaan-pertanyaan tersebut, siswa diharapkan bisa melakukan kegiatan mencoba berikut dengan bimbingan guru.

2.6.2.3. Mencoba

Percobaan 1

Untuk mengetahui jawaban dari pertanyaan pertama yaitu menentukan berapakah jarak kedua titik pada ruang, salin dan lengkapilah program berikut ke dalam Free Pascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Jarak3d;
var
    a: array[1..2,1..2,1..2] of real;
    jar : real;
begin
    write('masukkan nilai x pada titik A : ');
    readln(a[1,1,1]);
    write('masukkan nilai y pada titik A : ');
    readln(a[1,1,2]);
    write('masukkan nilai z pada titik A : ');
    readln(a[1,2,1]);
    write('masukkan nilai x pada titik B : ');
    readln(a[1,2,2]);
    write('masukkan nilai y pada titik B : ');
    readln(a[2,1,1]);
    write('masukkan nilai z pada titik B : '');
```

```

readln(a[2,1,2]);

jar:=sqrt((sqr(a[1,1,1]-a[1,2,2]))+(sqr(a[1,1,2]-
a[2,1,1]))+sqr((a[1,2,1])-(a[2,1,2])));
writeln('jarak kedua titik adalah : ',jar:0:2);
readln;
end.

```

Hasil Percobaan Program 1:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

2.6.2.4. Mengasosiasi

Setelah siswa melakukan percobaan yang sesuai dengan pertanyaan yang ada, siswa diharapkan melakukan kegiatan menalar yang bisa dilakukan secara individu oleh masing-masing siswa atau bisa juga berkelompok. Beberapa hal yang bisa dijadikan arahan untuk menalar:

1. Perhatikan hasil kompilasi, apakah berhasil atau justru terjadi kesalahan.
2. Perhatikan urutan proses dari setiap program yang sudah dicoba.
3. Perhatikan jarak kedua titik dalam ruang pada percobaan.
4. Cobalah membuat kesimpulan dari percobaan yang telah dilakukan.

2.6.3. Rangkuman

Dari percobaan-percobaan yang telah dilakukan, bisa ambil kesimpulan bahwa:

1. Array tiga dimensi dapat digunakan untuk mencari jarak dua titik dalam ruang

2.6.4. Tugas

Siswa diminta untuk berkreasi membuat program yang memanfaatkan array tiga dimensi atau bisa juga array multidimesni.

2.6.5. Uji Kompetensi

1. Jika terdapat pabrik A yang terletak pada A(2,4,1) dan pabrik B terletak pada B(4,1,9). Buatlah program Pascal untuk menentukan letak gudang untuk menyimpan hasil produksi kedua pabrik, dimana lokasi gudang yang akan dibuat sama-sama dekat dari lokasi kedua pabrik.
2. Apakah yang dimaksud dengan inisialisasi array dan bagaimana melakukannya untuk array tiga dimensi?

Uji Kompetensi Bab II

Pilihan Ganda

Pilihlah salah satu jawaban yang paling tepat

1. Pengertian dari Array adalah....
 - a. Sekumpulan data angka
 - b. Sekumpulan data yang memiliki tipe sama
 - c. Sekumpulan data yang memiliki tipe data berbeda
 - d. Sekumpulan data yang memiliki tipe data heterogen
 - e. Sekumpulan data yang tidak memiliki tipe data
2. Pendeklarasian array yang benar adalah....
 - a. nama array : array[1..10,4..2] of integer;
 - b. nama array : array[1..10,2..5] ofinteger;
 - c. nama_array: array[1..10] of integer
 - d. nama array : array[1...10] of integer;
 - e. nama_array : array[1..10] of Integer;
3. Tentukan nilai elemen array pada data[7] pada program berikut adalah....

```
for i := 1 to n do
begin
    data[i]:=i;
end;
```

- a. 1
- b. 7

- c. 14
 d. 21
 e. 49
4. Nilai elemen array dalam matriks M untuk i ke-1 dan j ke-3 pada program berikut adalah

```

begin
    clrscr;
    {matriks M}
    m[1,1]:=70;  m[1,2]:=70;  m[1,3]:=65;
    m[2,1]:=80;  m[2,2]:=75;  m[2,3]:=85;
    for i:=1 to 3 do {baris matriks M}
    begin
        for j:=1 to 2 do {kolom matriks M}
        begin
            write(m[i,j], '');
        end;
        writeln;
    end;
    readln;
end.
```

- a. 65
 b. 70
 c. 75
 d. 80
 e. 85
5. Perhatikan inisialisasi array berikut:

```

nilai : array [2..4,5..6,3..5] of integer
= (((90,65,70),(85,70,75)),((90,65,70),(80,75,70)),
  ((85,70,75),(80,75,70)));
```

Nilai dalam elemen array nilai[4,6,4] adalah

- a. 65
 b. 70
 c. 75
 d. 80
 e. 85

Lembar kreativitas siswa pembuatan program array multidimensi.

Program

Uses

Type

Var

Begin

End.

Output program:

BAB

III

OPERASI STRING DAN KONVERSI DATA

Kompetensi Dasar:

- Memahami operasi string
- Mendeskripsikan konversi antar tipe data
- Menalar suatu alur proses kerja ke dalam bentuk kode program
- Memecahkan masalah konversi antar tipe data

BAB III. Operasi String Dan Konversi Data

Dewasa ini, banyak sekali informasi yang bentuknya adalah rangkaian kata. Informasi yang berbentuk rangkaian kata ini dapat kita temukan dalam banyak hal kehidupan sehari-hari kita, seperti buku, koran, web, dan lain-lain. Kata adalah rangkaian dari beberapa huruf atau karakter. Seiring kemajuan teknologi informasi, sudah banyak berkembang aplikasi-aplikasi berbasis kata yang digunakan dalam kehidupan sehari-hari.

Kita semua pasti sudah tidak asing lagi dengan aplikasi *Microsoft Word* yang mampu mengolah kata dan menampilkan hasilnya ke layar. Dalam *Microsoft Word*, terdapat proses manipulasi antara kata satu dengan kata yang lain, seperti contoh ketika proses cut, copy, dan paste. Proses tersebut dalam pemrograman merupakan manipulasi dari string. Selain itu terdapat proses mengubah kata, seperti contohnya Uppercase dan Lowercase. Masih banyak lagi proses pengolahan kata yang terdapat dalam *Microsoft Word*.

3.1. Kegiatan Belajar 1. Pengertian String

Alokasi Waktu : 2 x 45 menit

3.1.1. Tujuan Pembelajaran

Dalam kegiatan belajar ini siswa diharapkan dapat:

1. Memberikan pengertian string
2. Memahami konsep string

3.1.2. Aktivitas Belajar Siswa

3.1.2.1. Mengamati

Seperti yang kita bahas di awal, *Microsoft Word* terdapat dapat melakukan proses manipulasi antara kata satu dengan kata yang lain, seperti contoh ketika proses cut, copy, dan paste. Proses tersebut dalam pemrograman merupakan manipulasi dari string.

String adalah rangkaian karakter 8-bit ASCII dengan ukuran tertentu yang diapit dengan tanda kutip ('') dengan panjang maksimal adalah 255. Karakter tersebut dapat berupa huruf, angka, kosong, karakter khusus atau kombinasi dari semua. Ukuran dalam karakter tersebut dapat diubah oleh programmer sesuai dengan keinginan, akan tetapi jika ukuran tidak ditentukan maka ketentuan diatas yang berlaku.

Di bab sebelumnya kita telah membahas array. Dalam bab ini terdapat kaitannya antara array dan string, bahwa string bisa disebut sebuah array yang bertipe char.

```
var data: array [1..15] of char;
```

Sesuai dengan definisi string yang merupakan rangkaian karakter dengan ukuran tertentu, pada contoh array di atas kita hanya dapat memberikan karakter sesuai dengan ukuran tersebut.

Pendeklarasian string dapat dilakukan dengan cara sebagai berikut :

1. Deklarasi tipe

```
type namaTipe = string;  
  
var namaVar : namaTipe;
```

Keterangan:

namaTipe dan namaVar bisa diganti sesuai dengan kebutuhan

Contoh:

```
type StrTipe = string;  
  
var kata : StrTipe;
```

2. Deklarasi variabel

```
var namaVar: string;
```

Pendeklarasian string dapat dilakukan dengan menyertakan ukuran dari karakter. Misalkan kita deklarasikan seperti berikut:

```
var namaVar: string[ukuran];  
  
Keterangan: ukuran berupa bilangan bulat
```

Contoh:

```
var kata: string[15];
```

Pada contoh tersebut bisa diartikan membuat suatu string yang terdiri atas 15 karakter. Untuk ukuran karakter itu sendiri dapat kita ubah sesuai keinginan. Apabila ukuran karakter tidak didefinisikan, dalam bahasa Pascal menganggap ukurannya adalah 255 karakter (ukuran maksimum). String juga bisa diinisialisasi atau diberi nilai awal sehingga pada awalnya string tidak kosong atau berisi data yang tidak jelas. Cara inisialisasi string adalah sebagai berikut.

```
var kata : string ='ini isi awal';
```

Cara inisialisasi lain:

```
const  
    sdefault = 'Ini adala string diinisialisasi';  
var  
    s:string = sdefault;
```

Untuk

Untuk mengakses karakter pada variabel string, dapat menggunakan syntax berikut.

```
namaVariabelString[posisi_karakter]
```

Contohnya :

```
program nama;  
var  
    data: string[5]='tujuh';  
begin  
  
    write(data[3]);  
    readln;  
end.
```

data[1] adalah karakter pertama dari string

data[2] adalah karakter kedua dari string

data[3] adalah karakter ketiga dari string

data[4] adalah karakter keempat dari string

data[5] adalah karakter kelima dari string

Kita juga dapat mengakses semua karakter pada variabel string menggunakan perintah write atau writeln.

```
Write(var_string);  
Writeln(var_string);
```

3.1.2.2. Menanya

1. Apakah yang terjadi apabila kita mengakses elemen yang lebih besar dari ukuran string yang telah kita tentukan?
2. Jika suatu string ditentukan berukuran 10, kemudian dilakukan pemberian nilai dengan suatu kata yang memiliki 12 karakter, apakah yang terjadi ?
3. Jika ukuran string adalah 0, apakah yang terjadi ?
4. Jika ukuran string adalah negatif, apakah yang terjadi ?
5. Apakah yang terjadi apabila kita mengakses string pada posisi nol atau negatif ?

3.1.2.3. Mencoba

Percobaan 1

Untuk mengetahui jawaban dari pertanyaan pertama yaitu apakah yang terjadi apabila kita mengakses elemen yang lebih besar dari ukuran string yang telah kita tentukan, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Akses_string;
uses crt;
var data : string[10]='saya makan';
begin
    writeln(data[11]);
end.
```

Hasil Percobaan Program Akses_string :

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

Percobaan 2

Untuk mengetahui jawaban dari pertanyaan kedua yaitu apa yang terjadi jika suatu string ditentukan berukuran 10, kemudian dilakukan pemberian nilai dengan suatu kata yang memiliki 12 karakter, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Karakter_lebih;
uses crt;
var data : string[10];
begin
    write('Masukkan kata dg jumlah karakter lebih dari 10: ');
    readln(data);
    writeln(data);
end.
```

Hasil Percobaan Program Karakter_lebih :

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

```
program Init_lebih;
uses crt;
var data : string[10]='saya makan nasi goreng';
begin
end.
```

Hasil Percobaan Program Init_lebih :

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 3

Untuk mengetahui jawaban dari pertanyaan ketiga yaitu apa yang terjadi jika ukuran string adalah 0, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Ukuran_nol;
uses crt;
var data : string[0];
begin
    write('Masukkan kata: ');
    readln(data);
    writeln(data);
end.
```

Hasil Percobaan Program Ukuran_nol :

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

Percobaan 4

Untuk mengetahui jawaban dari pertanyaan keempat yaitu apa yang terjadi jika ukuran string adalah negatif, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Ukuran_negatif;
uses crt;
var data : string[-1];
begin
    write('Masukkan kata: ');
    readln(data);
    writeln(data);
end.
```

Hasil Percobaan Program Ukuran_negatif :

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

Percobaan 5

Untuk mengetahui jawaban dari pertanyaan kelima yaitu apakah yang terjadi apabila kita mengkases string pada posisi nol? Salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program StringPosisiNol;
uses crt;
var data : string[10]='saya makan';
begin
    writeln(data[0]);
end.
```

Hasil Percobaan Program StringPosisiNol :

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

```
program StringPosisiNegatif;
uses crt;
var data : string[10];
begin
    data := 'saya makan nasi goreng';
    writeln(data[-1]);
end.
```

Hasil Percobaan Program StringPosisiNegatif :

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

3.1.2.4. Menalar

Pada Percobaan 1 dan Percobaan 2 perhatikan dan bandingkan setiap pesan hasil kompilasi. Lihat posisi/letak kesalahan berada pada baris dan kolom berapa pada baris perintah yang ada ! Kemudian perhatikan hasil eksekusi pada Percobaan 2, informasi apakah yang dapat kamu ambil ?

Pada Percobaan 3 dan Percobaan 4 perhatikan dan bandingkan setiap pesan hasil kompilasi. Lihat posisi/letak kesalahan berada pada baris dan kolom berapa pada baris perintah yang ada ! Adakah kesamaan kesalahan yang terdapat pada kedua percobaan tersebut ? Informasi apakah yang dapat kamu ambil ?

Pada Percobaan 5 nomor 1 dan 2, perhatikan hasil eksekusinya ! Adakah kesamaan hasilnya? Informasi apakah yang dapat kamu ambil ?

3.1.2. Rangkuman

1. String adalah barisan character 8-bit ASCII yang diapit dengan tanda kutip ('') dengan panjang maksimal adalah 255. Ukuran ini dapat diubah oleh programmer sesuai dengan keinginan, akan tetapi jika tidak ukuran tidak ditentukan maka ketentuan diatas yang berlaku.
2. String bisa ditentukan ukurannya sesuai kebutuhan program. Inisialisasi dan assignment melebihi ukuran maka program hanya akan menampilkan karakter sesuai dengan ukuran yang telah ditentukan.
3. Untuk menentukan ukuran karakter dari string tidak diperkenankan memberikan ukurannya 0 (nol) atau negatif.
4. Mengakses string pada posisi nol akan memberikan karakter null (karakter kosong, tidak ada karakter) dan mengakses pada posisi negatif tidak diperbolehkan.

3.1.4. Tugas

Tugas 1 : Individu

Buatlah sebuah program sederhana berkaitan dengan string dengan ketentuan berikut:

1. Program yang dapat menginputkan nama lengkap, TTL, dan alamat rumah.
2. Buatlah ke dalam pilihan menu!

Tugas 2 : Kelompok

Buatlah sebuah program sederhana berkaitan dengan string dengan ketentuan berikut:

1. Dapat menentukan banyaknya sebuah karakter dari rangkaian karakter yang diberikan.
2. Buatlah ke dalam pilihan menu!

3.1.5. Uji Kompetensi

1. Jelaskan apa yang anda ketahui tentang string !
2. Jelaskan teknik pendeklarasian string !
3. Bagaimana caramekakses string !

3.2. Kegiatan Belajar 2. Operasi string

Alokasi Waktu : 2 x 45 Menit

3.2.1. Tujuan Pembelajaran

Dalam kegiatan belajar ini siswa diharapkan dapat:

1. Mengidentifikasi operasi penggabungan pada string
2. Mengidentifikasi operasi pemecahan pada string

3.2.2. Aktivitas Belajar Siswa

3.2.2.1. Mengamati

Seperti yang kita bahas di awal, *Microsoft Word* terdapat dapat melakukan proses manipulasi antara kata satu dengan kata yang lain, seperti contoh ketika proses cut, copy, dan paste. Proses tersebut dalam pemrograman merupakan manipulasi dari string. String itu sendiri memiliki banyak fungsi, akan tetapi pada Kegiatan Belajar kali ini kita akan membahas beberapa operasi pada string.

Penggabungan String

Dalam string operasi penggabungan string disebut append. Operasi penggabungan adalah proses menggabungkan dua atau lebih data yang bertipe string menjadi sebuah string yang baru. Prosedurnya bernama concat. Berikut adalah sintaknya :

```
concat(string1, string2, ..., stringN);
```

Keterangan :

- string1, nilai string ke satu yang akan digabungkan
- string2, nilai string ke dua yang akan digabungkan
- stringN, nilai string ke N yang akan digabungkan

Berikut adalah contoh operasi penggabungan string :

```
program penggabungan;
var n,m: string;
begin
  n:='aris';
  m:='wara';
  //menggabungkan variabel n dan m menggunakan concat
  writeln(concat(n,m));
end.

//output yang dikeluarkan adalah "ariswara"
```

Penghapusan Substring

Operasi penghapusan adalah proses menghapus sebagian karakter atau substring yang dimulai dari posisi x sebanyak n karakter. Prosedurnya bernama delete. Berikut adalah sintaknya :

```
delete(string, x, n);
```

Keterangan :

- string, nilai yang string akan dihapus substringnya
- x, posisi awal substring yang akan dihapus
- n, banyaknya karakter yang akan dihapus

Berikut adalah contoh penghapusan substring :

```
program hapus;
var n: string;
begin
    n:='aris';
    //dihapus substring dari string dengan variabel n
    delete(n,2,2);
    writeln(n);
end.

//output yang dikeluarkan adalah "as"
```

Menyisipkan Substring

Operasi penyisipan adalah proses menyisipkan sebagian karakter atau substring ke dalam substring yang lain yang dimulai dari posisi x. Prosedurnya bernama insert. Berikut adalah sintaknya :

```
insert(string1,string2,n);
```

Keterangan :

- string1, nilai string yang akan disisipkan
- string2, nilai string yang akan mendapat sisipan dari string1
- n, posisi karakter string2 yang akan mendapat sisipan dari string1

Berikut adalah contoh penyisipan substring :

```
program sisipan;
var n,m: string;
begin
    n:='aris';
    m:='wara';
    //disisipkan string variabel m ke string variabel n
    insert(m,n,2);
    writeln(n);
end.

//output yang dikeluarkan adalah "awararis"
```

Penyalinan Substring

Operasi penyalinan adalah proses menyalin sebagian karakter atau substring yang dimulai dari posisi x sebanyak n karakter. Prosedurnya bernama copy. Berikut adalah deklarasinya :

```
copy(string,x,n);
```

Keterangan :

- string, nilai string akan disalin
- x, posisi awal substring yang akan disalin
- n, banyaknya karakter yang akan disalin

Berikut adalah contoh penyisipan substring :

```
program copy;
var n,m: string;
begin
    n:='ariswara';
    //disalin substring dari string variabel n
    m:=copy(n,2,2);
    writeln(m);
end.

//output yang dikeluarkan adalah "ri"
```

Mendapatkan Posisi String

Operasi ini berfungsi untuk mencari posisi suatu substring di dalam string dengan output data berupa integer. Operasi ini akan mengembalikan nilai indeks substring ke dalam string jika string mengandung substring yang dicari. Dalam susbstring tidak ditemukan, nilai 0 akan dikembalikan. Prosedurnya bernama pos. Berikut adalah deklarasinya :

```
pos(substring,string);
```

Keterangan :

- string, nilai string akan dicari posisi substringnya
- substring, nilai substring yang akan dicari posisinya

Berikut adalah contohnya :

```
program posisi_string;
var n: string;
begin
    n:='ariswara sita riyanda';
    //dicari substring sita dalam string variabel n
    writeln(pos('sita',n));
end.

//output yang dikeluarkan adalah "10"
```

Menghitung Panjang String

Menghitung panjang string atau banyaknya karakter dalam string dapat dilakukan menggunakan fungsi length. Berikut deklarasi dari fungsi length :

```
length(string);
```

Keterangan :

- string, nilai string akan dicari panjangnya

Berikut adalah contohnya :

```
program panjang_string;
var n: string;
    panj: integer;
begin
    n:='ariswara sita riyanda';
    panj:=length(n);
    writeln(panj);
end.

{output yang dikeluarkan adalah "21" }
```

Membandingkan Dua Buah String

Membandingkan elemen dua string dapat dilakukan dengan cara membandingkan banyaknya elemen karakter dalam string terlebih dahulu untuk kemudian membandingkan tiap elemen dalam string. Berikut adalah contohnya :

```

program perbandingan_string;
uses crt;
var i:integer;
    s,t:string;
begin
    s:='sekolah';
    t:='sekolah';
    if length(s)=length(t) then
    begin
        writeln('string s : ',s);
        writeln('string t : ',t);
        //membandingkan string s dengan string t
        //memanfaatkan panjang string
        for i:=1 to length(s) do
        begin
            if s[i]=t[i] then
            begin
                writeln('elemen ke ',i,' adalah ',t[i]);
            end
            else
            begin
                writeln('elemen ke ',i,' adalah ',t[i],' tidak sama');
            end;
        end;
        readln;
    end.

//output yang dikeluarkan adalah:
string s : sekolah
string x : SekOlaH
elemen ke 1 adalah s
elemen ke 2 adalah e
elemen ke 3 adalah k
elemen ke 4 adalah u tidak sama
elemen ke 5 adalah l
elemen ke 6 adalah a
elemen ke 7 adalah h

```

Dan berikut ini adalah beberapa operasi operasi string yang lain.

No	Fungsi dan Prosedur String
1.	Fungsi AnsiCompareStr (const s1:; const s2:): integer; Membandingkan dua buah string
2.	Fungsi AnsiCompareText (const s1:; const s2:): integer; Membandingkan dua buah string yang tidak sensitif terhadap huruf

3.	Fungsi AnsiExtractQuotedStr (var Src: PChar; Quote: Char);; Menghapus kutipan dalam string
4.	Fungsi AnsiLastChar (const s:): PChar; Mengambil karakter terakhir dari string
5.	Fungsi AnsiLowerCase (const s:): Mengubah string menjadi huruf kecil semua
6.	Fungsi Ansi QuotedStr (const s:; Quote: Char);; Kutipan sebuah string
7.	Fungsi AnsiStrComp (s1: PChar; s2: PChar): integer; Membandingkan string yang sensitif terhadap huruf
8.	Fungsi AnsiStrIComp (s1: PChar; s2: PChar): integer: Membandingkan string yang tidak sensitif terhadap karakter
9.	Fungsi AnsiStrLComp (s1: PChar; s2: PChar; MaxLen: Cardinal): integer; Membandingkan L karakter dari string yang sensitif terhadap huruf
10.	Fungsi AnsiStrLIComp (s1: PChar; s2: PChar; MaxLen: Cardinal): integer; Membandingkan L karakter dari string yang tidak sensitif terhadap huruf
11.	Fungsi AnsiStrLastChar (Str: PChar): PChar; Mengambil karakter terakhir dari string
12.	Fungsi AnsiStrLower (Str: PChar): PChar; Mengubah string menjadi huruf kecil semua
13.	Fungsi AnsiStrUpper (Str: PChar): PChar; Mengubah string menjadi huruh besar(kapital) semua
14.	Fungsi AnsiUpperCase (const s):; Mengubah string menjadi huruf besar(kapital) semua
15.	Prosedur AppendStr (var Dest:; const s:); Menyisipkan dua karakter string
16.	Prosedur AssignStr (var P: PString; const s:); Memberikan nilai dari string yang berada pada rangkaian
17.	Fungsi CompareStr (const s1:; const s2:): integer; overload; Membandingkan dua string yang sensitif terhadap huruf
18.	Fungsi CompareText (const s1:; const s2:): integer; Membandingkan dua string yang tidak sensitif terhadap huruf
19.	Prosedur DisposeStr (s : PString); overload; Menghapus string dari rangkaian
20.	Prosedur DisposeStr (s: PShortString); overload; Menghapus string dari rangkaian
21.	Fungsi IsValidIdent (const Ident:): Boolean;

	String pengidentifikasi bahasan pascal yang sah
22.	Fungsi LastDelimiter (const Delimiters: ; const s:): integer; Memunculkan karakter terakhir di dalam string
23.	Fungsi LeftStr (const s:; count: integer);; Mengambil N karakter pertama dalam string
24.	Fungsi LoadStr (Ident: integer);; Membuka string dari asalnya
25.	Fungsi LowerCase (const s):; overload; Mengubah string menjadi huruf kecil semua
26.	Fungsi LowerCase (constV: variant);; overload; Mengubah string menjadi huruf kecil semua
27.	Fungsi NewStr (const s:): PString; overload; Menempatkan string baru ke dalam rangkaian
28.	Fungsi RightStr (const s:; count: integer);; Mengambil N karakter terakhir dari string
29.	Fungsi StrAlloc (size: cardinal): PChar; Menempatkan memori dari string
30.	Fungsi StrBufSize (Str: PChar): SizeUInt; Mencadangkan memori dari string
31.	Prosedur StrDispose (Str: PChar); Menghapus string dari rangkaian
32.	Fungsi StrPas (Str: PChar);; Mengubah Pchar menjadi string
33.	Fungsi StrPCopy (Dest: PChar; Source:): PChar; Menyalin string
34.	Fungsi StrPLCopy (Dest: PChar; Source:; MaxLen: SizeUInt): PChar; Menyalin N bytes dari string
35.	Fungsi UpperCase (const s:);; Mengubah string menjadi huruf besar(kapital) semua

3.2.2.2. Menanya

1. Jika dua string kita dikenai operator (+), apakah yang terjadi?
2. Coba kita cari kata ‘Malang’ dari kata ‘Pemalang’, apakah yang terjadi ?
3. Akan kita bandingkan kata surakarta dengan SURAKARTA, apakah memiliki nilai yang sama atau tidak ?

3.2.2.3. Mencoba

Percobaan 1

Untuk mengetahui jawaban dari pertanyaan pertama yaitu penggunaan operator ‘+’ pada dua string, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
Program StringPlus;
var nama1,nama2,nama3, data : string;
begin
    nama1:='aris';
    nama2:='wara';

    nama3:=nama1+nama3;
    writeln(nama3);

    data:='1''+'2';
    writeln(data);
    readln;
end.
```

Hasil Percobaan StringPlus

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 2

Untuk mengetahui jawaban dari pertanyaan kedua, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan

kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Cari_substring;
var n: string;
begin
    n:='Pemalang';
    //dicari substring Malang dalam string Pemalang
    writeln(pos('Malang',n));
end.
```

Hasil Percobaan Program Cari_substring :

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 3

Untuk mengetahui jawaban dari pertanyaan ketiga, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program BandingString;
uses sysutils;
var kota1,kota2: string;
    banding:longint;
begin
    kota1:='SURAKARTA';
    kota2:='surakarta';
```

```
//membandingkan kota1 dengan kota2  
banding:=comparestr(kota1,kota2);  
write(kota1, ' adalah ');  
if banding<0 then  
begin  
    write('kurang dari ');\nend  
else if banding=0 then  
begin  
    write('sama dengan ');\nend  
else  
begin  
    write('lebih dari ');\nend; writeln(kota2);  
readln;  
end.
```

Hasil Percobaan Program BandingString :

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

3.2.2.4. Mengasosiasi

Pada Percobaan 1, perhatikan hasil eksekusinya ! Adakah kesamaan hasilnya ? Informasi apakah yang dapat kamu ambil ?

Pada Percobaan 2, perhatikan hasil eksekusinya ! Informasi apakah yang dapat kamu ambil ? Mengapa tidak ditemukan kata ‘Malang’ dari string ‘Pemalang’?

Pada Percobaan 3, perhatikan hasil eksekusinya ! Informasi apakah yang dapat kamu ambil?

3.2.3. Rangkuman

1. Jika sebuah tipe data string dideklarasikan, tipe data string memiliki panjang maksimal 30 karakter dimulai dari indeks ke-1 hingga indeks ke-30. Indeks ke-0 merupakan indeks yang mengandung informasi banyaknya karakter yang tersimpan dalam string. Oleh karena itu, untuk mengetahui panjang karakter dalam string digunakan kode untuk mengakses elemen string indeks ke-0.
2. Untuk menggabungkan dua buah string menjadi string yang baru operasi string memiliki operator (+). Di dalam string, operator string (+) hanya berfungsi untuk menggabungkan dua buah string bukan untuk menjumlahkan.
3. Dalam string mengandung karakter 8 bit ASCII dimana karakter huruf kapital nilainya berbeda dengan huruf kecil. Jika kita membandingkannya huruf kapital tidak sama dengan huruf kecil.
4. Dalam menentukan posisi substring, apabila susbstring yang kita cari tidak ditemukan, nilai 0 akan dikembalikan atau saat kita eksekusi outputnya adalah 0.
5. Untuk membandingkan dua buah string, operator =, <, >, >=, <=, dan <> dapat digunakan untuk membandingkan elemen karakter dalam string.

3.2.4. Tugas

Tugas 1 : Individu

Buatlah sebuah progam sederhana memanfaatkan operasi string untuk mengambil karakter kanan dan kiri dari sebuah rangkaian karakter !

Tugas 2 : Kelompok

Buatlah sebuah progam sederhana memanfaatkan operasi string untuk membalikkan kata, misal makan menjadi nakam.

3.2.5. Uji Kompetensi

1. Jelaskan prosedur penggabungan dua buah string ?
2. Jelaskan prosedur menghitung panjang string ?
3. Sebutkan dan jelaskan operator perbandingan string ?

3.3. Kegiatan Belajar 3. Pemanfaatan String Dalam Aplikasi

Alokasi Waktu : 2 x 45 menit

3.3.1. Tujuan Pembelajaran

Dalam kegiatan belajar ini siswa diharapkan dapat:

- 1) Mengetahui penerapan dari operasi string
- 2) Membuat aplikasi sederhana dengan menerapkan operasi string

3.3.2. Aktivitas Belajar Siswa

3.3.2.1. Mengamati

Operasi string mempunyai banyak manfaatnya di kehidupan sehari-hari. Kita telah mengetahui bahwa palindrom adalah sebuah kata, frasa, angka maupun susunan lainnya yang dapat dibaca dengan sama baik dari depan maupun belakang (spasi antara huruf-huruf biasanya diperbolehkan). Kata-kata yang termasuk palindrom misalnya: ada, apa, ara, asa, bab, ini, katak, kodok, makam, malam, radar, taat, dll. Dalam bab sebelumnya kita juga telah membuat pemrograman palindrom.

3.3.2.2. Menanya

1. Bagaimana membuat pemrograman palindrom menggunakan operasi string?
2. Bagaimana penerapan string dalam aplikasi bisnis ? Berikut adalah daftar gaji sebuah perusahaan berdasarkan jabatannya

Jabatan	Gaji	Tunjangan
Direktur	5000000	25% gaji
Manager	3000000	10% dari gaji
Karyawan	1500000	5% dari gaji

Dikenakan PPN sebesar 10% dan gaji bersih didapat dari gaji ditambah tunjangan dikurangi PPN. Hitung gaji bersih tiap jabatan tersebut !

3.3.2.3. Mencoba

Percobaan 1

Untuk mengetahui jawaban dari pertanyaan pertama, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program StringPalindrom;
var i,n: integer;
    kata1,kata2:string;
begin
    //input kata yg akan dicek
    writeln('masukkan kata: ');
    readln(kata1);
    kata2:='';
    //copy kata1 ke kata2 secara urutan terbalik
```

```

n:=length(kata1);
for i:=... downto ... do
kata2:=kata2+kata1[i];
//bandingkan kedua kata
if ..... then
  write('palindrom')
else
  write('bukan palindrom');
readln;
end.

```

Hasil Percobaan Program StringPalindrom

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 2

Untuk mengetahui jawaban dari pertanyaan kedua, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```

program Gajil;
uses crt;
var nama,jabatan:string;
    gaji:longint;
    tunj,ppn,total:real;
begin
    //input nama dan jabatan
    write('Masukkan Nama : '); readln(nama);
    write('Masukkan Jabatan : '); readln(jabatan);
    //membuat rumus gaji sesuai jabatan

```

```
if  (jabatan='direktur') then
begin
    gaji:.....;
    tunj:=0.25*gaji;
end else
    if  (jabatan='manager') then
begin
    gaji:.....;
    tunj:=0.1*gaji;
end else
begin
    gaji:.....;
    tunj:=0.05*gaji;
end;
//tampilkan rincian gajinya
writeln('Gaji Kotor      : Rp.',gaji);
writeln('Tunjangan Jabatan   : Rp.',tunj:9:2);
ppn:=0.1*....;
writeln('PPN 10 %       : Rp.',ppn:9:2);
total:=(gaji+tunj)-ppn;
writeln('Gaji bersih    : Rp.',total:9:2);
readln;
end.
```

Hasil Percobaan Program Gaji1

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

```

program Gaji2;
uses crt;
var nama,jabatan:string;
    gaji:longint;
    tunj,ppn,total:real;
begin
    write('Masukkan Nama      : '); readln(nama);
    write('Masukkan Jabatan   : '); readln(jabatan);
    if LowerCase(jabatan)='direktur' then
    begin
        gaji:.....;
        tunj:=0.25*gaji;
    end
    else
        if LowerCase(jabatan)='Manager' then
        begin
            gaji:.....;
            tunj:=0.1*gaji;
        end else
        begin
            gaji:.....;
            tunj:=0.05*gaji;
        end;
    //tampilkan rincian gajinya
    writeln('Gaji Kotor      : Rp.',gaji);
    writeln('Tunjangan Jabatan   : Rp.',tunj:9:2);
    ppn:=0.1*....;
    writeln('PPN 10 %       : Rp.',ppn:9:2);
    total:=(gaji+tunj)-ppn;
    writeln('Gaji bersih     : Rp.',total:9:2);
    readln;
end.

```

Hasil Percobaan Program Gaji2

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

3.3.2.4. Mengasosiasi

Pada Percobaan 1, perhatikan hasil eksekusinya ! Kemudian mengapa kita perlu mengkopi kata1 ke kata2 dengan urutan terbalik ? Informasi apakah yang dapat kamu ambil ?

Pada Percobaan 2, perhatikan dan bandingkan hasil eksekusinya ! Kemudian mengapa kita perlu fungsi LowerCase? Informasi apakah yang dapat kamu ambil ?

3.3.3. Rangkuman

1. Untuk membuat program yang dapat memeriksa sebuah kata apakah termasuk palindrom atau bukan kita dapat memanfaatkan salah satu operasi string yaitu panjang string. langkah awalnya kita inputkan sebuah kata, kemudian kata tersebut kita cari panjangnya memanfaatkan operasi length. Setelah itu kita copy kata tersebut dengan urutan terbalik. Jika pada saat dibalik kata tersebut memiliki kesamaan rangkaian elemen pada saat kita inputkan, maka kata tersebut bisa disebut palindrom. Dan jika pada saat dibalik kata tersebut tidak memiliki kesamaan rangkaian elemen pada saat kita inputkan, maka kata tersebut tidak bisa disebut palindrom.
2. Pada percobaan program perhitungan gaji. Untuk membuatnya, langkah awal yang kita akan kita lakukan adalah menginputkan nama kemudian jabatannya. Acuan penghitungan gaji adalah berdasar jabatan. Oleh karena itu, dengan memanfaatkan rumus if then kita buat aturan untuk merumuskan gaji berdasarkan jabatannya. Kita juga menggunakan fungsi LowerCase dikarenakan untuk mencegah eksekusi yang tidak berjalan dikarenakan dalam string bersifat case insensitive atau terdapat perbedaan karakter antara huruf kapital dan huruf kecil yang dapat memungkinkan user menginputkan beberapa kemungkinan penulisan. Langkah terakhir dari program ini yaitu menampilkan rincian gajinya berdasarkan rumus yang telah kita tentukan.

3.3.4. Tugas

Tugas Kelompok

Sebuah perusahaan memiliki 5 unit penjualan dengan setiap unit memiliki laba (jabarkan labanya). Kemudian akan dicari laba total, rata-rata laba, dan unit mana yang labanya sama dengan rata-rata laba, unit mana yang labanya di bawah rata-rata laba, dan unit mana yang labanya di atas rata-rata laba. Buatlah program untuk kasus tersebut.

3.4. Kegiatan Belajar 4. Konversi Data

Alokasi Waktu : 2 x 45 menit

3.4.1. Tujuan Pembelajaran

Dalam kegiatan belajar ini siswa diharapkan dapat:

- 1) Mengetahui pengertian konversi data
- 2) Mengidentifikasi operasi konversi data

3.4.2. Aktivitas Belajar Siswa

3.4.2.1. Mengamati

Konversi data adalah mengadaptasi tipe data tertentu pada sebuah program agar dapat dioperasikan ke dalam tipe data yang lain dengan fungsi dan prosedur yang sesuai.

Mengkonversi data dari String ke Integer

Fungsi untuk konversi dari string ke integer adalah

```
StrToInt(const s: string): integer
```

StrToInt akan mengkonversi string ke integer. Jika string mengandung karakter yang tidak valid atau memiliki format yang tidak valid, maka Error akan muncul. Untuk berhasil dikonversi, string dapat berisi kombinasi karakter numerik, mungkin diawali dengan tanda minus (-). Spasi tidak diperbolehkan. String S dapat berisi angka dalam desimal, heksadesimal, format biner atau oktal.

Berikut adalah contoh pemrogramannya :

```
program konversi;
uses sysutils;
var a,b:string[10];
    i,j,k:integer;
begin
    a:='1';
    b:='2';
    //mengkonversi string ke integer
    i:=StrToInt(a); writeln(i);
    j:=StrToInt(b); writeln(j);
    k:=i*j;
    writeln(k);
    readln;
end.
```

Mengkonversi data dari string ke integer juga dapat menggunakan prosedur berikut :

```
val(string,var_string,kode);
```

Keterangan :

- string, nilai string yang akan dikonversi
- var_string, variabel tempat penyimpanan hasil konversi
- kode, variabel tempat penyimpanan kode kesalahan

Berikut contoh pemrogramannya :

```
program mencoba;
var s: String;
    i: integer;
    x,j,kode: integer;
begin
    s:='100';
    //mengkonversi string variabel s menjadi integer
    val(s,i,kode);
    x:=100;
    j:=x + i;
    writeln('hasil penjumlahan x dan n adalah', j);
    readln;
end.
```

Mengkonversi data dari Integer ke String

Fungsi untuk konversi dari integer ke string adalah

```
IntToStr(value: string): integer
```

IntToStr mengkonversi numerik ke representasi string itu. String yang dihasilkan hanya sebanyak karakter yang diperlukan untuk mewakili nilai. Jika nilai negatif tanda minus akan ditambahkan ke string.

Berikut adalah contoh pemrogramannya :

```
program konversi;
uses sysutils;
var a,b,k:string[10];
    i,j:integer;
begin
    i:=11;
    j:=22;
    //mengkonversi integer ke string
    a:=IntToStr(i); writeln(a);
    b:=IntToStr(j); writeln(b);
    k:=a+b;
    writeln(k);
end.
```

Mengkonversi data dari integer ke string juga dapat menggunakan prosedur berikut :

```
str(x[:lebar[:desimal]],string);
```

Keterangan :

- x, nilai integer yang akan dikonversi
- lebar, menyatakan lebar hasil string
- desimal, menentukan banyak digit dari x
- string, variabel tempat menyimpan hasil konversi

Berikut contoh pemrogramannya :

```
program konversi;
var a,b,k:string;
    i,j:integer;
begin
    i:=1000;
    j:=200;
    //mengkonversi integer ke string
    str(i,a); writeln(a);
    str(j,b); writeln(b);
    k:=a+b;
    writeln(k);
end.
```

Mengkonversi data dari String ke Float

Fungsi untuk konversi dari string ke float adalah

```
StrToFloat(const s: string): extended
```

StrToFloat mengkonversi string S ke float (real). S harus berisi representasi string yang valid dari nilai float (baik dalam desimal atau notasi ilmiah).

Mengkonversi data dari Float ke String

Fungsi untuk konversi dari float ke string adalah

```
FloatToStr(value: double): string
```

FloatToStr mengkonversi variabel float (real) ke string.

Berikut adalah contoh pemrogramannya :

```
program konversi;
uses sysutils;
var a,b,k:string[10];
    i,j:real;
begin
    i:=1.25;
    j:=-2.24;
    //mengkonversi float ke string
    a:=FloatToStr(i); writeln(a);
    b:=FloatToStr(j); writeln(b);
    k:=a+b;
    writeln(k);
readln;
end.
```

Mengkonversi data dari String ke Boolean

Fungsi untuk konversi dari string ke boolean adalah

```
StrToInt(const s: string): boolean
```

StrToInt akan mengkonversi string S ke nilai boolean. String S dapat berisi salah satu dari *True* atau *False* atau nilai numerik. Jika berisi nilai numerik 0 akan diubah menjadi *False*, dan semua nilai-nilai yang lain adalah *True*. Jika string S tidak mengandung boolean yang valid, maka pengecualian EconvertError akan muncul.

Mengkonversi data dari Boolean ke String

Fungsi untuk konversi dari boolean ke string adalah

```
BoolToStr(b: boolean; useboolstrs:boolean): string
```

BoolToStr mengkonversi boolean ke salah satu string *True* atau *False*.

3.4.2.2 Menanya

- 1) Bagaimana kita mengakses fungsi konversi data tanpa assignment?
- 2) Apakah yang dimaksud dengan karakter yang tidak valid dalam konversi string menjadi integer sehingga menyebabkan error saat dieksekusi?

3.4.2.3 Mencoba

Percobaan 1

Untuk mengetahui jawaban dari pertanyaan pertama, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Konversi;
uses sysutils;

begin
    //mengkonversi string ke integer
    writeln(StrToFloat('1,22'));
    writeln(StrToFloat('2,11'));
readln;
end.
```

Hasil Percobaan program

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 2

Untuk mengetahui jawaban dari pertanyaan kedua, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program KarakterInvalid1;
uses sysutils;
var i:integer;
begin
```

```
//mengkonversi string ke integer  
i:=StrToInt('1234567890');  
writeln(i);  
readln;  
end.
```

Hasil Percobaan Program KarakterInvalid1 :

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

```
program KarakterInvalid2;  
uses sysutils;  
var i:integer;  
begin  
    //mengkonversi string ke integer  
    i:=StrToInt('1a');  
    writeln(i);  
    readln;  
end.
```

Hasil Percobaan Program KarakterInvalid2 :

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

3.4.2.4 Menalar

Pada Percobaan 1, perhatikan hasil eksekusi ! Informasi apakah yang dapat kamu ambil ?

Pada percobaan 2, perhatikan setiap pesan pada hasil kompilasi ! Kemudian perhatikan dan bandingkan hasil eksekusinya ! Apabila terdapat error, pesan apa yang tercantum pada error tersebut ? Informasi apa yang dapat kamu ambil ?

3.4.3 Rangkuman

1. Untuk mengakses hasil konversi data tidak harus dengan assignment.
2. Karakter yang tidak valid dalam konversi string menjadi integer sehingga menyebabkan error saat dieksekusi adalah ketika kita menginputkan karakter yang tidak sesuai dengan variabel. Misalkan dalam kita mendeklarasikan sebuah variabel ke dalam integer, karakter yang kita inputkan bertipe longint. Maka pada saat dieksekusi, akan muncul error.

3.4.4. Tugas

Tugas 1 : Individu

Buatlah sebuah program sederhana memanfaatkan konversi untuk mengubah string ke dalam biner.

Tugas 2 : Kelompok

Buatlah sebuah program sederhana memanfaatkan konversi data dengan kombinasi konversi data string ke integer dan string ke float.

3.4.5. Uji Kompetensi

1. Jelaskan apa yang dimaksud dengan konversi data ?
2. Jelaskan dua cara mengkonversi string menjadi integer ?
3. Jelaskan dua cara mengkonversi integer menjadi string ?

Catatanku:

3.5. Kegiatan Belajar 5. Pemanfaatan Konversi Data Dalam Aplikasi

Alokasi Waktu : 2 x 45 menit

3.5.1. Tujuan Pembelajaran

Dalam kegiatan belajar ini siswa diharapkan dapat:

- 1) Mengetahui penerapan dari operasi string
- 2) Membuat aplikasi sederhana dengan menerapkan operasi string

3.5.2. Aktivitas Belajar Siswa

3.5.2.1. Mengamati

Kita pasti tahu bahwa kita memiliki Jembatan Suramadu yang menghubungkan Pulau Jawa da Pulau Madura. Pembangunan jembatan tersebut pasti memerlukan biaya yang sangat mahal. Dengan memanfaatkan konversi data kita dapat memprediksi biaya yang dikeluarkan pemerintah untuk mrmbangun Jembatan Suramadu. Pada kegiatan belajar kali ini kita dapat menerapkan konversi data dengan mengkonversi sebuah nilai string ke bentuk tipe data integer atau real dan sebaliknya untuk menyelesaikan persoalan sehari-hari.



Gambar 3.1. Jembatan Suramadu (<http://id.wikipedia.org>)

3.5.2.2. Menanya

1. Panjang dari Jembatan Suramadu adalah 6km=6000m. Misalkan biaya pembangunan jalan per meter sebesar Rp 100 juta. Berapa biaya minimal yg harus disiapkan untuk membangun Jembatan Suramadu tersebut ? Akan kita inputkan nilai string kemudian dikonversi ke dalam integer untuk menghitungnya !

2. Dua kota A berlokasi di titik (x1,y1) dan kota B berlokasi di titik (x2,y2) akan dihubungkan dengan jalan tol. Biaya pembangunan jalan per meter sebesar Rp 100 juta. Berapa biaya minimal yg harus disiapkan untuk membangun jalan tol tersebut ? Akan kita inputkan nilai string kemudian dikonversi ke dalam integer untuk menghitungnya !
3. Bagaimana penerapan string dalam aplikasi bisnis ? Berikut adalah daftar gaji sebuah perusahaan berdasarkan jabatannya

Jabatan	Gaji	Tunjangan
Direktur	5000000	25% gaji
Manager	3000000	10% dari gaji
Karyawan	1500000	5% dari gaji

Dikenakan PPN sebesar 10% dan gaji bersih didapat dari gaji ditambah tunjangan dikurangi PPN. Hitung gaji bersih tiap jabatan tersebut !

3.5.2.3. Mencoba

Percobaan 1

Untuk mengetahui jawaban dari pertanyaan pertama, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Suramadu;
uses sysutils;
var i,biaya: longint;
    panj:string;
begin
    //input panjang jembatan dengan nilai string
    panj:='6000';
    writeln('Panjang jembatannya adalah : ',.....);
    //ubah nilai panj ke dalam integer
    i:=StrToInt(....);
    //hitung biaya yang dikeluarkan
    biaya:=100*i;
    writeln('Biaya Jembatan Suramadu sebesar : ',.....,' juta');
    readln;
end.
```

Hasil Percobaan Program Suramadu ;

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 2

Untuk mengetahui jawaban dari pertanyaan kedua, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program BiayaTol;
uses sysutils;
var a,b,c,d,jarak,biaya:real;
    x1,y1,x2,y2:string;
begin
    //input x1,y1,x2,y2 dengan nilai string
    writeln('masukkan koordinat titik-titiknya');
    writeln('x1: ');readln(...);
    writeln('y1: ');readln(...);
    writeln('x2: ');readln(...);
    writeln('y2: ');readln(...);

    //konversi x1,y1,x2,y2 menjadi tipe data real
    a:=StrToFloat(...);
    b:=StrToFloat(y1);
    c:=StrToFloat(...);
    d:=StrToFloat(y2);

    //menghitung jarak dua titik dari nilai konversi string
    jarak:=sqrt(...);
    writeln('jaraknya adalah: ',jarak:3:4);
```

```
//menghitung biaya minimum yang disiapkan
biaya:=100*....;
writeln('biaya yang dipersiapkan: ',biaya:3:2,'juta');
readln;
end.
```

Hasil Percobaan Program BiayaTol :

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 1

Untuk mengetahui jawaban dari pertanyaan kedua yaitu Apakah yang terjadi apabila kita memberikan nilai dengan null karakter? Salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program HitGaji;
uses sysutils;
var nama,jabatan,gaji:string;
    gajil:longint;
    tunj,ppn,total:real;
begin
    //input nama dan jabatan
    writeln('Masukkan Nama      : '); readln(.....);
    writeln('Masukkan Jabatan   : '); readln(.....);
    //membuat rumus gaji sesuai jabatan
    if  LowerCase(jabatan)='direktur' then
```

```
begin
    //konversi nilai string menjadi integer
    gaji:='5000000';
    gajil=StrToInt(.....);
    tunj:=0.25*.....;
end else
    if .....(jabatan)='manager' then
begin
    //konversi nilai string menjadi integer
    gaji:='3000000';
    gajil=StrToInt(.....);
    tunj:=0.1*.....;
end else
begin
    //konversi nilai string menjadi integer
    gaji:='1500000';
    gajil=StrToInt(.....);
    tunj:=0.05*.....;
end;
//tampilkan rincian gajinya
writeln('Gaji Kotor      : Rp.',gaji);
writeln('Tunjangan Jabatan   : Rp.',tunj:9:2);
ppn:=0.1*.....
writeln('PPN 10 %       : Rp.',ppn:9:2);
total:=(gaji+tunj)-ppn;
writeln('Gaji bersih    : Rp.',total:9:2);
readln;
end.
```

Hasil Percobaan Program HitGaji :

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

3.5.2.4 Mengasosiasi

Pada Percobaan 1, perhatikan eksekusinya ! Informasi apakah yang dapat kamu ambil ?

Pada Percobaan 2, perhatikan eksekusinya ! Kemudian mengapa kita perlu fungsi or ? Informasi apakah yang dapat kamu ambil ?

Pada Percobaan 3, perhatikan eksekusinya ! Kemudian mengapa kita perlu fungsi or ? Informasi apakah yang dapat kamu ambil ?

3.5.3. Rangkuman

1. Dalam membuat pemrograman untuk memprediksi biaya yang dikeluarkan pemerintah saat membuat Jembatan Suramadu, langkah pertama adalah menginputkan panjang dari Jembatan Suramadu dalam nilai string kemudian kita konversi menjadi integer agar dapat kita hitung biayanya. Langkah yang terakhir adalah menghitung biayanya dengan rumus yang telah kita tentukan.
2. Dalam membuat program menentukan biaya pembangunan jalan tol yang menghubungkan jarak antara dua titik untuk menerapkan konversi data di dalamnya kita perlu menginputkan semua titik-titiknya dalam variabel nilai string kemudian kita konversi menjadi tipe data real karena titik-titik dalam koordinat dapat berupa bilangan real oleh karena itu kita memerlukan operasi konversi data string to float. Selanjutnya kita hitung jaraknya menggunakan rumus jarak antara dua titik. Langkah terakhir dengan menghitung biayanya berdasarkan rumus yang kita tentukan.
3. Pada percobaan program perhitungan gaji. Untuk membuatnya, langkahnya seperti pada kegiatan belajar sebelumnya. Langkah awal yang kita akan kita lakukan adalah menginputkan nama kemudian jabatannya dan membuat semua inputannya ke dalam string. Pada ini acuan yang kita gunakan adalah besarnya gaji berdasarkan jabatannya. Oleh itu, dengan memanfaatkan rumus if then kita buat aturan untuk merumuskan gaji berdasarkan jabatannya. Selanjutnya variabel gaji dengan nilai string akan kita konversi menjadi bentuk integer. Langkah terakhir dari program ini yaitu menampilkan rincian gajinya berdasarkan rumus yang telah kita tentukan.

3.5.4. Tugas

Tugas Kelompok

Sebuah perusahaan memiliki 5 unit penjualan dengan setiap unit memiliki laba (jabarkan labanya). Kemudian akan dicari laba total, rata-rata laba, dan unit mana yang labanya sama dengan rata-rata laba, unit mana yang labanya di bawah rata-rata laba, dan unit mana yang labanya di atas rata-rata laba. Buatlah program yang memanfaatkan konversi data untuk kasus tersebut.

3.5.5. Uji Kompetensi

1. Jelaskan pengertian dari string !

2. Jelaskan beberapa teknik mendeklarasikan string !
3. Jelaskan dua metode menghitung panjang string !
4. Jelaskan perbedaan operasi perbandingan pada string !
5. Jelaskan perbedaan operasi penggabungan string menggunakan operator dan fungsi concat !
6. Jelaskan apa yang dimaksud dengan palindrom !
7. Jelaskan apa yang dimaksud dengan konversi data !
8. Jelaskan fungsi yang digunakan untuk mengkonversi string ke integer !
9. Jelaskan fungsi yang digunakan untuk mengkonversi string ke float !
10. Jelaskan kegunaan fungsi LowerCase!

BAB

IV

POINTER

Kompetensi Dasar:

- Memahami konsep pointer
- Memodifikasi data melalui pointer

BAB 4. POINTER

Pada Bab sebelumnya telah dijelaskan mengenai variabel yang bertipe Array, suatu tipe data yang bersifat statis (ukuran dan urutannya sudah pasti). Salah satu kelemahan data bertipe array adalah bahwa variabel tersebut harus dideklarasikan terlebih dahulu dengan menyebutkan ukurannya. Dengan demikian ukuran variabel tersebut tidak dapat kita ubah selama program dijalankan. Lalu bagaimakah kalau kita akan membuat program untuk menangani data dalam jumlah yang besar? Bagaimana pula kalau jumlah datanya tidak diketahui sebelumnya?. Bagaimana cara ruang memori yang dipakai oleh variabel tersebut dapat dihapus jika variabel tersebut tidak digunakan kembali..

Pertanyaan pertama mungkin dapat diselesaikan dengan memesan variabel array dengan ukuran sebanyak data yang akan dimasukkan. Tetapi sebenarnya ide ini belum tentu dapat dijalankan (terutama kalau implementasi programnya menggunakan bahasa yang mensyaratkan ukuran maksimum variabel dalam suatu program, seperti Pascal yang membatasi jumlah ukuran semua variabel dalam suatu program tidak boleh melebihi 64 Kb). Sebagai contoh, berikut ini adalah deklarasi dalam Pascal yang akan ditolak:

1. var banyak : array[1..35000] of integer;
2. var larik : array[1..12000] of real;
3. var larik1 : array[1..30000] of char;
4. var larik2 : array[1..35000] of byte;

Pendeklarasian di atas akan ditolak oleh Pascal karena:

1. ukuran variabel banyak adalah 35000×2 byte = 70000 byte
2. ukuran variabel larik adalah 12000×6 byte = 72000 byte
3. ukuran variabel larik1 = 30000×1 byte = 30000 byte dan ukuran variabel larik2 = 35000×1 byte = 35000 byte, sehingga total ukuran kedua variabel adalah 65000 byte

Permasalahan kedua juga menyulitkan pemrograman. Bila kita alokasikan variabel dengan ukuran yang besar sementara nantinya data yang dipakai hanya sedikit maka ini berarti pemborosan memori (memori sudah dipesan tetapi tidak dipergunakan) sedang bila dialokasikan dengan ukuran kecil sementara nantinya data yang digunakan cukup banyak sehingga deklarasi yang dibuat tidak cukup maka tentunya permasalahan tidak akan terselesaikan.

Untuk mengatasi permasalahan di atas, kita dapat menggunakan pointer yang bersifat dinamis. Variabel akan dialokasikan hanya pada saat dibutuhkan dan setelah tidak digunakan dapat didealokasikan lagi.

5.1. Kegiatan Belajar 1. Pointer ke Variabel

Alokasi Waktu : 2 x 45 menit

5.1.1. Tujuan Pembelajaran

Tujuan pembelajaran pada Kegiatan Belajar 1 tentang Pointer ke Variabel adalah:

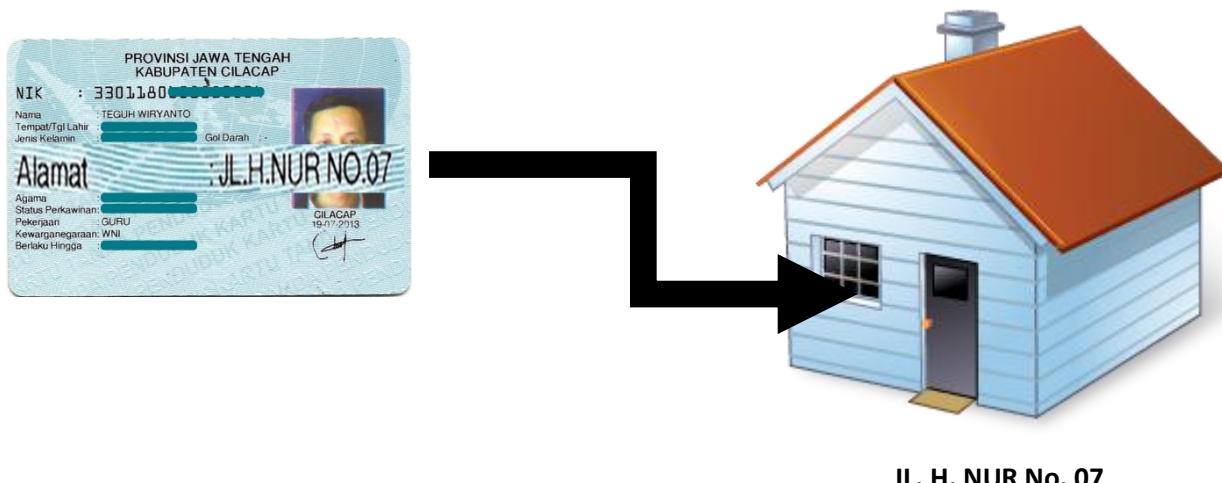
1. Siswa dapat memahami Pembuatan pointer ke variabel dan penyimpanannya pada memori
2. Siswa dapat menganalisis kegunaan pointer dalam program untuk mengendalikan variabel dan manajemen memori

5.1.2. Aktivitas belajar siswa

5.1.2.1. Mengamati

Definisi Pointer

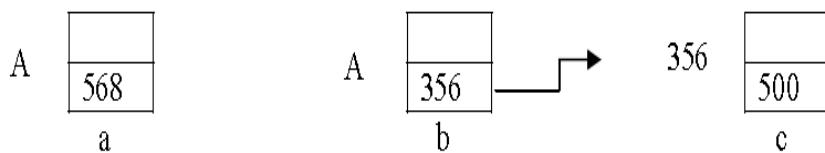
Pointer sering disebut juga dengan istilah link, referensi, penunjuk adalah suatu variabel yang menyimpan atau menunjuk alamat dari suatu variabel yang lain. Pointer menunjuk suatu data yang tersimpan dalam sebuah variabel dengan cara menyimpan alamat dari variabel penyimpan data tersebut. Pointer dalam kehidupan sehari – hari dianalogikan seperti kartu alamat seseorang (KTP, atau yg lain) maka kartu tersebut dapat menunjukkan dimana orang tersebut berada, bagaimana caranya? ya tentu saja dengan menyimpan alamat tersebut pada kartu.



Gambar 4.1. Analogi Pointer

(Sumber https://www.iconfinder.com/icons/34226/home_house_icon)

Secara umum, dalam pembicaraan tentang pointer, tipe data pointer digambarkan sebagai tanda panah sedang variabel yang ditunjuk digambarkan sebagai kotak. Suatu pointer dapat menunjuk ke data bertipe elementer, terstruktur, pointer yang lain, atau tidak bertipe. Jika suatu pointer tidak menunjuk ke mana-mana, pointer itu dinamakan dangling, sedangkan bagian memori yang tidak dapat diakses karena tidak ada pointer yang menunjuk dinamakan garbage (sampah).



Gambar 5.2. Pembuatan dan Penugasan Variabel Biasa dan Pointer

Gambar a merupakan variabel statis yang berisi suatu nilai statis, sedangkan variable A pada gambar b merupakan variable dinamis yang menunjuk pada suatu alamat tertentu (356), dimana alamat yang ditunjuk tersebut berisi nilai data 500 (gambar c). Nilai data yang ditunjuk ini biasanya disebut node atau simpul.

Sehingga secara umum perbedaan antara variabel bertipe Array dan variabel bertipe Pointer adalah sebagai berikut:

Kriteria	Array	Pointer
Sifat	Statis	Dinamis
Ukuran	Pasti	Sesuai Kebutuhan
Alokasi Memori	Saat program dijalankan sampai dengan selesai	Dapat diatur sesuai dengan kebutuhan

Deklarasi Pointer ke Variabel

Pendeklarasian variabel Pointer tidak jauh berbeda dengan pendeklarasian variabel biasa, hanya perlu ditambahkan simbol topi (^), biasa juga disebut ceret, topi atau circumflex. Simbol topi tersebut menandakan bahwa variabel tersebut menunjuk ke lokasi tertentu di memori.

Deklarasi secara umum adalah sebagai berikut :

Cara 1. Deklarasi type diikuti dengan ddeklarasi variabel:

```
Type
    nama_pointer:^tipe_data;
Var
    nama_var_pointer:nama_pointer;
```

Cara 2. Deklarasi variabel secara langsung:

```
Var
    nama_pointer:^tipe_data;
```

Keterangan:

nama_pointer adalah definisi pointer pada type

nama_var_pointer adalah deklarasi variabel pointer

tipe_data adalah tipe data dari variabel yang ditunjuk variabel pointer

Contoh 1: Pendeklarasian type pointer

```
Type
  Ptrinteger = ^integer;
Var
  Ptrbilangan : Ptrinteger;
  Bilangan : integer;
```

Keterangan:

Ptrinteger adalah type variabel pointer yang menunjuk suatu variabel bertipe integer
Ptrbilangan adalah variabel pointer dengan type **Ptrinteger**
Bilangan adalah variabel data yang bertipe integer

Contoh 2: Pendeklarasian variabel pointer

```
Var
  JumlahData : ^Integer;
  {variabel JumlahData menunjuk data bertipe
  Integer}
  NamaSiswa : ^String[25];
  {variabel NamaSiswa menunjuk data bertipe String
  dengan panjang karakter maksimum 25}
  NilaiSiswa : ^Real;
  {Variabel NilaiSiswa menunjuk data bertipe Real}
```

Keterangan :

Variabel **JumlahData**, **NamaSiswa**, **NilaiSiswa** adalah variabel pointer yang menunjuk tipe data yang berbeda(**integer**, **String**, **Real**).

Mengisi Suatu Variabel Pointer

Variabel Pointer berisi alamat dari suatu variabel yang lain. Adapun cara untuk mengisikan alamat dari suatu variabel ke variabel pointer adalah sebagai berikut:

```
var_pointer:=@variabel_data;
```

Keterangan:

var_pointer adalah nama variabel pointer
variabel_data adalah nama variabel yang alamatnya akan disimpan pointer

Contoh 3

```
program IsiPointer;
{ Membuat program pascal dengan nama IsiPointer }
uses crt;
{ Memanggil unit crt yang kita gunakan untuk memanggil
fungsi clrscr yang berfungsi untuk membersihkan layar }
Var
    Ptrbilangan : ^integer;
    { Mendeklarasikan Variabel Ptrbilangan menunjuk tipe
data integer }
    Bilangan : integer;
    { Mendeklarasikan Variabel Bilangan bertipe integer }
Begin
    clrscr;
    { Membersihkan tampilan layar }
    Bilangan := 10;
    { Memberi nilai 10 pada variabel Bilangan }
    Writeln ('Isi Variabel Bilangan =', Bilangan);
    { Menampilkan nilai/isi dari variabel Bilangan }
    Ptrbilangan :=@Bilangan ;
    { Variabel Ptrbilangan menunjuk alamat variabel
Bilangan }

    Writeln ('Nilai yang ditunjuk oleh Ptrbilangan
= ', ptrbilangan^);
    { Mengambil isi data dari variabel yang ditunjuk
pointer Ptrbilangan }

    Readln;
    { Perintah ini digunakan untuk menunggu user menekan
tombol dari keyboard }

End.
```

Keterrangan:

1. Variabel Bilangan diberi nilai awal 10 dengan perintah penugasan (:=)
2. Variabel pointer Ptrbilangan Menunjuk atau menyimpan alamat dari variabel Bilangan (Ptrbilangan :=@Bilangan;)
3. Untuk menampilkan isi atau nilai yang ditunjuk variabel pointer Ptrbilangan menggunakan perintah ptrbilangan^

4. Program tersebut jika dijalankan akan memberikan keluaran:

Isi Variabel Bilangan = 10

Nilai yang ditunjuk oleh Ptr Bilangan = 10

Perlu diingat fungsi @ hanya menunjuk alamat dari suatu variabel bukan menunjuk nilainya secara langsung. Adapun untuk mengambil isi data yang ditunjuk variabel pointer dapat digunakan perintah berikut:

var_pointer^

Menginisialisasi Pointer

Menginisialisasi pointer artinya memberi isi alamat mana yang akan ditunjuk. Tidak melakukan hal ini akan mengakibatkan pointer berada di sembarang alamat. Kalau kebetulan alamat yang ditunjuk daerah kosong, maka program akan berjalan baik, tapi kalau alamat yang ditunjuk sudah digunakan untuk sesuatu yang penting, program akan terhenti (halt).

Contoh 4

<pre>program Tanpa_inisialisasi; Var P : ^real; Begin P^ :=13; ... end.</pre>	<pre>program Dengan_inisialisasi; Var P : ^real; I : integer; Begin I :=13; {Mulai baris ini } P :=@I ; {inisialisasi dilakukan...} end.</pre>
---	--

Variabel pointer yang tidak dipakai sebaiknya diisi sesuatu yang kira-kira artinya tidak menunjuk kemana-mana. Pascal menyediakan nilai tersebut pada variabel NIL. Kita dapat mengisi pointer dengan NIL sebelum memakainya. . Ini dapat dilakukan dengan memberikan assignment sebagai berikut:

var_Pointer:=nil

Perlu dibedakan pengertian variabel pointer yang bernilai NIL dengan variabel pointer yang tak terdefinisi (tak tentu). Variabel pointer bernilai NIL berarti ia tidak menunjuk ke suatu variabel dinamis, sedang variabel pointer yang bernilai tak tentu berarti ia mungkin mengacu (menunjuk) ke suatu lokasi sembarang di memori.

Contoh 5. Nil Pointer

```

program NilPointer;
{Membuat program pascal dengan nama NilPointer }

var
    iptr: ^integer;
    { Mendeklarasikan Variabel pointer iptr yang
      menunjuk tipe data integer }

    y: ^word;
    { Mendeklarasikan variabel pointer y yang
      menunjuk ke tipe data word}

begin
    iptr := nil;
    { Memberi nilai nil pada variabel pointer iptr }
    y := addr(iptr);
    { Variabel menunjuk alamat dari variabel pointer
      iptr}
    writeln('Alamat dari Variabel yang ditunjuk iptr
= ', y^);
    { Menampilkan alamat dari variabel yang ditunjuk
      iptr}
    Readln;
end.

```

Keterangan:

1. Pointer iptr diinisiasikan sebagai Nil pointer,
2. Variabel pointer y menyimpan alamat dari pointer iptr
3. Selanjutnya ketikan menampilkan alamat yang ditunjuk pointer iptr adalah 0, dengan kata lain pointer iptr tidak menunjuk lokasi manapun.

Menampilkan Alamat Pointer

Pada pascal disediakan fungsi untuk menampilkan alamat dari suatu pointer, sehingga keberadaannya kita ketahui. Adapun perintah yang digunakan adalah:

addr (Var_Pointer)

Contoh 6:

```
program AlamatPointer;
var
    nilai: integer;
    score: ^integer;
    alamat: ^word;

begin
    nilai := 500;
    writeln('Nilai =', nilai);
    score := @nilai;
    { Mengisi variabel pointer dengan alamat variabel
bilai}
    writeln('Score Menuju Nilai = ', score^);
    { Mengambil isi data yang ditunjuk pointer }
    score^ := 1000;
    writeln('Nilai Sekarang = ', Nilai);
    writeln('Score Menuju Nilai = ', score^);
    alamat := addr(score);
    writeln('Alamat Variabel score = ', alamat^);
    readln;
end.
```

Keterangan:

1. Variabel `nilai` diberi nilai awal 500 dengan operasi penugasan (`:=`)
2. Variabel pointer `score` menunjuk alamat dari variabel `nilai`, yaitu dengan perintah `score := @nilai`
3. Program memanipulasi isi dari variabel `nilai` dengan menggunakan pointer yaitu dengan perintah `score^:=1000`, hal ini berarti mengubah isi dari variabel `nilai` menjadi 1000
4. Program menampilkan alamat dari variabel yang ditunjuk pointer `score` dengan perintah `addr(score)`
5. Program tersebut jika dijalankan akan memberikan keluaran:
Nilai = 500
Score Menuju Nilai = 500
Nilai Sekarang = 1000

Score Menuju Nilai 1000

Alamat Variabel score = 36864

Alokasi Memori Untuk Pointer

Pascal menyediakan prosedur untuk memberi alamat pada pointer dan sekaligus mengalokasikan sejumlah byte di memori untuk dipakai menyimpan data yang ditunjuk oleh pointer yang bersangkutan.

Prosedur ini akan mengalokasikan sejumlah memori pada bagian HEAP dan sering disebut sebagai Variabel Dinamis

1. Menggunakan New dan Dispose

Prosedur ini akan mengalokasikan memori dan mengeset pointer untuk menunjuk ke alamat baru tersebut. Pointer `Var_pointer` bertipe pointer sehingga dapat digunakan untuk mengalokasikan berbagai tipe (char, integer, real, record maupun lainnya), adapun cara penggunaanya adalah sebagai berikut :

```
New (Var_pointer);
```

Keterangan:

`Var_pointer` adalah nama variabel pointer yang akan dialokasikan.

Prosedur `New` digunakan untuk memesan memori untuk pointer bertipe, prosedur ini akan membentuk suatu variabel dinamik yang diletakkan dalam Heap. Heap adalah memori-memori di komputer yang belum dialokasikan, yaitu memori yang tidak digunakan oleh DOS, oleh program-program resident, oleh program Pascal dan variabel-variabel di data segmen.

Suatu pointer yang telah dialokasikan pada waktu program dieksekusi, dapat didealokasikan kembali pada saat eksekusi program tersebut. Setelah suatu pointer dihapus, maka lokasi yang semula ditempati oleh variabel yang ditunjuk oleh pointer tersebut akan bebas dan dapat digunakan oleh peubah yang lain Deklarasi penghapusan suatu pointer adalah sebagai berikut :

```
Dispose (Var_pointer);
```

Keterangan:

`Var_pointer` adalah nama variabel pointer yang akan didealokasikan atau dihapus.

Contoh 7:

```
program NewdanDispose;
{Membuat program pascal dengan nama NewdanDispose }
Var
  Ptrdata : ^string;
{Mendeklarasikan variabel pointer Ptrdata yang}
```

```

        menunjuk tipe data integer}

Begin
    New (Ptrdata);
    {Memesan memori/ mengalokasikan memori untuk
     variabel pointer Ptrdata}
    Ptrdata^:= 'Teguh';
    {Menginisiasi nilai yang ditunjuk pointer
     Ptrdata}
    writeln('Nilai yang ditunjuk pointer =
', Ptrdata^);
    {Menampilkan nilai atau isi yang ditunjuk
     variabel pointer Ptrdata }
    Dispose(Ptrdata);
    {Menghapus atau mendealokasikan memori dari
     variabel pointer Ptrdata}
    readln;
End.

```

Keterangan:

1. program mengalokasikan pointer baru bertipe string dengan nama Ptrdata dengan perintah New (Ptrdata)
2. Pointer tersebut menunjuk suatu alamat pada heap yang berisi string 'Teguh'
3. Program menampilkan isi dari Alamat yang ditunjuk oleh pointer dengan perintah Ptrdata^
4. Setelah pointer Ptrdata selesai digunakan, program menghapus memori penyimpanan dari pointer tersebut dengan perintah dispose(Ptrdata). Sehingga beban memori pada heap menjadi berkurang.
5. Jika program di atas dijalankan maka akan memberikan keluaran sebagai berikut:
Nilai yang ditunjuk pointer = Teguh

2. Menggunakan Getmem dan Freemem

Untuk menggunakan Getmem dapat digunakan perintah sebagai berikut:

```
getmem(Var_ptr, Ukuran);
```

Keterangan:

Var_ptr adalah nama Variabel Pointer yang akan dialokasikan

Ukuran adalah jumlah byte yang akan dialokasikan (max 64 KB)

Metode ini memberikan kebebasan menentukan besar byte jumlah memori yang akan dialokasikan pada sebuah variabel pointer. Sedangkan cara untuk mendealokasikannya dapat digunakan perintah Freemem sebagai berikut:

```
freemem(var_ptr, ukuran);
```

Keterangan:

Var_ptr adalah nama Variabel Pointer yang akan didealokasikan

Ukuran adalah jumlah byte yang akan didealokasikan (max 64 KB)

Contoh 8:

```
Program alokasiDgnGetMem;
Uses crt;
Const
    MaxHeapAlloc = 65521 ; {maksimal +64KB}
Type
    Arraysemu = array [1..1] of real;
Var
    p : ^arraysemu;
    I,ukuran, totalbyte : longint;

Begin
    Clrscr;
    Write ('Berapa Ukuran array yang diinginkan :');
    readln(ukuran);
    totalbyte :=ukuran * SizeOf (real);
    If (MemAvail > totalbyte) and (totalbyte <
MaxHeapAlloc)
        begin
            GetMem( p, totalbyte );
            For I := 1 to ukuran do p^[I] :=9;
            For I := 1 to ukuran Do write (p^[I]:4:0);
            FreeMem( p, totalbyte );
        End
    Else
        Writeln ('terlalu besar, memori yang ada
tidak tersedia sebesar itu');
End.
```

3. Menggunakan MARK dan RELEASE

Untuk menggunakan Mark dapat digunakan perintah sebagai berikut:

```
Mark (Var_ptr);
```

Keterangan:

Var_ptr adalah Variabel Pointer

Prosedur ini hanya digunakan untuk melakukan perekaman keadaan heap memori ke dalam sebuah pointer. Artinya, apabila kita akan mengalokasikan beberapa buah pointer dengan prosedur New atau GetMem, kita tidak perlu repot melakukan dealokasi satu persatu. Sedangkan cara mendealokasikannya adalah :

```
Release (var_ptr);
```

Perintah Release akan menghapus seluruh variabel dinamis yang terletak antara yang ditunjuk oleh var_ptr sampai sebelum yang ditunjuk oleh HEAPPTR.

Operasi pada Pointer

1. Operasi Penugasan

Suatu variable pointer seperti halnya variable yang lain, juga bisa mengalami operasi penugasan. Nilai dari suatu variable pointer dapat disalin ke variable pointer yang lain. Operasi mengkopi pointer akan mengakibatkan sebuah variabel akan ditunjuk oleh lebih dari satu pointer.

Bentuk deklarasi statemen:

```
T1:=T2;
```

Keterangan:

T1 dan T2 adalah nama dari Variabel pointer

Alamat dari Variabel pointer T2 disimpan dalam variabel pointer T1, dengan kata lain, sekarang variabel pointer menunjuk alamat yang sama dengan variabel pointer T2

2. Operasi Logika

Operasi logika juga dapat dilakukan pada sebuah variabel pointer. Berikut ini contoh operasi untuk membandingkan isi variabel:

```
If T1 = T2 then ..... .
While T1 <> T2 then .....
While T2 <> nil do .....
```

3. Operasi Aritmatika

Suatu variabel pointer hanya dapat dilakukan operasi aritmatika dengan nilai integer saja. Operasi yang biasa dilakukan adalah operasi penambahan satu (increment), pengurangan satu (decrement), + dan -. Operasi penambahan satu menunjukkan lokasi data berikutnya (index selanjutnya) dalam memori. Begitu juga operasi pengurangan satu. Untuk Operasi Arimatika akan dibahas lebih dalam pada kegiatan belajar 2.

4.1.2.2. Menanya

Berdasarkan kegiatan mengamati, ada beberapa hal yang bisa diperhatikan dan dibahas lebih lanjut. Beberapa pertanyaan yang berkaitan dengan Pointer ke Variabel adalah:

1. Bagaimanakan jika pointer yang menunjuk tipe data integer, digunakan untuk menunjuk alamat dari variabel dengan tipe data yang lain, misalnya string? apakah yang akan terjadi.
2. Dapatkah kita mengubah alamat atau lokasi yang ditunjuk oleh suatu pointer? Dan apakah nilai yang ditunjuk pointer tersebut juga berubah?
3. Dapatkah kita mengubah nilai dari variabel yang ditunjuk oleh suatu pointer menggunakan pointer?
4. Jika ada dua pointer menunjuk lokasi yang sama, apakah nilai dari variabel yang ditunjuk pointer juga sama?
5. Dapatkah kita melakukan operasi dalam pascal hanya menggunakan variabel pointer, tanpa menggunakan variabel dengan tipe data tertentu seperti integer, string, array ataupun yang lain?

Untuk menjawab pertanyaan-pertanyaan tersebut, siswa diharapkan bisa melakukan kegiatan mencoba berikut dengan bimbingan guru.

4.1.2.3. Mencoba

Percobaan 1

Untuk mengetahui jawaban dari pertanyaan pertama, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program PointerSederhana;  
var    data :^byte; {Mendeklarasikan data sebagai variabel pointer}  
.....:byte; {Mendeklarasikan variabel dengan tipe byte}
```

```

.....:string; {Mendeklarasikan variabel dengan tipe string}
begin
    {Variabel nama_siswa diberi nilai awal 'Teguh'}
    .....:='Teguh';
    {Variabel tinggi_siswa diberi nilai awal 165}
    .....:=165;
    {Variabel pointer data menunjuk variabel nama_siswa}
    data:=@.....
    {Menampilkan nilai yang ditunjuk variabel pointer data}
    writeln('Data menunjuk variabel dengan nilai :',.....);
    readln;
end.

```

Hasil Percobaan Program Pointer Sederhana:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Analisis:

1. Apakah program dapat decompile dan dijalankan?
2. Apakah nilai yang ditunjuk pointer data adalah 'Teguh' bandingkan dengan output program?
3. Bandingkan jika data pada program diatas menunjuk variabel tinggi siswa (data:=@tinggi_siswa;), apa hasil output program

Percobaan 2

Untuk mengetahui jawaban dari pertanyaan kedua, salin dan lengkapilah program berikut ke dalam Free Pascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan

pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program PilihBuah;
uses crt;
var pilih_buah:^string;
    buah1,buah2:string;
begin
    buah1:='jeruk';
    buah2:='apel';
    {Variabel pilih_buah menunjuk buah1}
    pilih_buah:=.....;
    {Menampilkan nilai dari variabel yang ditunjuk pilih_buah}
    writeln('Buah yang ditunjuk pertama adalah ',.....);
    {Variabel pilih_buah menunjuk buah2}
    pilih_buah:=.....;
    {Menampilkan nilai dari variabel yang ditunjuk pilih_buah}
    writeln('Buah yang ditunjuk kedua adalah ',.....);
    readln;
end.
```

Hasil Percobaan Program PilihBuah :

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 3

Untuk mengetahui jawaban dari pertanyaan ketiga, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Remidi;
uses ..... //Memanggil unit untuk memanggil fungsi membersihkan layar
var
.....:^byte; //Mendeklarasikan variabel pointer
.....:byte; // Mendeklarasikan variabel bertipe byte
begin
..... // Fungsi untuk membersihkan tampilan layar
nilai_akhir:=50;
// Variabel pointer menunjuk lokasi variabel nilai_akhir
ulangan:=....;
// Menampilkan nilai dari variabel yang ditunjuk pointer nilai
writeln('Nilai ulangan kamu adalah ',.....);
ulangan^:=70;
//Menampilkan nilai dari variabel yang ditunjuk pointer ulangan
writeln('Nilai kamu setelah ulangan remidi menjadi ',.....);
// Menampilkan nilai dari variabel nilai_akhir
writeln('Nilai Akhir kamu menjadi ',.....);
readln;
end.
```

Hasil Percobaan Program Remidi:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 4

Untuk mengetahui jawaban dari pertanyaan keempat, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program Duapointer;
var
    pointer1,pointer2:^integer;
    data:integer;
begin
    data:=100;
    pointer1:=.....
    {pointer1 menunjuk variabel data}
    pointer2:=.....
    {pointer2 menunjuk variabel data}
    writeln('Nilai dari variabel data adalah ',.....);
    {menampilkan nilai dari variabel data}
    writeln('Nilai dari variabel yang ditunjuk pointer1
    adalah ',.....);
    {menampilkan nilai dari variabel yang ditunjuk pointer1}
    writeln('Nilai dari variabel yang ditunjuk pointer2
    adalah ',.....);
    {menampilkan nilai dari variabel yang ditunjuk pointer2}
    if pointer1=pointer2 then writeln('Pointer1 dan pointer2
    menunjuk lokasi yang sama')
```

```

{memeriksa apakah alamat dari kedua pointer sama}
else writeln('Pointer1 dan pointer2 menunjuk lokasi yang
berbeda');
pointer2^:=90;
{Mengubah nilai yang ditunjuk pointer2}
writeln('Nilai dari variabel data sekarang adalah
',.....);
{menampilkan nilai dari variabel data}
writeln('Nilai dari variabel yang ditunjuk pointer1
sekarang adalah ',pointer1^);
{menampilkan nilai dari variabel yang ditunjuk pointer1}
writeln('Nilai dari variabel yang ditunjuk pointer2
sekarang adalah ',pointer2^);
{menampilkan nilai dari variabel yang ditunjuk pointer2}
readln;
end.

```

Hasil Percobaan Program DuaPointer:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 5

Untuk mengetahui jawaban dari pertanyaan kelima, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```

program PointerSaja;
uses crt;

```

```
var
    nilai1,nilai2:^byte;
    hasil:^real;
begin
    clrscr;
    new(nilai1);
    new(nilai2);
    new(hasil);
    write('Masukan nilai pertama : ');
    readln(nilai1^);
    write('Masukan nilai kedua : ');
    readln(nilai2^);
    hasil^:=nilai1^+nilai2^;
    writeln('Hasil dari ',nilai1^,' + ',nilai2^,' = ',hasil^:0:0);
    hasil^:=nilai1^-nilai2^;
    writeln('Hasil dari ',nilai1^,' - ',nilai2^,' = ',hasil^:0:0);
    hasil^:=nilai1^*nilai2^;
    writeln('Hasil dari ',nilai1^,' x ',nilai2^,' = ',hasil^:0:0);
    hasil^:=nilai1^/nilai2^;
    writeln('Hasil dari ',nilai1^,' : ',nilai2^,' = ',hasil^:0:2);
    dispose(nilai1);
    dispose(nilai2);
    dispose(hasil);
    readln;
end.
```

Hasil Percobaan Program PointerSaja:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

4.1.2.4. Mengasosiasi/ menalar

Setelah siswa melakukan percobaan yang sesuai dengan pertanyaan yang ada, siswa diharapkan melakukan kegiatan menalar yang bisa dilakukan secara individu oleh masing-masing siswa atau bisa juga berkelompok. Beberapa hal yang bisa dijadikan arahan untuk menalar:

1. Perhatikan hasil kompilasi, apakah berhasil atau justru terjadi kesalahan.
2. Perhatikan urutan proses dari setiap program yang sudah dicoba.
3. Cobalah membuat kesimpulan dari percobaan yang telah dilakukan.

4.1.3. Rangkuman

Dari percobaan-percobaan yang telah dilakukan, bisa ambil kesimpulan bahwa:

1. Pointer adalah suatu variabel yang menyimpan atau menunjuk alamat dari suatu variabel yang lain
2. Pointer yang menunjuk tipe data tertentu, hanya dapat menyimpan lokasi dari variabel yang tertentu pula
3. Lokasi yang ditunjuk oleh suatu pointer dapat diubah
4. Pointer dapat mengubah dan memanipulasi nilai dari variabel yang ditunjuk
5. Jika ada dua pointer menunjuk lokasi yang sama, maka nilai dari variabel yang ditunjuk pointer juga sama
6. Dengan menggunakan pointer kita melakukan operasi dalam pascal, tanpa menggunakan variabel dengan tipe data tertentu seperti integer, string, array ataupun yang lain, sehingga setelah variabel terpakai dapat dihapus dari memori

4.1.4. Tugas

1. Buatlah Program untuk menggambarkan konsep Pointer: menampilkan alamat pointer dan isi pointer, dengan ketentuan sbb,

Variabel bilangan bulat bil= 20

dan variabel pointer .Vpoin

Susun laporan yang terdiri atas kode program, penjelasan program, dan output dari program.

2. Buatlah program penjumlahan dengan menggunakan pointer. Operasi harus menggunakan pointer. Dan hasilnya juga harus berupa pointer. Diketahui variabel a=5 dan b=10. Dan c adalah hasil pertambahan a+b. tetapi operasi pertambahan tidak boleh $c=a+b$; harus menggunakan pointer. Hasilnya harus dimasukkan ke dalam pointer juga.

Susun laporan yang terdiri atas kode program, penjelasan program, dan output dari program.

3. . Bila diberikan deklarasi pointer dalam Pascal :

```
var
  p,q,r : ^integer;
  x,y,z : integer;
```

4.

Tentukan statemen yang manakah di bawah ini yang legal/benar dan yang manakah yang tidak benar/illegal? Berikan alasannya masing-masing.

- | | | |
|------------|----------------|----------------|
| a) NEW(p) | f) r := NIL | k) dispose(r) |
| b) NEW(q^) | g) z := p^ | l) x := NEW(p) |
| c) NEW(x) | h) p := ^x | m) q^ := NIL |
| d) p:=r | i) dispose(y) | n) p^ := x^ |
| e) q := y | j) dispose(p^) | o) z := NIL |
5. Diberikan sepenggal program pendek sebagai berikut, lengkapi keterangan untuk setiap baris programnya dan tentukan output programnya!

```
Program coba_pointer;
var
  p : ^integer;
  { ..... }
  nilai : integer;
  { ..... }
begin
  nilai:=23;
  { ..... }

  p:=@nilai;
  { ..... }
  writeln(p^);
  { ..... }
  p^:=1992;
  { ..... }
  writeln(p^);
  { ..... }
```

```

        writeln(nilai);
        { ..... . }
        readln;
        { ..... . }
    end.

```

6. Perhatikan Program untuk menentukan suatu kata merupakan polindrom atau bukan berikut!

```

program polindrom;
uses crt;
var

    a : string;
    b:^string;
    { ..... . }
    i      : integer ;
begin
    clrscr;
    { ..... . }
    write('masukkan kata:'); readln(a);
    { ..... . }
    new(b);
    { ..... . }
    b^:=a;
    { ..... . }

        for i:=1 to length(a) do
            b^[i]:=a[length(a)+1-i];
    { ..... . }

        if a=b^ then writeln('Kata : ',a,' adalah
palindrom')
        { ..... . }
        else writeln('Kata : ',a,' bukan palindrom karena
jika dibalik menjadi : ',b^);
        { ..... . }
    dispose(b);
    { ..... . }
    readln;

```

end.

Apa yang akan terjadi jika program diberi masukan KODOK, ADA, KASUR INI RUSAK, DIDIK jelaskan jalannya program!

4.1.5. Uji Kompetensi

Pilihlah salah satu jawaban yang paling tepat!

1. Arti dari istilah pointer adalah....
 - a. Pengenal
 - b. Penunjuk
 - c. Pencari
 - d. Penanda
 - e. Penghitung
2. Definisi dari Pointer yang tepat dari pernyataan di bawah ini adalah....
 - a. Variabel yang menunjuk alamat dari suatu variabel yang lain
 - b. Variabel yang mencari alamat dari suatu variabel yang lain
 - c. Variabel yang menandai alamat dari suatu variabel yang lain
 - d. Variabel yang menghitung alamat dari suatu variabel yang lain
 - e. Variabel yang mengenali alamat dari suatu variabel yang lain
3. Salah satu Kelebihan dari variabel pointer dibanding variabel yang lain adalah kecuali....
 - a. Bersifat dinamis
 - b. Dapat dipanggil ketika dibutuhkan saja
 - c. Dapat dihancurkan/didealokasikan ketika sudah tidak terpakai
 - d. Efisien memori
 - e. Bersifat statis
4. Cara pendeklarasian pointer yang tepat adalah....
 - a. Var pointer:*integer;
 - b. Var pointer:&integer;
 - c. Var pointer:^integer;
 - d. Var pointer:=^integer;
 - e. Var pointer=@integer;
5. Untuk mengisi suatu variabel pointer dengan alamat variabel lain menggunakan simbol....
 - a. *
 - b. &
 - c. #
 - d. @

- e. ^
6. Untuk Mengakses nilai dari suatu variabel menggunakan pointer vpointer digunakan perintah.....
- a. ^vpointer
 - b. vpointer*
 - c. vpointer^
 - d. @vpointer
 - e. &pointer
7. Variabel pointer bernilai Nil adalah variabel pointer yang
- a. Tak tentu
 - b. Tak terdefinisi
 - c. Tak hingga
 - d. Menunjuk ke suatu lokasi tertentu
 - e. Tidak menunjuk ke suatu lokasi tertentu
8. Perintah addr(Var_pointer) adalah perintah untuk
- a. Menunjuk suatu variabel
 - b. Menampilkan nilai dari suatu variabel
 - c. Merubah nilai dari suatu variabel
 - d. Menampilkan Alamat dari suatu variabel
 - e. Menandai suatu variabel
9. Untuk mengalokasikan suatu pointer bisa digunakan perintah
- a. Dispose
 - b. Get
 - c. New
 - d. Release
 - e. Freemem
10. Untuk Mendealokasikan suatu pointer yang sudah tidak pakai, digunakan perintah
- f. Dispose
 - g. Get
 - h. New
 - i. Mark
 - j. Getmem
11. Perhatikan kode program berikut:

```
program Pointer1;
var
    iptr: ^integer;
    Angka: integer;
```

```

begin
    Angka := 100;
    writeln('Besar angka: ', Angka);
    iptr := @Angka;
    writeln('iptr menunjuk variabel dengan nilai: ',
    iptr^);
    readln;
end.

```

Keluaran atau output dari program tersebut adalah....

- a. Angka
 - b. 100
 - c. Error
 - d. Iptr
 - e. Integer
12. Diketahui Nama adalah variabel string yang bernilai ‘SMK’, jika sptr adalah pointer yang menunjuk Nama maka perintah sptr^[3] menghasilkan keluaran
- a. SMK
 - b. 3
 - c. K
 - d. Alamat dari Nama
 - e. Error
13. Bila diberikan deklarasi pointer dalam Pascal :

```

var
    p,q,r : ^integer;
    x,y,z : integer;

```

Manakah statemen di bawah ini yang legal/benar....

- a. New(q^)
 - b. New(x)
 - c. p:=x
 - d. p:=q
 - e. p^:=x^
14. Pada deklarasi pointer no. 13, manakah statemen di bawah ini yang tidak legal
- a. New(r)
 - b. y:=@q
 - c. q^:=x

- d. $r:=p$
e. $x:=z$
15. Jika kita memiliki 10 variabel bertipe integer, berapa memori minimal yang dibutuhkan aplikasi....
a. 10 Bytes
b. 20 Bytes
c. 30 Bytes
d. 40 Bytes
e. 50 Bytes

Jawablah Pertanyaan berikut dengan singkat dan jelas.

1. Jelaskan Pengertian dari pointer yang anda ketahui!
2. Sebut dan jelaskan kelebihan variabel pointer dari variabel yang lain!
3. Buatlah Program sederhana menggunakan pointer, untuk melakukan operasi aritmetika sederhana (jumlah, kurang, bagi dan kali), misal $a=15$ dan $b=5$, tentukan $a+b$, $a-b$, $a*b$ dan a/b
4. Buatlah Program sederhana menggunakan pointer, untuk menampilkan alamat dan isi atau nilai dari suatu variabel yang ditunjuk pointer.

4.2. Kegiatan Belajar 2. Pointer ke Array

Alokasi Waktu : 2 x 45 menit

4.2.1. Tujuan Pembelajaran

Tujuan pembelajaran pada Kegiatan Belajar 2 tentang Pointer ke Array adalah siswa dapat memahami Pembuatan pointer ke Array dan penyimpanannya pada memori

4.2.2. Aktivitas belajar siswa

4.2.2.1. Mengamati

Deklarasi Pointer ke Array

Untuk membuat pointer ke array, pertama kamu harus mendeklarasikan sebuah pointer yang menunjuk tipe data yang sama dengan tipe data dari elemen array. Sehingga deklarasinya adalah sebagai berikut:

```
Var
    nama_pointer:^tipe_data;
    nama_array:array[iAwal..iAkhir] of tipe_data;
```

Keterangan :

1. `nama_pointer` adalah pointer yang menunjuk array
2. `nama_array` adalah nama variabel array
3. `tipe_data` adalah tipe data yang digunakan pada array
4. `iAwal` adalah indeks awal dari array
5. `iAkhir` adalah indeks akhir dari array
6. Banyaknya elemen yang dimiliki array adalah (`iAkhir - iAwal`)

Contoh 1. Pendeklarasian Pointer ke Array

```
Var
    ipoin :^integer;
    nilai : array[1..10] of integer;
```

Keterangan :

1. `ipoin` adalah pointer yang menunjuk data dengan tipe integer
2. `nilai` adalah array yang memiliki elemen berjumlah 10 dengan tipe data integer

Menginisiasi Pointer ke Array

Untuk menginisiasi pointer ke Array dapat digunakan perintah sebagai berikut:

```
nama_pointer:= @nama_array[Index_awal];
```

Keterangan :

1. `nama_pointer` adalah pointer yang menunjuk array
2. `nama_array` adalah nama variabel array
3. `iAwal` adalah indeks awal dari array

Contoh 2. Inisiasi Pointer ke Array

```
ipoin := nilai[1];
```

Keterangan :

1. `ipoin` adalah pointer yang menunjuk data dengan tipe integer
2. `nilai` adalah array yang memiliki elemen berjumlah 10 dengan tipe data integer

Dengan demikian pointer `nama_pointer` telah menunjuk elemen pertama dari array `nilai`

Mengakses elemen Array dengan Pointer

Untuk mengakses elemen Array dapat digunakan perintah sebagai berikut:

```
nama_pointer^
```

Aritmatika Pointer pada Array

Array menempati slot memori berturut-turut di memori komputer. Di sinilah aritmatika pointer berguna. Jika Anda membuat pointer ke elemen pertama, penambahan satu (increment) akan membuat sebuah titik ke elemen berikutnya. Misalkan ipoin adalah pointer integer yang menunjuk alamat dari suatu variabel integer yaitu 1000, maka dengan perintah:

```
Inc (nama_pointer);
```

Atau sama dengan:

```
nama_pointer:=nama_pointer+1;
```

Sekarang pointer ipoin menunjuk lokasi 1002 karena setiap ipoin ditambahkan satu, pointer akan menunjuk alamat integer selanjutnya, dimana letaknya 2 bytes dari alamat semula. Jika ptr menunjuk suatu variabel dengan type char, dengan alamat 1000, maka operasi diatas akan menunjuk alamat 1001 karena variabel char selanjutnya berada di 1001 atau 1 bytes dari alamat semula. Demikian juga sebaliknya untuk pengurangan satu (decrement).

```
Dec (nama_pointer);
```

Atau sama dengan:

```
nama_pointer:=nama_pointer-1;
```

Contoh 3. Increment pada pointer

```
Inc (ipoin);
```

Keterangan :

Dengan perintah Inc maka sekarang pointer ipoin menunjuk elemen kedua dari variabel nilai

Contoh 4. Decrement pada pointer

```
Dec (ipoin);
```

Keterangan :

Dengan perintah Dec maka sekarang pointer ipoin kembali menunjuk elemen pertama dari variabel nilai

Selain operasi increment dan decrement, pada pointer juga berlaku operasi penambahan dan pengurangan, akan tetapi hanya untuk penambahan dan pengurangan dengan nilai integer saja. Jika suatu pointer ditambahkan 1 berarti pointer akan menunjuk alamat selanjutnya tepat yang letaknya sesuai bytes tipe data dari alamat semula, demikian juga jika ditambah dua berarti pointer akan menunjuk alamat yang berlokasi 2 kali ukuran bytes dari alamat semula. Misalnya Variabel pointer iptr menunjuk alamat 1000 dari suatu variabel yang bertipe data integer. Jika dilakukan operasi :

```
iptr:=iptr+2;
```

Maka pointer sekarang menunjuk alamat $1000+2\times 2$ bytes=1004, dengan kata lain pointer menunjuk alamat yang berjarak 4 dari alamat semula, demikian juga seterusnya.

Sama halnya dengan penambahan, Jika suatu pointer dikurangi 1 berarti pointer akan menunjuk alamat sebelumnya tepat dari alamat semula. Sehingga jika ditambah dua berarti pointer akan menunjuk alamat yang berlokasi 2 kali ukuran bytes sebelumnya dari alamat semula.

Contoh 4. Penjumlahan pada pointer

```
ipoin:= ipoin+3;
```

Keterangan :

Dengan perintah diatas maka apabila pointer awalnya menunjuk elemen pertama, sekarang pointer ipoin menunjuk elemen keempat dari variabel array nilai.

Contoh 5. Pengurangan pada pointer

```
ipoin:= ipoin-2;
```

Keterangan :

Dengan perintah diatas maka apabila pointer awalnya menunjuk elemen keempat, sekarang pointer ipoin menunjuk elemen kedua dari variabel array nilai.

4.2.2.2. Menanya

Berdasarkan kegiatan mengamati, ada beberapa hal yang bisa diperhatikan dan dibahas lebih lanjut. Beberapa pertanyaan yang berkaitan dengan Pointer ke Variabel adalah:

1. Dapatkah pointer bekerja jika Pointer menunjuk tipe data yang berbeda dari elemen suatu array.
2. Dapatkah kita menginisiasikan pointer pada elemen terakhir pada suatu array?
3. Dapatkah kita menginisiasikan pointer pada diluar dari indeks array?
4. Dapatkah kita mengubah atau memanipulasi nilai elemen pada array menggunakan pointer?

Untuk menjawab pertanyaan-pertanyaan tersebut, siswa diharapkan bisa melakukan kegiatan mencoba berikut dengan bimbingan guru.

4.2.2.3. Mencoba

Percobaan 1

Untuk mengetahui jawaban dari pertanyaan pertama yaitu Dapatkah pointer bekerja jika Pointer menunjuk tipe data yang berbeda dari elemen suatu array.. Salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```

program PointerArray1;
var
    isi:^string;
    vect:array[1..4] of byte;
    i:byte;
begin
    for i:=1 to 4 do
        begin
            write('Masukan elemen ke-',i,' : ');
            readln(.....);
            {Menyimpan masukan user ke dalam elemen ke I dari array vect}
        end;
    isi:=.....
    {Variabel isi menunjuk elemen pertama dari array vect}
    writeln(.....) ;
    {Menampilkan nilai yang ditunjuk pointer isi}
    readln;
end.

```

Hasil Percobaan Program PointerArray1:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 2

Untuk mengetahui jawaban dari pertanyaan kedua, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program PointerArray2;
uses .....
var
    isi:^ byte;
    vect:array[1..4] of byte;
    i:byte;
begin
    for i:=1 to 4 do
        begin
            write('Masukan elemen ke-',i,' : ');
            readln(.....);
            {Menyimpan masukan user ke dalam elemen ke I dari array vect}
        end;
    isi:=.....
    {Variabel isi menunjuk elemen terakhir dari array vect}
    writeln(.....);
    {Menampilkan nilai yang ditunjuk pointer isi}
    readln;
end.
```

Hasil Percobaan Program PointerArray2:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 3

Untuk mengetahui jawaban dari pertanyaan ketiga, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program PointerArray3;
uses .....
var
    isi:^ byte;
    vect:array[1..4] of byte;
    i:byte;
begin
    for i:=1 to 4 do
    begin
        write('Masukan elemen ke-',i,' : ');
        readln(.....);
        {Menyimpan masukan user ke dalam elemen ke I dari array vect}
    end;
    isi:=.....
    {Variabel isi menunjuk elemen kelima dari array vect}
    writeln(.....) ;
    {Menampilkan nilai yang ditunjuk pointer isi}
    readln;
end.
```

Hasil Percobaan Program PointerArray3:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Salin pesan asli kompilasi disini:

.....
.....
.....

Letak kesalahan (baris, kolom)	Terjadi karena

Percobaan 4

Untuk mengetahui jawaban dari pertanyaan keempat, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

Program 4.

```
program PointerArray4;
var    isi:^byte;
       vect:array[1..4] of byte;
       i:byte;
begin
  for i:=1 to 4 do begin
    write('Masukan elemen ke-',i,' : ');
    readln(.....);
    {Menyimpan masukan user ke dalam elemen ke I dari array vect}
  end;
  isi:=.....
  {Variabel isi menunjuk elemen pertama dari array vect}
  for i:=1 to 4 do begin
    isi^:=isi^+1;
    writeln('Elemen ke ',i,' = ',.....);
    {Menampilkan elemen ke I dari array vect}
    .....
    {Pointer menunjuk satu lokasi tepat didepannya}
  end;
  readln;
end.
```

Hasil Percobaan Program PointerArray4:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

4.2.2.4. Mengasosiasi

Setelah siswa melakukan percobaan yang sesuai dengan pertanyaan yang ada, siswa diharapkan melakukan kegiatan menalar yang bisa dilakukan secara individu oleh masing-masing siswa atau bisa juga berkelompok. Beberapa hal yang bisa dijadikan arahan untuk menalar:

1. Perhatikan hasil kompilasi, apakah berhasil atau justru terjadi kesalahan.
2. Perhatikan urutan proses dari setiap program yang sudah dicoba.
3. Perhatikan letak/posisi kesalahan, coba pikirkan mengapa terjadi kesalahan pada posisi tersebut.
4. Cobalah membuat kesimpulan dari percobaan yang telah dilakukan.

4.2.3. Rangkuman

1. Pointer hanya bekerja jika Pointer menunjuk tipe data yang sama dari elemen suatu array.
2. Pointer dapat diinisiasikan pada elemen berapapun pada suatu array
3. Pointer tidak dapat diinisiasikan pada elemen diluar indeks pada suatu array
4. Pointer dapat mengubah atau memanipulasi nilai elemen pada array

4.2.4. Tugas

1. Buatlah program untuk menampilkan isi dan alamat pointer larik (array)

Bilangan : Array[1..9] of integer =(5,7,8,3,4,9,1,2,0);

Susun laporan yang terdiri atas kode program, penjelasan program, dan output dari program.

2. Buatlah program menggunakan pointer untuk menjumlahkan dua vektor a dan b, dimana:

a=(2,4,1) dan b=(5,3,1)

Susun laporan yang terdiri atas kode program, penjelasan program, dan output dari program.

4.2.5. Uji Kompetensi

- Perhatikan Deklarasi pointer berikut:

```
Var
    Iptr:^integer;
    Sptr:^string;
    Nilai :array[1..10] of integer;
    Hari :array[1..7] of
        string= ('Minggu','Senin','Selasa','Rabu','Kamis','Jumat','Sabtu');
```

Variabel manakah yang merupakan pointer yang menunjuk suatu Array....

- a. Iptr
 - b. Minggu
 - c. Integer
 - d. Hari
 - e. Array
- Untuk menginisiasi pointer ke array digunakan perintah
 - a. Var_Pointer:= @Var_Array[1];
 - b. Var_Pointer = @Var_Array[1];
 - c. Var_Pointer: @Var_Array[1];
 - d. Var_Pointer:= ^Var_Array[1];
 - e. Var_Pointer= ^Var_Array[1];
 - Dari deklarasi pointer no. 1, manakah statemen di bawah yang legal....
 - a. Sptr:='Sekolah'
 - b. Iptr:=100
 - c. Iptr:=@Nilai[8]
 - d. Sptr:=@Hari[10]
 - e. Nilai:=Iptr
 - Cara mengakses isi array yang benar menggunakan pointer adalah
 - a. Var_Pointer^
 - b. ^Var_Pointer
 - c. Var_Pointer^[index]
 - d. @Var_Pointer
 - e. @Var_Pointer^
 - Perhatikan Kode program berikut:

```

Program CetakHari;
{Membuat Program dengan nama CetakHari}

Var
    Sptr:^string;
    Hari :array[1..7] of String
    =('Minggu','Senin','Selasa','Rabu','Kamis','Jumat','Sabtu');
    urut:integer;
    pHari: array[1..7] of String;

Begin

    for urut:= 1 to 7 do
        pHari[urut]:= @Hari[urut];

        Write('Masukan sebarang angka : ');readln(urut);
        if urut<8 then
            begin
                if urut>0 then
                    Write('Hari ke - ',urut,' :
',pHari[urut]^)
                else writeln('Masukan yang di izinkan antara 1
sampai 7');
            end
        else writeln('Masukan yang di izinkan antara 1
sampai 7');
        readln;
End.

```

Jika program di atas diberi masukan 2, maka akan memberi keluaran

- Selasa
 - Jumat
 - Senin
 - Kamis
 - Rabu
6. Untuk menggeser lokasi yang ditunjuk pointer satu lokasi setelahnya digunakan perintah....
- Inc(Var_Pointer)
 - Dec(Var_Pointer)
 - Increment(Var_Pointer)
 - Decrement(Var_Pointer)

- e. @Var_Pointer+1
7. Jika suatu pointer integer menunjuk alamat 1002 dari suatu variabel bertipe integer, jika dilakukan operasi aritmetika pada pointer sehingga pointer menunjuk alamat 1010. Hal ini berarti pointer menunjuk
- Lokasi ke 10 setelah alamat semula
 - Lokasi ke 5 setelah alamat semula
 - Lokasi pertama setelah alamat semula
 - Lokasi ke-5 sebelum alamat semula
 - Lokasi ke-10 sebelum alamat semula
8. Jika pointer Ipoint menunjuk elemen ke 2, pernyataan manakah yang membuat pointer menunjuk elemen ke 7 dari suatu array....
- Ipoint:=7
 - Ipoint:=Ipoint+7
 - Ipoint:=Ipoint+5
 - Ipoint:=Ipoint^+7
 - Ipoint^:=Ipoint+5
9. Untuk mengakses elemen atau isi array dengan pointer dapat digunakan perintah....
- Var_Pointer^
 - Var_Pointer^[index]
 - @Var_Pointer
 - @Var_Pointer[index]^
 - *Var_Pointer[index]^
10. Perhatikan potongan program berikut:

```
Var
    Sptr:^string;
    Hari :array[1..7] of
        string=('Minggu','Senin','Selasa','Rabu','Kamis','Jumat','Sabtu')
    ;
Begin
    Sptr:=@hari[2]
    Sptr:=Sptr+4
End.
```

Pointer Sptr menunjuk elemen ke.....

- 2
- 3
- 4
- 5
- 6

1.1 Kegiatan Belajar 3 Pemanfaatan Pointer

Alokasi Waktu : 2 x 45 menit

4.3.1. Tujuan Pembelajaran

Tujuan pembelajaran pada Kegiatan Belajar 3 tentang Pemanfaatan Pointer adalah:

1. Siswa dapat memanfaatkan pointer dalam memecahkan masalah kehidupan sehari - hari

4.3.2. Aktivitas belajar siswa

Pada kegiatan belajar kali ini akan lebih fokus pada pemanfaatan dan penggunaan pointer dalam menyelesaikan berbagai macam kasus yang ada.

4.3.2.1. Menanya

1. Apakah Pointer dapat memecahkan masalah terkait dengan palindrom?
2. Apakah Pointer dapat memecahkan masalah terkait terkait laba dan rugi suatu perusahaan
3. Apakah Pointer dapat memecahkan masalah terkait dengan jarak dari dua titik

4.3.2.2. Mencoba

Percobaan 1

Kata "palindrom" berasal dari bahasa Yunani: palin ("balik") dan dromos ("pacuan kuda"). Palindrom adalah sebuah kata, kalimat, frase, atau sekumpulan bilangan yang memiliki arti yang sama bila dibaca dari arah berbeda (dari kiri ke kanan atau dari kanan ke kiri).. Misalnya kata "tamat", "kakak", "malam", dan lain-lain. Dalam bentuk kalimat salah satu contoh yang bisa diberikan adalah "kasur ini rusak".

Untuk mengetahui jawaban dari pertanyaan pertama, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program PointerPalindrom1;
uses .....
var
    kata: string;
    ikata:^string;
    { ..... }
    i      : byte ;
begin
    .....
    { Fungsi untuk membersihkan layar}
    write('masukkan kata:'); readln(.....);
```

```

    { menyimpan masukan user ke dalam variabel
      string}
      new(.....);
      {Mengalokasikan memori untuk pointer}
      ikata^:=kata;
      {Menginisiasi nilai awal yang ditunjuk pointer}
      { ..... }
      for i:=1 to length(kata) do
        ikata^[i]:= kata [length(kata)+1-i];
      { ..... }
      if kata=ikata^ then writeln('Kata : ', kata,
      , ' adalah palindrom')
      { ..... }
      else writeln('Kata : ', kata,' bukan palindrom
      karena jika dibalik menjadi : ', ikata^);
      { ..... }
      dispose(ikata);
      { ..... }
      readln;
    end.
  
```

Hasil Percobaan Program PointerPalindrom1:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

```

program PointerPalindrom2;

uses crt;

const max =25;
  
```

```

var arr: array [1..max] of String;
    n,i: byte;
    spoin:^String;
    ispalindrom:boolean;
begin
    clrscr;
    write('Masukan banyaknya elemen palindrom : ');
    readln(n);
    for i:=1 to n do begin
        write('Masukan elemen ke-',i,' : ');
        readln(Arr[i]);
    end;
    spoin:=@arr[n];
    ispalindrom:=true;
    for i := 1 to n do begin
        if arr[i]=spoin^ then begin
            ispalindrom:=(ispalindrom and true);
            dec(spoin);
        end else begin
            ispalindrom:=(ispalindrom and false);
        end;
    end;
    if ispalindrom then
        writeln('Palindrom')
    else
        writeln('Bukan palindrom');
    readln;
end.

```

Hasil Percobaan Program PointerPalindrom2:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 2

Jika seorang pengusaha memiliki 10 buah perusahaan, dimana perusahaan pertama memperoleh laba 5.000.000, perusahaan kedua memperoleh laba 1.000.000, perusahaan ketiga memperoleh laba 4.000.000, perusahaan keempat memperoleh laba 6.000.000, perusahaan kelima memperoleh laba 4.000.000, perusahaan keenam memperoleh laba 5.000.000, perusahaan ketujuh memperoleh laba 10.000.000, perusahaan kedelapan memperoleh laba 7.000.000, perusahaan kesembilan memperoleh laba 3.000.000, perusahaan kesepuluh memperoleh laba 8.000.000, berapakah laba total yang diperoleh pengusaha tersebut? Berapakah rata-rata laba pengusaha tersebut? Apakah ada perusahaan yang mempunyai laba sama dengan rata-rata, lebih dari rata-rata, dan lebih dari rata-rata?

Untuk mengetahui jawaban dari pertanyaan kedua, salin dan lengkapilah program berikut ke dalam Freepascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program LabaPrshn;
uses crt;
const max =20;
var arr: array [1..max] of integer;
    n,i: byte;
    ipoin, temp:^integer;
begin
    clrscr;
    write('Masukan banyaknya unit perusahaan : ');
    readln(n);
    for i:=1 to n do begin
        write('Masukan Laba Unit ke-',i,' : ');
        readln(arr[i]);
    end;
    ipoin:=@arr[1];
    New(temp);
    temp^:=0;
    for i := 1 to n do begin
        temp^:=temp^+ipoin^;
        {Menghitung Jumlahan setiap elemen array}
        inc(ipoin);
        {Pointer sekarang menunjuk lokasi tepat
        setelahnya dari lokasi semula}
```

```
end;

Writeln('Total Laba dari setiap unit : ', temp^);
Writeln('Rata rata laba : ', temp^/n:2:2);
ipoin:=@arr[1];
for i :=1 to n do
if ipoin^>temp^/n then begin
    writeln('Unit ke ',i,' di atas rata laba unit');
    inc(ipoin);
end else
if ipoin^=temp^/n then begin
    writeln('Unit ke ',i,' sama dengan rata laba
unit');
    inc(ipoin);
end else begin
    writeln('Unit ke ',i,' di bawah rata laba
unit');
    inc(ipoin);
end;
readln;
dispose(temp);
end.
```

Hasil Percobaan Program LabaPrshn:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

Percobaan 3

Letak titik pada bidang XoY ditulis dalam bentuk pasangan bilangan (x,y) dimana x disebut absis dan y disebut ordinat. Format yang dipakai selalu (x,y) dengan urutan yang tidak dapat dibalik. Pada bidang koordinat tersebut, titik A terletak pada koordinat (1,0), ditulis A(1,0), titik B terletak pada koordinat (2,4), ditulis B(2,4), titik C terletak pada koordinat (5,7), ditulis dengan C(5,7), dan titik D terletak pada koordinat (6,4) ditulis D(6,4).

Untuk mencari jarak antara dua titik dapat digunakan rumus

$$\text{Jarak} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Keterangan:

Jarak = jarak dua titik

x_1 = nilai x pada titik pertama

x_2 = nilai x pada titik kedua

y_1 = nilai y pada titik pertama

y_2 = nilai y pada titik kedua

Untuk mengetahui jawaban dari pertanyaan Apakah Pointer dapat memecahkan masalah terkait dengan jarak antara dua titik. Salin dan lengkapilah program berikut ke dalam FreePascal, kemudian lakukan kompilasi (Compile / Alt+F9). Jika kompilasi menghasilkan pesan kesalahan maka perhatikan kesalahan yang terjadi. Jika kompilasi berhasil lanjutkan dengan menjalankan program (Ctrl+F9) dan amati hasilnya.

```
program PointerJarak;
uses crt;
var x1,y1,x2,y2:^integer;
    jarak:^real;
begin
    clrscr;
    New(x1);
    New(y1);
    New(x2);
    New(y2);
    New(temp);
    write('Masukan nilai x1 : ');
    readln(x1^);
    write('Masukan nilai y1 : ');
    readln(y1^);
```

```

        write('Masukan nilai x2 : ');
        readln(x2^);
        write('Masukan nilai y2 : ');
        readln(y2^);
        writeln(x1^,y1^,x2^,y2^);
        jarak^:=sqrt((x1^-x2^)*(x1^-x2^)+(y1^-y2^)*(y1^-y2^));
        Writeln('Jarak dari Dua titik tersebut : ', jarak^:0:2);
        readln;
        dispose(x1);
        dispose(y1);
        dispose(x2);
        dispose(y2);
        dispose(jarak);
end.

```

Hasil Percobaan Program PointerJarak:

Hasil Kompilasi (beri tanda silang pada bagian yang sesuai)

- Berhasil, tanpa kesalahan
- Tidak berhasil, ada kesalahan

Output Program:

4.3.2.4. Mengasosiasi/ menalar

Setelah siswa melakukan percobaan yang sesuai dengan pertanyaan yang ada, siswa diharapkan melakukan kegiatan menalar yang bisa dilakukan secara individu oleh masing-masing siswa atau bisa juga berkelompok. Beberapa hal yang bisa dijadikan arahan untuk menalar:

1. Perhatikan hasil kompilasi, apakah berhasil atau justru terjadi kesalahan.
2. Perhatikan urutan proses dari setiap program yang sudah dicoba.
3. Perhatikan letak/posisi kesalahan, coba pikirkan mengapa terjadi kesalahan pada posisi tersebut.
4. Cobalah membuat kesimpulan dari percobaan yang telah dilakukan.

4.3.3. Rangkuman

1. Pointer dapat menyelesaikan dengan baik permasalahan mengenai polindrom, baik menggunakan pointer ke string ataupun pointer ke array
2. Pointer dapat menyelesaikan dengan baik masalah terkait laba dan rugi suatu perusahaan.
3. Pointer juga dapat menyelesaikan masalah dengan baik mengenai jarak antara dua titik.

4.3.4. Tugas/Portofolio

1. Buatlah program persamaan linier yang menerima inputan x dan menghasilkan nilai y dengan persamaan $y = 3x + 4$ dimana x dan y adalah variabel pointer dan dealokasikan variabel x , dan y setelah selesai digunakan.

Susun laporan yang terdiri atas kode program, penjelasan program, dan output dari program.

2. Buatlah program menggunakan pointer untuk menyelesaikan masalah berikut:

Diketahui kota P terletak pada $P(2,7)$ dan kota Q terletak pada $Q(10,3)$ akan dibuat jalan tol yang menghubungkan antara kota P dan kota Q, jika biaya per satuan jarak jalan tol adalah Rp 100.000.000,00 maka berapakah dana minimal yang harus disiapkan?

Susun laporan yang terdiri atas kode program, penjelasan program, dan output dari program.

3. Buatlah program menggunakan pointer untuk menghitung frekuensi huruf "a" yang terdapat pada kalimat : "ada apa dengan SMK".

Susun laporan yang terdiri atas kode program, penjelasan program, dan output dari program.

4. Buatlah program menggunakan pointer untuk mengubah derajat suhu dari celsius ke Fahrenheit, Reamur dan Kelfin.

Susun laporan yang terdiri atas kode program, penjelasan program, dan output dari program.

5. Buatlah program menggunakan pointer untuk menyelesaikan masalah anagram (hurufnya sama tetapi diacak). Program yang menerima masukan berupa dua buah kata, kemudian memeriksa apakah kedua kata tadi termasuk anagram atau tidak.

Contoh:

"SEBAB" dan "BEBAS" adalah anagram

"KAPAS" dan "PASA" adalah anagram

Susun laporan yang terdiri atas kode program, penjelasan program, dan output dari program.

DAFTAR PUSTAKA

- Cantu, Marco. 2008. *Essential Pascal 4th Edition*. Piacenza Italy: Marco Cantu.
- Kadir, Abdul.1997.*Pemrograman Pascal*.Yogyakarta:Penerbit Andi
- Santosa, P. Insap. 1997. *Struktur Data Menggunakan Turbo Pascal 6.0*. Yogyakarta: Andi Offset.
- Suprapto. 2008. *Bahasa Pemrograman Untuk Sekolah Menengah Kejuruan*. Jakarta: DitPSMK Depdiknas
- Van Canneyt, M. 2013. *Free Pascal Programmer's Guide, Programmer's Guide for Free Pascal, Version 2.6.2.* (pdf, freepascal.org)
- Van Canneyt, M. 2013. *Free Pascal Reference Guide, Reference Guide for Free Pascal, Version 2.6.2.* (pdf, freepascal.org)
- Van Canneyt, M. 2013. *Free Pascal User's Guide, User's Guide for Free Pascal, Version 2.6.2.* (pdf, freepascal.org)
- Zarlis, Muhammad dan Handrizal.2008. *Algoritma & Pemrograman: Teori dan Praktik dalam Pascal*. Medan: USU Press
- _____. *Pascal Tutorial*. (pdf, tutorialspoint.com)

Pemrograman Dasar

SMK/MAK Kelas XI

Semester 1

Buku Pemrograman Dasar ini ditujukan untuk SMK/MAK bidang keahlian Teknologi Informasi dan Komunikasi kelas XI semester 1.

Buku ini dirancang untuk menjadi buku pegangan siswa agar bisa digunakan untuk membantu pembelajaran yang menerapkan Kurikulum 2013 (K13).

Bahasa pemrograman Pascal dipilih untuk digunakan dalam buku ini karena kesederhaannya tetapi sangat ampuh untuk digunakan untuk mengenal dan menguasai dasar-dasar pemrograman. Compiler yang digunakan dalam penyusunan buku adalah Free Pascal dan menggunakan Integrated Development Environment (IDE) non visual.

Materi:

- Bab I. Aritmatika dan Logika
- Bab II. Array
- Bab III. Operasi String dan Konversi Data
- Bab IV. Pointer

Buku ini menyajikan beberapa pertanyaan pancingan dan disediakan kegiatan percobaan yang bisa dilakukan siswa, sehingga siswa diajak untuk melakukan percobaan langsung untuk mengetahui jawaban dari setiap pertanyaan yang ada