

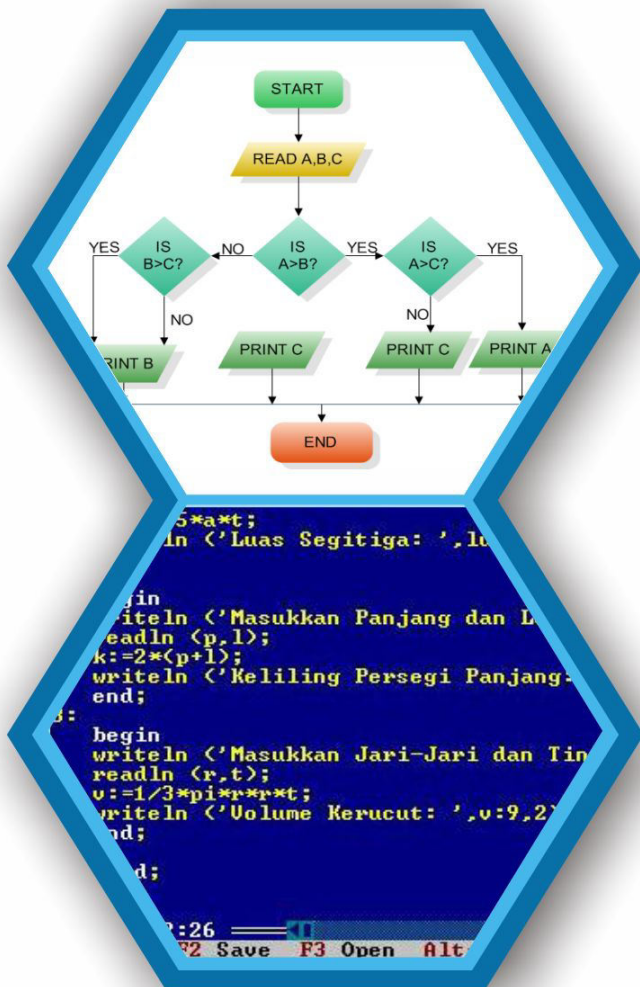


Kementerian Pendidikan dan Kebudayaan
Republik Indonesia
2014



Pemrograman Dasar

Untuk SMK/MAK



Kelas X
Semester 2

Mahasiswa Ke-1 = Edi Yulianto
Nilai Mahasiswa Ke-1 = 85
Mahasiswa Ke-2 = Tarkiman
Nilai Mahasiswa Ke-2 = 87
Mahasiswa Ke-3 = Abdul Kohar
Nilai Mahasiswa Ke-3 = 81
Mahasiswa Ke-4 = Chandra Kusuma
Nilai Mahasiswa Ke-4 = 75
Mahasiswa Ke-5 = Dadan Setiawan
Nilai Mahasiswa Ke-5 = 80

Dwi Maryono
Dhidhi Pambudi

Hak Cipta pada Kementerian Pendidikan dan Kebudayaan
Dilindungi Undang-Undang

Milik Negara
Tidak Diperdagangkan

Kontributor

Penyunting Materi : Rosihan Ari Yuana
Penyunting Bahasa : Rosihan Ari Yuana
Ilustrasi Sampul : Adhi Ginanjar
Hilda Debora
Penyeliapenerbitan :

Kotak Katalog dalam Terbitan (KDT)

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 (*passive 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 Dasar 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, mengasosiasikan, 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 Dasar” Kelas X Semester 2 Sekolah Menengah Kejuruan (SMK).

Jakarta, 12 Desember 2014

Menteri Pendidikan dan Kebudayaan

DAFTAR ISI

Kata Pengantar	iii
Daftar Isi	v
Pendahuluan	1
Bab 1. Pengenalan Bahasa Pemrograman	5
1.1 Kegiatan Belajar 1 Pengenalan Bahasa Pemrograman	5
1.2 Kegiatan Belajar 2 Struktur Bahasa Pemrograman Pascal	21
1.3 Kegiatan Belajar 3 Standar Input Output.....	33
1.4 Kegiatan Belajar 4Kompilasi Program dan Perbaikan Kesalahan	45
Bab 2. Variabel, Tipe Data dan Operator	55
2.1 Kegiatan Belajar 1 Variabel, Tipe Data dan Konstanta	55
2.2 Kegiatan Belajar 2 Operator dan Ekspresi	74
Bab 3. Struktur Percabangan	87
3.1 Kegiatan Belajar 1 Percabagnan 1 Kondisi.....	87
3.2 Kegiatan Belajar 2 Percabangan Dengan 2 Kondisi	97
3.3 Kegiatan Belajar 3 Percabangan Lebih Dari 2 Kondisi.....	107
3.4 Kegiatan Belajar 4 Percabangan Bersarang	118
Bab 4. Struktur Perulangan	125
4.1 Kegiatan Belajar 1 Perulangan Dengan Kondisi di Awal	125
4.2 Kegiatan Belajar 2 Perulangan Dengan Kondisi di Akhir.....	139
4.3 Kegiatan Belajar 3 Perulangan Dengan Kondisi Dinputkan User	147
4.4 Kegiatan Belajar 4 Perulangan Dengan Pernyataan Break & Continue	154
Bab 5. Pengembangan Algoritma Aplikasi.....	161
5.1 Kegiatan Belajar 1 Pengembangan Aplikasi dan Analisis Pemecahan Masalah	161
5.2 Kegiatan Belajar 2 <i>Error Handling</i>	169
5.3 Kegiatan Belajar 3 <i>Debugging</i>	178
5.4 Kegiatan Belajar 4 Study Kasus Proyek Pengembangan Aplikasi	187
DAFTAR PUSTAKA	190

PENDAHULUAN

Deskripsi:

Pemrograman Dasar adalah pemrograman yang memberikan dasar-dasar logika dimana sintak-sintak yang diberikan bersifat universal dan lebih mengedepankan pembentukan pola pikir siswa tentang bagaimana membuat sebuah program yang efektif dan efisien.

Materi Pelajaran Pemrograman Dasar dibagi menjadi 4 bagian, di mana satu semester untuk tiap bagian. Pada bagian kedua di Kelas X Semester 2 ini, siswa dikenalkan pada bahasa pemrograman untuk membuat program atau aplikasi. Bahasa pemrograman yang digunakan adalah Bahasa Pemrograman Pascal dengan *compiler* Free Pascal.

Pengenalan Bahasa Pascal dalam pembahasan ini meliputi pengenalan struktur Bahasa Pascal, *compiler* Free Pascal, konsep variabel, tipe data, dan operator, struktur percabangan dan perulangan. Dengan materi dasar tersebut diharapkan siswa mampu membuat program sederhana untuk menyelesaikan masalah sehari-hari.

Prasyarat:

- **Dasar-dasar Algoritma**

Petunjuk Penggunaan:

1. Bacalah modul di setiap Kegiatan Belajar dengan seksama, ikuti semua proses mulai mengamati, menanya, mencoba, mengasosiasikan, dan mengkomunikasikan
2. Kerjakan Lembar Kerja Siswa dengan berkelompok dan berdiskusi secara aktif
3. Jika ada yang kurang jelas tanyakan kepada instruktur/guru pembimbing Anda
4. Kerjakan tugas pada setiap Kegiatan Belajar dan laporkan hasil tugas kalian kepada Guru
5. Kerjakan tes Uji Kompetensi untuk mengukur pencapaian hasil belajar

Tujuan Akhir :

Diharapkan siswa mampu

1. Mengetahui bahasa pemrograman
2. Mengetahui tools/framework pengembangan program
3. Melakukan instalasi tools bahasa pemrograman
4. Memahami Struktur bahasa pemrograman
5. Memahami Standar input dan output dalam bahasa pemrograman
6. Melakukan kompilasi dan eksekusi program
7. Memperbaiki Kesalahan

8. Memahami Tipe Data, Variabel, Operator dan Ekspresi
9. Memahami Struktur Kontrol Percabangan dengan 1 kondisi, 2 kondisi, lebih dari 2 kondisi, dan percabangan bersarang
10. Memahami Struktur Kontrol Perulangan dengan kondisi diawal, di akhir, dan diinputkan user
11. Memahami perulangan dengan pernyataan continue dan break
12. Memahami langkah-langkah pengembangan aplikasi
13. Memahami langkah-langkah analisis pemecahan permasalahan
14. Memahami error handling dalam pascal dan menerapkannya pada masalah pemrograman.
15. Memahami proses debugging dan menerapkannya pada masalah pemrograman.
16. Membuat program aplikasi dalam kehidupan sehari-hari

Kompetensi Inti dan Kompetensi Dasar

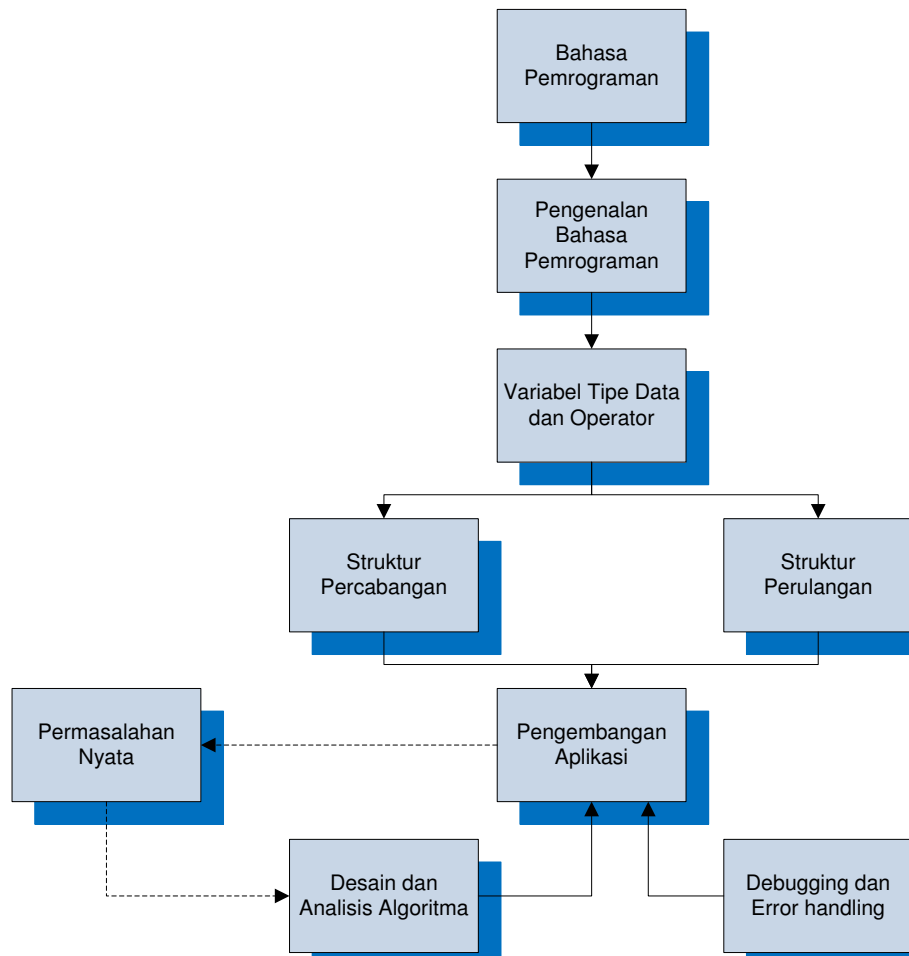
Kompetensi Inti

- KI 1 : Menghayati dan mengamalkan ajaran agama yang dianutnya.
- KI 2 : Menghayati dan Mengamalkan perilaku jujur, disiplin, tanggungjawab, peduli (gotong royong, kerjasama, toleran, damai), santun, responsif dan proaktif 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.
- KI 3 : Memahami, menerapkan dan menganalisis pengetahuan faktual, konseptual, dan procedural berdasarkan rasa inginnya tentang ilmu pengetahuan, teknologi, seni, budaya, dan humaniora dalam wawasan kemasyarakatan, kebangsaan, kenegaraan, dan peradaban terkait penyebab fenomena dan kejadian dalam bidang kerja yang spesifik untuk memecahkan masalah.
- KI 4 : Mengolah, menalar, dan menyaji dalam ranah konkret dan ranah abstrak terkait dengan pengembangan dari yang dipelajarinya di sekolah secara mandiri, 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. Mendiskripsikan kebesaran Tuhan yang menciptakan Pelbagai sumber energi di alam.
- 1.3. Mengamalkan nilai-nilai keimanan sesuai dengan ajaran agamanya dalam kehidupan sehari-hari
- 2.1. 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
- 2.2. Menghargai kerja individu dan kelompokdalam aktivitas sehari-hari sebagai wujudimplementasi melaksanakan percobaandan melaporkan hasil percobaan
- 3.4. Menerapkan bahasa pemrograman
- 4.4. Mengolah algoritma ke dalam bentuk kode program komputer
- 3.5. Menerapkan penggunaan tipe data, variabel, konstanta, operator dan ekspresi
- 4.5. Mengolah data menggunakan konsep tipe data, variabel, konstanta, operator dan ekspresi.
- 3.6. Menerapkan struktur kontrol percabangan dalam bahasa pemrograman
- 4.6. Memecahkan masalah menggunakan struktur kontrol percabangan
- 3.7. Menerapkan struktur kontrol perulangan dalam bahasa pemrograman.
- 4.7. Memecahkan masalah menggunakan struktur kontrol perulangan.
- 3.8. Menerapkan keseluruhan konsep algoritma dalam penyelesaian masalah kompleks
- 4.8. Menganalisa kesalahan dalam program

Peta Konsep

BAB 1

PENGENALAN BAHASA PEMROGRAMAN PASCAL

Saat kalian belanja di swalayan ataupun super market pernahkah kalian memperhatikan bagaimana kasir bekerja? Apakah dari banyaknya belanjaan pelanggan, sang kasir menghitung manual satu persatu belanjaan mereka? Iya benar, tentu saja mereka tidak menghitung secara manual, karena bisa dibayangkan antrian yang terjadi mungkin bisa sangat panjang. Hal ini tentu merugikan pelanggan dan supermarket itu sendiri. Pelanggan merasa jenuh, dan banyak waktu yang terbuang sia-sia untuk mengantri, sehingga kepuasan pelanggan menurun. Tentu ini juga akan berdampak terhadap swalayan itu sendiri. Untuk itu mereka menggunakan aplikasi atau software khusus yang digunakan untuk kasir. Dengan software ini, kasir dapat memeriksa barang belanjaan dengan cepat (menggunakan *barcode reader*), sehingga dapat menentukan harga setiap barang yang dibeli dan menjumlahkan semuanya dengan sangat mudah dan cepat.

Demikian juga ketika kita bertransaksi di bank, membayar pajak listrik, membayar pajak motor, melakukan pemeriksaan di rumah sakit dan masih banyak lagi yang menggunakan aplikasi atau software khusus untuk melayani pengguna/pelanggannya sehingga menjadi lebih cepat dan praktis. Inilah mengapa kehidupan modern tidak dapat dipisahkan dengan perkembangan teknologi informasi, di mana salah satu bentuknya adalah penerapan aplikasi atau software.

Salah satu hal yang harus disyukuri bahwa Tuhan menciptakan manusia, membekali dengan kecerdasan sehingga dengan kehendakNya manusia bisa membuat sebuah aplikasi yang amat membantu pekerjaan manusia termasuk membantu kasir dalam bekerja. Pernahkah terlintas dipikiran kalian bagaimana cara membuat aplikasi perhitungan yang bermanfaat itu? Bagaimana aplikasi itu bisa menghitung dengan cepat? Bagaimana memerintahnya mengerjakan menghitung? Nah untuk memahami bagaimana suatu aplikasi dikembangkan dan dibuat, kalian harus mengerti dulu apa itu bahasa pemrograman.

1.1. KEGIATAN BELAJAR 1 PENGENALAN BAHASA PEMROGRAMAN

1.1.1 Tujuan Pembelajaran

1. Siswa memahami pengertian software bahasa pemrograman, konsep dan manfaatnya dalam kehidupan nyata
2. Menginstall salah satu bahasa pemrograman, dalam hal ini bahasa pemrograman Pascal

1.1.2 Aktivitas Kegiatan Belajar

1.1.2.1. Mengamati

Kasus 1

Marwa Berlatih Membuat Kue

Marwa ingin sekali bisa membuat kue tapi dia tidak bisa membuatnya sendiri. Akhirnya dia mendekati ibu dan meminta ibu mengajarnya.

Marwa : “Ibu, bagaimana cara membuat kue? Aku mau belajar membuatnya”

Ibu : “Oke, pertama kamu harus menyiapkan semua bahan yang dibutuhkan ada telur, tepung terigu mentega, coklat, dan pengembang, sana ambil dulu dikulkas”

Marwa : “Siap.... Ini bu semua sudah komplit”

Ibu : “Bagus, sekarang kamu kocok telurnya, setelah itu masukan tepung terigu mentega, coklat dan pengembangnya hingga tercampur rata”

Marwa : “Begini ya bu?”

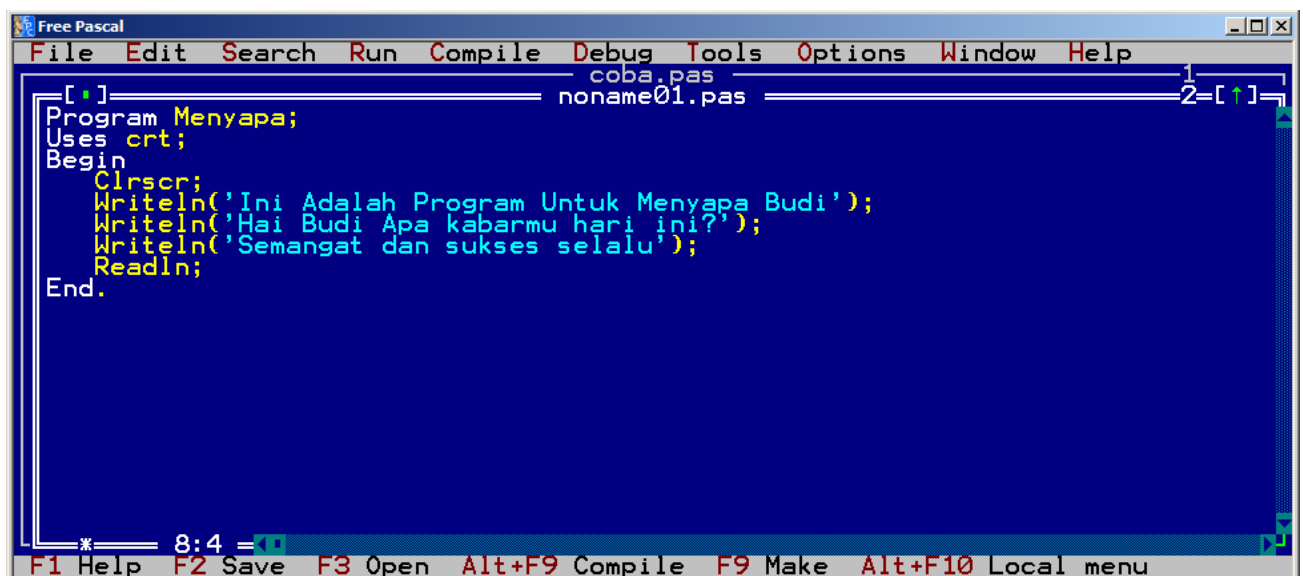
Ibu : “Ya benar, kalau sudah tuang ke cetakan dan masukan ke dalam oven sampai mengembang, kalo sudah segera dikeluarkan biar tidak gosong”

Marwapun mengikuti semua instruksi yang dianjurkan ibu. Dan akhirnya kue siap Dihidangkan.

Kasus 2

Budi Membuat Program Komputer Yang Menyapa

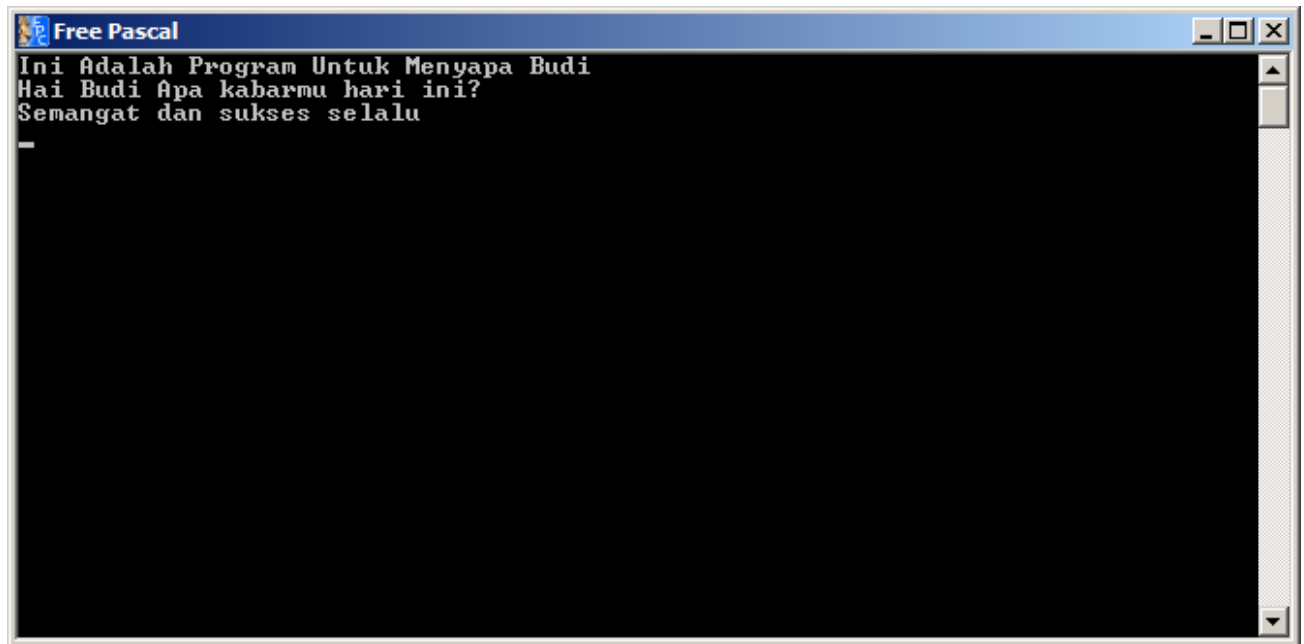
Budi ingin membuat aplikasi menggunakan komputer, maka diapun mulai meminta komputer melakukan setiap instruksi darinya. Untuk itu dia menuliskan kode dalam suatu bahasa pemrograman untuk membuat program tersebut.

The image shows a screenshot of the Free Pascal IDE. The title bar reads 'Free Pascal'. The menu bar includes 'File', 'Edit', 'Search', 'Run', 'Compile', 'Debug', 'Tools', 'Options', 'Window', and 'Help'. The toolbar shows icons for opening, saving, and running files. The main editor window displays a Pascal program named 'coba.pas' with the following code:

```
[.]
Program Menyapa;
Uses crt;
Begin
  Clrscr;
  Writeln('Ini Adalah Program Untuk Menyapa Budi');
  Writeln('Hai Budi Apa kabarmu hari ini?');
  Writeln('Semangat dan sukses selalu');
  Readln;
End.
```

The status bar at the bottom shows the cursor position '8:4' and a series of keyboard shortcuts: 'F1 Help', 'F2 Save', 'F3 Open', 'Alt+F9 Compile', 'F9 Make', 'Alt+F10 Local menu'.

Setelah dijalankan maka muncul



1.1.2.2. Menanya

Setelah melakukan pengamatan, coba buatlah pertanyaan seperti contoh berikut.

1. Apa yang dilakukan Marwa agar dia bisa membuat kue?
2. Apa yang dilakukan Budi agar komputer bisa menyapanya?
3. Pada kasus 1, siapa yang memberi intruksi dan kepada siapa? Sedangkan pada kasus 2, siapa ke siapa?
4. Adakah perbedaan instruksi yang diberikan ibu dengan instruksi yang diberikan budi?
5. Apa itu program komputer?
6. Kode apakah yang dituliskan Budi?

1.1.2.3. Mencari Informasi/Mencoba

Dari 2 contoh diatas kita temui bahwa Marwa dan Budi sama-sama menginginkan sesuatu. Marwaingin bisa membuat kue, kemudian apa yang dia lakukan? Ya, Marwa berbicara kepada ibu untuk membantunya membuat kue.Dan ibu menyampaikan instruksi-instruksi dengan jelas agar Marwa mengerti.

Begitu pula Budi yang menginginkan komputer bekerja membuatkan Budi sebuah program agar komputer bisa menyapanya, maka apa yang Budi lakukan? Untuk mengutarakan keinginannya maka dia berkomunikasi dan memberi instuksi kepada komputer melalui suatu bahasa tertentu, yang disebut sebagai **Bahasa Pemrograman**. Dia memberikan instruksi:

Program Menyapa -> mengidentifikasi judul program

Clrscr -> instruksi membersihkan layar

BEGIN -> Instruksi untuk mulai membuat

Writeln-> Instruksi untuk menuliskan

Readln-> Instruksi untuk membaca dan menampilkan

END -> Instruksi mengakhiri program.

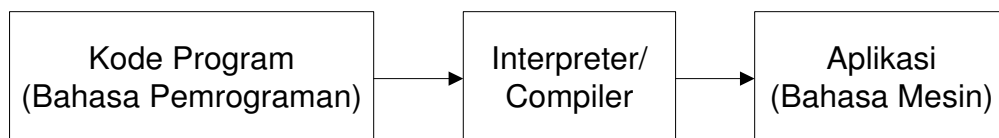
Jadi, untuk membuat sebuah program bisa berjalan sesuai runtutan kerja atau algoritma sesuai yang kita kehendaki maka kita harus menggunakan bahasa atau suatu intruksi standar yang bisa dimengerti komputer, bahasa itulah yang dikenal dengan **Bahasa Pemrograman**.

Program dan Bahasa Pemrograman

Kalian tentu tidak asing lagi dengan program komputer, kan? Setiap hari tentu kalian menggunakan program komputer atau disingkat program saja. Lalu apa definisi program? Bagaimana cara kerja dan pembuatannya?

Menurut definisi, **program komputer** adalah kumpulan instruksi yang digunakan untuk mengatur komputer agar melakukan suatu tindakan tertentu. Jadi komputer tidak akan bisa apa-apa jika tidak ada program di dalamnya. Sebuah komputer atau laptop secanggih apa pun spesifikasinya tanpa program sistem operasi maka tidak ada gunanya. Demikian juga untuk lebih optimal kemanfaatannya perlu dipasang berbagai program seperti pengolah kata, pengolah angka, pengolah grafis, program video, musik hingga game. Semua program tersebut pada dasarnya memberikan instruksi kepada komputer sedemikian rupa sehingga bisa melakukan berbagai macam hal sesuai dengan fungsi program tersebut. Orang yang mengembangkan program disebut dengan **programmer**.

Komputer bekerja dengan operasi-operasi biner, yaitu mengolah data-data berupa barisan biner 0 dan 1 sehingga tidak mungkin ketika seorang programmer memerintahkan untuk menulis pesan langsung dimengerti oleh komputer. Oleh karena itu harus ada suatu bahasa yang penghubung agar komputer mengerti instruksi atau perintah yang diberikan. Bahasa penghubung inilah yang disebut dengan **bahasa pemrograman**. **Bahasa Pemrograman** adalah suatu instruksi standar untuk memerintah komputer. Untuk menjadi sebuah program maka kode-kode yang sudah dituliskan dalam bahasa pemrograman diubah ke dalam bahasa mesin yang dimengerti oleh komputer dengan menggunakan **interpreter** atau **compiler**. Proses perubahan ini disebut dengan interpretasi atau kompilasi. Hasil kompilasi inilah yang menghasilkan program komputer. Penjelasan lebih detail mengenai keduanya akan dibahas pada bagian akhir Bab ini.



Menurut kedekatannya dengan mesin komputer, bahasa pemrograman di bagi menjadi beberapa tingkatan diantaranya:

1. Bahasa Mesin

Bahasa mesin merupakan rangkaian instruksi yang terdiri dari kode biner, contohnya 0001110011. Bahasa ini menempati tingkatan yang paling rendah karena bahasa ini bisa langsung dieksekusi oleh processor, sehingga bahasa mesin sering kali digunakan untuk kompilasi oleh compiler pada bahasa pemrograman lain, yang lebih tinggi tingkatnya.

2. Bahasa Tingkat Rendah

Bahasa tingkat rendah merupakan rangkaian instruksi menggunakan kode-kode singkat (kode mnemonic) contohnya bahasa Assembly.

3. Bahasa Tingkat Menengah

Bahasa tingkat menengah merupakan rangkaian instruksi gabungan antara bahasa manusia dengan kode-kode simbolik, contohnya {,},?,<,>&&,\$,dsb. Yang termasuk bahasa menengah yakni bahasa C.

4. Bahasa Tingkat Tinggi

Bahasa tingkat tinggi merupakan rangkaian instruksi yang menggunakan unsur kata-kata manusia sehingga lebih mudah untuk dimengerti contohnya begin, end, if, for, while, and, or dsb. Untuk bahasa ini diperlukan *compiler* untuk mengompilasi ke dalam bahasa mesin sehingga dimengerti komputer. Yang termasuk kategori ini adalah Pascal dan Basic

Mengenal Bahasa Pascal

Dalam buku Pemrograman Dasar ini, digunakan bahasa Pemrograman Pascal. Mengapa Pascal? Seperti yang dijelaskan sebelumnya bahasa pemrograman Pascal, selanjutnya disingkat bahasa Pascal, adalah salah satu bahasa tingkat tinggi, yang menggunakan perintah-perintah sebagaimana bahasa sehari-hari. Dalam Bahasa Pascal digunakan perintah-perintah seperti **read**, **write**, pernyataan **if then**, **while do**, **repeat until**, **begin**, **end**, dan sebagainya yang dimengerti oleh kebanyakan orang, sehingga menjadi bahasa Pascal mudah dipelajari, khususnya untuk mengenalkan pemrograman, melatih kemampuan algoritma dan pemecahan masalah.



Gambar 1.1 Blaise Pascal

sumber : <http://www.iitten.com/wp-content/uploads/2014/06/blaise-pascal.jpg>

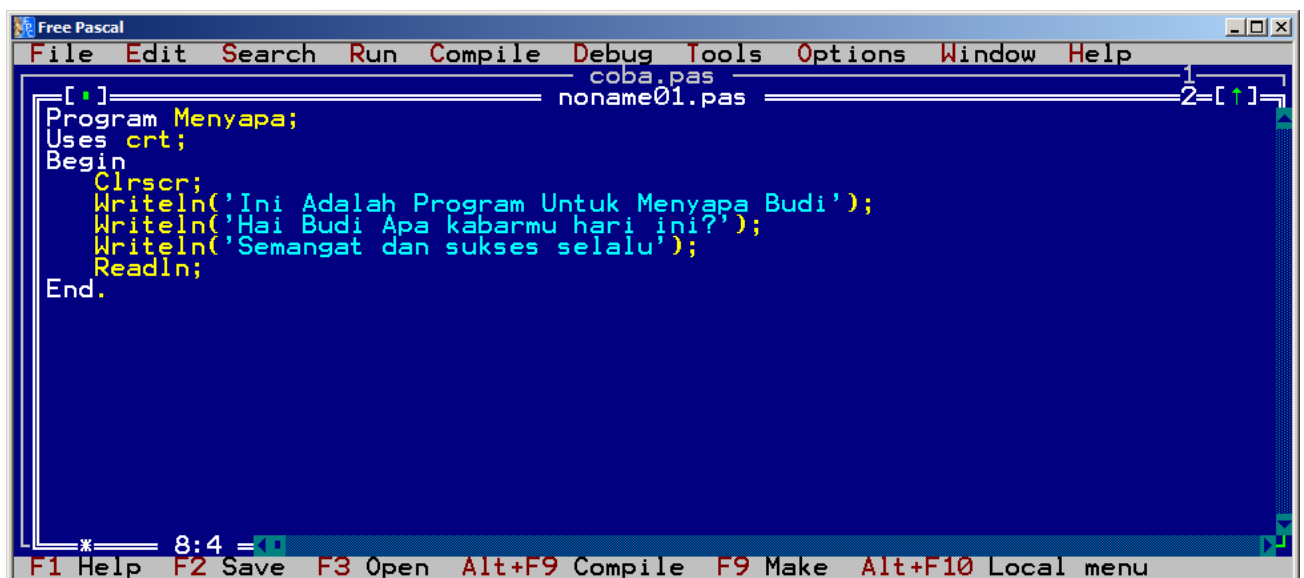
Bahasa Pascal pertama kali dirancang oleh Profesor “Niklaus Wirth” dari Technical University di Zurich, Switzerland pada tahun 1971. Nama Pascal sendiri terinspirasi nama seorang matematikawan Perancis, Blaise Pascal yang pertama kali menciptakan mesin penghitung. Bahasa Pascal kala itu dikembangkan untuk memudahkan para mahasiswa dalam mempelajari pemrograman secara sistematis.

Mengenal Tools/Framework Pengembangan Program

Seperti yang telah disampaikan sebelumnya bahwa Pascal termasuk bahasa pemrograman tingkat tinggi sehingga dibutuhkan sebuah compiler atau tools pengembang yang berfungsi menterjemahkan bahasa kedalam bahasa yang dimengerti komputer.

Beberapa *compiler* Pascal diantaranya GNU Pascal, Free Pascal, Dev Pascal dan Turbo Pascal, dan Delphi. Dari beberapa compiler tersebut, untuk pembelajaran buku ini menggunakan **FreePascal**. FreePascal memiliki keunggulan yakni cocok digunakan untuk berbagai platform Sistem Operasi diantaranya: DOS, Linux, Windows, FreeBSD, PalmOs dan masih banyak yang lain. Tools ini gratis (*freeware*) dan dapat didownload di www.freepascal.org. Selain sebagai *compiler*, kita bisa langsung menggunakan FreePascal sebagai editor, sehingga kita bisa membuat dan mengedit kode sekaligus mengkompilasi menjadi sebuah aplikasi.

Berikut ini tampilan program FreePascal yang digunakan hingga pembahasan nanti di kelas XI.

The image shows the Free Pascal IDE window. The title bar says 'Free Pascal'. The menu bar includes 'File', 'Edit', 'Search', 'Run', 'Compile', 'Debug', 'Tools', 'Options', 'Window', and 'Help'. The editor area shows a Pascal program with the following code:

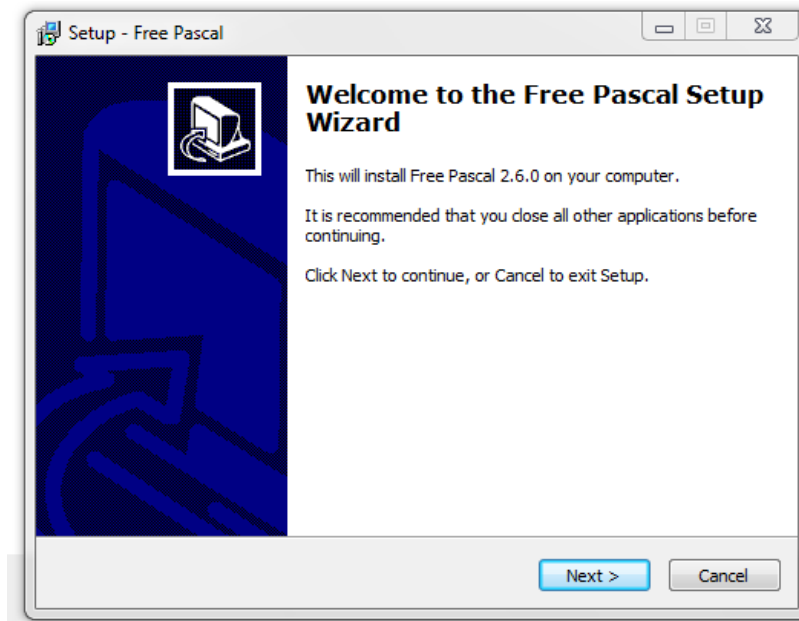
```
[.]
Program Menyapa;
Uses crt;
Begin
  Clrscr;
  Writeln('Ini Adalah Program Untuk Menyapa Budi');
  Writeln('Hai Budi Apa kabarmu hari ini?');
  Writeln('Semangat dan sukses selalu');
  Readln;
End.
```

The status bar at the bottom shows '8:4' and several keyboard shortcuts: 'F1 Help', 'F2 Save', 'F3 Open', 'Alt+F9 Compile', 'F9 Make', and 'Alt+F10 Local menu'.

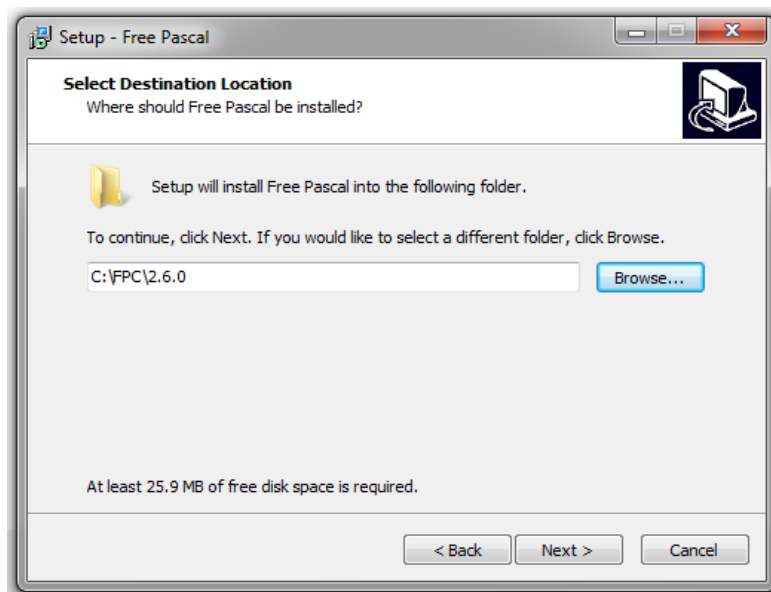
Instalasi Free Pascal

Pernahkan kalian menginstal suatu aplikasi di komputer? Mudah, bukan? Nah sekarang mari kita melihat bagaimana cara melakukan instalasi Free Pascal. Sebelumnya download dulu program instalasi Free Pascal di alamat website ressminya, www.freepascal.org. Setelah file instalasi selesai didownload, lakukan langkah-langkah berikut.

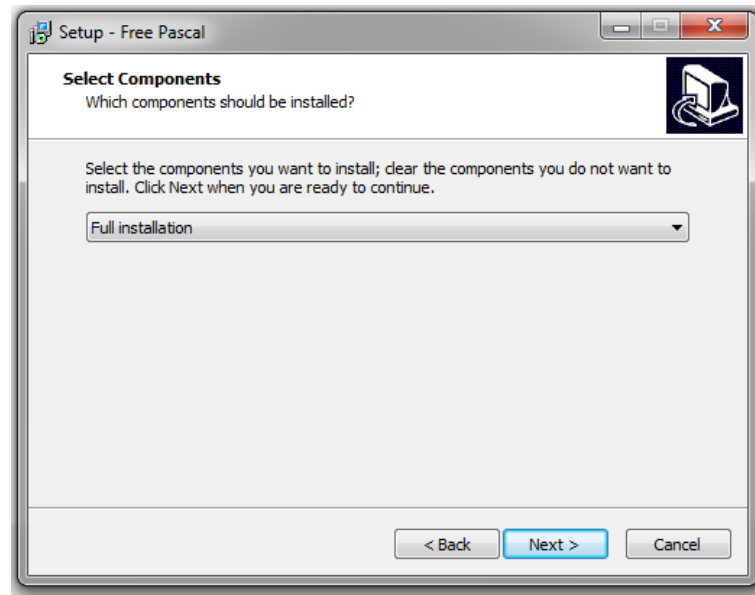
1. Double klik pada freepascal.exe setelah itu tunggu hingga muncul tampilan seperti berikut, kemudian klik next.



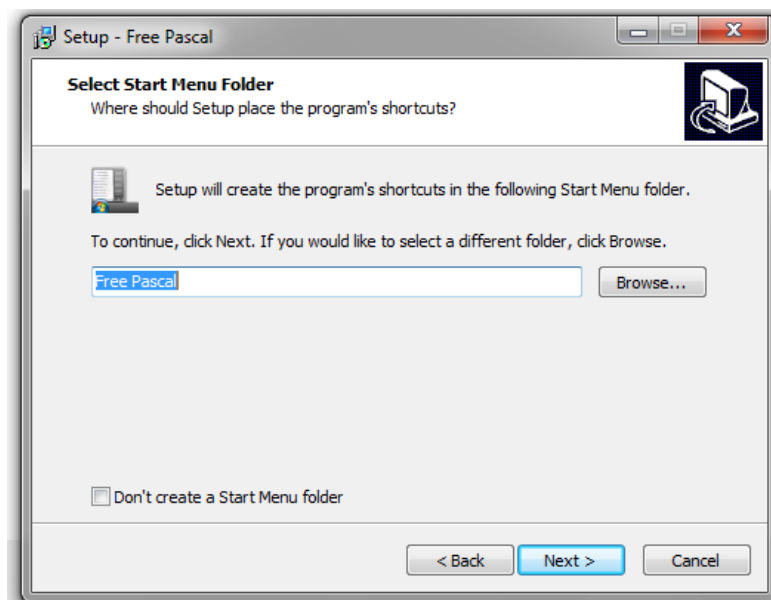
2. Kemudian akan muncul pengaturan untuk memilih lokasi penyimpanan file instalasi, secara default akan tersimpan pada folder C tapi jika ingin mengganti maka klik *browse*, kemudian klik *next*.



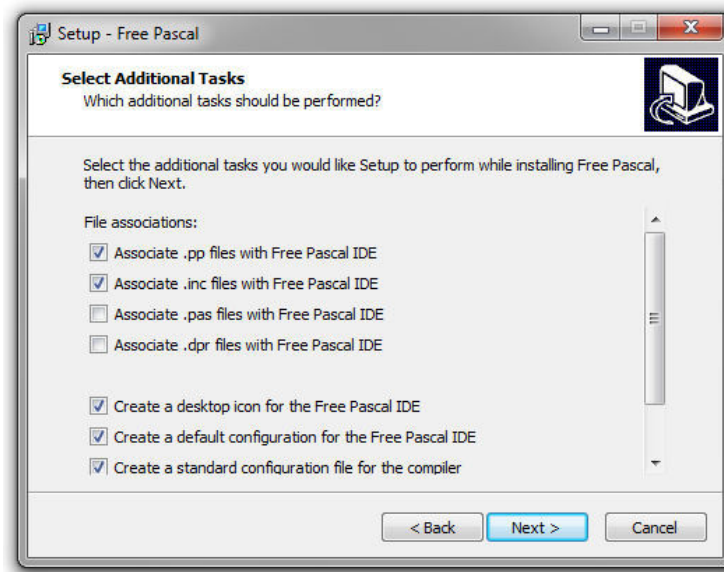
3. Memilih komponen apa saja yang akan diinstall, pilih *full installation* agar komponen yang terinstall lengkap, kemudian klik *next*.



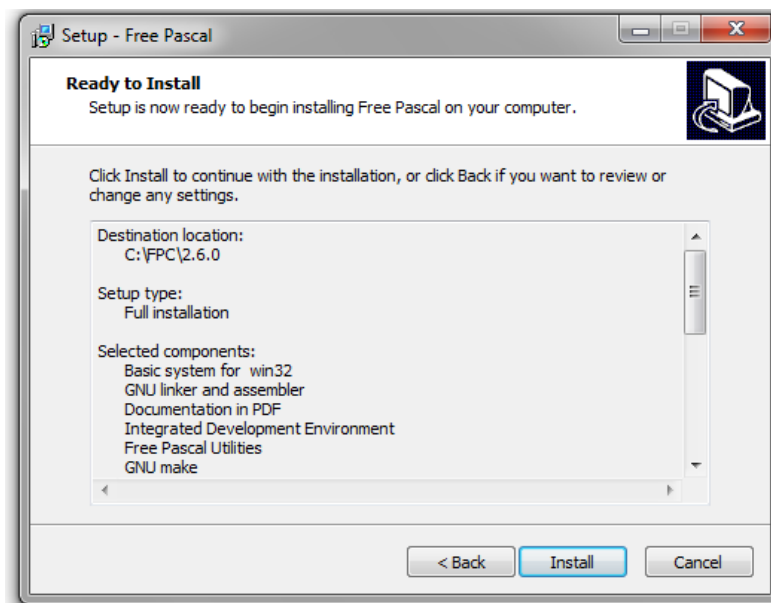
4. Memilih folder untuk penempatan setup *Free Pascal*, untuk pengaturan default maka diletakan di folder *Free Pascal*, kemudian klik *next*.



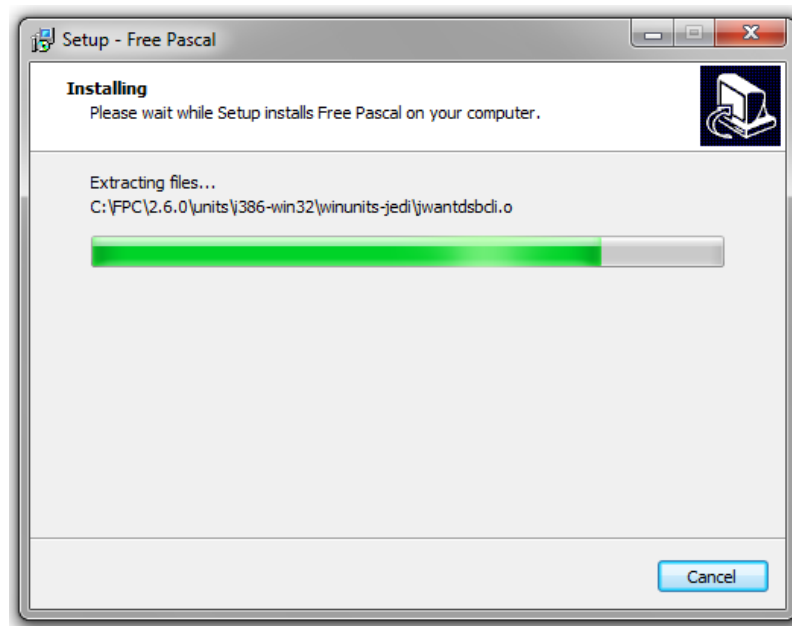
5. Memilih *Additional Task* atau fungsi tambahan.



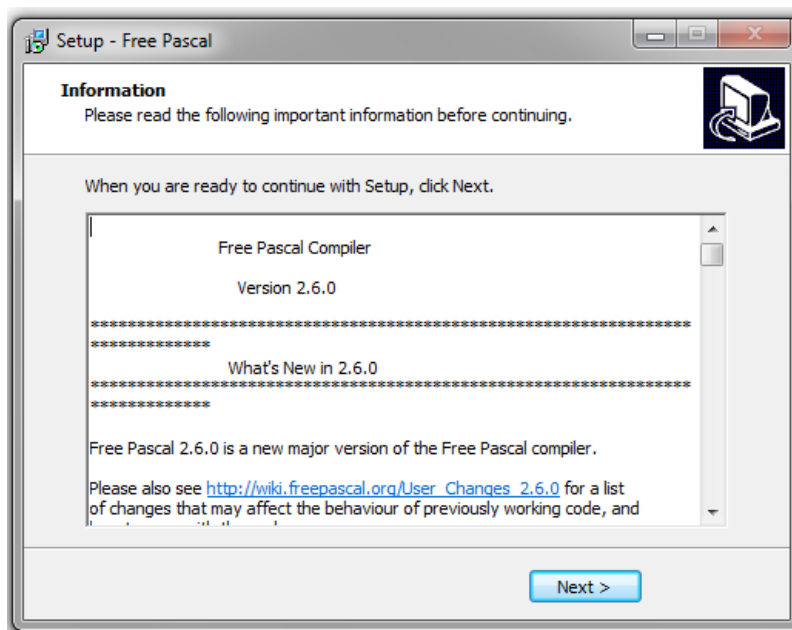
6. Kemudian muncul kotak info bahwa aplikasi siap diinstall, kemudian klik *next*.



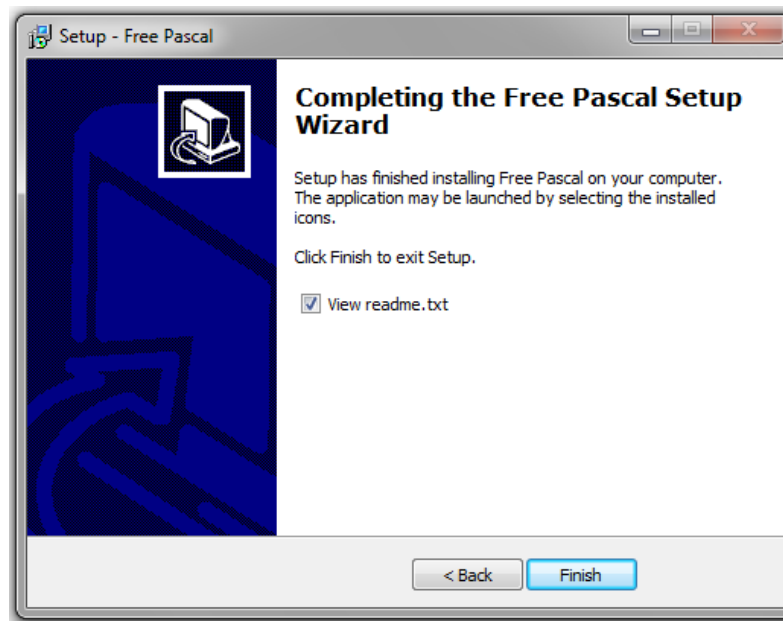
7. Tunggu hingga proses selesai.



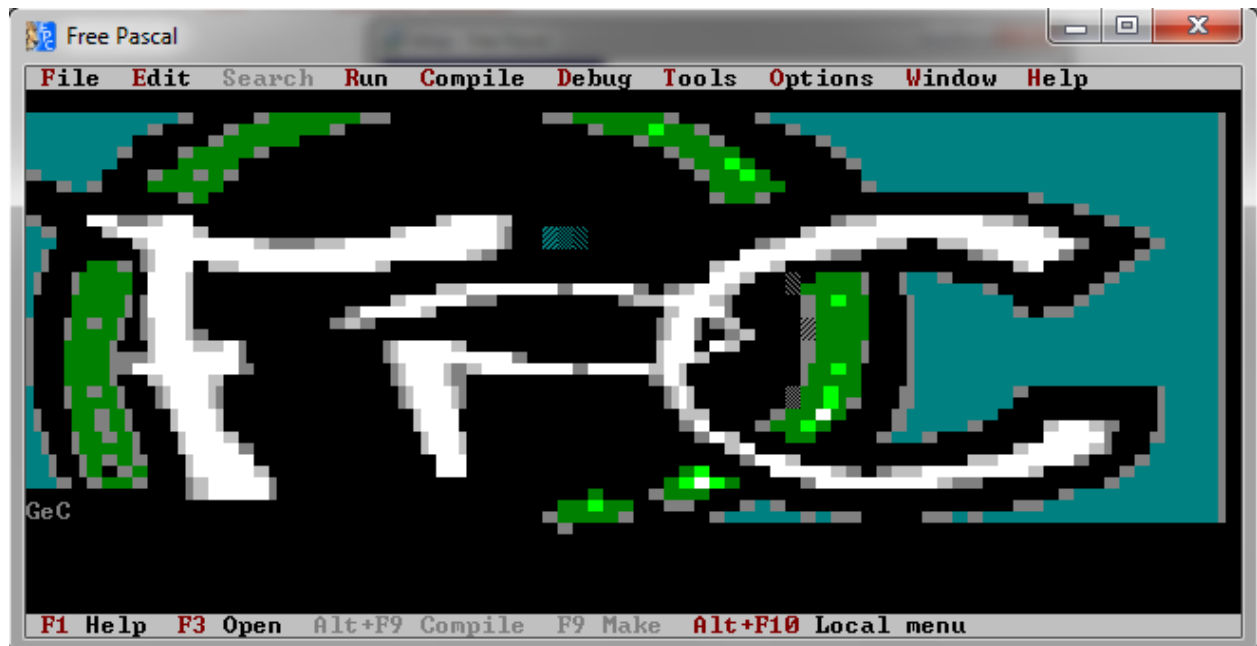
8. Kemudian akan muncul kotak informasi terkait *Free Pascal*. Kemudian klik *Next*



9. Klik *Finish*, setelah selesai



Setelah selesai double klik icon Free Pascal dan akan muncul halaman depan seperti ini:



Dengan aplikasi inilah nantinya kita akan bekerja untuk membuat program, diawali dengan menyusun **algoritma** untuk menentukan input, proses dan output yang dikehendaki kemudian menuliskan algoritma dalam bentuk kode program menggunakan format bahasa Pascal (mengkode), setelah itu mengkompilasi untuk menghasilkan program dalam bentuk file berekstensi *exe* (*executable*). Bahasan tentang Struktur Bahasa dan Kompilasi akan dikupas pada kegiatan belajar berikutnya.

Agar mengenal lebih dekat mengenai Free Pascal, lakukan kegiatan pada LKS berikut.

LEMBAR KERJA SISWA

Lakukan instruksi kerja berikut!

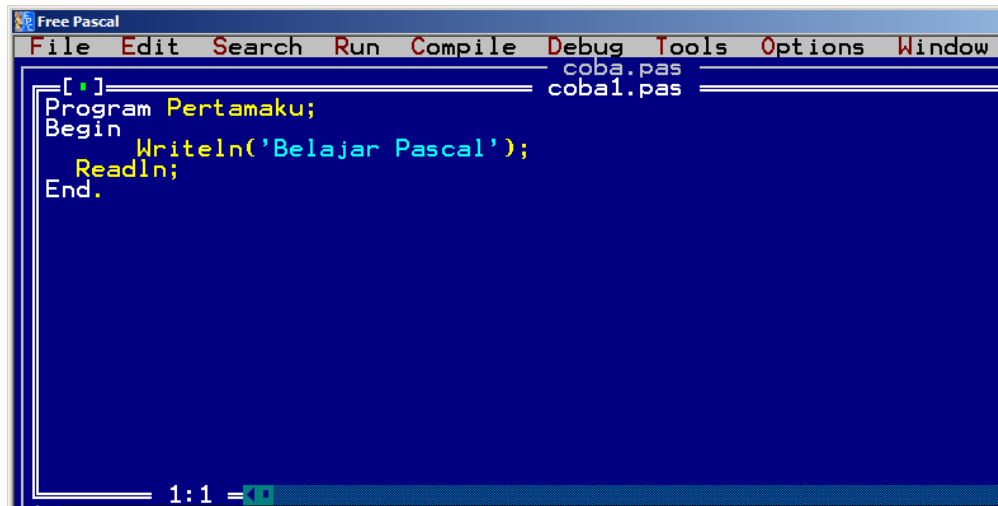
1. Lakukan Instalasi Free Pascal sesuai langkah-langkah yang telah dipaparkan sebelumnya. File bisa di unduh di <http://www.freepascal.org/download.var>.
2. Setelah selesai coba lakukan langkah-langkah berikut ini:
 - a. Double Click pada ikon Free Pascal untuk menjalankan aplikasi.
 - b. Pada Menu bar pilih File -> New
 - c. Kemudian Tuliskan kode berikut

```

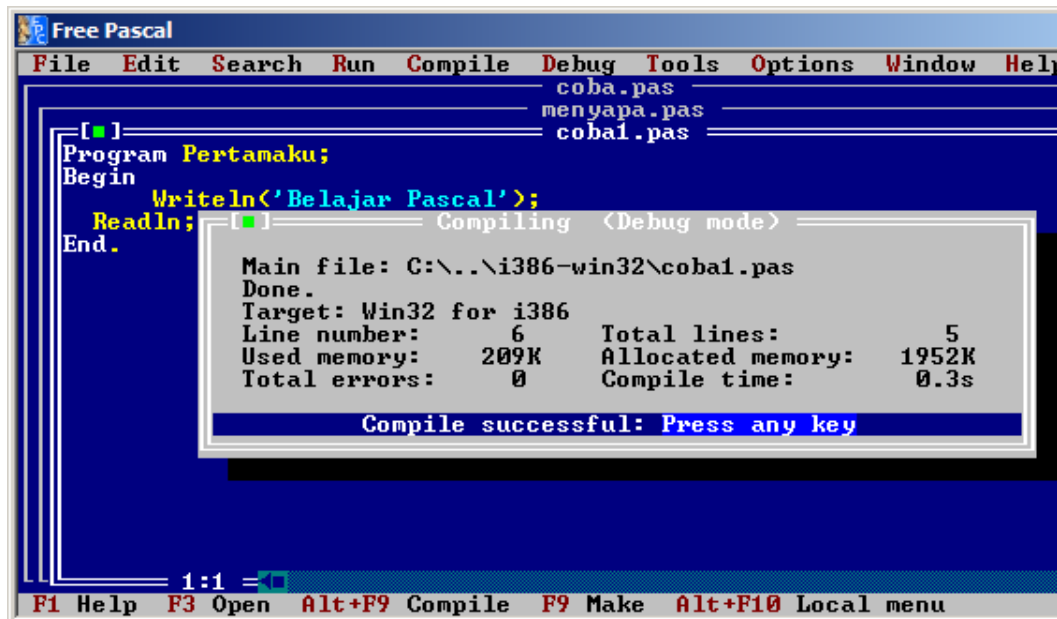
Program Pertamaku;
Begin
  Writeln('Belajar Pascal');
  Readln;
End.

```

Simpan dengan mengklik menu File->Save as, kemudian beri nama **coba1.pas**. Seperti ini tampilan kode program kalian.



Setelah itu pada Menu Bar klik Compile, kemudian pilih Compile lagi, atau tekan tombol pada keyboard Alt+F9. Berikut ini hasilnya.



Dapatkan kalian jelaskan ini proses apa? Apa arti *Compile successful*? Coba periksa pada folder di mana file coba1.pas, apakah ada file lain dengan nama coba1 juga terbentuk, apa ekstensinya?

Jawab:.....
.....
.....
.....
.....
.....

- d. Kembali ke kode program kalian di Free Pascal. Setelah kompilasi berhasil (sukses), berikutnya klik pada Menu Bar kemudian pilih Run atau tekan Ctrl + F9, apa hasilnya?

Jawab:.....
.....
.....
.....

- e. Dapatkan kalian menyebutkan tujuan program diatas? Apakah output yang muncul?

Jawab:
.....
.....
.....
.....

3. Berikut adalah program perkalian sederhana menggunakan bahasa pascal.

```
Program Perkalian;  
var; bil1, bil2, hasil : real;  
begin  
    write('Silahkan Masukan Bilangan Pertama :');  
    readln(bil1);  
    write('Silahkan Masukan Bilangan Kedua:');  
    readln(bil2);  
    hasil = bil1 * bil2;  
    writeln('Hasil perkalian =',hasil);  
    readln;  
end.
```

- a. Coba tuliskan kembali program diatas menggunakan Free Pascal. Setelah itu tekan Alt + F9 untuk mengkompilasi, jika sudah sukses, tidak ada kesalahan tekan Ctrl+F9 untuk menjalankan programnya. Jelaskan apa yang dikerjakan program tersebut?
- b. Tuliskan Algoritma dari program tersebut menggunakan bahasa kalian sendiri (bahasa natural), mulai dari input, proses dan hasilnya sebagaimana pada Semester 1.

Jawab:

- a.....
-
-
-
-
-
-
- b.....
-
-
-
-

1.1.2.4. Mengasosiasikan

Tuliskan kesimpulan kalian mengenai beberapa hal berikut.

No	Istilah	Arti/Keterangan
1	Program
2	Bahasa Pemrograman
3	Kompilasi
4	<i>Compiler</i>

	
5	Free Pascal
6	Alt+F9
7	Ctrl+F9

1.1.2.5. Mengkomunikasikan

Presentasikan hasil LKS dan kesimpulan kalian di depan kelas, diskusikan dengan teman-teman sekelas yang lain.

1.1.2.3 Rangkuman

1. Bahasa pemrograman adalah instruksi standar yang dimengerti komputer dan digunakan untuk menyampaikan kumpulan instruksi dalam alur Algoritma program.
2. Tingkatan bahasa pemrograman:
 1. Bahasa Mesin
 2. Bahasa Tingkat Rendah
 3. Bahasa Tingkat Menengah
 4. Bahasa Tingkat Tinggi
3. Untuk bahasa pemrograman tingkat tinggi diperlukan adanya jembatan berupa tools atau *compiler* agar instruksi dalam bahasa tingkat tinggi bisa dieksekusi oleh komputer. Contoh *compiler* untuk Bahasa Pascal adalah Free Pascal.

1.1.3.4 Tugas

Cermati kembali soal nomer 3, Modifikasi program tersebut menjadi program penjumlahan, Jelaskan algoritma penyelesaian program tersebut.

1.1.3.5. Uji Kompetensi

- A. Soal Pilihan Ganda

Pilihlah jawaban yang paling benar diantara pilihan a,b,c,d dan e!

1. Tingkatan paling rendah suatu bahasa pemrograman adalah...
 - a. Algoritma
 - b. Bahasa Pemrograman
 - c. Bahasa Mesin
 - d. Bahasa Assembly
 - e. Bahasa Tingkat Rendah
 2. Tingkatan bahasa pemrograman yang menggunakan unsur kata-kata manusia adalah...
 - a. Bahasa Mesin
 - b. Bahasa Biner
 - c. Bahasa Tingkat Tinggi
 - d. Bahasa Tingkat Rendah
 - e. Bahasa Menengah
 3. Tingkat bahasa pemrograman yang menggunakan gabungan antara bahasa manusia dengan kode-kode simbolik adalah...
 - a. Bahasa Tingkat Tinggi
 - b. Bahasa Mesin
 - c. Bahasa Rendah
 - d. Bahasa Tingkat Menengah
 - e. Bahasa Biner
 4. Berikut ini contoh bahasa pemrograman tingkat tinggi **kecuali...**
 - a. Pascal
 - b. PHP
 - c. Java
 - d. C
 - e. Visual Basic
 5. Berikut ini termasuk tools atau framework pengembangan pascal **kecuali...**
 - a. Free Pascal
 - b. Neat Beans
 - c. Delphi
 - d. Dev Pascal
 - e. Turbo Pascal
- B. Soal Essay
- Jawablah dengan benar pertanyaan berikut ini!
1. Apa yang dimaksud dengan bahasa pemrograman?
 2. Sebutkan fungsi tools/framework bahasa pemrograman!
 3. Apakah kelebihan tools/framework Free Pascal dibanding yang lain?
 4. Sebutkan klasifikasi tingkatan bahasa pemrograman, dan masing-masing perbedaannya.
 5. Sebutkan langka-langkah instalasi Free Pascal!

1.2. KEGIATAN BELAJAR 2 STRUKTUR BAHASA PEMROGRAMAN PASCAL

1.2.1. Tujuan Pembelajaran

Siswa memahami struktur bahasa pemrograman Pascal.

1.2.2. Aktivitas Kegiatan Belajar

Pada kegiatan sebelumnya kita telah memahami apa itu bahasa pemrograman dan kaitannya dengan algoritma, nah kali ini kita akan belajar tentang struktur dan cara penuliskode program menggunakan Bahasa Pascal. Untuk memahaminya mari lakukan kegiatan belajar berikut.

1.2.2.1. Mengamati

1. Perhatikan contoh kode program penghitung luas lingkaran berikut

```
Program Luas_lingkaran;  
uses crt;  
var r : integer;  
L : real;  
begin  
    clrscr;  
    write('Masukan nilai jari-jari = ');  
    readln (r);  
    L := 3.14*r * r * ;  
    writeln ('Luas lingkarannya adalah=',L);  
    readln;  
end.
```

2. Berikut contoh kode program perhitungan diskon

```
Program Menghitung_diskon;  
Uses crt;  
var harga,diskon,harga_akhir: real;  
begin  
    write('masukan harga barang=');  
    readln(harga);  
    write('masukan prosentase diskon barang, skala 1-100 =');  
    readln(diskon);  
    harga_akhir :=harga – (harga * diskon/100);  
    writeln('harga barang setelah didiskon adalah=', harga_akhir);  
    readln;  
end.
```

1.2.2.2. Menanya

Berdasarkan pengamatan kalian di atas, buatlah pertanyaan-pertanyaan seperti contoh berikut.

1. Dari kedua contoh diatas adakah persamaan dalam hal penulisan kode program?
2. Coba kalian sebutkan pada bagian mana saja yang terlihat sama.
3. Dari kedua contoh diatas adakah perbedaan dalam hal penyusunan program?
4. Coba kalian sebutkan pada bagian mana saja yang terlihat berbeda!
5. Bagaimana struktur kode program dalam Pascal?

1.2.2.3. Mengumpulkan Informasi/Mencoba

Kalian sudah belajar Algoritma pada materi semester lalu bukan? Nah, pada subbab inilah kalian akan mengaplikasikan ilmu yang telah kalian alami sebelumnya untuk berlatih membuat suatu program. Caranya adalah dengan menerjemahkan algoritma baik dalam bentuk pseudocode, bahasa natural ataupun flowchart ke bentuk kode (*source code*) bahasa pemrograman tertentu, atau selanjutnya disingkat kode program saja. Setiap bahasa pemrograman memiliki aturan atau tata cara penulisan kata, ekspresi maupun pernyataan disebut dengan *sintaks*.

Secara umum, struktur kode dari sebuah program komputer terdiri dari dua bagian utama yaitu bagian judul dan blok program. Sedangkan blok program terdiri dari bagian deklarasi dan pernyataan. Struktur kode program mirip atau dapat dikatakan sama dengan struktur algoritma, ada judul, deklarasi dan pernyataan (deskripsi). Yang membedakan hanya pada aturan penulisan, di mana dalam bahasa pemrograman, bahasa yang digunakan sudah baku dan detail, sehingga tidak menoleransi kesalahan-kesalahan kecil yang tidak sesuai dengan sintaksnya, meskipun hanya titik, koma, atau tanda petik saja.

Contoh struktur kode program dengan Bahasa Pascal dapat dilihat pada Gambar 1.1.

Judul Program

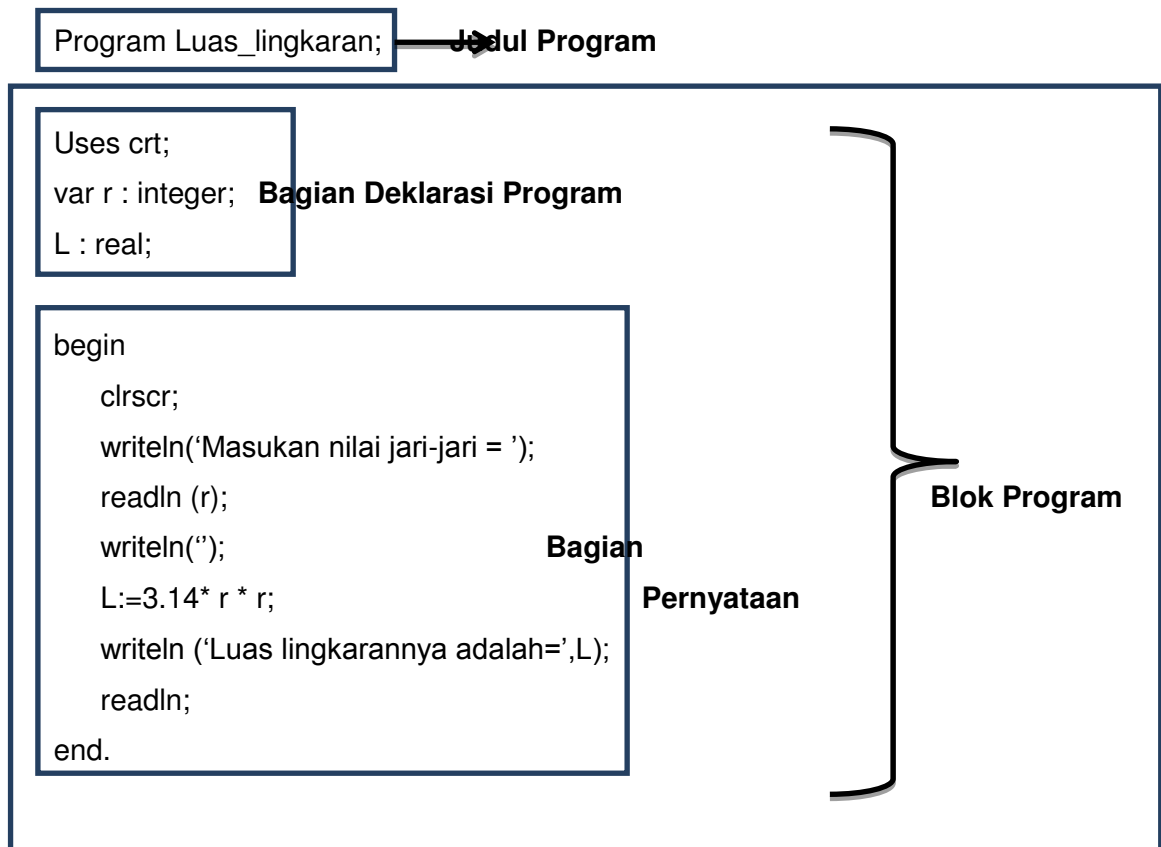
Judul program digunakan untuk mengidentifikasi nama program yang dibuat. Untuk penulisan judul program sifatnya optional jadi boleh dicantumkan atau tidak, namun untuk keperluan identifikasi lebih baik judul ada dan diberi nama sesuai dengan tujuan program. Nama program diawali dengan kata Program, sebagai kata baku dalam Pascal, diikuti dengan nama program atau *identifier*.

Aturan pemberian nama identifier, dalam hal ini termasuk judul program adalah sebagai berikut.

1. Nama identifier harus dimulai dengan karakter huruf alfabet a-z atau A-Z.
2. Setelah karakter pertama, dapat dilanjutkan dengan karakter alfanumerik (angka maupun huruf) dan underscore (_).
3. Dalam penulisan identifier tidak boleh menggunakan karakter-karakter berikut ini.
~ ! @ # \$ % ^ & * () + ` - = { } [] : " ; ' < > ? , . / |
4. Tidak boleh menuliskan identifier dengan nama yang sama pada kata kunci dari Pascal, seperti

and, array, begin, case, const, div, do, downto, else, end, file, for, forward, function, goto, if, in, label, mod, nil, not, of, or, packed, procedure, program, record, repeat, set, then, to, type, until, var, while, with

5. Jika ingin tetap digunakan, maka kata tersebut harus dirangkai dengan kata/karakter yang lain. Misal program_if, begin1.



Gambar 1.1. Struktur kode program dengan Bahasa Pascal

Berikut ini contoh pemberian nama Program yang benar.

```

Program LuasLingkaran;
Program Luas_Lingkaran
Program Luas2;
Program L_uas;
  
```

Berikut ini contoh nama program yang salah

```

Program Luas Lingkaran;
Program Luas-Lingkaran;
Program Luas^Lingkaran.
Program 2Luas;
  
```

Di akhir nama program harus diikuti dengan tanda semicolon/titik koma (;).

Identifier juga digunakan untuk memberi nama variabel ,konstanta, sub program seperti function dan prosedur, sehingga format penamaan harus sesuai dengan aturan di atas.

Deklarasi Program

Sebagaimana dalam algoritma, bagian deklarasi adalah bagian persiapan di mana semua hal yang diperlukan oleh program pada bagian deskripsi didefinisikan di sini. Yang termasuk dari bagian deklarasi adalah sebagai berikut.

1. Unit, yaitu fungsi-fungsi tambahan dari Pascal, cara memanggilnya adalah dengan menggunakan kata kunci **Uses**.

Sebagai contoh pada program di atas adalah penggunaan **Uses crt**. **Uses crt** diperlukan karena kode program menggunakan perintah **clrscr** untuk membersihkan layar. Perintah ini disediakan oleh unit **crt**.

2. Konstanta, yaitu suatu nilai yang besarnya tetap (konstan) selama program dijalankan
3. Deklarasi Variabel, yaitu *identifer* atau nama-nama untuk menyimpan nilai beserta denan tipe datanya.
4. Deklarasi Tipe data baru yang didefinisikan oleh *programmer*
5. Prosedur dan fungsi

Penjelasan lengkap mengenai komponen bagian deklarasi akan dibahas pada Bab 2 nanti.

Bagian Pernyataan/Statement

Pada bagian ini dituliskan urutan-urutan instruksi yang perlu dieksekusi oleh program. Aturan penulisan;

1. Bagian program inti diawali dengan Begin(tanpa diikuti tanda baca) dan diakhiri dengan End (diikuti tanda **titik (.)**)
2. Dalam Pascal penggunaan huruf besar dan kecil tidak dipermasalahkan (tidak sensitive-case)
3. Untuk setiap perintahatau pernyataan, diakhiri dengan titik koma (;), boleh ada lebih dari 1 pernyataan dalam tiap baris.

Contoh:

Pernyataan1;

Pernyataan2;

Boleh juga dituliskan

Pernyataan1;Pernyataan2;

Namun demikian agar penulisan lebih terstruktur dan mudah dalam menemukan adanya kesalahan, lebih baik satu pernyataan satu baris dan digunakan aturan indentasi sebagaimana ketika menyusun algoritma pada Semester 1.

4. Jika ada struktur percabangan (kondisi bersyarat) atau perulangan dengan lebih dari satu pernyataan, diawali dengan begin dan diakhiri dengan end dan titik koma (;)

Contoh:

If (kondisi) then

Begin

Pernyataan_If1;

```
Pernyataan_If2;
```

```
End;
```

Jika pernyataan dalam struktur hanya satu saja maka Begin dan End boleh dihilangkan

```
If (kondisi) then
```

```
    Pernyataan_If1;
```

5. Komentar dituliskan di antara tanda kurung {}

Contoh

```
Program Luas_lingkaran; {Program ini menghitung luas lingkaran}
```

```
uses crt;
```

```
var r : integer;
```

```
L : real;
```

Tulisan “Program ini menghitung luas lingkaran” tidak akan tercetak pada program yang sudah jadi, karena diletakkan di antara kurung { } maka ini hanya sebagai komentar untuk menerangkan bagian Judul mengenai apa tujuan program. Hal ini memudahkan orang lain untuk memahami atau mengedit program tersebut.

Sebagai tambahan, dalam bahasa Pascal, penulisan write atau WRITE atau Write dianggap sama karena tidak case-sensitive seperti di Bahasa C. Demikian juga dengan nama identifier a atau A, luas atau Luas, Phi atau PHI dianggap sebagai *identifier* yang sama.

Untuk lebih jelasnya perhatikan contoh-contoh berikut.

Contoh 1

Program Pascal yang paling sederhana adalah sebagai berikut.

```
Begin
```

```
End.
```

Program di atas hanya terdiri dari begin dan end saja, tapi dapat dijalankan oleh Pascal, tanpa ada kesalahan. Namundemikian tidak ada output apapun dalam program ini.

Contoh 2

```
Var a:integer
```

```
Begin
```

```
    a:=5;    {Memberikan nilai a dengan 5}
```

```
    writeln(a);    {Menuliskan isi a}
```

```
    readln;
```

```
End.
```

Program di atas meskipun tanpa judul dapat dijalankan oleh Pascal, di mana dalam program tersebut menampilkan isi dari variabel *a*, yang sudah dideklarasikan di awal. Tulisan di antara kurung { } adalah komentar yang tidak akan dieksekusi oleh program. Komentar ini bertujuan untuk memberikan informasi kepada pengguna mengenai kode program yang dituliskan.

Contoh 3

Program Menghitung Luas Segitiga

```
Program Luas_Segitiga;  
uses crt;  
var Luas,a,t : real;  
begin  
    clrscr;  
    write('masukan nilai alas=');  
    readln(a);  
    write('masukan nilai tinggi=');  
    readln(t);  
    Luas:=1/2 (a * t);  
    writeln('luas segitiganya adalah=', Luas);  
    readln;  
end.
```

Kode program ini lebih lengkap, ada judul, bagian deklarasi, dan bagian pernyataan. Dari program di atas maka dapat diketahui algoritma dalam bahasa natural sebagai berikut.

Algoritma LuasSegitiga

Persiapkan variable *Luas,a,t* bertipe real

Langkah-langkah

1. Meminta inputan nilai alas
2. Membaca nilai input dan menyimpan dalam variable *a*
3. Meminta inputan nilai tinggi
4. Membaca nilai input dan menyimpan dalam variable *t*
5. Melakukan perhitungan luas segitiga dengan rumus $\frac{1}{2} (a * t)$, dan menyimpannya dalam variable *Luas*
6. Menampilkan hasil *Luas*

Berikut ini penjelasan kode program satu demi satu.

1. **Judul Program**, nama programnya adalah Luas_segitiga
2. **Bagian Deklarasi**, diantaranya
 - ✓ **Uses crt**; ini digunakan untuk menyiapkan layar di awal program.
 - ✓ **Var Luas,a,t: real**, artinya menyiapkan variabel-variabel seperti Luas, a, t sekaligus tipe datanya yaitu bilangan real.
3. **Bagian Pernyataan**, terdiri dari beberapa urutan instruksi:
 - ✓ **Begin** Digunakan sebagai penanda akan memulai instruksi program
 - ✓ **Clrscr**, digunakan untuk membersihkan layar
 - ✓ **Write('Masukan nilai alas=');** Meminta input nilai alas
 - ✓ **Readln(a);** Menyimpan inputan nilai dalam variable a
 - ✓ **Writeln('Masukan nilai tinggi=');** Meminta input nilai tinggi
 - ✓ **Readln(t);** Menyimpan inputan nilai dalam variable t
 - ✓ **Luas:=1/2 (a * t);** Proses perhitungan Luas Segitiga
 - ✓ **Writeln('Luas Segitiganya adalah', Luas);** Menampilkan luas segitiga
 - ✓ **End.** Digunakan untuk mengakhiri instruksi program.

Agar lebih jelas pemahaman kalian mengenai struktur kode program dan sintaks penulisan dalam bahasa Pascal, kerjakan LKS berikut.

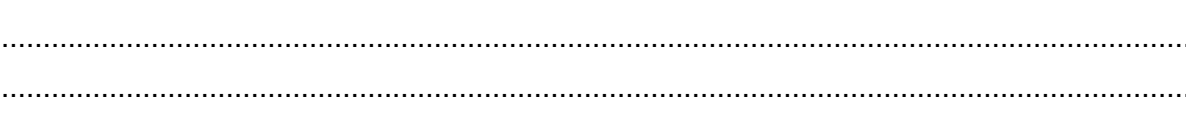
LEMBAR KERJA SISWA

1. Perhatikan kode program berikut

```
Program Konversi Panjang;  
Uses crt;  
Var cm, m, mm:real;  
Begin  
    Clrscr;  
    Write('Masukan nilai panjang dalam Centimeter=')  
    Readln(cm)  
    m=cm/100  
    mm=cm*10  
    Writeln('Konversi nilai dalam Meter=',m)  
    Writeln('Konversi nilai dalam Mili Meter=',mm)  
    Readln  
End
```


- Jawab:

a.

[illegible][illegible]

[illegible]

Dari hasil pembahasan dan pekerjaan LKS, tuliskan kesimpulan kalian mengenai hal-hal berikut.

No	Istilah	Arti/Keterangan
1	Bagian-bagian Kode Program
2	Fungsi Judul Program
3	Fungsi Bagian deklarasi
4	Fungsi bagian pernyataan

5	Identifier
6	Sintaks

1.2.2.5. Mengkomunikasikan

Presentasikan hasil pekerjaan LKS dan kesimpulan kalian di depan kelas, diskusikan dengan teman-teman sekelas yang lain.

1.2.3. Rangkuman

Struktur bahasa pemrograman pascal terdiri dari:

1. Judul Program, untuk mengidentifikasi nama program.
2. Bagian Deklarasi, digunakan untuk mengenalkan berbagai pengenalan atau kebutuhan yang diperlukan program ketika dijalankan.
3. Bagian Pernyataan, untuk menjelaskan urutan instruksi dalam menyelesaikan program

1.2.4. Tugas

Carilah contoh program sederhana menggunakan bahasa pascal, kemudian coba jelaskan bagaimana algoritma untuk menyelesaikan program tersebut, sebutkan pula struktur programnya.

1.2.5. Uji Kompetensi

A. Soal Pilihan Ganda

Pilihlah jawaban yang paling benar diantara pilihan a,b,c,d dan e!

1. Berikut ini bagian dari struktur pascal yang berfungsi untuk mengidentifikasi nama program adalah...
 - a. Judul Program
 - b. Header Program
 - c. Footer Program
 - d. Deklarasi
 - e. Statement
2. Bagian dari struktur pascal yang berfungsi untuk urutan-urutan instruksi yang perlu dieksekusi oleh program adalah...
 - a. Deklarasi
 - b. Judul Program
 - c. Statement
 - d. Header

- e. Content
- 3. Bagian dari struktur pascal yang berfungsi untuk mengenalkan berbagai pengenalan adalah...
 - a. Deklasrasi
 - b. Statement
 - c. Judul Program
 - d. Content
 - e. Header
- 4. Begin pada struktur pascal berfungsi untuk...
 - a. Mengulangi Program
 - b. Memulai eksekusi statement
 - c. Mengakhiri eksekusi
 - d. Menghapus Program
 - e. Menghapus Layar
- 5. Tanda semicolon (;) pada struktur bahasa pascal digunakan untuk...
 - a. Mendeklarasikan tipe data variable
 - b. Mendeklarasikan nilai suatu variable
 - c. Menutup statement
 - d. Memulai Program
 - e. Mengakhiri Program

B. Soal Essay

Jawablah dengan benar pertanyaan berikut ini!

- 1. Apakah yang dimaksud dengan struktur bahasa pemrograman?
- 2. Sebutkan apa saja struktur bahasa pascal!
- 3. Sebutkan aturan penulisan masing-masing bagian pada struktur bahasa pascal!

Cermati soal berikut ini:

```
Program hitung_diskon;  
Uses crt;  
Var a,b,c :Real;  
Begin  
End;  
Writeln ('Masukan Harga Asal=');  
Readln(a);  
Writeln('Masukan Besarnya Diskon=');  
Readln(b);  
c:=a-(a*b);  
Writeln('Harga Setelah didiskon =',c);  
Readln;  
End.
```

- 4. Tuliskan kembali program diatas dan tunjukan strukturnya!
- 5. Dengan menggunakan bahasa natural tuliskan alur algoritma program tersebut!

1.3. KEGIATAN BELAJAR 3 STANDAR INPUT OUTPUT

Pada kegiatan belajar sebelumnya kalian telah memahami struktur kode program dengan bahasa Pascal. Dalam sebuah kode program terdapat Judul Program, Deklarasi dan Pernyataan. Pada bagian pernyataan dituliskan rangkaian instruksi untuk menyelesaikan suatu permasalahan atau program, dimulai dari instruksi untuk memasukan data atau **INPUT** data kemudian data itu diproses atau **PROSES** data dan menampilkan hasilnya atau **OUTPUT** data. Kalian sudah diperkenalkand engan perintah read dan write di kegiatan belajar sebelumnya. Dua perintah ini adalah standar input dan output dalam Bahasa Pascal.

1.3.1. Tujuan Pembelajaran

Siswa memahami standar input dan output dalam bahasa pemrograman Pascal dan menggunakannya untuk membuat program.

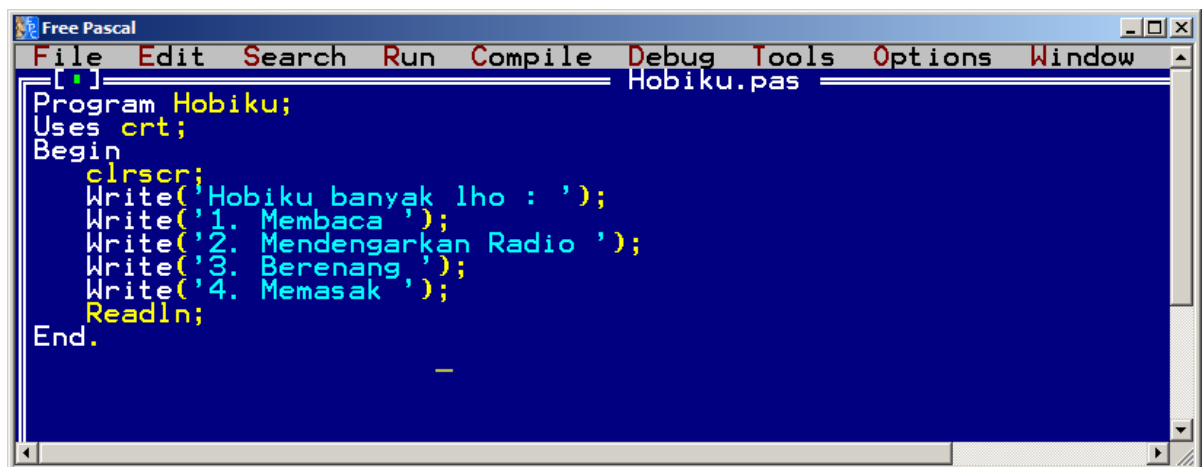
1.3.2. Aktivitas Kegiatan Belajar

1.3.2.1. Mengamati

Kasus 1

Perhatikan kode program dan keluaran dari dua program berikut.

1. Program Pertama



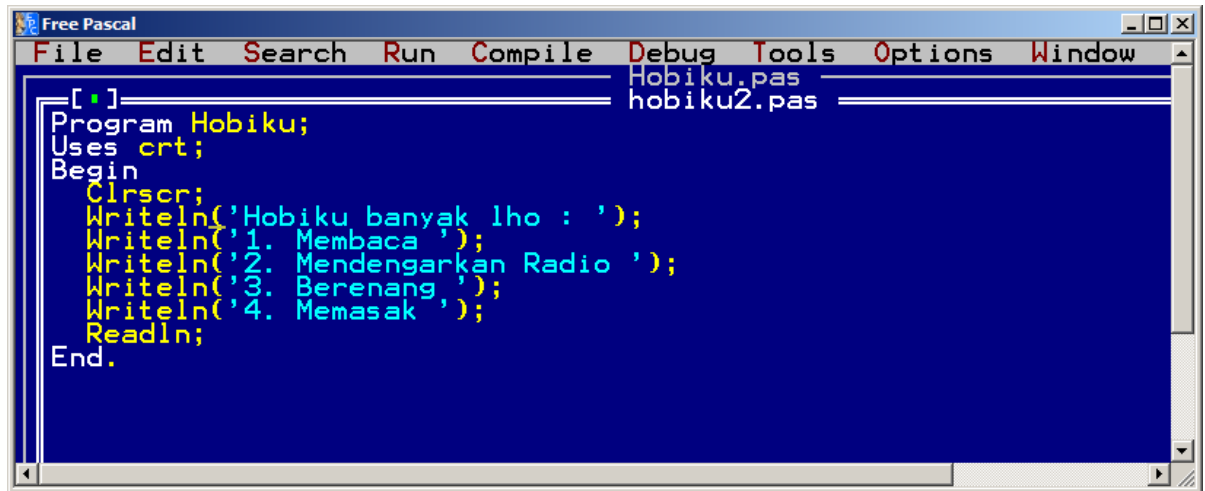
```
Free Pascal
File Edit Search Run Compile Debug Tools Options Window
[.] Hobiku.pas
Program Hobiku;
Uses crt;
Begin
  clrscr;
  Write('Hobiku banyak lho : ');
  Write('1. Membaca ');
  Write('2. Mendengarkan Radio ');
  Write('3. Berenang ');
  Write('4. Memasak ');
  Readln;
End.
```

Output:



```
Free Pascal
Hobiku banyak lho : 1. Membaca 2. Mendengarkan Radio 3. Berenang 4. Memasak
```

2. Program Kedua

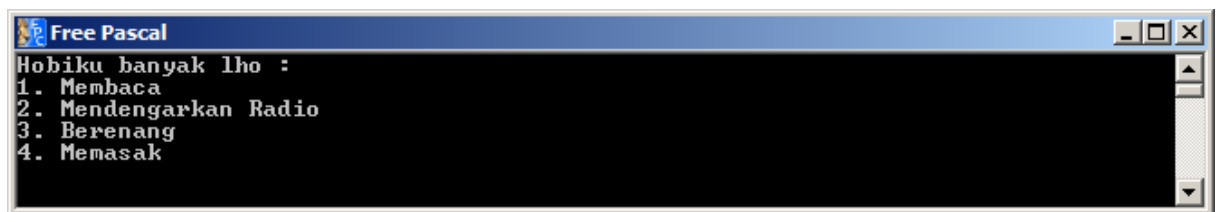


```

[.]
Program Hobiku;
Uses crt;
Begin
  Clrscr;
  Writeln('Hobiku banyak lho : ');
  Writeln('1. Membaca ');
  Writeln('2. Mendengarkan Radio ');
  Writeln('3. Berenang ');
  Writeln('4. Memasak ');
  Readln;
End.

```

Output:



```

Free Pascal
Hobiku banyak lho :
1. Membaca
2. Mendengarkan Radio
3. Berenang
4. Memasak

```

Kasus 2

Hasil menjalankan sebuah kode program, nampak di layar berupa inputan dan tampilan berikut.

1. Program pertama

2

3

4

Hasil Penjumlahan = 9

2. Program Kedua

2 3 4

Hasil Penjumlahan = 9

3. Program Ketiga

2 3

4

Hasil Penjumlahan = 9

Bilangan 2, 3 dan 4 adalah inputan dari user sedangkan Hasil Penjumlahan adalah outputnya.

1.3.2.2. MENANYA

Setelah mengamati, coba buat pertanyaan seperti pertanyaan berikut ini:

Kasus1

1. Apa perbedaan diantara kedua algoritma tampilan hobi program diatas?
2. Instruksi apa yang membuat tampilan dari kedua program berbeda?
3. Perintah apa saja dalam Pascal untuk menampilkan output? Apa fungsi tiap perintah input dan bagaimana sintaksnya?

Kasus 2

1. Apa perbedaan diantara ke dua cara membaca nilai input diatas?
2. Instruksi apa yang membuat cara membaca nilai inputan di atas?
3. Perintah apa saja dalam Pascal untuk menampilkan output? Apa fungsi tiap perintah input dan bagaimana sintaksnya?

1.3.2.3. Mengumpulkan Informasi/Mencoba

Dalam pemrograman, secara garis besar ada tiga tahap, yaitu INPUT, PROSES, dan OUTPUT. Ketiganya terangkai dalam bagian pernyataan dalam kode program. Standar masukan pada Pascal adalah keyboard, sedangkan standar output dalam Pascal adalah monitor. Artinya dalam Pascal umumnya semua input dimasukkan lewat keyhasilnya juga pada file. board dan output ditampilkan melalui layar monitor. Selain itu, dalam Pascal juga dikenal standar input output lain, yaitu dengan melakukan operasi pada file, artinya kita bisa membaca input dari file dan mencetak pada file. Namun demikian pembahasan mengenai operasi file akan dibahas di kelas XI nanti.

Standar Output.

Lihat kembali algoritma pada kasus 1

Untuk menampilkan data ke layar digunakan instruksi Write dan Writeln. Apa bedanya?

{Kode Program 1}

```
Program Hobiku;
Uses crt;
Begin
  Clrscr;
  Write('Hobiku banyak lho : ');
  Write('1. Membaca ');
  Write('2. Mendengarkan Radio ');
  Write('3. Berenang ');
  Write('4. Memasak ');
  Readln;
End.
```

{Kode Program 2}

```
Program Hobiku;
Uses crt;
Begin
  Clrscr;
  Write('Hobiku banyak lho : ');
  Writeln('1. Membaca ');
  Writeln('2. Mendengarkan Radio ');
  Writeln('3. Berenang ');
  Writeln('4. Memasak ');
  Readln;
End.
```


Kode program 1 hanya menggunakan perintah `write` sedangkan kode program 2 menggunakan perintah `writeln`. Hasilnya pada kode program satu menghasilkan output semua dalam satu baris, sedangkan kode program 2, satu perintah cetak ditulis pada satu baris. Apa kesimpulannya?

Ada beberapa perbedaan untuk instruksi `write` dan `writeln`:

1. **Write** digunakan untuk menampilkan data ke layar tanpa disertai pergantian baris baru, jadi data berikutnya akan menyambung pada baris yang sama
2. **Writeln** digunakan untuk menampilkan data ke layar kemudian mencetak baris baru.

Perintah **write** atau **writeln** dapat digunakan untuk menuliskan beberapa hal berikut:

a. Nilai

Contoh :

<code>Write (5);</code>	{menuliskan nilai bilangan bulat 5}
<code>Writeln('Halo apa kabar');</code>	{menuliskan string 'halo apa kabar'}
<code>Write ('a');</code>	{menuliskan karakter 'a'}
<code>Writeln(3.14);</code>	{menuliskan bilangan real 3.14}

b. Variabel

Contoh:

<code>A:=5;</code>	
<code>Kata:='Halo apa kabar';</code>	
<code>StatusLulus:=True;</code>	
<code>Write(A);</code>	{menuliskan isi variabel A}
<code>Writeln(Kata);</code>	{menuliskan isi variabel Kata}
<code>Writeln(StatusLulus);</code>	{menuliskan isi variabel StatusLulus}

c. operasi nilai-nilai atau variabel

Contoh:

<code>writeln(3+2);</code>	{menuliskan hasil jumlahan }
<code>write(A*2);</code>	{menuliskan hasil perkalian A*2}
<code>write('Aku'+ 'Bisa');</code>	{menuliskan string 'Aku' diikuti 'bisa'}

d. Gabungan ketiga bentuk di atas (nilai, variabel, dan operasinya)

Contoh:

<code>Write('Luas segitiga= ', Luas)</code>	{menuliskan 'Luas segitiga = ', diikuti nilai variabel luas}
<code>Write(a,b);</code>	{menuliskan nilai variabel a diikuti nilai variabel b}
<code>Writeln (Hasil penjumlahan adalah ', a+b);</code>	
<code>Write ('Luas segitiga dengan alas ', a, 'dan tinggi ', t, 'adalah ', Luas);</code>	
	{menuliskan string diikuti nilai a, kemudian string, nilai t, string, dan nilai Luas}

Standar Output

Perhatikan kembali berbagai macam bentuk output pada Kasus 2.

2	2 3 4	2 3
3	Hasil Penjumlahan =9	4
4		Hasil Penjumlahan =9
Hasil Penjumlahan =9		

Input berupa bilangan 2, 3, dan 4 dibaca oleh program melalui perintah membaca input. Dalam Pasca perintah dasar membaca input adalah Read dan Readln. Apa bedanya? Kode-kode berikut adalah kode dari pembacaan ketiga inputan di atas.

{Kode 1} Readln(A); Readln(B); Readln (C); Writeln('Hasil penjumlahan =',A+B+C);	{Kode 2} Read (A); Read(B); Readln(C); Writeln('Hasil penjumlahan =',A+B+C); Alternatif : Readln (A, B, C); Writeln('Hasil penjumlahan =',A+B+C);	{Kode 3} Read(A); Readln(B); Readln(C); Writeln('Hasil penjumlahan =',A+B+C); Alternatif : Readln (A, B); Readln (C); Writeln('Hasil penjumlahan =',A+B+C);
Output: 2 3 4 Hasil Penjumlahan =9	Output 2 3 4 Hasil Penjumlahan =9	Output 2 3 4 Hasil Penjumlahan =9

Berdasarkan contoh kode program di atas, apa perbedaan antara perintah read dan readln?

1. **Read** mengambil/membaca nilai dari layar yang diinputkan melalui keyboard menyimpannya dalam variable dan kursor tetap pada baris input.
2. **Readln** mengambil/membaca nilai dari layar yang diinputkan melalui keyboard kemudian memerintahkan ganti baris (kursor turun pada baris berikutnya)

Untuk lebih memahami perintah Write, Writeln, Read dan Readln perhatikan contoh berikut:

Contoh1

```

Program Latihan1;
Uses crt;
Var a: String[30];
    b : Integer;
    c: String[3];
Begin
    Clrscr;

```

```

Write('Masukan Nama:');           {Menampilkan string 'Masukkan nama:'}
Readln(a);                         {Menyimpan data dalam variabel a bertipe string}
Write('Masukan Umur:');            {Menampilkan string 'Masukkan Umur'}
Readln(b);                         {Menyimpan data dalam variabel b bertipe Integer}
Write('Masukan Pendidikan Terakhir:');
Readln(c);                         {Menyimpan data dalam variabel c bertipe String}
Write('Nama Saya',a, 'Umur Saya',b, 'Pendidikan Terakhir:',c);{Menampilkan hasil}
Readln;
End.

```

Maka ketika dijalankan akan muncul:

```

Masukan Nama:Anggun
Masukan Umur: 20
Masukan Pendidikan terakhir: S2
Nama Saya Anggun Umur Saya 20 Pendidikan Terakhir S2

```

Mengatur format penulisan

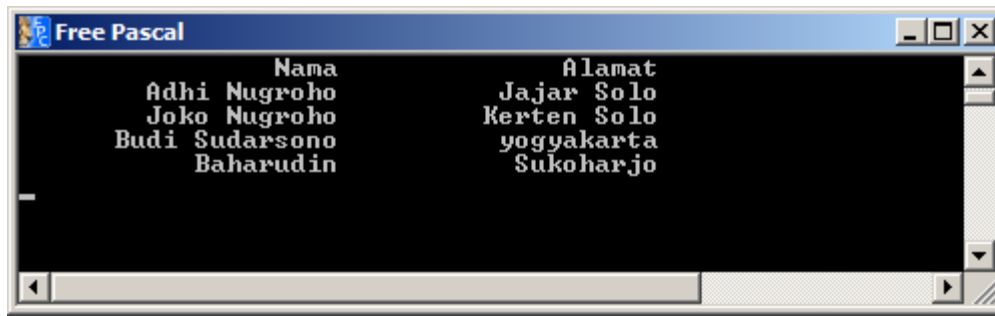
Secara default, perintah write atau writeln menuliskan dengan rata kiri, namun kita juga bisa menuliskan dengan rata kanan, dengan memberikan ruang yang lebih pada data yang akan dituliskan. Perhatikan contoh berikut.

```

Free Pascal
File Edit Search Run Compile Debug Tools Options Window Help
[ ] rata.pas 6=[ ]
Program TulisRata;
uses crt;
begin
  clrscr;
  write('Nama':20);writeln('Alamat':20);
  write('Adhi Nugroho':20);writeln('Jajar Solo':20);
  write('Joko Nugroho':20);writeln('Kerten Solo':20);
  write('Budi Sudarsono':20);writeln('yogyakarta':20);
  write('Baharudin':20);writeln('Sukoharjo':20);
  readln;
end.
4:10
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

```

Keluarannya adalah sebagai berikut.



Dengan menambahkan :20 di belakang nilai yang dituliskan akan memberikan ruang 20 karakter untuk menulis, sehingga penulisannya menjadi rata kanan. Ini bisa diterapkan terhadap nilai maupun variabel.

Contoh :

```

Write (nama:20);           {nama adalah variabel bertipe string}
Write(tinggi_badan:10);     {tinggi_badan adalah variabel dengan tipe integer}
Write(5/2:10);              {menuliskan nilai 5/2 dengan diberikan ruang 10 karakter}
  
```

Khusus untuk penulisan bilangan real, dapat juga dituliskan format berapa angka di belakang koma. Misalkan sebagai berikut.

```

Write(Rerata:10:2)
  
```

Artinya nilai rerata dituliskan pada ruang dengan 10 digit dengan dua angka di belakang koma.

Agar lebih memahami bagaimana format input dan output, kerjakan LKS berikut dengan teman sekelompok kalian.

LEMBAR KERJA SISWA

1. Berikut tampilan suatu program setelah dijalankan

```

===Bintangku===
*
*
**
*
**
**
  
```

- a. Dengan konsep `write` dan `writeln` tuliskan program yang menampilkan pola bintang diatas, perintah hanya boleh dituliskan `write` (*) atau `writeln`(*) saja, jadi hanya satu bintang untuk satu kali cetak.
- b. Untuk mengetahui kebenaran program maka coba jalankan program tersebut di Free Pascal

Jawab:

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Lihat kembali kode program pada Contoh 1

```
Program Latihan1;  
Uses crt;  
Var a: String[30];  
    b : Integer;  
    c: String[3];  
Begin  
    Clrscr;  
    Write('Masukan Nama:');  
    Readln(a);  
    Write('Masukan Umur:');  
    Readln(b);  
    Write('Masukan Pendidikan Terakhir:');  
    Readln(c);  
    Write('Nama Saya',a, 'Umur Saya',b, 'Pendidikan Terakhir:',c);  
    Readln;  
End.
```

- a. Salin kode program di atas ke dalam Free Pascal, kemudian ganti write diganti dengan writeln. Tuliskan bagaimana outputnya!
- b. Jelaskan apa perbedaanya dibandingkan dengan menggunakan perintah write saja, manakah yang lebih sesuai?

Jawab

.....

.....

.....

.....

.....

.....

.....

.....

.....

3. Berikut ini program untuk perhitungan keliling lingkaran.

- a. Lengkapilah program berikut ini, program membaca inputan jari-jari dan menampilkan hasil perhitungannya.

```

Program VolumeKerucut;
Uses crt;
Var phi, r, t, Volume : Real;
Begin
    Clrscr;
    phi := .....;
    write('.....'); {meminta inputan jari-jari}
    readln(.....); {membaca nilai jari-jari}
    write('.....'); {meminta inputan tinggi}
    readln('.....'); {membaca nilai tinggi}
    Volume:= .....; {menghitung volume}
    Writeln(.....); {mencetak output}
    Readln;
End.

```

- b. Untuk mengetahui kebenarannya salin kedalam Free Pascal

Jawab:

- a. Program Keliling_lingkaran;

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

1.3.2.4. Mengasosiasikan

Tuliskan kesimpulan kalian mengenai perintah input dan output pada Bahasa Pascal dengan melengkapi tabel isian berikut.

No	Istilah	Arti/Keterangan
1	Perintah write ()
2	Perintah writeln()
3	Perintah read()
4	Perintah readln()

1.3.2.5. Mengkomunikasikan

Presentasikan hasil pekerjaan kalian di depan kelas, diskusikan dengan teman sekelas yang lain.

1.3.3. Rangkuman

1. Perintah membaca input dalam Bahasa Pascal menggunakan perintah dasar berikut.

- Read** mengambil/membaca nilai dari layar yang diinputkan melalui keyboard menyimpannya dalam variable dan kursor tetap pada baris input.

- b. **Readln** mengambil.membaca nilai dari layar yang diinputakan melalui keyboard kemudian kursor turun pada baris berikutnya
2. Perintah cetak dalam Bahasa Pascal menggunakan perintah dasar berikut.
- a. **Write** digunakan untuk menampilkan data ke layar tanpa disertai mencetak baris baru jadi data berikutnya akan menyambung pada baris yang sama
 - b. **Writeln** digunakan untuk menampilkan data ke layar kemudian mencetak baris baru.

1.3.4. Tugas

1. Buatlah program sederhana untuk mencetak tampilan berikut (tanpa inputan)
Nama : Dwi Septiningsih
Alamat : Solo
Sekolah : SMK Negeri 7 Surakarta
Bidang : Multimedia
Rata-rata UAN : 8.5
2. Buatlah program sederhana untuk membaca inputan bilangan berikut.
1
2 3
4 5 6
7 8 9 10

1.3.5. Uji Kompetensi

A. Soal Pilihan Ganda

1. Kode pascal yang digunakan untuk menampilkan data ke layar tanpa disertai mencetak baris baru adalah...
 - a. Write
 - b. Writeln
 - c. Read
 - d. Readln
 - e. Create
2. Kode pascal yang digunakan untuk menampilkan data ke layar disertai mencetak baris baru adalah...
 - a. Create
 - b. Createln
 - c. Readln
 - d. Writeln
 - e. Write
3. Kode pascal yang digunakan untuk mengambil/membaca nilai dari layar dan kursor tetap pada baris input adalah...
 - a. Readln
 - b. Read
 - c. Create
 - d. Createln
 - e. Ln

4. Kode pascal yang digunakan untuk mengambil/membaca nilai dari layar dan kursor tetap pada baris input dan kursor turun pada baris berikutnya adalah...
 - a. Read
 - b. Readln
 - c. Createln
 - d. Write
 - e. Writeln
5. Untuk menampilkan kata Nama Saya maka kode yang benar adalah...
 - a. Read("Nama Saya");
 - b. Write(Nama Saya);
 - c. Write('Nama Saya');
 - d. Writeln(Nama Saya).
 - e. Readln(Nama Saya)

B. Soal Essay

Berikut program untuk soal nomer 1- 5.

```

Program hitung_nilai;
Uses crt;
Var jumlah_benar, nilai : integer;
Begin
  1. ....
  2. ....
  3. ....
  4. ....
  5. ....
End.

```

Program diatas digunakan untuk menghitung nilai ujian pilihan ganda, dimana nilai diperoleh dari jumlah benar dibagi banyaknya soal.

1. Tuliskan kode untuk menampilkan Judul Program === NILAI UJIAN MTK ===
2. Tuliskan kode untuk menampilkan perintah meminta input jumlah soal yang dijawab benar.
3. Tuliskan kode untuk menyimpan jumlah soal yang dijawab benar dalam variable jumlah_benar
4. Tuliskan kode untuk menghitung nilai
5. Tuliskan kode untuk menampilkan hasil perhitungan.

1.4. KEGIATAN BELAJAR 4 KOMPILASI PROGRAM DAN PERBAIKAN KESALAHAN

Di awal Bab kita sudah mengenal istilah kompilasi program? Masih ingat, kan apa maksud dari mengkompilasi program? Kemudian Pernahkah kalian mendapati error saat menjalankan program? Bagaimana cara memperbaikinya? Untuk memahami dan menemukan jawabannya maka mari lakukan kegiatan belajar berikut ini:

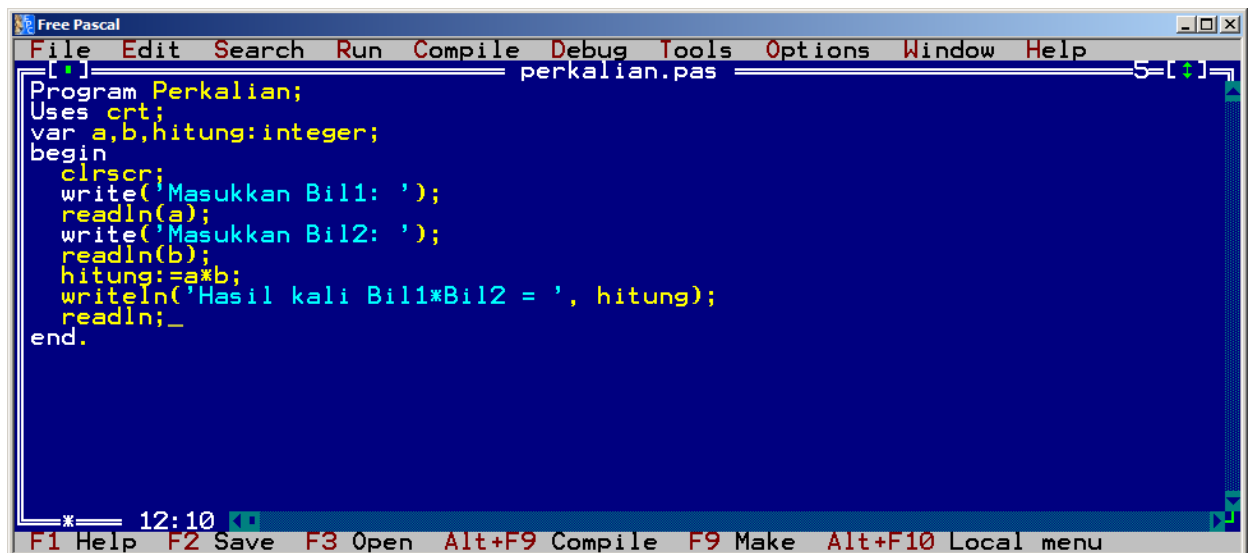
1.4.1. Tujuan Pembelajaran

Siswa memahami cara mengkompilasi program dan memperbaiki kesalahan-kesalahan penulisan kode program dalam Bahasa Pascal.

1.4.2. Aktivitas Kegiatan belajar

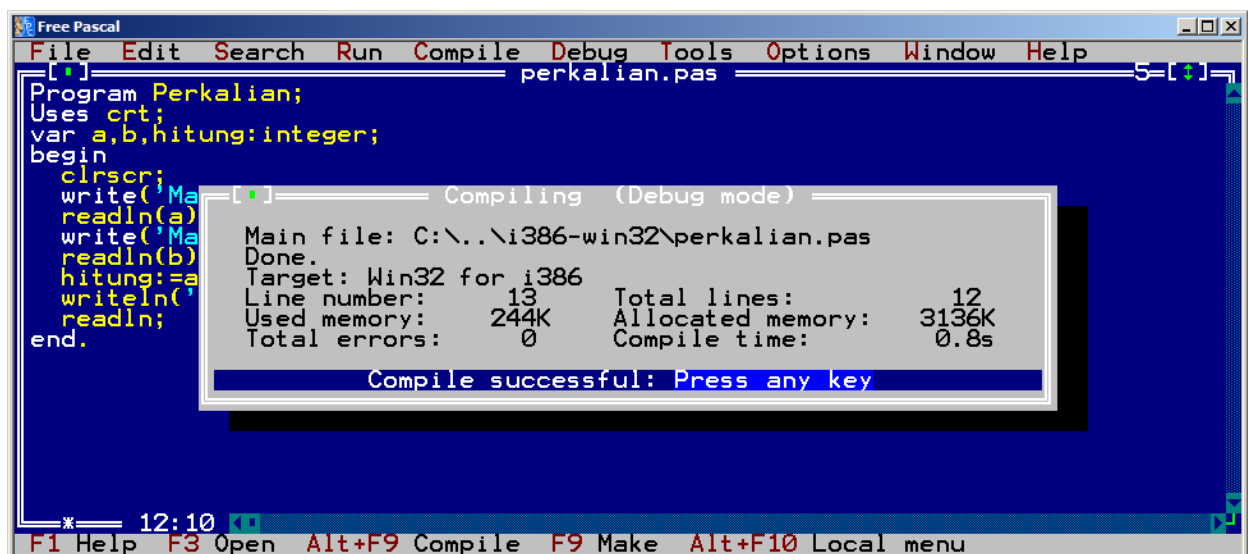
1.4.2.1. Mengamati

1. Suatu program disusun sebagai berikut:



```
Free Pascal
File Edit Search Run Compile Debug Tools Options Window Help
[.] perkalian.pas
Program Perkalian;
Uses crt;
var a,b,hitung:integer;
begin
  clrscr;
  write('Masukkan Bil1: ');
  readln(a);
  write('Masukkan Bil2: ');
  readln(b);
  hitung:=a*b;
  writeln('Hasil kali Bil1*Bil2 = ', hitung);
  readln;_
end.
* 12:10
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu
```

Kemudian ketika ditekan Alt + F9 muncul kotak peringatan sebagai berikut:



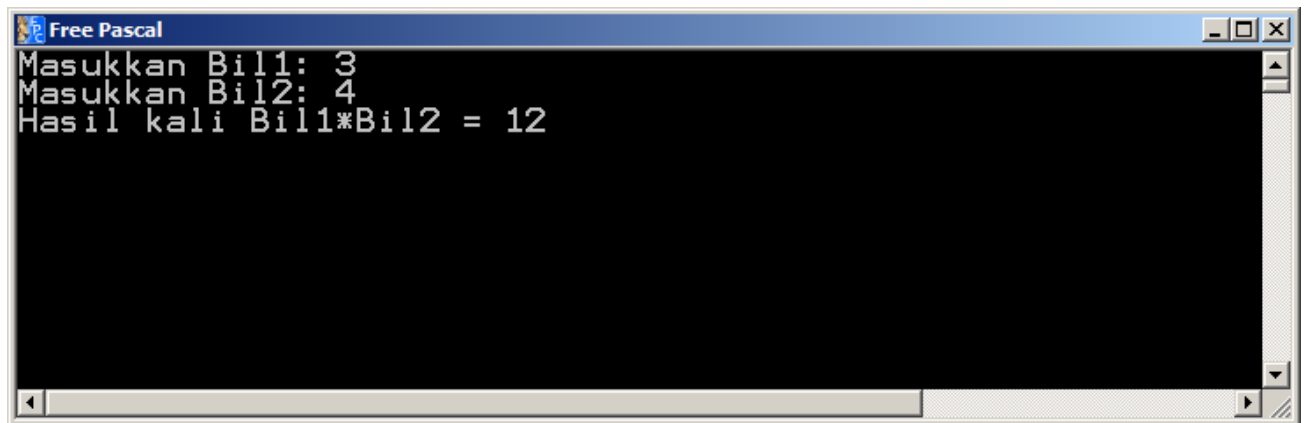
```
Free Pascal
File Edit Search Run Compile Debug Tools Options Window Help
[.] perkalian.pas
Program Perkalian;
Uses crt;
var a,b,hitung:integer;
begin
  clrscr;
  write('Ma
  readln(a)
  write('Ma
  readln(b)
  hitung:=a
  writeln('
  readln;
end.
* 12:10
F1 Help F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu
```

Compiling (Debug mode)

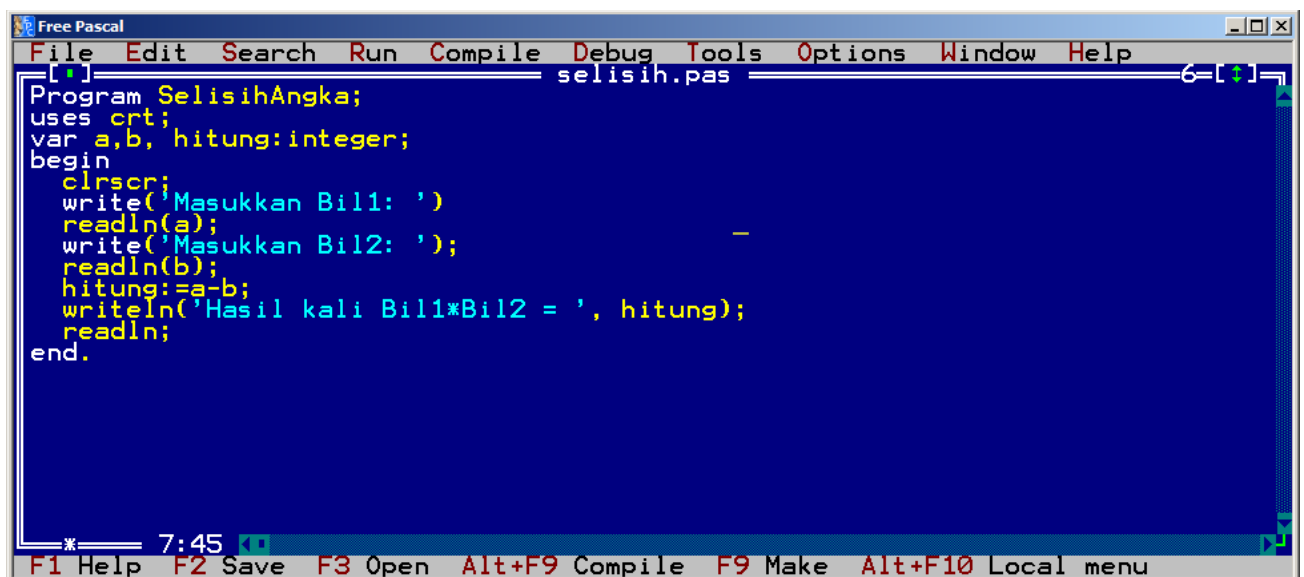
Main file: C:\..\i386-win32\perkalian.pas
Done.
Target: Win32 for i386
Line number: 13 Total lines: 12
Used memory: 244K Allocated memory: 3136K
Total errors: 0 Compile time: 0.8s

Compile successful: Press any key

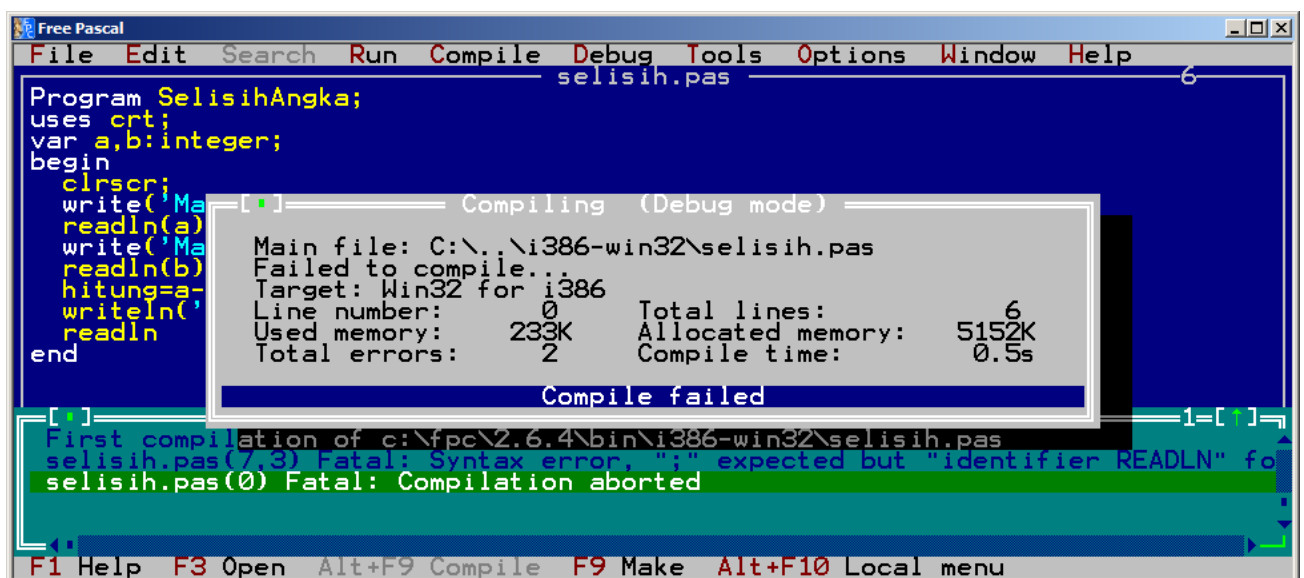
Kemudian ketika ditekan Ctrl + F9 muncul tampilan berikut:



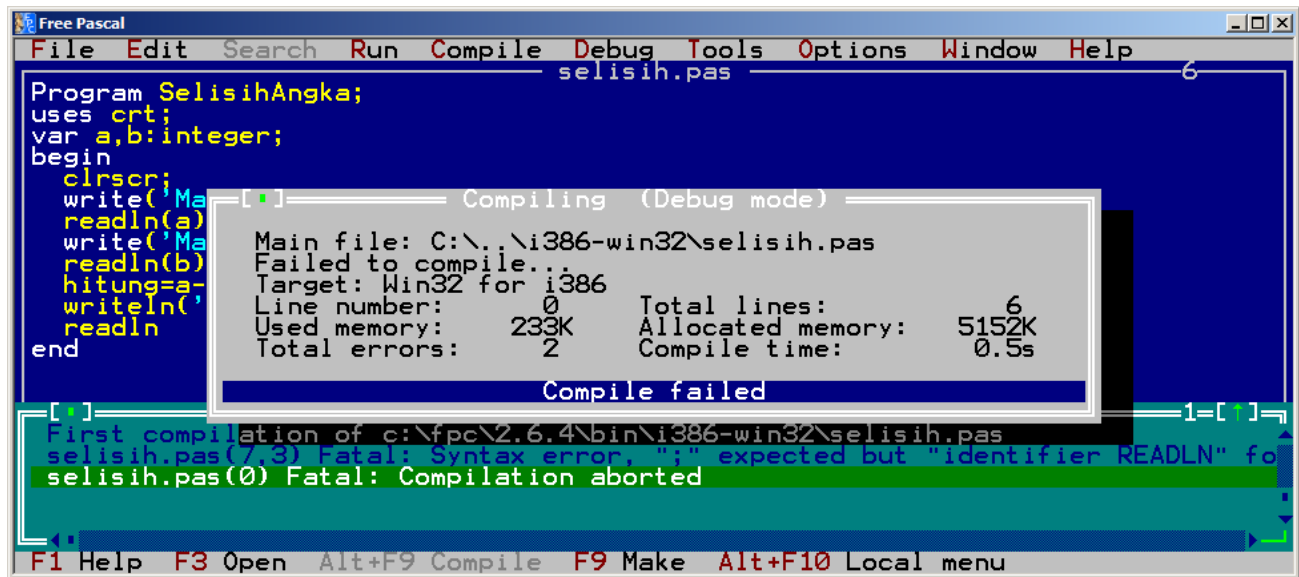
2. Program yang kedua disusun sebagai berikut



Kemudian ketika ditekan Alt + F9 muncul kotak peringatan sebagai berikut:



Kemudian ketika ditekan Ctrl + F9 tampil peringatan yang sama:



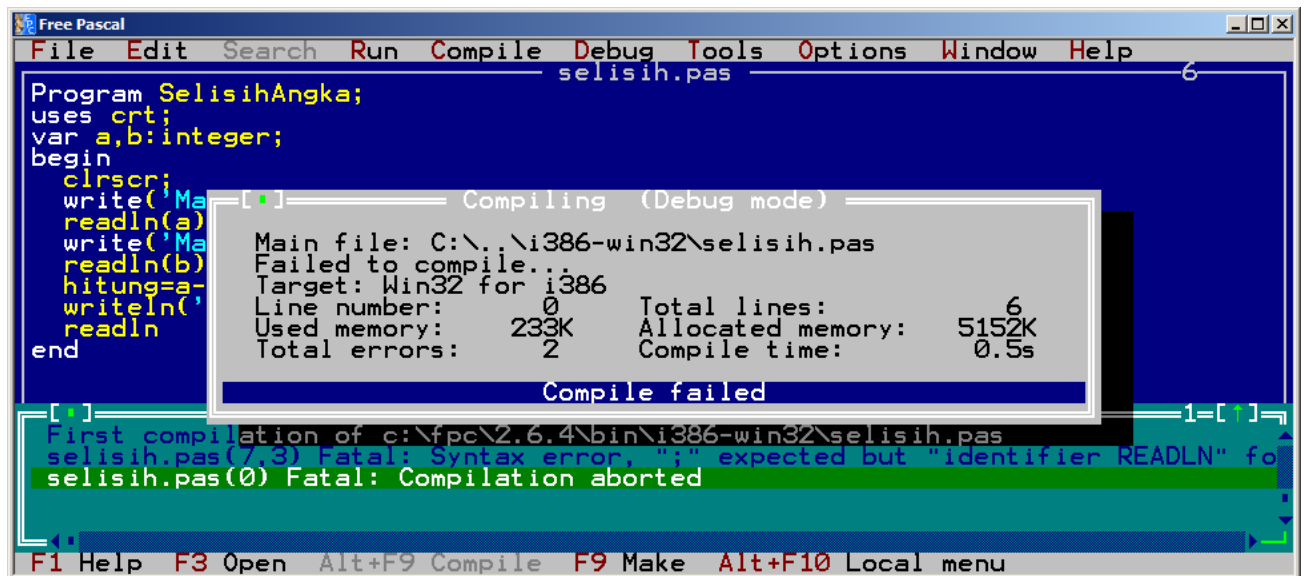
1.4.2.2. Menanya

Dari hasil pengamatan kalian di atas, buatlah pertanyaan seperti contoh berikut.

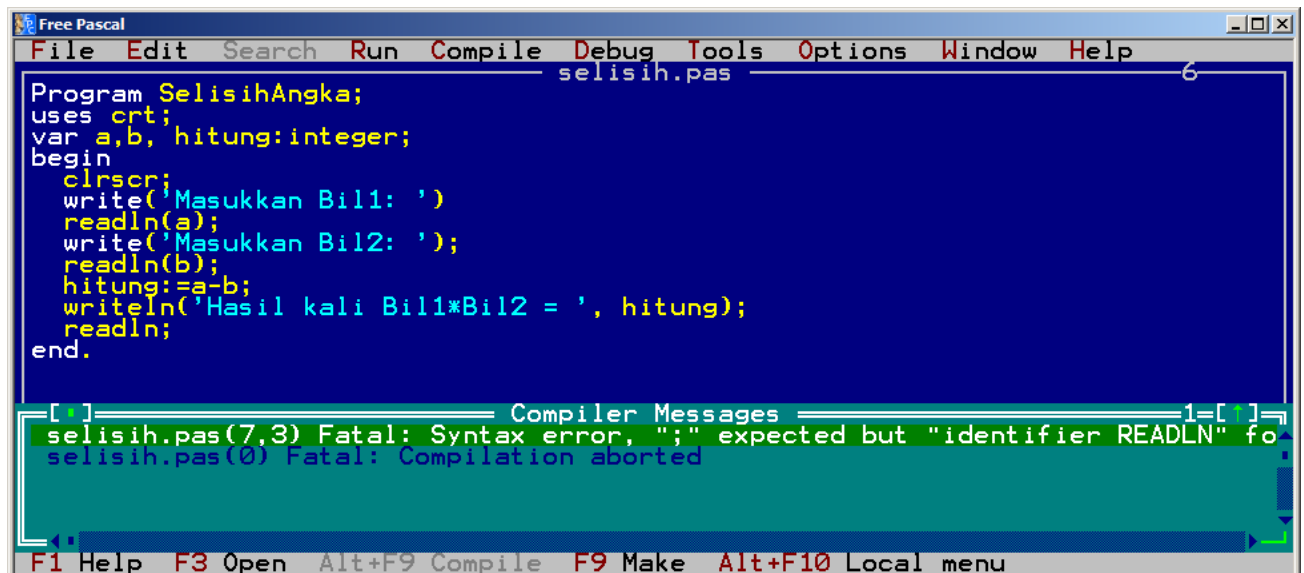
1. Dari duakode program di atas, manakah yang berhasil dikompilasi? Program manakah yang sukses dan dapat dijalankan?
2. Mengapa ada program yang bisa sukses saat di jalankan? Dan mengapa program yang lain tidak bisa untuk dijalankan?
3. Apa fungsi perintah untuk menekan Alt + F9 dan Ctrl + F9?
4. Jika ada pesan Complie failed, apa artinya? Apa bedanya dengan Compile Successfull?
5. Kesalahan-kesalahan apa sajakah yang sering muncul dalam penulisan kode program dengan Bahasa Pascal?
6. Bagaimana memperbaiki kesalahan-kesalahan penulisan dalam bahasa Pascal?

1.4.2.3. Mengumpulkan Informasi/Mencoba

Mari kita analisa bersama, dari kegiatan mengamati diatas diberikan dua buah contoh kode program. Kode pertama berhasil dikompilasi, dan dijalankan programnya dengan baik. Ini berarti tidak ada kesalahan sintaks atau penulisan pada kode program. Sedangkan pada kode program kedua ditemukan sebuah kesalahan dalam penulisan kode program seperti berikut.



Jika ditekan enter maka program Free Pascal akan mengarahkan di mana letak kesalahan, dan apa kesalahan kode tersebut.



Pesan yang muncul adalah:

“selisih.pas(7,3) Fatal:Syntax error, “;” expected but “identifier READLN” for.....”

, artinya kesalahan terjadi pada baris 7, kolom 3 di mana diharapkan ada tanda “;” sebelum perintah readln. Jika ditekan enter maka posisi kursor akan menuju tempat kesalahan dan kita dapat memperbaiki langsung kode program di atas dengan menambahkan tanda “;” sebelum readln seperti berikut.

```

File Edit Search Run Compile Debug Tools Options Window Help
[.] selisih.pas
Program SelisihAngka;
uses crt;
var a,b, hitung:integer;
begin
  clrscr;
  write('Masukkan Bil1: ');
  readln(a);
  write('Masukkan Bil2: ');
  readln(b);
  hitung:=a-b;
  writeln('Hasil kali Bil1*Bil2 = ', hitung);
  readln;
end.
7:4
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

```

Tekan Alt+F9 hasilnya adalah sebagai berikut.

```

File Edit Search Run Compile Debug Tools Options Window Help
[.] selisih.pas
Program SelisihAngka;
uses crt;
var a,b, hitung:integer;
begin
  clrscr;
  write('Masukkan Bil1: ');
  readln(a);
  write('Masukkan Bil2: ');
  readln(b);
  hitung:=a-b;
  writeln('Hasil kali Bil1*Bil2 = ', hitung);
  readln;
end.
7:4
F1 Help F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

```

Compiling (Debug mode)

Main file: C:\..\i386-win32\selisih.pas
 Done.
 Target: Win32 for i386
 Line number: 13 Total lines: 12
 Used memory: 309K Allocated memory: 8864K
 Total errors: 0 Compile time: 0.9s

Compile successful: Press any key

Meskipun sudah benar, tapi peletakan tanda “,” tadi dapat diperbaiki, dengan meletakkan tanda “,” pada akhir baris di atasnya, karena tanda tersebut adalah akhir dari pernyataan pada baris di atasnya.

```

File Edit Search Run Compile Debug Tools Options Window Help
[.] selisih.pas
Program SelisihAngka;
uses crt;
var a,b, hitung:integer;
begin
  clrscr;
  write('Masukkan Bil1: ');
  readln(a);
  write('Masukkan Bil2: ');
  readln(b);
  hitung:=a-b;
  writeln('Hasil kali Bil1*Bil2 = ', hitung);
  readln;
end.
7:3
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

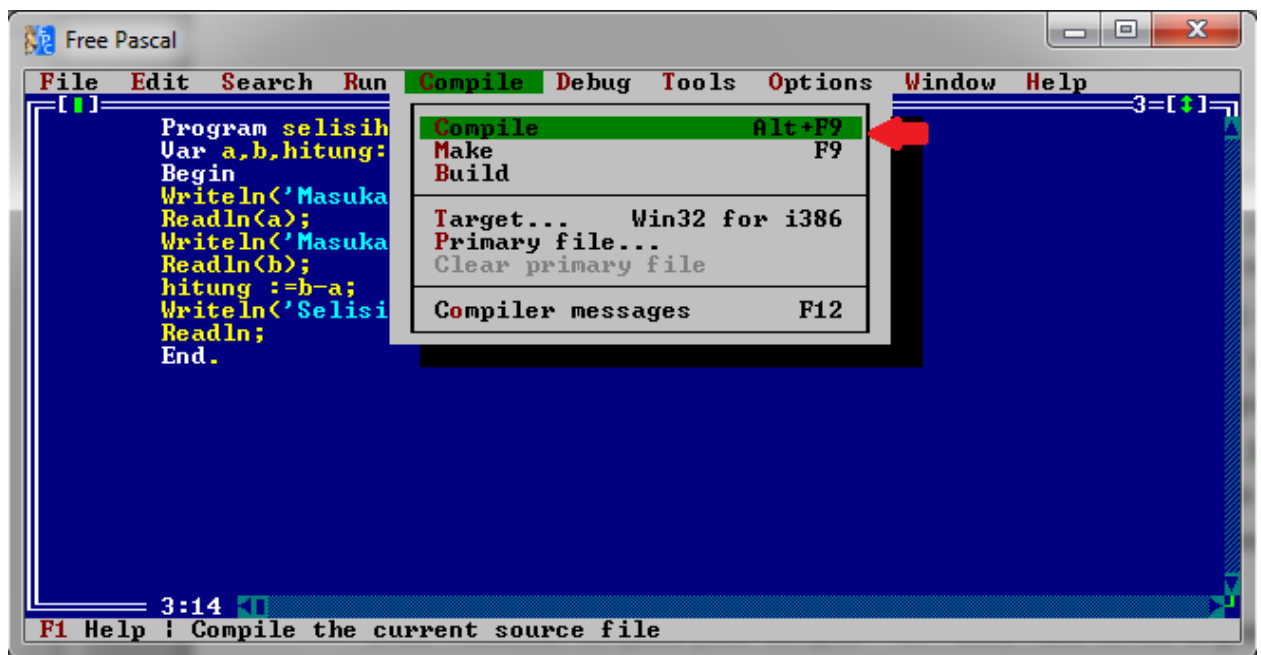
```

Kompilasi Program

Pada kegiatan belajar sebelumnya, telah dijelaskan sekilas mengenai apa itu kompilasi. Yaitu proses menerjemahkan kode program dalam bahasa pemrograman, ke bahasa mesin yang sehingga dapat dimengerti dan dijalankan oleh komputer. Proses kompilasi sendiri dibantu oleh **translator** yaitu program yang mengambil inputan sebuah program yang ditulis pada satu bahasa program ke bahasa yang lain. Pada bahasa tingkat tinggi termasuk di dalamnya bahasa Pascal translatornya adalah **Compiler**, di mana salah satunya adalah Free Pascal. Dalam proses kompilasi, dilakukan proses pengecekan untuk mengetahui kebenaran dalam hal penulisan instruks, apakah sudah sesuai kaidah dan struktur penulisan Bahasa Pemrogrman tersebut atau belum. Jika belum, maka proses kompilasi tidak akan diteruskan hingga benar.

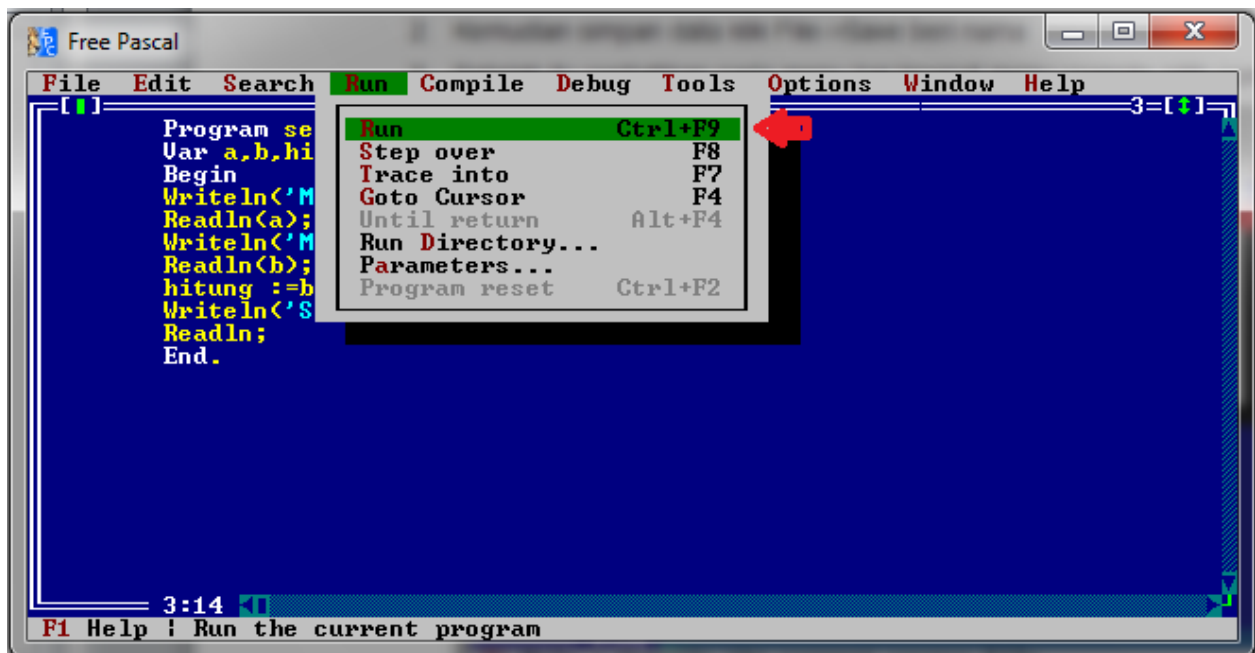
Untuk melakukan kompilasi pada Compiler Free Pascal maka berikut langkah-langkahnya:

1. Membuat File baru klik File->New File
2. Kemudian simpan data klik File->Save beri nama
3. Setelah itu perhatikan pada menu bar terapat menu compile pilih compile.



Atau juga bisa menekan Alt + F9

4. Kemudian untuk menjalankan program maka pada menu bar pilih menu Run atau dengan menekan Ctrl + F9



Namun demikian meskipun dilihat dari kaidah (sintaks) sudah benar belumdum berarti program yang dihasilkan sudah benar. Perlu dilihat juga apakah output yang dihasilkan sudah sesuai dengan yang diharapkan atau belum. Jika belum bisa jadi terjadi kesalahan di luar sintaks, seperti kesalahan logika (*logical error*) atau *run-time error*.

Memperbaiki Kesalahan

Dari penjelasan di atas ada beberapa tipe kesalahan yang mungkin bisa terjadi. Secara garis besar, kita dapat membedakan jenis kesalahan-kesalahan dalam pembuatan program sebagai berikut.

1. Syntax Error

Jenis eror ini adalah jenis yang paling sering dijumpai, erornya program dikarenakan adanya penulisan perintah yang diketik tidak sesuai dengan aturan bahasa pemrograman. Contohnya penulisan perintah tidak diakhiri dengan semicolon (;), lupa menyebutkan tipe data pada variabel, salah dalam penulisan identifer, salah penulisan dalam operator-operator perhitungan, pemberian tanda titik atau tanda petik, dan sebagainya. Meskipun sering muncul eror jenis ini pada umumnya mudah untuk diatasi karena *Compiler Message* akan mendeteksi kesalahan pada baris ke berapa, biasanya kesalahan penulisan ada pada baris tersebut dan atau pada baris sebelum/sesudahnya.

2. Run-Time Error

Runtime Error terjadi manakala program menuliskan kode untuk sesuatu yang tidak memungkinkan. Misalnya mengakses file yang tidak ada, terjadi kesalahan alokasi memori, atau bisa juga bukan karena penulisan program tapi karena tidak sengaja menghapus file penting aplikasi. Untuk eror jenis ini dapat dilakukan pencegahan dengan memberikan validasi

atau penanganan error (*error handling/exception*) sehingga kesalahan tidak berakibat berhentinya program di tengah jalan.

3. Logical Error

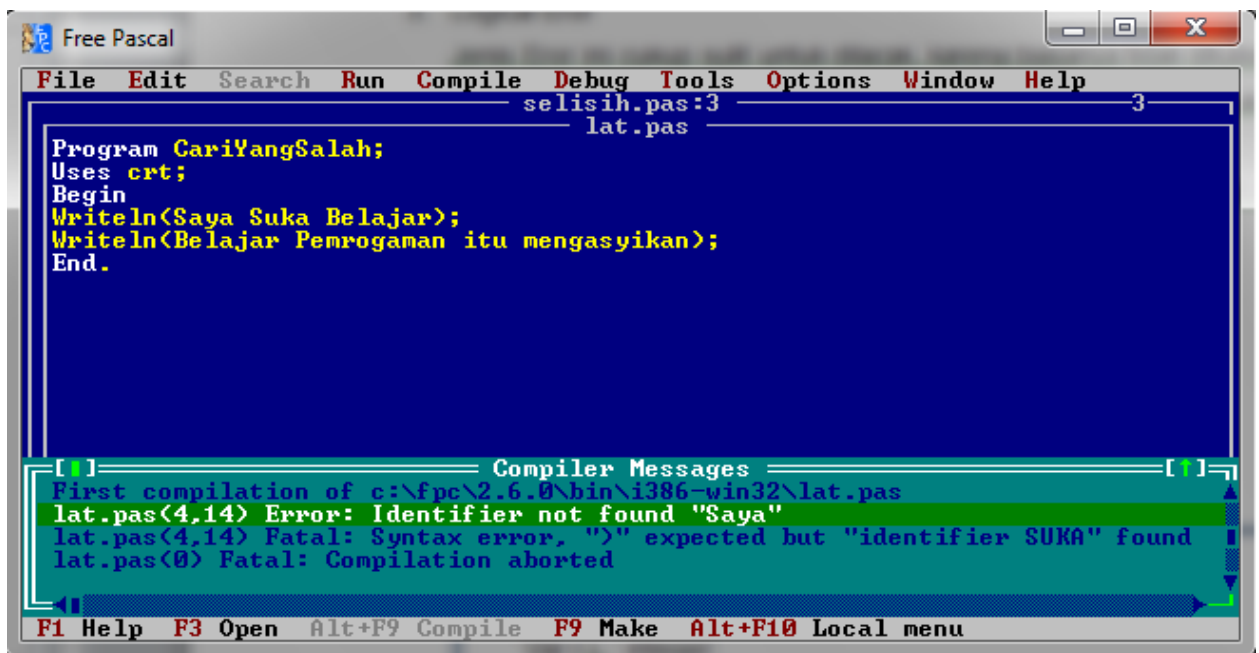
Jenis Error ini cukup sulit untuk dilacak, karena biasanya tidak dituliskan kesalahan pada compiler message, namun saat program dijalankan belum sesuai dengan yang kita harapkan. Untuk mengatasinya maka perlu dilakukan pengecekan secara runtut alur kerja program yang berkaitan.

Pada pembahasan awal ini kita hanya fokus pada syntax error saja, sedangkan untuk jenis error 2 dan 3 akan di bahas pada Bab 5.

Untuk memperjelas pemahaman kalian mengenai cara memperbaiki kesalahan, khususnya kesalahan sintaks dalam bahasa Pascal, kerjakan LKS berikut.

LEMBAR KERJA SISWA

1. Berikut ini adalah hasil dari kompilasi suatu program



- Apa saja kesalahan yang ada dalam program tersebut?
- Bagaimana cara memperbaikinya? Lakukan pengecekan menggunakan Free Pascal

2. Salinlah Program berikut kedalam Free Pascal

```
Program Luas PersegiPanjang;
var Luas, panjang, lebar= integer
Begin
    Clrscr;
    Write ('Masukan panjang persegi panjang = ');
    Readln(panjang);
```

```

Write (Masukan lebar persegi panjang = );
Readln(lebar);
L= panjang x lebar;
Writeln ('Luas lingkarannya adalah='Luas);
Readln;
End

```

- Tuliskan semua kesalahan penulisan pada program diatas? Tuliskan juga pesan yang muncul untuk setiap kesalahan.
- Bagaimana cara memperbaiki kesalahan tersebut?

Jawab:

-
.....
.....
.....
.....
.....
.....
.....
.....
-
.....
.....

1.4.2.4. Mengasosiasikan

Dari hasil pekerjaan LKS kalin, tuliskan kesalahan-kesalahan sintaks yang sering muncul dalam penulisan kode progam menggunakan Bahasa Pascal.

No	Jenis Kesalahan Sintaks
1
2
3
4
5

6
7
8

1.4.2.5. Mengkomunikasi

Presentasikan hasil pekerjaan LKS dan kesimpulan kalian di depan kelas.

1.4.3. Rangkuman

1. Proses kompilasi adalah proses menerjemahkan bahasa pemrograman ke bahasa mesin sehingga dapat dijalankan oleh komputer.
2. Jenis kesalahan yang sering muncul:
 - *Syntax Error*
 - *Run-Time Error*
 - *Logical Error*
3. Cara memperbaiki kesalahan adalah dengan meneliti kembali sintaks penulisan maupun alur algoritma penyelesaian program.

1.4.4. Tugas

Cari kesalahan-kesalahan sintaks lain yang bisa muncul pada penulisan algoritma, minimal 3 macam beserta contohnya.

1.4.5. Uji Kompetensi

Soal Essay

Jawablah dengan benar pertanyaan berikut ini.

1. Apa yang dimaksud dengan mengkompilasi program?
2. Bagaimana cara mengkompilasi program menggunakan Free Pascal?
3. Sebutkan perbedaan antara proses compile dan running?
4. Bagaimana cara me_Running program pada Free Pascal?
5. Jelaskan penyebab jenis kesalahan Syntac Eror !
6. Bagaimana cara memperbaiki jenis kesalahan Syntac Eror ?
7. Jelaskan penyebab jenis kesalahan Runtime Eror !
8. Bagaimana cara memperbaiki jenis kesalahan Runtime Eror ?
9. Jelaskan Penyebab jenis kesalahan Logical Eror !
10. Bagaimana cara memperbaiki jenis kesalahan Logical Eror ?

BAB 2

VARIABEL, TIPE DATA, KONSTANTA, OPERATOR, DAN EKSPRESI

2.1. KEGIATAN BELAJAR 1 VARIABEL, TIPE DATA DAN KONSTANTA

Dalam pembahasan di semester satu, kalian sudah mengenal apa itu variabel dan tipe data, bukan? Bagaimana di Bahasa Pascal? Hal ini tidak jauh berbeda, hanya sintaks atau penulisan saja yang harus tepat sesuai dengan kaidah pemrograman Bahasa Pascal. Pada pembelajaran ini pembahasan mengenai variabel dan tipe data akan lebih jelas.

2.1.1. Tujuan Pembelajaran

1. Siswa memahami penulisan variabel, tipe data dan konstanta dalam Bahasa Pascal
2. Siswa mampu menentukan variabel, tipe data dan konstanta dalam permasalahan pemrograman

2.1.2. Aktivitas kegiatan belajar

2.1.2.1. Mengamati

Coba kalian amati dan bandingkan beberapa kode program berikut :

Kasus 1

```
program biodata;
var nama, alamat : string[30];
    umur:integer;
begin
    write ('Masukkan nama Anda: ');
    readln(nama);
    write('Masukkan alamat : ');
    readln(alamat);
    write('Masukkan Umur Anda:');
    readln(umur);
    writeln('Nama Anda adalah : ', nama);
    writeln('Alamat Anda adalah : ', alamat);
    writeln('Umur Anda adalah ', umur);
    readln;
```

end.

Kasus 2

```
program VolumeTabung;  
const Phi=3.14;  
var r, t, V;  
begin  
    readln(r);  
    readln(t);  
    V:=Phi*r*r*t;  
    writeln('Volume Tabung adalah ',V);  
    readln;  
end.
```

Kasus 3

```
program UAN;  
var NilaiMat, Nilailnd, Nilailngg:integer;  
    rerata:real;  
    stsLulus:booean;  
begin  
    readln(NilaiMat);  
    readln(Nilailnd);  
    readln(Nilailngg);  
    rerata:=(NilaiMat_Nilailnd+Nilialngg)/3;  
    stsLulus:=(rerata>=60);  
    writeln('Rata-rata Nilia UAN adalah ', rerata);  
    writeln('Status Kelulusan adalah ', stsLulus);  
    readln;  
end.
```

2.1.2.2. Menanya

Dari ketiga kode program diatas, buatlah pertanyaan seperti berikut.

1. Apa tujuan dari setiap program di atas?
2. Bagian manakah yang mendeklarasikan variabel?
3. Variabel apa saja yang ada/diperlukan tiap program? Bagaimana cara menentukannya?
4. Apa itu konstanta, bagian manakah yang mendeklarasikan konstanta?
5. Sebutkan tipe data dari tiap variabel pada source code di atas! Mengapa harus berbeda-beda?

2.1.2.3. Mengumpulkan Informasi/Mencoba

Variabel

Kalian masih ingat apakah arti variabel? Iya, variabel adalah suatu nama atau simbol untuk menyimpan sebuah nilai. Atau lebih tepatnya variabel adalah sebuah nama yang diberikan oleh programmer yang berfungsi sebagai tempat penyimpanan data, sehingga dapat dimanipulasi oleh program. Seperti pada contoh pada source code di atas, Program Biodata diperlukan variabel **nama**, **alamat** dan **umur**. Pada program menghitung volume diperlukan variabel **r**, **t**, **V** yang mewakili jari-jari, tinggi, dan volume tabung. Sedangkan pada Program UAN diperlukan variabel **NilaiMat**, **NilaiIngg**, **NilaiInd**, **rerata**, dan **stsLulus**.

Variabel ini ditentukan sesuai dengan kebutuhan dari solusi permasalahan. Karena Volume tabung adalah $\text{Volume} = \pi r^2 t$, maka tentu kita membutuhkan variabel untuk mewakili Volume, jari-jari dan tinggi, dan satu lagi adalah π sebagai konstanta. Jadi ada 3 variabel dan 1 konstanta.

Demikian juga dengan kelulusan, karena tergantung dari nilai ketiga mata pelajaran, maka variabel-variabel yang dibentuk pun menyesuaikan.

Jenis nilai yang disimpan variabel tentu berbeda-beda sesuai dengan kebutuhan, ingat kembali materi pada semester 1. Variabel **nama** dan **alamat** tentu menyimpan nilai berupa susunan huruf, atau yang dulu dikenalkan sebagai *string* (dalam Pascal disebut juga *string*), Volume, **r**, dan **t** menyimpan nilai bilangan real, sedangkan variabel **nilaiMat**, **nilaiInd** menyimpan nilai bilangan bulat (mungkin juga bilangan real, sesuai kebutuhan). Jenis-jenis data inilah yang disebut sebagai TIPE DATA. Oleh karena itu dalam mendeklarasikan variabel harus didefinisikan juga tipe data yang akan digunakan. Hal ini sekaligus digunakan program untuk menentukan besarnya memori sebagai tempat penyimpanan variabel tersebut.

Sintaks Penulisan Variabel

Berikut ini adalah sintaks pendeklarasian variabel.

```
var nama_variabel:tipe_data;
```

Jika ada beberapa variabel dengan tipe data sama, sintaksnya adalah:

```
var namaVar1, namaVar2, namaVar3:tipe_data;
```

Jika ada beberapa variabel dengan tipe data berbeda-beda sintaksnya adalah:

```
var namaVar1:tipe_data1;
    namavar2:tipe_data2;
    namaVar3:tipe_data3;
```

Pemberian Nama Variabel

Pemberian nama variabel sebagaimana pemberian nama identifier seperti yang dijelaskan pada Bab 1, sebagai berikut.

1. Nama identifier harus dimulai dengan karakter huruf alfabet a-z atau A-Z.
2. Setelah karakter pertama, dapat dilanjutkan dengan karakter afanumerik (angka maupun huruf) dan underscore (_).
3. Dalam penulisan identifier tidak boleh menggunakan karakter-karakter berikut ini.
~ ! @ # \$ % ^ & * () + ` - = { } [] : " ' < > ? , . / |
4. Tidak boleh menuliskan identifier dengan nama yang sama pada kata kunci dari Pascal, seperti
and, array, begin, case, const, div, do, downto, else, end, file, for, forward, function, goto, if, in, label, mod, nil, not, of, or, packed, procedure, program, record, repeat, set, then, to, type, until, var, while, with
5. Jika ingin tetap digunakan, maka kata tersebut harus dirangkai dengan kata/karakter yang lain.
Misal program_if, begin1.

Aturan tersebut juga berlaku untuk pemberian nama konstanta, tipe data buatan, dan sub program (*procedure* dan *function*).

Berikut ini contoh pemberian nama variabel yang benar.

Jari_jari, jari2, Volume_Balok, LuasPersegi, Luas_segi3, record3.

Berikut ini contoh nama variabel yang salah

Jari^2, jari-jari, Volume Balok, p*l, 2jari, alam@t

Konstanta

Variabel dapat diturunkan dari kebutuhan input, output, dan segala sesuatu yang dilibatkan dalam pengolahan input menjadi output. Pemberian nilai variabel dapat dilakukan melalui perintah baca (input) maupun dari pernyataan pemberian nilai (assignment), dengan demikian nilai variabel dapat berubah-ubah selama proses dijalankan. Jika diperlukan suatu nilai yang tetap, maka dibutuhkan konstanta. Dalam hal ini, sebagai contoh karena besaran nilai π adalah tetap, tidak berubah selama perhitungan maka π yang diwakili dengan variabel Phi dideklarasikan sebagai

konstanta. Jadi konstanta adalah variabel yang mempunyai nilai tunggal, dimana nilai tunggal tersebut tidak dapat berubah ketika program sudah dieksekusi. Sebagaimana juga variabel konstanta mempunyai tipe data tertentu yang menyesuaikan nilai yang diberikan dalam deklarasi konstanta.

Sintaks deklarasi penggunaan konstanta adalah sebagai berikut :

```
ConstNama_konstanta = nilai_konstanta;
```

Contoh deklarasi konstanta pada beberapa tipe :

```
ConstA = 100;
```

```
ConstPi = 3.14;
```

```
ConstOperator = '+';
```

```
ConstPesan = 'Selamat Datang';
```

Tipe Data

Tipe data merupakan klasifikasi jenis data dari variabel atau object yang terdapat dalam program computer. Jenis data yang digunakan dalam variabel merupakan bagian penting dalam pengkodean pemrograman computer. Ketepatan pemilihan data pada variabel atau konstanta akan sangat menentukan pemakaian sumber daya komputer (terutama memori komputer). Sehingga dalam pemilihan tipe data yang akan digunakan dalam variabel dan konstanta akan mempengaruhi efisiensi dan kinerja dari program tersebut.

Secara garis besar tipe data dalam Pascal dibagi menjadi 3 kategori, yaitu tipe data sederhana (tunggal), Tipe data pointer dan tipe data terstruktur (structured data). Masing-masing kategori dapat dibagi lagi menjadi beberapa jenis lagi seperti pada Gambar 2.1.

1. Tipe Data Sederhana

Tipe data sederhana merupakan tipe data paling dasar yang terdapat dalam pascal. Tipe data ini mewakili satu objek yaitu hanya satu nilai yang dapat disimpan dalam tipe data ini sehingga disebut juga tipe data tunggal. Tipe data sederhana dibagi menjadi dua yaitu tipe data standar dan User Defined.

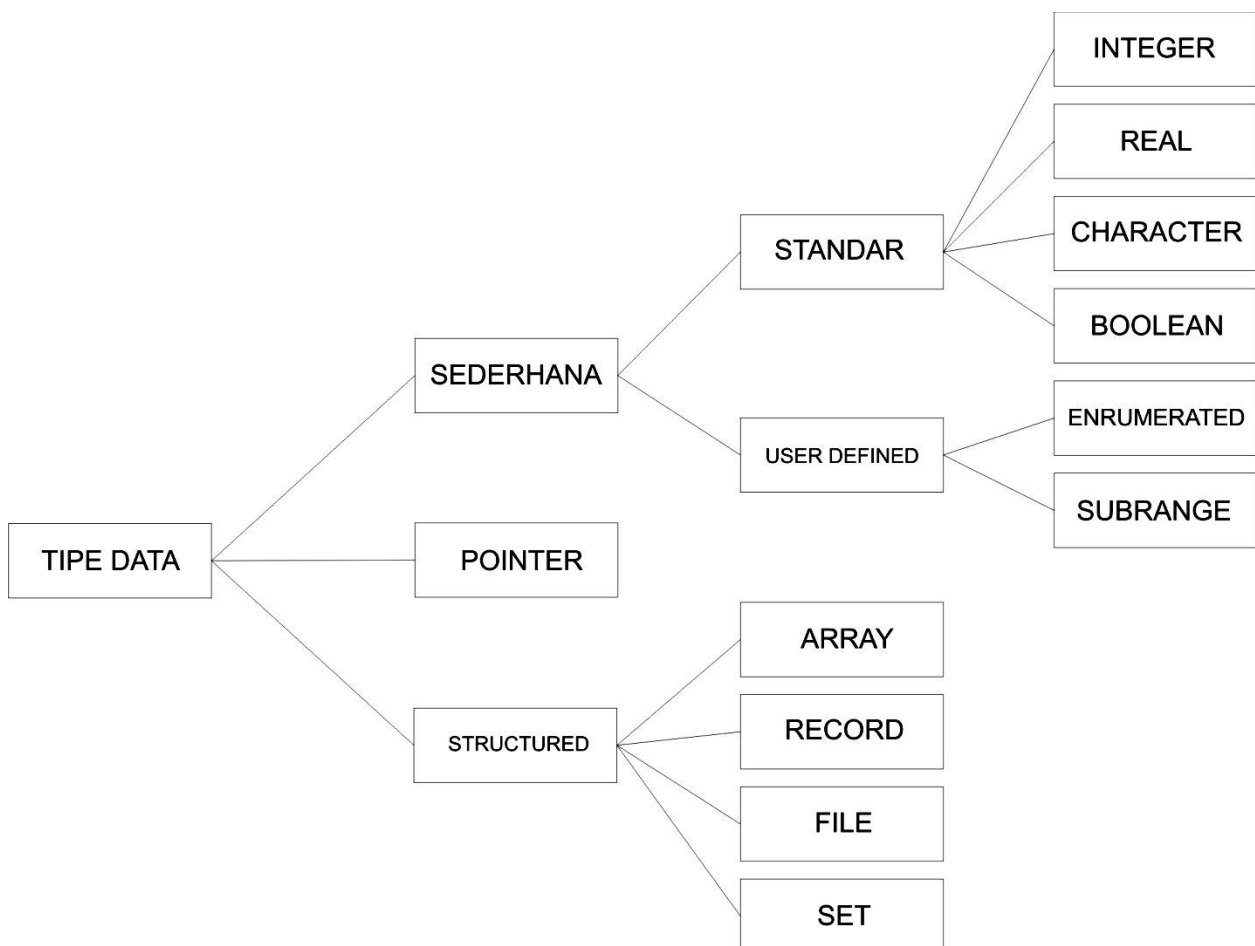
- a. Standar, merupakan tipe data yang umum dijumpai dalam berbagai bahasa pemrograman. Tipe data standar meliputi integer, real, character dan boolean.

1) Integer

Tipe data integer merupakan tipe data berupa bilangan bulat. Tipe data ini cocok untuk variabel-variabel seperti: jumlah anak, bulan, tahun, dan sebagainya. Berat badan, tinggi, suhu, panjang, lebar, nominal uang dapat juga dimasukkan dalam tipe data ini, meskipun dapat juga dimasukkan ke dalam tipe data lain seperti bilangan real.

Dalam bahasa Pemrograman, terkait dengan penggunaan memori untuk menyimpan variabel, maka tipe data integer, dibagi menjadi beberapa kelas sesuai dengan kebutuhan. Sebagai contoh untuk variabel jumlah anak, maka kita dapat memperkirakan batasan berapa maksimalnya. Jumlah anak biasanya tidak leboh dari 10, maka dipilih tipe data integer dengan jangkauan yang kecil, yaitu byte. Sedangkan untuk nominal uang, karena tidak mungkin bernilai negatif, dan jangkauannya yang tinggi (bisa mencapai ratusan juta atau milyar) bisa digunakan tipe data cardinal, dan sebagainya. Hal ini penting untuk diperhatikan agar program kita lebih efisien.

Tabel jenis-jenis tipe data integer dapat dilihat pada Tabel 2.1.



Gambar 2.1. Macam-macam tipe data dalam Pascal

Tabel 2.1. Jenis-jenis tipe data integer yang terdapat dalam pascal.

Tipe	Batas Bawah	Batas Atas
Integer	-32768	32767
Cardinal	0	4294967295
Smallint	-128	127
Word	0	65535
Longint	-2147483648	2147483647
Byte	0	255

Penulisan deklarasi dengan tipe data integer adalah sebagai berikut :

```
var nama_variabel : integer;
atau
var nama_variabel : byte;
atau
var nama_var1, nama_var2,nama_var3:integer;
```

2) Real

Tipe data real merupakan tipe data yang berisi bilangan yang bisa memuat angka di belakang koma. Berbeda dengan integer yang hanya menyimpan bilangan bulat. Dalam tipe data real nilai yang tersimpan dapat berupa bilangan desimal. Untuk mengidentifikasi variabel dengan tipe data real, dapat dilihat dari operasi yang dibutuhkan dalam proses perhitungan. Jika variabel tersebut diperoleh dari operasi dari bilangan lain bertipe data real maka tipe datanya juga real. Sebagai contoh Luas lingkaran, volume tabung dan kerucut termasuk bertipe data real karena melibatkan perkalian dengan $\Phi \approx 3.14$ sehingga hasilnya pasti ada angka di belakang koma. Selain itu adanya operasi bagi (‘/’) juga menghasilkan bilangan real. Sebagai contoh Luas segitiga = $(\text{alas} \times \text{tinggi}) / 2$ termasuk mempunyai tipe data real, demikian juga rata-rata, suhu konversi dalam Reamur dan Fahrenheit, karena melibatkan operasi pembagian.

Selain berdasarkan operasi, variabel dengan tipe data bilangan real tergantung kebutuhan. Misalkan menghitung tinggi badan. Kita bisa melakukan pembulatan tanpa koma sehingga tinggi badan bisa kita masukkan tipe data integer. Jika ingin lebih teliti, menggunakan koma, bisa menggunakan tipe data real.

Sebagaimana pada tipe data integer, tipe data bilangan real ini dapat dibagi menjadi beberapa kelas sesuai dengan jangkauannya seperti pada Tabel 2.2.

Tabel 2.2. Jenis-jenis tipe data real berdasarkan jangkauannya

Tipe	Batas Bawah	Batas Atas
Real	2.9×10^{-39}	1.7×10^{38}
Single	1.5×10^{-45}	3.4×10^{38}
Double	5.0×10^{-324}	1.7×10^{308}
Extended	3.4×10^{-4932}	1.1×10^{4932}

Sedangkan penulisan deklarasi variabel dengan tipe data real adalah sebagai berikut.

```
var nama_variabel : real;
atau
var nama_variabel : double;
atau
var nama_var1, nama_var2,nama_var3:real;
```

3) *Character*

Tipe data *character* merupakan tipe data yang menyimpan karakter yang berupa angka maupun huruf. Tipe data character dalam pascal meliputi *char* dan *string*. Tipe data *char* hanya meyimpan 1 karakter saja. Variabel yang cocok untuk tipe data karakter adalah variabel yang nilainya hanya terdiri dari satu karakter saja, contoh: JenisKelamin (nilainya bisa diwakili 'L' untuk laki-laki atau 'P' untuk perempuan), Pilihan yang bernilai iya dan tidak (bisa diwakili dengan 'Y' atau 'N'), dan lain-lain.

Tipe data *string* merupakan susunan dari karakter-karakter dengan batas maksimum 255 karakter, tipe data *string* digunakan untuk menyimpan data text, misal nama, alamat dan sebagainya. Penulisan deklarasi variaebel dengan tipe data *character* dan *string* adalah sebagai berikut :

```
var nama_variabel : char;
atau
var nama_variabel : string[batasMaksimal];
```

Keterangan: batasMaksimal diganti dengan banyaknya karakter yang maksimal dapat ditampung oleh variabel tersebut.

Contoh:

```
var Gender:char;
var pilihan:char;
var nama:string[20];
```

```
var alamat:string [50];
```

4) Boolean

Tipe data Boolean merupakan tipe data yang hanya mempunyai dua nilai yaitu *true* atau *false*. Tipe data ini biasanya digunakan variabel yang hanya mempunyai nilai benar atau salah saja. Misalkan statusLulus, bisa bernilai True atau False; remidi, bisa bernilai true atau false, Menikah, bisa bernilai true atau false, pdan sebagainya.

Penulisan deklarasi variabel dengan tipe data ini adalah ebagai berikut.

```
Var nama_variabel:boolean;
```

b. User defined

Tipe data ini merupakan tipe data yang dibuat oleh sang *programmer*. Jadi setiap *programmer* dapat membuat tipe data baru yang dibutuhkannya. *User defined* meliputi *enumerated* dan *subrange*.

1) Enumerated

Tipe data enumerated merupakan tipe data yang didefinisikan oleh *programmer*. Hal ini memungkinkan programmer untuk membuat tipe datanya sendiri, yang terdiri dari kumpulan symbol. Contoh penulisan code program tipe data enumerated adalah sebagai berikut:

```
type hari = (senin,selasa,rabu,kamis,jumat,sabtu,minggu);  
var nama_hari : hari;
```

Dari contoh diatas, baris pertama menunjukkan pembuatan tipe data baru yaitu hari yang berisi nama-nama hari, kemudian di baris kedua menunjukkan deklarasi penggunaan tipe data yang telah dibuat tadi.

2) Subrange

Tipe data subrange merupakan tipe data yang memungkinkan programmer untuk mendefinisikan beberapa nilai dari tipe data tertentu. Misalkan seorang programmer ingin membuat tipe data nilai yang dimana tipe data ini berisi bilangan integer dari 1 sampai 10. Contoh penulisan code program tipe data enumerated adalah sebagai berikut:

```
type nilai = 1..10;  
var nilai_mtk : nilai;
```

Dari contoh diatas, baris pertama merupakan tipe data baru yang berisi bilangan integer dari satu sampai 10. Berikut beberapa contoh lain untuk tipe data subrange :

```
Positif = 1..32500;  
Skor = 0.00..4.00;  
Abjad = 'Z'..'A';  
Range = '1'..9;  
Count = -15..15;
```

2. Tipe Data Pointer

Sama seperti tipe data sederhana, pointer merupakan tipe data yang dapat menyimpan satu nilai saja. Bedanya adalah, sesuai dengan namanya tipe data ini berfungsi menunjuk, yaitu alamat memori yang digunakan suatu variabel dengan tipe data tertentu. Misalkan **a** adalah variabel dengan tipe data integer. Variabel **point** adalah variabel dengan tipe data pointer dari integer. Ketika variabel point diberi nilai dengan alamat a (alamat memori a), maka kita bisa mengubah isi a melalui variabel **point** tersebut dengan langsung mengakses ke memori tanpa melalui variabel a. Tipe data ini sangat berguna untuk membangun suatu program dengan data yang dinamis. Lebih jelasnya akan dipelajari di kelas XI nanti.

3. Tipe Data Terstruktur

Berbeda dengan tipe data sederhana dan pointer, yang hanya dapat menyimpan satu nilai saja, tipe data terstruktur dapat menyimpan beberapa nilai dalam satu variabel. Tipe data terstruktur adalah tipe data yang terdiri dari data-data tunggal, yang diorganisasi oleh suatu tipe data terstruktur. Sebagai contoh, terdapat data mengenai nilai ulangan 20 anak, apakah kita harus membuat 20 variabel untuk menyimpan masing-masing nilai tersebut? Tentu ini akan menjadikan program komputer menjadi tidak efisien. Mungkinkah kita bisa mengumpulkan semua variabel tersebut dalam satu wadah? Iya, bisa yaitu menggunakan tipe data terstruktur. Inilah pentingnya data terstruktur dalam algoritma atau pemrograman. Berikut ini jenis-jenis tipe data terstruktur.

a. Array

Array dapat diartikan sebagai larik, atau barisan. Array digunakan untuk menyimpan banyak data dengan tipe data yang sama. Sebagai contoh, dalam sebuah algoritma diberikan input sejumlah data ulangan anak, misalkan 40 anak. Jika untuk menyimpan data itu salah satu alternatifnya adalah memberikan variabel Nilai1, Nilai2, dan seterusnya, hingga Nilai40. Bagaimana jika 100 anak? Maka cara ini sangatlah tidak efisien. Data seperti ini dapat disimpan dalam satu wadah yang disebut dengan Array. Untuk lebih jelasnya perhatikan ilustrasi berikut :

indeks	1	2	n
nilai(indeks)	nilai(1)	nilai(2)	nilai(n)



Elemen pertama



Elemen terakhir

Berikut deklarasi variabel dengan tipe data array pada Pascal.

```
Typenama_array : Array[nilai array] of tipe_data;
```

Contoh :

```
Typenilai : Array[1..40] of integer;
```

b. *Record*

Jika Array hanya mampu menyimpan sejumlah data yang bertipe sama, **record**, bisa menampung data-data dengan tipe yang berbeda. Sebagai contoh data seorang siswa, bisa terdiri dari Nama, Nomor Induk Siswa, Alamat, Tempat Lahir, Jenis Kelamin, Kelas Umur. Agar data tersebut menjadi satu kesatuan informasi maka perlu suatu wadah untuk menampung tipe data seperti itu, yaitu RECORD. Berikut deklarasi tipe data record dalam pascal :

```
Typenama_record = record
    variabel_1 : tipe_data_1;
    variabel_2 : tipe_data_2;
    ....
    variabel_n : tipe_data_n;
end;
```

Contoh:

```
Typebuku = record
    judul :string[30];
    tahun_terbit = integer;
    pengarang = string[30];
end;
```

c. File

Dalam pemrograman pascal terdapat tipe data file, dimana data input dan output dari suatu program dapat disimpan dalam bentuk file.txt. Selain menyimpan pascal dapat juga membaca dan mengedit file.txt tersebut. Sebagai contoh kita ingin memproses suatu nilai ulangan dari 20 anak yang telah tersimpan dengan format text. Tentunya sangat tidak efisien jika data harus diinput satu persatu. Agar data yang sudah ada dapat terbaca dalam pascal maka kita perlu tipe data file. Penjelasan mengenai penggunaan tipe data ini dibahas pada Buku Pemrograman Dasar Kelas XI.

Berikut ini contoh bagaimana menentukan variabel dan tipe data dalam suatu permasalahan.

Contoh 1.

Budi seorang programmer di SMP Nusantara akan membuat program untuk menampilkan data nilai siswa, dimana data yang akan ditampilkan meliputi : nomor induk siswa, nama siswa, nilai matematika, nilai bahasa Indonesia, nilai bahasa Inggris dan rata-rata nilai. Dimana data yang diinputkan mempunyai peraturan yaitu

- 1) nomor induk siswa terdiri dari 6 karakter,
- 2) nama siswa maksimal 30 karakter,
- 3) nilai matematika, nilai bahasa Indonesia dan nilai bahasa Inggris merupakan bilangan bulat skala 1 – 100,
- 4) rata – rata nilai berupa bilangan desimal dengan dua digit angka di belakang koma, yang diperoleh dari penjumlahan nilai matematika, bahasa Indonesia dan bahasa Inggris dibagi tiga.

Contoh data yang akan diinputkan adalah sebagai berikut :

Nomor induk siswa	: A45789
Nama Siswa	: Slamet Sentosa
Nilai Matematika	: 80
Nilai Bahasa Indonesia	: 76
Nilai Bahasa Inggris	: 70
Rata – rata nilai	: 75.33

Bantulah budi untuk menentukan variabel dan tipe data yang tepat dengan deklarasinya dalam Program!

Jawab:

Untuk menentukan variabel dan tipe data yang tepat, kita harus memperhatikan data apa saja yang akan diinputkan ke dalam program. Pada contoh diatas data yang diinputkan meliputi : nomor induk siswa, nama siswa, nilai matematika, nilai bahasa Indonesia, nilai bahasa inggris dan rata-rata nilai. Kemudian kita perhatikan contoh data yang akan diinputkan. Dengan demikian dapat ditentukan variabel dan tipe data sebagai berikut.

- 1) Coba perhatikan data nomor induk siswa pada contoh di atas. Jika kita perhatikan, data nomor induk siswa terdiri dari 1 huruf dan 5 angka, maka tipe data yang tepat adalah string. Karena string dapat menyimpan data berupa huruf atau karakter dan angka. Sehingga dapat ditentukan nama variabel **NIS** sebagai string dengan panjang maksimal 6 karakter.
- 2) Kemudian perhatikan data nama siswa pada contoh diatas. Data nama siswa terdiri dari huruf, maka tipe data yang tepat adalah string. Karena string dapat menyimpan data berupa huruf atau karakter. Sehingga dapat ditentukan nama variabel **Nama** sebagai string. Panjangnya dapat diperkirakan maksimal 30.
- 3) Selanjutnya, perhatikan data nilai matematika. Data tersebut berupa bilangan bulat maka tipe data yang tepat adalah integer. Jika kita perhatikan, data nilai bahasa Indonesia dan data nilai bahasa inggris mempunyai data yang sama dengan nilai matematika. Maka tipe data yang tepat adalah integer. Karena tipe data integer dapat menyimpan bilangan bulat. Sehingga dapat ditentukan nama variabel Nilai_Mat, Nilai_Blnd, Nilai_Ingggris sebagai integer.
- 4) Berikutnya, perhatikan data rata-rata nilai. Data tersebut di dapat dari penjumlahan nilai matematika, bahasa Indonesia dan bahasa inggris dibagi tiga. Karena ada operasi pembagian, maka tipe data yang tepat adalah real.

Deklarasi variabel dan tipe data

```
Var NIS : string[6];  
Nama: string[30];  
Nilai_Mat : integer;  
Nilai_Blnd : integer;  
Nilai_Ingggris : integer;  
Rerata : real;
```

Pada nomor induk siswa akan menyimpan data dengan maksimal 6 karakter, sedangkan pada nama siswa maksimal 30 karakter. Sehingga, walaupun user menginputkan data lebih dari 30 karakter, maka data yang tersimpan tetap 30 karakter.

LEMBAR KERJA SISWA

- Berikut contoh output dari program yang akan dibuat

Analisalah, tentukan variabel dan jelaskan tipe data yang tepat untuk digunakan pada program tersebut dan deklarasikan tipe data tersebut pada pemrograman pascal?

[illegible]

2. Terdapat suatu program dengan source code berikut ini :

```
Program dataSiswa;  
Var NIS : integer;  
Nama_siswa : string;  
Alamat : char;  
Goldar : string;  
Gaji_ortu : integer;  
Begin  
    Write('Masukkan Nomor induk siswa : ');  
    Readln(nis);  
    Write('Masukkan nama Anda : ');  
    Readln(nama_siswa);  
    Write('Masukkan Alamat Rumah : ');  
    Readln(alamat);  
    Write('Masukkan Golongan Darah Anda : ');  
    Readln(goldar);  
    Write('Masukkan gaji orang tua anda : ');  
    Readln(gaji_ortu);  
    Writeln('NIS : ',nis);  
    Writeln('Nama : ',nama_siswa);  
    Writeln('Alamat : ',alamat);  
    Writeln('Golongan Darah : ',goldar);  
    Writeln('Gaji Orang Tua : ',gaji_ortu);  
    Readln;  
End.
```

Ketika budi melakukan penginputan data nis, program menjadi error dan penginputan data tidak dapat dilanjutkan. Nis budi adalah A314, tentukan penyebab errornya program tersebut?. Amati dan temukan kesalahan sintaks pada kode program di atas. Beri penjelasan dan kemudian perbaiki sehingga kode program di atas menjadi benar.

Jawab:

.....

.....

.....

.....

[illegible]

3. Ardan ingin membuat suatu program untuk persewaan CD film yang meliputi Judul Film, Tahun, Genre, Pemeran, jumlah copy, rating (Dewasa, Remaja atau Anak-anak), serta status tersedianya, iya atau tidak.

Berikut ini contoh form isiannya:

Judul : Beautiful Mind
Tahun : 2005
Genre : Drama
Pemeran :
Jumlah Copy : 2
Rating : Remaja
Tersedia : True

Bantulah ardan untuk menentukan variabel dan tipe data yang akan digunakan dalam program tersebut?

[illegible]

3.1.2.4. Mengasosiasikan

Berdasarkan Lembar Kerja Siswa di atas, buatlah kesimpulan mengenai hal berikut.

1. Dari mana sajakah kita menurunkan variabel dalam menyusun kode program?
2. Pertimbangan apa saja yang dibutuhkan ketika menentukan tipe data dari variabel
3. Sebutkan karakteristik variabel dengan tipe data : integer, real, karakter, string, dan boolean

3.1.2.5. Mengkomunikasikan

Presentasikan jawaban kalian di depan kelas, kemudian diskusikan jawaban kalian dengan teman anda.

4.3. Rangkuman

1. Variabel adalah simbol yang digunakan untuk menyimpan nilai dalam pemrograman
2. Tipe data adalah jenis data yang dimiliki oleh variabel
3. Tipe data terdiri dari tipe data sederhana, tipe data pointer, dan tipe data terstruktur
4. Tipe data sederhana terdiri dari tipe data standar seperti integer, real, character, string, dan boolean; dan tipe data user defined seperti *enumerator* dan *range*.
5. Tipe data terstruktur adalah tipe data yang terdiri dari tipe data-tipe data tunggal
6. Tipe data terstruktur meliputi array, record, file dan set.

4.4. Tugas

1. Buatlah bagian deklarasi variabel dari program berikut.
 - a. program menghitung pajak kendaraan bermotor, di mana besar pajak ditentukan oleh jenis kendaraan bermotor, tahun keluaran dan CC kendaraan.
 - b. Algoritma menyimpan data koleksi buku perpustakaan yang terdiri dari judul buku, pengarang, tahun, penerbit, dan harganya.
2. Carilah sebuah formulir pendaftaran, kemudian buatlah variabel sesuai informasi yang didapat dari formulir tersebut dan tipe datanya, kemudian buatlah deklarasi dalam bahasa pemrograman pascal.
3. Buatlah 5 konstanta dari rumus matematika, fisika, atau kimia yang kalian ketahui. Dan buatlah deklarasi konstanta tersebut ke dalam pemrograman pascal.

4.5. Uji Kompetensi

A. Pilihan Ganda

1. Berikut beberapa contoh penamaan variabel :
 1. nomor_induk_siswa
 2. namasiswa

3. alamat
4. gaji.orangtua
5. golongan_darah
6. peringkat.kelas

daricontohdiatasmanakahpenamaanvariabel yang sesuaipadapemrogramanpascal?

- a. 1,2,3
- b. 3,4,6
- c. 1,5,6
- d. 2,3,4
- e. 1,3,5

2. Diketahui sebuah data dengan tipe datanya sebagai berikut :

1. Nama : Ardan Setiawan
 2. No HP : 085477345123
 3. Tinggi badan : 170
 4. Berat Badan : 65.5
 5. Golongan darah : O
- A. String
 - B. Integer
 - C. Char
 - D. Real
 - E. Boolean

Manakah pasangan data dengan tipe datanya di bawah ini yang benar ?

- a. 1-A,2-B,3-C,4-D,5-E
- b. 1-A,2-B,3-B,4-D,5-C
- c. 1-A,2-A,3-D,4-D,5-C
- d. 1-A,2-A,3-B,4-D,5-C
- e. 1-C,2-D,3-D,4-B,5-E

3. Budi memiliki data nilai matematika dari 10 siswa, budi ingin membuat 10 data tersebut menjadi satu tipe data terstruktur. Manakah tipe data terstruktur yang sesuai dengan masalah di atas?

- a. Array[1..10] of integer;
- b. Array[1..10] of string;
- c. Type

Nilai = record

Nilai1 = integer;

.....

Nilai10 = integer;

End;

d. Array [1..10] of real;

e. Type

Nilai = record

Nilai1 = real;

.....

Nilai10 = real;

End;

4. Berikut ini variabel yang pasti mempunyai tipe data real yaitu

a. Umur

b. JumlahAnak

c. TahunLahir

d. BeratBadan

e. LuasSegitiga

5. Di antara variabel-variabel berikut memerlukan tipe data string, kecuali

a. Pekerjaan

b. NomorHP

c. Agama

d. Sekolah

e. Penghasilan

B. Essay

Buatlah bagian deklarasi dari sebuah program untuk menghitung besar tagihan penggunaan air PDAM, jika pada struk pembayaran diperoleh seperti ini.

STRUK PEMBAYARAN TAGIHAN PDAM			
ID. Pelanggan	: 199102009	Periode	: MAR2013
Nama	: NURSIHAH	Meteran	: 00001075-00001075
Tagihan	: Rp.....10.500		
Admin Bank	: Rp.....2.500		
Total	: Rp.....13.000		
Terbilang	: TIGA BELAS RIBU		
<p>"Terima kasih atas kepercayaan Anda membayar melalui loket kami." Simpanlah struk ini sebagai bukti pembayaran anda. Struk ini merupakan dokumen elektronik dan alat bukti hukum yang sah (PASAL 5 AYAT (1) UU ITE)</p>			

2.2 KEGIATAN BELAJAR 2 OPERATOR DAN EKSPRESI

Pada semester lalu kalian sudah dikenalkan mengenai operator dan ekspresi. Masih ingatkah kalian apa itu operator dan ekspresi? Pada pembahasan kali ini kita akan menggunakannya dalam menyusun kode program.

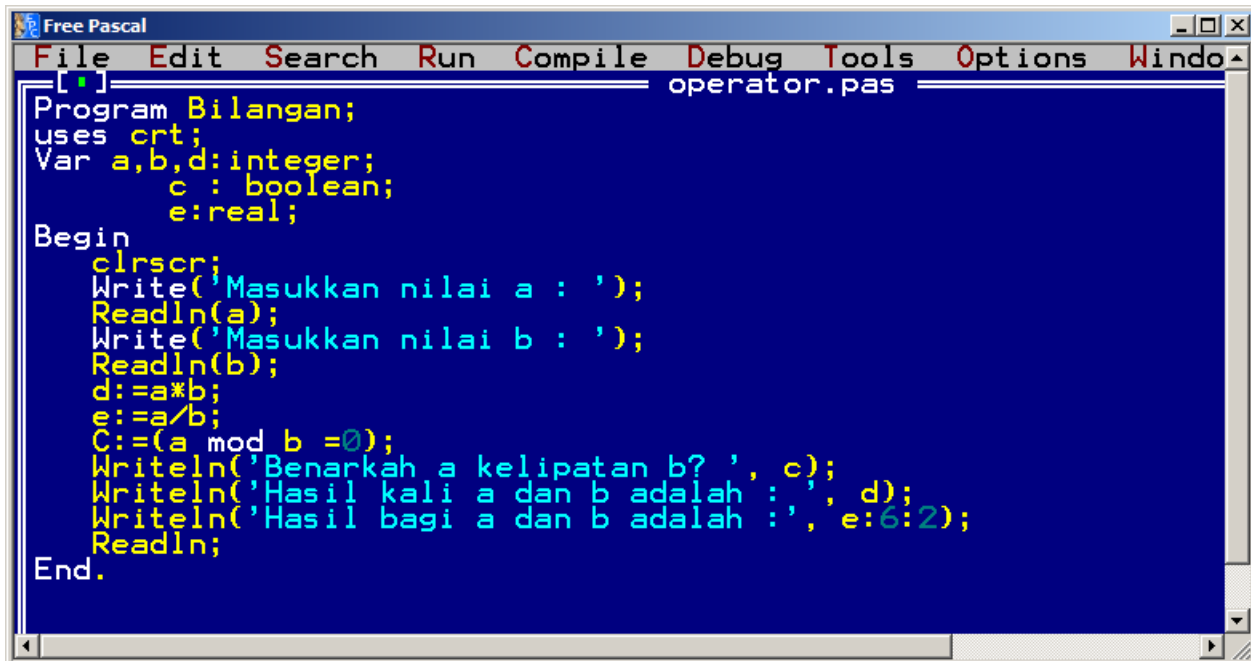
1.2.1. Tujuan Pembelajaran

1. Siswa memahami macam-macam operator dan ekspresi dalam Bahasa Pascal.
2. Siswa mampu menggunakan operator dan ekspresi yang tepat dalam menyelesaikan masalah pemrograman.

1.2.2. Aktivitas Kegiatan Belajar

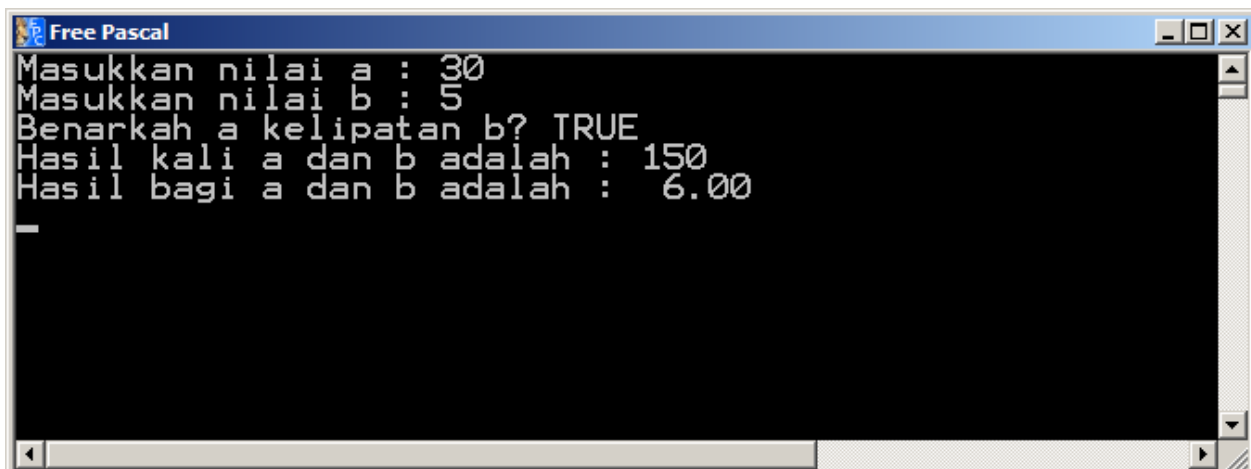
2.2.2.1. Mengamati

Amatilah kode program berikut :



```
Free Pascal
File Edit Search Run Compile Debug Tools Options Window
[.] operator.pas
Program Bilangan;
uses crt;
Var a,b,d:integer;
    c : boolean;
    e:real;
Begin
  clrscr;
  Write('Masukkan nilai a : ');
  Readln(a);
  Write('Masukkan nilai b : ');
  Readln(b);
  d:=a*b;
  e:=a/b;
  c:=(a mod b =0);
  Writeln('Benarkah a kelipatan b? ', c);
  Writeln('Hasil kali a dan b adalah : ', d);
  Writeln('Hasil bagi a dan b adalah : ', e:6:2);
  Readln;
End.
```

Berikut ini contoh outputnya.



```
Free Pascal
Masukkan nilai a : 30
Masukkan nilai b : 5
Benarkah a kelipatan b? TRUE
Hasil kali a dan b adalah : 150
Hasil bagi a dan b adalah : 6.00
_
```

2.2.2.2. Menanya

Setelah mengamati kedua source code diatas, buatlah pertanyaan yang berkaitan dengan kedua source code tersebut, misalkan

1. Operator apa sajakah yang digunakan dalam kode program di atas?
2. Bagian manakah yang disebut ekspresi?
3. Bagian manakah yang disebut assignment?
4. Mengapa variabel d,e,c, mempunyai tipe yang berbeda, ada yang integer, real, dan boolean?
5. Apa maksud $a \bmod b$, $a \bmod b = 0$, dan $c := (a \bmod b)$?

2.2.2.3. Mengumpulkan Informasi/Mencoba

Dari contoh di atas, ada beberapa pernyataan yang melibatkan operasi dari dua buah variabel atau lebih, misalkan $a*b$, a/b , $a \bmod b$, dan $(a \bmod b) = 0$. Bentuk-bentuk ini disebut sebagai ekspresi, yaitu sesuatu yang jika dievaluasi/diproses menghasilkan suatu nilai. Sebagai contoh dimasukkan nilai a adalah 30 dan b adalah 5.

$a*b$ memberikan nilai 150

a/b memberikan nilai 6.00

$a \bmod b$ memberikan nilai 0

$(a \bmod b) = 0$ memberikan nilai TRUE.

Selanjutnya simbol-simbol $*$, $/$, \bmod dan $=$ disebut sebagai operator dan nilai yang dioperasikan disebut operan. Ketika nilai dari ekspresi diberikan kepada variabel lain maka disebut sebagai *assignment*.

$d := a*b;$

$e := a/b;$

$c := (a \bmod b = 0)$

Dalam Pascal assignment ditandai dengan $:=$ (titik dua sama dengan). Selain itu perlu diperhatikan bahwa operasi yang dilakukan menentukan tipe data dari variabel yang diberi nilai. Variabel d tetap integer, e bertipe real, sedangkan c menjadi boolean. Mengapa? Simak jenis-jenis operator berikut untuk mengetahui jawabannya.

Operator

Operator adalah sebuah symbol yang memberitahukan compiler pascal untuk melakukan sebuah operasi matematika maupun operasi logika. Ada beberapa jenis operator dalam pemrograman pascal diantaranya operator aritmatika, operator relasional, operator Boolean

1. Operator aritmatika

Operator aritmatika adalah operasi hitung dari bilangan-bilangan. Terdapat dua macam tipe data bilangan yaitu integer dan bilangan real. Operator-operator yang berlaku keduanya hampir sama, kecuali untuk bagian pembagian. Berikut ini adalah operator-operator untuk tipe data integer dan bilangan real. Operator Aritmatika pada Bilangan Integer

Operator	Keterangan	Contoh
+	Menjumlahkan dua bilangan baik integer maupun real	$7+3=10$
-	Mengurangi bilangan pertama dengan bilangan kedua baik integer maupun real	$7-3=4$
*	Mengalikan dua bilangan, baik integer maupun real	$7 \times 3=21$
/	Membagi dua bilangan, baik integer maupun real	$10 / 2 = 5$
Div	Membagi bilangan integer pertama dengan bilangan integer kedua, di mana hasilnya adalah bagian bulat dari hasil pembagiannya, selanjutnya disebut pembagian bilangan bulat.	$7 \text{ div } 3 = 2$
Mod	Sisa bagi dari hasil pembagian bilangan bulat antara bilangan pertama dengan kedua (keduanya integer)	$7 \text{ mod } 3= 1$

Perhatikan bahwa $7 = 3 \times 2 + 1$, artinya $7 \text{ div } 3 = 2$ dan $7 \text{ mod } 3= 1$

Contoh program dengan operator aritmatika

Program hitung;

var

a,b,c : integer;

d: real;

begin

a:=21;

b:=10;

c := a + b;

writeln('21 + 10 = ', c);

```

c := a - b;
writeln('21 - 10 = ', c);
c := a * b;
writeln('21 x 10 = ', c);
d := a / b;
writeln('21 / 10 = ', d);
c := a mod b;
writeln('21 mod 10 = ', c);
c := a div b;
writeln('21 div 10 = ', c);
end.

```

Dari kode program di atas, dapat dilihat bahwa khusus untuk d bertipe data real, karena merupakan hasil pembagian, yang memungkinkan terjadinya bilangan real.

2. Operator relasional

Operator relasional adalah operator yang menghubungkan nilai dari variabel. Hasil ekspresi dengan operator ini adalah nilai kebenaran True atau False. Berikut operator relasional dalam Pascal.

Operator	Penjelasan	Contoh
=	Periksa apakah nilai keduanya sama	a=10, periksa apakah nilai variabel a sama dengan 10
<>	Periksa apakah nilai keduanya tidak sama	a<>10, periksa apakah nilai variabel a tidak sama dengan 10
>	Periksa apakah nilai pertama lebih besar dari nilai kedua	a>10, periksa apakah nilai variabel a lebih besar 10
<	Periksa apakah nilai pertama lebih kecil dari nilai kedua	A<10, periksa apakah nilai variabel a lebih kecil 10
>=	Periksa apakah nilai pertama lebih besar atau sama dengan nilai kedua	a>10, periksa apakah nilai variabel a lebih besar sama dengan 10
<=	Periksa apakah nilai pertama lebih kecil atau sama dengan nilai kedua	A<10, periksa apakah nilai variabel a lebih kecil sama dengan 10

Operator relasional tidak hanya berlaku pada tipe data bilangan (integer dan real) namun berlaku pula untuk tipe data lain seperti karakter, string maupun boolean.

Sebagai contoh:

'a' < 'c' memberikan nilai true

if (Nama)='Dwi' then

If (StatusLulus=True) then

While (n>100) do

Operasi relasional seringkali muncul pada kondisi bersyarat, yaitu pemenuhan suatu syarat untuk melakukan proses lanjutan. Meskipun pembahasan kondisi bersyarat masih pada Bab selanjutnya, tak ada salahnya kita mengenal bentuknya terlebih dahulu untuk memahami pentingnya operasi relasional.

Contoh program dengan operator relasional

```
Program relasional;  
var  
a, b: integer;  
c,d,e,f : boolean;  
begin  
  a := 21;  
  b := 10;  
  c := a > b;  
  d := a < b;  
  e := a >= b;  
  f := a <= b;  
  writeln('Hasil c = ',c);  
  writeln('Hasil d = ',d);  
  writeln('Hasil e = ',e);  
  writeln('Hasil f = ',f);  
  readln;  
  
end.
```

Pertama kita perhatikan bagian deklarasi variabel dan tipe data yang digunakan dalam program di atas. Nama variabel pada program di atas adalah a, b, c, d, e, f dimana variabel a dan b menggunakan tipe data integer dan variabel c, d, e, f menggunakan tipe data Boolean. Mengapa demikian? Perhatikan main program pada source code di atas, variabel a dan b digunakan untuk menyimpan nilai berupa angka, maka tipe data yang digunakan adalah integer. Selanjutnya perhatikan variabel c, d, e, f pada main program, variabel c nantinya akan menyimpan nilai dari hasil operasi perbandingan variabel a dan variabel b. Pada variabel c akan terjadi proses perbandingan apakah variabel a lebih besar dari variabel b. Jika proses

tersebut dipenuhi artinya variabel a lebih besar dari variabel b maka variabel c akan bernilai TRUE, sedangkan jika proses tersebut tidak dipenuhi maka variabel c akan bernilai FALSE. Selanjutnya variabel d, e, f akan diproses sesuai dengan operasi relasional yang digunakan. Jika proses tersebut terpenuhi maka akan bernilai TRUE sedangkan jika tidak terpenuhi maka akan bernilai FALSE.

Dari contoh program diatas kita dapat mengetahui fungsi dari operator relasional yaitu untuk membandingkan dua nilai yang berbeda. Perbandingan yang terjadi pada dua nilai sesuai dengan operator relasional yang digunakan. Apabila perbandingan tersebut terpenuhi maka akan bernilai TRUE, dan jika tidak terpenuhi akan bernilai FALSE.

3. Operator Boolean

Operator Boolean adalah operator yang digunakan pada sebuah statement. Berikut beberapa operator Boolean pada pascal :

Operator	Penjelasan	Contoh
AND	Jika kedua pernyataan yang dihubungkan bernilai benar maka hasilnya adalah benar, jika salah satu saja salah maka hasilnya adalah salah	(D<0) AND (a>0), artinya D kurang dari 0 dan a lebih dari 0
OR	Jika salah satu saja pernyataan benar maka hasilnya adalah benar, jika keduanya salah maka hasilnya akan salah	(Umur>=60) OR (Umur<=6), artinya jika umur lebih besar 60 atau kurang dari 6
NOT	Digunakan untuk membalikkan nilai kebenaran (ingkaran), jika pernyataan bernilai benar maka akan menjadi salah dan sebaliknya jika pernyataan salah akan menjadi benar	NOT(Profesi=="guru"), artinya Profesinya selain guru

Contoh program dengan operator Boolean

```
Program operasiBoolean;
var
a, b, c, d: integer;
e, f, g, h : boolean;
begin
    a := 21;
    b := 10;
    c := 15;
    d := 30;
    e := (a >b) and (c>d) ;
    f := (a >b) and (c<d) ;
    g:= (a <b) or (c<d);
    h:= (a<b) or (c>d);
    writeln('Hasil e = ',e);
    writeln('Hasil f = ',f);
    writeln('Hasil g = ',g);
    writeln('Hasil h = ',h);
    readln;
end.
```

Dari contoh program diatas dapat kita lihat fungsi dari operator boolean yaitu untuk menghubungkan dua pernyataan atau lebih. Setiap pernyataan akan diproses apakah pernyataan tersebut bernilai TRUE atau FALSE. Selanjutnya program akan mengoperasikan operasi Boolean pada pernyataan tersebut, untuk mengetahui apakah operasi tersebut bernilai TRUE atau FALSE.

Untuk lebih jelasnya, mengenai operator Boolean dapat melihat tabel berikut :

AND

Pernyataan pertama	Pernyataan kedua	Hasil
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
FALSE	TRUE	FALSE
FALSE	FALSE	FALSE

OR

Pernyataan pertama	Pernyataan kedua	Hasil
--------------------	------------------	-------

.....

.....

.....

.....

.....

.....

3.1.2.4. Mengasosiasikan

Berdasarkan Lembar Kerja Siswa di atas, buatlah kesimpulan mengenai apa yang kalian pelajari seperti berikut.

1. Operator adalah
.....
2. Macam – macam operator
 - a. Operator aritmatika adalah
.....
Operator arimatika meliputi
.....
.....
 - b. Operator relasional adalah
.....
Operator relasional meliputi
.....
.....
 - c. Operator Boolean adalah
.....
Operator Boolean meliputi
.....
.....
3. Cara kerja operator dalam pemrograman pascal
 - a. Operator aritmatika.....
.....
.....
 - b. Operator relasional.....
.....
.....

- c. Operator Boolean
-
-
4. Ekspresi adalah
-
5. Assignment adalah
-

3.1.3. Rangkuman

1. Operator adalah sebuah symbol yang memberitahukan compiler pascal untuk melakukan sebuah operasi matematika maupun operasi logika
2. Macam-macam operator: operator aritmatika, operator relasional, dan operator boolean
3. Ekspresi adalah sesuai yang dievaluasi memberikan sebuah nilai

3.1.4. Tugas

1. Buatlah program untuk mengetahui berapa lama Budi membayar angsuran sebuah mobil. Jika diketahui harga mobil tersebut adalah Rp. 375.000.000, gaji Budi tiap bulan adalah Rp. 3.500.000 dengan tunjangan anak sebesar Rp. 350.000 dan tunjangan istri sebesar Rp. 350.000 serta untuk kebutuhan setiap bulan budi adalah Rp 2.750.000.
2. Buatlah sebuah program untuk menghitung besar tabungan tiap bulan. Besar tabungan dihitung dengan carasaldo awal ditambah besar bunga yang didapatkan dikurangi dengan biaya administrasi. Input berupa besar tabungan awal, besarnya prosentase bunga dan biaya administrasi tiap bulan, outputnya adalah besar tabungan akhir.

3.1.5. Uji Kompetensi

A. Pilihan Ganda

1. Diketahui sebuah rumus volume bola adalah $\frac{4}{3}\pi r^3$, bagaimanakah penulisan rumus tersebut dalam pemrograman pascal?
 - a. $4 / 3 \times 3.14 \times r^3$
 - b. $4 : 3 * 3.14 \times r^3$
 - c. $4 / 3 * 3.14 * r * r * r$
 - d. $4 / 3 * 3.14 * r^3$
 - e. $4 / 3 \times 3.14 \times r \times r \times r$
2. Diketahui kode suatu program adalah sebagai berikut :

```
a := 3;
```

```
b := 2;
```

```
c := 5;
```

```
nilai1 := a < b;
```

```
nilai2 := b < c;
```

```
nilai3 := a > c;
```

Apakah nilai1, nilai2, nilai3 ketika program dijalankan ?

- a. TRUE, TRUE, TRUE
- b. TRUE, FALSE, TRUE
- c. FALSE, TRUE, TRUE
- d. FALSE, TRUE, FALSE
- e. FALSE, FALSE, FALSE

3. Perhatikan source code berikut :

```
a = 20;
```

```
b = 10;
```

```
c = a + b;
```

```
writeln('21 + 10 = ', c);
```

```
c = a - b;
```

```
writeln('21 - 10 = ', c);
```

apa yang terjadi ketika program dijalankan?

- a. Program berjalan dengan baik
- b. Terjadi error : illegal expression
- c. Terjadi error pada operator program
- d. Program berhenti di tengah proses
- e. Tidak terjadi apa – apa pada program

4. Diketahui source code program sebagai berikut

```
a := 3;
```

```
b := 5;
```

```
c := 4;
```

```
d := 8;
```

```
e := a > b AND c > d;
```

Agar nilai e menjadi TRUE, maka rumus pada variabel e yang benar adalah

- a. e := a < b AND c > d;
- b. e := a > c AND d > b;
- c. e := b < d; AND c > a;

- d. $e := a > d \text{ AND } c < b;$
 - e. $e := b < c \text{ AND } a > d;$
5. Boim membuat program dengan main program sebagai berikut
- a := '32';
 - b := '12';
 - c := '45';
 - d := a+b+c;
- Apa hasil dari nilai d ketika program dijalankan ?
- a. 89
 - b. 322145
 - c. 98
 - d. 321245
 - e. 542123

B. Essay

Buatlah sebuah program untuk menentukan besar gaji pegawai, di mana gaji pegawai ditentukan oleh gaji pokok, tunjangan keluarga, dan upah lembur. Tunjangan keluarga adalah 15 % dari gaji pokok baik sudah menikah atau belum. Jika sudah punya anak ditambahkan 5% dari gaji pokok untuk tunjangan tiap anak yang dimiliki. Upah lembur dihitung banyak hari lembur dikalikan 100 ribu. Inputan berupa nama pegawai, bagian, status menikah, jumlah anak, jumlah jam lembur. Outputnya adalah total gaji yang diterimakan.

BAB 3

STRUKTUR PERCABANGAN

Pada semester sebelumnya sudah dibahas mengenai struktur percabangan atau kondisi bersyarat pada algoritma. Pada pembahasan kali ini, kita akan menerapkannya pada Bahasa Pemrograman sehingga menghasilkan suatu program yang dapat digunakan untuk menyelesaikan permasalahan sehari-hari.

Banyak permasalahan sehari-hari yang memerlukan struktur percabangan untuk menyelesaikannya. Misalkan seorang siswa bisa dikatakan lulus jika rata-rata nilai ujiannya lebih dari 60. Seorang peminjam buku akan dikenai denda jika terlambat mengembalikan buku pinjaman. Pelanggan listrik akan didenda jika membayar tagihan lebih dari tanggal 20. Seorang pengguna facebook bisa login ke facebook jika user dan passwordnya benar, dan masih banyak lagi yang lain. Kesemua permasalahan yang disebutkan di atas memerlukan persyaratan untuk dapat dilanjutkan dengan sejumlah aksi tertentu. Bisa dikatakan tidak ada program tanpa struktur percabangan di dalamnya.

Sebagaimana pembahasan pada algoritma, dalam bahasa pemrograman ini struktur percabangan terdiri dari beberapa macam, yaitu percabangan dengan 1 kondisi, 2 kondisi, lebih dari dua kondisi dan percabangan bersarang.

3.1. KEGIATAN BELAJAR 1 PERCABANGAN 1 KONDISI

3.1.1. Tujuan Pembelajaran

Siswa memahami struktur percabangan dengan 1 kondisi pada bahasa pemrograman Pascal dan menggunakannya untuk menyelesaikan permasalahan sehari-hari.

3.1.2. Aktivitas Kegiatan belajar

3.1.2.1. Mengamati

Perhatikan permasalahan berikut.

1. Sebuah Toko Baju memberlakukan diskon akhir tahun sebesar 20% untuk semua koleksi yang dijual. Namun diskon akan dinaikkan menjadi 25% jika total belanjaan lebih dari 200 ribu rupiah. Untuk itu toko tersebut membutuhkan sebuah program untuk menghitung jumlah uang yang harus dibayarkan oleh pelanggan untuk baju-baju yang telah dibeli.
2. Di sebuah perusahaan gaji pegawai terdiri dari gaji pokok, tunjangan keluarga dan dikurangi 30 ribu untuk asuransi kesehatan pekerja. Gaji pokok tiap pekerja berbeda-beda sehingga

perlu diinputkan oleh operator, sedangkan tunjangan keluarga dihitung sama untuk semua karyawan, sebesar 10% dari gaji pokok. Khusus bagi karyawan dengan masa kerja lebih dari 20 tahun maka total gajinya akan ditambahkan dengan tunjangan pengabdian sebesar 500 ribu. Untuk itu perusahaan tersebut membutuhkan program perhitungan gaji para karyawannya.

3.1.2.2. Menanya

Dari hasil pengamatan di atas coba buatlah pertanyaan pertanyaan misal :

1. Bagaimana algoritma kedua permasalahan di atas?
2. Berapa banyak persyaratan pada kedua permasalahan di atas?
3. Manakah persyaratan/kondisi pada kedua algoritma di atas?
4. Bagaimana membuat struktur percabangan di Bahasa Pascal?

3.1.2.3. Mengumpulkan Informasi/Mencoba

Ingat kembali algoritma percabangan pada semester 1, di mana dengan pseudocode, struktur percabangan dengan satu kondisi dapat dituliskan

```
IF (kondisi) THEN
    Pernyataan1
    Pernyataan 2
dst
```

Jika dilihat pada permasalahan di atas, pada kasus 1, setiap pembelian dapat diskon sebesar 20% dari harga normal. Namun jika total belanjaan lebih dari 200 ribu maka diskonnya menjadi 25%, maka hal ini dapat dinyatakan dalam algoritma sebagai berikut.

Pertama tentukan semua variabel terlebih, dahulu. Ada hargaTotalBelanja, prosentase diskon (skala 1-100), besar diskon, dan TotalBayar. Selanjutnya dibuat algoritma berikut.

Algoritma DiskonBaju

Variabel hargaNormal, ProsenDiskon,:integer

BesarDiskon, hargaAkhir:bilangan real

- 1) READ TotalBelanja
- 2) ProsenDiskon=20
- 3) IF (TotalBelanja>=200000) THEN
- 4) ProsenDiskon=25
- 5) BesarDiskon=(ProsenDiskon/100)*TotalBelanja
- 6) TotalBayar=TotalBelanja-BesarDiskon
- 7) WRITE TotalBayar
- 8) END

Mengapa data harga normal, prosen diskon dengan besar diskon dan harga akhir mempunyai tipe data yang berbeda? Hal ini dikarenakan, dalam perhitungan besar diskon melibatkan operasi / yang akan menghasilkan bilangan real, sedangkan harga akhir hasil mengurangi harga normal (integer) dengan harga diskon (real) sehingga menghasilkan bilangan real juga.

Selanjutnya algoritma di atas akan dikonveris ke dalam bahasa pemrograman Pascal. Karena pseudocode yang kita bahas adalah pseudocode berbasis Pascal, maka tidak ada perubahan yang berarti. Dalam bahasa pascal, struktur percabangand engan 1 kondisi dituliskan,

if (kondisi) then

Pernyataan;

Jika pernyataan lebih dari satu maka dituliskan

if (kondisi) then

Begin

Pernyataan1;

Pernyataan2;

dst;

end;

Sekarang kita lihat bagaimana menyajikan algoritma di atas dalam bahasa Pascal.

```
Program DiskonBaju;
Uses crt;
var hargaNormal, ProsenDiskon:integer;
    BesarDiskon, hargaAkhir: real;
Begin
    Clrscr;
    write('Masukkan total harga pembelian: ');
    readln( TotalBelanja);
    ProsenDiskon:=20;
    if (TotalBelanja>=200000) then
        ProsenDiskon:=25;
    BesarDiskon=(ProsenDiskon/100)*TotalBelanja;
    TotalBayar=TotalBelanja-BesarDiskon;
    writeln('Total harga yang harus dibayar adalah : ', TotalBayar:6:2);
end.
```

Untuk permasalahan kedua kita tinggalkan untuk diskusi pada LKS berikut.

LEMBAR KERJA SISWA

1. Perhatikan kembali permasalahan kedua

Di sebuah perusahaan gaji pegawai terdiri dari gaji pokok, tunjangan keluarga dan dikurangi 30 ribu untuk asuransi kesehatan pekerja. Gaji pokok tiap pekerja berbeda-beda sehingga perlu diinputkan oleh operator, sedangkan tunjangan keluarga dihitung sama untuk semua karyawan, sebesar 10% dari gaji pokok. Khusus bagi karyawan dengan masa kerja lebih dari 20 tahun maka total gajinya akan ditambahkan dengan tunjangan pengabdian sebesar 500 ribu. Untuk itu perusahaan tersebut membutuhkan program perhitungan gaji para karyawannya.

a. Buatlah algoritma dalam bentuk pseudocode menggunakan struktur perulangan dengan 1 kondisi, dengan input: nama karyawan, gaji pokok, masa kerja dan outputnya adalah total gaji karyawan tersebut.

b. Tuliskan kode pemrograman dalam bahasa Pascal dengan ketentuan sebagai berikut:

Format input:

Nama karyawan :

Besar Gaji Pokok:.....

Masa Kerja:.....

FormatOutput:

Total gaji karyawan bernama **Nama_Karyawan** sebesar **Rp00**.

c. Kompilasi program yang sudah kalian buat hingga berhasil dijalankan

Jawab :

a. Algoritma dalam pseudocode

Algoritma GajiKaryawan

Variabel

.....

1) READ

2) READ

3) READ

4) Tunj_Keluarga =

5) Total_Gaji =

6) IF (.....) THEN

7) Total_Gaji=Total_Gaji +

8) WRITE

9) END

b. Kode pemrograman dengan bahasa Pascal

Program

Var

.....

```

.....
begin
writeln('PROGRAM GAJI KARYAWAN');
  Write ('.....');
readln(.....);
write('.....');
readln(.....);
write('.....');
readln(.....);
.....:= 0.1 *.....;
.....:= .....+ .....;
if (.....>=20) then
  .....:=.....+ .....
  writeln('Total gaji karyawan bernama ',.....,'sebesar Rp. ', ..... : .... : ....);
readln;
end.

```

- c. Kompilasi kode program yang sudah dibuat dengan Free Pascal. Perbaiki hingga kode kalian berhasil.

2. Perhatikan masalah berikut:

Sebuah persewaan komik memberikan batas waktu menyewa adalah 3 hari dengan biaya sewa per tiap buku adalah Rp1.500,00. Jika peminjam terlambat maka dikenakan denda sebesar Rp 1000,00 per buku per hari. Buatlah program persewaan buku tersebut dengan format tampilan program sebagai berikut.

PROGRAM PERSEWAAN BUKU

Nama Peminjam :

Jumlah Buku :

Lama Pengembalian (hari) :

Biaya Sewa :

Biaya Denda :

Total yang harus dibayar :

Ingat, kalian hanya boleh menggunakan percabangan dengan 1 kondisi saja, yaitu bentuk IF THEN saja, tidak boleh yang lain.

Jawab:

Kalian dapat menggunakan bantuan kode berikut, yaitu dengan mengisi titik-titik yang tersedia, atau sialakan boleh menggunakan ide sendiri jauh lebih baik.


```

Program.....
Var .....
.....
.....
Begin
  Writeln(.....);
  Write( .....);
  readln(.....);
  write(.....);
  readln(.....);
  write(.....);
  readln(.....);
  biaya_sewa:= ..... * .....;
  denda:=0;
  if (lama_kembali>3) then
    denda:=..... * .....*1000;
  total_bayar:=.....+ .....;
  writeln(.....);
  writeln(.....);
  writeln(.....);
  readln;
End.

```

3. Permasalahan berikut diambil dari buku semester 1 Bab 2, pada LKS Kegiatan belajar 1. Subsidi biaya pendidikan diberikan oleh pemerintah kepada semua penduduk dengan besaran 50 ribu tiap anak tiap bulannya, yang hanya diberikan kepada semua keluarga yang memiliki anak usia sekolah (6-17 tahun). Bagi keluarga yang kurang mampu, subsidi dinaikkan menjadi 60 ribu per bulan ditambah dengan bantuan sosial sebesar 100 ribu per bulan. Ukuran keluarga tidak mampu dilihat dari penghasilan dan besarnya konsumsi listrik tiap bulan. Keluarga dikatakan kurang mampu jika penghasilannya kurang dari 1 juta per bulan dan konsumsi listrik kurang dari 100 ribu per bulan. Buatlah program dengan Bahasa Pascal dengan struktur percabangan dengan 1 kondisi, dengan format input dan output sebagai berikut.

Input:

Nama Kepala Keluarga	:
Jumlah anak usia sekolah (6-17 tahun)	:
Penghasilan per bulan	:
Konsumsi listrik tiap bulan	:

Output:

Besar Subdisidi Pendidikan :

Besar bantuan Sosial :

Untuk membuat program ini, kalin bisa membuka lagi algoritma pada pembahasan semester lalu sehingga dengan mudah menerapkannya pada bahasa Pascal. Jalankan program sampai berhasil.

3.1.2.4. Mengasosiasikan

Dari LKS yang telah kalian kerjakan buatlah kesimpulan mengenai bagaimna penerapan struktur percabangan dengan satu kondisi pada bahasa Pascal?

3.1.2.5. Mengkomunikasikan

Presentasikan jawaban kalian di depan kelas, diskusikan hasilnya dengan kelompok lain.

3.1.3. Rangkuman

Struktur percabangan dengan satu kondisi pada Bahasa Pascal mempunyai sintaks sebagai berikut.

if (kondisi) then

Pernyataan;

Jika pernyataan lebih dari satu maka dituliskan

if (kondisi) then

Begin

Pernyataan1;

Pernyataan2;

dst;

end;

3.1.4. Tugas**2.1.4. Tugas**

1. Di suatu daerah ada kebijakan khusus mengenai pembayaran pajak bumi dan bangunan. Setiap pencatatn pembayaran pajak dicatat setiap tahun, dan dicatat apakah pembayar pajak selalu tepat waktu atau tidak. Jika tahun lalu pembayar pajak tidak terlambat membayar dan saat pembayaran juga tidak terlambat maka dia akan mendapat keringanan potongan sebesar 10% dari nilai pajak ditambah potongan tunai Rp 10000,00. Buatlahgkaprogram dengan Bahasa pascal untuk menyelesaikan permasalahan di atas. Inputnya adalah nama pelanggan, besar pajak yang harus dibayar, status pembayaran tahun lalu (terlambat atau tidak) dan outputnya adalah Total Besar Pajak yang harus dibayar.

2. Telusuri algoritma berikut.

Program KondisiAngka**Var a,b,c :integer****Begin** **readln(a,b,c);** **IF (a mod b=0) THEN** **a:=b;** **IF (b mod c=1) THEN** **B:=c+1;** **IF ((b mod 2=0) AND (c mod 2 =1) THEN** **C:=20;** **writeln(a,b,c);** **readln;****END.**

- a. Tentukan output dari program di atas jika a=44, b=22, c =3
 - b. Tentukanlah satu contoh inputan sehingga nilai a,b, dan c tidak berubah, jika bisa.
3. Cari permasalahan yang ada di sekitarmu yang dapat diselesaikan ke dalam bentuk percabangan 1 kondisi. Buatlah program dengan bahasa Pascal untuk menyelesaikannya.

3.1.5. Uji Kompetensi**A. Pilihan Ganda**

1. Format penulisan struktur percabangan dengan 1 kondisi di dalam pascal adalah ...
 - a. if (kondisi) then
 Pernyataan;
 - b. if (kondisi) do
 Pernyataan;
 - c. if (kondisi)
 Pernyataan;
 - d. for(kondisi)then
 Pernyataan;
 - e. for (kondisi) do
 Pernyataan;
2. Perhatikan potongan kode program berikut.
 diskon:=0;
 if harga>500000 then
 diskon := harga*0.05;
 hargatotal:=harga-diskon;

Jika kita menginputkan harga sebesar 500000 maka nilai harga total menjadi ..

- a. 475000
- b. 500000
- c. 25000
- d. 0
- e. 525000

3. Diberikan potongan kode program berikut.

```
p:=4;  
q := 7;  
if (p>q) then  
    p=p-q;  
writeln(p);  
Keluaran dari algoritma di atas adalah ....
```

- a. 4
- b. 7
- c. 3
- d. -3
- e. 11

4. Perhatikan potongan kode program berikut.

```
if bil<=3 then  
    write('bagus');  
if bil>=3 then  
    write('biasa');
```

Jika sebelumnya diberikan nilai bil := 3 maka outputnya adalah ..

- a. error
- b. Bagus
- c. Biasa
- d. Bagusbiasa
- e. Tidak muncul apa apa

5. Perhatikan potongan kode program berikut.

```
Hasil:=b+c;  
If a>b then  
begin  
    c:=a;  
    b:=a;  
    a:=c;  
end;
```

hasil:=a-b;

Jika sebelumnya diberikan nilai a:=9 dan b:=3 maka outputnya adalah ...

- a. 6
- b. -6
- c. 9
- d. 3
- e. 12

B. Essay

1. Buatlah program, dimana user diminta menginputkan 3 bilangan dan outputnya adalah bilangan terbesar!

3.2. KEGIATAN BELAJAR 2PERCABANGAN DENGAN 2 KONDISI

3.2.1. Tujuan Pembelajaran

Siswa memahami algoritma percabangan dengan dua kondisi

3.2.2. Aktivitas Kegiatan Belajar

3.2.2.1. Mengamati

Perhatikan kedua source code berikut :

Kode Program 1

```
Program DiskonBaju;
var hargaNormal, ProsenDiskon:integer;
    BesarDiskon, hargaAkhir: real;
Begin
    write('Masukkan total harga pembelian: ');
    readln( TotalBelanja);
    ProsenDiskon:=20;
    if (TotalBelanja>=200000) then
        ProsenDiskon:=25;
    BesarDiskon=(ProsenDiskon/100)*TotalBelanja;
    TotalBayar=TotalBelanja-BesarDiskon;
    writeln('Total harga yang harus dibayar adalah : ', TotalBayar:6:2);
end.
```

Kode Program 2

```
Program DiskonBaju;
var hargaNormal, ProsenDiskon:integer;
    BesarDiskon, hargaAkhir: real;
Begin
    write('Masukkan total harga pembelian: ');
    readln( TotalBelanja);
    if (TotalBelanja>=200000) then
        ProsenDiskon:=25
    Else
        ProsenDiskon:=25;
    BesarDiskon=(ProsenDiskon/100)*TotalBelanja;
    TotalBayar=TotalBelanja-BesarDiskon;
    writeln('Total harga yang harus dibayar adalah : ', TotalBayar:6:2);
```

end.

3.2.2.2. Menanya

Dari kedua source code diatas buat lah pertanyaan seperti :

1. Adakah perbedaan antara 2 source code diatas ?
2. Apakah kedua source code tersebut jika dibuat program akan menghasilkan output yang sama?
3. Manakah yang lebih baik di antara kode di atas?
4. Kode 1 menggunakan percabangan dengan 1 kondisi sedangkan kode 2 menggunakan percabangan dengan dua kondisi, mengapa disebut demikian?
5. Bagaimana format penulisan percabangan 2 kondisi dalam Bahasa pascal?
6. Bagaimana cara kerja dari percabangan 2 kondisi?

3.2.2.3. Mengumpulkan Informasi/Mencoba

Pada dasarnya program di atas adalah program untuk menyelesaikan permasalahan yang sama, hanya beda dalam penyajian saja. Hal ini sebagaimana pembahasan pada semester 1, bahwa ada percabangan dengan 1 kondisi, 2 kondisi atau lebih.

Pada percabangan 1 kondisi, ketika kondisi dipenuhi maka proses tertentu akan dijalankan, jika tidak maka tidak dilakukan apa-apa. Padahal dalam kehidupan nyata, permasalahan seringkali terjadi melibatkan dua kondisi. Misalkan “Jika nilai lebih besar 60 maka lulus, jika tidak, maka tidak lulus”; “Jika rajin belajar maka naik kelas, jika tidak maka tinggal kelas”; “Jika bobot truk ≥ 100 ton maka tidak boleh lewat, jika kurang dari itu boleh lewat”. Kenyataannya seperti itu dalam kehidupan sehari-hari sehingga percabangan dengan dua kondisi lebih banyak digunakan. Seperti halnya pada contoh di atas, sebenarnya mempunyai dua kondisi, jika total belanja < 200 ribu maka diskonnya 20% jika lebih besar atau sama dengan 200 ribu maka diskonnya menjadi 25%. Oleh karena itu pada percabangan dengan 1 kondisi pernyataan

ProsenDiskon:=20

diletakkan sebelum struktur if then, karena digunakan sebagai nilai default dari prosen diskon. Hal ini dapat disajikan dalam bentuk yang lebih cocok menggunakan struktur if then else sebagai bentuk percabangan dengan dua kondisi.

Dalam percabangan 2 kondisi ini digunakan struktur if then else. Format penulisannya hampir sama dengan percabangan 1 kondisi bedanya di tambah dengan else.

Bentuk umum penulisan if then else sebagai berikut :

If (syarat) then

Pernyataan1

```
else
```

```
    Pernyataan2;
```

Sintaks di atas jika hanya menggunakan pernyataan tunggal untuk setiap syarat. Jika pernyataan lebih dari satu untuk tiap syarat, maka dapat ditambahkan begin dan end seperti berikut.

```
If (syarat1) then
```

```
    begin
```

```
        Pernyataan1;
```

```
        Pernyataan2;
```

```
    end
```

```
else
```

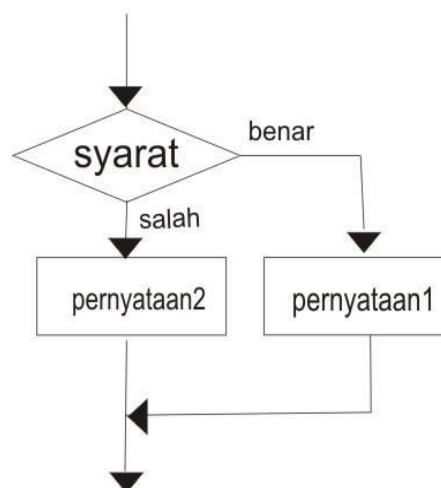
```
    begin
```

```
        Pernyataan3;
```

```
        Pernyataan4;
```

```
End;
```

Cara kerja nya bisa dijelaskan melalui gambar flowchart berikut :



Bentuk flowchart seperti di atas sudah kita pelajari pada semester sebelumnya.

Agar lebih jelas kerjakan LKS berikut secara berkelompok.

LEMBAR KERJA SISWA

1. Toko buah membuat suatu promo jika pembeli membeli apel lebih dari 2 kg dan jeruk lebih dari 3 kg maka akan mendapat diskon sebesar 10%, pelanggan yang tidak membeli apel lebih dari 2 kg dan jeruk 3 kg maka hanya mendapatkan diskon 5%. Harga untuk 1 kg jeruk adalah 10000 sedangkan harga untuk 1 kg apel adalah 15000.
 - a. Dari permasalahan tersebut kode programnya menggunakan pascal!
 - b. Jelaskan cara kerja if di dalam program tersebut!

Jawab

a. Kode Program

```

program diskonbuah;
var.....: real;
.....: integer;
begin
    write('Jumlah jeruk : ');
    readln(.....);
    write('Jumlah apel  : ');
    readln(.....);
    ..... := .....;
    ..... := .....;
    ..... :=.....;
    if (.....) and (.....) then
        .....:= .....
    else
        .....:= .....;
        .....:=.....;
    write('Total uang yang harus dibayar adalah : ', total:8:2);
    readln;
end.

```

b. Penjelasan:

[illegible]

2. Perhatikan kedua source berikut

Kode Program 1

```
program uji1;  
var test : integer;  
begin  
    write('masukkan input : ');  
    readln(test);  
    if test<=5 then  
        writeln('bilangan Anda kurang dari atau sama dengan 5');  
    if test>5 then  
        writeln('bilangan Anda lebih besar 5');  
    readln;  
end.
```

Kode Program 2

```
program uji2;  
var test : integer;  
begin  
    write('masukkan input : ');  
    readln(test);  
    if test<=5 then  
        writeln('bilangan Anda kurang dari atau sama dengan 5');  
    else  
        writeln('bilangan Anda lebih besar 5');  
    readln;  
end.
```

- Inputkan bilangan 3, telusuri dengan source code 1
- Inputkan bilangan 3, telusuri dengan source code 2
- Apakah hasil keduanya sama? Lantas apa perbedaan keduanya? Manakah yang lebih efektif? Lihat berapa kali harus memeriksa syarat, jika banyaknya pemeriksaan lebih sedikit maka berarti lebih efektif.

jawab

- Hasil penelusuran source code 1

.....

.....

.....

.....

.....

.....

.....

b. Hasil penelusuran source code 2

.....

.....

.....

.....

.....

.....

.....

c. Penjelasan

.....

.....

.....

.....

.....

.....

.....

3. Buatlah program dengan struktur percabangan dengan dua kondisi untuk permasalahan berikut. (permasalahan ini diambil pada LKS semester sebelumnya)

Pembayaran gaji karyawan di Perusahaan X terdiri dari 3 komponen utama, gaji Pokok, Tunjangan masa kerja (TMK), dan tunjangan keluarga (TK). TMK dihitung berdasarkan masa kerja, jika masa kerja >10 tahun maka TMKnya adalah 500 ribu jika kurang dari itu TMK 100 ribu. Tunjangan keluarga dilihat dari status, Menikah atau Belum Menikah, jika sudah menikah maka tunjangannya dihitung sebagai berikut

Tunjangan Istri=10% x Gaji Pokok

Tunjangan Anak=banyak anak x 2,5% x Gaji Pokok

Tunjangan Keluarga=Tunjangan Istri+Tunjangan Anak

Sehingga gaji Totalnya adalah Gaji Pokok + Tunjangan masa kerja + Tunjangan Keluarga

Jika belum menikah maka Tunjangan Keluarga diganti dengan nama Tabungan Berkeluarga sebesar 5% x Gaji Pokok, sehingga total penerimaan gaji adalah gaji Pokok + Tunjangan Masa kerja + Tabungan Berkeluarga.

Kompilasi program yang sudah kalian buat hingga berhasil.

3.2.2.4. Mengasosiasikan

Berdasarkan hasil pembahasan dan pengerjaan LKS kalian, apa yang dapat kalian simpulkan mengenai struktur percabangan dengan dua kondisi dengan Bahasa Pascal. Jelaskan perbedaannya dengan satu kondisi, mana yang lebih sering digunakan?

3.2.2.5. Mengkomunikasikan

Presentasikan jawaban kalian di depan kelas, diskusikan dengan kelompok/teman kalian yang lain.

3.2.3. Rangkuman

Bentuk umum struktur percabangan dengan dua kondisi menggunakan bahasa Pascal adalah sebagai berikut

If (syarat) then

Pernyataan1

else

Pernyataan2;

untuk pernyataan tunggal pada tiap syarat, atau

If (syarat1) then

begin

Pernyataan1;

Pernyataan2;

end

else

begin

Pernyataan3;

Pernyataan4;

End;

Untuk pernyataan lebih dari satu untuk tiap syarat.

3.2.4. Tugas

1. Diberikan masalah berikut

Di sebuah loket stasiun menyediakan tiket untuk dua kereta api, bisnis dan ekonomi. Harga tiket masing-masing kereta adalah sebagai berikut

Kategori/Kelas	Bisnis	Ekonomi
Dewasa	Rp 100.000,-	Rp 100.000,-
Anak	Rp 50.000,-	Rp 50.000,-
Infant	Rp 5.000,-	Rp 5000,-

Dari masalah di atas dibuatkan sebuah program untuk menentukan harga tiket, di mana masukan terdiri dari jenis kereta api yang dipilih, jumlah tiket yang dipesan, terdiri dari tiket dewasa, anak-anak, dan infant. Keluarannya adalah besar uang yang harus dibayar.

Tidak berhenti di sini, program juga meminta input berapa besar uang yang dibayarkan dan mencetak besar kembalian. Buatlah program dengan Bahasa Pascal

2. Carilah permasalahan di sekitarmu yang dapat diselesaikan dengan membuat program yang memuat struktur percabangan dengan dua kondisi.

3.2.5. Uji Kompetensi

A. Pilihan ganda

1. Jika seorang pegawai mempunyai masa kerja lebih dari 5 tahun maka dia akan mendapatkan tambahan gaji sebesar 500000 sedangkan untuk yang lainnya mendapat tambahan gaji 200000

Penulisan struktur percabangan yang sesuai dengan masalah diatas adalah ...

- a. `if masakerja<=5 then`
`tambahan:=500000;`
`else`
`tambahan := 200000;`
- b. `if masakerja<=5 then`
`tambahan:=200000;`
`else`
`tambahan := 500000;`
- c. `if masakerja>=5 then`
`tambahan:=500000;`
`else`
`tambahan := 200000;`
- d. `if masakerja<5 do`
`tambahan:=500000;`
`else`
`tambahan := 200000;`
- e. `if masakerja>5 do`
`tambahan:=500000;`
`else`
`tambahan := 200000;`

2. Perhatikan potongan kode program berikut :
`if (masakerja>=5) and (gajitotal>3000000) then`
`pajak := gajitotal*0.1;`

```

else
    pajak := gajitotal*0.05;
gajibersih := gajitotal-pajak;

```

Jika Agus sudah bekerja di perusahaan tersebut selama 4,5 tahun dan dikarenakan dia banyak lembur sehingga gajitotalnya pada bulan ini adalah 3500000 maka gaji bersih yang diterima agus adalah

- a. 3482500
- b. 3325000
- c. 3150000
- d. 3500000
- e. 3000000

3. Perhatikan potongan kode program berikut

```

if 10<=a<=20 then
    write('bilangan diantara 10 sampai 20');
if 21<=a<=40 then
    write('bilangan diantara 21 sampai 40');

```

Jika nilai a adalah 9 maka outputnya adalah ...

- a. bilangan diantara 10 sampai 20
- b. bilangan diantara 21 sampai 40
- c. program error
- d. tak ada yang terjadi
- e. muncul output bilangan diantara 10 sampai 20 dan bilangan diantara 21 sampai 40

4. Perhatikan potongan program berikut

```

if(1 <= peringkat) and (peringkat<= 4) then
    beasiswa:=1000000;
else
    if status = 'kurang mampu' then
        beasiswa := 500000;

```

Tono selalu mendapat peringkat 1 di kelasnya dan dia adalah siswa kurang mampu maka total beasiswa yang di dapat adalah ...

- a. 1500000
- b. 1000000
- c. 750000
- d. 500000
- e. 0

5. Perhatikan potongan algoritma berikut.

```

Readln(a);

```

```
Readln(b);  
if ((a mod 2 =1) or (b mod a=0 ) then  
    begin  
        c:=a+b  
        d:=a-b  
    end;  
else  
    begin  
        c:=2*a+b  
        d:=2*b-a  
    end;  
writeln(c,' dan ', d);  
Apa keluaran berturut-turut c dan d jika diberikan a=4 dan b = 10  
a. 24 dan 16  
b. 14 dan 6  
c. 14 dan -6  
d. 24 dan 4  
e. 8 dan 2
```

Essay

1. Buatlah program untuk menentukan kelulusan berdasarkan rata-rata nilai UAN yang terdiri dari matematika, bahasa indonesia, bahasa inggris. Siswa dinyatakan lulus jika Rata-ratanya lebih besar sama dengan 60 dan tidak ada nilai yang kurang dari 50.

3.3. KEGIATAN BELAJAR 3 PERCABANGAN LEBIH DARI DUA KONDISI

Pembahasan pada materi algoritma yang lalu telah menjelaskan bagaimana struktur percabangan dengan dua macam penyajian, menggunakan struktur if then atau switch of. Secara garis besar, karena pseudocode yang dibahas adalah pseudocode Pascal-like (mirip Pascal) maka tidak ada perbedaan yang berarti. Untuk itu pada pembahasan kali ini lebih banyak latihan penyelesaian permasalahan.

3.3.1. Tujuan Pembelajaran

Siswa memahami percabangan lebih dari dua kondisi dengan Bahasa Pascal

3.3.2. Aktivitas Kegiatan Belajar

3.3.2.1. Mengamati

Perhatikan masalah berikut :

1. Perhitungan Gaji

Misalkan di sebuah perusahaan yang bergerak di bidang pemasaran, gaji pegawai dihitung berdasarkan gaji pokok dan bonus dari omset penjualan

Jika Omset > 100 juta maka bonusnya adalah $1.5\% \times \text{Omset Penjualan}$

Jika Omset antara 50-100 juta, bonusnya adalah $1\% \times \text{Omset Penjualan}$

Jika kurang dari 50 juta maka ia tidak mendapatkan bonus.

Bagaimana mengimplementasikan aturan tersebut dalam program komputer?

2. Persewaan Mobil

Sebuah persewaan mobil mempunyai 4 jenis koleksi mobil, Avanza, Xenia, Innova dan Alphard.

Biaya persewaan mobil terdiri dari dua komponen, yaitu Biaya Sewa dan Biaya Asuransi.

Berikut ini adalah Tabel biaya sewa mobil dan asuransi

Jenis Mobil	Biaya sewa/hari	Biaya Asuransi
Avanza	300 ribu	15 ribu
Xenia	300 ribu	15 ribu
Innova	500 ribu	25 ribu
Alphard	750 ribu	30 ribu

Akan dirancang sebuah program untuk menghitung biaya persewaan mobil. Algoritma akan meminta inputan berupa Jenis Mobil yang disewa, dan lama hari menyewa. Asuransi dihitung sekali bayar untuk berapa pun lama peminjaman. Biaya persewaan dihitung dengan menjumlahkan total biaya sewa untuk n hari ditambah dengan biaya asuransi.

3.3.2.2. Menanya

Dari masalah diatas coba buatlah pertanyaan pertanyaan misal :

- Bagaimana menyajikan persyaratan (percabangan) dengan lebih dari dua kondisi dalam bahasa Pascal
- Apakah bisa menggunakan bentuk if then else?
- Jika dalam algoritma ada bentuk switch of, apakah di bahasa pemrograman juga bisa?

d. Kapan bentuk if then atau switch of digunakan?

3.3.2.3. Mengumpulkan Informasi/Mencoba

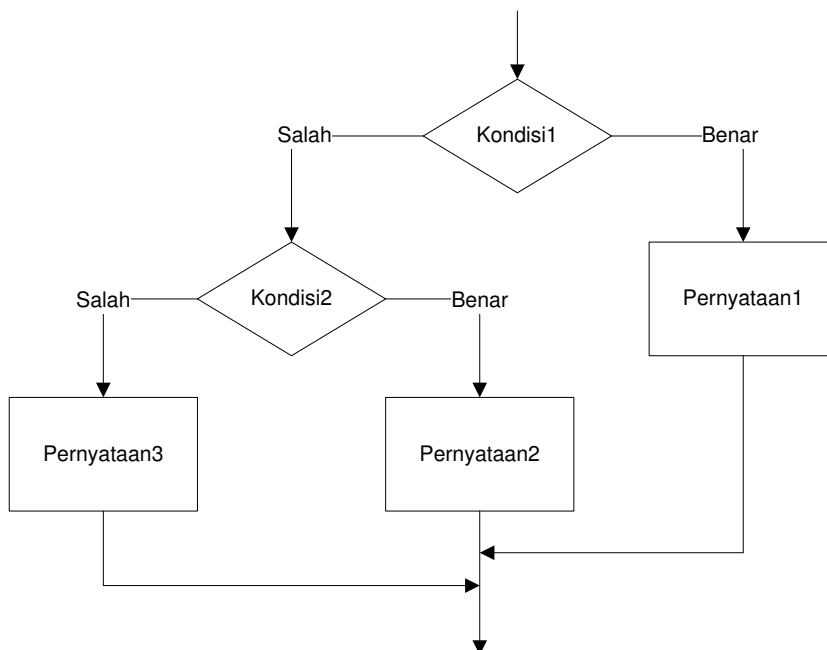
Sebagaimana dalam penyajian algoritma dengan pseudocode, di Bahasa Pascal kita bisa menggunakan dua macam struktur untuk menyajikan percabangan dengan lebih dari dua kondisi, yaitu dengan menggunakan struktur if then else if atau dengan menggunakan bentuk switch of.

Struktur if then else if

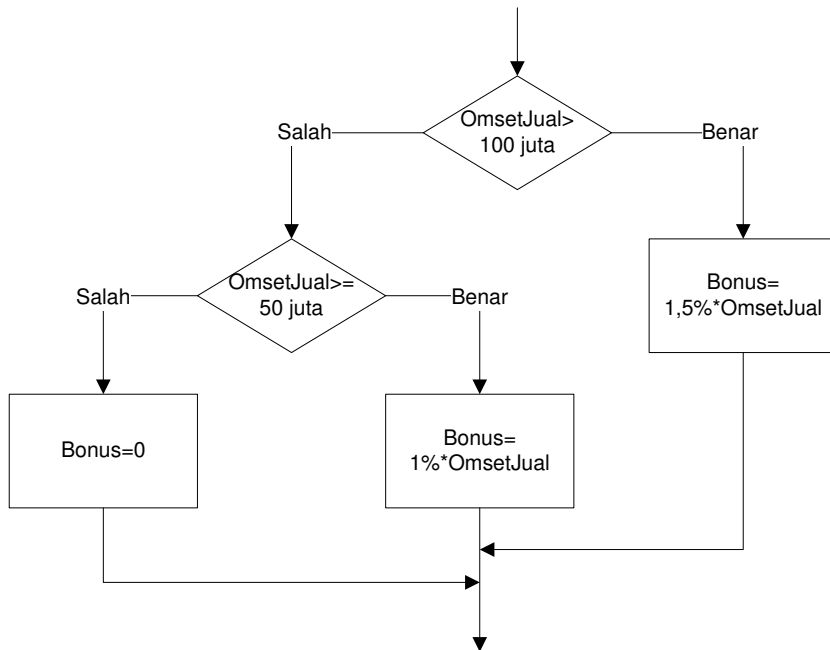
Dalam pseudocode, percabangan dengan lebih dari dua kondisi dituliskan sebagai berikut.

```
IF (kondisi1) THEN
  Pernyataan1
ELSEIF (kondisi2) THEN
  Pernyataan2
ELSE
  Pernyataan3
```

Sehingga dapat dibuat flowchartnya dalam bentuk sebagai berikut.



Dalam hal ini untuk kasus dapat dituliskan flowchart percabangannya dalam bentuk berikut.



Demikian juga pada bahasa Pascal, kita bisa menyajikan permasalahan di atas dalam sintaks sebagai berikut.

```

if (kondisi1) then
  pernyataan1
else if (kondisi2) then
  pernyataan2
else
  pernyataan3;
  
```

sehingga untuk permasalahan 1, dapat dituliskan bagian percabngannya adalah

```

if (Omset_Jual>100000000) then
  bonus:=(1.5/100)*Omset_Jual
else if (Omset_Jual>=50000000) then
  bonus:=(1/100)*Omset_Jual
else
  bonus:=0;
  
```

Bagaimana dengan permasalahan kedua? Pada permasalahan kedua ini, kondisi jauh lebih banyak, dalam hal ini ada 5 cabang atau kondisi sesuai dengan banyaknya jenis mobil. Selain itu dalam setiap cabang ada lebih dari satu pernyataan sehingga sehingga dapat dituliskan kode dalam Pascalnya adalah sebagai berikut.

```

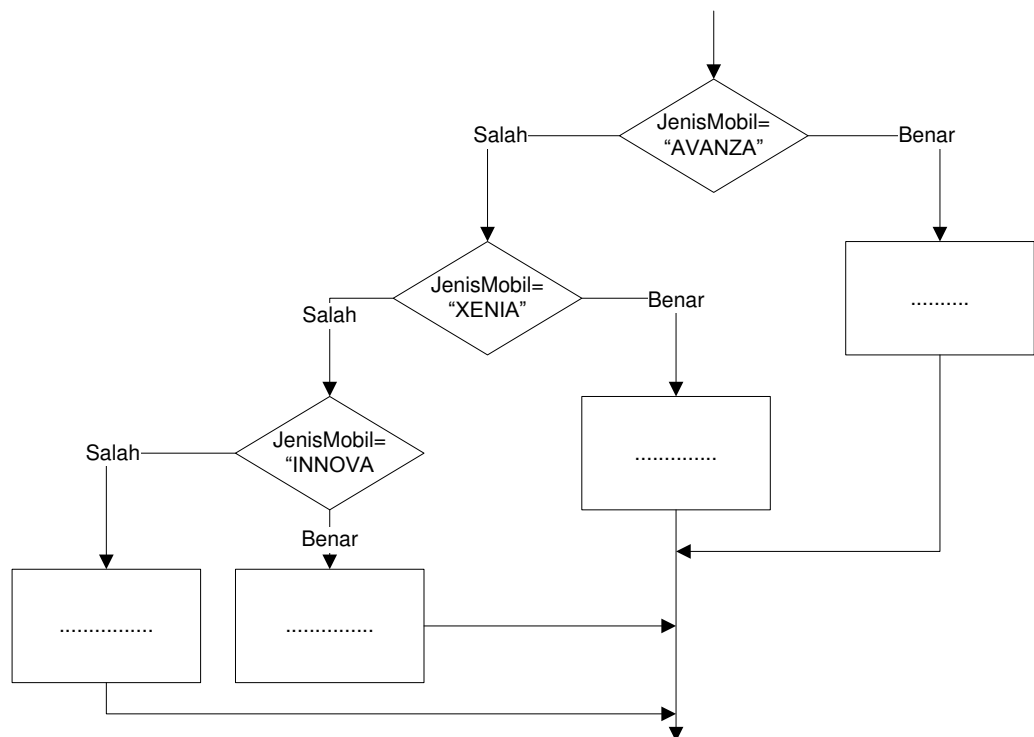
If (jenis_Mobil='Avanza') then
  Begin
  
```

```

Pernyataan1;
Pernyataan2;
End
Else if (jenis_Mobil='Avanza') then
    Begin
        Pernyataan1;
        Pernyataan2;
    End
Else if (jenis_mobil='Innova') then
    Begin
        Pernyataan1;
        Pernyataan2;
    End
Else
    Pernyataan1;
    Pernyataan2;

```

Sehingga dapat dituliskan bentuk flowchartnya adalah sebagai berikut.



Detail program akan dilanjutkan dalam diskusi pengerjaan LKS.

Struktur case of

Perhatikan permasalahan kedua, dengan 5 kondisi (percabangan) maka penyajian algoritma menjadi memanjang ke kanan. Dan ini akan menjadi kurang menarik dan efektif dalam penulisan,

meskipun hasilnya tetap sama. Alternatifnya adalah dengan menggunakan struktur case of sebagai mana pada pembahasan algoritma.

Secara umum, sintaks struktur case of adalah sebagai berikut.

```
Case (variabelpilihan) of
  Nilai1: pernyataan1;
  Nilai2:pernyataan2;
  Nilai3:pernyataan3;
  ...
Else
    PernyataanElse;
end;
```

Pada struktur case of, seluruh kemungkinan nilai didaftar dan diberikan pernyataan yang sesuai dengan nilai yang dimiliki variabel. Ini tentu akan lebih efektif dibandingkan menggunakan if then else if secara berulang-ulang. Kekurangannya bahwa struktur ini hanya bekerja pada variabel dengan tipe data ordinal, yaitu tipe data yang dapat dihitung, dalam hal ini adalah integer, karakter atau tipe data set (tipe data terstruktur). Jika variabel yang akan diubah dalam bentuk case of belum menjadi salah satu bentuk ini maka perlu diubah ke salah satu bentuk tipe data. Caranya? Sudah kalian pelajari di pembahasan semester lalu.

Lihat kembali permasalahan kedua, yaitu persewaan mobil. Kita bisa membuatkan menu pilihan untuk mewakili setiap jenis mobil. Misalkan ditampilkan di awal program keterangan berikut.

Mobil yang disewakan:

1. **Avanza**
2. **Xenia**
3. **Innova**
4. **Alphard**

Masukkan nomor pilihan mobil:{*baca sebagai variabel var_pilih*}

Selanjutnya kita bisa menggantikan setiap jenis dengan nilai var_pilih dari 1, 2, 3, atau 4 untuk masing-masing kategori mobil. Dengan ini maka var_pilih bertipe ordinal, sehingga dapat diterapkan struktur case sebagai berikut.

```
case (var_pilih) of
  1: begin
    Pernyataan1a;
    Pernyataan1b;
```

```

        end;
    2: begin
        Pernyataan2a;
        Pernyataan2b;
    end;
    3: begin
        Pernyataan 3a;
        Pernyataan 3b;
    End;
Else
    Begin
        PernyataanElse1;
        PernyataanElse2;
    End;
End;
{ini akhir dari struktur case of}

```

Selain kondisi berupa data tunggal, struktur case of juga dapat diterapkan pada interval data ordinal, emisalkan sebagai berikut. Dalam algoritma penghitungan bonus karyawan berdasarkan omset jual di atas. Agar dapat diterapkan struktur case of, maka besar omset jual dituliskan sebagai variabel dengan tipe integer (pembulatan) di mana nilainya dalam satuan juta. Shingga jika variabel omset_jual= 125 artinya 125 juta, sehingga bonusnya dihitung $(1.5/100) * \text{omset_jual} * 1000000$. Sekali lagi, ini adalah salah satu bentuk strategi untuk memecahkan masalah. Dengan demikian kita bisa memberikan alternatif solusi kondisi percabangan adalah sebagai berikut:

```

readln(omset_jual);      {dalam juta}
case (omset_jual) of
    0..49      : bonus:=0;
    50..100    : bonus:=(1/100)*omset_jual*1000000;
else
    bonus:=(1.5/100)*omset_jual*1000000;
end;

```

Struktur di atas meberikan gambaran bahwa struktur case of dapat diterapkan pada rentang nilai dari suatu data ordinal. Bisa juga dengan menggunakan tipe data karakter seperti berikut.

```

case (karakter) of
    'a'..'j'    : pernyataan1;

```

```
'k'..'p' :pernyataan2;  
'q'..'u' : pernyataan3  
else  
    PernyataanElse;  
end;
```

Untuk lebih jelasnya kerjakan LKS berikut.

LEMBAR KERJA SISWA

1. Sempurnakan program untuk permasalahan 1 dengan struktur if then else if!
2. Sempurnakan program untuk permasalahan 2 dengan struktur if tehcn else if dan case of!
3. Perhatikan masalah berikut :

Rizal ingin membuat program untuk menghitung :

- Luas persegi
- Luas persegi panjang
- Luas segitiga
- Luas lingkaran

Dia ingin membuat user bisa memilih menu, setiap menu akan di digunakan untuk menghitung luas suatu bidang.

- a. Buatlah program yang sesuai dengan keinginan rizal!
- b. Agar lebih jelas lagi terangkan alur programnya dengan flowchart!

3.3.2.4. Mengasosiasikan

Setelah mengerjakan soal soal diatas tuliskan kesimpulan kalian terkait dengan struktur percabangan dengan lebih dari dua kondisi!

- a. Bagaimana penyajian struktur percabangan dengan bahasa Pascal?
- b. Kapan saat yang tepat dalam penggunaan if then else dan case of di dalam percabangan lebih dari 2 kondisi?
- c. Apa kelebihan dan kekurangan masing-masing struktur if then else if dan case of.

3.3.2.5. Mengkomunikasikan

Presentasikan jawaban kalian di depan kelas, kelompok lain menanggapi. Jalannya diskusi dimimbing oleh Guru.

3.3.3. Rangkuman

Struktur percabangaan dengan lebih dari dua kondisi

1. **Menggunakan pernyataan if then else if**

```

if (kondisi1) then
    pernyataan1
else if (kondisi2) then
    pernyataan2
else
    pernyataan3;

```

2. Menggunakan pernyataan case of

Case (variabelpilihan) of

```

    Nilai1: pernyataan1;
    Nilai2:pernyataan2;
    Nilai3:pernyataan3;
    ...
Else
    PernyataanElse;
end;

```

3.3.4. Tugas

1. Penilaian di sebuah sekolah menggunakan format A, B, C, D, dan E dengan kriteria sebagai berikut

Skor	Huruf	Kategori
80-100	A	Sangat Baik
70-79	B	Baik
60-69	C	Cukup
40-59	D	Tidak Baik
Kurang dari 40	E	Gagal

Buatlah program permasalahan di atas dengan inputan Nama siswa, skor dan menampilkan nilai dalam huruf dan kategori prestasinya. Buatlah dua program, satu dengan struktur if then else if dan case of.

2. . Buatlah sebuah algoritma dalam bentuk pseudocode untuk penggajian yang ditentukan berdasarkan lama bekerja sebagai berikut.

Lama Kerja	Gaji Pokok	Tunjangan Kinerja	Tunjangan keluarga	Potongan premi Asuransi
1-5 tahun	1.200.000	20% x gaji pokok	15% x gaji pokok	15.000
6-10 tahun	1.500.000	20% x gaji pokok	15% x gaji pokok	15.000
10-15 tahun	2.000.000	25% x gaji pokok	15% x gaji pokok	20.000
Lebih dari	3.000.000	25% x gaji pokok	15% x gaji pokok	20.000

15 tahun				
----------	--	--	--	--

Diasumsikan semua pegawai mendapatkan Tunjangan Kinerja dan tunjangan keluarga. Keseluruhan penerimaan gaji dipotong premi asuransi.

- a. Buatlah program dengan struktur CASE OF.
3. Temukan sebuah permasalahan yang memerlukan percabangan dengan lebih dari dua kondisi. Buat program penyelesaiannya masing-masing dengan pernyataan if then else dan case of.

3.3.5. Uji Kompetensi

1. Perhatikan potongan kode program berikut.
case nilai of

```
38..48:write('E');
49..59:write('D');
60..70:write('C');
71..81:write('B');
else write('A')
```

end;

saat kita menginputkan nilai 100 outputnya adalah ...

- a. A
 - b. B
 - c. C
 - d. D
 - e. tidak ada
2. Perhatikan potongan kode program berikut :
if nilai >= 70 then
 write('bagus')
else if nilai >= 60 then
 write('cukup');
if nilai < 60 then
 write('remidi');

 else write('tidak remidi');

Jika nilainya adalah 65 maka outputnya adalah:

- a. bagus
 - b. cukup
 - c. tidak remidi
 - d. cukup tidak remidi
 - e. remidi
3. Jika aturan penilaian sebagai berikut
 Jika nilai >= 70 maka dicetak "bagus"
 Jika 60 <= nilai < 70 maka dicetak "cukup"
 Jika kurang dari 60 maka dicetak "remidi"
Jika aturan tersebut ditulis dengan menggunakan case of menjadi ...

- a. case nilai of
0-61:write('remidi');
60-70:write('cukup');
70-100:write('bagus');
end;
 - b. case nilai of
0..59:write('remidi');
60..69:write('cukup');
70..100:write('bagus');
end;
 - c. case nilai of
0..60:write('remidi');
61..70:write('cukup');
71..100:write('bagus');
end;
 - d. case nilai of
0-60:write('remidi');
61-70:write('cukup');
71-100:write('bagus');
end;
 - e. case nilai of
0<61:write('remidi');
60<70:write('cukup');
70<100:write('bagus');
end;
4. Case pilihan of
'1': hasil:=a+b;
'2': hasil:=a-b;
'3': hasil:=a*b;
Else hasil:=a/b;
end;
Jika input a =15, b=3 dan pilihan=5 maka outputnya adalah...
- a. 18
 - b. 13
 - c. 45
 - d. 3
 - e. 0
5. Perhatikan potongan kode program berikut.
Case tahun of
2000..2004: pajak:=150000;
2005..2009: pajak:=200000;
2010..2012: pajak:=300000;

2013..2014: pajak:=350000;

End;

Jika aji membeli motor pada tahun 2007 maka dia harus membayar pajak motornya sebesar...

- a. 150000
- b. 200000
- c. 300000
- d. 350000
- e. Tidak ada

Essay

1. Buatlah program dimana pengguna dapat memilih menghitung volume kubus, volume tabung dan volume bola!

3.4. KEGIATAN BELAJAR 4 PERCABANGAN BERSARANG

3.4.1. Tujuan Pembelajaran

Siswa memahami percabangan bersarang pada bahasa Pascal.

3.4.2. Aktivitas Kegiatan Belajar

3.4.2.1. Mengamati

Diberikan masalah sebagai berikut :

Sebuah dealer memberikan daftar harga per merk berikut adalah daftar harga motor di dealer tersebut:

- a. Honda
 - Supra x 125 : 15000000
 - Megapro : 18000000
 - Tiger : 25000000
- b. Yamaha
 - Jupiter : 17000000
 - Mio : 13000000
 - Vixion : 22000000
- c. Suzuki
 - Nex : 12000000
 - Satria FU : 18000000
 - Smash : 13000000

Dari Ketentuan diatas akan dibuat algoritma untuk mengetahui harga motor yang di pilih pelanggan.

3.4.2.2. Menanya

Buatlah pertanyaan dari masalah diatas misal :

- a. Bagaimana cara menyingkat penulisan menggunakan if bersarang?
- b. Bagaimana bentuk umum if bersarang?
- c. Apakah bisa menggunakan case of pada if bersarang?

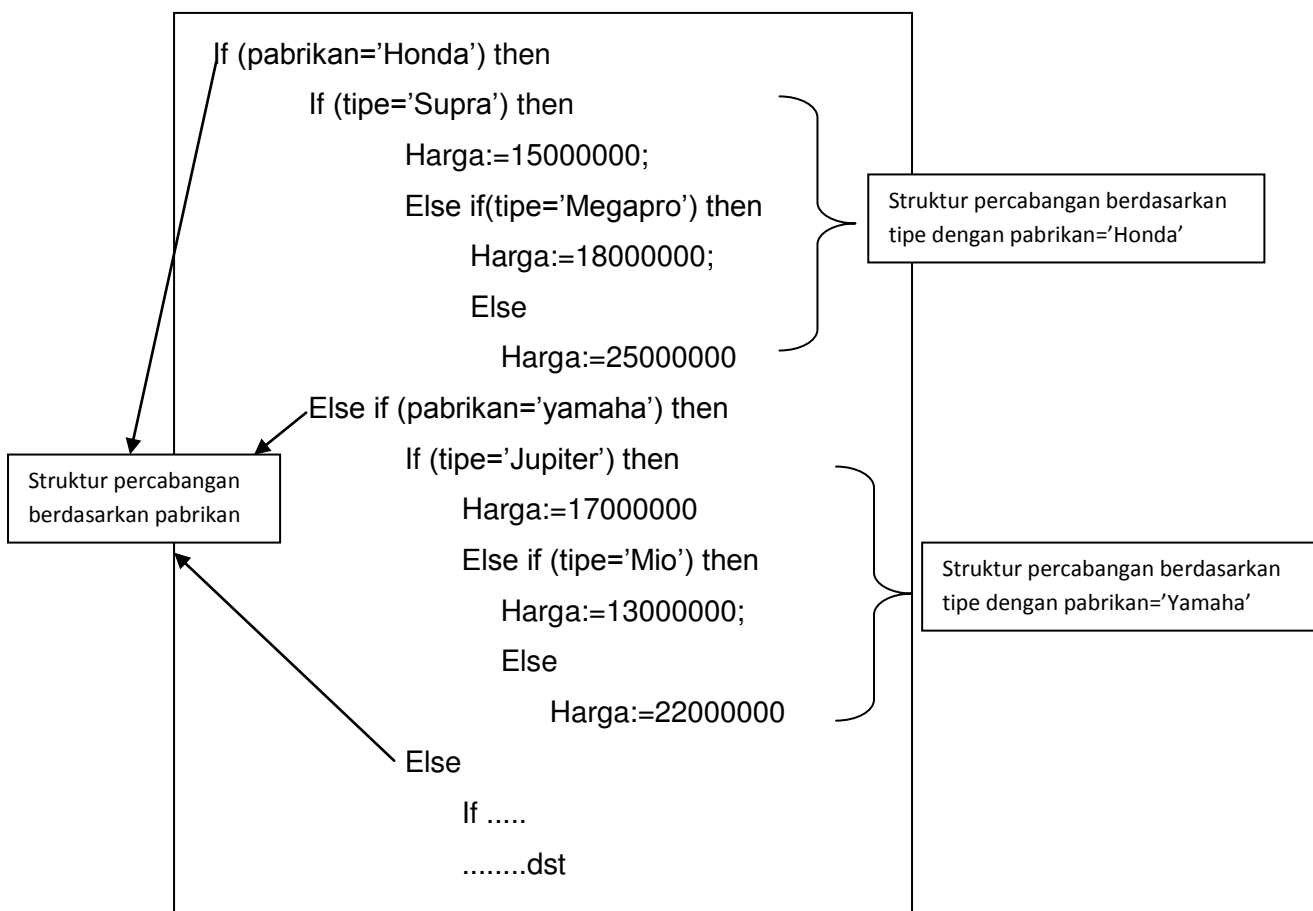
Pembahasan

If bersarang atau nested if adalah percabangan di dalam percabangan. Kita bisa memasukkan struktur if then di dalam struktur if then yang lain sehingga bisa membentuk sesuatu yang lebih kompleks berbeda saat kita hanya menggunakan percabangan biasa. Bentuk umum dari if bersarang adalah sebagai berikut :

```
If (kondisi1) then
    If (kondisi2) then
        Pernyataan1;
    Else
        Pernyataan2;
```

Else**Pernyataan3;**

Perhatikan contoh di atas, kita bisa membagi permasalahan dulu berdasarkan pabrikan, yaitu Honda, Yamaha, dan Suzuki, selanjutnya tiap-tiap pabrikan dipecah lagi berdasarkan jenis motor. Misalkan Honda menjadi Supra, Megapro, dan Tiger; Yamaha menjadi Jupiter, Mio dan Vixion. Dengan demikian kita dapat menuliskan struktur algoritma seperti berikut.



Flowchart dari struktur di atas adalah seperti Gambar 3.1.

Agar lebih praktis, kita dapat menggunakan struktur case of untuk salah satu kategori apakah pabrikan atau tipe atau keduanya. Untuk percabangan case of untuk variabel pabrikan. Pertama tentu kita harus memberikan daftar menu pilihan pada bagian awal seperti berikut.

Jenis-jenis motor berdasarkan pabrikan:

1. Honda
2. Yamaha
3. Suzuki

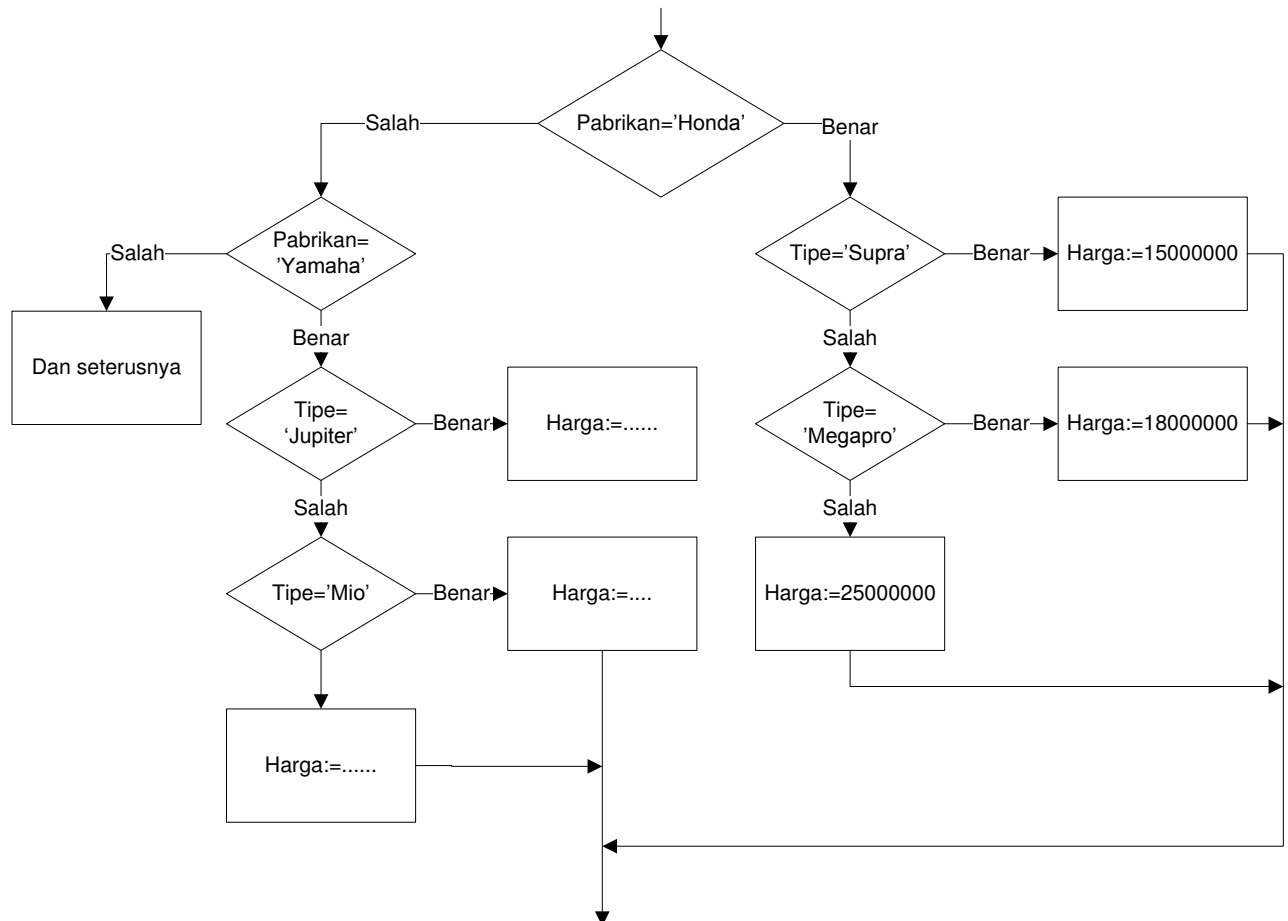
Masukkan nomor pabrikan motor yang akan dibeli: {input dari pengguna, dibaca sebagai pilihanmerk}

Jika user memasukkan angka 1, maka user dihadapkan pada menu tipe motor untuk merk Honda

Tipe Motor merk Honda

1. Supra
2. Megapro
3. Tiger

Masukkan nomor tipe motor yang akan dibeli{input dari pengguna dibaca sebagai pilihantipe}



Gambar 3.1. Flowchart percabangan bersarang

Begitu juga dengan Merk Yamaha dan Suzuki.

Tipe Motor merk yamaha

1. Jupiter
2. Mio
3. Vixion

Masukkan nomor tipe motor yang akan dibeli{input dari pengguna dibaca sebagai pilihantipe}

Tipe Motor merk Suzuki

1. Nex

2. Satria
3. Smash

Masukkan nomor tipe motor yang akan dibeli{input dari pengguna dibaca sebagai pilihantipe}

Selanjutnya dapat dibentuk struktur case of nya sebagai berikut.

```

readln(pilihanmerk);
case pilihanmerk of
  1: begin
    Writeln('.....'); {menu pilihan}
    Readln(pilihantipe);
    Case pilihantipe of
      1: harga:=15000000;
      2: harga:=18000000;
      3: harga:=25000000
    end;
  end;
  2: begin
    Writeln('.....'); {menu pilihan}
    Readln(pilihantipe);
    Case pilihantipe of
      1: harga:=17000000;
      2: harga:=13000000;
      3: harga:=22000000
    end;
  end;
  3: begin
    .....
    .....
    Dst.
  End;
End;

```

Kelebihan dari penggunaan menu adalah bahwa dengan menomori menu maka tingkat kesalhan memasukkan input oleh user dapat diminimalisir. Karena user mengetikkan 'Honda', atau 'HONDA' atau 'honda' dibaca sebagai inputan yang berbeda sehingga bisa jadi dengan maksud

yang sama tapi hasil yang dikehendaki bisa lain. Oleh karena itu penggunaan penomoran menu sangat diperlukan untuk kasus-kasus seperti ini.

Untuk lebih memperjelas pemahaman kalian kerjakan LKS berikut secara berkelompok.

LEMBAR KERJA SISWA

1. Sempurnakan kode program pada permasalahan di atas dengan dua macam struktur if then else dan case of.
2. Buatlah program untuk menentukan besari hadiah lomba lari untuk tiga kategori umur.

Kategori Umur	Juara		
	1	2	3
12-15 tahun	3 juta	2 juta	1,5 juta
16-21 tahun	4 juta	3 juta	2 juta
Di atas 21 tahun	5 juta	4 juta	3 juta

3.4.2.3. Mengasosiasikan

Berdasarkan hasil pembahasan dan pengerjaan LKS di atas, tuliskan kesimpulan mengenai beberapa hal berikut.

1. Kapan percabangan bersarang diperlukan?
2. Bagaimana sintaks percabangan bersarang pada Bahasa Pascal? Struktur apa saja yang dapat digunakan?
3. Apa kelebihan dan kekurangan struktur if then else dan case of untuk percabangan bersarang?

3.4.2.5. Mengkomunikasikan

Presentasikan jawaban kalian di depan kelas, kelompok lain menanggapi. Jalannya diskusi dimimbing oleh Guru.

3.4.3. Rangkuman

1. Percabangan tersarang adalah struktur percabangan di mana ada percabangan di dalam percabangan
2. Percabangan tersarang dapat menggunakan struktur if then else atau case of atau gabungan keduanya

3.4.4. Tugas

1. Misalkan Tabel berikut adalah tabel mengenai besaran pajak kendaraan bermotor dilihat

Jenis Kendaraan	Tahun Keluaran		
	Lebih dari 2012	2008-2012	<2008
Bebek	220 ribu	195 ribu	180 ribu
Matic	225 ribu	205 ribu	190 ribu
Sport	240 ribu	225 ribu	200 ribu

Buatlah program untuk menentukan besarnya biaya pajak berdasarkan tabel di atas.

2. Misalkan di sebuah perusahaan sistem penggajian adalah berdasarkan masa kerja dan golongan. Masa kerja menentukan gaji pokok, sedangkan golongan pegawai menentukan tunjangan jabatan. Gaji pokok menurut Masa kerja adalah sebagai berikut.

Masa Kerja	Gaji Pokok
Kurang dari 5 tahun	2.000.000
6-10 tahun	2.500.000
Lebih dari 10 tahun	3.000.000

Tunjangan jabatan menurut Golongan pegawai adalah sebagai berikut

Golongan	Tunjangan
1	1.000.000
2	1.500.000
3	2.000.000

Selain itu setiap pegawai mendapatkan tunjangan keluarga sebesar 15 % dari gaji pokok.

Susunlah sebuah program yang meminta masukan berupa nama pegawai, masa kerja dan Golongan pegawai dan menampilkan output berupa penerimaan gaji tiap bulan.

3. Carilah masalah yang memerlukan percabangan bersarang, buatlah program untuk menyelesaikannya.

3.4.5. Uji Kompetensi

Buatlah program dengan algoritma tersarang untuk menentukan besar tarif dasar listrik berdasarkan tabel berikut.

Kategori Pemakai	Besaran tarif untuk tiap Daya (per kWh)		
	900 VA	1300 VA	2200 VA
Rumah Tangga	500	700	1000
Perusahaan	800	1000	1200
Fasilitas Umum	400	500	700

Program dibuat dengan meminta inputan berupa nama pelanggan, nomor meteran, kategori, besar daya, jumlah pemakaian (dalam kWh). Outputnya adalah besar tagihan listrik yang harus dibayarkan.

BAB 4

STRUKTUR PERULANGAN

Apa yang terjadi apabila seorang programmer ingin membuat sebuah proses yang sama tapi dalam jumlah yang lebih banyak? Apakah dia harus menuliskannya berulang kali? Ternyata, untuk menyelesaikan masalah ini, kita hanya perlu menggunakan sebuah struktur control pengulangan. Lalu apa fungsi dari struktur control pengulangan itu?

Struktur perulangan atau biasa disebut dengan looping digunakan untuk mengatur proses yang dijalankan secara berulang-ulang. Sehingga programmer tidak perlu menuliskan scriptnya secara berulang kali. Struktur kontrol perulangan memungkinkan kita untuk mengeksekusi sebuah statement atau sekelompok statement dalam jumlah yang banyak.

Pembahasan mengenai apa itu struktur perulangan dalam Algoritma sudah dijelaskan dengan detail di semester 1, termasuk apa itu perulangan, bagaimana bentuk-bentuknya. Pada bab ini kita akan mengimplementasikan algoritma perulangan tersebut dalam kode bahasa Pascal.

4.1. KEGIATAN BELAJAR 1 PERULANGAN DENGAN KONDISI DIAWAL

4.1.1. Tujuan pembelajaran

Siswa memahami konsep dan struktur perulangan dengan kondisi di awal.

4.1.2. Aktivitas Belajar Siswa

4.1.2.1. Mengamati

Coba perhatikan contoh kode program berikut.

```
program whiledo
var i : integer;
begin
    i := 1;
    while i <= 5 do
    begin
        writeln('Pascal');
        i:=i+1;
    end;
end.
```

Program FORDO

```
Var i : integer;  
Begin  
    For i := 1 to 5 Do  
        Writeln('Pascal');  
End.
```

Program NestedLoops; { * contoh aplikasi mencari nilai prima * }

```
var  
    i, j:integer;  
begin  
    for i := 2 to 50 do  
        begin  
            for j := 2 to i do  
                if (i mod j)=0 then  
                    break; { * if faktor ketemu, berarti bukan prima *}  
            if(j = i) then  
                writeln(i , ' adalah nilai prima' );  
        end;  
    end.
```

Ketiga kode program di atas adalah contoh dari perulangan dengan kondisi diawal.

4.1.2.2. Menanya

Dari ketiga contoh kode program di atas, buat pertanyaan yang berkaitan dengannya, misalkan:

1. Apakah yang membedakan 3 source code di atas?
2. Bagaimana cara kerja dari 3 kode program di atas?
3. Lalu, apa hasil eksekusi dari 3 kode program di atas?
4. Ketiga kode program di atas merupakan contoh dari perulangan dengan kondisi di awal, lalu kapan kita akan menggunakannya?

4.1.2.3. Mengumpulkan Informasi/Mencoba

Jika kita lihat sekilas, maka kita bisa menyimpulkan bahwa kode program pertama dan kedua memiliki struktur yang hampir sama, yang membedakan hanya yang pertama menggunakan pernyataan WHILE-DO, sedangkan yang kedua menggunakan pernyataan FOR-DO. Bagaimana dengan yang ketiga? Yang ketiga menunjukkan sebuah contoh penggunaan dari sebuah nested loops (perulangan bersarang/ perulangan di dalam perulangan). Bagaimana cara kerja, hasil, dan kapan ketiga kode program (pernyataan perulangan) tersebut di gunakan? Akan kita bahas pada pembahasan berikut ini.

Sebagaimana yang pernah dijelaskan pada semester lalu bahwa struktur perulangan merupakan struktur yang tidak terpisahkan dengan algoritma dan pemrograman. Struktur perulangan memungkinkan program untuk melakukan serangkaian perintah secara berulang-ulang. Dan untuk memenuhi syarat bahwa algoritma harus finite (terbatas) maka dalam perulangan pasti ada titik pemberhentian. Jika ternyata dalam sebuah kasus perulangan tidak mencapai titik berhenti maka dapat dikatakan algoritma tersebut salah.

Titik pemberhentian dapat diberikan dengan beberapa cara, sebagai berikut.

1. Pemberhentian dengan syarat

Pemberhentian dengan syarat artinya ada sebuah kondisi yang akan menyebabkan perulangan berhenti. Pemberian syarat ini juga dapat dilakukan dengan dua cara, yaitu

- a. Syarat diberikan di awal, di mana selama persyaratan dipenuhi maka dilakukn serangkain perintah
- b. Syarat diberikan di akhir, di mana proses akan diulang-ulang sampai syarat dipenuhi.

Perbedaan antara a dan b adalah ketika syarat diberikan di awal, maka selama persyaratan itu dipenuhi , maka perulangan dilakukan. Jika kondisi sudah tidak dipenuhi maka berhenti. Sedangkan pada kasus syarat di akhir, dikerjakan serangkaian langkah. Setiap selesai rangkaian langkah diperiksa apakah kondisi sudah dicapai, jika belum maka proses diulangi lagi. Struktur yang sesuai dengan perulangan dengan kondisi di awal adalah struktur WHILE DO sedangkan untuk kondisi di akhir adalah dengan REPEAT UNTIL.

2. Pemberhentian dengan pencacah

Pemberhentian dengan pencacah, artinya dari awal sudah ditentukan bahwa perulangan akan dilakukan berapa kali. Pencacah ini juga ada dua macam cara, yaitu

- a. Pencacah naik

Pemberhentian dengan pencacah naik artinya untuk suatu pencacah, misalkan i dari 1 sampai 100 lakukan rangkain langkah x. Artinya langkah tersebut akan diulangi sebanyak seratus kali. Dalam pascal untuk kasus pencacah naik digunakan struktur FOR TO DO.

- b. Pencacah turun

Pemberhentian dengan pencacah turun , artinya sebaliknya untuk suatu pencacah i dari 100 sampai 1 lakukan rangkaian langkah x, artinya langkah x akan diulangi sebanyak seratus kali. Dalam pascal untuk kasus pencacah naik digunakan struktur FOR DOWNT0 DO

Penggunaan beberapa macam pemberhentian di atas tergantung dari situasi dan kondisi, untuk lebih jelasnya akan dijelaskan seperti berikut.

Perulangan dengan struktur WHILE DO

Dalam berbagai bahasa Pascal dikenal struktur WHILE DO, yang berarti selama memenuhi kriteri/kondisi tertentu, maka dilakukan serangkaian proses. Struktur ini dapat diterapkan pada contoh di atas sebagai berikut.

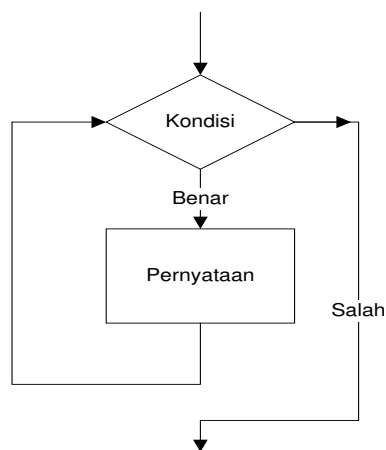
Bentuk Umum:

```
WHILE (kondisi) DO  
  Pernyataan;
```

Atau jika lebih dari satu pernyataan dapat dituliskan

```
WHILE (kondisi) DO  
  Begin  
    Pernyataan1;  
    Pernyataan2;  
    Pernyataan3;  
  End;
```

Perhatikan Flow Diagram berikut:



Untuk lebih jelasnya, perhatikan contoh kode program berikut, beserta dengan hasilnya.

Contoh 1

Kode program:

```
program whileLoop;  
var  
  a: integer;  
begin
```

```
a := 10;
while a < 20 do
begin
  writeln('nilai dari a adalah: ', a);
  a := a + 1;
end;
end.
```

Hasil:

```
nilai dari a adalah: 10
nilai dari a adalah: 11
nilai dari a adalah: 12
nilai dari a adalah: 13
nilai dari a adalah: 14
nilai dari a adalah: 15
nilai dari a adalah: 16
nilai dari a adalah: 17
nilai dari a adalah: 18
nilai dari a adalah: 19
```

Perulangan dengan struktur FOR DO

Perulangan dengan pencacah, yaitu menggunakan struktur FOR DO dimasukkan dalam pembahasan pada kegiatan Belajar ini. Struktur ini digunakan untuk mengulang statement atau satu blok statement berulang kali sejumlah yang ditentukan.

Bentuk umum :

```
for nama_variabel := nilai_awal to nilai_akhir do {untuk pencacah naik}
  Pernyataan;

atau

for nama_variabel := nilai_awal downto nilai_akhir do {untuk pencacah turun}
  Pernyataan;
```

Untuk lebih jelasnya, perhatikan contoh kode program berikut, beserta hasilnya.

Program FORDO:

```
Var I : integer;
Begin
  For I := 1 to 5 Do
    Writeln('Pascal');
```

End.

Output :

Pascal

Pascal

Pascal

Pascal

Pascal

Penulisan pernyataan yang akan diproses berulang kali tersebut dapat juga ditulis dalam bentuk blok pernyataan (diawali dengan BEGIN dan diakhiri dengan END), walaupun hanya berisi sebuah statement saja, sebagai berikut :

Contoh 2

Program Ulang;

Var I : integer;

Begin

For I := 1 to 5 Do

Begin

Writeln('Belajar Pascal');

Writeln ("Sampai Bisa');

End;

End.

Output:

Belajar Pascal

Sampai Bisa

Belajar Pascal

Sampai Bisa

Belajar Pascal

Sampai Bisa

Belajar Pascal

Sampai Bisa

Belajar Pascal

Sampai Bisa

Dalam menyelesaikan suatu permasalahan kita bisa menggunakan salah satu bentuk perulangan, apakah While Do ataupun For Do. Manakah yang lebih baik? Keduanya mempunyai

kelebihan dan kekurangan sendiri-sendiri. Ada permasalahan yang dapat diubah dalam dua bentuk perulangan tapi ada yang lebih mudah menggunakan bentuk For Do, tapi ada juga yang hanya bisa digunakan bentuk While Do.

Perhatikan dua contoh pertama pada bagian mengamati.

program whiledo

```
var i : integer;
begin
    i := 1;
    while i <= 5 do
    begin
        writeln('Pascal');
        i:=i+1;
    end;
end.
```

Program FORDO:

```
Var I : integer;
Begin
    For i:= 1 to 5 Do
        Writeln('Pascal');
End.
```

Kedua program di atas mempunyai keluaran yang sama yaitu mencetak 'Pascal' sebanyak 5 kali, yaitu untuk i=1 sampai 5. Dalam While Do,

- pertama i diberi nilai 1,
- Periksa apakah i <= 5? Iya maka tuliskan 'Pascal' kemudian nilai i ditambah 1 menjadi 2
- Periksa apakah i<=5? Iya , maka tuliskan Pascal, i=i+1=3
- Periksa apakah i<=5? Iya , maka tuliskan Pascal, i=i+1=4
- Periksa apakah i<=5? Iya , maka tuliskan Pascal, i=i+1=5
- Periksa apakah i<=5? Iya , maka tuliskan Pascal, i=i+1=6
- Periksa apakah i<=5?Tidak, maka STOP

Di sini i berfungsi sebagai iterator dan 5 sebagai nilai sentinel.

Dalam struktur FOR DO, iterator dan sentinel diringkas dalam bentuk for i:=1 to 5, di mana i otomatis bertambah naik, mulai dari nilai awal 1 menjadi 5. Oleh karena itu struktur For Do disebut perulangan dengan pencacah.

Contoh berikut kode program dengan dua struktur berbeda untuk masalah yang sama.

Contoh 3.

Buatlah program untuk menentukan suku ke-n dari barisan berikut.

2,7,12,17,22,

Jawab:

Barisan di atas adalah barisan aritmatika di mana selisih tiap suku berturutan adalah 5. Dengan struktur For Do.

Program Barisan;


```
var i,n,suku:integer;
begin
  readln(n);
  suku:=2;
  for i:=2 to n do
    suku:=suku+5;
  writeln('suku ke- ', n, ' adalah : ', suku);
end.
```

Dari kode program di atas dapat dilihat bahwa suku pertama adalah 2, sehingga dinyatakan nilai awal suku:=2. Berikutnya untuk suku kedua, ketiga dan seterusnya (i=2, 3, 4, ..) dihitung nilai suku setelahnya adalah suku sebelumnya ditambah 5 (suku:=suku+5).

Berikut ini programnya dengan struktur WHILE DO.

```
Program Barisan;
var i,n,suku:integer;
begin
  readln(n);
  suku:=2;
  i:=2
  While i<=n do
    Begin
      suku:=suku+5;
      i:=i+1;
    End;
  writeln('suku ke- ', n, ' adalah : ', suku);
end.
```

Contoh 4.

Buatlah program untuk menentukan suku terbesar yang kurang dari n dari barisan 2, 4, 8, 16,

Jawab:

Masalah ini kurang cocok diselesaikan dengan struktur FOR DO karena banyaknya perulangan tidak diketahui dengan pasti sehingga sulit menentukan iterator dan nilai sentinelnya. Dengan menggunakan struktur While Do dapat diperoleh solusi sebagai berikut.

```
Program Barisan2;
var n,suku:integer;
begin
```

```
readln(n);
suku:=2;
  While suku<=n do
    suku:=suku*2;
  suku:=suku/2;
  writeln('suku ke- ', n, ' adalah : ', suku);
end.
```

Misalkan n adalah 50, kita dapat menelusuri program di atas sebagai berikut.

Suku=2, periksa apakah suku<50? Ya, maka suku=suku*2=4

Suku=4, periksa apakah suku<50? Ya, maka suku=suku*2=8

Suku=8, periksa apakah suku<50? Ya, maka suku=suku*2=16

Suku=16, periksa apakah suku<50? Ya, maka suku=suku*2=32

Suku=32, periksa apakah suku<50? Ya, maka suku=suku*2=64

Suku=64, periksa apakah suku<50? STOP, lihat nilai terakhir suku adalah 64, karena ketika suku=32, pasti masih tetap dikalikan 2, sehigga setelah iterasi berakhir maka perlu dibagi lagi dengan 2.

Nested Loops (perulangan bersarang)

Dalam beberapa permasalahan sering dijumpai perulangan di dalam perulangan. Bisa menggunakan berbagai struktur WHILE DO atau FOR DO atau kombinasi dari struktur perulangan yang berbeda-beda.

```
for variabel1:=nilai_awal1 to [downto] nilai_akhir1 do
begin
  for variabel2:=nilai_awal2 to [downto] nilai_akhir2 do
  begin
    statement;
  end;
end;
```

Sedangkan bentuk umum untuk perulangan bersarang **WHILE DO** adalah sebagai berikut

```
while(kondisi1)do
begin
  while(kondisi2) do
  begin
    statement(s);
  end;
  statement(s);
end;
```

Untuk lebih jelasnya, perhatikan contoh kode program berikut, beserta dengan hasilnya.

```
program NestedLoops; {* contoh aplikasi mencari nilai prima *}
var
  i, j:integer;
begin
  for i := 1 to 5 do
  begin
    for j := 1 to i do
      write('*');
    writeln;
  end;
end.
```

Output:

```
*
**
***
****
*****
```

Dalam program di atas diulangi dari i=1 sampai 5.

i=1, ulangi dari j=1 sampai 1 tuliskan '*', kemudian ganti baris (writeln)

i=2, ulangi dari j=1 sampai 2 tuliskan '*' sehingga ada dua '*', kemudian ganti baris (writeln)

i=3, ulangi dari j=1 sampai 3 tuliskan '*' sehingga ada tiga '*', kemudian ganti baris (writeln)

i=4, ulangi dari j=1 sampai 4 tuliskan '*' sehingga ada empat '*', kemudian ganti baris (writeln)

i=5, ulangi dari j=1 sampai 5 tuliskan '*' sehingga ada lima '*', kemudian ganti baris (writeln)

Agar pemahaman kalian jelas mengenai penggunaan struktur perulangan kerjakan LKS berikut.

4.1.2.3. Mencoba/Mengumpulkan Informasi

Lembar Kegiatan Siswa

1. Diberikan algoritma dalam pseudocode sebagai berikut.

Algoritma Kelipatan5

Variabel i, a:integer

i =1 {i sebagai iterator}

a=5

WHILE (a< 50) DO

WRITE a

i=i+1

a=5*i

END

- a. Ubahlah dalam bentuk kode Program menggunakan Pascal
 - b. Tuliskan output dari program yang kalian buat.
2. Buatlah program untuk menentukan suku ke-n dari barisan:
2, 5, 8, 11,
Dengan struktur WHILE DO dan FOR DO.
 3. Buatlah program untuk menampilkan data seperti berikut:

1	2	3	4	5
1	2	3	4	
1	2	3		
1	2			
1				

1				
1	2			
1	2	3		
1	2	3	4	
1	2	3	4	5

(a)

(b)

4.1.2.4. Mengasosiasikan

4.1.2.4. Berdasarkan Lembar Kerja Siswa di atas, buatlah kesimpulan mengenai apa yang kalian pelajari.

1. Perulangan dengan kondisi di awal adalah

.....

.....

2. Di dalam pascal sendiri, terdapat tiga kontrol perulangan dengan kondisi di awal, yaitu:

- a. **FOR DO**, digunakan ketika

.....

.....

- b. **WHILE DO**, digunakan ketika

.....

- c. **NESTED LOOP**, digunakan ketika

.....

4.1.2.5. Mengkomunikasikan

Presentasikan jawaban kalian di depan kelas, murid yang lain menanggapi. Jalannya diskusi dibiimbing oleh Guru.

4.1.3. Rangkuman

1. Struktur perulangan adalah struktur dalam bahasa pemrograman yang mengulangi sejumlah perintah sampai batas yang ditentukan
2. Struktur perulangan memerlukan iterator dan sentinel sebagai titik henti
3. Struktur perulangan dengan kondisi di depan menggunakan struktur **WHILE DO**
4. Struktur perulangan dengan pencacah menggunakan struktur **FOR TO DO** atau **FOR DOWNTONTO DO**.

4.1.4. Tugas

Buatlah kode program sederhana yang menggunakan struktur kontrol perulangan dengan kondisi di awal untuk menyelesaikan suatu masalah, boleh menggunakan struktur kontrol **FOR DO**, **WHILE DO**, atau **NESTED LOOP**. Jelaskan atau presentasikan pada pertemuan yang akan datang.

4.1.5. Uji Kompetensi

1. Perhatikan kode program berikut.

```
Var I : Integer ;
Begin
  For I := 1 To 2 Do
    Write(' Belajar ');
    Writeln('Oke');
  End.
```

Output dari program diatas adalah

a. Belajar Belajar Oke	b. Belajar Oke Belajar Oke
c. Belajar Oke Oke	d. Belajar Oke Belajar
e. Belajar Belajar Oke Oke	

2. Perhatikan program dibawah ini, outputnya adalah

```
Var i,j : byte;
```

```

Begin
  For i := 1 To 3 Do
    Begin
      For j := 3 DownTo i Do
        Write(j);
      writeln;
    End;
  End.

```

a. 123 12 1	b. 321 32 3
c. 1 12 123	d. 3 32 321
e. 321 21 1	

3. Program untuk menghasilkan output : 5 4 3 2 1 adalah.....

a. var i: byte; begin For i:= 5 to 1 Do Write(i); end.	b. var i: integer; begin For i:= 5 downto 1 Do Writeln(i); end.
c. var i: integer; begin For i:= 1 to 5 Do Write(i); end.	d. var i: byte begin For i:= 5 downto 1 Do Write(i); end.
e. var i: integer; begin For i:= 1 to 5 Do Writeln(i); end.	

4. Perhatikan program berikut.

```

Var      a,b : integer;
Begin

```

```

    For a := 1 to 5 Do
      For b:= 1 to a Do
        Write(b);
      End.
    End.

```

Output dari program berikut :

- a. 1 b. 12345 c. 12345 d. 112123123412345
- 12 1234
- 123 123
- 1234 12

5. Output dari program berikut :

```

Var      i : integer;
Begin
  i :=1;
  While i <= 5 Do
    Begin
      If 10 Mod i = 0 Then
        Writeln(10 Div I);
      i := I + 1;
    End;
  End.

```

a. 1 2 5	b. 10 5 2
c. 5	d. 2 5 10
e. 5 2 1	

4.2. KEGIATAN BELAJAR 2 PERULANGAN DENGAN KONDISI DI AKHIR

4.2.1. Tujuan pembelajaran

Dengan Kegiatan Belajar ini diharapkan siswa mampu memahami perulangan dengan kondisi berada di akhir dan menerapkannya untuk menyelesaikan masalah pemrograman.

4.2.2. Aktivitas belajar siswa

4.2.2.1. Mengamati

Perhatikan Source code berikut

PROGRAM REPEATUNTIL

```
var
digit := 1;
begin
    repeat
        write (digit);
        digit := digit + 1
    until digit > 5 ;
end.
```

Program REPEATUNTIL

```
var
    a: integer;
begin
    a := 10;
    (* repeat until loop execution *)
    repeat
        writeln('nilai dari a adalah ', a);
        a := a + 1
    until a = 20;
end.
```

Kedua contoh di atas adalah contoh kode program dari struktur kontrol yang akan dibahas selanjutnya

4.2.2.1. Menanya

Dari kedua contoh kode program di atas, buat pertanyaan yang berkaitan dengan keduanya, misalkan:

1. Apakah hasil/output dari kode program diatas?
2. Bagaimana proses dari kode program diatas sehingga menghasilkan hasil/output tersebut?

Hasil/output, serta proses dari kode program di atas akan di jelaskan pada pembahasan berikut ini

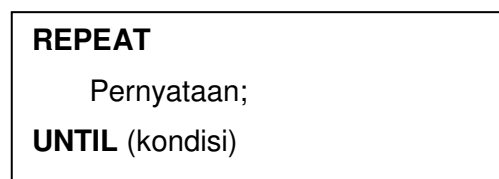
4.2.2.3. Mengumpulkan Informasi

Sebagaimana yang dijelaskan pada KB sebelumnya bahwa perulangan dapat dilakukan dengan menyatakan kondisi di akhir. Berbeda dengan kondisi di awal, di mana selama syarat (kondisi) dipenuhi maka perintah dijalankan, pada kondisi di akhir, perintah dijalankan sampai memenuhi kondisi yang diinginkan. Struktur yang digunakan adalah struktur Repeat Until.

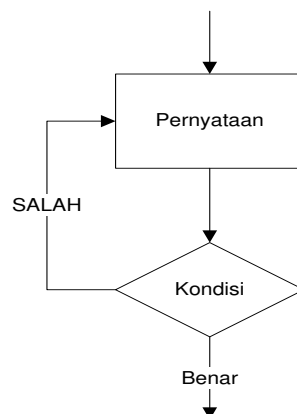
REPEAT UNTIL

Digunakan untuk mengulang (repeat) statement atau blok statement sampai (until) kondisi yang diseleksi di Until tidak terpenuhi.

Bentuk Umum :



Coba perhatikan flow diagram berikut ini



Untuk lebih jelasnya, perhatikan contoh kode program berikut, beserta dengan hasilnya.

Contoh 1.

Kode Program

```
Var digit := 1;  
begin  
    repeat  
        writeln (digit);  
        digit := digit + 1  
    until digit > 5 ;  
end.
```

Output:

1

2
3
4
5

Program di atas diawali dengan nilai awal digit=1, kemudian dilakukan perintah

```
writeln (digit);
```

```
digit := digit + 1
```

hingga nilai digit>5.

Contoh 2.

Kode program:

```
var
```

```
  a: integer;
```

```
begin
```

```
  a := 10;
```

```
  repeat
```

```
    writeln('nilai dari a adalah ', a);
```

```
    a := a + 2
```

```
  until a = 20;
```

```
end.
```

Hasil:

nilai dari a adalah 10

nilai dari a adalah 12

nilai dari a adalah 14

nilai dari a adalah 16

nilai dari a adalah 18

nilai dari a adalah 20

Bagaimana jika kondisi diganti dengan a=21? Smaapi kapanpun syarat ini tidak akan dipenuhi karena nilai dari a pasti genap, sedangkan 21 ganjil. Dengan demikian terjadi perulangan terus menerus.

Lalu,kapan kita akan menggunakan Perulangan dengan kondisi di awal, dalam hal ini adalah **WHILE DO** dengan Perulangan dengan kondisi di akhir yaitu **REPEAT UNTIL**, adalah

Perbedaan antara struktur REPEAT-UNTIL dengan WHILE-DO adalah sebagai berikut :

- 1) Paling sedikit statement-statement di dalam perulangan REPEAT-UNTIL diproses sekali, karena seleksi kondisi ada pada statement Until yang terletak dibawah. Sedang pada struktur WHILE-DO paling sedikit dikerjakan nol kali, karena seleksi kondisi ada pada statement While

2) Pada REPEAT-UNTIL dapat tidak dipergunakan blok statement (tidak diperlukan Begin dan End untuk menunjukkan batas perulangannya), karena batas perulangannya sudah ditunjukkan oleh Repeat sampai dengan Until (Repeat dan Until sebagai pengganti Begin dan End).

LEMBAR KERJA SISWA

- Jawab:

[illegible]

- Jawab:

[illegible]

.....

4.2.2.4. Mengasosiasikan

Berdasarkan Lembar Kerja Siswa di atas, buatlah kesimpulan mengenai apa yang kalian pelajari.

1. Perulangan dengan kondisi di akhir adalah
2. **Cara Kerja Struktur REPEAT UNTIL**, adalah.....

4.2.2.5. Mengkomunikasikan

Presentasikan jawaban kalian di depan kelas, kelompok lain menanggapi. Jalannya diskusi dibiimbing oleh Guru.

4.2.3. Rangkuman

1. Perulangan dengan kondisi di akhir maksudnya adalah perulangan sejumlah perintah hingga suatu kondisi dipenuhi
2. Perulangan dengan kondisi di akhir menggunakan struktur REPAT UNTIL.

4.2.4. Tugas

- 1) Buatlah program menu tampilan login dengan memasukkan PIN dengan menggunakan struktur kontrol **REPEAT-UNTIL** dengan syarat apabila user salah memasukkan PIN maka program menu akan tampil kembali. Tampilan menu seperti berikut:

Login Account

PIN anda :

- 2) Buatlah kode program sederhana untuk menentukan apakah bilangan yang diinputkan ganjil atau genap. User dapat melakukan perhitungan ulang hingga 5 kali. Gunakan struktur Repeat Until.

4.2.5. Uji Kompetensi

1. Looping pada WHILE DO dikerjakan minimal ... kali, sedangkan pada REPEAT...UNTIL dikerjakan minimal kali.

- a. 1 0
- b. 1 1
- c. 0 1
- d. 0 0
- e. 2 2

2. Perhatikan Kode Program berikut

Var a: integer;

```

begin
  a := 2;
  repeat
    write(a, ' ');
    a := 2
  until a = 3;
end.

```

Output dari program di atas adalah

- a. 2 2 2
- b. 2
2
2
- c. 2 2 2
2 2 2
2 2 2
- d. 2
- e. Infinite Loops

3. Perhatikan Kode Program berikut.

```

var i : integer;
begin
  i:=0;
  repeat
    i:=i+1;
    writeln(i);
  until i = 5;
  readln;
end.

```

Output dari program di atas adalah

- | | |
|--------------|----------------|
| a. 1 2 3 4 5 | d. 0 1 2 3 4 5 |
| b. 1 | e. 0 |
| 2 | 1 |
| 3 | 2 |
| 4 | 3 |
| 5 | 4 |
| | 5 |
| c. 5 | |

4. Perhatikan kode program berikut.

```
Var i: integer;  
Begin  
    I:=1;  
    Repeat  
        Write('ini yang ke ',i);  
        I:=i+1;  
    Until i=2;  
End.
```

Output dari program di atas berupa

- a. Ini yang ke 1
- b. Ini yang ke 2
- c. Ini yang ke 1
 Ini yang ke 2
- d. Ini yang ke 1 ini yang ke 2
- e. tidak ada output

5. Perhatikan kode program berikut

```
var a: integer;
```

```
begin
```

```
    a:=5;  
    repeat  
        write(a,' ');  
        a:=a+5;  
    until a>25;
```

```
end.
```

Output dari program diatas berupa...

- | | |
|-------------------|------------------|
| a. 5 10 15 20 25 | d. 25 20 15 10 5 |
| b. 5 | e. 25 |
| 10 | 20 |
| 15 | 15 |
| 20 | 10 |
| 25 | 5 |
| c. Infinite Loops | |

4.3. KEGIATAN BELAJAR 3 PERULANGAN DENGAN KONDISI DIINPUTKAN USER

4.3.1. Tujuan Pembelajaran

Siswa memahami perulangan dengan kondisi diinputkan user dan menggunakannya untuk menyelesaikan masalah pemrograman.

4.3.2. Aktivitas Kegiatan Belajar

Dalam bab ini, hampir sama dengan bab-bab sebelumnya. Tetapi, yang membedakan adalah dalam bab ini, kita akan belajar bagaimana caranya kita akan menggunakan perulangan dengan kondisi sesuai dengan apa yang akan di inputkan oleh si user. Sehingga nilai yang akan di hasilkan pun akan dinamis atau menyesuaikan.

4.3.2.1. Mengamati

Perhatikan Program berikut.

```
Program UlangVolumeBalok;
Uses crt;
Var   p,l,t, V:integer;
      pilihan:char;
Begin
  Clrscr;
  Repeat
    Readln(p);
    Readln(l);
    Readln(t);
    V:= p*l*t;
    writeln (v);
    Write ('ingin menghitung lagi y/n');
    Readln (pilihan);
  until (pilihan=="n");
  readln;
End.
```

Program ini diambil dari semester yang lalu mengenai algoritma perulangan dengan kondisi dari user.

4.3.2.2. Menanya

Dari contoh diatas, buat pertanyaan seperti contoh berikut.

1. Tentukan apa maksud dari program atas!
2. Kapan program berhenti?
3. Variabel apakah yang menjadi iterator?
4. Berapa kali user bisa menghitung volume balok?

4.3.2.3. Mencoba/Mengumpulkan Informasi

Aplikasi pemrograman dalam kehidupan nyata, khususnya yang terkait dengan struktur perulangan, memerlukan syarat yang diinputkan oleh pengguna. Sebagai contoh, dalam sebuah aplikasi kasir, ketika melayani pembeli, seorang kasir akan memasukkan data barang-barang yang dibeli dengan barcode reader, hingga akhirnya kasir akan mengklik sebuah tombol untuk menghitung total belanjaan. Proses memasukkan data dari barang pertama, kedua, dan ke barang-barang berikutnya, adalah suatu bentuk perulangan. Dan perulangan ini akan berhenti hingga akhirnya kasir mengeklik tombol untuk menghitung total belanjaan.

Contoh lain misalkan dalam sebuah video game dengan koin, ketika seorang anak memasukkan koin, kemudian bermain dan akhirnya terjadi game over. Berikutnya biasanya, video game akan menampilkan hitungan mundur menunggu anak untuk memasukkan koin berikutnya agar permainan dapat dilanjutkan. Bentuk hitungan mundur ini juga menggunakan perulangan, dan akan berhenti perulangan ini untuk dua kemungkinan, si anak memasukkan koin atau perhitungan mundur selesai dan akhirnya masuk ke menu awal game kembali.

Kondisi yang diinputkan dari user bisa diperoleh sebelum perulangan atau di dalam perulangan itu sendiri. Contoh di atas adalah kondisi dimasukkan oleh user di dalam perulangan itu sendiri. Contoh pengambilan kondisi sebelum perulangan adalah menentukan suku ke- n dari sebuah barisan. Misalkan menentukan suku ke n dari barisan 1,2,3,5,8, Nilai n diperoleh dengan menerima masukan dari user sebelum perulangan dilakukan.

Diskusikan LKS berikut agar lebih jelas.

Lembar Kerja Siswa

1. Tuliskan kode program untuk permasalahan berikut
 - a. Menuliskan semua suku dari barisan 2,4,6,8... hingga suku ke-n
 - b. Mencari suku terbesar dari barisan 1,2,4,7,... yang kurang dari n

Jalankan program yang telah kalian buat!

Jawab :

[illegible]

[illegible]

- Perhitungan biaya terdiri dari

- Biaya Beban: Rp 15.000,-
- Biaya pemakaian dengan aturan 10 m³ pertama, biayanya Rp. 1200,- / m³ Selebihnya dikenai biaya Rp 2000/m³.

Selanjutnya, program meminta input pengguna apakah akan menghitung lagi atau tidak, hingga akhirnya pengguna menekan “Y” untuk keluar dari algoritma.

Jawab:

This image shows a single sheet of white paper with horizontal blue ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

3. Buatlah program untuk menampilkan angka sebagai berikut:

1			
2	2		
3	3	3	
n.....n			

dengan nilai n, sesuai dengan masukkan dari user

Jawab:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

4.3.2.4. Mengasosiasikan

Berdasarkan Lembar Kerja Siswa di atas, buatlah kesimpulan mengenai apa yang kalian pelajari mengenai maksud perulangan dengan kondisi diinputkan user. Jelaskan mengapa struktur seperti ini penting di dunia pemrograman.

4.3.2.5. Mengkomunikasikan

Presentasikan jawaban kalian di depan kelas, murid yang lain menanggapi. Jalannya diskusi dibimbing oleh Guru.

4.3.3. Rangkuman

1. Dalam struktur perulangan , kondisi dapat ditentukan oleh user
2. Kondisi dapat diperoleh dari user sebelum perulangan atau di dalam perulangan itu sendiri.

4.3.4. Tugas

1. Buatlah sebuah program untuk menentukan suku ke-n dari barisan 2, 5, 8, 11, 14,

2. Buatlah sebuah program untuk menginputkan data mengonversi suhu dalam celcius ke Reamur, fahrenheit dan Kelvin. Dalam algoritma tersebut pengguna dapat melakukan perhitungan secara berulang-ulang hingga mengehendaki untuk mengakhiri perhitungan.
3. Tentukan jumlahan dari deret matematika berikut
 $2+4+8+16+\dots+2^n$
 Dengan n ditentukan oleh pengguna.
4. Carilah sebuah permasalahan di sekitarmu yang membutuhkan perulangan dengan kondisi yang diinputkan user. Buatlah program untuk menyelesaikannya

4.3.5. Uji Kompetensi

Pilihlah jawaban yang benar dengan memberi tanda silang pada huruf a, b, c, d, atau e.

1. Perhatikan kode program dibawah ini, jika $n=5$ maka outputnya adalah

```

Var i,n,bil : integer;
Begin
    Readln(n);
    bil:=1;
    For i := 1 to n Do
        bil := bil * i;
    Writeln(bil);
End.

```

Output dari program di atas adalah.....

- a. 120
 - b. 16
 - c. 24
 - d. 5
 - e. 1
2. Perhatikan kode program berikut.
- ```

var i,j,jml: byte;
begin
 write('jumlah bintang: '); readln(jml);
 j:=0;
 repeat
 i:=0;
 j:=j+1;
 repeat
 write('*');
 i:=i+1;
 until i=j;

```

```

 writeln;
 until j=jml;
end.

```

Output dari program di atas jika jml = 3 adalah

- |                |        |
|----------------|--------|
| a. *           | b. *** |
| **             | ***    |
| ***            | ***    |
| c. ***         | d. *   |
| **             | *      |
| *              | *      |
| e. Salah semua |        |

3. Perhatikan kode program berikut.

```

Var l, n: integer;
Begin
 Write('banyaknya pengulangan ?');
 readln(n);
 i:=1;
 repeat
 begin
 write(i);
 i:=i+1;
 end;
 until (i<=n)
end.

```

Output dari program di atas apabila n=0 adalah

|        |         |
|--------|---------|
| a. 1   | b. 12   |
| c. 123 | d. 1234 |
| e.     |         |

4. Perhatikan kode program berikut.

```

Var i,n: integer;
Begin
 Write('banyaknya pengulangan ?');
 readln(n);
 for i:=1 to n do
 write(i);
 end;
end.

```

end.

Output program di atas jika  $n=0$  adalah

|          |       |
|----------|-------|
| a. 1     | b. 12 |
| c. 123   | d.    |
| e. ERROR |       |

5. Berikut ini pernyataan yang benar mengenai perulangan dengan kondisi akhir diinputkan pengguna, kecuali ....
- Kondisi dapat di awal ataupun di akhir perulangan
  - Inputan dari pengguna sebagai sentinel dapat diberikan sebelum perulangan
  - Inputan pengguna sebagai sentinel dapat diberikan di dalam perulangan
  - Inputan pengguna sebagai sentinel dapat diberikan setelah perulangan selesai
  - Dapat menggunakan REPEAT UNTIL

#### 4.4. KEGIATAN BELAJAR 4 PERULANGAN DENGAN PERNYATAAN BREAK & CONTINUE

##### 4.4.1. Tujuan Pembelajaran

Siswa memahami penggunaan pernyataan break dan continue dalam perulangan dengan bahasa Pascal.

##### 4.4.2 Aktivitas kegiatan Belajar

###### 4.4.2.1. Mengamati

Coba perhatikan contoh kode program berikut

```
Program Breaking;
var
 a: integer;
begin
 a := 10;
 while a < 20 do
 begin
 writeln('nilai dari a adalah ', a);
 a:=a +1;
 if(a > 15) then
 break; {menghentikan perulangan}
 end;
end
```

```
program Lanjut;
var a: integer;
begin
 a := 10;
 repeat
 if(a = 15) then
 begin
 a := a + 1;
 continue;
 end;
 writeln('nilai a adalah ', a);
 a := a+1;
 until (a = 20);
end.
```

Kedua Kode program diatas adalah contoh penggunaan struktur kontrol yang akan kita bahas berikut ini

#### 4.2.2.2. Menanya

Dari kedua contoh diatas, buat pertanyaan yang berkaitan dengan keduanya, misalkan:

1. Apakah yang membedakan kedua kode program diatas?
2. Apakah hasil/output pada kedua kode program tersebut?
3. Bedakan antara kode program diatas dengan kode program yang tidak menggunakan pernyataan seperti di atas!

#### 4.4.2.3. Mengumpulkan Informasi/Mencoba

Perbedaan dari kedua kode program di atas adalah pada penggunaan pernyataan BREAK dan CONTINUE. Walaupun pada dasarnya hasil/output nya sama, tetapi pada penggunaan time execution (waktu yang dibutuhkan untuk eksekusi) lebih cepat dibandingkan dengan yang tidak menggunakan kedua pernyataan tersebut.

Di dalam pascal sendiri kedua pernyataan tersebut di sampaikan dalam bentuk sebagai berikut

##### 1) **BREAK**

Pernyataan **BREAK** di dalam Pascal memiliki 2 buah kegunaan, yaitu

1. Pernyataan break yang ditemukan di dalam sebuah perulangan, otomatis akan menghentikan kontrol perulangan tersebut dan kontrol program selanjutnya akan dilanjutkan pada pernyataan setelah perulangan
2. Pernyataan break juga dapat digunakan untuk menghentikan pernyataan case of yang sudah dibahas pada bab sebelumnya

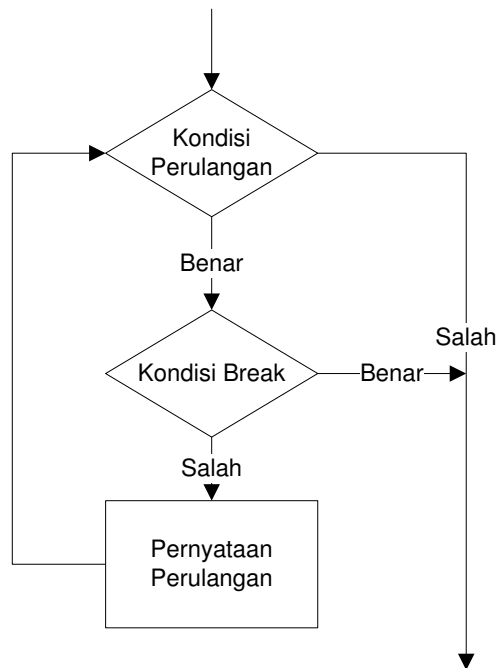
Bentuk umum:

    If (kondisi)

        Break;

Coba perhatikan flow diagram berikut, yaitu penggunaan break pada struktur WHILE DO.





Untuk lebih jelasnya, perhatikan contoh kode program berikut, beserta hasilnya berikut.

#### Contoh 1.

##### Kode program:

```

var
 a: integer;
begin
 a := 10;
 (* while loop execution *)
 while a < 20 do
 begin
 writeln('nilai dari a adalah ', a);
 a:=a +1;
 if(a > 15) then
 (* menghentikan proses looping menggunakan pernyataan break *)
 break;
 end;
 end
 end
end

```

##### Hasil:

```

nilai dari a adalah 10
nilai dari a adalah 11
nilai dari a adalah 12
nilai dari a adalah 13

```

nilai dari a adalah 14

nilai dari a adalah 15

Perulangan di atas akan dihentikan ketika  $a > 15$  sehingga yang ditampilkan

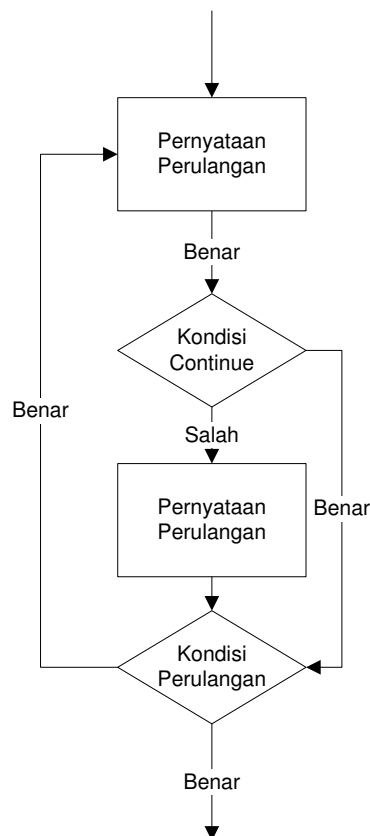
## 2) CONTINUE

Pernyataan continue di dalam Pascal bekerja seperti halnya pernyataan break. Break digunakan untuk menghentikan secara paksa sebuah proses, tetapi continue secara paksa akan melanjutkan perulangan selanjutnya, dan mengabaikan kode-kode diantara keduanya.

Bentuk umum:

```
Continue;
```

Coba perhatikan flow diagram berikut, yaitu penggunaan continue pada struktur repeat until.



Untuk lebih jelasnya, perhatikan contoh kode program berikut, beserta hasilnya

Contoh 2.

**Kode program:**

```
var
 a: integer;
begin
```

```

a := 10;
(* repeat until loop execution *)
repeat
 if(a = 15) then
 begin
 a := a + 1;
 continue; {langsung menuju ke repeat}
 end;
 writeln('nilai a adalah ', a);
 a := a+1;
until (a = 20);
end.

```

**Hasil:**

```

nilai a adalah 10
nilai a adalah 11
nilai a adalah 12
nilai a adalah 13
nilai a adalah 14
nilai a adalah 16
nilai a adalah 17
nilai a adalah 18
nilai a adalah 19

```

Kerjakan lembar kerja siswa berikut secara berkelompok.

### LEMBAR KERJA SISWA

1. Buatlah program komputer menggunakan perulangan dengan perintah continue untuk menampilkan:
  - a. semua bilangan yang bukan kelipatan 3 dari 1-30
  - b. menjumlahkan semua yang bukan bilangan kelipatan 2 atau 5 dari 1 sampai n (diinputkan) user dengan pernyataan:

Jawab:

a. ....

.....

.....

.....

[illegible]

- Jawab:

This image shows a full page of white paper with horizontal dashed lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

#### 4.4.2.4. Mengasosiasikan

Berdasarkan Lembar Kerja Siswa di atas, buatlah kesimpulan mengenai apa yang kalian pelajari.

1. Pernyataan **Break** digunakan ketika

.....  
.....

2. Pernyataan **Continue** digunakan ketika

.....  
.....

#### 4.4.2.5. Mengkomunikasikan

Presentasikan jawaban kalian di depan kelas, kelompok lain menanggapi. Jalannya diskusi dibiimbing oleh Guru.

#### 4.4.3. Rangkuman

1. Pernyataan break yang ditemukan di dalam sebuah perulangan, otomatis akan menghentikan kontrol perulangan tersebut dan kontrol program selanjutnya akan dilanjutkan pada pernyataan setelah perulangan
2. Pernyataan continue secara paksa akan melanjutkan perulangan selanjutnya, dan mengabaikan kode-kode diantara keduanya.

#### 4.4.4. Tugas

Carilah sebuah permasalahan sederhana yang memerlukan perulangan dengan pernyataan break atau continue di dalamnya, masing-masing 1.

#### 4.4.5. Uji Kompetensi

Buatlah program berikut:

1. Menuliskan semua bilangan dari 1 sampai n yang bukan kelipatan 3 atau 7 menggunakan perulangan dengan perintah continue
2. Memasukkan inputan nama, alamat, Sekolah, Umur, dan menampilkannya secara berulang-ulang menggunakan perulangan dengan perintah break.

## BAB 5

## PENGEMBANGAN ALGORITMA APLIKASI

Apa yang sudah kita pelajari dari semester 1 hingga semester 2, adalah meletakkan dasar-dasar kemampuan pemecahan masalah dengan algoritma dan pemrograman. Dengan menguasai dasar-dasar algoritma dan bahasa pemrograman menjadi suatu langkah untuk dapat mengembangkan program atau aplikasi seperti aplikasi kasir, rental, perpustakaan, sistem informasi sekolah, dan sebagainya.

Pada bab terakhir ini kalian akan dihadapkan pada berbagai permasalahan yang erat kaitannya dalam algoritma pembuatan aplikasi dan bagaimana implementasinya menggunakan bahasa Pemrograman. Jadi dalam membuat aplikasi, harus diperhatikan urutan langkah dan strategi mulai melakukan analisis terhadap permasalahan, merumuskan kebutuhan, membuat desain, mengimplementasikan dengan bahasa pemrograman, dan melakukan tes atau uji coba terhadap program yang dihasilkan.

### 5.1. KEGIATAN BELAJAR 1 PENGEMBANGAN APLIKASI DAN ANALISIS PEMECAHAN MASALAH

#### 5.1.1. Tujuan Pembelajaran

1. Siswa memahami langkah-langkah pengembangan aplikasi
2. Siswa memahami langkah-langkah analisis pemecahan permasalahan

#### 5.1.2. Aktivitas Kegiatan belajar

##### 5.1.2.1. Mengamati

Perhatikan Tarif Dasar Listrik pada Gambar 5.1 dan struk pembayaran listrik pada Gambar

##### 5.1.2.2. Menanya

Buatlah pertanyaan terkait dengan hasil pengamatan seperti contoh

1. Bagaimana cara perhitungan tagihan listrik?
2. Informasi apa saja yang dicantumkan pada struk rekening listrik
3. Apa saja yang diperlukan untuk membuat aplikasi pembayaran tagihan listrik
4. Bagaimana mengembangkan algoritma/aplikasi dari permasalahan nyata seperti pembayaran rekening listrik?

**TARIF TENAGA LISTRIK UNTUK KEPERLUAN RUMAH TANGGA  
BERLAKU MULAI 1 NOVEMBER 2014**

| NO. | GOL. TARIF | BATAS DAYA          | REGULER                    |                                                                                                               | PRA BAYAR (Rp/kWh) |
|-----|------------|---------------------|----------------------------|---------------------------------------------------------------------------------------------------------------|--------------------|
|     |            |                     | BIAYA BEBAN (Rp/kVA/bulan) | BIAYA PEMAKAIAN (Rp/kWh)                                                                                      |                    |
| 1.  | R-1/TR     | s.d.450 VA          | 11.000                     | Blok I : 0 s.d. 30 kWh : 169<br>Blok II : di atas 30 kWh s.d. 60 kWh : 360<br>Blok III : di atas 60 kWh : 495 | 415                |
| 2.  | R-1/TR     | 900 VA              | 20.000                     | Blok I : 0 s.d. 20 kWh : 275<br>Blok II : di atas 20 kWh s.d. 60 kWh : 445<br>Blok III : di atas 60 kWh : 495 | 605                |
| 3.  | R-1/TR     | 1.300 VA            | *)                         | 1.352                                                                                                         | 1.352              |
| 4.  | R-1/TR     | 2.200 VA            | *)                         | 1.352                                                                                                         | 1.352              |
| 5.  | R-2/TR     | 3.500 s.d. 5.500 VA | *)                         | 1.352                                                                                                         | 1.352              |
| 6.  | R-3/TR     | 6.600 VA keatas     | *)                         | 1.352                                                                                                         | 1.352              |

Catatan :  
\*) Diterapkan Rekening Minimum (RM):  
RM1 = 40 (Jam Nyala) x Daya Tersambung (kVA) x Biaya Pemakaian.

Gambar 5.1. Tarif dasar Listrik

**BPR KS**  
**STRUK PEMBAYARAN TAGIHAN LISTRIK**

IDPEL : 536440031179  
 Nama : NYAI  
 Tarif/Daya : R1 / 000000450 VA  
 BL/TH : Jan10, Feb10  
 Stand Meter : 00820400 - 00837400  
 Non Subsidi : Rp. 0  
 RPTAG PLN : Rp. 71.382  
 MLPO Ref : 1011002618010066254354

PLN menyatakan struk ini sebagai bukti pembayaran yang sah.  
 Mohon disimpan

Admin Bank : Rp. 4.000  
 Total Bayar : Rp. 75.382

Gambar 5.2. Struk Pembayaran Tagihan Listrik

### 5.1.2.3. Mengumpulkan Informasi/Mencoba

Dalam membangun suatu program atau aplikasi komputer, diperlukan strategi dan langkah-langkah sehingga program dapat berjalan seperti yang diharapkan oleh pengguna. Langkah-langkah inilah yang menjadi strategi bagaimana masalah dapat dipecahkan. Di sinilah peran penting

algoritma dalam pengembangan aplikasi. Namun demikian sebelum membuat algoritma atau langkah-langkah penyelesaian masalah maka harus terlebih dahulu diketahui apa permasalahan dan apa saja yang dibutuhkan sebelum memecahkan masalah. Ketika langkah-langkah penyelesaian masalah sudah diperoleh maka langkah berikutnya adalah mengimplementasikan dalam bentuk kode program sehingga menghasilkan program. Program pun juga harus diuji cobakan terlebih dahulu sehingga terhindar dari kesalahan-kesalahan baik kesalahan teknis ataupun algoritmis. Program yang sudah lolos uji coba dapat digunakan oleh pengguna, dengan tetap senantiasa evaluasi kesesuaian program dengan kebutuhan pengguna.

Dari uraian singkat di atas, menggambarkan bahwa dalam mengembangkan aplikasi tidak sekedar koding dan koding tapi perlu strategi dan langkah-langkah yang terstruktur.

Untuk menyusun sebuah komputer diperlukan langkah-langkah berikut.

Untuk menyusun sebuah program yang besar dan kompleks, pemrogram membutuhkan tahapan penyusunan yang sistematis dan terpadu, yaitu:

1. Definisi Masalah

Pada tahap ini programmer harus mengetahui masalah yang akan diselesaikan, apa tujuan dan kegunaan program yang akan dibuat. Untuk itu diperlukan pengumpulan data baik lewat wawancara terutama pengguna program, dan ketersediaan sumber daya yang akan digunakan dalam pengembangan dan penerapan program yang akan dibuat. Sebagai contoh program pembayaran rekening listrik di atas. Program harus bisa otomatis menghitung besarnya tarif listrik berdasarkan kelas pengguna, batas daya dan besarnya penggunaan daya.

2. Analisis Kebutuhan

Berdasarkan data yang diperoleh mengenai masalah yang akan diselesaikan, langkah selanjutnya adalah mengumpulkan kebutuhan untuk pengembangan program. Untuk bisa menghitung besar tarif listrik, pengguna harus menyiapkan tabel penentuan TDL berdasarkan ketentuan dan cara perhitungannya sehingga bisa menyusun algoritmanya. Selain itu kebutuhan-kebutuhan yang mendukung fungsi tujuan juga harus diperhatikan, misalkan bagaimana hasil perhitungan harus ditampilkan, apakah perlu dicetak atau tidak, atau misalkan data pelanggan dan pemakaian daya harus diambil langsung ke database PLN. Maka ini kebutuhan yang harus disiapkan agar program dapat berjalan.

3. Penyusunan Algoritma

Jika kebutuhan program sudah teridentifikasi, langkah selanjutnya adalah menyusun algoritma atau langkah-langkah penyelesaiannya. Hal ini penting untuk memudahkan penyusunan program. Tahap ini juga disebut sebagai tahap desain. Desain program



ini dapat disajikan dalam bentuk *flowchart* sehingga mudah oleh *programmer* untuk menuangkannya ke dalam bahasa Pemrograman.

#### 4. Pengkodean/Pemrograman

Setelah proses desain atau penyusunan algoritma dilakukan, tahap selanjutnya adalah menuangkannya ke dalam bahasa pemrograman. Bahasa Pemrograman dipilih sesuai dengan hasil analisis kebutuhan program. Jika program memerlukan database untuk menyimpan, mengakses informasi, atau program harus mempunyai tampilan menarik dengan berbagai tombol dan menu, maka bahasa Pascal tidak cocok untuk itu karena masih sangat sederhana. Untuk itu diperlukan pemrograman visual seperti Delphi, VB net, dan sebagainya.

#### 5. *Testing* dan *Debugging*

Setelah proses pengkodean dilakukan langkah selanjutnya adalah melakukan uji coba (*testing*) dan *debugging*. Langkah uji coba ini penting untuk mengetahui apakah program yang telah dihasilkan memberikan solusi yang benar sesuai dengan yang diharapkan. Jika belum maka perlu proses *debugging* untuk mengetahui di mana letak kesalahannya dan bagaimana cara memperbaikinya.

#### 6. Pemeliharaan

Program yang sudah berhasil diujicobakan kemudian diaplikasikan oleh pengguna. Selama program digunakan kekurangan-kekurangan dicatat dan digunakan untuk memperbaiki program yang sedang dijalankan. Pada proses ini modifikasi program dapat dilakukan untuk memperbaiki kesalahan dan peningkatan performance.

Langkah-langkah pengembangan ini dikenal dengan Model Waterfall (air Terjun)

### Analisis dan Desain Algoritma

Algoritma yang baik tidak hanya sekedar benar dalam menentukan solusi, namun juga harus efektif dan efisien. Kriteria efektif berdasarkan seberapa bagus (optimal) penyelesaian yang dihasilkan, sedangkan efisien ini biasanya terkait dengan waktu eksekusi untuk menyelesaikan suatu masalah. Dua buah algoritma memberikan hasil yang benar untuk suatu permasalahan. Salah satu algoritma disebut lebih efisien jika waktu yang diperlukan untuk mengakses proses lebih sedikit dibandingkan dengan yang lain. Namun demikian dengan kecanggihan komputer sekarang, selisih waktu eksekusi biasanya sangat kecil sehingga seakan-akan sama, padahal berbeda seper sekian detik.

Sebagai contoh kalian bisa membandingkan dua buah mesin pencari. Mesin cari bisa kita katakan efektif jika konten yang diberikan benar-benar sesuai dengan harapan user. Sedangkan efisien terkait dengan berapa lama waktu pencarian.

Untuk suatu permasalahan yang rumit ada beberapa strategi yang diterapkan. Strategi-strategi ini dituangkan dalam desain algoritma. Ada beberapa strategi teknik pemecahan masalah dalam algoritma, di antaranya adalah:

1. *Divide and Conquer*

Teknik ini digunakan untuk memecahkan masalah dalam skala yang besar kemudian membagi-baginya menjadi masalah yang lebih sederhana. Dari masalah yang sederhana ini diselesaikan dan hasilnya digabungkan dengan solusi dari bagian yang lain sehingga didapatkan solusi dari permasalahan utama.

2. *Decrease and Conquer*

Berbeda dengan divide and conquer, teknik ini tidak membagi masalah utama menjadi beberapa bagian, tapi mengurangi ukuran dari permasalahan utama sehingga menjadi lebih sederhana.

3. *Transform and Conquer*

Teknik ini merubah penyajian masalah ke dalam bentuk lain agar lebih mudah diselesaikan. Sebagai contoh : Menyelesaikan sistem persamaan linear dengan matriks.

Untuk melatih kemampuan pemecahan masalah dengan algoritma, coba kerjakan LKS berikut secara berkelompok.

## LEMBAR KERJA SISWA

Perhatikan kembali tabel TDL di atas.

| TARIF TENAGA LISTRIK UNTUK KEPERLUAN RUMAH TANGGA<br>BERLAKU MULAI 1 NOVEMBER 2014                      |            |                     |                               |                                                                                                               |                       |
|---------------------------------------------------------------------------------------------------------|------------|---------------------|-------------------------------|---------------------------------------------------------------------------------------------------------------|-----------------------|
| NO.                                                                                                     | GOL. TARIF | BATAS DAYA          | REGULER                       |                                                                                                               | PRA BAYAR<br>(Rp/kWh) |
|                                                                                                         |            |                     | BIAYA BEBAN<br>(Rp/kVA/bulan) | BIAYA PEMAKAIAN (Rp/kWh)                                                                                      |                       |
| 1.                                                                                                      | R-1/TR     | s.d.450 VA          | 11.000                        | Blok I : 0 s.d. 30 kWh : 169<br>Blok II : di atas 30 kWh s.d. 60 kWh : 360<br>Blok III : di atas 60 kWh : 495 | 415                   |
| 2.                                                                                                      | R-1/TR     | 900 VA              | 20.000                        | Blok I : 0 s.d. 20 kWh : 275<br>Blok II : di atas 20 kWh s.d. 60 kWh : 445<br>Blok III : di atas 60 kWh : 495 | 605                   |
| 3.                                                                                                      | R-1/TR     | 1.300 VA            | *)                            | 1.352                                                                                                         | 1.352                 |
| 4.                                                                                                      | R-1/TR     | 2.200 VA            | *)                            | 1.352                                                                                                         | 1.352                 |
| 5.                                                                                                      | R-2/TR     | 3.500 s.d. 5.500 VA | *)                            | 1.352                                                                                                         | 1.352                 |
| 6.                                                                                                      | R-3/TR     | 6.600 VA keatas     | *)                            | 1.352                                                                                                         | 1.352                 |
| Catatan :                                                                                               |            |                     |                               |                                                                                                               |                       |
| *) Diterapkan Rekening Minimum (RM):<br>RM1 = 40 (Jam Nyala) x Daya Tersambung (kVA) x Biaya Pemakaian. |            |                     |                               |                                                                                                               |                       |

Aturan perhitungan:

. Ada dua macam pemakaian yaitu REGULER dan PRA BAYAR. Kita hanya fokus pada pemakaian REGULER. Ada dua macam biaya yang harus ditanggung, yaitu Biaya Beban dan

biaya Pemakaian. Berikut ini beberapa contoh cara penghitungan biaya tagihan listrik tiap bulannya.

### **Batas daya 450 dan 900 VA**

Biaya Beban= Rp 11.000

Biaya Pemakaian dihitung bertahap (per blok)

- Blok 1: 0-30 kWh dengan biaya Rp 169,-/kWh
- Blok 2 : di atas 30 sampai 60 kWh dengan biaya Rp. 360,-/kWh
- Blok 3: di atas 60 kWh dengan biaya Rp 495,-/kWh.

Berikut ini contoh perhitungannya.

Pemakaian kurang dari 30 kWh

Misalkan pemakaian daya listrik adalah 25 kWh

Biaya Beban=Rp. 11.000,-

Biaya pemakaian hanya dihitung  $25 \times \text{Rp } 169,- = \text{Rp } 4.425,-$ .

Total Biaya Listrik Rp 11.000,-+Rp. 4.425,-=Rp. 15.425,-

Pemakaian di atas 30 kWh hingga 60 kWh

Misalkan pemakain daya listrik adalah 55 kWh.

Biaya Beban=Rp.11.000,-

Biaya Pemakaian:

Blok 1: 30 kWh pertama, biayanya =  $30 \times \text{Rp } 169,- = \text{Rp } 5.070,-$

Blok 2: 25 kWh berikutnya, biayanya =  $25 \times \text{Rp. } 360,- = \text{Rp } 9.000,-$

Total Biaya Pemakaian =  $\text{Rp } 5.070,- + \text{Rp. } 9.000,- = \text{Rp } 14.070,-$ .

Total Biaya Listrik=  $\text{Rp. } 11.000,- + \text{Rp. } 14.070,- = \text{Rp}15.070,-$

Pemakain di atas 60 kWh

Misalkan pemakain daya listrik adalah 100 kWh.

Biaya Beban=Rp. 11.000,-

Biaya Pemakaian:

Blok 1: 30 kWh pertama, biayanya =  $30 \times \text{Rp } 169,- = \text{Rp } 5.070,-$

Blok 2: 30 kWh berikutnya, biayanya =  $30 \times \text{Rp. } 360,- = \text{Rp } 10.800,-$

Blok 3: 40 kWh berikutnya, baiaynya =  $40 \times \text{Rp. } 495,- = \text{Rp. } 19.800,-$

Total biaya pemakaian =  $\text{Rp. } 35.670,-$

Total Biaya Listrik =  $\text{Rp. } 11.000 + \text{Rp. } 35.670,- = \text{Rp. } 46.670,-$

Perhitungan yang sama dilakukan untuk daya pemakai 900 VA.

### **Batas Daya 1300 VA ke atas**

Untuk kategori ini Biaya Beban dihitung dengan Tarif Minimum dengan rumus

Tarif Minimum =  $40 \times \text{Daya tersambung (kVA)} \times \text{Biaya Pemakaian}$

Biaya Pemakaian = Rp. 1.352,-

Tarif minimum berarti jika pemakaian kurang dari atau sama dengan  $40 \times \text{Daya Tersambung (kVA)}$  maka pelanggan tetap harus membayar sebesar tarif minimum. Jika lebih dari itu, dapat langsung dihitung besarnya biaya pemakaian saja (biaya beban sudah otomatis masuk pada perhitungan).

Contoh :

Daya Tersambung 2.200 VA = 2,2 kV

Batas pemakaian untuk tarif minimum =  $40 \text{ jam} \times 2,2 \text{ kV} = 88 \text{ kWh}$ .

Jika pemakaian daya adalah 50 kWh, karena di bawah Batas pemakaian tarif minimum maka biaya yang dibayarkan sebesar  $88 \times \text{Rp } 1.352,- = \text{Rp. } 118.976,-$ .

Jika pemakaian daya adalah 100 kWh, karena di atas batas pemakaian tarif minimum maka, biayanya tinggal dikalikan dengan biaya pemakaian, yaitu  $100 \times \text{Rp } 1.352,- = \text{Rp } 135.200,-$

Perhitungan yang sama diterapkan untuk daya tersambung 1.300 VA, 3500-5.500 VA dan 6.500 VA ke atas.

Berdasarkan tabel di atas, dengan anggota kelompok yang lain, buatlah rancangan algoritma untuk menentukan besaran tarif listrik pelanggan. Bagilah tugas pembuatan algoritma misalkan berdasarkan kategori/kelas tertentu. Ini sekaligus menerapkan salah satu teknik yaitu *Divide and Conquer*. Gabungkan hasil dari pekerjaan menjadi satu kesatuan algoritma. Sajikan algoritma kalian dalam bentuk flowchart.

#### 5.1.2.4. Mengasosiasikan

Dari hasil kegiatan belajar kali ini, tuliskan kesimpulan kalian mengenai beberapa hal berikut.

1. Langkah-langkah pembuatan program
2. Arti algoritma yang efektif dan efisien
3. Bagaimana strategi untuk menyelesaikan masalah yang rumit dengan algoritma

#### 5.1.2.5. Mengkomunikasikan

Presentasikan pekerjaan kalian di depan kelas, dan diskusikan dengan teman-teman yang lain.

#### 5.1.3. Rangkuman

1. Langkah-langkah pembuatan program meliputi : definisi masalah, analisis kebutuhan, perancangan algoritma (desain), pembuatan kode atau programming, uji coba dan debugging dan pemeliharaan
2. Analisis dan desain algoritma diperlukan untuk mendapatkan algoritma yang efektif dan efisien

#### 5.1.4. Tugas

Carilah struk pembayaran rekening air (PDAM), susun algoritma dalam bentuk flowchart.

### 5.1.5. Uji Kompetensi

1. Berikut ini adalah yang termasuk dalam proses pengemabangan aplikasi, kecuali ....
  - a. analisis kebutuhan
  - b. membuat desain/algoritma
  - c. membuat kode program
  - d. memasarkan program
  - e. pemeliharaan program.
2. Mengetahui kegunan dan tujuan pembuatan program termasuk dari bagian proses.....
  - a. definisi masalah
  - b. membuat desain/algoritma
  - c. membuat kode program
  - d. memasarkan program
  - e. pemeliharaan program.
3. Proses di mana program senantiasa dievaluasi ketika sudah digunakan oleh pelanggan dalam rangka meningkatkan kinerja program tersebut termasuk dalam proses .....
  - a. definisi masalah
  - b. membuat desain/algoritma
  - c. membuat kode program
  - d. memasarkan program
  - e. pemeliharaan program.
4. Efisiensi dari sebuah algoritma dapat dilihat berdasarkan .....
  - a. kebenaran output
  - b. optimlnya penyelesaian
  - c. running time
  - d. harga
  - e. keterbukaan untuk dapat digunakan atau diubah pengguna
5. Seorang programmer membduat program mencari rute terpendek di antara dua buah kota, dengan cara memodelkan ke dalam bentuk data yang dapat diolah ke komputer. Permasalahan dimodelkan dalam bentuk titik-titik sebagai kota dan sisi sebagai jalan. Dari hasil model ini di bawa ke bentuk matriks angka. Teknik yang programmer tersebut termasuk kategori .....
  - a. divide and conquer
  - b. brute force
  - c. dynamic
  - d. decrease and conquer
  - e. Transform and Conquer

## 5.2. KEGIATAN BELAJAR 2 *ERROR HANDLING*

Pernah kalian menggunakan program dan di tengah kalian asik bekerja tiba-tiba hang, atau menutup sendiri? Ini bisa disebabkan adanya error yang muncul tapi belum diantisipasi oleh program tersebut.

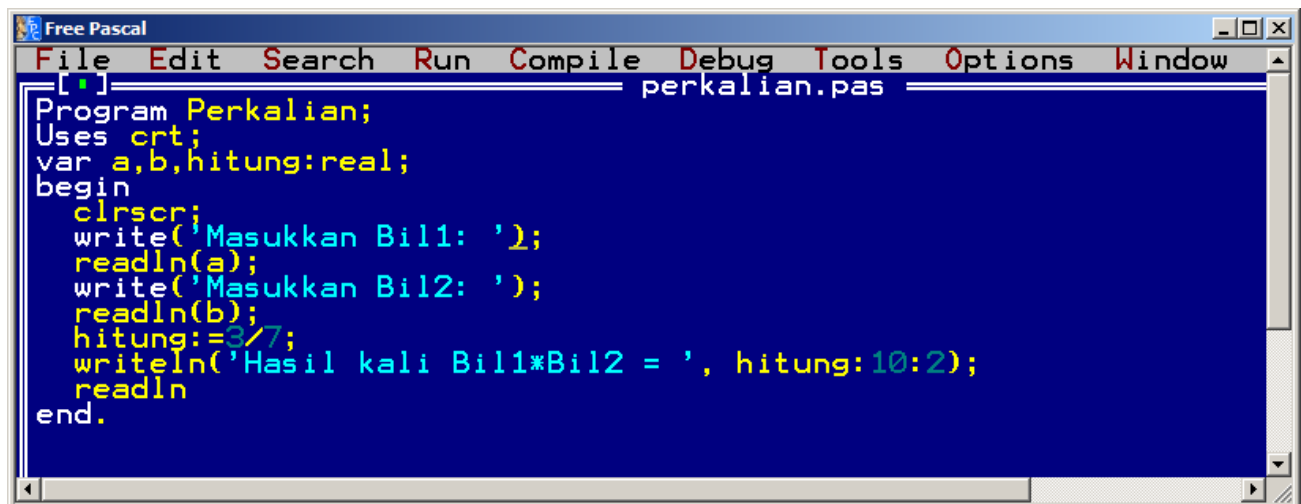
### 5.1.1. Tujuan Pembelajaran

Siswa memahami error handling dalam pascal dan menerapkannya pada masalah pemrograman.

### 5.1.2. Aktivitas Kegiatan belajar

#### 5.1.2.1. Mengamati

Perhatikan kode program berikut.

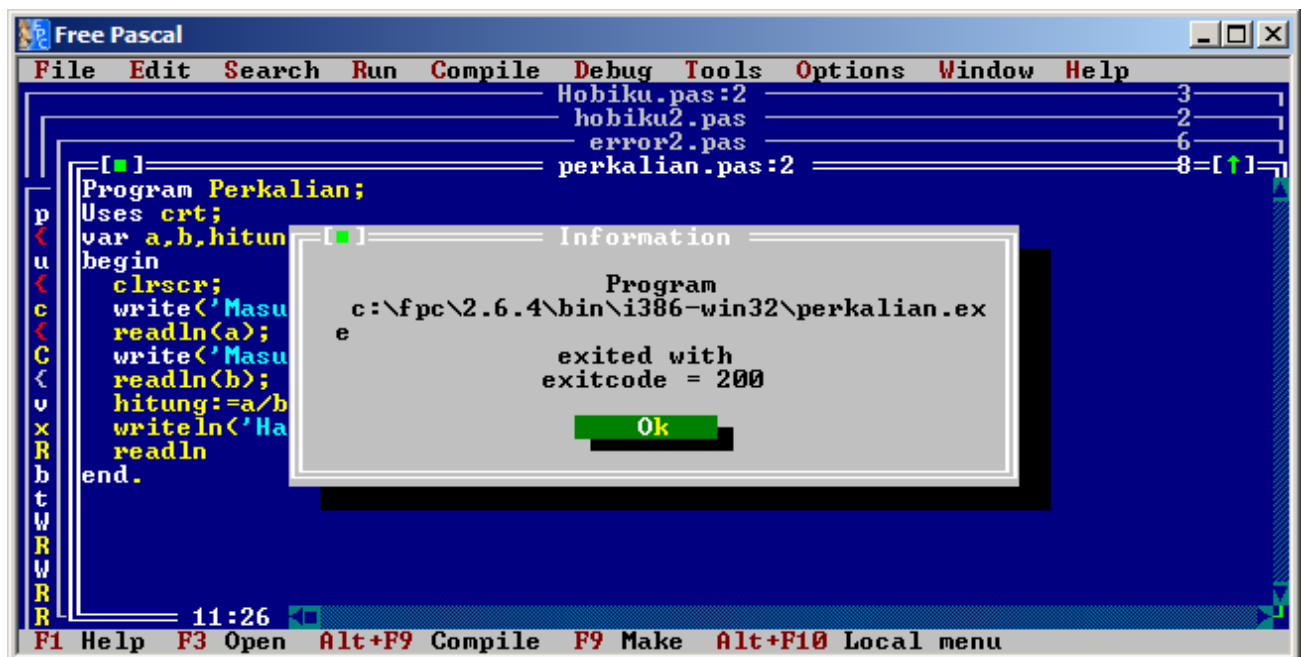


```

[.]
Program Perkalian;
Uses crt;
var a,b,hitung:real;
begin
 clrscr;
 write('Masukkan Bil1: ');
 readln(a);
 write('Masukkan Bil2: ');
 readln(b);
 hitung:=a/b;
 writeln('Hasil kali Bil1*Bil2 = ', hitung:10:2);
 readln
end.

```

Program di atas berhasil dikompilasi dengan baik tapi apa jadinya jika dimasukkan nilai a adalah 4 dan b adalah nol.



The screenshot shows the Free Pascal IDE with the same source code as before. An 'Information' dialog box is displayed in the foreground, indicating the program's execution status:

```

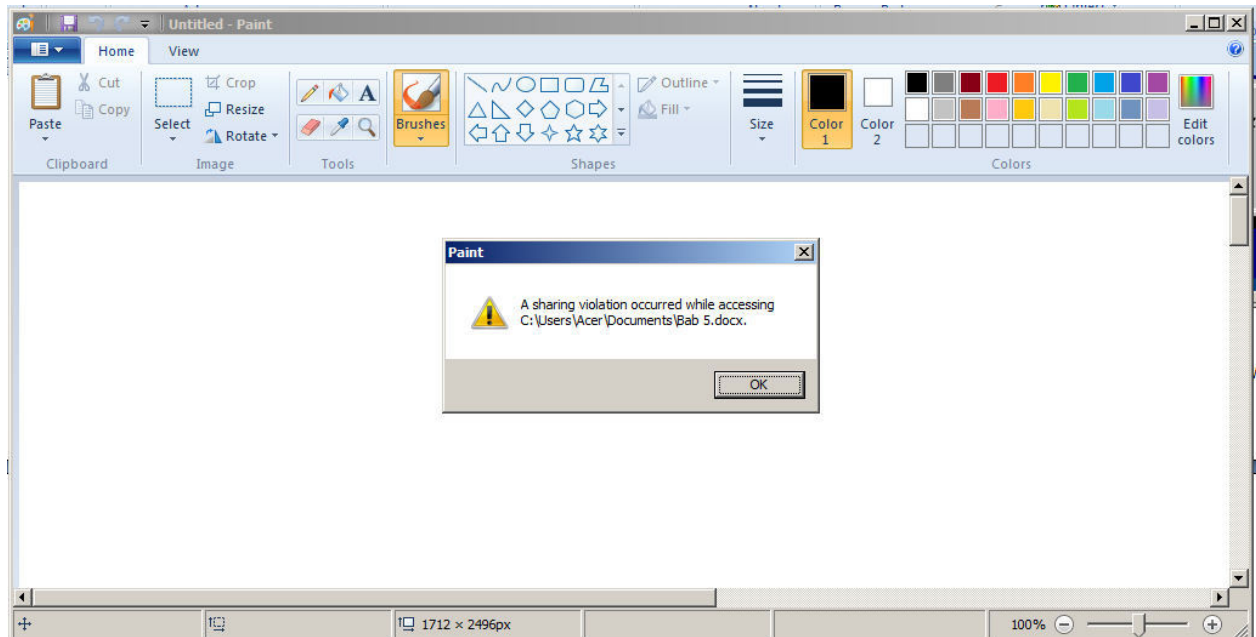
Information
Program
c:\fpc\2.6.4\bin\i386-win32\perkalian.exe
exited with
exitcode = 200
Ok

```

The background code is partially visible, showing the same Pascal program. The status bar at the bottom indicates the time as 11:26 and provides keyboard shortcuts for various actions.

Sebagai perbandingan:

Dengan menggunakan software Micosoft paint, akan dicoba membuka file WORD, apa yang terjadi?



#### 5.1.2.2. Menanya

1. Mengapa program pascal berhenti/keluar setelah kita inputkan karakter
2. Pesan apa yang muncul?
3. Mengapa program Microsoft Paint tidak berhenti keluar ketika salah memasukkan input?
4. Pesan apa yang muncul di Microsoft paint
4. Strategi apa agar program di pascal tidak berhenti ketika menjumpai kesalahan?

#### 5.1.2.3. Mengumpulkan Informasi/Mencoba

Dalam pembuatan program, kesalahan tidak hanya terjadi pada kode program (*syntax error*), kesalahan mungkin juga terjadi di pihak pengguna seperti kesalahan dalam menginputkan nilai ke dalam program. Program yang baik adalah yang bisa mengantisipasi kesalahan seperti itu sehingga tidak berhenti di tengah jalan. Jika program berhenti di tengah jalan, maka pengguna harus memulai dari awal, dan tentu ini menyebabkan banyaknya waktu yang hilang. Belum lagi jika user melakukan kesalahan lagi.

Selain kesalahan penulisan (*syntax error*) dikenal juga *run-time error*, yaitu kesalahan yang muncul saat program dijalankan. Sebagai contoh pembagian dengan nol.

Hasil:=a/b;

Agar program tidak langsung berhenti, maka perlu kesalahan perlu diantisipasi. Penanganan kesalahan ini disebut *error handling* atau *exception*. Kemampuan menangani error mutlak harus dikuasai oleh seorang programmer.

Di dalam Free Pascal disediakan beberapa metode penanganan error, di antaranya adalah menggunakan pernyataan Try ...Except, Try...Finally, dan Raise Exception.

### 1. Pernyataan Try ....Except.

Pernyataan ini menjalankan serangkaian proses dengan perintah **Try** di depan, kemudian setelah itu diikuti *Except*, untuk menandakan kejadian khusus yang dideteksi sebagai error (kesalahan). Sintaks dari pernyataan Try Except adalah sebagai berikut.

```
Try
 Pernyataan1;
 Pernyataan2;
except
on e: exception do // Exception handling
 begin
 Writeln('Error: ' + e.message);
 end;
end;
```

Untuk dapat menggunakan pernyataan Try Except di bagian awal program harus dipanggil dulu unit-unit yang terkait dengan penggunaan pernyataan ini.

#### Contoh 1.

```
program ExceptionHandling;
{$mode objfpc}{$H+}
uses
 {$IFDEF UNIX}{$IFDEF UseCThreads}
 cthreads,
 {$ENDIF}{$ENDIF}
 Classes, sysutils;
var
 a, b: Integer;
 Hasil: Real;
begin
 try
 Write('masukkan nilai a: ');
 Readln(a);
 Write('Masukkan nilai b: ');
 Readln(b);
 Hasil:= a/ b;
 Writeln('a/ b = ', Hasil);
 except
 on e: exception do
 begin
 Writeln('Telah terjadi error: ', e.message);
 end;
 end;
 Readln;
end.
```



Coba kita jalankan dengan dengan masukan =4 dan b=0

```
Running "c:\fpc\2.6.4\bin\i386-win32\errorhand.exe "
masukkan nilai a: 4
Masukkan nilai b: 0
Telah terjadi error: Division by zero
-
```

Ketika error terjadi, yaitu adanya pembagian dengan nol maka akan muncul pesan “Telah terjadi error” diikuti dengan tipe kesalahan yang terjadi. Kejadian exception (pengecualian) dideteksi pada pernyataan **Hasil:=a/b**. Jika nilai b = 0 maka perintah di bawah pengecualian akan dilakukan jika tidak maka langkah-langkah dalam pengecualian tidak dilakukan.

## 2. Pernyataan Try... Finally

Pernyataan Try .. Finally ini menjalankan erangkain proses setelah perintah Try. Jika ada error ma pada bagian error itu akan dilewati. Kemudian perintah-perintah setelah pernyataan Finally akan dieksekusi apapun yang terjadi baik ada kesalahan atau tidak.

Sintak pernyataan Try ...Finally

```
try
 Pernyataan1;
 Pernyataan2;
Finally
 Pernyataan3;
end;
```

Contoh 2.

```
program ExceptionHandling;
{$mode objfpc}{$H+}
uses
 {$IFDEF UNIX}{$IFDEF UseCThreads}
 cthreads,
 {$ENDIF}{$ENDIF}
 Classes, sysutils;
var
 a, b: Integer;
 Hasil: Real;
begin
 try
 Write('masukkan nilai a: ');
 Readln(a);
 Write('Masukkan nilai b: ');
 Readln(b);
```

```

 Hasil:= a/ b;
 Writeln('a/ b = ', Hasil);
 Finally
 Begin
 Writeln('Tekan Enter untuk menyelesaikan');
 Readln;
 End;
 end;
end.

```

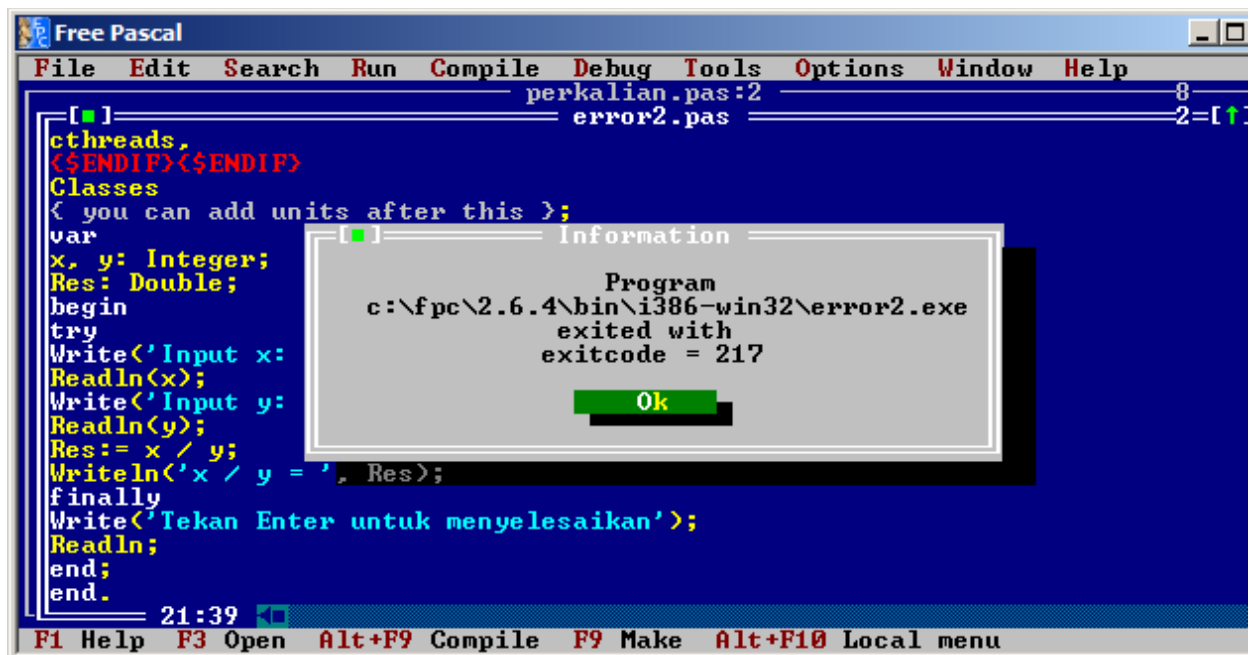
Dengan input a=4 dan b=0 maka keluarannya adalah:

```

Running "c:\fpc\2.6.4\bin\i386-win32\error2.exe "
Input x: 4
Input y: 0
Tekan Enter untuk menyelesaikan

```

Error sebenarnya terdeteksi tapi tidak ada tindakan khusus. Hal ini terlihat ketika keluar dari program, akan muncul pesan berikut.



### 3. Pernyataan Raise Exception

Dalam membuat program terkadang dibutuhkan suatu nilai dengan rentang tertentu, Misalkan variabel bulan, bernilai dari 1 sampai 12, tidak ada nilai 0 ataupun 13. Untuk menghindari kesalahan seperti ini digunakan pernyataan raise exception.

Berikut ini contoh penggunaannya:

#### Contoh 3.

```

program ExceptionHandling;
{$mode objfpc}{$H+}
uses
 {$IFDEF UNIX}{$IFDEF UseCThreads}

```



.....

.....

.....

2. Buatlah sebuah program untuk menentukan kecepatan rata-rata dari sebuah mobil dalam suatu perjalanan dengan masukan berupa jarak tempuh (dalam km) dan waktu perjalanan (dalam jam). Gunakan pernyataan Try ...Finally untuk penanganan kesalahan.

Jawab:

[illegible]

3. Buatlah sebuah proram untuk menentukan kelulusan berdasarkan rata-rata nilai UAN yang terdiri dari matematika, bahasa indonesia, dan bahasa inggris. Setiap nilai mempunyai range 0-100. Gunakan pernyataan **raise exception** untuk mengantisipasi kesalahan inputan.

Jawab:

This image shows a single sheet of white paper with horizontal blue ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins or other markings on the paper.

.....

.....

.....

#### 5.2.2.4. Mengasosiasikan

Dari hasil kegiatan belajar ini, apa kesimpulan kalian mengenai error handling, mengapa diperlukan dan bagaimana cara melakukannya dalam bahasa Pascal.

#### 5.2.2.5. Mengkomunikasikan

Presentasikan hasil pekerjaan LKS kalian dan kesimpulan kalian di depan kelas.

#### 5.2.3. Rangkuman

1. Ada 3 macam kesalahan dalam penyusunan program: syntax error, run-time error dan logical error
2. Run-time error harus diantisipasi agar program tidak berhenti di tengah jalan, cara mengantisipasi disebut dengan error handling
3. Penanganan error dalam Pascal menggunakan pernyataan *Try Except*, *Try Finally* dan *Raise Exception*

#### 5.2.4. Tugas

Buatlah sebuah program, masing-masing satu yang melibatkan penanganan error menggunakan pernyataan Try Except, Try Finally dan Raise Exception

#### 5.2.5. Uji Kompetensi

**Pilihlah jawaban yang benar dengan memberikan tanda silang pada huruf a, b, c, d, atau e.**

1. Berikut ini pernyataan yang benar mengenai kesalahan dalam pembuatan program, kecuali ....
  - a. kesalahan dapat terjadi pada penulisan kode
  - b. kesalahan algoritma menyebabkan hasil yang keliru
  - c. kesalahan inputan dapat menyebabkan program berhenti di tengah jalan
  - d. Jika langkah-langkah algoritmanya sudah benar dan dituliskan sesuai sintaks, maka program tidak akan berhenti di tengah jalan
  - e. kesalahan penulisan kode paling mudah dideteksi
2. Misalkan dalam sebuah program menghitung konversi suhu dari celcius ke satuan lain. Programmer ingin suhu dalam celcius mempunyai rentang dari -50<sup>0</sup>C hingga 100 <sup>0</sup>C. Untuk menjamin bahwa inputan sesuai dengan rentang itu digunakan pernyataan....
  - a. Try Except
  - b. Try Until
  - c. Try Finally

- d. Raise Exception
  - e. Raise Finally
3. Agar program memberikan pengecualian ketika menemukan kesalahan perhitungan, dengan tidak mengeksekusinya amun menampilkan pesan kesalahan digunakan pernyataan....
- a. Try Except
  - b. Try Until
  - c. Try Finally
  - d. Raise Until
  - e. Raise Finally
4. Agar program tidak mempedulikan kesalahan yang terjadi namun tetap bisa melanjutkan pada proses selanjutnya, digunakan pernyataan.....
- a. Try Except
  - b. Try Until
  - c. Try Finally
  - d. Raise Until
  - e. Raise Finally
5. Maksud dari **e.message** pada pernyataan Try Except adalah ....
- a. pesan yang diambil dari email
  - b. pesan dari pengguna
  - c. pesan yang didefinisikan di awal program
  - d. pesan mengenai kesalahan yang terjadi
  - e. pesan setelah program berhenti

### 5.3. DEBUGGING

Selain syntax error dan run-time error, ada satu macam kesalahan lagi yang paling sulit untuk dideteksi, yaitu kesalahan logika atau kesalahan algoritma.

#### 5.3.1. Tujuan Pembelajaran

Siswa memahami apa itu debugging dan bagaimana melakukan nya dalam pemrograman Pascal

#### 5.3.2. Aktivitas Kegiatan Belajar

Sebuah penilaian prestasi belajar diwakili dengan huruf. Huruf ini diperoleh berdasarkan nilai ujian skala 0-100 dengan kategori sebagai berikut

Nilai 80-100: A

70-79 : B

60-69 : C

50-59 : D

0-49: E

Dari Perhatikan Kode Program berikut, dan hasilnya.

```
Program KonversiNilai;
Uses c
Var nilai:integer;
 Huruf:char;
Begin
 Write('Masukkan nilai : ');
 Readln(nilai);
 If nilai>=80 then huruf:='A'
 Else if nilai>=70 then huruf :='B'
 Else huruf:='C';
 If nilai>=50 then huruf:='D'
 Else huruf:='E';
 Writeln('Nilai Anda dalam huruf adalah : ',huruf);
 Readln;
End.
```

Beberapa hasil output sebagai berikut:

Masukkan nilai : 55

Nilai Anda dalam huruf adalah : D

Masukkan nilai : 30

Nilai Anda dalam huruf adalah : E

Masukkan nilai: 100

**Nilai dalam huruf adalah : D**

#### **5.3.2.2. Menanya**

Dari hasil pengamatan kalian, jawablah pertanyaan berikut ini.

1. Mengapa untuk nilai =100 hurufnya adalah D?
2. Mengapa terjadi kesalahan seperti itu?
3. Bagaimana cara/teknik mencari kesalahan algortimik dalam kode program?

#### **5.3.2.3. Mengumpulkan Informasi/Mencoba**

Mencari kesalahan logika dalam alur algoritma pemrograman merupakan sesuatu yang cukup sulit. Oleh karena itu digunakan istilah *debugging* (mencari kutu). Kesalahan sekecil apapun jika itu merupakan kesalahan algortimik akan mempengaruhi hasil dari program tersebut. Semakin kecil kesalahan itu maka akan sulit mencarinya, sehingga diibaratkan sesulit mencari kutu.

Secara garis besar melakukan *debugging* program dapat dilakukan dengan beberapa cara, yaitu:

1. menjalankan kode program baris demi baris (pernyataan demi pernyataan)
2. menjalankan program hingga baris tertentu, dengan menggunakan breakpoint
3. Melihat isi dari variabel memori ketika program dijalankan.

#### **Menjalankan Kode Baris demi Baris**

Salah satu strategi mencari kesalahan adalah meneliti baris per baris pernyataan dan mencoba, apakah pada baris tersebut sudah benar atau belum. Ini bisa dilakukan secara manual atau menggunakan perintah yang ada pada tool pembat aplikasi, dalam hal ini Free Pascal. Mengecek baris per baris pernyataan apakah sudah benar dengan mencobakan beberapa nilai, tentunya yang menghasilkan keluaran salah.

Misalkan dalam kode program di atas, kita akan lihat satu persatu pernyataan dengan nilai input nilai=100. Berikut ini potongan bagian pernyataan. Akhir pernyataan biasanya diakhiri dengan titik koma. Selanjutnya adalah kita melakukan pemeriksaan apakah pernyataan tersebut sudah benar sesuai dengan harapan. Untuk itu kita dapat melakukan secara bertahap, caranya mulai dari atas, tentukan pernyataan manakah yang akan diperiksa, kemudian beri tanda komentar untuk semua pernyataan di bawahnya. Contoh sebagai berikut. Perintah `writeln` tidak dimasukkan dalam komentar karena dari sinilah kita mengetahui outputnya sudah sesuai atau belum.



```

Program KonversiNilai;
Uses crt;
Var nilai:integer;
 Huruf:char;
Begin
 Write('Masukkan nilai : ');
 Readln(nilai);
 <If nilai>=80 then huruf:='A'
 Else if nilai>=70 then huruf :='B'
 Else huruf:='C';
 If nilai>=50 then huruf:='D'
 Else huruf:='E';
 Writeln('Nilai Anda dalam huruf adalah : ',huruf);
 Readln;
End.

```

Kemudian jalankan, apakah hasilnya sudah sesuai? Tentu untuk kasus ini sudah, karena hanya meminta inputan nilai. Selanjutnya diambil pernyataan selanjutnya, keluarkan dari komentar.

```

Program KonversiNilai;
Uses crt;
Var nilai:integer;
 Huruf:char;
Begin
 Write('Masukkan nilai : ');
 Readln(nilai);
 If nilai>=80 then huruf:='A'
 Else if nilai>=70 then huruf :='B'
 Else huruf:='C';
 < If nilai>=50 then huruf:='D'
 Else huruf:='E';
 Writeln('Nilai Anda dalam huruf adalah : ',huruf);
 Readln;
End.

```

Jalankan dan beri nilai variabel nilai=100. Ternyata hasilnya sudah benar.

```

Running "c:\fpc\2.6.4\bin\i386-win32\logika.exe "
Masukkan nilai : 100
Nilai Anda dalam huruf adalah : A

```

Langkah berikutnya keluarkan pernyataan tersisa dari komentar.

```

Running "c:\fpc\2.6.4\bin\i386-win32\logika.exe "
Masukkan nilai : 100
Nilai Anda dalam huruf adalah : D

```

Hal ini menandakan bahwa kesalahan terletak pada pernyataan

```

If nilai>=50 then huruf:='D'
 Else huruf:='E';

```

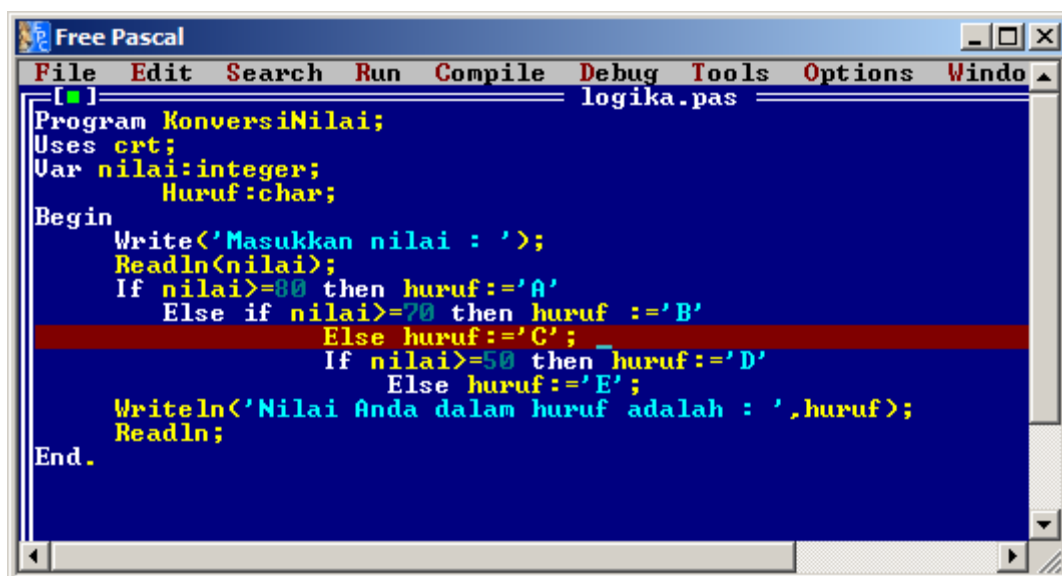
Setelah mengetahui letak kesalahannya, langkah selanjutnya adalah memperbaiki.

Kita lihat bahwa struktur percabangan ini ternyata terpisah dari percabangan sebelumnya. Pada struktur percabangan sebelumnya program sudah memberikan huruf = 'A' untuk nilai =100. Karena ada struktur percabangan baru maka nilai diperiksa kembali sehingga karena nilai  $\geq 50$  hurufnya diganti menjadi 'D'. Algoritma yang benar harusnya:

```
If nilai ≥ 80 then huruf:='A'
 Else if nilai ≥ 70 then huruf:='B'
 Else if nilai ≥ 60 then huruf:='C'
 Else If nilai ≥ 50 then huruf:='D'
 Else huruf:='E';
```

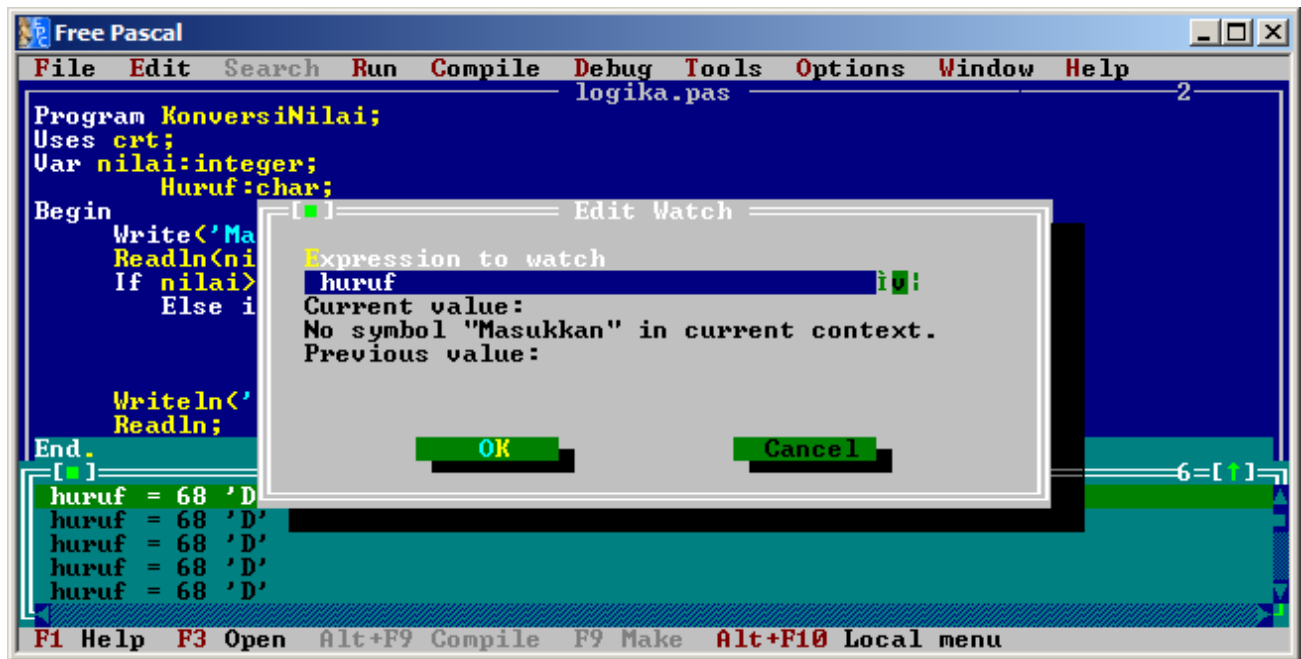
### Menggunakan *BreakPoint* dan *Watch*

Jika kode program cukup banyak, maka pengecekan baris demi baris memerlukan waktu yang lama. Hal ini bisa diatasi dengan menentukan breakpoint. Caranya letakkan kursor di mana program akan berhenti kemudian pilih menu Debug, pilih Breakpoint.

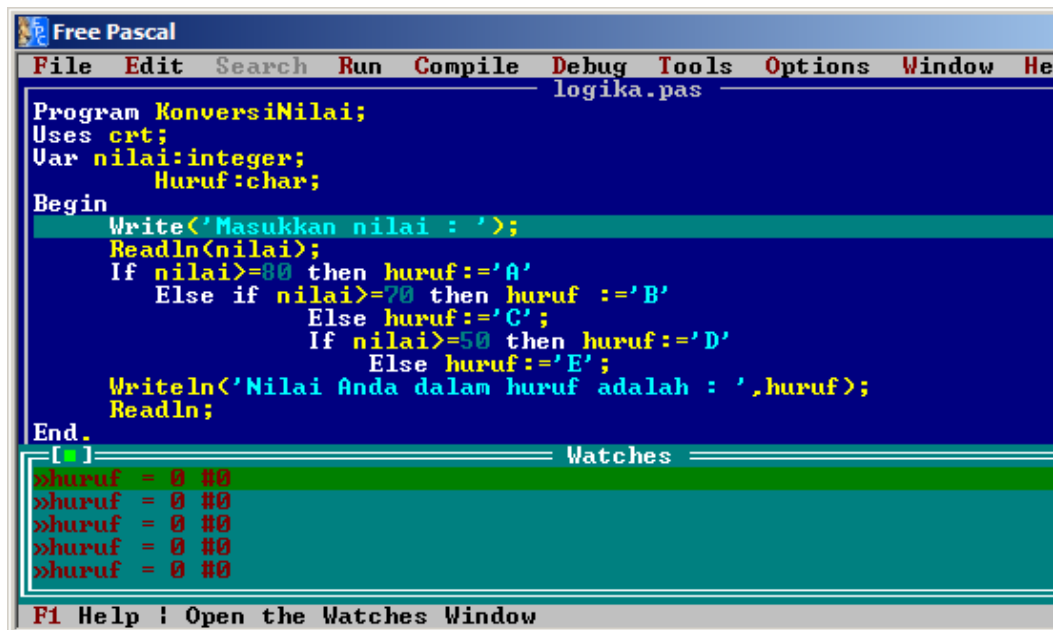


Breakpoint akan mengakibatkan program berhenti pada baris di mana breakpoint diset/ditentukan. Pada saat itu, proses dikembalikan pada IDE dan dimungkinkan untuk dijalankan kembali.

Fasilitas Watch adalah fasilitas untuk melihat isi variabel memori pada saat program dihentikan (oleh breakpoint). *Watch* ditampilkan di window baru dengan memilih menu *Debug*, pilih *Add Watch* atau dengan menekan *Ctrl+F7*. Isikan dengan nama variabel yang akan dilihat isinya.



Kemudian klik menu Debug, dan pilih Watch sehingga muncul jendela baru di bagian bawah. Untuk perubahan isi dari variabel huruf, tekan F7 secara berturutan. Cursor pada kode program berubah menunjukkan sejauh mana eksekusi dilakukan, dan berpengaruh pada nilai huruf atau tidak?



```

Free Pascal
File Edit Search Run Compile Debug Tools Options Window Help
logika.pas
Program KonversiNilai;
Uses crt;
Var nilai:integer;
 Huruf:char;
Begin
 Write('Masukkan nilai : ');
 Readln(nilai);
 If nilai>=80 then huruf:='A'
 Else if nilai>=70 then huruf :='B'
 Else huruf:='C';
 If nilai>=50 then huruf:='D'
 Else huruf:='E';
 Writeln('Nilai Anda dalam huruf adalah : ',huruf);
 Readln;
End.
Watches
huruf = 65 'A'
>huruf = 65 'A'
>huruf = 65 'A'
>huruf = 65 'A'
>huruf = 65 'A'
F1 Help ! Open the Watches Window

```

```

Free Pascal
File Edit Search Run Compile Debug Tools Options Window Help
logika.pas
Program KonversiNilai;
Uses crt;
Var nilai:integer;
 Huruf:char;
Begin
 Write('Masukkan nilai : ');
 Readln(nilai);
 If nilai>=80 then huruf:='A'
 Else if nilai>=70 then huruf :='B'
 Else huruf:='C';
 If nilai>=50 then huruf:='D'
 Else huruf:='E';
 Writeln('Nilai Anda dalam huruf adalah : ',huruf);
 Readln;
End.
Watches
huruf = 68 'D'
>huruf = 68 'D'
>huruf = 68 'D'
>huruf = 68 'D'
>huruf = 68 'D'
F1 Help ! Open the Watches Window

```

Perubahan nilai variabel huruf dapat dilihat langkah demi langkah sehingga kita bisa tahu kapan kesalahan itu terjadi.

## LEMBAR KERJA SISWA

Terapkan teknik debugging untuk menentukan kesalahan dari kode program berikut.

1. Program menentukan besar biaya sewa CD, di mana biayanya sama dengan jumlah CD dipinjam dikali biaya sewa per CD kemudian ditambahkan dengan denda yang besarnya 500 per CD per hari keterlambatan. Biaya

```

Program SewaCD;
Var jmlCD, HTerlambat, Sewa, Total, Denda:integer;
Begin

```

```

Write('Masukkan banyak CD :');readln(jmlCD);
Write ('Masukkan hari keterlambatan, isi 0 jika tidak :'); readln(Hterlambat);
BiayaSewa:=jmlCD*2000;
Denda:=jmlCD*500;
Total:=BiayaSewa*Denda;
Writeln('Total yang harus dibayar adalah : ', Total);
End.

```

Buatlah perhitungan manual sebagai perbandingan, langkah per langkah.

2. Program menentukan jumlahan deret bilangan 1+4+7+10+13+....yang kurang dari n

```

Program Deret;
Var suku, jum:integer;
Begin
 Readln(n);
 Suku:=1;
 i:=1;
 jum:=0
 While(suku<=n) do
 Begin
 Suku:=suku+3*i;
 Jum:=jum+suku;
 i:=i+1;
 End;
 Writeln(jum);
 Readln;
End.

```

- a. Apakah program di atas sudah menghasilkan output yang benar?
- b. Dapatkah kita terapkan teknik debugging untuk memperbaiki program di atas. Jika tidak jelaskan, jika bisa, lakukan.

#### 5.3.2.4. Mengasosiasikan

Dari hasil pekerjaan LKS, buat kesimpulan apa itu debugging, seberapa pentingnya proses debugging ini, dan bagaimana melakukannya?

#### 5.3.2.5. Mengkomunikasikan

Presentasikan pekerjaan LKS kalian di depan kelas, dan diskusikan dengan teman-teman yang lain.

#### 5.3.3. Rangkuman

1. Debugging adalah proses menemukan kesalahan, terutama kesalahan algoritmik dalam sebuah program
2. Debugging dapat dilakukan dengan menjalankan baris per baris perintah, atau sekumpulan perintah hingga titik tertentu (breakpoint) atau dengan melihat isi dari variabel dari langkah per langkah.

#### 5.3.4. Tugas

Buat sebuah program dalam permasalahan sehari-hari yang memerlukan beberapa rumus yang panjang, misalkan penggajian, persewaan mobil, dan sebagainya. Kemudian tuliskan strategi melacak kesalahan (debugging). Kumpulkan dalam bentuk laporan tertulis dalam Microsoft Word.

#### 5.3.5. Uji Kompetensi.

**Pilihlah jawaban yang benar dengan memberikan tanda silang pada huruf a, b, c, d, atau e.**

1. Yang termasuk kesalahan algoritmik adalah ....
  - a. menentukan operasi pada rumus
  - b. menuliskan format membaca inputan
  - c. memilih tipe data
  - d. pembagian dengan nol
  - e. memasukkan input di luar range
2. Salah dalam menentukan kondisi percabangan termasuk kesalahan ....
  - a. sintaks
  - b. struktur
  - c. run-time error
  - d. algoritmik
  - e. user
3. Untuk melihat isi variabel pada tiap langkah digunakan perintah ....
  - a. Breakpoint
  - b. Memory
  - c. Watch
  - d. Lookup
  - e. Step
4. Agar kita bisa memberhentikan program pada titik tertentu digunakan perintah ....
  - a. Breakpoint
  - b. Memory
  - c. Watch
  - d. Lookup
  - e. Step
5. Setelah tahap uji coba dan debugging, langkah pada pembuatan program adalah ....

- a. Pemeliharaan
- b. Pemrograman
- c. Desain
- d. Analisis kebutuhan
- e. Selesai

#### 5.4. KEGIATAN BELAJAR 4 STUDY KASUS PROYEK PENGEMBANGAN APLIKASI

Setelah hampir satu semester kalian belajar dasar-dasar pemrograman dengan Pascal, maka di akhir buku ini kalian akan menerapkan semuanya dalam satu tugas, yaitu proyek membuat program berskala besar dari permasalahan yang ada di kehidupan sehari-hari. Masing menyambung kegiatan sebelumnya, kita akan menggunakan permasalahan pada KB 1, pada Bab ini yaitu permasalahan pembuatan program penghitungan tagihan listrik.

##### 5.4.1. Tujuan Pembelajaran

Siswa mampu membuat aplikasi dalam kehidupan sehari-hari dengan Bahasa Pascal.

##### 5.4.2. Aktivitas kegiatan belajar Siswa

#### LEMBAR KERJA SISWA

Dengan kelompok kalian, buatlah program/aplikasi untuk menentukan besarnya tagihan listrik berdasarkan Tabel TDI dan keterangannya berikut untuk semua golongan dan batas daya. Tentukan input dan outputnya dan buatlah format nya semenarik mungkin. Hasil pekerjaan ini sekaligus sebagai evaluasi dari kegiatan belajar ini.

**TARIF TENAGA LISTRIK UNTUK KEPERLUAN RUMAH TANGGA  
BERLAKU MULAI 1 NOVEMBER 2014**

| NO.                                                                                                                                                    | GOL. TARIF | BATAS DAYA          | REGULER                    |                                                                                                               | PRA BAYAR (Rp/kWh) |
|--------------------------------------------------------------------------------------------------------------------------------------------------------|------------|---------------------|----------------------------|---------------------------------------------------------------------------------------------------------------|--------------------|
|                                                                                                                                                        |            |                     | BIAYA BEBAN (Rp/kVA/bulan) | BIAYA PEMAKAIAN (Rp/kWh)                                                                                      |                    |
| 1.                                                                                                                                                     | R-1/TR     | s.d.450 VA          | 11.000                     | Blok I : 0 s.d. 30 kWh : 169<br>Blok II : di atas 30 kWh s.d. 60 kWh : 360<br>Blok III : di atas 60 kWh : 495 | 415                |
| 2.                                                                                                                                                     | R-1/TR     | 900 VA              | 20.000                     | Blok I : 0 s.d. 20 kWh : 275<br>Blok II : di atas 20 kWh s.d. 60 kWh : 445<br>Blok III : di atas 60 kWh : 495 | 605                |
| 3.                                                                                                                                                     | R-1/TR     | 1.300 VA            | *)                         | 1.352                                                                                                         | 1.352              |
| 4.                                                                                                                                                     | R-1/TR     | 2.200 VA            | *)                         | 1.352                                                                                                         | 1.352              |
| 5.                                                                                                                                                     | R-2/TR     | 3.500 s.d. 5.500 VA | *)                         | 1.352                                                                                                         | 1.352              |
| 6.                                                                                                                                                     | R-3/TR     | 6.600 VA keatas     | *)                         | 1.352                                                                                                         | 1.352              |
| Catatan :<br>*) Diterapkan Rekening Minimum (RM):<br>$RM1 = 40 \text{ (Jam Nyala)} \times \text{Daya Tersambung (kVA)} \times \text{Biaya Pemakaian.}$ |            |                     |                            |                                                                                                               |                    |

Tabel di atas menggambarkan biaya yang harus dibayarkan oleh pelanggan PLN kategori Keperluan Rumah Tangga. Ada dua macam pemakaian yaitu REGULER dan PRA BAYAR. Kita hanya fokus pada pemakaian REGULER. Ada dua macam biaya yang harus ditanggung, yaitu Biaya Beban dan biaya Pemakaian. Berikut ini beberapa contoh cara penghitungan biaya tagihan listrik tiap bulannya.

##### Batas daya 450 dan 900 VA

Biaya Beban= Rp 11.000



Biaya Pemakaian dihitung bertahap (per blok)

- Blok 1: 0-30 kWh dengan biaya Rp 169,-/kWh
- Blok 2 : di atas 30 sampai 60 kWh dengan biaya Rp. 360,-/kWh
- Blok 3: di atas 60 kWh dengan biaya Rp 495,-/kWh.

Berikut ini contoh perhitungannya.

Pemakaian kurang dari 30 kWh

Misalkan pemakaian daya listrik adalah 25 kWh

Biaya Beban=Rp. 11.000,-

Biaya pemakaian hanya dihitung  $25 \times \text{Rp } 169,- = \text{Rp } 4.425,-$ .

Total Biaya Listrik Rp 11.000,-+Rp. 4.425,-=Rp. 15.425,-

Pemakaian di atas 30 kWh hingga 60 kWh

Misalkan pemakain daya listrik adalah 55 kWh.

Biaya Beban=Rp.11.000,-

Biaya Pemakaian:

Blok 1: 30 kWh pertama, biayanya =  $30 \times \text{Rp } 169,- = \text{Rp } 5.070,-$

Blok 2: 25 kWh berikutnya, biayanya =  $25 \times \text{Rp. } 360,- = \text{Rp } 9.000,-$

Total Biaya Pemakaian =  $\text{Rp } 5.070,- + \text{Rp. } 9.000,- = \text{Rp } 14.070,-$ .

Total Biaya Listrik=  $\text{Rp. } 11.000,- + \text{Rp. } 14.070,- = \text{Rp}15.070,-$

Pemakain di atas 60 kWh

Misalkan pemakain daya listrik adalah 100 kWh.

Biaya Beban=Rp. 11.000,-

Biaya Pemakaian:

Blok 1: 30 kWh pertama, biayanya =  $30 \times \text{Rp } 169,- = \text{Rp } 5.070,-$

Blok 2: 30 kWh berikutnya, biayanya =  $30 \times \text{Rp. } 360,- = \text{Rp } 10.800,-$

Blok 3: 40 kWh berikutnya, baiaynya =  $40 \times \text{Rp. } 495,- = \text{Rp. } 19.800,-$

Total biaya pemakaian =  $\text{Rp. } 35.670,-$

Total Biaya Listrik =  $\text{Rp. } 11.000 + \text{Rp. } 35.670,- = \text{Rp. } 46.670,-$

Perhitungan yang sama dilakukan untuk daya pemakai 900 VA.

### **Batas Daya 1300 VA ke atas**

Untuk kategori ini Biaya Beban dihitung dengan Tarif Minimum dengan rumus

Tarif Minimum =  $40 \times \text{Daya tersambung (kVA)} \times \text{Biaya Pemakaian}$

Biaya Pemakaian =  $\text{Rp. } 1.352,-$

Tarif minimum berarti jika pemakaian kurang dari atau sama dengan 40 x Daya Tersambung (kVA) maka pelanggan tetap harus membayar sebesar tarif minimum. Jika lebih dari itu, dapat langsung dihitung besarnya biaya pemakaian saja (biaya beban sudah otomatis masuk pada perhitungan).

Contoh :

Daya Tersambung 2.200 VA= 2,2 kV

Batas pemakaian untuk tarif minimum = 40 jam x 2,2 kV = 88 kWh.

Jika pemakaian daya adalah 50 kWh, karena di bawah Batas pemakaian tarif minimum maka biaya yang dibayarkan sebesar 88 x Rp 1.352,- = Rp. 118.976,-.

Jika pemakaian daya adalah 100 kWh, karena di atas batas pemakaian tarif minimum maka, biayanya tinggal dikalikan dengan biaya pemakaian, yaitu 100 x Rp 1.352,-= Rp 135.200,-

Perhitungan yang sama diterapkan untuk daya tersambung 1.300 VA, 3500-5.500 VA dan 6.500 VA ke atas.

Buatlah program/aplikasi perhitungan tagihan listrik di atas secara berkelompok.

Buatlah format input dan output semenarik mungkin.

## DAFTAR PUSTAKA

- Anany Levitin, 2002. Introduction to the Design & Analysis of Algorithms, Addison Wesley:United States
- Munir, Rinaldi, 2011. Algoritma dan pemrograman dalam bahasa Pascal dan C, Informatika Bandung
- Michaël Van Canneyt, 2010. Reference guide for Free Pascal, version 2.4.2. <http://web.fe.up.pt/~jmcruz/si/si.1011/doc/ref-fpc.pdf>. Diakses tanggal 25 Nopember 2014.
- Motaz Abdel Azeem,2012. Start programming using Object Pascal Free Pascal/Lazarus. <http://www.hnue.edu.vn/Portals/0/TeachingSubject/hoanglm/27736a95-2ee0-47c4-be13-d810aa36cd76ObjectPascalBasis.pdf>. Diakses tanggal 25 Nopember 2014.
- Pressman, RS, 2008. Software Engineering: A Practitioner's Approach. New York:McGraw-Hill.
- Tim Pembina TOKI, 2004. Seri Buku Persiapan Seleksi Tim Olimpiade Komputer Indonesia:Buku Untuk Siswa, Referensi Pemrograman Bahasa Pascal Menggunakan Free Pascal Versi 1.0.10. Departemen Pendidikan Nasional Indonesia. Jakarta
- <http://www.Freepascal.org>. Diakses tanggal 25 Nopember 2014.