

**LAPORAN HASIL PRAKTIKUM**

**ALGORITMA DAN PEMOGRAMAN I**



**Nama : Jordan Setiawan Nanyan**

**NIM : 213020503058**

**Kelas : D**

**Modul : RUNTUTAN (SEQUENTIAL)**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PALANGKA RAYA**

**2021**

## **BAB 1**

### **TUJUAN DAN LANDASAN TEORI**

#### **1.1. Tujuan :**

- 1.1.1. Dapat menggolongkan sebuah masalah ke dalam kategori runtunan (sequential).
- 1.1.2. Dapat memecahkan permasalahan yang sifatnya runtunan ke dalam sebuah program.

#### **1.2. Landasan Teori :**

Runtunan adalah struktur algoritma paling dasar yang berisi rangkaian instruksi yang diproses secara sekuensial, artinya dikerjakan satu per satu, mulai dari instruksi pertama sampai instruksi akhir.

Sebuah instruksi dieksekusi setelah instruksi sebelumnya selesai dieksekusi. Urutan instruksi menentukan keadaan hasil dari algoritma. Bila urutannya diubah, maka hasil akhirnya mungkin juga berubah. Instruksi algoritma pada struktur runtunan ini akan dikerjakan semua dan tidak ada yang diabaikan. Pada dasarnya, semua program dibuat berdasarkan struktur algoritma runtunan.

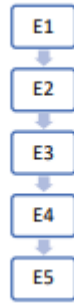
Menurut Goldshlager dan Lister (1998) struktur berurutan atau runtunan mengikuti ketentuan-ketentuan sebagai berikut.:

- a) Tiap instruksi dikerjakan satu persatu.
- b) Tiap instruksi dilaksanakan tepat sekali, tidak ada yang diulang.
- c) Urutan instruksi yang dilaksanakan pemroses sama dengan urutan aksi sebagaimana yang tertulis di dalam algoritmanya.
- d) Akhir dari instruksi terakhir merupakan akhir algoritma.

Setiap instruksi di dalam runtunan ditulis dalam satu baris, atau beberapa instruksi dalam baris yang sama tetapi antara setiap instruksi dipisahkan dengan tanda titik koma (;).

Contoh struktur runtunan adalah sebagai berikut.

Bila suatu instruksi runtunan dalam algoritma ditulis berturut-turut, kita lambangkan dengan variabel E1, E2, E3, E4, E5, maka urutan pelaksanaan instruksi tersebut adalah seperti gambar 1 dibawah ini:



Gambar 1 Runtunan yang Terdiri dari 5 Buah Instruksi

Keterangan :

Runtunan diatas, dimulai dari intruksi E1, kemudian E1 harus dikerjakan sampai selesai terlebih dahulu sebelum instruksi E2 dikerjakan, setelah instruksi E2 selesai dikerjakan juga, maka instruksi E3 dikerjakan lagi dan seterusnya sampai instruksi ke E5 setelah itu proses selesai dikerjakan dan baru berhenti dari proses pembacaan runtunan tersebut.

### 1.3. Tugas

1. Buatlah program untuk melakukan operasi aritmatika dimana bilangan pertama dan kedua merupakan inputan dari piranti masukan.
2. Buatlah program untuk mencari nilai x1 dan x2 dari persamaan linear  $ax^2 + bx + c = 0$  dengan rumus :

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Dimana nilai a, b dan c merupakan inputan dari piranti masukan.

3. Algoritma Hitung Buku

Deklarasi

Const Bsoni = 15

var

Badi, Banis, Bluki, Bnisa: integer

Deskripsi

```
Badi ← Bsoni-10  
Banis ← 2*(Bsoni+Badi)  
Bluki ← Bsoni+Badi+Banis-5  
Banis ← Bluki*4  
Write(Badi, Banis, Bluki, Banis)
```

Buatlah program pascal untuk menghitung buku Adi, Anis, Luki, dan Nisa berdasarkan algoritma diatas.

#### 4. Algoritma KonversiDetik

Deklarasi

var

Detik, KonvJam, KonvMnt, KonvDtk, Sisa : integer

Deskripsi

read(Detik)

KonvJam ← Detik div 3600

Sisa ← Detik mod 3600

KonvMnt ← Sisa div 60

KonvDtk ← Sisa mod 60

write(KonvJam, KonvMnt, KonvDtk)

Buatlah program pascal untuk mengkonversi waktu dari satuan detik ke satuan jam, menit, dan detik.

## BAB 2

### LANGKAH PENYELESAIAN DAN PEMBAHASAN TUGAS PRAKTIKUM

#### 2.1.Langkah Penyelesaian dan Penjelasan

##### 2.1.1. Langkah Penyelesaian dan Penjelasan Tugas No 1

Tugas No.1 meminta untuk membuat sebuah program yang melakukan operasi aritmatika pada dua bilangan yang dapat diinputkan sendiri dan kemudian memberikan outputnya. Algoritma programnya akan ditulis terlebih dahulu, dan akan terlihat seperti ini :

```
Program Aritmatika;

uses crt;
var a, b, c, d, e, f: real;

begin
  clrscr;

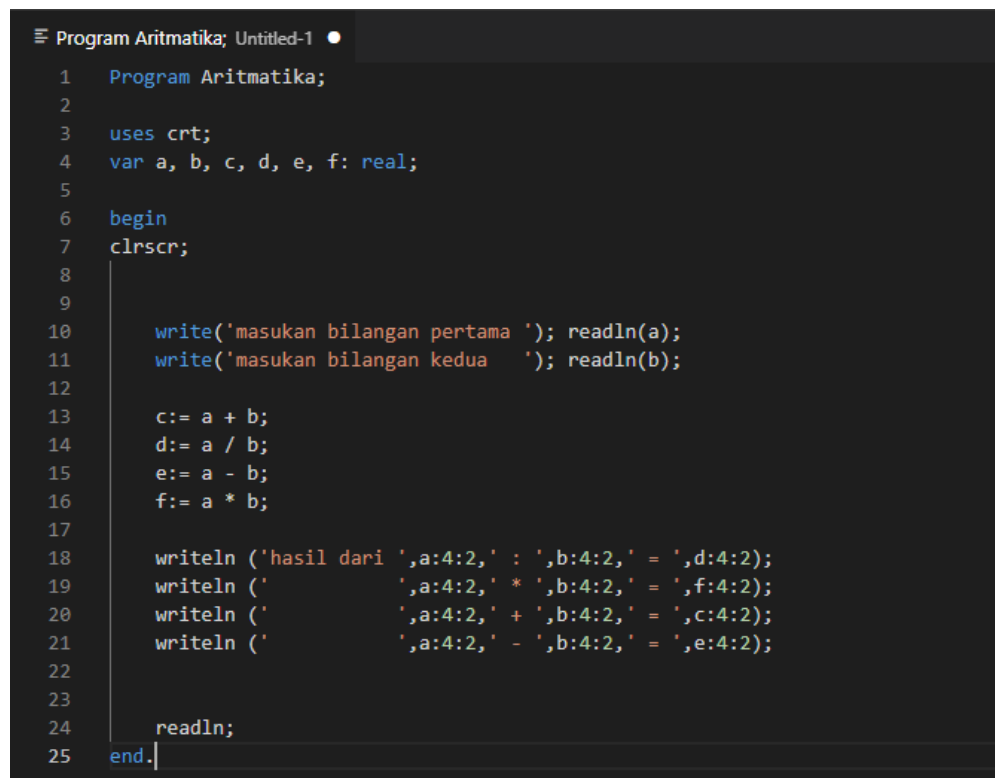
  writeln('Program Perhitungan Aritmatika')
  writeln('-----')
  write('masukan bilangan pertama '); readln(a);
  write('masukan bilangan kedua '); readln(b);

  c:= a + b;
  d:= a / b;
  e:= a - b;
  f:= a * b;

  writeln ('hasil dari ',a:4:2,' : ',b:4:2,' = ',d:4:2);
  writeln ('      ',a:4:2,' * ',b:4:2,' = ',f:4:2);
  writeln ('      ',a:4:2,' + ',b:4:2,' = ',c:4:2);
  writeln ('      ',a:4:2,' - ',b:4:2,' = ',e:4:2);
  readln;
end.
```

Algoritma tersebut menggunakan variabel berjenis real karena adanya operasi pembagian yang seringnya akan memberikan hasil pecahan, maka variabel real lebih cocok untuk digunakan. Terdapat enam variabel di algoritma tersebut, dua untuk bilangan yang akan dimasukan dan empat untuk hasil dari operasi aritmatikanya. Angka ':4:2' dituliskan di samping variabel hasil ketika ingin menuliskannya di output, adalah agar hasil yang tertulis hanya akan memiliki dua angka di belakang tanda koma.

Jika algoritma tersebut dimasukan kedalam aplikasi Visual Studio Code maka akan terlihat seperti berikut:

The image shows a screenshot of the Visual Studio Code editor interface. The title bar at the top reads "Program Aritmatika; Untitled-1". The editor contains a Pascal program with 25 lines of code. The code defines variables a, b, c, d, e, and f of type real. It prompts the user to input two numbers, a and b. Then it calculates the sum (c), product (d), difference (e), and quotient (f) of a and b. Finally, it prints out these results in a formatted manner using the ':4:2' format specifier for the output values. The code is as follows:

```
1 Program Aritmatika;
2
3 uses crt;
4 var a, b, c, d, e, f: real;
5
6 begin
7   clrscr;
8
9
10  write('masukan bilangan pertama '); readln(a);
11  write('masukan bilangan kedua   '); readln(b);
12
13  c:= a + b;
14  d:= a / b;
15  e:= a - b;
16  f:= a * b;
17
18  writeln ('hasil dari ',a:4:2,' : ',b:4:2,' = ',d:4:2);
19  writeln ('          ',a:4:2,' * ',b:4:2,' = ',f:4:2);
20  writeln ('          ',a:4:2,' + ',b:4:2,' = ',c:4:2);
21  writeln ('          ',a:4:2,' - ',b:4:2,' = ',e:4:2);
22
23
24  readln;
25 end.
```

Gambar Program Aritmatika Di Visual Studio Code

Dari program tersebut maka output dari Visual Studio Code akan terlihat seperti ini seperti berikut:

```
masukan bilangan pertama 18
masukan bilangan kedua 14
hasil dari 18.00 : 14.00 = 1.29
           18.00 * 14.00 = 252.00
           18.00 + 14.00 = 32.00
           18.00 - 14.00 = 4.00
```

Gambar Output Dari Program Aritmatika

Dari program diatas dapat disimpulkan beberapa hal, seperti jika ada operasi pembagian lebih baik menggunakan variabel real dan ketika menuliskan hasil dari operasi aritmatikanya, harus menulis':4:2'. Agar angka dari hasil yang ditulis hanya memiliki dua angka dibelakang tanda koma. Jika ingin agar hasil operasinya memiliki lebih dari dua angka dibelakang tanda koma maka cuma perlu mengubah angka '2' di ':4:2' menjadi jumlah angka di belakang koma yang diinginkan dibelakang tanda koma.

#### 2.1.2. Langkah Penyelesaian dan Penjelasan Tugas No 2

Tugas No.2 meminta untuk membuat sebuah program yang melakukan operasi perhitungan dengan rumus yang ada di soal. Dengan input a,b,dan c dapat diinputkan sendiri dan kemudian dapat memberikan output yang benar. Algoritma programnya akan ditulis terlebih dahulu, dan akan terlihat seperti ini :

Program Perhitungan No2;

uses crt;

var a, b, c, x1, x2: real;

begin

clrscr;

writeln('Program Untuk Mencari x1, dan x2');

writeln('-----');

write('masukan bilangan pertama '); readln(a);

write('masukan bilangan kedua '); readln(b);

write('masukan bilangan ketiga '); readln(c);

$x1 := (-b + \sqrt{b^2 - 4ac}) / 2a$ ;

$x2 := (-b - \sqrt{b^2 - 4ac}) / 2a$

writeln('hasil x1 bernilai = ', x1:4:2, ' dan hasil x2 adalah = ', x2:4:2);

readln;

end.

Untuk menjelaskan apa itu program yang digunakan, maka di programnya ditulis terlebih dahulu `writeln('Program untuk Mencari x1, dan x2')`. Setelah itu baru masuk ke dalam perhitungannya, dan algoritma tersebut menggunakan variabel real karena hasil yang diberi berkemungkinan merupakan bilangan pecahan. Di algoritma dapat dilihat setelah memasukan bilangan a, b, dan c, ketiga bilangan tersebut akan dimasukan kedalam rumus yang telah diberikan soal. Setelah mendapatkan hasilnya, maka akan ditulis dengan `':4:2'` agar angka dibelakang koma hanya ada dua.

Jika dimasukan kedalam Visual Studio Code, maka akan terlihat seperti ini:



```

1  Program persamaanlinear3variabel;
2
3  uses crt;
4  var a, b, c, x1, x2: real;
5
6  begin
7  clrscr;
8
9      writeln('Program Untuk Mencari x1, dan x2');
10     writeln('-----');
11     write('masukan bilangan a '); readln(a);
12     write('masukan bilangan b '); readln(b);
13     write('masukan bilangan c '); readln(c);
14
15     x1:= (-b + sqrt(sqr(b)-4*a*c))/2*a;
16     x2:= (-b - sqrt(sqr(b)-4*a*c))/2*a;
17     writeln;
18
19     writeln('hasil x1 adalah = ',x1:4:2, ' dan hasil x2 adalah = ',x2:4:2);
20     readln;
21 end.

```

Gambar Program Persamaanlinear3variabel di Visual Studio Code

Dan outputnya akan terlihat seperti ini:

```

PROBLEMS  TERMINAL  OUTPUT  DEBUG CONSOLE

Program Untuk Mencari x1, dan x2
-----
masukan bilangan a 12
masukan bilangan b 189
masukan bilangan c 12

hasil x1 adalah = -9.18 dan hasil x2 adalah = -2258.82

```

Output Program Linear3 variabel di Visual Studio Code

Dari kedua gambar tersebut dapat dibuktikan bahwa nilai outputnya bernilai benar. Karena telah sesuai dengan rumus yang diberikan oleh soal.

### 2.1.3. Langkah Penyelesaian dan Penjelasan Tugas No 3

Di tugas No. 3, telah diberikan algoritma untuk program yang akan kita buat. Algoritma tersebut jika kita tuliskan akan terlihat seperti ini:

```

Program No3;

uses crt;

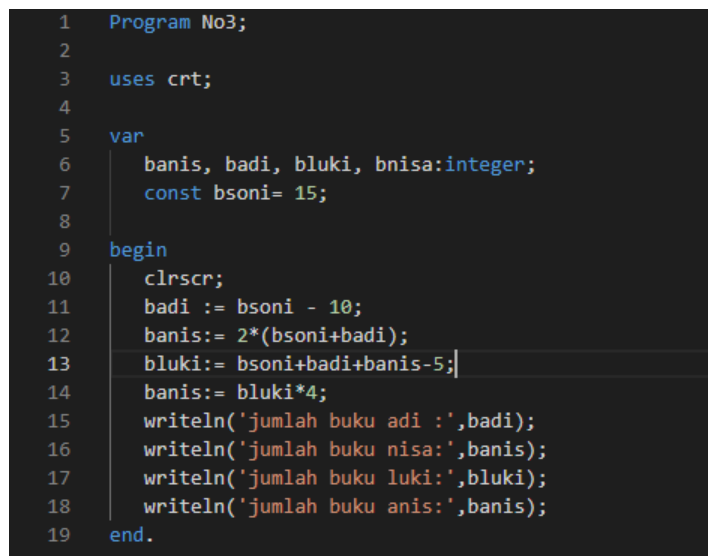
var
    banis, badi, bluki, bnisa:integer;
    const bsoni= 15;

begin
    clrscr;
    badi := bsoni - 10;
    banis:= 2*(bsoni+badi);
    bluki:= bsoni+badi+banis-5;
    banis:= bluki*4;
    writeln('jumlah buku adi :',badi);
    writeln('jumlah buku nisa:',banis);
    writeln('jumlah buku luki:',bluki);
    writeln('jumlah buku anis:',banis);
end.

```

Dari algoritma di atas, dapat dilihat bahwa programnya menggunakan variabel integer dan const. Di mana const adalah nilai tetap dan integer adalah nilai dari hasil perhitungan. Banis, badi, bluki, bnisa menggunakan integer dan bsoni menggunakan const dengan nilai 15.

Jika dituliskan di Visual Studio Code akan terlihat seperti ini:



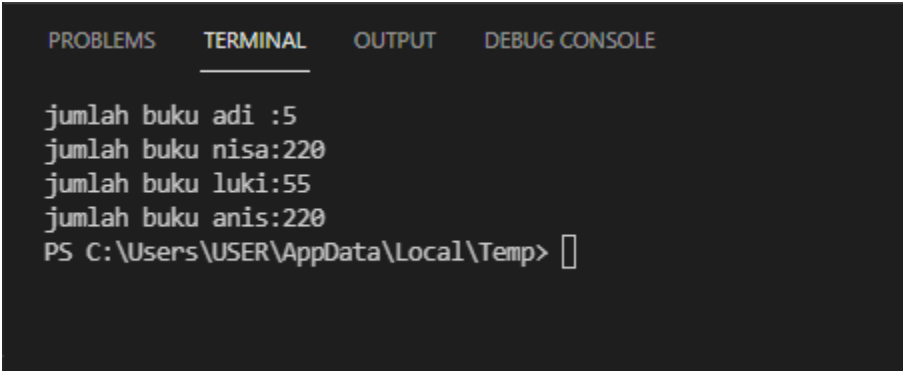
```

1  Program No3;
2
3  uses crt;
4
5  var
6      banis, badi, bluki, bnisa:integer;
7      const bsoni= 15;
8
9  begin
10     clrscr;
11     badi := bsoni - 10;
12     banis:= 2*(bsoni+badi);
13     bluki:= bsoni+badi+banis-5;
14     banis:= bluki*4;
15     writeln('jumlah buku adi :',badi);
16     writeln('jumlah buku nisa:',banis);
17     writeln('jumlah buku luki:',bluki);
18     writeln('jumlah buku anis:',banis);
19 end.

```

Gambar Program No. 3 di Visual Studio Code

Dan output dari program tersebut jika dijalankan adalah:



```
PROBLEMS  TERMINAL  OUTPUT  DEBUG CONSOLE

jumlah buku adi :5
jumlah buku nisa:220
jumlah buku luki:55
jumlah buku anis:220
PS C:\Users\USER\AppData\Local\Temp> 
```

Gambar Output Untuk Program No. 3

Dari gambar-gambar tersebut, apakah program tersebut dapat dibuktikan dengan menghitung sendiri dan jika dihitungkan hasilnya akan menjadi sama. Dari output dapat terlihat bahwa nisa dan anis memiliki nilai sama karena kedua buku tersebut menggunakan banis sebagai variabelnya dan karena banis dihitung dua kali, dan karena sifat runtutan maka perhitungan banis yang terakhir yang akan diambil, yaitu  $(banis = bluki * 4)$ , dan hasilnya 220.

#### 2.1.4. Langkah Penyelesaian dan Penjelasan Tugas No 4

Di tugas No. 4 juga telah diberikan algoritma dari soal, dan algoritma tersebut dibuat untuk mengkonversi waktu dari detik ke jam, menit, dan detik, jika ingin berbeda dengan contoh soal dan memberi hasil yang lebih baik maka akan terlihat seperti ini

```
Program konversidetik;

uses crt;
var detik, konvJam, konvMnt, konvDtk, sisa : real;

begin
  clrscr;

  writeln('Program Konversi Waktu');
  writeln('-----');
  write('masukan jumlah detik = '); readln(detik);
  konvJam := detik / 3600;
  konvMnt := detik / 60;
  konvDtk := detik;

  writeln('konversi jam adalah = ', konvJam:4:0);
  writeln('konversi menit adalah = ', konvMnt:4:2);
  writeln('dan konversi detik adalah = ', konvDtk:4:2);
  readln;

end.
```

Algoritma di atas tidak mengikuti contoh soal. Karena menggunakan variabel real, agar memberikan hasil yang sebenarnya. Karena pembagian sering memberikan hasil bilangan pecahan dan variabel integer tidak dapat menunjukkan bilangan pecahan. Program di atas menggunakan rumus yang cukup simpel, yaitu jumlah menit dibagi 3600 untuk konversi jam, dibagi 60 untuk konversi menit.

Jika dituliskan di dalam Visual Studio Code maka akan terlihat seperti ini:

```
1  Program konversidetik;
2
3  uses crt;
4  var detik, konvJam, konvMnt, konvDtk, sisa : real;
5
6  begin
7      clrscr;
8
9      writeln('Program Konversi Waktu');
10     writeln('-----');
11     write('masukan jumlah detik = '); readln(detik);
12     konvJam := detik / 3600;
13     konvMnt := detik / 60;
14     konvDtk := detik;
15
16     writeln('konversi jam adalah =      ',konvJam:4:0);
17     writeln('konversi menit adalah =    ',konvMnt:4:2);
18     writeln('dan konversi detik adalah = ',konvDtk:4:2);
19     readln;
20
21 end.
```

Gambar Program Konversidetik di Visual Studio Code.

Dan output dari program tersebut akan terlihat seperti ini:

```
PROBLEMS  TERMINAL  OUTPUT  DEBUG CONSOLE
Program Konversi Waktu
-----
masukan jumlah detik = 72000
konversi jam adalah =      20
konversi menit adalah =    1200.00
dan konversi detik adalah = 72000.00

```

Gambar Output Program konversidetik di Visual Studio Code

Dari contoh di atas dapat terlihat bahwa variabel program seharusnya real agar mendapatkan hasil yang benar. Karena jumlah yang di inputkan adalah jumlah detik maka

konversi detik tidak diperlukan dan jumlah konversi detik sama dengan jumlah detik yang di inputkan.

## **KESIMPULAN**

Di modul 2 yang dipelajari adalah runtutan (sequential), seperti apa itu runtutan, sifat runtutan, dan bagaimana ketentuan-ketentuan menggunakan runtutan di penulisan program Pascal. Di soal No. 3 hal ini terlihat jelas karena variabel banis memiliki dua perhitungan dan ketika kita ingin menuliskannya ke dalam output, hanya output dari perhitungan variabel banis yang terakhir yang terlihat. Hal ini dikarenakan sifat runtutan yang membuat intruksi yang paling akhir yang dianggap intruksi yang benar. Di program runtutan juga tidak ada pengulangan dan setiap kode hanya dikerjakan sekali dan intruksi akhir adalah akhir dari program tersebut. Maka pemahaman runtutan sangat penting dipahami sebagai dasar untuk menulis bahasa pemrograman

## **DAFTAR PUSTAKA**

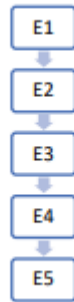
Fungsi Div Dan Mod Pada Pascal

<https://annur24.wordpress.com/2014/01/16/216/> [Diakses pada tanggal 15 Oktober 2021]

Runtutan Dalam Bahasa Pascal

<http://emerer.com/runtunan/> [Diakses pada tanggal 15 Oktober 2021]

## LAMPIRAN



Gambar 1 Runtunan yang Terdiri dari 5 Buah Instruksi

```
Program Aritmatika; Untitled-1
1  Program Aritmatika;
2
3  uses crt;
4  var a, b, c, d, e, f: real;
5
6  begin
7  clrscr;
8
9
10     write('masukan bilangan pertama '); readln(a);
11     write('masukan bilangan kedua   '); readln(b);
12
13     c:= a + b;
14     d:= a / b;
15     e:= a - b;
16     f:= a * b;
17
18     writeln ('hasil dari ',a:4:2,' : ',b:4:2,' = ',d:4:2);
19     writeln ('          ',a:4:2,' * ',b:4:2,' = ',f:4:2);
20     writeln ('          ',a:4:2,' + ',b:4:2,' = ',c:4:2);
21     writeln ('          ',a:4:2,' - ',b:4:2,' = ',e:4:2);
22
23
24     readln;
25 end.
```

Gambar Program Aritmatika Di Visual Studio Code



```
masukan bilangan pertama 18
masukan bilangan kedua 14
hasil dari 18.00 : 14.00 = 1.29
           18.00 * 14.00 = 252.00
           18.00 + 14.00 = 32.00
           18.00 - 14.00 = 4.00
```

Gambar Output Dari Program Aritmatika

```
1 Program persamaanlinear3variabel;
2
3 uses crt;
4 var a, b, c, x1, x2: real;
5
6 begin
7   clrscr;
8
9   writeln('Program Untuk Mencari x1, dan x2');
10  writeln('-----');
11  write('masukan bilangan a '); readln(a);
12  write('masukan bilangan b '); readln(b);
13  write('masukan bilangan c '); readln(c);
14
15  x1:= (-b + sqrt(sqr(b)-4*a*c))/2*a;
16  x2:= (-b - sqrt(sqr(b)-4*a*c))/2*a;
17  writeln;
18
19  writeln('hasil x1 adalah = ',x1:4:2, ' dan hasil x2 adalah = ',x2:4:2);
20  readln;
21 end.
```

Gambar Program Persamaanlinear3variabel di Visual Studio Code

```
PROBLEMS  TERMINAL  OUTPUT  DEBUG CONSOLE

Program Untuk Mencari x1, dan x2
-----
masukan bilangan a 12
masukan bilangan b 189
masukan bilangan c 12

hasil x1 adalah = -9.18 dan hasil x2 adalah = -2258.82
█
```

Output Program Linear3 variabel di Visual Studio Code

```
1  Program No3;
2
3  uses crt;
4
5  var
6      banis, badi, bluki, bnisa:integer;
7      const bsoni= 15;
8
9  begin
10     clrscr;
11     badi := bsoni - 10;
12     banis:= 2*(bsoni+badi);
13     bluki:= bsoni+badi+banis-5;
14     banis:= bluki*4;
15     writeln('jumlah buku adi :',badi);
16     writeln('jumlah buku nisa:',banis);
17     writeln('jumlah buku luki:',bluki);
18     writeln('jumlah buku anis:',banis);
19 end.
```

Gambar Program No. 3 di Visual Studio Code

```
PROBLEMS  TERMINAL  OUTPUT  DEBUG CONSOLE

jumlah buku adi :5
jumlah buku nisa:220
jumlah buku luki:55
jumlah buku anis:220
PS C:\Users\USER\AppData\Local\Temp> █
```

Gambar Output Untuk Program No. 3

```

1  Program konversidetik;
2
3  uses crt;
4  var detik, konvJam, konvMnt, konvDtk, sisa : real;
5
6  begin
7      clrscr;
8
9      writeln('Program Konversi Waktu');
10     writeln('-----');
11     write('masukan jumlah detik = '); readln(detik);
12     konvJam := detik / 3600;
13     konvMnt := detik / 60;
14     konvDtk := detik;
15
16     writeln('konversi jam adalah =      ',konvJam:4:0);
17     writeln('konversi menit adalah =    ',konvMnt:4:2);
18     writeln('dan konversi detik adalah = ',konvDtk:4:2);
19     readln;
20
21 end.

```

Gambar Program Konversidetik di Visual Studio Code.

```

PROBLEMS  TERMINAL  OUTPUT  DEBUG CONSOLE
Program Konversi Waktu
-----
masukan jumlah detik = 72000
konversi jam adalah =      20
konversi menit adalah =    1200.00
dan konversi detik adalah = 72000.00

```

Gambar Output Program konversidetik di Visual Studio Code

**LAPORAN RENCANA PRAKTIKUM  
ALGORITMA DAN PEMOGRAMAN I**



**Nama : Jordan Setiawan Nanyan**  
**NIM : 213020503058**  
**Kelas : D**  
**Modul : RUNTUTAN (SEQUENTIAL)**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS PALANGKA RAYA**

**2021**

## **MODUL II**

### **RUNTUTAN (SEQUENTIAL)**

#### **1.1.Tujuan :**

- 1.1.1. Dapat menggolongkan sebuah masalah ke dalam kategori runtunan (sequential).
- 1.1.2. Dapat memecahkan permasalahan yang sifatnya runtunan ke dalam sebuah program.

#### **1.2.Landasan Teori :**

Runtunan adalah struktur algoritma paling dasar yang berisi rangkaian instruksi yang diproses secara sekuensial, artinya dikerjakan satu per satu, mulai dari instruksi pertama sampai instruksi akhir.

Sebuah instruksi dieksekusi setelah instruksi sebelumnya selesai dieksekusi. Urutan instruksi menentukan keadaan hasil dari algoritma. Bila urutannya diubah, maka hasil akhirnya mungkin juga berubah. Instruksi algoritma pada struktur runtunan ini akan dikerjakan semua dan tidak ada yang diabaikan. Pada dasarnya, semua program dibuat berdasarkan struktur algoritma runtunan.

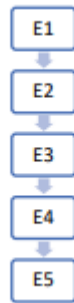
Menurut Goldshlager dan Lister (1998) struktur berurutan atau runtunan mengikuti ketentuan-ketentuan sebagai berikut.:

- e) Tiap instruksi dikerjakan satu persatu.
- f) Tiap instruksi dilaksanakan tepat sekali, tidak ada yang diulang.
- g) Urutan instruksi yang dilaksanakan pemroses sama dengan urutan aksi sebagaimana yang tertulis di dalam algoritmanya.
- h) Akhir dari instruksi terakhir merupakan akhir algoritma.

Setiap instruksi di dalam runtunan ditulis dalam satu baris, atau beberapa instruksi dalam baris yang sama tetapi antara setiap instruksi dipisahkan dengan tanda titik koma (;).

Contoh struktur runtunan adalah sebagai berikut.

Bila suatu instruksi runtunan dalam algoritma ditulis berturut-turut, kita lambangkan dengan variabel E1, E2, E3, E4, E5, maka urutan pelaksanaan instruksi tersebut adalah seperti gambar 1 dibawah ini:



Gambar 1 Runtunan yang Terdiri dari 5 Buah Instruksi

Keterangan :

Runtunan diatas, dimulai dari intruksi E1, kemudian E1 harus dikerjakan sampai selesai terlebih dahulu sebelum instruksi E2 dikerjakan, setelah instruksi E2 selesai dikerjakan juga, maka instruksi E3 dikerjakan lagi dan seterusnya sampai instruksi ke E5 setelah itu proses selesai dikerjakan dan baru berhenti dari proses pembacaan runtunan tersebut.

### 1.3.Tugas

1. Buatlah program untuk melakukan operasi aritmatika dimana bilangan pertama dan kedua merupakan inputan dari piranti masukan.
2. Buatlah program untuk mencari nilai  $x_1$  dan  $x_2$  dari persamaan linear  $ax^2 + bx + c = 0$  dengan rumus :

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Dimana nilai a, b dan c merupakan inputan dari piranti masukan.

3. Algoritma Hitung Buku

Deklarasi

```

    Const Bsoni = 15
var
    Badi, Banis, Bluki, Bnisa: integer
Deskripsi

```

```

    Badi ← Bsoni-10
    Banis ← 2*(Bsoni+Badi)
    Bluki ← Bsoni+Badi+Banis-5
    Banis ← Bluki*4
    Write(Badi, Banis, Bluki, Banis)

```

Buatlah program pascal untuk menghitung buku Adi, Anis, Luki, dan Nisa berdasarkan algoritma diatas.

#### 4. Algoritma KonversiDetik

```

Deklarasi
var
    Detik, KonvJam, KonvMnt, KonvDtk, Sisa : integer
Deskripsi
read(Detik)
KonvJam ← Detik div 3600
Sisa ← Detik mod 3600
KonvMnt ← Sisa div 60
KonvDtk ← Sisa mod 60
write(KonvJam, KonvMnt, KonvDtk)

```

Buatlah program pascal untuk mengkonversi waktu dari satuan detik ke satuan jam, menit, dan detik.