

Pertemuan 5



Contoh program Pencabangan

Mata Kuliah : Algoritma & Pemrograman
Dosen : Tessy Badriyah, SKom., MT., PhD.



Tujuan Pembelajaran

- Membahas contoh-contoh persoalan pencabangan yang meliputi :
 - Menemukan akar persamaan kuadrat Mencari tahun kabisat
 - Memeriksa apakah sebuah karakter termasuk alfabet atau bukan
 - Faktorial dari suatu bilangan
 - Mencari Faktor Persekutuan Terbesar (FPB)

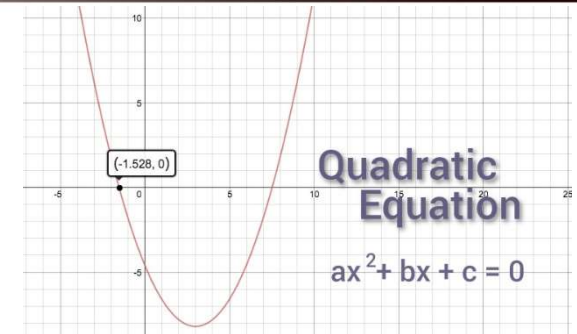


Mencari akar persamaan kuadrat

- Bentuk persamaan kuadrat:

$ax^2 + bx + c = 0$, dimana
 a, b dan c adalah bilangan real dan
 $a \neq 0$

- $b^2 - 4ac$ disebut dengan determinan dari persamaan kuadrat
- Determinan menentukan akar.
 - Jika determinan lebih dari 0, maka akar merupakan bilangan real dan berbeda.
 - Jika determinan sama dengan 0, maka akart adalah bilangan real dan sama.
 - Jika determinan kurang dari 0, maka akar merupakan bilangan kompleks dan berbeda.



If determinant > 0 ,	$\text{root1} = \frac{-b + \sqrt{(b^2 - 4ac)}}{2a}$ $\text{root2} = \frac{-b - \sqrt{(b^2 - 4ac)}}{2a}$
If determinant $= 0$,	$\text{root1} = \text{root2} = \frac{-b}{2a}$
If determinant < 0 ,	$\text{root1} = \frac{-b}{2a} + i \frac{\sqrt{-(b^2 - 4ac)}}{2a}$ $\text{root2} = \frac{-b}{2a} - i \frac{\sqrt{-(b^2 - 4ac)}}{2a}$



Mencari akar persamaan kuadrat

```
include <stdio.h>
#include <math.h>
int main()
{
    double a, b, c, determinan, akar1, akar2, realPart,
    imaginaryPart;
    printf("Masukkan koefisien a,b dan c: ");
    scanf("%lf %lf %lf", &a, &b, &c);
    determinan = b*b-4*a*c;
    // jika determinan >0 maka ada 2 akar yang berbeda
    if (determinan > 0)
    {
        // sqrt() mengembalikan nilai akar
        akar1 = (-b+sqrt(determinan))/(2*a);
        akar2 = (-b-sqrt(determinan))/(2*a);
        printf("akar1 = %.2lf and akar2 = %.2lf", akar1 ,
        akar2);
    }
}
```

```
//jika determinan=0 maka terdapat dua akar kembar
```

```
else if (determinan == 0)
```

```
{
```

```
    akar1 = akar2 = -b/(2*a);
```

```
    printf("akar1 = akar2 = %.2lf;", akar1);
```

```
}
```

```
// jika determinan <0 maka akarnya tidak real
(imaginer)
```

```
else
```

```
{
```

```
    realPart = -b/(2*a);
```

```
    imaginaryPart = sqrt(-determinan)/(2*a);
```

```
    printf("akar1 = %.2lf+%.2lfi and akar2 = %.2f-%.2fi",
    realPart, imaginaryPart, realPart, imaginaryPart);
```

```
}
```

```
    return 0;
```

```
}
```

Output :

Masukkan koefisien a,b dan c: 2.3

4

5.6

akars are: -0.87+1.30i and -0.87-1.30i



Memeriksa apakah suatu bilangan termasuk Kabisat atau tidak

- Tahun kabisat adalah tahun dimana bulan Februari sampai tanggal 29.
- Algoritma tahun kabisat telah dijelaskan pada pertemuan sebelumnya
- Berikut ini adalah program untuk menentukan apakah suatu tahun yang diinputkan adalah kabisat atau tidak.

```
#include <stdio.h>

int main()
{ int tahun;
  printf("Masukkan tahun: "); scanf("%d",&tahun);
  if(tahun%4 == 0)
  { if( tahun%100 == 0)
    { // jika tahun habis dibagi 400, maka tahun adalah kabisat
      if ( tahun%400 == 0)   printf("%d adalah kabisat.", tahun);
      else   printf("%d adalah bukan kabisat.", tahun);
    }
    else   printf("%d adalah kabisat.", tahun );
  }
  else   printf("%d adalah bukan kabisat.", tahun);
  return 0;
}
```

Output :

Masukkan tahun: 1900
1900 adalah bukan kabisat.

Masukkan tahun: 2012
2012 adalah kabisat.



Memeriksa apakah suatu karakter yang diinputkan termasuk alfabet atau bukan

```
#include <stdio.h>
int main()
{
    char c;
    printf("Masukkan karakter: ");
    scanf("%c",&c);

    if( (c>='a' && c<='z') || (c>='A' && c<='Z'))
        printf("%c is an alfabet.",c);
    else
        printf("%c bukan alfabet.",c);

    return 0;
}
```

Output :

Masukkan karakter: *
* bukan alfabet



Mencari Faktorial suatu bilangan

- Bilangan factorial dari n didapatkan dengan

Faktorial dari n ($n!$) = $1*2*3*4....n$

- Faktorial dari bilangan negatif adalah tidak ada, dan faktorial dari bilangan nol adalah satu, $0! = 1$

Output :

Masukkan integer: 10

Faktorial dari 10 = 3628800

```
#include <stdio.h>

int main()
{   int n, i;
    unsigned long long factorial = 1;
    printf("Masukkan integer: ");
    scanf("%d",&n);
    // tampilkan kerror jika user memasukkan integer negatif
    if (n < 0)
        printf("Salah! Faktorial dari bilangan negative itu tidak
ada.");
    else
    {
        for(i=1; i<=n; ++i)
        {
            factorial *= i;          // factorial = factorial*i;
        }
        printf("Faktorial dari %d = %llu", n, factorial);
    }
    return 0;
}
```



Mencari Faktor Persekutuan Terbesar (FPB)

- Faktor Persekutuan Terbesar (FPB) dari dua bilangan integer adalah bilangan integer terbesar yang dapat membagi kedua bilangan tanpa ada sisa.
- Pada program berikut, dua integer dimasukkan oleh user kemudian disimpan dalam variable n1 dan n2. Kemudian for loop akan melakukan iterasi sampai counter i bernilai kurang dari n1 dan n2.
- Pada setiap iterasi, kedua integer n1 dan secara tepat dapat dibagi oleh bilangan tanpa ada sisa.
- Setelah for loop selesai, maka bilangan pembagi terbesar akan disimpan dalam variable gcd.

```
#include <stdio.h>


int main()
{
    int n1, n2, i, gcd;
    printf("Masukkan dua bil. integer: ");
    scanf("%d %d", &n1, &n2);
    for(i=1; i <= n1 && i <= n2; ++i)
    {
        // periksa apakah i adalah faktor
        // dari kedua bilangan
        if(n1%i==0 && n2%i==0)
            gcd = i;
    }
    printf("FPB dari %d dan %d adalah %d",
n1, n2, gcd);
    return 0;
}
```




Mencari Faktor Persekutuan Terbesar (FPB)

Penjelasan program :

- Faktor Persekutuan Terbesar (FPB) dari dua bilangan integer adalah bilangan integer terbesar yang dapat membagi kedua bilangan tanpa ada sisa.
- Pada program berikut, dua integer dimasukkan oleh user kemudian disimpan dalam variable $n1$ dan $n2$. Kemudian for loop akan melakukan iterasi sampai counter i bernilai kurang dari $n1$ dan $n2$.
- Pada setiap iterasi, kedua integer $n1$ dan $n2$ secara tepat dapat dibagi oleh bilangan tanpa ada sisa.
- Setelah for loop selesai, maka bilangan pembagi terbesar akan disimpan dalam variable gcd .



Mencari Faktor Persekutuan Terbesar (FPB) (2)


- Berikut ini alternative program dengan cara lain untuk mencari factor persekutuan terbesar (FPB)

Output:

Masukkan dua bil. integer: 81
153
GCD = 9

```
#include <stdio.h>
int main()
{
    int n1, n2;

    printf("Masukkan dua bil. integer: ");
    scanf("%d %d",&n1,&n2);
    while(n1!=n2)
    {
        if(n1 > n2)
            n1 -= n2;
        else
            n2 -= n1;
    }
    printf("FPB = %d",n1);
    return 0;
}
```



Mencari Faktor Persekutuan Terbesar (FPB) (3)

```
#include <stdio.h>


int main()
{ int n1, n2;

  printf("Masukkan dua bil. integer: ");
  scanf("%d %d",&n1,&n2);

  // Jika input bil. Negatif, maka tanda akan diubah ke positif
  n1 = ( n1 > 0) ? n1 : -n1;
  n2 = ( n2 > 0) ? n2 : -n2;
  while(n1!=n2)
  { if(n1 > n2)
    { n1 -= n2;
    }
    else
    { n2 -= n1;
    }
  }
  printf("FPB = %d",n1);
  return 0;
}
```

Output :

Masukkan dua bil. integer: 81
-153
FPB = 9



Mencari Faktor Persekutuan Terbesar (FPB) (3)

Penjelasan program :

- Dua program untuk mencari factor persekutuan terbesar sebelumnya menerima input 2 bilangan integer positif.
- Program berikut ini dapat menerima input bilangan integer positif maupun negatif.



Membalik Bilangan

- Program berikut akan meminta input bilangan (n).
- Kemudian n digunakan sebagai counter pada While loop selama $n \neq 0$.
- Pada setiap iterasi dari while loop, sisa dari pembagian n dengan 10 dihitung kemudian nilai dari n direduksi setiap kali iterasi.

Output :

Masukkan integer: 2345
Reversed Number = 5432

```
#include <stdio.h>

int main()
{
    int n, reversedNumber = 0, remainder;
    printf("Masukkan integer: ");
    scanf("%d", &n);
    while(n != 0)
    {
        sisa = n%10;
        reversedNumber = reversedNumber*10 + sisa;
        n /= 10;
    }
    printf("Reversed Number = %d", reversedNumber);
    return 0;
}
```



Pangkat dari suatu bilangan

- Program berikut ini mencari pangkat dari dua bilangan.
- Misal dua bilangan yang diinputkan adalah a dan b, a adalah bilangan dasar dan b adalah pangkat.
- Contoh: Jika inputnya 23
 - 2 adalah bilangan dasar
 - 3 adalah pangkat
 - Sehingga 2 dipangkatkan dengan 3 sama dengan $2*2*2$

```
#include <stdio.h>

int main()
{
    int dasar, pangkat;
    long long result = 1;
    printf("Masukkan bilangan dasar: ");
    scanf("%d", &dasar);
    printf("dipangkatkan dengan : ");
    scanf("%d", &pangkat);
    while (pangkat != 0)
    {
        result *= dasar;
        --pangkat;
    }
    printf("Hasil = %lld", result);
    return 0;
}
```

Output:

Masukkan bilangan dasar: 3
dipangkatkan dengan : 4
Hasil = 81



Pangkat dari suatu bilangan

- Alternatif program yang lain untuk mencari pangkat dari suatu bilangan
- Dengan menggunakan fungsi `pow()` yang ada pada `math.h`

Output:

Masukkan bilangan dasar: 2.3
dipangkatkan dengan : 4.5
 $2.3^{4.5} = 42.44$

```
#include <stdio.h>
#include <math.h>
int main()
{
    double dasar, pangkat, result;
    printf("Masukkan bilangan dasar: ");
    scanf("%lf", &dasar);
    printf("dipangkatkan dengan : ");
    scanf("%lf", &pangkat);
    // calculates the power
    result = pow(dasar, pangkat);
    printf("%.1lf^%.1lf = %.2lf", dasar, pangkat, result);
    return 0;
}
```



Palindrome atau bukan?

- Program akan memeriksa suatu integer termasuk palindrome atau bukan.
- Bilangan palindrome adalah bilangan yang sama dengan kebalikannya.

```
#include <stdio.h>

int main()
{
    int n, reversedInteger = 0, sisa, originalInteger;
    printf("Masukkan integer: ");
    scanf("%d", &n);
    originalInteger = n;
    // reversed integer is stored in variable
    while( n!=0 )
    {
        sisa = n%10;
        reversedInteger = reversedInteger*10 + sisa;
        n /= 10;
    }
    // palindrome jika originalInteger dan reversedInteger sama
    if (originalInteger == reversedInteger)
        printf("%d adalah palindrome.", originalInteger);
    else
        printf("%d adalah bukan palindrome.", originalInteger);

    return 0;
}
```

Output :

Masukkan integer: 1001
1001 adalah palindrome.



Yang sudah dipelajari

- Contoh-contoh dari pencabangan dalam program Bahasa C yang dapat menyelesaikan persoalan dari pengambilan keputusan (*decision making*)



Referensi

- Robertson, Lesley Anne. (1992). *Students' guide to program design*. Oxford : Newnes
- Santner, Williams, and Notz (2003), *Design and Analysis of Computer Experiments*, Springer.
- Deitel & Deitel, *C How to Program*, Prentice Hall 1994 (2nd edition)
- Brookshear, J.G., *Computer Science: An Overview*, Benjamin-Cummings 2000 (6th edition)
- Kernighan & Ritchie, *The C Programming Language*, Prentice Hall