

### **JURUSAN INFORMATIKA**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS SANATA DHARMA YOGYAKARTA**

**LAPORAN PRAKTIKUM ALGORITMA DAN PEMROGRAMAN**

**Nama : Johanes Yogtan Wicaksono Raharja**

**NIM : 215314105**

1. **Uraian Soal**

Modifikasilah program latihan nomor 2 di atas sehingga program Anda dapat menghitung besarnya C(n,r) = n ! / [ (n – r) ! r ! ] di mana besarnya n dan r dimasukkan lewat keyboard serta memakai syarat bahwa r ≤ n.

1. **Rancangan Program**

**Output**

Program Menghitung Kombinasi C(n,r)

Masukkan Berapa Hitungan Kombinasi : 2

Masukkan Nilai N : 4

Masukkan Nilai R : 2

Hasil Kombinasi C(4,2) adalah 6

---------------------------------------

Masukkan Nilai N : 4

Masukkan Nilai R : 1

Hasil Kombinasi C(4,1) adalah 4

---------------------------------------

**Input**

Memesan memori untuk bilangan dengan nama variabel n,r, kombCNR, p, i, u, hasFak, dan nr

**Proses**

Membaca data dari keyboard (Long) disimpan ke variable p

//Awal For Bersarang

For, exspresi (int u = 1; u <= p; u++) //Mengulang 2 kali

//Awal Do

Do, menampilkan blok pernyataan ...

Membaca data dari keyboard (Int) disimpan ke variable n

Membaca data dari keyboard (Int) disimpan ke variable r

If, Jika (r>n) Maka, memberi tahu user (False)

While, apabila (r > n) perintah di blok pernyataan do akan terus diulang (False)

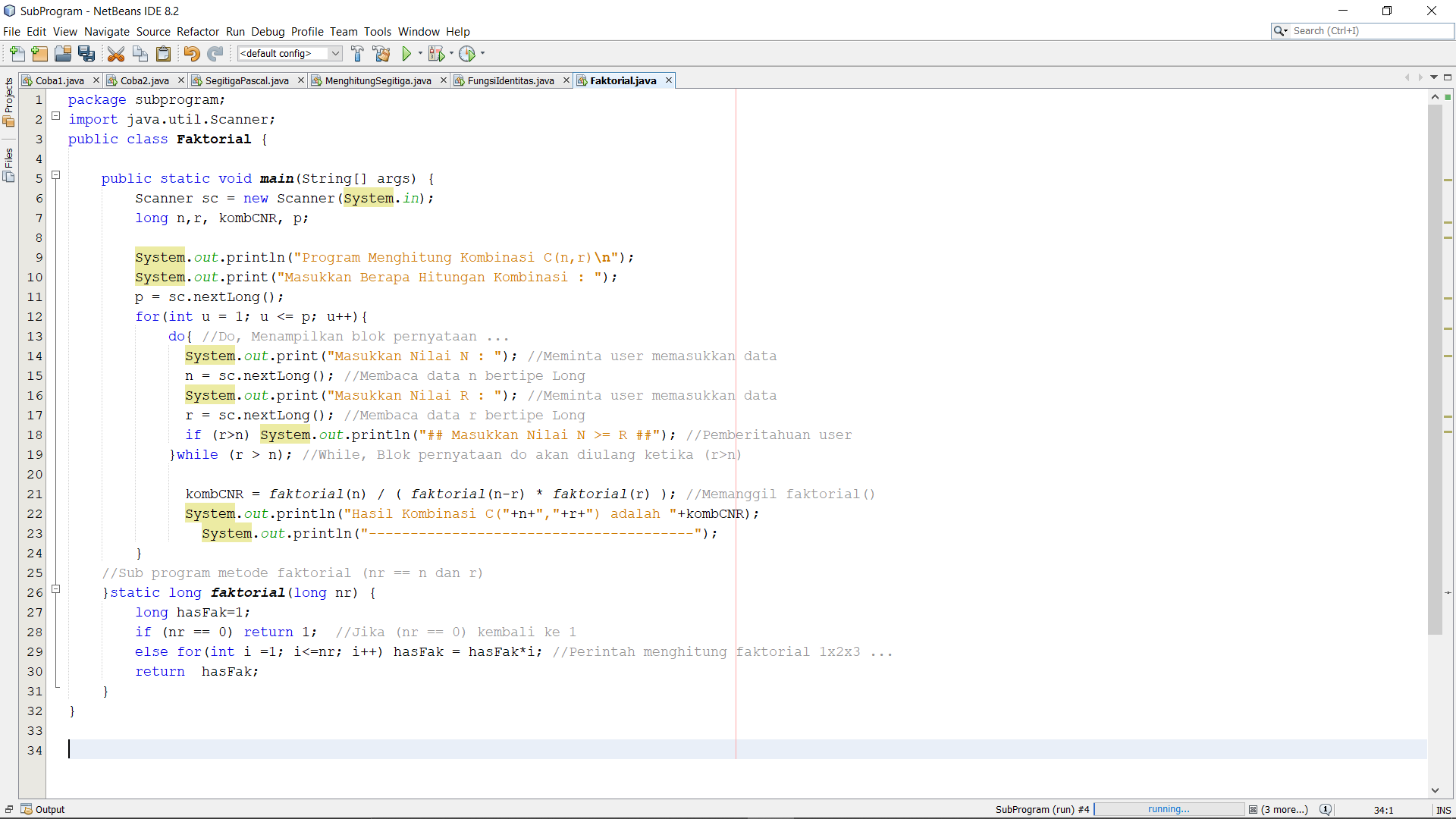
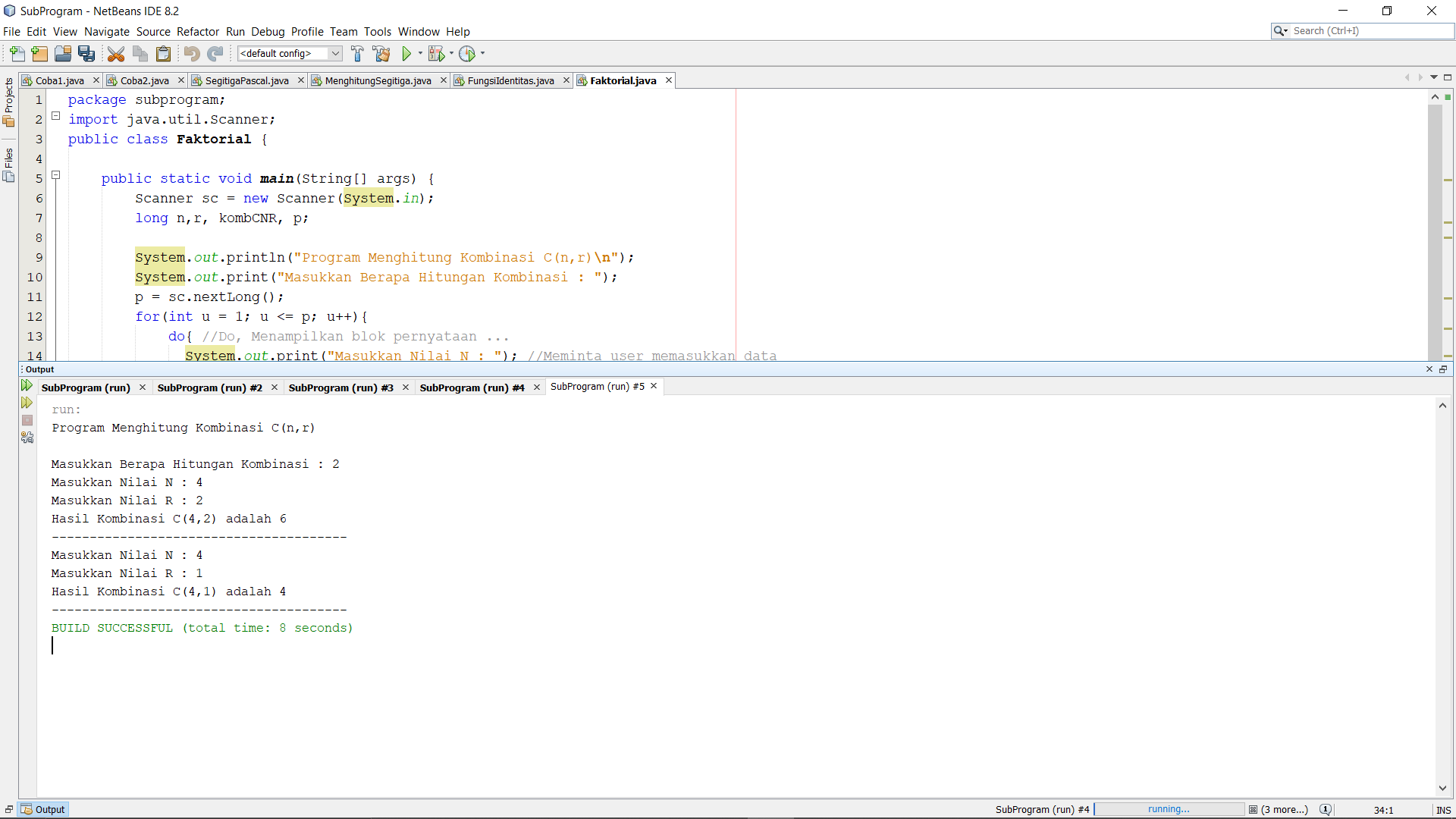
//Akhir Do

Memanggil dan menghitung factorial(n) dan (r) disimpan ke variabel kombCNR

//Akhir For Bersarang

Sub program faktorial, nr sebagai parameter sesuai pemanggilan

Else for, exspresi (int i =1; i<=nr; i++) menghitung hasil kali hasFak dan mengirimnya

1. **Capture Program**
2. **Capture Output**
3. **Uraian Soal**

Segitiga Pascal adalah daftar bilangan bulat yang disusun berdasarkan jumlah dua bilangan sebelah kiri dan kanan dari deret sebelumnya seperti dalam gambar di bawah

Angka-angka tersebut di atas juga dapat dihitung memakai rumus Kombinasi ( C(n,r) seperti dalam diagram berikut

C(n,r) dapat dihitung memakai rumus n ! / [(n-r) ! r !] di mana n ! = 1 x 2 x 3 x … x n

Sebagai contoh, C(4,2) = 4 ! / [(4-2) ! 2 !] = (1 x 2 x 3 x 4) / [ (1 x 2) (1 x 2)] = 6

Tugas Anda adalah menampilkan bilangan dari segitiga Pascal untuk baris atau n tertentu yang dimasukkan lewat keyboard.

Jika nilai n adalah 3 maka program Anda menampilkan angka 1 3 3 1 dan angka-angka ini dapat dihitung masing-masing dengan C(3,0), C(3,1), C(3,2), C(3,3).

Dengan kata lain, berdasarkan input besarnya n, program Anda harus menghitung dan menampilkan nilai dari C(n,0), C(n,1), C(n,2), …, C(n,(n-1)), C(n,n).

Jika Anda sudah selesai mengerjakan tugas ini, program dapat Anda modifikasi sehingga mampu menghasilkan semua bilangan dalam segitiga Pascal mulai dari n = 0 sampai n bernilai tertentu yang dimasukkan lewat keyboard.

1. **Rancangan Program**

**Output**

Program Segitiga Pascal

Masukkan Nilai N : 5

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

**Input**

Memesan memori untuk bilangan dengan nama variabel n, i, k, j, cij, r, fak, dan I

**Proses**

//Awal Do

Do, menampilkan blok pernyataan ...

Membaca data dari keyboard (Long) disimpan ke variable n

If, Jika (n < 0)Maka, memberi tahu user (False)

While, apabila (n < 0)perintah di blok pernyataan do akan terus diulang (False)

//Akhir Do

//Awal For Bersarang

For, exspresi (int i = 0; i<=n; i++) //Note : Diulang hingga 5x

For, exspresi (long k = n - 1 ; k >= i; k-=1) //Membentuk output seperti segitiga

For, exspresi (int j = 0 ; j <= i; j++)

Memanggil kombinasi(I,j) dan disimpan ke variabel cij

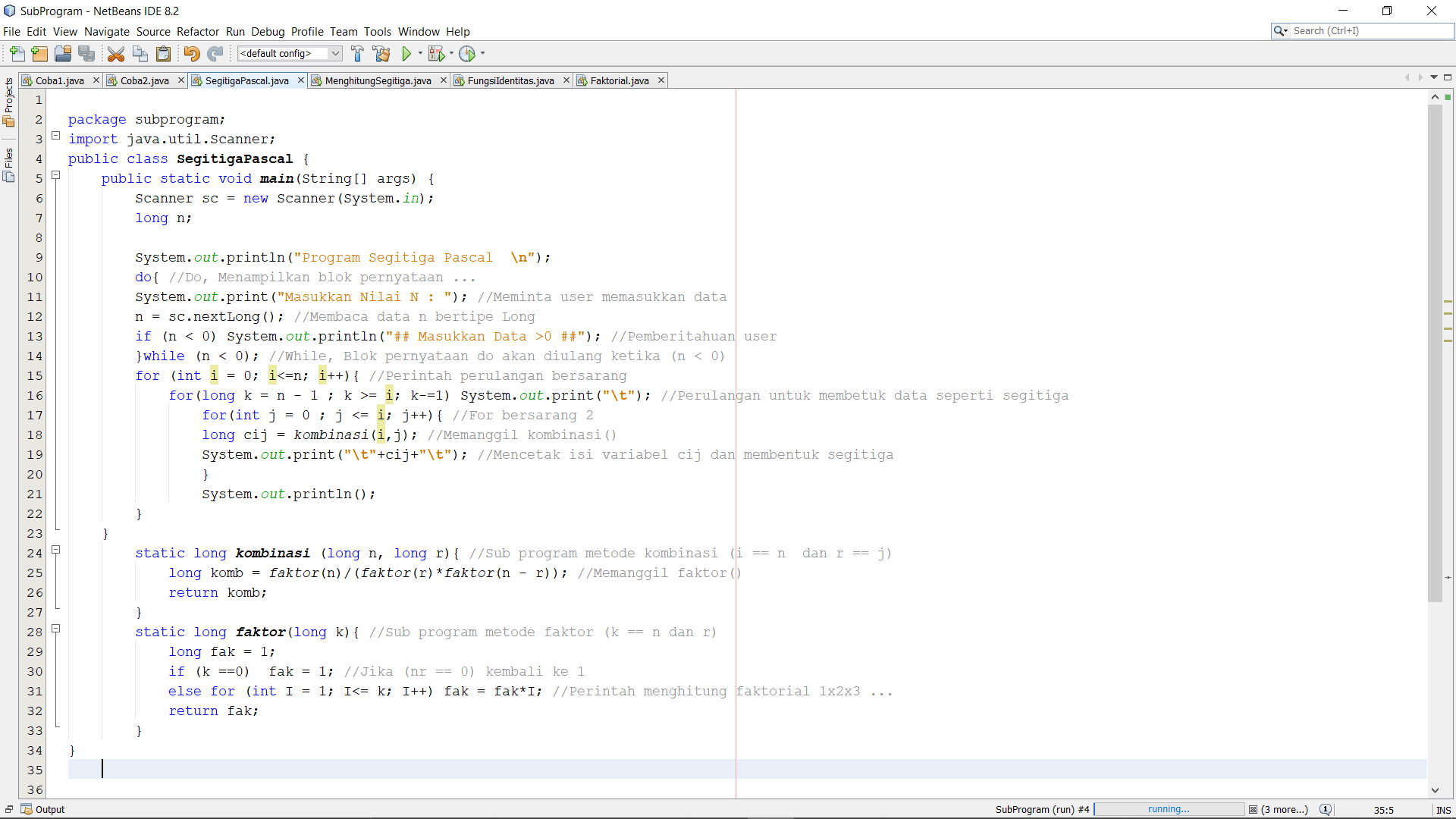
Sub program kombinasi, n dan r sebagai parameter sesuai pemanggilan

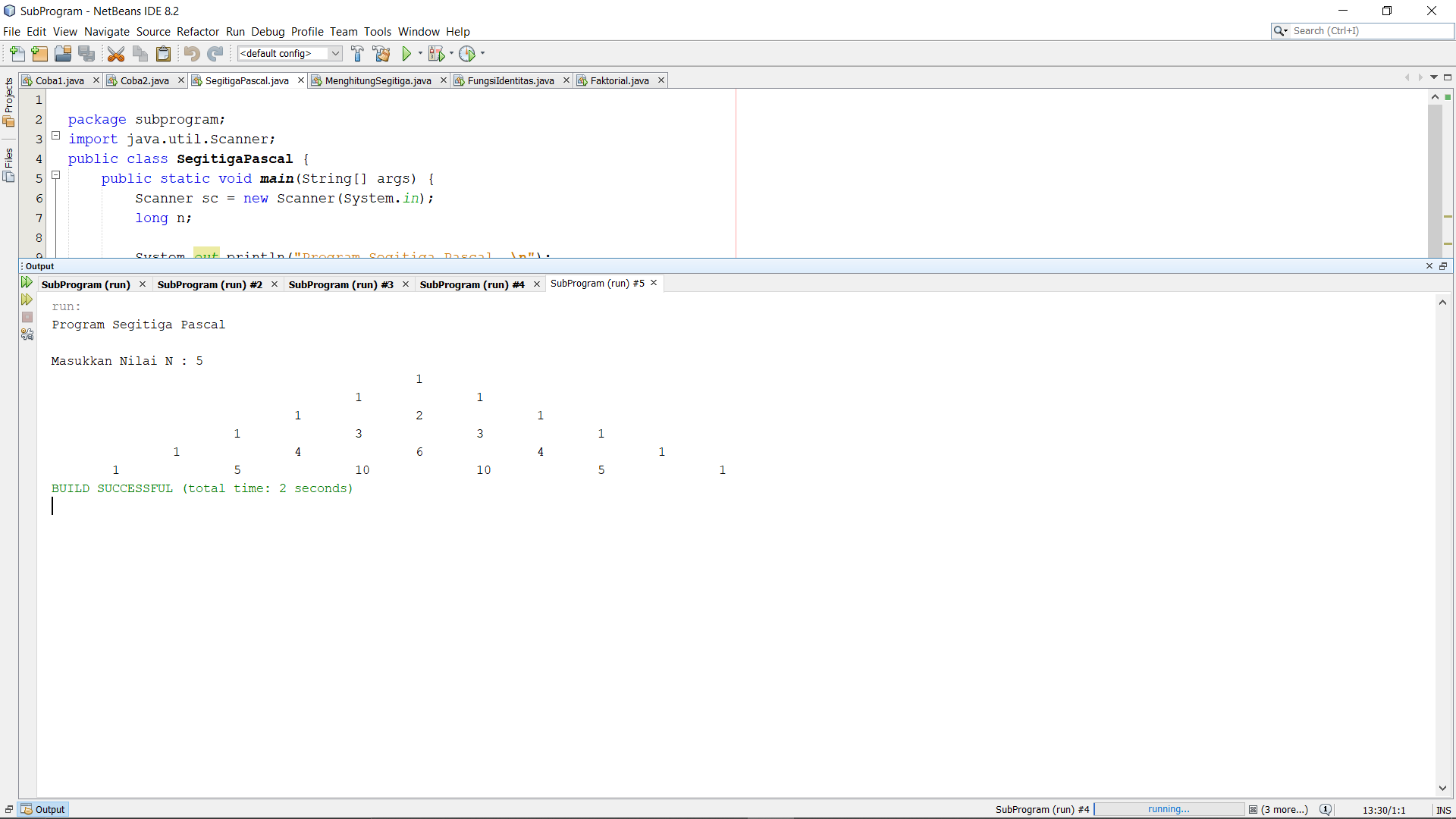
Memanggil faktor n dan r dan disimpan ke variabel komb menghitungnya serta mengirimnya

Sub program faktor, k sebagai parameter sesuai pemanggilan

If, Jika (k == 0), Maka fak = 1

Else for, exspresi (int I = 1; I<= k; I++) menghitung hasil kali hasFak dan mengirimnya

1. **Capture Program**
2. **Capture Output**

****

1. **Uraian Soal**

Buat program untuk menghitung luas segitiga berdasarkan koordinat 3 titik sudutnya yang dimasukkan lewat keyboard. Program Anda harus memakai beberapa metode berikut

1. bacaTitik() // membaca absis dan ordinat suatu titik dan tanpa nilai balikan
2. hitungJarak(int x1, int y1, int x2, int y2) // menghitung jarak antara 2 titik (x1,y1) dan (x2,y2) dengan nilai balikan (return) jarak dua titik
3. hitungKeliling(double sisiA, double sisiB, double sisiC) //dengan nilai balikan keliling segitiga atau jumlah sisiA, sisiB dan sisiC.
4. hitungLuas() // menghitung dan menampilkan luas segitiga memakai rumus yang sudah pernah diajarkan

1. **Rancangan Program**

**Output**

Menghitung Segitiga

===========================================

Masukkan Nilai Absis dan Ordinat

Titik A, B, dan C!!

> Titik A (X1, Y1)

Absis (X1) : -3

Ordinat (Y1) : 7

> Titik B (X2, Y2)

Absis (X2) : 1

Ordinat (Y2) : 4

> Titik C (X3, Y3)

Absis (X3) : -4

Ordinat (Y3) : 3

--------------------------------------------

Kita dapat mengetahui jarak antara titik,

keliling dan luas segitiga sebagai berikut ...

jarak AB : 5.0

jarak BC : 5.0990195135927845

jarak AC : 4.123105625617661

Keliling : 14.222125139210444

Luas : 9.499999999999996

**Proses**

Memanggil bacaTitik()

SubProgram bacaTitik, perintah ini akan membaca data dari keyboard sebagai input awal

Memanggil hitungJarak(x2, x1, y2, y1) hitungJarak(x3, x2, y3, y2) hitungJarak(x3, x1, y3, y1) masing masing disimpan ke variabel AB BC dan AC

SubProgram hitungJarak, a1, a2, b1, b2 sebagai parameter sesuai pemanggilan dan menghitung hasil kali dan mengirimnya

Memanggil hitungKeliling() dan disimpan ke variabel kel

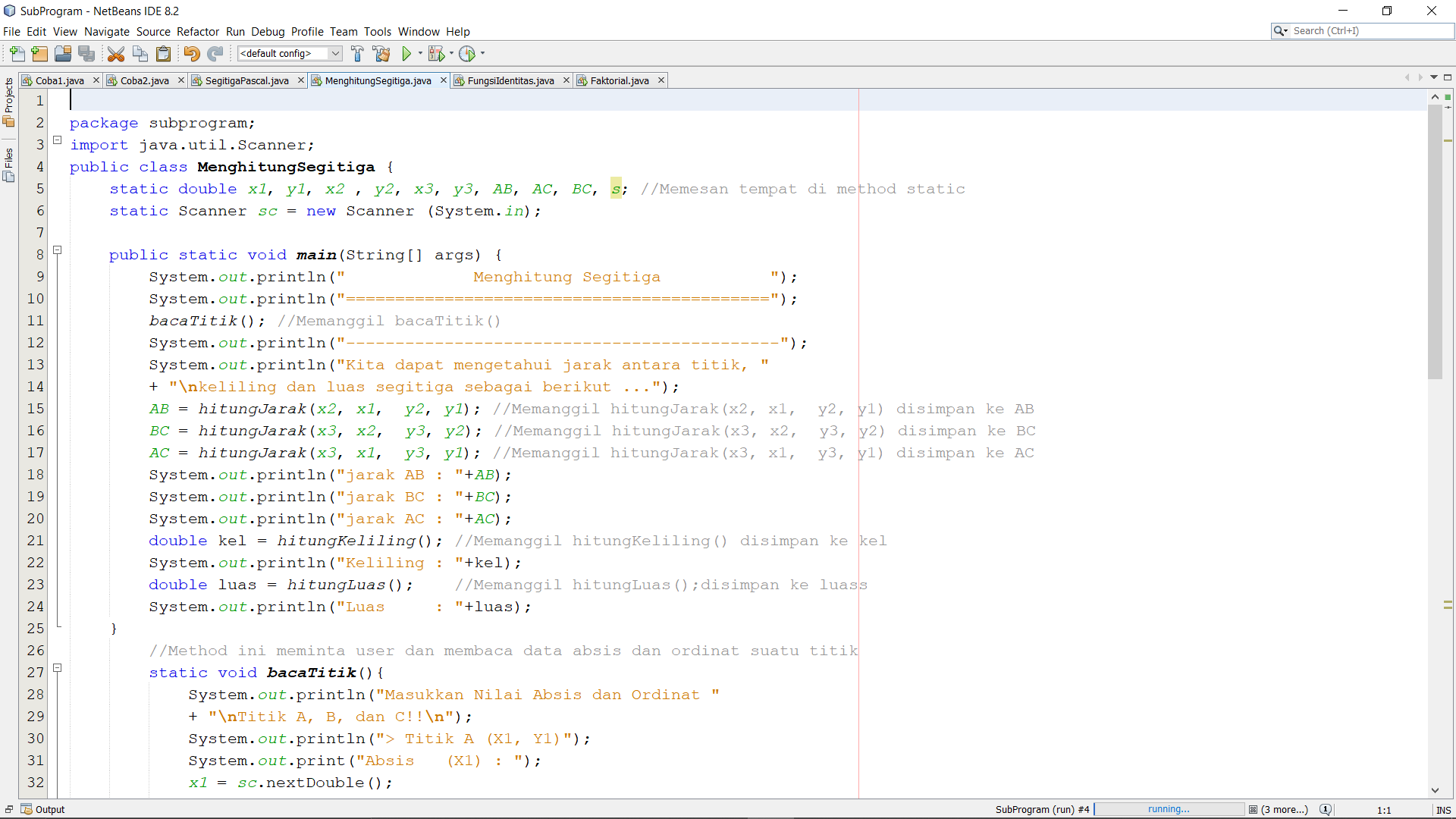
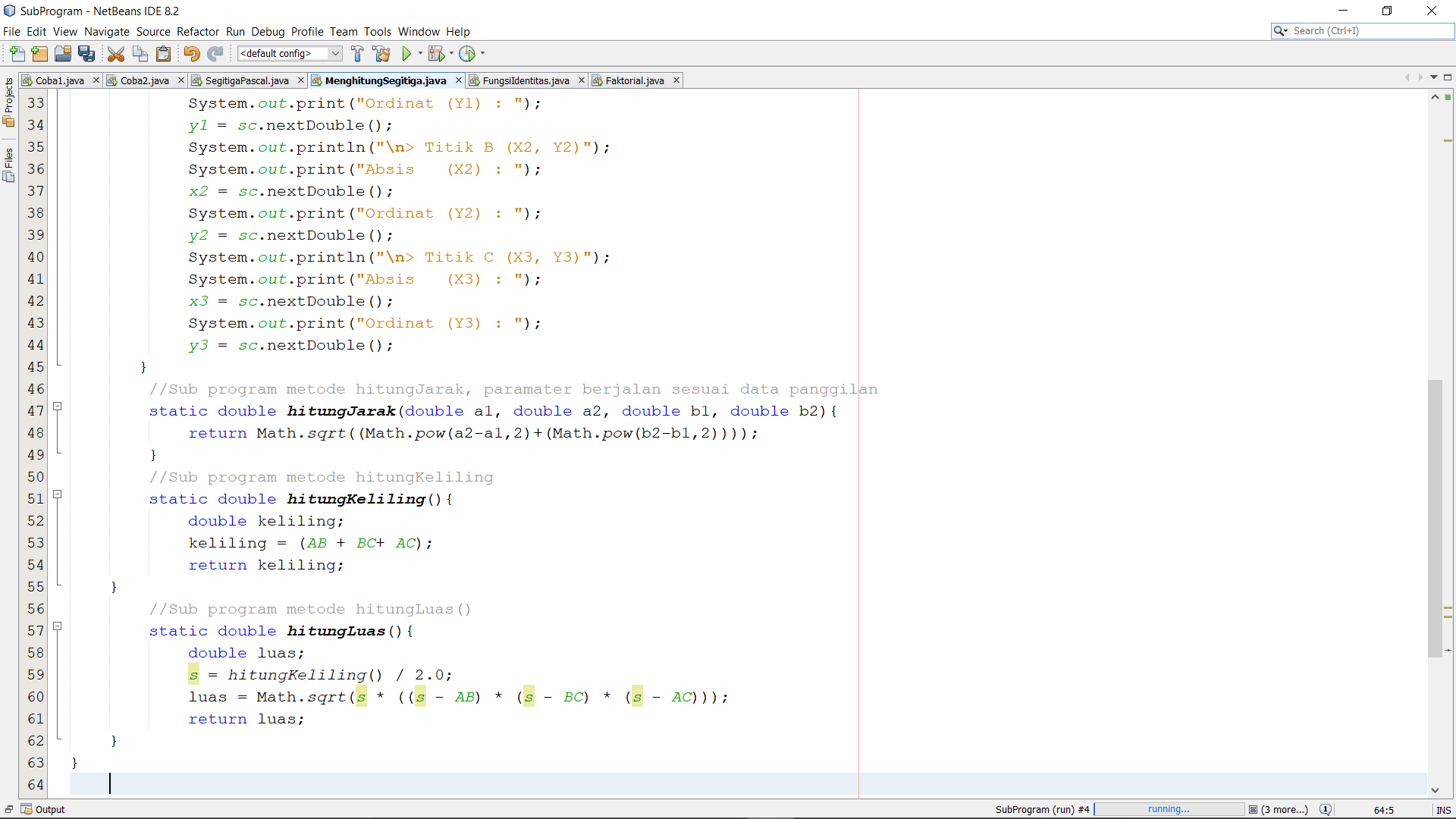
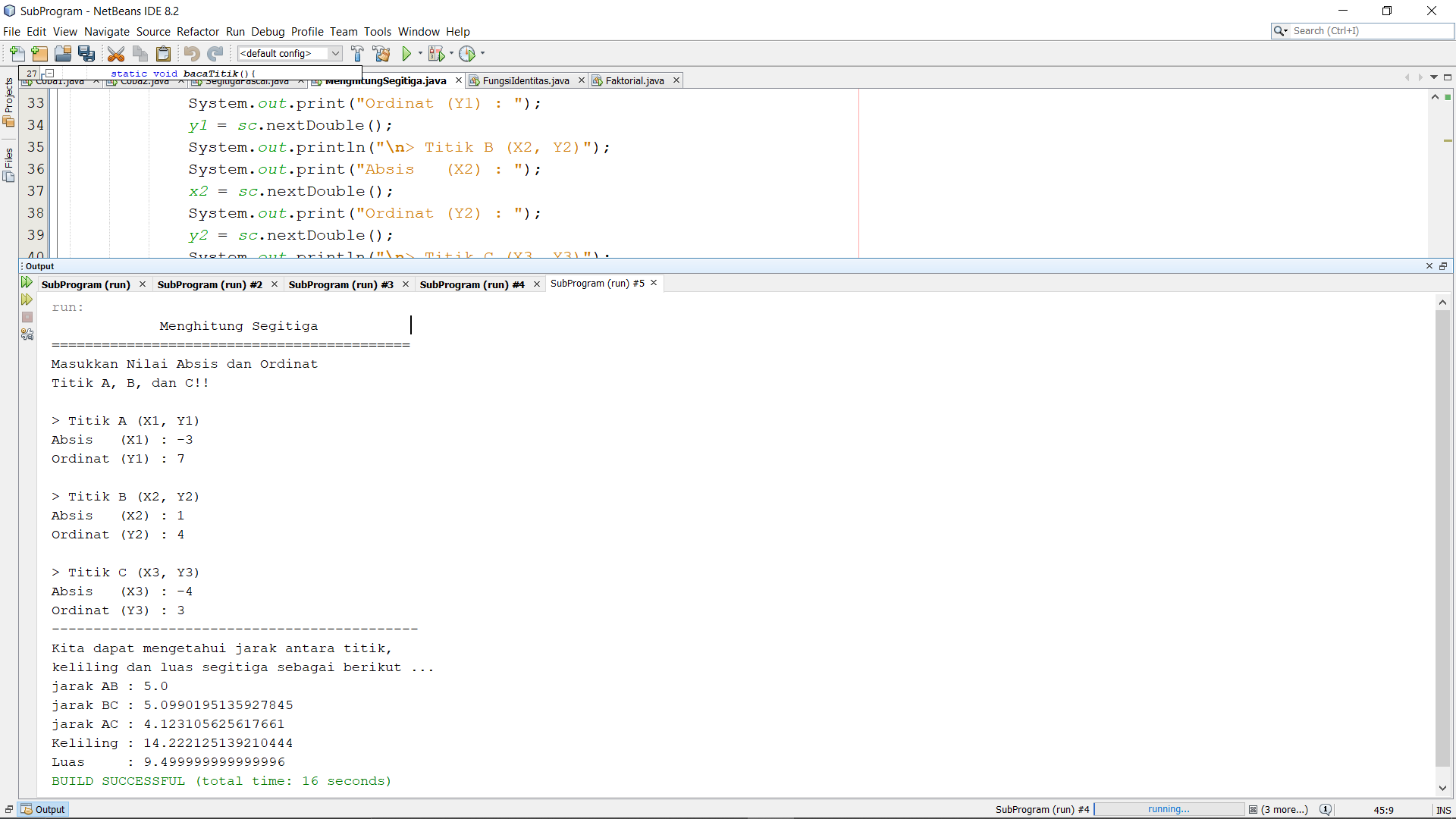
SubProgram hirungKeliling() perintah ini membaca dan menghitung hasil kali variabel SubProgram hitungJarak() serta mengirimnya kembali

Memanggil hitungLuas () dan disimpan ke variabel luas

SubProgram hirungLuas() perintah ini, menghitung isi variabel s dan membaca dan menghitung hasil kali variabel SubProgram hitungJarak() serta mengirimnya kembali

**Input**

Memesan memori untuk bilangan dengan nama variabel x1, y1, x2 , y2, x3, y3, AB, AC, BC, s, a1, a2, b1, b2, keliling, kel dan luas

1. **Capture Program**
2. **Capture Output**
3. **Uraian Soal**

Fungsi densitas/kepadatan normal baku yang berbentuk lonceng atau bel dapat dirumuskan memakai fungsi berikut ini.



Di mana e adaah bilangan alam yang besarnya sama dengan 2,71828

Kalau digambar maka fungsi f(z) di atas akan seperti ini

f(Z)

Z

z1

z2

0

1

2

3

4

5

6

-6

-5

-4

-3

-2

-1

Gambar di atas juga bisa disajikan menjadi

f(Z)

Z

z1

z2

0

1

2

3

4

5

6

-6

-5

-4

-3

-2

-1

Tugas Anda adalah membuat grafik/gambar fungsi normal di atas yang disederhananakan menjadi

\*

\*\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

Caranya adalah:

1. Buat metode yag menghitung nilai f(z) tetapi karena f(z) nilai maksimumnya hanya 1 maka buatlah metode tersebut menjadi menghitung nilai dari n f(z) yang memberikan nilai balikan (return) int sebagi hasil pembulatan dari n f(z). Dengan kata lain Anda harus membuat metode seperti berikut

static int normal(int n, double z) {



return (int) Math.round(n \* f) ;

}

1. Buat metode untuk mencetak bintang sebanyak m buah yakni

static void bintang(int m) {

for(int i = 1; i <= m; i++) System.out.print (“\*”);

System.out.println();

}

1. Buat class main yang memuat dua metode di atas serta memanggil metode bintang(m) di mana m = normal(n, z) dan diulang untuk z mulai dari -4 sampai dengan 4 dengan pertambahan nilai 0.5 serta besarnya n dimasukkan lewat keyboard yang sebaiknya bernilai antara 30 sampai dengan 60. Dengan kata lain metode main Anda pada dasarnya berbentuk sbb

baca n // membaca besarnya n (antara 10 – 20)

for (double z = -4; z <= 4; z= z+0.5) {

int jumBintang = normal(n,z);

bintang(jumBintang);

}

1. **Rancangan Program**

**Output**

Grafk Fungsi Identitas/Kepadatan Normal Baku

Masukkan Nilai N : 40

\*

\*

\*

\*

\*

\*\*

\*\*

\*\*\*

\*\*\*

\*\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*\*

\*\*\*

\*\*\*

\*\*

\*\*

\*

\*

\*

\*

\*Distribusi Usia Ke-3 Penduduk di Atas Adalah :

0 - 20 : \*

21 - 40 : \*\*

41 - 60 :

61 - 80 :

81 - 100 :

------------------------------------------------

Pilihan : 5

Terima Kasih

------------------------------------------------

**Output**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*\*

\*\*\*

\*\*\*

\*\*

\*\*

\*

\*

\*

\*

\*

**Proses**

//Awal Do

Do, menampilkan blok pernyataan ...

Membaca data dari keyboard (Long) disimpan ke variable n

If, Jika (n < 20 && n > 100 ) Maka, memberi tahu user (False)

While, apabila (n < 20 && n > 100 ) perintah di blok pernyataan do akan terus diulang (False)

//Akhir Do

//Awal For

For, exspresi (int i = 0; i<=n; i++) //Note : Diulang hingga memenuhi

Memanggil normal(n,z) dan disimpan ke variabel jumBintang

Memanggil bintang(jumBintang)

Sub program normal, n dan z sebagai parameter sesuai pemanggilan

Menyimpan bilangan variabel menghitung isi variabel fz sera memanggil dengan syarat Math.round(n \* fz)

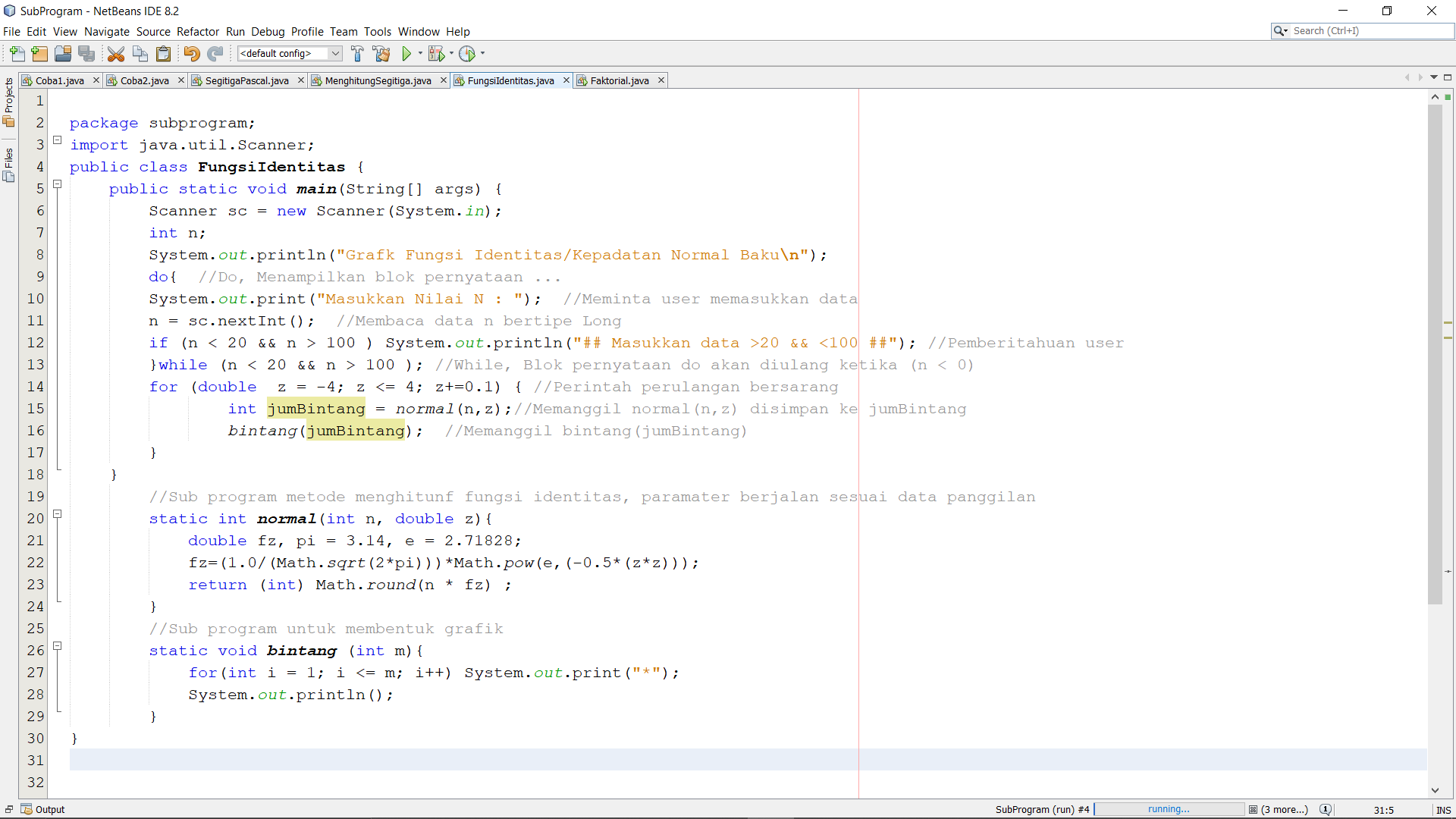
Sub program bintang, m sebagai parameter sesuai pemanggilan

For, exspresi (int i = 1; i <= m; i++) menampilkan \* secara berulang jika memenuhi

Memberi space kosong

**Input**

Memesan memori untuk bilangan dengan nama variabel n, z, jumBintang, fz, pi, e, I, dan m

1. **Capture Program**
2. **Capture Output**