

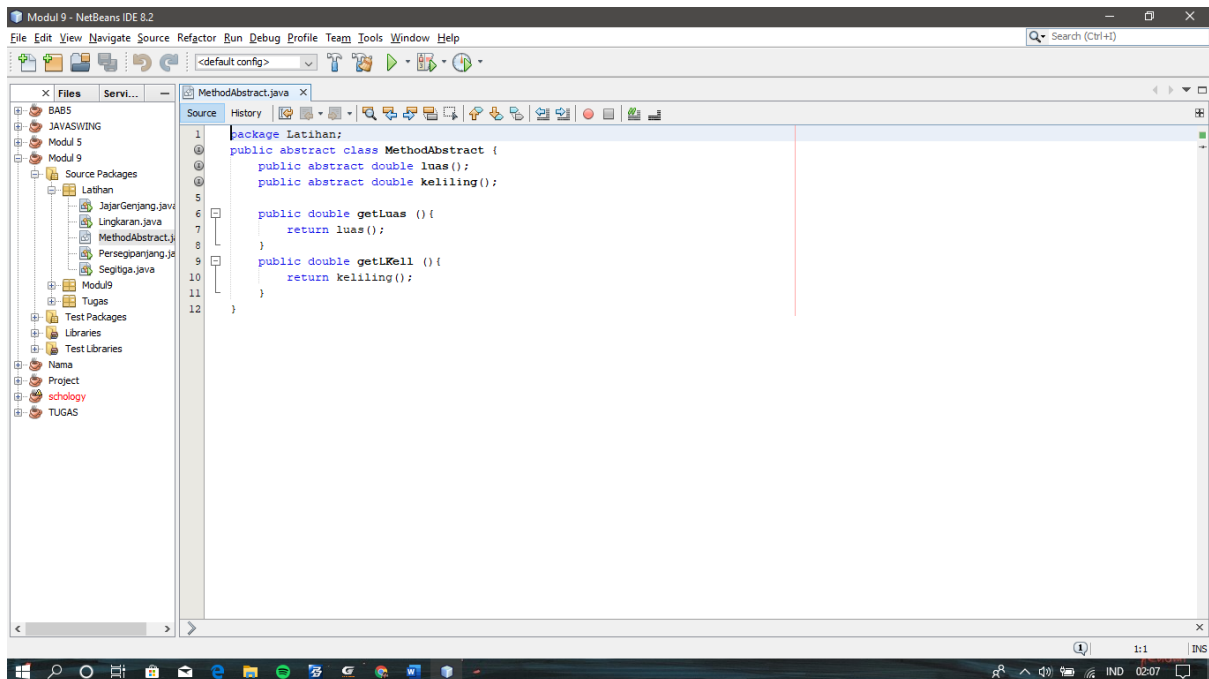
Nama = Chandika Aulia CS

NIM = L200180097

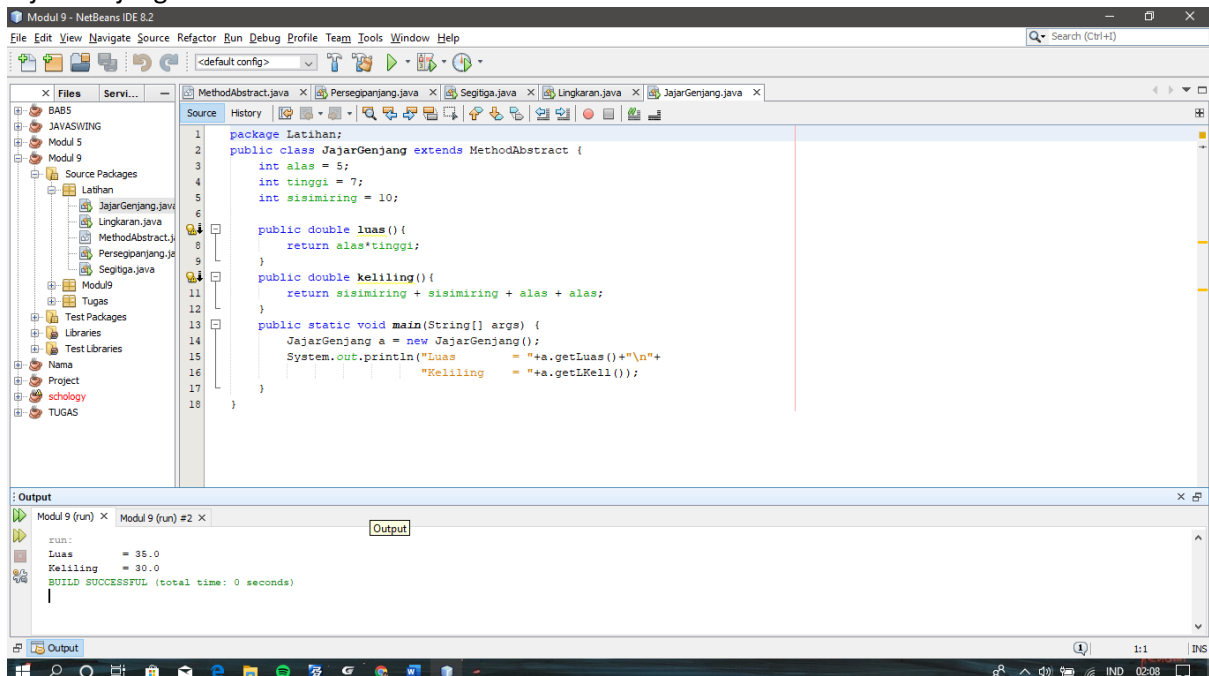
Kelas = C

Latihan.

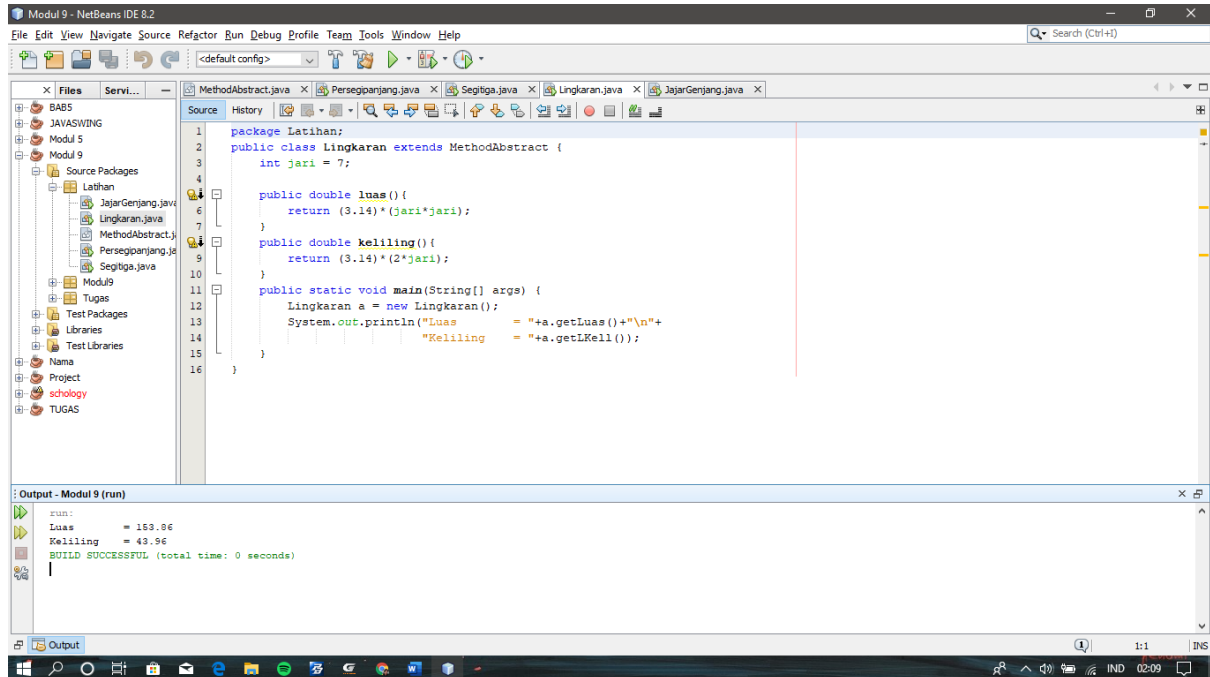
Method Abstract.



Jajar Genjang.



Lingkaran.

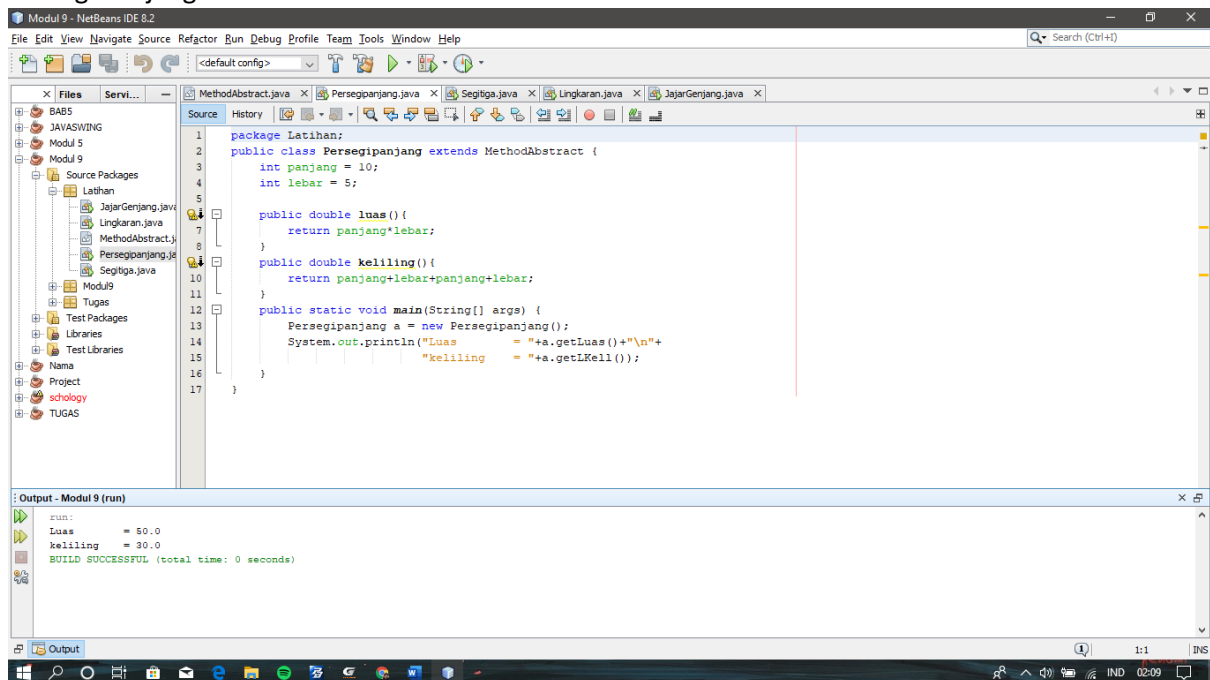


```
1 package Latihan;
2 public class Lingkaran extends MethodAbstract {
3     int jari = 7;
4
5     public double luas() {
6         return (3.14)*(jari*jari);
7     }
8     public double keliling() {
9         return (3.14)*(2*jari);
10    }
11
12    public static void main(String[] args) {
13        Lingkaran a = new Lingkaran();
14        System.out.println("Luas      = "+a.getLuas()+"\n"+
15                           "Keliling  = "+a.getLKell());
16    }
17 }
```

Output - Modul 9 (run)

```
run:
Luas      = 159.86
Keliling  = 43.96
BUILD SUCCESSFUL (total time: 0 seconds)
```

Persegi Panjang.

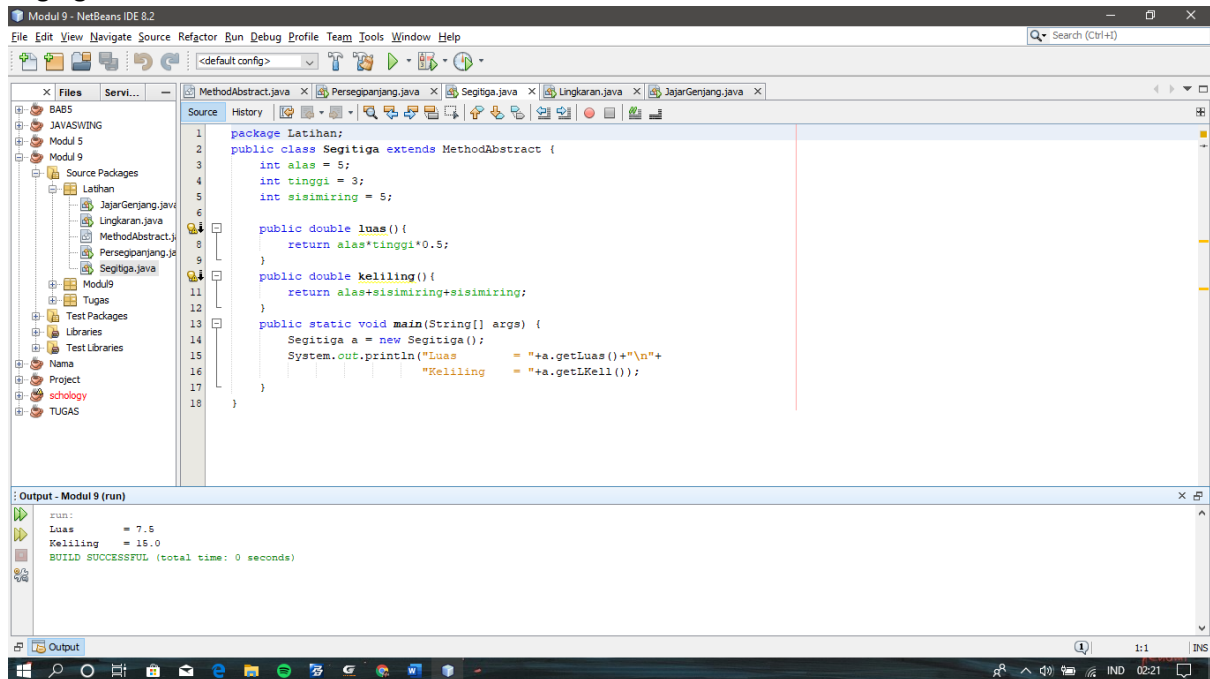


```
1 package Latihan;
2 public class Persegipanjang extends MethodAbstract {
3     int panjang = 10;
4     int lebar = 5;
5
6     public double luas() {
7         return panjang*lebar;
8     }
9     public double keliling() {
10        return panjang+lebar+panjang+lebar;
11    }
12
13    public static void main(String[] args) {
14        Persegipanjang a = new Persegipanjang();
15        System.out.println("Luas      = "+a.getLuas()+"\n"+
16                           "keliling  = "+a.getLKell());
17    }
18 }
```

Output - Modul 9 (run)

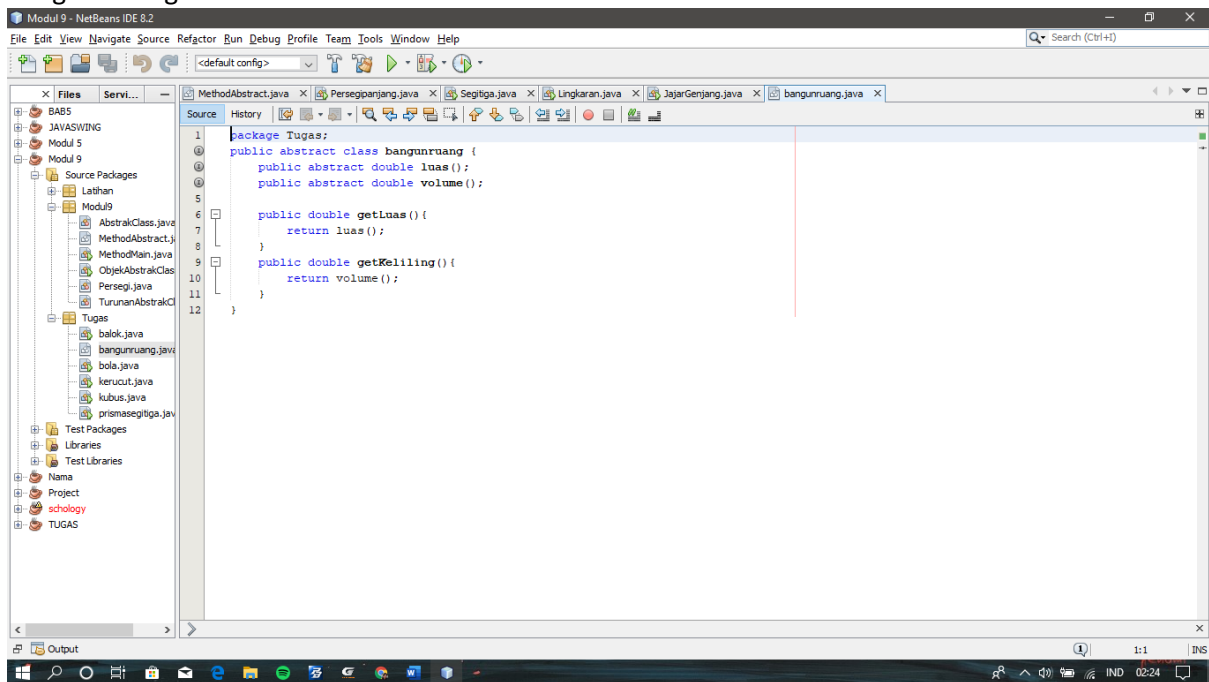
```
run:
Luas      = 50.0
keliling  = 30.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

Segitiga.

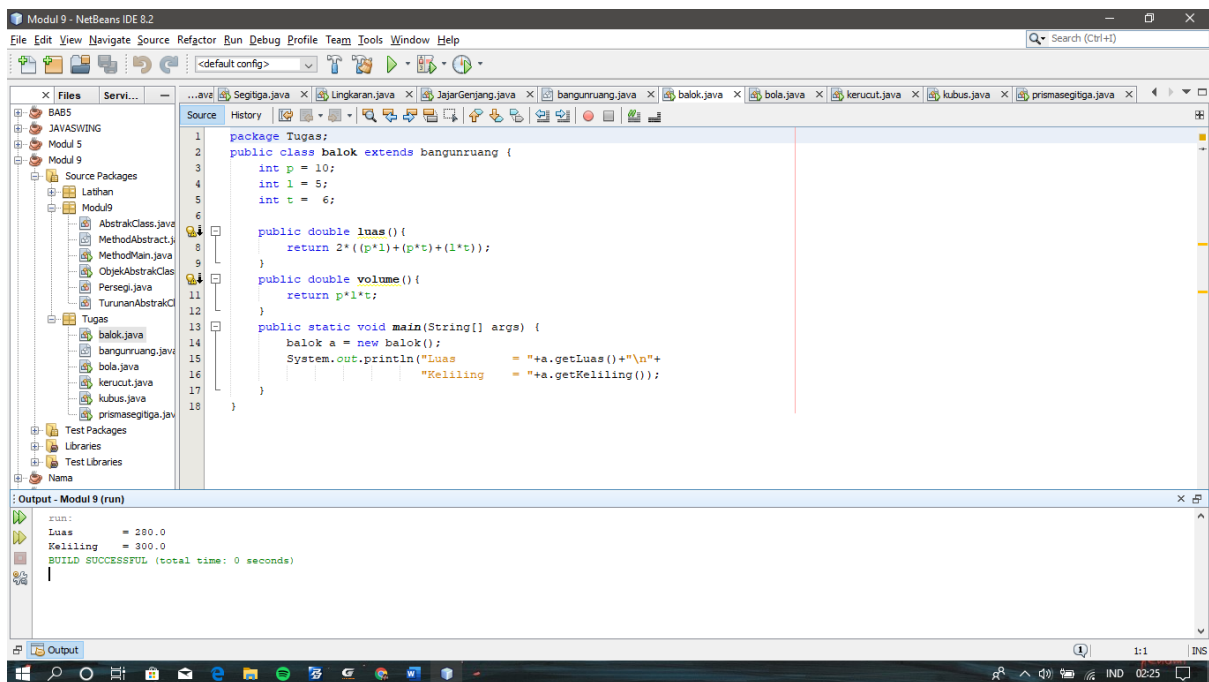


Tugas.

Bangun Ruang.



Balok.



Bola.

The screenshot shows the NetBeans IDE 8.2 interface. The 'Files' view on the left displays a project structure with 'Modul 9' containing 'Source Packages' and 'Test Packages'. The 'Source' view shows the 'Bola' class, which extends 'BangunRuang'. The code defines a constant `int r = 7;` and methods `luas()` and `volume()`. The `main` method creates a `Bola` object and prints its `luas` and `keliling` values. The 'Output' view at the bottom shows the results of running the program: `Luas = 615.44` and `Keliling = 343.0`. The status bar indicates 'BUILD SUCCESSFUL (total time: 0 seconds)'.

```
1 package Tugas;
2 public class bola extends bangunruang{
3     int r = 7;
4
5     public double luas() {
6         return 4 * (3.14) * (r*r);
7     }
8     public double volume() {
9         return (4/3) * (r*r*r);
10    }
11
12    public static void main(String[] args) {
13        bola a = new bola();
14        System.out.println("Luas      = "+a.getLuas()+"\n"+
15                           "Keliling  = "+a.getKeliling());
16    }
17 }
```

run:
Luas = 615.44
Keliling = 343.0
BUILD SUCCESSFUL (total time: 0 seconds)

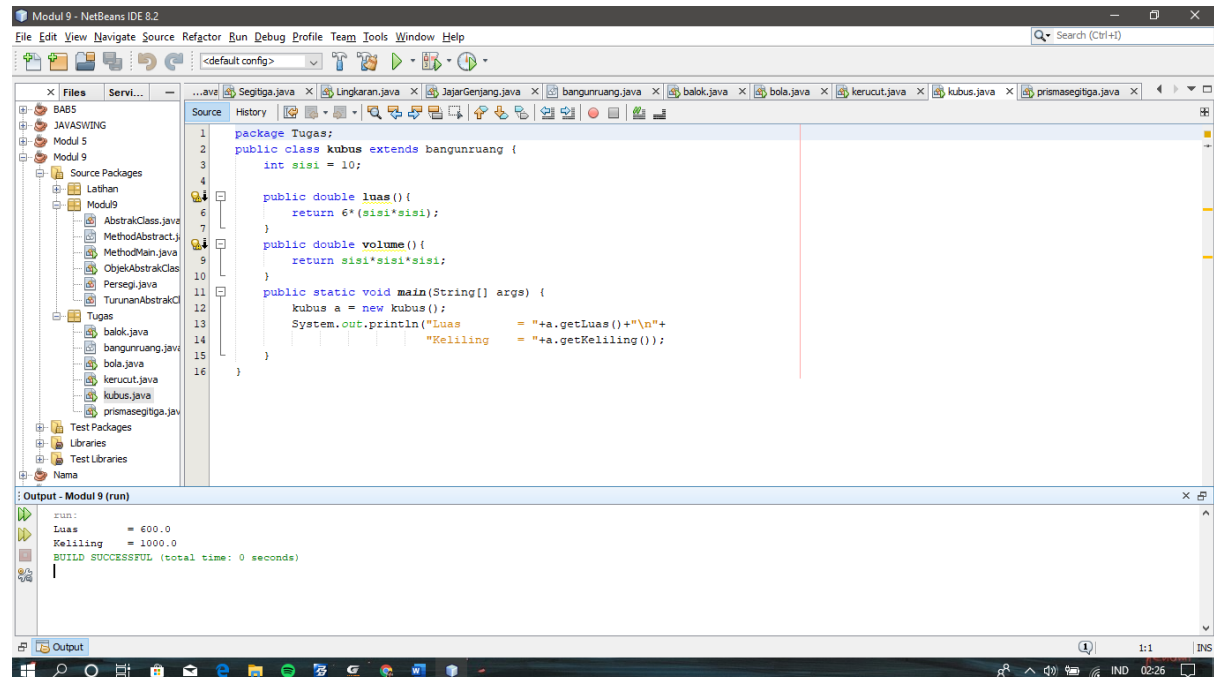
Kerucut.

The screenshot shows the NetBeans IDE 8.2 interface. The 'Files' view on the left displays a project structure with 'Modul 9' containing 'Source Packages' and 'Test Packages'. The 'Source' view shows the 'Kerucut' class, which extends 'BangunRuang'. The code defines constants `int r = 7;`, `int s = 10;`, and `int t = 15;` and methods `luas()` and `volume()`. The `main` method creates a `Kerucut` object and prints its `luas` and `keliling` values. The 'Output' view at the bottom shows the results of running the program: `Luas = 1538.6000000000001` and `Keliling = 769.3000000000001`. The status bar indicates 'BUILD SUCCESSFUL (total time: 0 seconds)'.

```
1 package Tugas;
2 public class kerucut extends bangunruang {
3     int r = 7;
4     int s = 10;
5     int t = 15;
6
7     public double luas() {
8         return (3.14) * r * (r*s);
9     }
10    public double volume() {
11        return ((3.14) * (r*r) * t) / 3;
12    }
13
14    public static void main(String[] args) {
15        kerucut a = new kerucut();
16        System.out.println("Luas      = "+a.getLuas()+"\n"+
17                           "Keliling  = "+a.getKeliling());
18    }
19 }
```

run:
Luas = 1538.6000000000001
Keliling = 769.3000000000001
BUILD SUCCESSFUL (total time: 0 seconds)

Kubus.



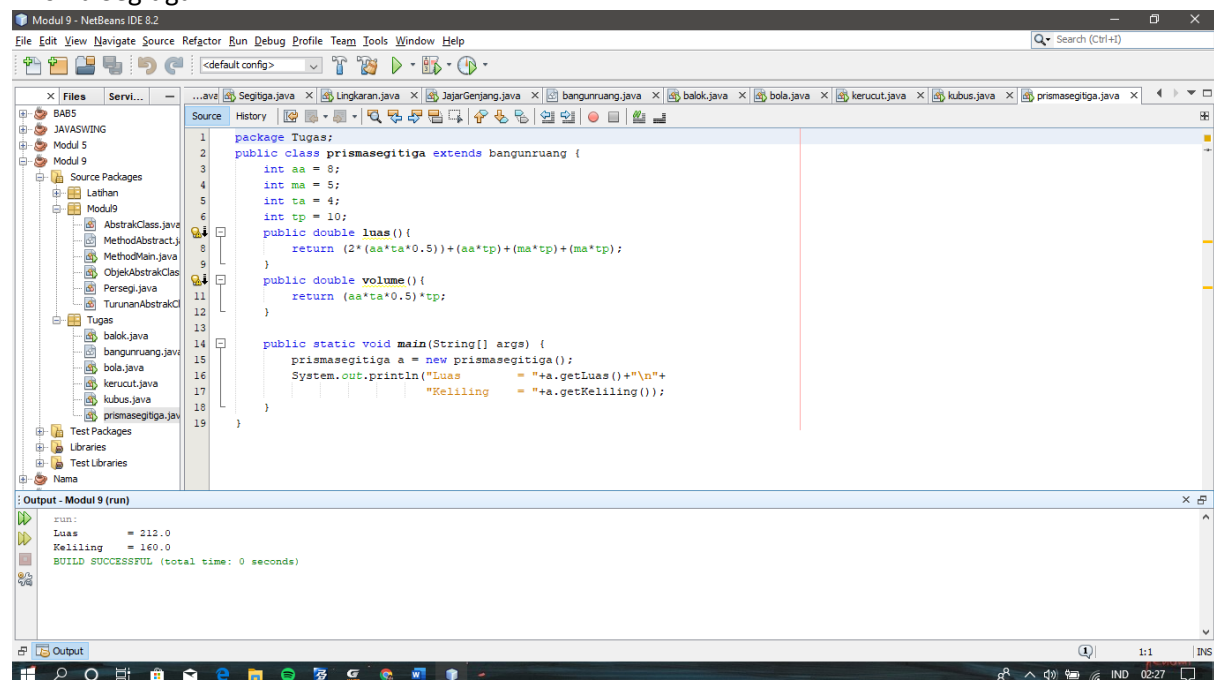
The screenshot shows the NetBeans IDE 8.2 interface. The 'Files' view on the left displays a project structure with 'Modul 9' containing 'Source Packages' and 'Test Packages'. The 'Source' view shows the code for 'kubus.java' in the 'Tugas' package. The code defines a class 'kubus' extending 'bangunruang', with a static variable 'sisi' set to 10. It includes methods 'luas()' and 'volume()' for calculating area and volume, and a 'main' method that creates a 'kubus' object and prints its 'luas' and 'keliling' (perimeter) values. The 'Output' view at the bottom shows the results of running the program: 'luas = 600.0' and 'keliling = 1000.0'. The status bar at the bottom indicates 'BUILD SUCCESSFUL (total time: 0 seconds)'.

```
1 package Tugas;
2 public class kubus extends bangunruang {
3     int sisi = 10;
4
5     public double luas() {
6         return 6*(sisi*sisi);
7     }
8     public double volume() {
9         return sisi*sisi*sisi;
10    }
11    public static void main(String[] args) {
12        kubus a = new kubus();
13        System.out.println("Luas      = "+a.getLuas()+"\n"+
14                           "Keliling   = "+a.getKeliling());
15    }
16 }
```

Output - Modul 9 (run)

```
run:
Luas      = 600.0
Keliling   = 1000.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

Prisma Segitiga.



The screenshot shows the NetBeans IDE 8.2 interface. The 'Files' view on the left displays a project structure with 'Modul 9' containing 'Source Packages' and 'Test Packages'. The 'Source' view shows the code for 'prismasegitiga.java' in the 'Tugas' package. The code defines a class 'prismasegitiga' extending 'bangunruang', with static variables 'aa', 'ma', 'ta', and 'tp' set to 8, 5, 4, and 10 respectively. It includes methods 'luas()' and 'volume()' for calculating area and volume, and a 'main' method that creates a 'prismasegitiga' object and prints its 'luas' and 'keliling' (perimeter) values. The 'Output' view at the bottom shows the results of running the program: 'luas = 212.0' and 'keliling = 160.0'. The status bar at the bottom indicates 'BUILD SUCCESSFUL (total time: 0 seconds)'.

```
1 package Tugas;
2 public class prismasegitiga extends bangunruang {
3     int aa = 8;
4     int ma = 5;
5     int ta = 4;
6     int tp = 10;
7     public double luas() {
8         return (2*(aa*ta*0.5))+(aa*tp)+(ma*tp)+(ma*tp);
9     }
10    public double volume() {
11        return (aa*ta*0.5)*tp;
12    }
13
14    public static void main(String[] args) {
15        prismasegitiga a = new prismasegitiga();
16        System.out.println("Luas      = "+a.getLuas()+"\n"+
17                           "Keliling   = "+a.getKeliling());
18    }
19 }
```

Output - Modul 9 (run)

```
run:
Luas      = 212.0
Keliling   = 160.0
BUILD SUCCESSFUL (total time: 0 seconds)
```