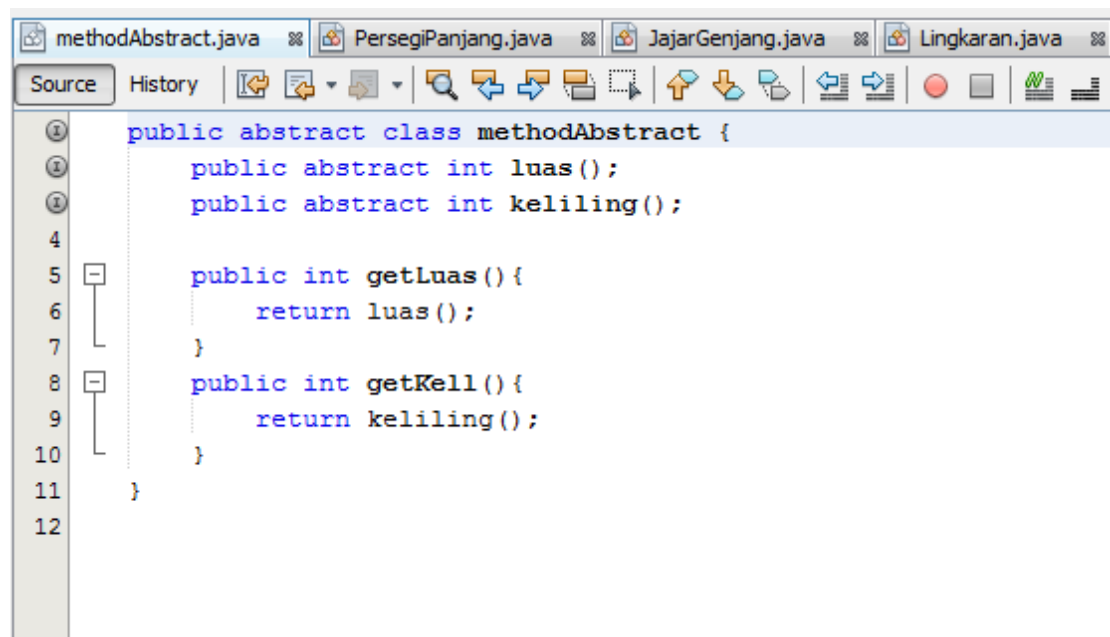
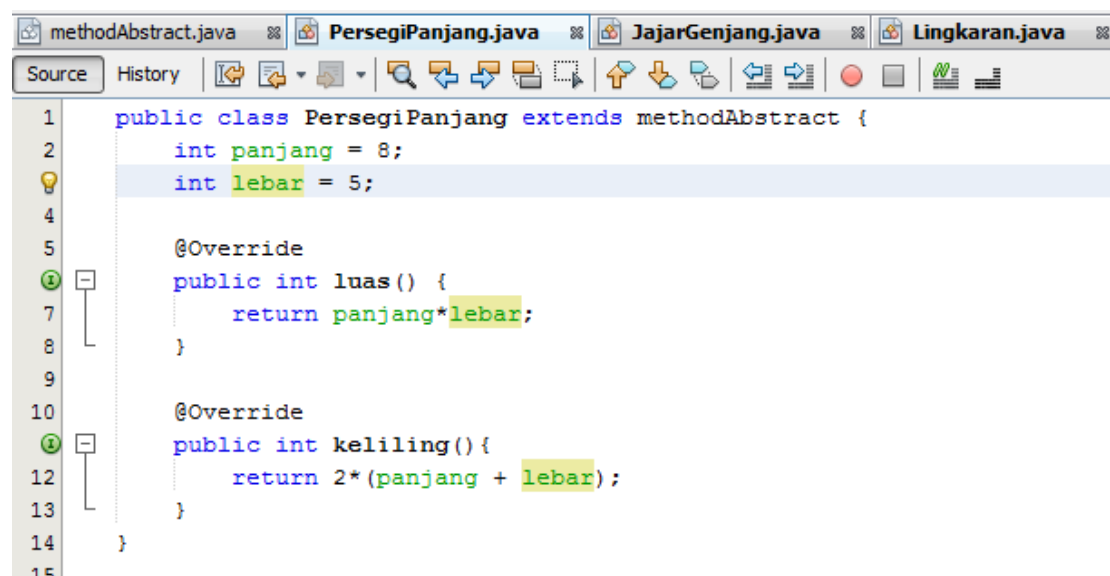


Nama : Ilham Anwar Yazied
NIM : L200180049
Kelas : B
Modul : Modul 9 (Abstract Class)

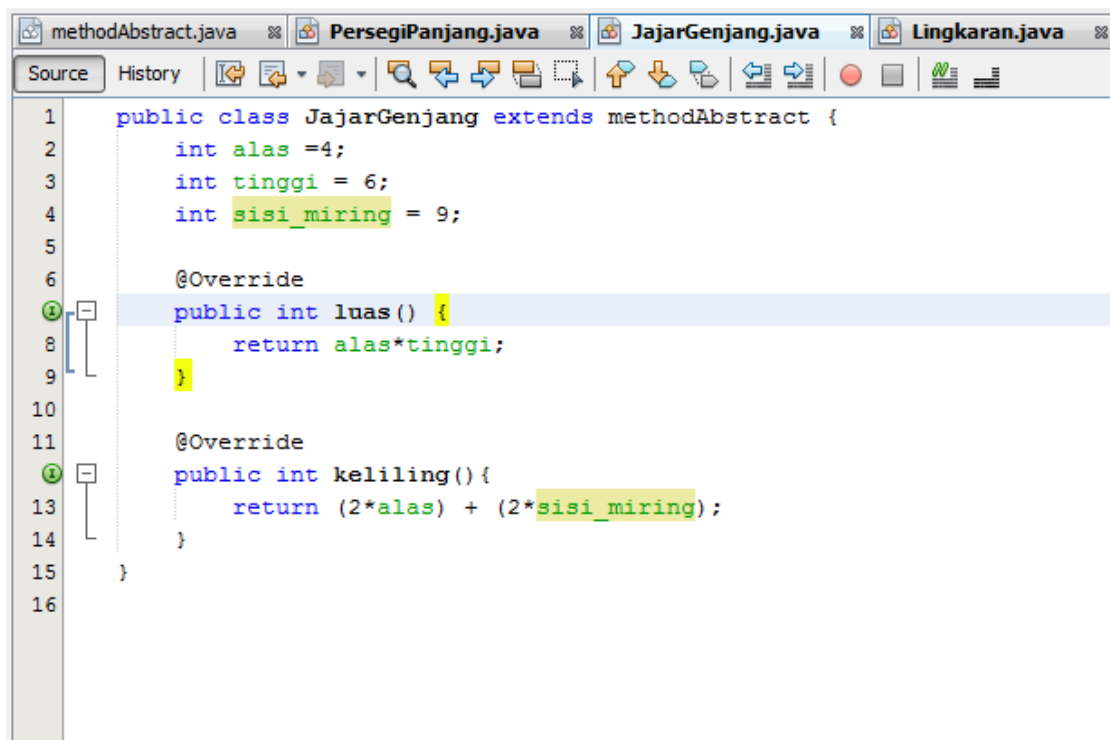
1. Latihan :



```
1 public abstract class methodAbstract {  
2     public abstract int luas();  
3     public abstract int keliling();  
4  
5     public int getLuas() {  
6         return luas();  
7     }  
8     public int getKell() {  
9         return keliling();  
10    }  
11 }  
12
```



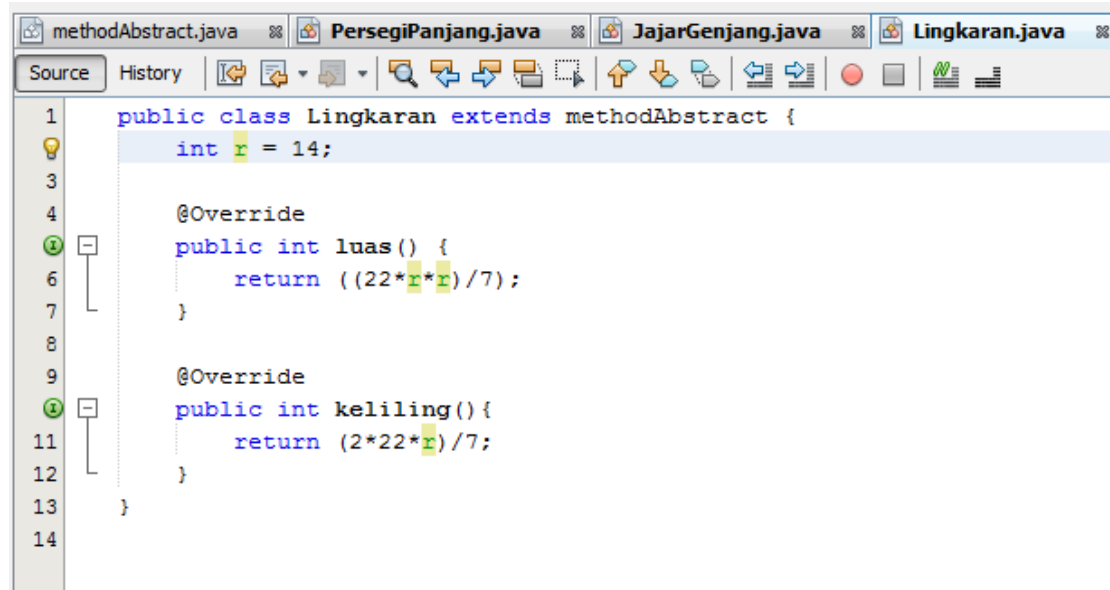
```
1 public class PersegiPanjang extends methodAbstract {  
2     int panjang = 8;  
3     int lebar = 5;  
4  
5     @Override  
6     public int luas() {  
7         return panjang * lebar;  
8     }  
9  
10    @Override  
11    public int keliling() {  
12        return 2 * (panjang + lebar);  
13    }  
14 }  
15
```



The screenshot shows an IDE with four tabs: `methodAbstract.java`, `PersegiPanjang.java`, `JajarGenjang.java` (active), and `Lingkaran.java`. The `JajarGenjang.java` tab displays the following code:

```
1 public class JajarGenjang extends methodAbstract {
2     int alas =4;
3     int tinggi = 6;
4     int sisi_miring = 9;
5
6     @Override
7     public int luas() {
8         return alas*tinggi;
9     }
10
11    @Override
12    public int keliling(){
13        return (2*alas) + (2*sisi_miring);
14    }
15 }
16
```

The code defines a `JajarGenjang` class that inherits from `methodAbstract`. It contains two instance variables: `alas` (base) and `sisi_miring` (slanted side). The `luas()` method calculates the area by multiplying `alas` and `tinggi`. The `keliling()` method calculates the perimeter by summing twice the `alas` and twice the `sisi_miring`. The `sisi_miring` variable is highlighted in yellow in the original image.

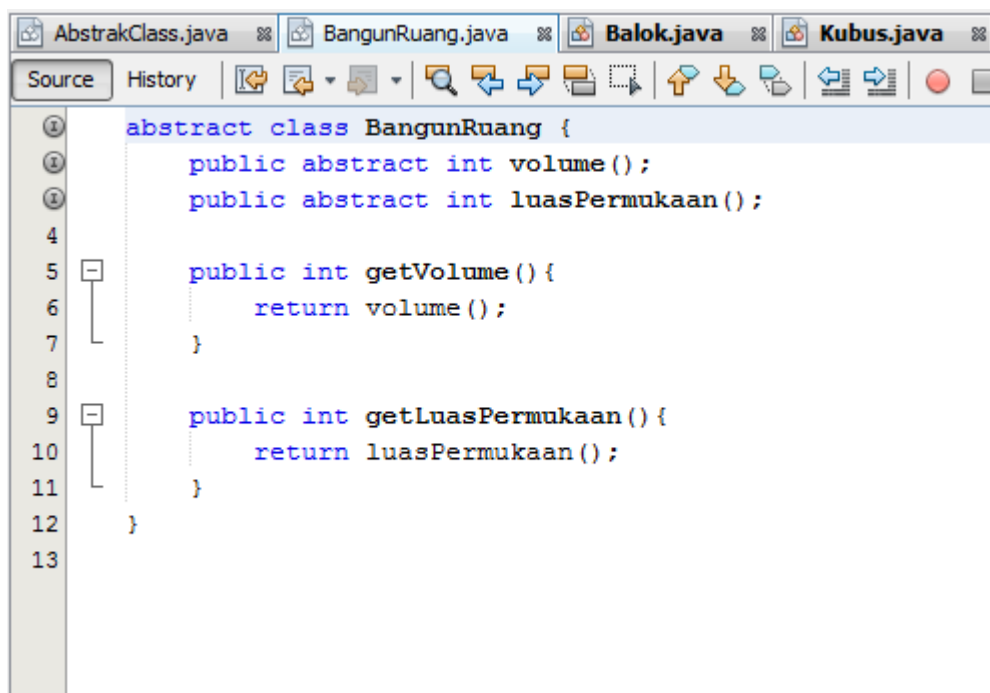


```
1 public class Lingkaran extends methodAbstract {
2     int r = 14;
3
4     @Override
5     public int luas() {
6         return ((22*r*r)/7);
7     }
8
9     @Override
10    public int keliling(){
11        return (2*22*r)/7;
12    }
13 }
14
```

```
methodAbstract.java PersegiPanjang.java JajarGenjang.java Lingkaran.java Segitiga.java
Source History
1 public class Segitiga extends methodAbstract {
2     int alas = 9;
3     int tinggi = 5;
4     int sisiMiring = 7;
5
6     @Override
7     public int luas() {
8         return (alas*tinggi)/2;
9     }
10
11    @Override
12    public int keliling(){
13        return alas + tinggi + sisiMiring;
14    }
15 }
16
```

```
methodAbstract.java PersegiPanjang.java JajarGenjang.java Lingkaran.java Segitiga.java MethodMain.java
Source History
1 public class MethodMain {
2     public static void main(String[] args){
3         PersegiPanjang pp = new PersegiPanjang();
4         System.out.println("Latihan : "+"\\n"+"\\n");
5         System.out.println("Rumus Persegi Panjang : ");
6         System.out.println("    1.Keliling = " + pp.getKell());
7         System.out.println("    2.Luas = " + pp.getLuas()+"\\n");
8
9         JajarGenjang jg = new JajarGenjang();
10        System.out.println("Rumus Jajar Genjang : ");
11        System.out.println("    1.Keliling = " + jg.getKell());
12        System.out.println("    2.Luas = " + jg.getLuas()+"\\n");
13
14        Lingkaran L = new Lingkaran();
15        System.out.println("Rumus Lingkaran : ");
16        System.out.println("    1.Keliling = " + L.getKell());
17        System.out.println("    2.Luas = " + L.getLuas()+"\\n");
18
19        Segitiga S = new Segitiga();
20        System.out.println("Rumus Segitiga : ");
21        System.out.println("    1.Keliling = " + S.getKell());
22        System.out.println("    2.Luas = " + S.getLuas()+"\\n");
23    }
24 }
25
```

2. Tugas :



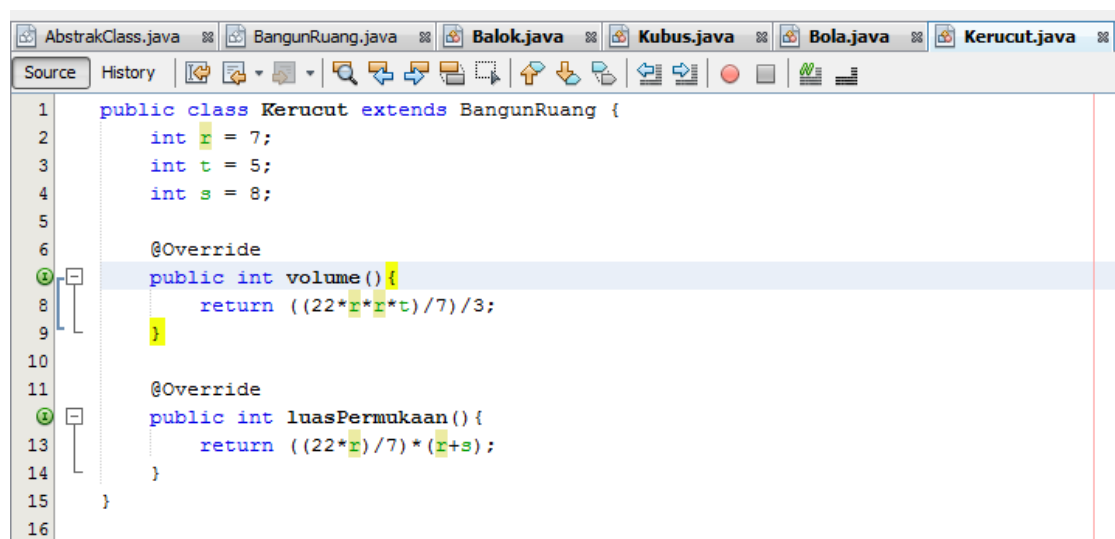
The screenshot shows an IDE window with four tabs: AbstrakClass.java, BangunRuang.java, Balok.java, and Kubus.java. The 'BangunRuang.java' tab is active, displaying the source code of an abstract class. The code defines an abstract class 'BangunRuang' with two abstract methods: 'volume()' and 'luasPermukaan()'. It also includes two concrete methods: 'getVolume()' and 'getLuasPermukaan()', both of which return the values of the abstract methods. The code is formatted with syntax highlighting, and the IDE includes a toolbar with various editing and navigation tools.

```
1 abstract class BangunRuang {  
2     public abstract int volume();  
3     public abstract int luasPermukaan();  
4  
5     public int getVolume() {  
6         return volume();  
7     }  
8  
9     public int getLuasPermukaan() {  
10        return luasPermukaan();  
11    }  
12 }  
13
```

```
1 public class Balok extends BangunRuang {
2     int panjang = 7;
3     int lebar = 5;
4     int tinggi = 10;
5
6     @Override
7     public int volume() {
8         return panjang*lebar*tinggi;
9     }
10
11    @Override
12    public int luasPermukaan() {
13        return 2*((panjang*lebar)+(panjang*tinggi)+(lebar*tinggi));
14    }
15 }
16
```

```
1 public class Kubus extends BangunRuang {
2     int sisi = 9;
3
4     @Override
5     public int volume() {
6         return sisi*sisi*sisi;
7     }
8
9     @Override
10    public int luasPermukaan() {
11        return 6*(sisi*sisi);
12    }
13 }
14
```

```
1 public class Bola extends BangunRuang {
2     int r = 7;
3
4     @Override
5     public int volume() {
6         return ((4*22*r*r*r)/7)/3;
7     }
8
9     @Override
10    public int luasPermukaan() {
11        return (4*22*r*r)/7;
12    }
13 }
14
```



The screenshot shows an IDE window with several tabs: AbstrakClass.java, BangunRuang.java, Balok.java, Kubus.java, Bola.java, and Kerucut.java. The 'Kerucut.java' tab is active, displaying the following Java code:

```
1 public class Kerucut extends BangunRuang {
2     int r = 7;
3     int t = 5;
4     int s = 8;
5
6     @Override
7     public int volume() {
8         return ((22 * r * t) / 7) / 3;
9     }
10
11     @Override
12     public int luasPermukaan() {
13         return ((22 * r) / 7) * (r + s);
14     }
15 }
16
```

The code defines a `Kerucut` class that extends `BangunRuang`. It includes two methods: `volume()` and `luasPermukaan()`, both annotated with `@Override`. The `volume()` method calculates the volume using the formula $\frac{22 \times r \times t}{7} \div 3$, and the `luasPermukaan()` method calculates the surface area using the formula $\frac{22 \times r}{7} \times (r + s)$. The variables `r`, `t`, and `s` are initialized with values 7, 5, and 8, respectively.


```
AbstrakClass.java  BangunRuang.java  Balok.java  Kubus.java  Bola.java  Kerucut.java  PrismaSegitiga.java
Source  History  [Icons]

1  public class PrismaSegitiga extends BangunRuang {
2      int luasAlas = 64;
3      int kelilingAlas = 32;
4      int tinggi = 10;
5
6      @Override
7      public int volume() {
8          return luasAlas*tinggi;
9      }
10
11     @Override
12     public int luasPermukaan() {
13         return (2*luasAlas)+(kelilingAlas*tinggi);
14     }
15 }
16
```

```
Balok.java  Kubus.java  Bola.java  Kerucut.java  PrismaSegitiga.java  BangunRuangMain.java
Source  History  [Icons]

1  public class BangunRuangMain {
2      public static void main (String[] args){
3          Balok bal = new Balok();
4          System.out.println("Tugas : "+"\\n"+"\\n");
5          System.out.println("Rumus Balok : ");
6          System.out.println("    1. Keliling = "+bal.getVolume());
7          System.out.println("    2. Luas = "+bal.getLuasPermukaan()+"\\n");
8
9          Kubus kub = new Kubus();
10         System.out.println("Rumus Kubus : ");
11         System.out.println("    1. Keliling = "+kub.getVolume());
12         System.out.println("    2. Luas = "+kub.getLuasPermukaan()+"\\n");
13
14         Bola bol = new Bola();
15         System.out.println("Rumus Bola : ");
16         System.out.println("    1. Keliling = "+bol.getVolume());
17         System.out.println("    2. Luas = "+bol.getLuasPermukaan()+"\\n");
18
19         Kerucut ker = new Kerucut();
20         System.out.println("Rumus Kerucut : ");
21         System.out.println("    1. Keliling = "+ker.getVolume());
22         System.out.println("    2. Luas = "+ker.getLuasPermukaan()+"\\n");
23
24         PrismaSegitiga prseg = new PrismaSegitiga();
25         System.out.println("Rumus Balok : ");
26         System.out.println("    1. Keliling = "+prseg.getVolume());
27         System.out.println("    2. Luas = "+prseg.getLuasPermukaan());
28     }
29 }
```

