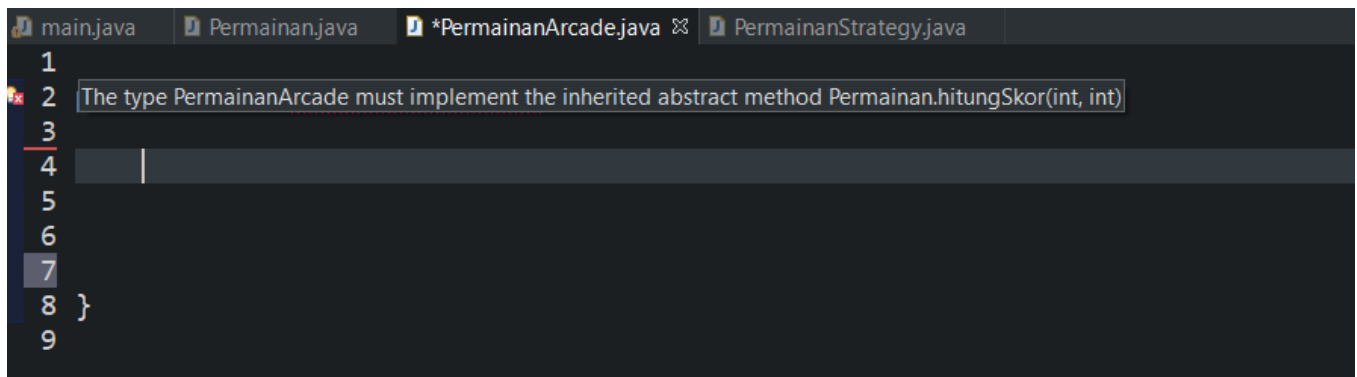
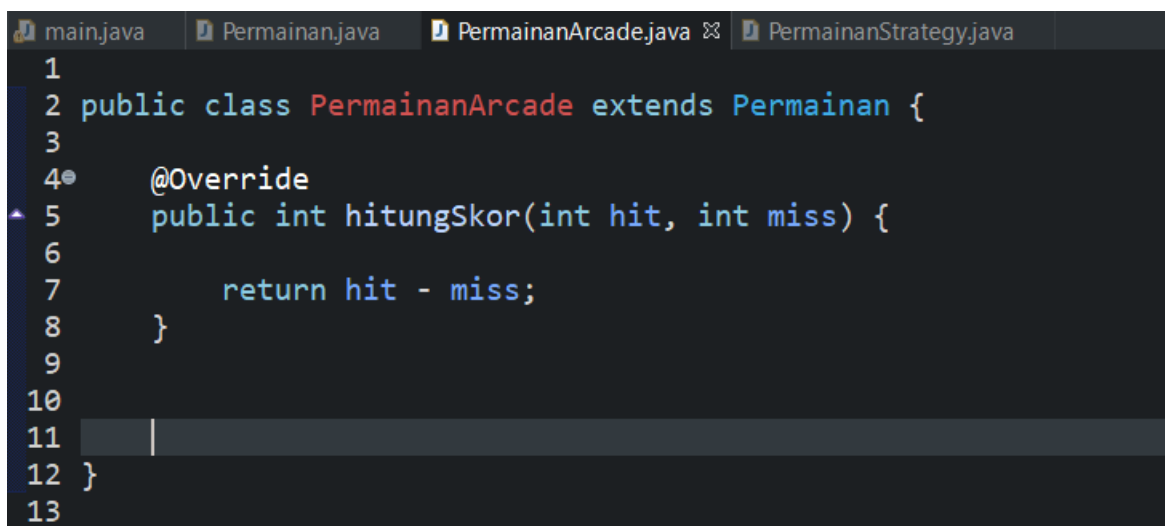


c. Pembuktian bahwa abstract method memastikan method di over-ride oleh Sub-Class



```
main.java  Permainan.java  *PermainanArcade.java  PermainanStrategy.java
1
2 The type PermainanArcade must implement the inherited abstract method Permainan.hitungSkor(int, int)
3
4
5
6
7
8 }
9
```

Terlihat dalam screen shot diatas, jika Sub-class belum terdapat method yang ada pada Super-class atau abstract class (Class Permainan) maka akan terdapat error. Namun, jika sudah terdapat method yang telah diinisialisasi pada abstract method maka tidak akan ada error seperti yang ada pada code berikut.



```
main.java  Permainan.java  PermainanArcade.java  PermainanStrategy.java
1
2 public class PermainanArcade extends Permainan {
3
4     @Override
5     public int hitungSkor(int hit, int miss) {
6
7         return hit - miss;
8     }
9
10
11
12 }
13
```

d. Pembuktian bahwa object dari abstract class tidak dapat dibentuk

```
main.java  Permainan.java  PermainanArcade.java  PermainanStrategy.java
5
6     System.out.println("Permainan Arcade");
7
8     PermainanArcade pa = new PermainanArcade();
9
10    pa.setNamaPemain("Aether");
11    pa.setLevelPemain(100);
12    pa.jalankan();
13    System.out.println("Skor:" + pa.hitungSkor(3, 1));
14
15    System.out.println("Permainan Strategy");
16
17    PermainanStrategy ps = new PermainanStrategy();
18
19    ps.setNamaPemain("Lumine");
20    ps.setLevelPemain(90);
21    ps.jalankan();
22    System.out.println("Skor:" + ps.hitungSkor(5, 0));
23
24
25    //membuat objek dari abstract class
26    Permainan pe = new Permainan();
27
28 }
29
30 }
```

Jika kita membentuk objek dari abstract class (Class Permainan), maka akan dihasilkan error sebagaimana code diatas. Oleh karena itu, object tidak akan bisa dibentuk oleh Abstract Class.

Nama: Ariq Fachry Ramadhan

NIM: 19051397047

Prodi: D4 Manajemen Informatika 19B