

Nama : Asih Putri Retno Kedhaton

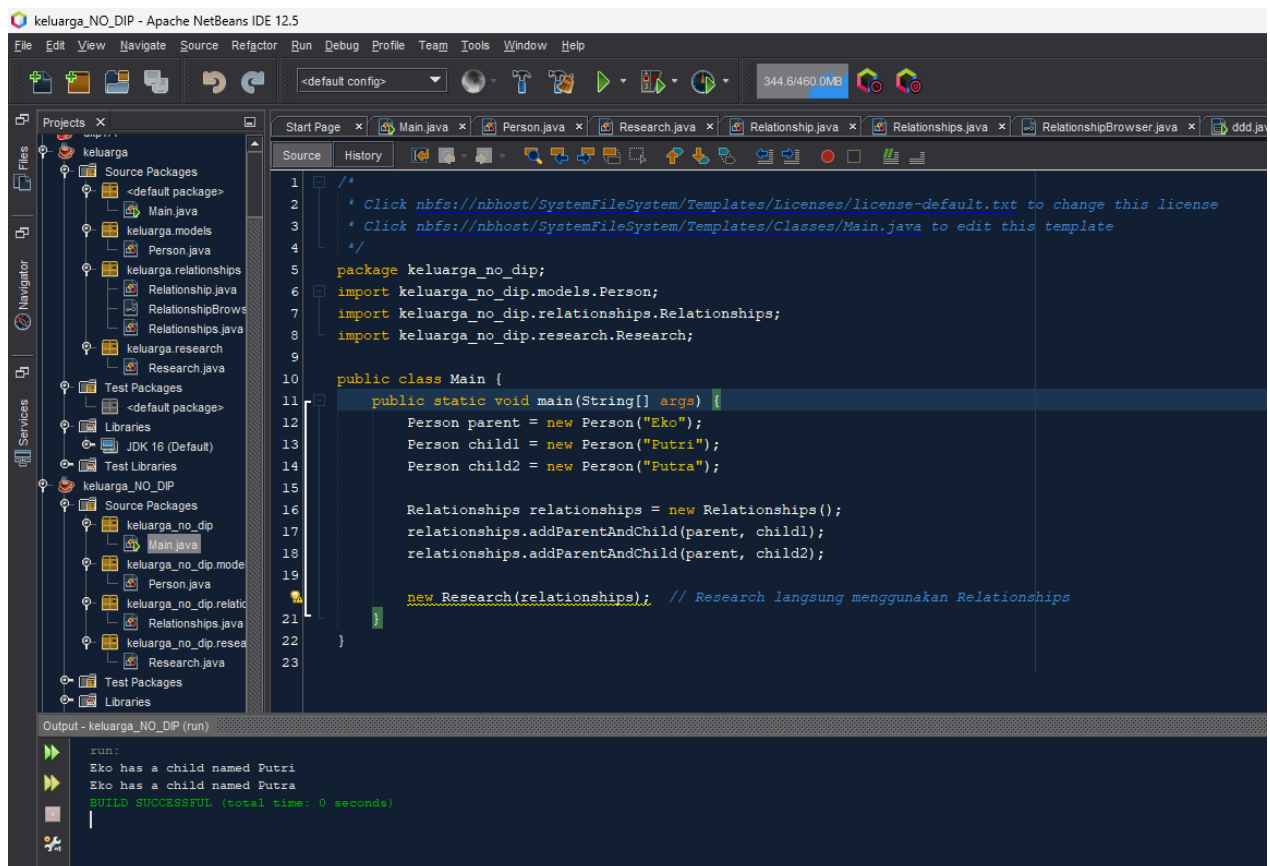
NIM : 21120122130068

Mata Kuliah : Rekayasa Perangkat Lunak Berbasis Komponen

Kelas : C

Running Example SOLID, Dependency Inversion Principle (DIP) pada java mengenai anggota keluarga. Bukti Screenshot:

- Tanpa DIP:





- Dengan DIP

keluarga - Apache NetBeans IDE 12.5

```

1 import keluarga.models.Person;
2 import keluarga.relationships.Relationships;
3 import keluarga.research.Research;
4
5 public class Main {
6     public static void main(String[] args) {
7         Person parent = new Person("Eko");
8         Person child1 = new Person("Putri");
9         Person child2 = new Person("Putra");
10
11         Relationships relationships = new Relationships();
12         relationships.addParentAndChild(parent, child1);
13         relationships.addParentAndChild(parent, child2);
14
15         new Research(relationships); // Research menggunakan interface RelationshipBrowser
16     }
17 }
18

```

Output - keluarga (run)

```

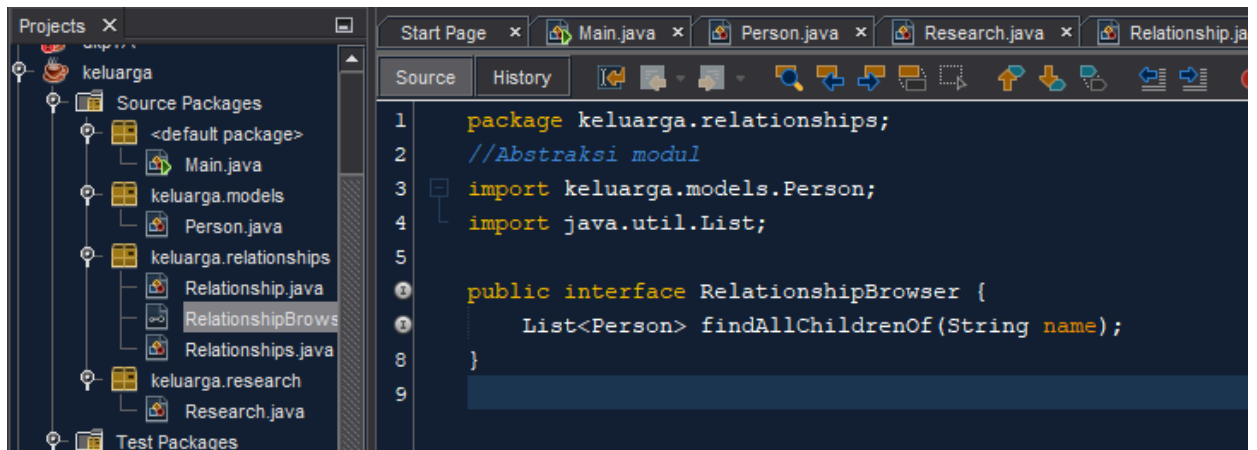
run:
Eko has a child named Putri
Eko has a child named Putra
BUILD SUCCESSFUL (total time: 0 seconds)

```

```

1 package keluarga.relationships;
2
3 import keluarga.models.Person;
4
5 public class Relationship {
6     private Person parent;
7     private Person child;
8     private String relationshipType;
9
10    public Relationship(Person parent, Person child, String relationshipType) {
11        this.parent = parent;
12        this.child = child;
13        this.relationshipType = relationshipType;
14    }
15
16    public Person getParent() {
17        return parent;
18    }
19
20    public Person getChild() {
21        return child;
22    }
23
24    public String getRelationshipType() {
25        return relationshipType;
26    }
27 }
28

```



Kode dengan DIP ditambahkan abstraksi diwujudkan dengan interface RelationshipBrowser yang hanya mendefinisikan bahwa harus ada metode findAllChildrenOf(String name) yang mengembalikan list anak-anak dari orang tua yang namanya diberikan. Namun, interface ini tidak peduli bagaimana detail implementasinya akan dilakukan. class relationship internal juga diberikan implementasi yg berbeda dari Relationships utama untuk memecah kebergantungan tiap implementasinya. Apapun yang diubah pada relationships tidak akan memengaruhi research karena sudah mendapatkan kelas abstrak.