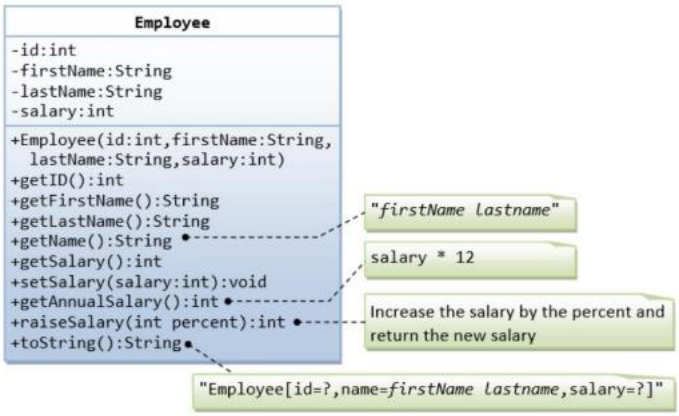
Nama : La Ode Muhammad Gazali
NIM : 222212696
Kelas : 2KS2

TUGAS PRA PERTEMUAN KE-2

PEMROGRAMAN BERBASIK OBJEK

1. Class vs Objek

A. Buatlah Kelas dari Class Diagram Berikut ini dan buatlah kelas untuk testingnya



- **Employee.java**

```
package employee;

/**
 *
 * @author U53R
 */
public class Employee {
    private int id;
    private String firstName;
    private String lastName;
    private int salary;

    public Employee(int id, String fName, String lName, int salary)
    {
        this.id=id;
        this.firstName=fName;
        this.lastName=lName;
        this.salary=salary;
    }

    public int getID()
    {
        return id;
    }

    public String getFirstName()
```

```
    {
        return firstName;
    }

    public String getLastName()
    {
        return lastName;
    }

    public String getName()
    {
        return (firstName + " " + lastName);
    }

    public int getSalary()
    {
        return salary;
    }

    public void setSalary(int salary)
    {
        this.salary = salary;
    }

    public int getAnnualSalary()
    {
        return (this.salary*12);
    }

    public int raiseSalary(int percent){
        this.salary = (int) (this.salary * (1 + percent / 100.0));
        return this.salary;
    }


    public String toString()
    {
        return ("\nEmployee[id="+this.id+",name="+this.firstName+"
"+this.lastName+",salary="+this.salary+"]");
    }
}
```

- **Main_Employee.java**

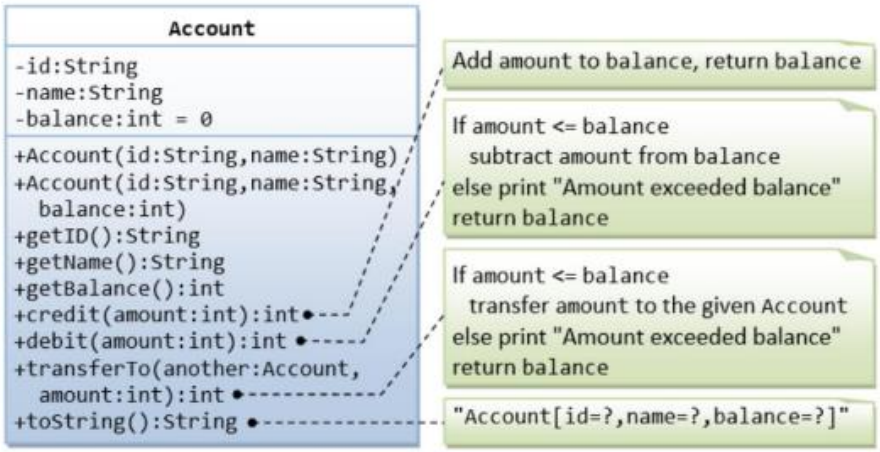
```
package employee;

/**
 *
 * @author U53R
 */
public class Main_Employee {
    public static void main(String[] args)
    {
        Employee employee1 = new Employee(1,"La Ode Muhammad","Gazali",5000);
```

```
        System.out.println("id :"+employee1.getID());
        System.out.println("Name :"+employee1.getName());
        System.out.println("Salary :"+employee1.getSalary());
        System.out.println("Annual Salary :"+employee1.getAnnualSalary());
        System.out.println("Raise salary by 5% :"+employee1.raiseSalary(5));
        System.out.println(employee1.toString());
    }
}
```

- Output

```
run:
id :1
Name :La Ode Muhammad Gazali
Salary :5000
Annual Salary :60000
Raise salary by 5% :5250

Employee[id=1,name=La Ode Muhammad Gazali,salary=5250]
BUILD SUCCESSFUL (total time: 0 seconds)
```

B. Buatlah Kelas dari Class Diagram Berikut ini dan buatlah kelas untuk testingnya



Ketentuan:

- Jumlah amount yang di debit atau di transfer harus maksimal sama dengan balance
- Buatlah dua object account dari kelas Account dan praktekkan metode transferTo() antar objek acoount

- **Account.java**

```
package account;

/**
 *
 * @author U53R
```

```
 */
public class Account {
    private String id;
    private String name;
    private int balance;


    public Account(String id, String name) {
        this.id = id;
        this.name = name;
    }

    public Account(String id, String name, int balance) {
        this.id = id;
        this.name = name;
        this.balance = balance;
    }

    public String getID() {
        return id;
    }

    public String getName() {
        return name;
    }

    public int getBalance() {
        return balance;
    }

    public int credit(int amount) {
        this.balance+=amount;
        return balance;
    }

    public int debit(int amount){
        if(amount <= this.balance)
        {
            this.balance-=amount;
        }
        else
            System.out.println("Amount exceeded balance");

        return balance;
    }

    public int transferTo(Account another, int amount) {
        if(amount <= this.balance) {
            this.debit(amount);
            another.credit(amount);
        }
        else
            System.out.println("Amount exceeded balance");
        return balance;
    }

    public String toString() {
```

```
        return ("Account[id="+id+",name="+name+",balance="+balance+"]");
    }
}
```

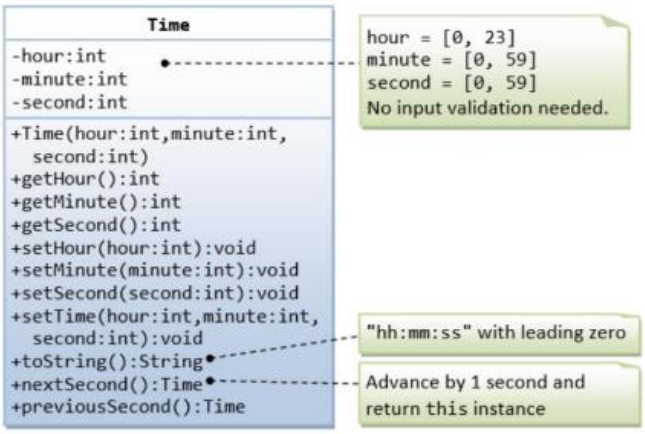
- **Main_Account.java**

```
package account;

/**
 *
 * @author U53R
 */

public class Main_Account {
    public static void main(String[] args)
    {
        Account Gazali = new Account("001","Gazali",5000);
        Account Heri = new Account("002","Heri",1000);
        System.out.println("Informasi Akun Sebelum Transfer");
        System.out.println(Gazali.toString());
        System.out.println(Heri.toString());

        Gazali.transferTo(Heri, 2000);
        System.out.println("\nInformasi Akun Setelah Transfer 2000");
        System.out.println(Gazali.toString());
        System.out.println(Heri.toString());

    }
}
```

- Output

```
run:
Informasi Akun Sebelum Transfer
Account[id=001,name=Gazali,balance=5000]
Account[id=002,name=Heri,balance=1000]

Informasi Akun Setelah Transfer 2000
Account[id=001,name=Gazali,balance=3000]
Account[id=002,name=Heri,balance=3000]
BUILD SUCCESSFUL (total time: 0 seconds)
```

C. Buatlah Kelas dari Class Diagram Berikut ini dan buatlah kelas untuk testingnya



Ketentuan:

- Untuk method nextSecond(), misalnya
  - Jika saat ini waktu menunjukkan pukul 10:00:59 maka next second adalah 10:01:00
  - Jika saat ini waktu menunjukkan pukul 10:59:59 maka next second adalah 11:00:00
  - Jika saat ini waktu menunjukkan pukul 23:59:59 maka next second adalah 00:00:00
- Untuk method previousSecond(), misalnya
  - Jika saat ini waktu menunjukkan pukul 10:01:00 maka previous second adalal 10:00:59
  - Jika saat ini menunjukkan pukul 10:00:00 maka previous second adalah 09:59:59
  - Jika saat ini waktu menunjukkan pukul 00:00:00 maka previous second adalal 23:59:59

- **Time.java**

```
package time;

/**
 *
 * @author U53R
 */
public class Time {
    private int hour;
    private int minute;
    private int second;

    public Time(int hour, int minute, int second){
        this.hour = hour;
        this.minute = minute;
        this.second = second;
    }

    public int getHour(){
        return hour;
    }

    public int getMinute(){
```

```
        return minute;
    }

    public int getSecond(){
        return second;
    }

    public void setHour(int hour){
        this.hour = hour;
    }

    public void setMinute(int minute){
        this.minute = minute;
    }

    public void setSecond(int second){
        this.second = second;
    }

    public void setTime(int hour, int minut, int second){
        this.hour = hour;
        this.minute = minute;
        this.second = second;
    }

    public Time nextSecond() {
        int nextSecondValue = (second + 1) % 60;
        int nextMinuteValue = minute;
        int nextHourValue = hour;

        if (nextSecondValue == 0) {
            nextMinuteValue = (minute + 1) % 60;

            if (nextMinuteValue == 0) {
                nextHourValue = (hour + 1) % 24;
            }
        }

        return new Time(nextHourValue, nextMinuteValue, nextSecondValue);
    }

    public Time previousSecond() {
        int previousSecondValue = second - 1;
        int previousMinuteValue = minute;
        int previousHourValue = hour;

        if (previousSecondValue < 0) {
            previousSecondValue = 59;

            if (minute == 0) {
                previousMinuteValue = 59;

                if (hour == 0) {
                    previousHourValue = 23;
                } else {
                    previousHourValue = hour - 1;
                }
```

```
            } else {
                previousMinuteValue = minute - 1;
            }
        }

        return new Time(previousHourValue, previousMinuteValue,
previousSecondValue);
    }

    public String toString()
    {
        return (this.hour+":"+this.minute+":"+this.second);
    }
}
```

- **Main_Time.java**

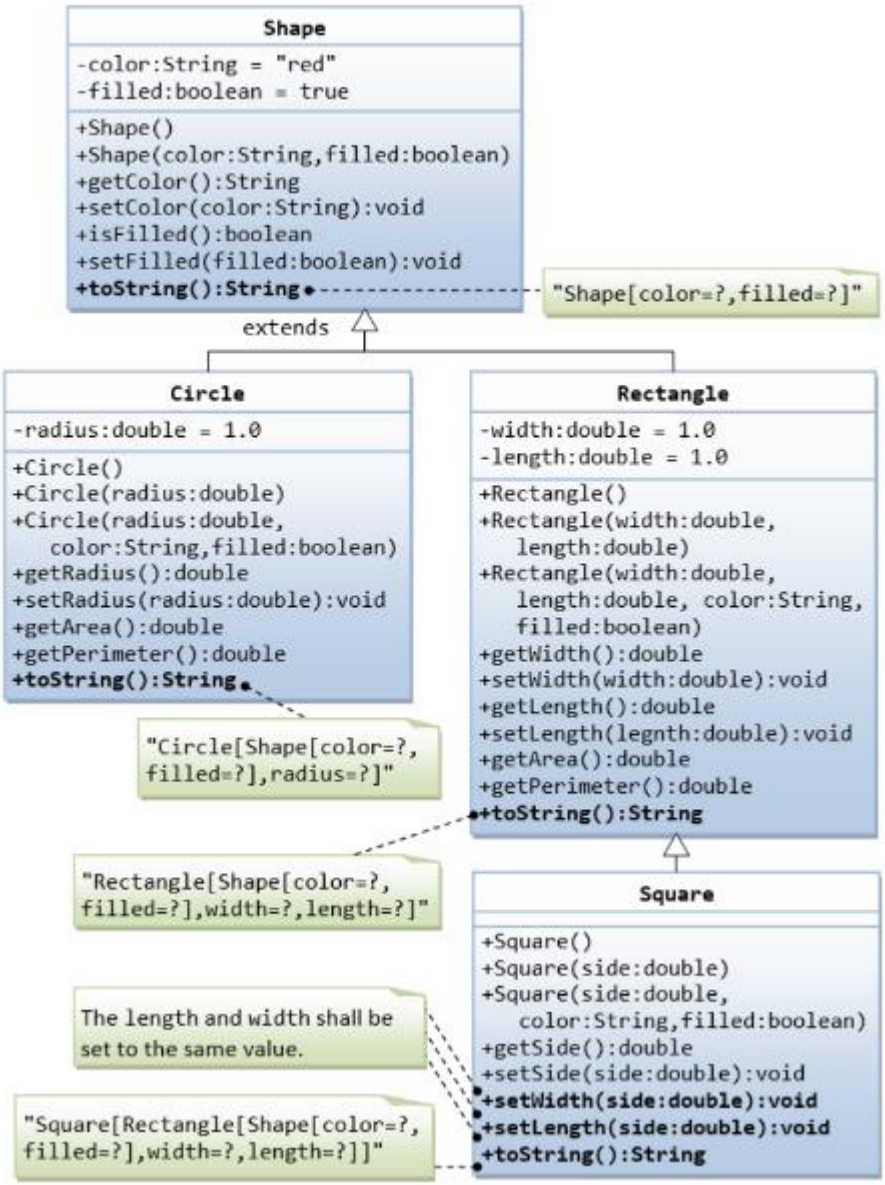
```
package time;

/**
 *
 * @author U53R
 */
public class Main_Time {
    public static void main(String[] args){
       Time currentTime1 = new Time(10, 0, 59);
       System.out.println("Current Time1: " + currentTime1.toString());
       Time nextSecond1 = currentTime1.nextSecond();
       System.out.println("Next Second1: " + nextSecond1.toString());
       Time previousSecond1 = currentTime1.previousSecond();
       System.out.println("Previous Second1: " + previousSecond1.toString());

       Time currentTime2 = new Time(10, 0, 0);
       System.out.println("\nCurrent Time2: " + currentTime2.toString());
       Time nextSecond2 = currentTime2.nextSecond();
       System.out.println("Next Second2: " + nextSecond2.toString());
       Time previousSecond = currentTime2.previousSecond();
       System.out.println("Previous Second2: " + previousSecond.toString());

    }

}
```

- Output

```
run:
Current Time1: 10:0:59
Next Second1: 10:1:0
Previous Second1: 10:0:58

Current Time2: 10:0:0
Next Second2: 10:0:1
Previous Second2: 9:59:59
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. Inheritence

Buatlah Kelas dari Class Diagram Berikut ini dan buatlah kelas untuk testingnya



- **Shape.java**

```
package shape;

/**
 *
 * @author U53R
 */
public class Shape {

    private String color;
    private boolean filled;
```

```
    public Shape(){
        color = "Red";
        filled = true;
    }

    public Shape(String color, boolean filled){
        this.color = color;
        this.filled = filled;
    }

    public String getColor(){
        return color;
    }

    public void setColor(String color){
        this.color = color;
    }

    public boolean isFIlled(){
        return filled;
    }

    public void setFilled(boolean filled){
        this.filled = filled;
    }

    public String toString(){
        return("Shape [color = "+this.color+", filled = "+ this.filled+"]");
    }

}
```

- **Main_Shape.java**

```
package shape;

/**
 *
 * @author U53R
 */
public class Main_Shape {
    public static void main(String[] args){
        Shape s1 = new Shape();
        System.out.println(s1);

        Shape s2 = new Shape("Blue", false);
        System.out.println(s2);
    }
}
```

- Output

```
run:
Shape [color = Red, filled = true]
Shape [color = Blue, filled = false]
BUILD SUCCESSFUL (total time: 0 seconds)
```

- **Circle.java**

```
package shape;

/**
 *
 * @author U53R
 */

public class Circle extends Shape {
    private double radius;

    public Circle() {
        super();
        radius = 1.0;
    }

    public Circle(double radius) {
        super();
        this.radius = radius;
    }

    public double getRadius() {
        return radius;
    }

    public void setRadius(double radius) {
        this.radius = radius;
    }

    public double getArea() {
        return Math.PI*radius*radius;
    }

    public double getPerimeter() {
        return Math.PI*radius*2;
    }

    @Override
    public String toString() {
        return "Circle[: subclass of " + super.toString() + " radius = " +
radius+"]";
    }
}
```

- **Main_Circle.java**

```
package shape;

/**
 *
 * @author U53R
 */

public class Main_Circle {
    public static void main(String[] args) {
```

```
        // TODO code application logic here
        Circle c1 = new Circle();
        System.out.println(c1);
        System.out.println("Area = "+c1.getArea());
        System.out.println("Perimeter = "+c1.getPerimeter());

        Circle c2 = new Circle(7.0);
        c2.setColor("Green");
        c2.setFilled(false);
        System.out.println("\n"+c2);
        System.out.println("Area = "+c2.getArea());
        System.out.println("Perimeter = "+c2.getPerimeter());

    }
}
```

- Output

```
run:
Circle[: subclass of Shape [color = Red, filled = true] radius = 1.0]
Area = 3.141592653589793
Perimeter = 6.283185307179586

Circle[: subclass of Shape [color = Green, filled = false] radius = 7.0]
Area = 153.93804002589985
Perimeter = 43.982297150257104
BUILD SUCCESSFUL (total time: 0 seconds)
```

- **Rectangle.java**

```
package shape;

/**
 *
 * @author U53R
 */

public class Rectangle extends Shape {
    private double width;
    private double length;

    public Rectangle() {
        super();
        width = 1.0;
        length = 1.0;
    }

    public Rectangle(double width, double length) {
        super();
        this.width = width;
        this.length = length;
    }

    public Rectangle(double width, double length, String color, boolean
filled) {
```

```
        super(color,filled);
        this.width = width;
        this.length = length;
    }

    public double getWidth() {
        return width;
    }

    public void setWidth(double width) {
        this.width = width;
    }

    public double getLength() {
        return length;
    }

    public void setLength(double length) {
        this.length = length;
    }

    public double getArea() {
        return length*width;
    }

    public double getPerimeter() {
        return 2*(length+width);
    }

    @Override
    public String toString() {
        return "Rectangle[: subclass of " + super.toString() + " width = " +
width+ ", length = " + length+"]";
    }
}
```

- **Main_Rectangle.java**

```
package shape;

/**
 *
 * @author U53R
 */

public class Main_Rectangle {
    public static void main(String[] args) {
        Rectangle rectangle1 = new Rectangle();
        System.out.println("---Rectangle 1---");
        System.out.println(rectangle1);
        System.out.println("Area: " + rectangle1.getArea());
        System.out.println("Perimeter: " + rectangle1.getPerimeter());

        Rectangle rectangle2 = new Rectangle(5.0, 3.0);
        System.out.println("\n---Rectangle 2---");
        System.out.println(rectangle2);
```

```
        System.out.println("Area: " + rectangle2.getArea());
        System.out.println("Perimeter: " + rectangle2.getPerimeter());

        Rectangle rectangle3 = new Rectangle(4.0, 5.0, "Yellow", false);
        System.out.println("\n---Rectangle 3---");
        System.out.println(rectangle3);
        System.out.println("Area: " + rectangle3.getArea());
        System.out.println("Perimeter: " + rectangle3.getPerimeter());
    }
}
```

- Output

```
run:
---Rectangle 1---
Rectangle[: subclass of Shape [color = Red, filled = true] width = 1.0, length = 1.0]
Area: 1.0
Perimeter: 4.0

---Rectangle 2---
Rectangle[: subclass of Shape [color = Red, filled = true] width = 5.0, length = 3.0]
Area: 15.0
Perimeter: 16.0

---Rectangle 3---
Rectangle[: subclass of Shape [color = Yellow, filled = false] width = 4.0, length = 5.0]
Area: 20.0
Perimeter: 18.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

- **Square.java**

```
package shape;
/**
 *
 * @author U53R
 */

public class Square extends Rectangle {
    public Square() {
        super();
    }

    public Square(double side) {
        super(side, side);
    }

    public Square(double side, String color, boolean filled) {
        super(side, side, color, filled);
    }

    public double getSide() {
        return getWidth();
    }

    public void setSide(double side) {
        setWidth(side);
```

```
        setLength(side);
    }

    public void setWidth(double side) {
        super.setWidth(side);
        super.setLength(side);
    }

    public void setLength(double side) {
        super.setWidth(side);
        super.setLength(side);
    }

    @Override
    public String toString() {
        return "Square [subclass of " + super.toString() + ", side=" +
getSide() + "]";
    }
}
```

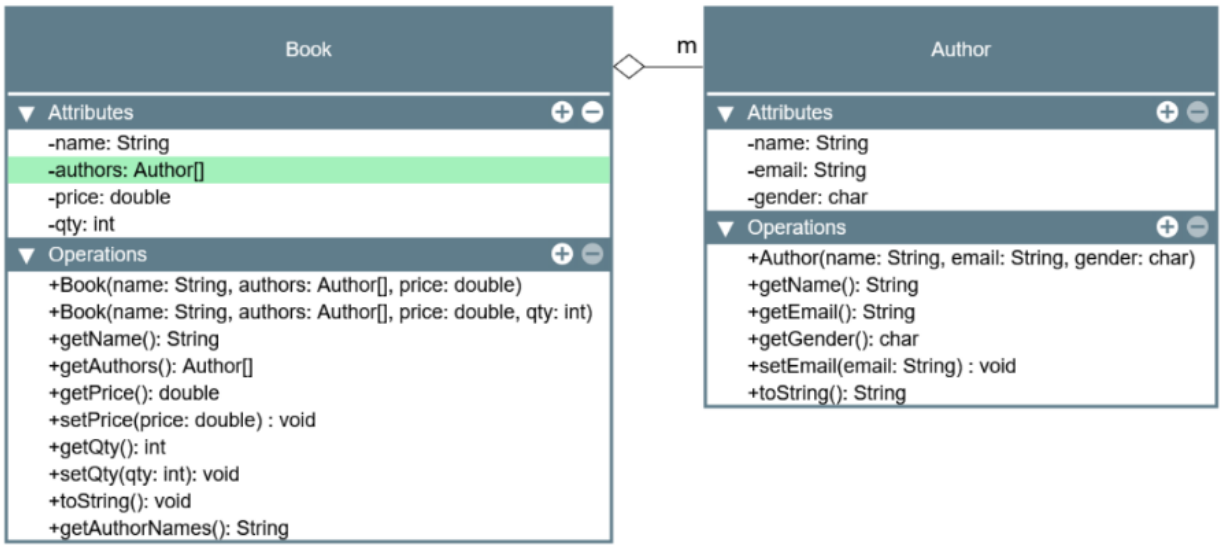
- **Main_Square.java**

```
package shape;
/**
 *
 * @author U53R
 */
public class Main_Square {
    public static void main(String[] args) {
        System.out.println("Before Modification:");
        Square square = new Square(5.0);
        System.out.println(square);
        System.out.println("Area: " + square.getArea());
        System.out.println("Perimeter: " + square.getPerimeter());

        square.setColor("Blue");
        square.setFilled(false);
        System.out.println("\nAfter Modification:");
        System.out.println(square);
        System.out.println("Area: " + square.getArea());
        System.out.println("Perimeter: " + square.getPerimeter());
    }
}
```

- Output

```
run:
Before Modification:
Square [subclass of Rectangle[: subclass of Shape [color = Red, filled = true] width = 5.0, length = 5.0], side=5.0]
Area: 25.0
Perimeter: 20.0

After Modification:
Square [subclass of Rectangle[: subclass of Shape [color = Blue, filled = false] width = 5.0, length = 5.0], side=5.0]
Area: 25.0
Perimeter: 20.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

3. Composition



Ubahlah hubungan Author dan buku sehingga 1 buku ditulis oleh banyak author (one to many) Gunakan array of Author.
Perubahan dari sebelumnya adalah:

1) Variable author menjadi bertipe array of Author
2) Konstruktor Book yang pertama, salah satu parameternya berisi array of Author
3) Konstruktor Book yang kedua, salah satu parameternya berisi array of Author
4) Method getAuthors() akan mengembalikan array of Author
5) Method toString(), pada kelas Book akan berisi semua authornya.
   "Book[name=?,authors={Author[name=?,email=?,gender=?],...},price =?,qty=?]"
6) Method getAuthorNames() akan mengembalikan nama semua author dalam String. Misal akan mengembalikan "Author1, Author2, ... "

- **Author.java**

```
package bookauthor;

/**
 *
 * @author U53R
 */
public class Author{
    private String name;
    private String email;
    private char gender;

    public Author(String name, String email, char gender){
        this.name = name;
        this.email = email;
        this.gender = gender;
    }
```

```
    public String getName(){
        return name;
    }

    public String getEmail(){
        return email;
    }

    public char getGender(){
        return gender;
    }

    public void setEmail(String email){
        this.email = email;
    }

    @Override
    public String toString() {
        return "\n" +
                "        name = " + name + ",\n" +
                "        email = " + email + ",\n" +
                "        gender = " + gender ;
    }

}
```

- **Book.java**

```
package bookauthor;

/**
 *
 * @author U53R
 */

public class Book {
    private String name;
    private Author[] authors;
    private double price;
    private int qty;

    public Book(String name, Author[] authors, double price) {
        this.name = name;
        this.authors = authors;
        this.price = price;
        this.qty = 0;
    }

    public Book(String name, Author[] authors, double price, int qty) {
        this.name = name;
        this.authors = authors;
        this.price = price;
        this.qty = qty;
    }

    public String getName() {
```

```
        return name;
    }

    public Author[] getAuthors() {
        return authors;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public int getQty() {
        return qty;
    }

    public void setQty(int qty) {
        this.qty = qty;
    }

    @Override
    public String toString() {
        StringBuilder authorString = new StringBuilder();
        for (int i = 0; i < authors.length; i++) {
            authorString.append("  ").append(authors[i]);
            if (i < authors.length - 1) {
                authorString.append("\n");
            }
        }

        return "[\n" +
               "  name = " + name + "\n" +
               "  price = " + price + "\n" +
               "  qty = " + qty + "\n" +
               "  authors :" + authorString + "\n" +
               "]";
    }

    public String getAuthorNames() {
        StringBuilder authorNames = new StringBuilder();
        for (int i = 0; i < authors.length; i++) {
            authorNames.append(authors[i].getName());
            if (i < authors.length - 1) {
                authorNames.append(", ");
            }
        }
        return authorNames.toString();
    }
}
```

- **BookAuthor.java**

```
package bookauthor;

/**
 *
 * @author U53R
 */

public class BookAuthor {
    public static void main(String[] args) {
       // Membuat objek Authors
       Author nano = new Author("Nano Yulian P.", "nano@bps.go.id", 'm');
       Author yeni = new Author("Wa Ode Zuhayeni M.", "yeni@bps.go.id", 'f');

       // Membuat array of Authors
       Author[] authors = {nano, yeni};

       // Membuat objek Book dengan menggunakan array of Authors
       Book oopBook = new Book("OOP for dummies", authors, 50000, 100);

       // Mendapatkan informasi buku dan penulis
       System.out.println("Book Info: " + oopBook);
       System.out.println("\nAuthor Names: " + oopBook.getAuthorNames());
    }
}
```

- Output

```
run:
Book Info: [
  name = OOP for dummies
  price = 50000.0
  qty = 100
  authors :
        name = Nano Yulian P.,
        email = nano@bps.go.id,
        gender = m

        name = Wa Ode Zuhayeni M.,
        email = yeni@bps.go.id,
        gender = f
]

Author Names: Nano Yulian P., Wa Ode Zuhayeni M.
BUILD SUCCESSFUL (total time: 0 seconds)
```