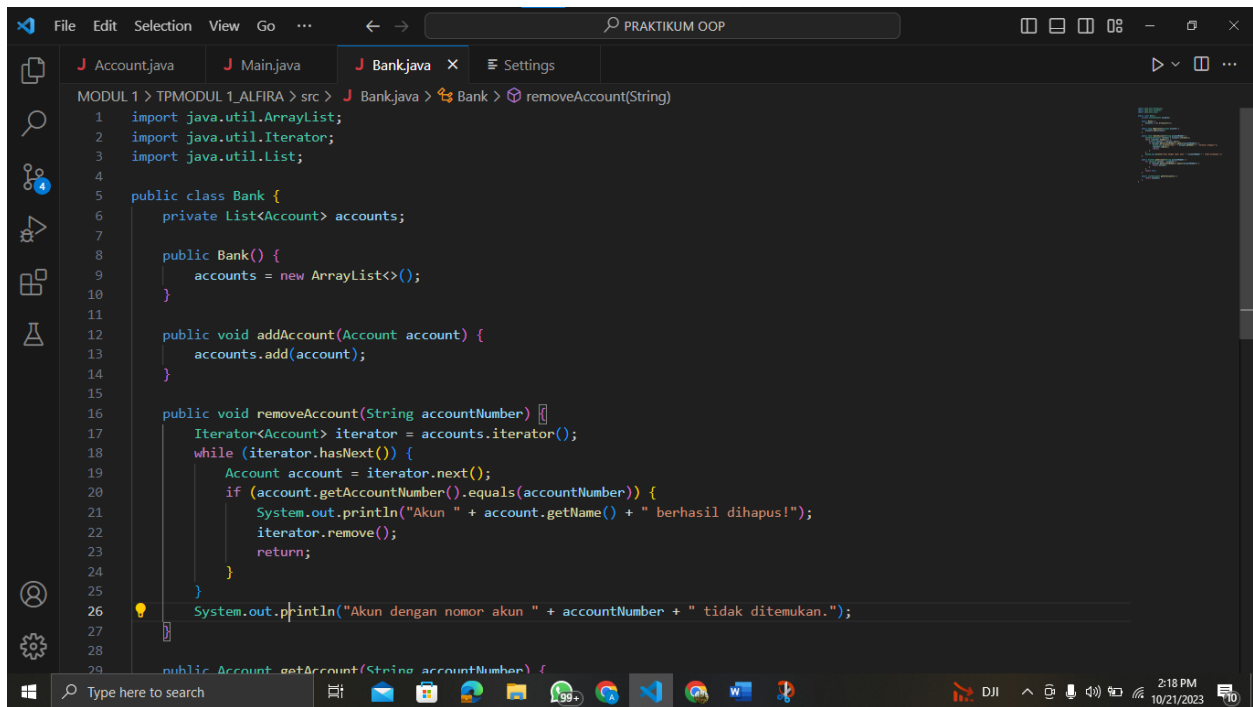
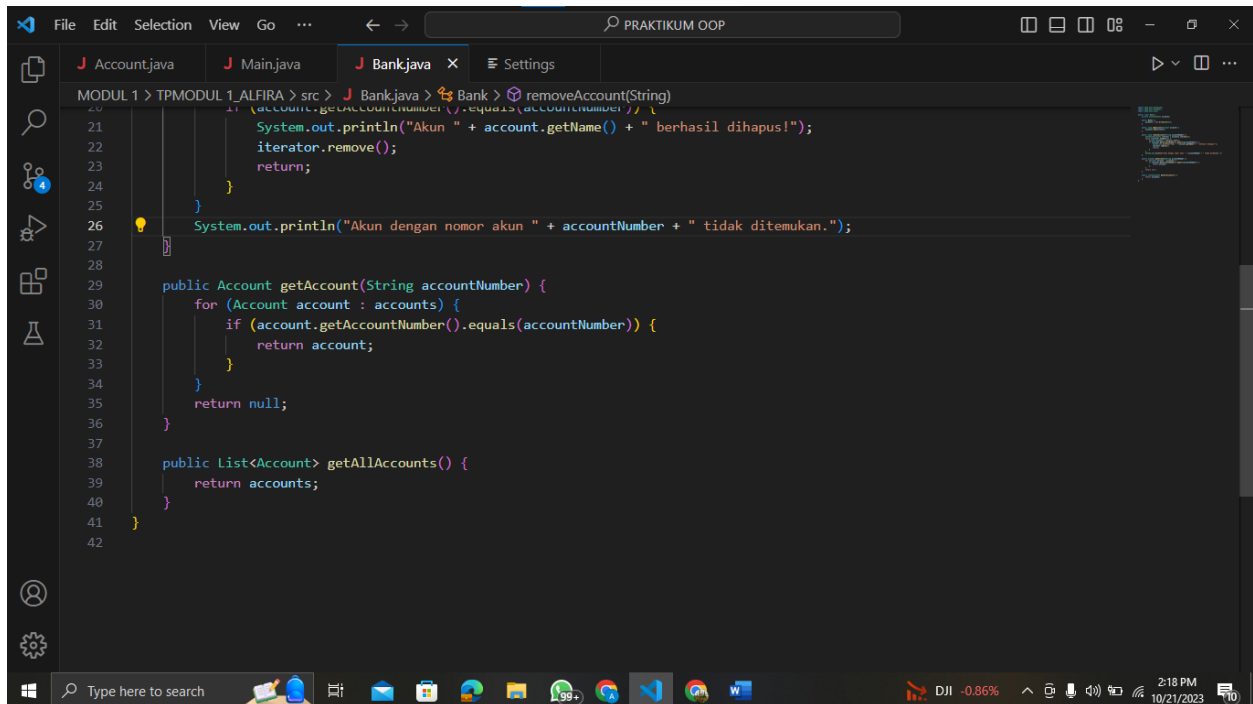


Class Bank

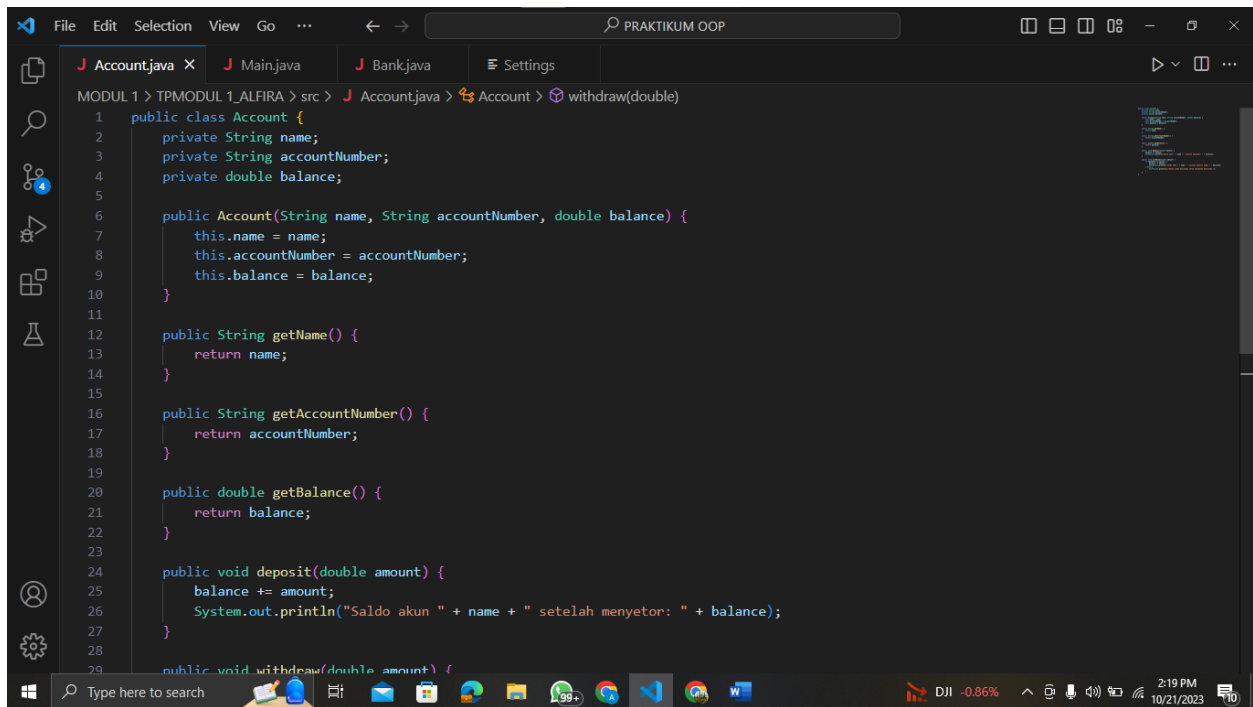


```
File Edit Selection View Go ... PRAKTIKUM OOP
J Account.java J Main.java J Bank.java X Settings
MODUL 1 > TPMODUL 1_ALFIRA > src > J Bank.java > Bank > removeAccount(String)
1 import java.util.ArrayList;
2 import java.util.Iterator;
3 import java.util.List;
4
5 public class Bank {
6     private List<Account> accounts;
7
8     public Bank() {
9         accounts = new ArrayList<>();
10    }
11
12    public void addAccount(Account account) {
13        accounts.add(account);
14    }
15
16    public void removeAccount(String accountNumber) {
17        Iterator<Account> iterator = accounts.iterator();
18        while (iterator.hasNext()) {
19            Account account = iterator.next();
20            if (account.getAccountNumber().equals(accountNumber)) {
21                System.out.println("Akun " + account.getName() + " berhasil dihapus!");
22                iterator.remove();
23                return;
24            }
25        }
26        System.out.println("Akun dengan nomor akun " + accountNumber + " tidak ditemukan.");
27    }
28
29    public Account getAccount(String accountNumber) {
```



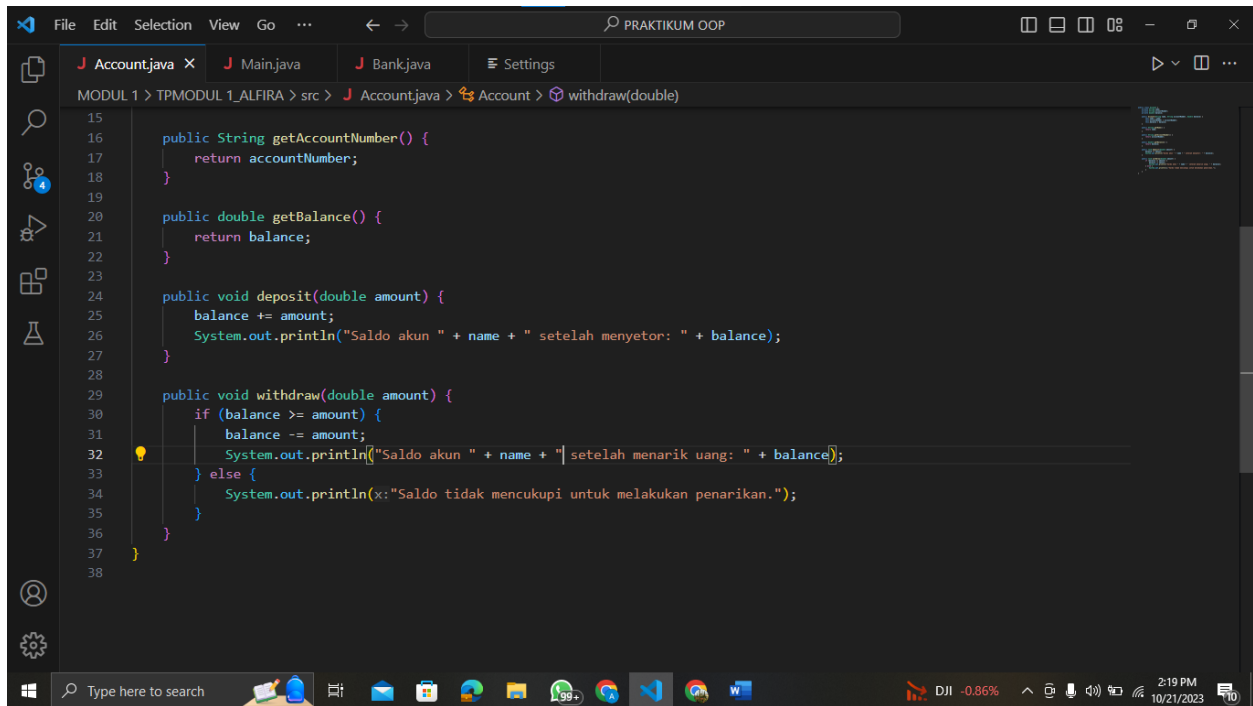
```
File Edit Selection View Go ... PRAKTIKUM OOP
J Account.java J Main.java J Bank.java X Settings
MODUL 1 > TPMODUL 1_ALFIRA > src > J Bank.java > Bank > removeAccount(String)
20 (account.getAccountNumber().equals(accountNumber)) {
21     System.out.println("Akun " + account.getName() + " berhasil dihapus!");
22     iterator.remove();
23     return;
24 }
25
26 System.out.println("Akun dengan nomor akun " + accountNumber + " tidak ditemukan.");
27
28
29 public Account getAccount(String accountNumber) {
30     for (Account account : accounts) {
31         if (account.getAccountNumber().equals(accountNumber)) {
32             return account;
33         }
34     }
35     return null;
36 }
37
38 public List<Account> getAllAccounts() {
39     return accounts;
40 }
41
42 }
```

Class Account



The screenshot shows a code editor with the following Java code for the Account class:

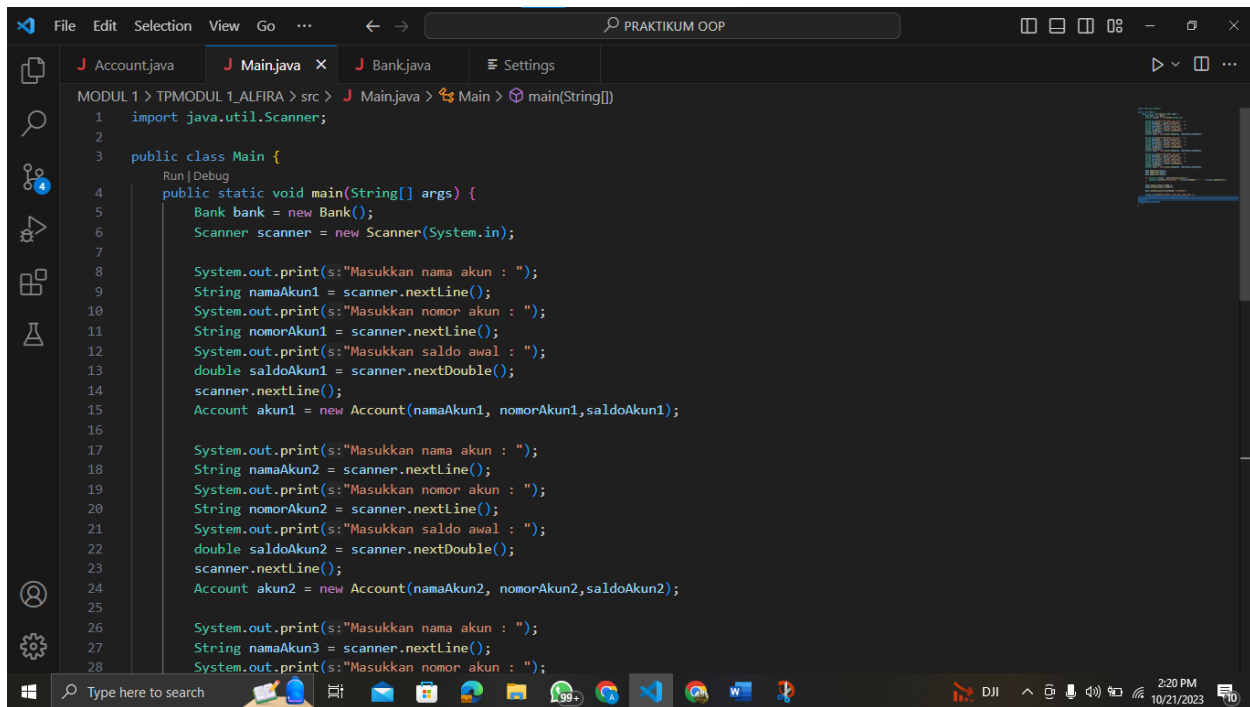
```
MODUL 1 > TPMODUL 1_ALFIRA > src > J Account.java > Account > withdraw(double)
1 public class Account {
2     private String name;
3     private String accountNumber;
4     private double balance;
5
6     public Account(String name, String accountNumber, double balance) {
7         this.name = name;
8         this.accountNumber = accountNumber;
9         this.balance = balance;
10    }
11
12    public String getName() {
13        return name;
14    }
15
16    public String getAccountNumber() {
17        return accountNumber;
18    }
19
20    public double getBalance() {
21        return balance;
22    }
23
24    public void deposit(double amount) {
25        balance += amount;
26        System.out.println("Saldo akun " + name + " setelah menyetor: " + balance);
27    }
28
29    public void withdraw(double amount) {
```



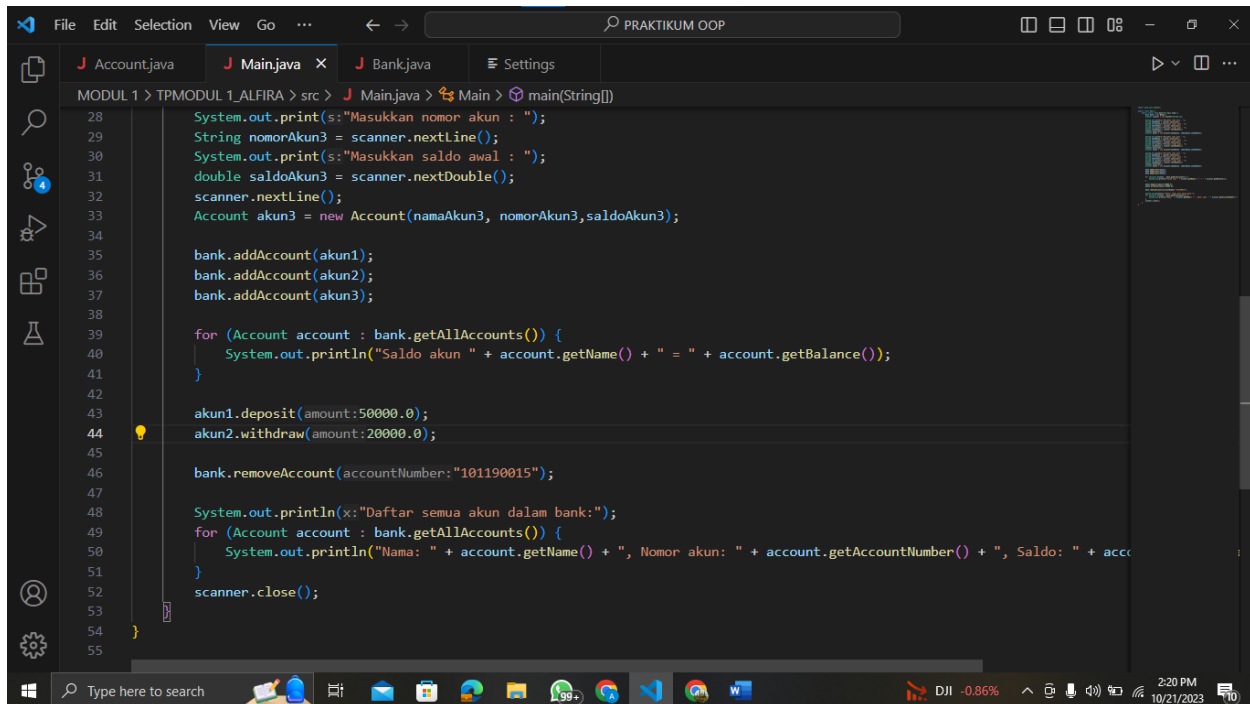
The screenshot shows the continuation of the Java code for the Account class:

```
15
16    public String getAccountNumber() {
17        return accountNumber;
18    }
19
20    public double getBalance() {
21        return balance;
22    }
23
24    public void deposit(double amount) {
25        balance += amount;
26        System.out.println("Saldo akun " + name + " setelah menyetor: " + balance);
27    }
28
29    public void withdraw(double amount) {
30        if (balance >= amount) {
31            balance -= amount;
32            System.out.println("Saldo akun " + name + " setelah menarik uang: " + balance);
33        } else {
34            System.out.println(x:"Saldo tidak mencukupi untuk melakukan penarikan.");
35        }
36    }
37
38 }
```

Class Main

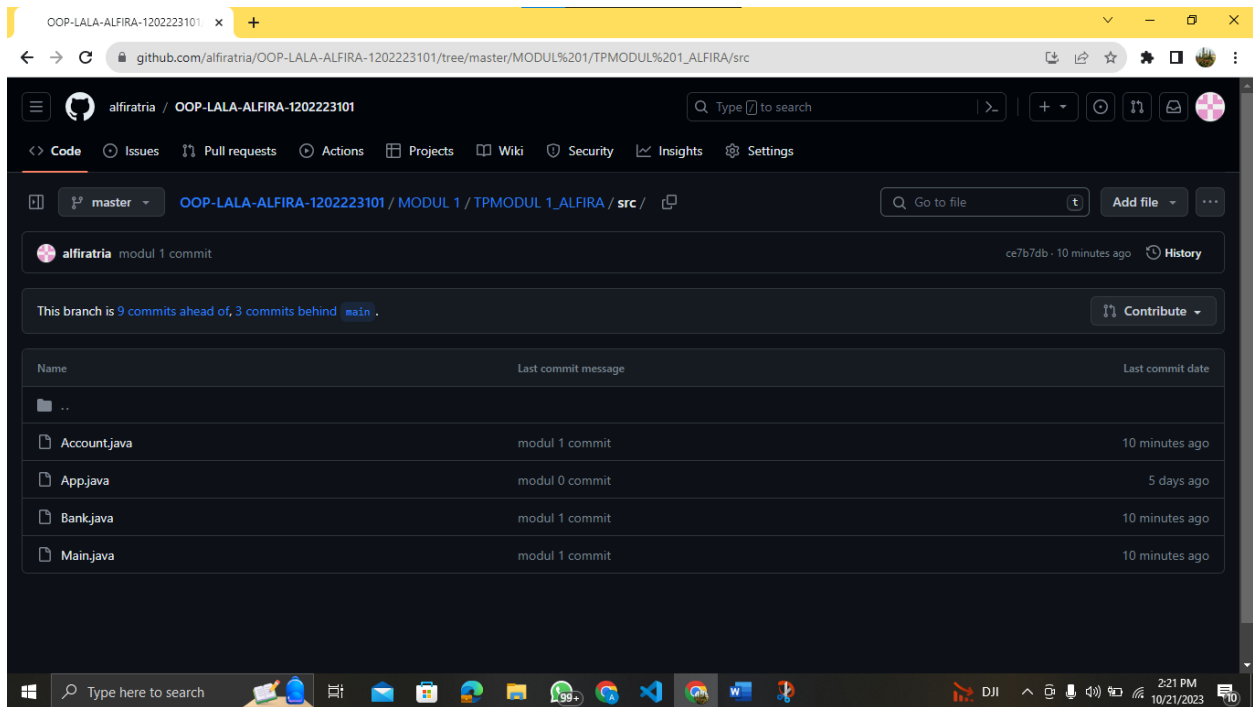


```
1 import java.util.Scanner;
2
3 public class Main {
4     public static void main(String[] args) {
5         Bank bank = new Bank();
6         Scanner scanner = new Scanner(System.in);
7
8         System.out.print(s:"Masukkan nama akun : ");
9         String namaAkun1 = scanner.nextLine();
10        System.out.print(s:"Masukkan nomor akun : ");
11        String nomorAkun1 = scanner.nextLine();
12        System.out.print(s:"Masukkan saldo awal : ");
13        double saldoAkun1 = scanner.nextDouble();
14        scanner.nextLine();
15        Account akun1 = new Account(namaAkun1, nomorAkun1,saldoAkun1);
16
17        System.out.print(s:"Masukkan nama akun : ");
18        String namaAkun2 = scanner.nextLine();
19        System.out.print(s:"Masukkan nomor akun : ");
20        String nomorAkun2 = scanner.nextLine();
21        System.out.print(s:"Masukkan saldo awal : ");
22        double saldoAkun2 = scanner.nextDouble();
23        scanner.nextLine();
24        Account akun2 = new Account(namaAkun2, nomorAkun2,saldoAkun2);
25
26        System.out.print(s:"Masukkan nama akun : ");
27        String namaAkun3 = scanner.nextLine();
28        System.out.print(s:"Masukkan nomor akun : ");
```



```
28        System.out.print(s:"Masukkan nomor akun : ");
29        String nomorAkun3 = scanner.nextLine();
30        System.out.print(s:"Masukkan saldo awal : ");
31        double saldoAkun3 = scanner.nextDouble();
32        scanner.nextLine();
33        Account akun3 = new Account(namaAkun3, nomorAkun3,saldoAkun3);
34
35        bank.addAccount(akun1);
36        bank.addAccount(akun2);
37        bank.addAccount(akun3);
38
39        for (Account account : bank.getAllAccounts()) {
40            System.out.println("Saldo akun " + account.getName() + " = " + account.getBalance());
41        }
42
43        akun1.deposit(amount:50000.0);
44        akun2.withdraw(amount:20000.0);
45
46        bank.removeAccount(accountNumber:"101190015");
47
48        System.out.println(x:"Daftar semua akun dalam bank:");
49        for (Account account : bank.getAllAccounts()) {
50            System.out.println("Nama: " + account.getName() + ", Nomor akun: " + account.getAccountNumber() + ", Saldo: " + account.getBalance());
51        }
52        scanner.close();
53    }
54 }
55
```

Push ke github



Penamaan di vscode

