Struktur Data dan Algoritma
Abstract Data Type

Nama:	
Kelas:	**NPM:**

Tujuan latihan:

- membuat program Java dengan menggunakan Abstract Data Type dalam Java.

Analisa kompleksitas dari program di bawah ini. Ubahlah program di bawah ini menggunakan ADT yang sesuai!

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class SDA13141LV2
{
    public static void main (String[] args) throws IOException
    {
        BufferedReader reader = new BufferedReader (new InputStreamReader (
                System.in));

        Bank bank = new Bank (5);

        String line = reader.readLine ();
        String[] lineSplit = line.split (" ");
        bank.addNewAccount (new BankAccount ("Ani", Long
                .parseLong (lineSplit[0])));
        bank.addNewAccount (new BankAccount ("Budi", Long
                .parseLong (lineSplit[1])));
        bank.addNewAccount (new BankAccount ("Chanek", Long
                .parseLong (lineSplit[2])));
        bank.addNewAccount (new BankAccount ("Dadang", Long
                .parseLong (lineSplit[3])));
        bank.addNewAccount (new BankAccount ("Emon", Long
                .parseLong (lineSplit[4])));

        int totalTransaction = Integer.parseInt (reader.readLine ());
        for (int ii = 0; ii < totalTransaction; ii++) {
            line = reader.readLine ();
            String[] split = line.split (" setor ");
            long uang = Long.parseLong (split[1]);
            BankAccount account = bank.getAccount (split[0]);
            account.deposit (uang);
        }
        reader.close ();

        bank.printAccounts ();
    }
}
```

```
1 2 3 4 5
6
Chanek setor 1000
Ani setor 2000
Chanek setor 4000
Budi setor 5000
Emon setor 6000
Dadang setor 2500
```

```
class Bank
{
    int lastIndex;
    private BankAccount[] accounts;

    public Bank (int totalAccount)
    {
        lastIndex = 0;
        accounts = new BankAccount[5];
    }

    public void addNewAccount (BankAccount account)
    {
        accounts[lastIndex] = account;
        lastIndex++;
    }

    public BankAccount getAccount (String name)
    {
        for (int ii = 0; ii < lastIndex; ii++) {
            if (name.equals (accounts[ii].getName ())) {
                return accounts[ii];
            }
        }
        throw new IllegalArgumentException ("name not found");
    }

    public void printAccounts ()
    {
        for (int ii = 0; ii < lastIndex; ii++) {
            System.out.println (accounts[ii]);
        }
    }
}
```

```
class BankAccount
{
    private String name;
    private long balance;

    public BankAccount (String name)
    {
        this (name, 0);
    }

    public BankAccount (String name,
        long balance)
    {
        this.name = name;
        this.balance = balance;
    }

    public String getName ()
    {
        return name;
    }

    public long getBalance ()
    {
        return balance;
    }

    public void deposit (long balance)
    {
        this.balance += balance;
    }

    public String toString ()
    {
        return name + " " + balance;
    }
```