Nama : Ananda Luthfiah Febiani
NIM : 2309106022
Praktikkum : Pemograman Berorientasi Objek

USER

```java
import java.util.*;
import java.io.*;
import java.util.ArrayList;
import java.util.Scanner;

public class App {
    static abstract class User {
        private final String username;
        private final String password;
        protected String role;

        public User(String username, String password, String role) {
            this.username = username;
            this.password = password;
            this.role = role;
        }
        public String getUsername() {
            return username;
        }
        public final boolean checkPassword(String pw) {
            return password.equals(pw);
        }
        public abstract void displayInfo();
    }

    static class AppAdmin extends User {
        public AppAdmin(String username, String password) {
            super(username, password, role:"admin");
        }
        public void accessPanel() throws Exception {
            Admin.main(new String[]{});
        }
        @Override
        public void displayInfo() {
            System.out.println("[ADMIN] Username: " + getUsername());
        }
    }

    public static class Customer extends User {
        private int saldo;
        public Customer(String username, String password, int saldo) {
```

```java
            super(username, password, role:"user");
            this.saldo = saldo;
        }
        public int getSaldo() {
            return saldo;
        }
        public void topUp(int amount) {
            if (amount > 0) {
                saldo += amount;
                System.out.println("Top-up berhasil. Saldo: " + saldo);
            }
        }
        public boolean kurangiSaldo(int amount) {
            if(amount <= saldo) {
                saldo -= amount;
                return true;
            }
            return false;
        }
        @Override
        public void displayInfo() {
            System.out.println("[CUSTOMER] Username: " + getUsername() + ", Saldo: " + saldo);
        }
    }

    private static ArrayList<User> users = new ArrayList<>();
    Run main | Debug main | Run | Debug
    public static void main(String[] args) throws Exception {
        Scanner sc = new Scanner(System.in);
        users.add(new AppAdmin(username:"admin",password:"admin123"));
        users.add(new Customer(username:"user",password:"user123",saldo:100_000));
```

```
        int choice;
        do {
            System.out.println(x:"\n=== SELAMAT DATANG DI PEMESANAN TIKET KONSER SAMARINDA ===");
            System.out.println(x:"1. Login");
            System.out.println(x:"2. Register (Customer)");
            System.out.println(x:"3. Exit");
            System.out.print(s:"Pilih: ");
            choice = sc.nextInt(); sc.nextLine();

            switch(choice) {
            case 1: loginFlow(sc); break;
            case 2: registerFlow(sc); break;
            case 3: System.out.println(x:"Keluar program."); break;
            default: System.out.println(x:"Pilihan invalid.");
            }
        } while(choice!=3);
        sc.close();
    }

    private static void loginFlow(Scanner sc) throws Exception {
        System.out.print(s:"Username: ");
        String u = sc.nextLine();
        System.out.print(s:"Password: ");
        String p = sc.nextLine();

        for (User usr: users) {
            if (usr.getUsername().equals(u) && usr.checkPassword(p)) {
                System.out.println("Login berhasil sebagai " + usr.role);
                if (usr instanceof AppAdmin) {
                    ((AppAdmin)usr).accessPanel();
                } else {
                    customerMenu(sc, (Customer)usr);
                }
                return;
            }
        }
        System.out.println(x:"Login gagal!");
    }
```

```
    private static void registerFlow(Scanner sc) {
        System.out.print(s:"Username baru: ");
        String u = sc.nextLine();
        System.out.print(s:"Password baru: ");
        String p = sc.nextLine();
        for (User usr: users) {
            if (usr.getUsername().equals(u)) {
                System.out.println(x:"Username sudah ada!");
                return;
            }
        }
        Customer c = new Customer(u,p,saldo:0);
        users.add(c);
        System.out.println("Registrasi sukses. Saldo awal: " + c.getSaldo());
    }

    private static void customerMenu(Scanner sc, Customer c) {
        int m;
        do {
            System.out.println("\n-- MENU CUSTOMER ("+c.getUsername()+") --");
            System.out.println(x:"1. Top-up Saldo");
            System.out.println(x:"2. Lihat Saldo");
            System.out.println(x:"3. Lihat Konser yang Tersedia");
            System.out.println(x:"4. Pesan Tiket Konser");
            System.out.println(x:"5. Lihat Riwayat Transaksi");
            System.out.println(x:"6. Lihat Tiket yang Sudah Dibeli");
            System.out.println(x:"7. Logout");
            System.out.print(s:"Pilih: ");
            m = sc.nextInt(); sc.nextLine();

            switch(m) {
                case 1:
                    System.out.print(s:"Jumlah top-up: ");
                    int amt = sc.nextInt(); sc.nextLine();
                    c.topUp(amt);
                    break;
                case 2:
                    System.out.println("Saldo Anda: " + c.getSaldo());
                    break;
                case 3:
```

```
                tampilkanKonser();
                break;
            case 4:
                pesanTiket(sc, c);
                break;
            case 5:
                tampilkanTransaksiUser(c.getUsername());
                break;
            case 6:
                tampilkanTiketUser(c.getUsername());
                break;
            case 7:
                System.out.println(x:"Logout Customer.");
                break;
            default:
                System.out.println(x:"Invalid.");
        }
    } while(m!=7);
}

private static void tampilkanKonser() {
    try (BufferedReader br = new BufferedReader(new FileReader(fileName:"database.txt"))) {
        System.out.println(x:"\n| Hari/Tanggal | Konser | Lokasi | Tiket Tersedia | Deskripsi |");
        String line;
        while ((line = br.readLine()) != null) {
            String[] f = line.split(regex:",", limit:6);
            System.out.printf(format:"|    %-12s    | %-10s  | %-10s  |      %-14s     |     %s    |\n", f[1], f[2], f[3], f[4], f[5]);
        }
    } catch (IOException e) {
        System.err.println(x:"Gagal membaca konser.");
    }
}
```

```
private static void pesanTiket(Scanner sc,
    try (BufferedReader br = new BufferedR
        ArrayList<String> tiketList = new ArrayList<>();
        String line;
        int index = 1;
        System.out.println(x:"\nDaftar Tiket:");
        while ((line = br.readLine()) != null) {
            String[] f = line.split(regex:",");
            if (f[3].equalsIgnoreCase(anotherString:"tersedia")) {
                tiketList.add(line);
                System.out.printf(format:"%d. ID: %s | Konser: %s | Harga: %s\n", index++, f[0], f[1], f[2]);
            }
        }
        if (tiketList.isEmpty()) {
            System.out.println(x:"Tidak ada tiket tersedia.");
            return;
        }
        System.out.print(s:"Pilih tiket nomor: ");
        int pilih = sc.nextInt(); sc.nextLine();
        if (pilih < 1 || pilih > tiketList.size()) {
            System.out.println(x:"Pilihan tidak valid.");
            return;
        }
        String[] t = tiketList.get(pilih-1).split(regex:",");
        int harga = Integer.parseInt(t[2]);
        if (!c.kurangiSaldo(harga)) {
            System.out.println(x:"Saldo tidak cukup.");
            return;
        }
        t[3] = c.getUsername();
        try (BufferedWriter bw = new BufferedWriter(new FileWriter(fileName:"transaksi.txt", append:true))) {
            bw.write(UUID.randomUUID() + "," + c.getUsername() + "," + t[1] + ",1," + harga);
            bw.newLine();
        }
        File inputFile = new File(pathname:"tiket.txt");
        File tempFile = new File(pathname:"temp_tiket.txt");
        try (BufferedReader reader = new BufferedReader(new FileReader(inputFile));
             BufferedWriter writer = new BufferedWriter(new FileWriter(tempFile))) {
            String currentLine;
            while ((currentLine = reader.readLine()) != null) {
```

Constructs an empty list with an initial capacity of ten.

```
                    if (currentLine.equals(tiketList.get(pilih-1))) {
                        writer.write(String.join(delimiter:",", t));
                    } else {
                        writer.write(currentLine);
                    }
                    writer.newLine();
                }
            }
            inputFile.delete();
            tempFile.renameTo(inputFile);
            System.out.println(x:"Tiket berhasil dipesan.");
        } catch (IOException e) {
            System.err.println(x:"Gagal memesan tiket.");
        }
    }

    private static void tampilkanTransaksiUser(String username) {
        try (BufferedReader br = new BufferedReader(new FileReader(fileName:"transaksi.txt"))) {
            System.out.println(x:"\nRiwayat Transaksi:");
            String line;
            while ((line = br.readLine()) != null) {
                String[] f = line.split(regex:",");
                if (f[1].equals(username)) {
                    System.out.printf(format:"ID: %s | Konser: %s | Jumlah: %s | Total: %s\n",
                            f[0], f[2], f[3], f[4]);
                }
            }
        } catch (IOException e) {
            System.err.println(x:"Gagal membaca transaksi.");
        }
    }

    private static void tampilkanTiketUser(String username) {
        try (BufferedReader br = new BufferedReader(new FileReader(fileName:"tiket.txt"))) {
            System.out.println(x:"\nTiket yang Dimiliki:");
            String line;
            while ((line = br.readLine()) != null) {
                String[] f = line.split(regex:",");
                if (f[3].equals(username)) {
```

```
                    System.out.printf(format:"ID: %s | Konser: %s | Harga: %s\n", f[0], f[1], f[2]);
                }
            }
        } catch (IOException e) {
            System.err.println(x:"Gagal membaca tiket.");
        }
    }
}
```

ADMIN

```java
import java.io.*;
import java.util.Arrays;
import java.util.Scanner;
import java.util.StringTokenizer;

public class Admin {
    public static void main(String[] args) throws IOException {
        Scanner input = new Scanner(System.in);
        int choice;
        boolean lanjut = true;

        do {
            clearScreen();
            System.out.println(x:"Selamat Datang Admin di penjualan Tiket Konser Samarinda\n");
            System.out.println(x:"1. Menu Lihat");
            System.out.println(x:"2. Cari Konser");
            System.out.println(x:"3. Tambah Konser");
            System.out.println(x:"4. Ubah Konser");
            System.out.println(x:"5. Hapus Konser");
            System.out.println(x:"6. Keluar");
            System.out.print(s:"\nPilih menu: ");
            choice = input.nextInt();
            input.nextLine();

            switch (choice) {
                case 1:
                    showSubMenu(input);
                    break;
                case 2:
                    System.out.println(x:"\n=========\nCARI KONSER\n=========");
                    cariData();
                    break;
                case 3:
                    System.out.println(x:"\n================\nTAMBAH KONSER\n================");
                    tambahData();
                    break;
                case 4:
                    System.out.println(x:"\n==============\nUBAH KONSER\n==============");
                    ubahData();
                    break;
                case 5:
```

```
                    System.out.println(x:"\n===============\nHAPUS KONSER\n===============");
                    hapusData();
                    break;
                case 6:
                    System.out.println(x:"Keluar dari Admin Panel.");
                    lanjut = false;
                    break;
                default:
                    System.err.println(x:"Pilihan tidak ditemukan. Silakan coba lagi.");
                }

                if (lanjut) {
                    lanjut = getYesorNo(message:"Apakah Anda ingin melanjutkan?");
                }
            } while (lanjut);

            input.close();
        }

        private static void showSubMenu(Scanner scanner) throws IOException {
            System.out.println(x:"\n=================");
            System.out.println(x:"LIHAT SELURUH DATA");
            System.out.println(x:"1. Konser");
            System.out.println(x:"2. Tiket");
            System.out.println(x:"3. Transaksi");
            System.out.println(x:"4. Kembali");
            System.out.print(s:"Pilih: ");
            int sub = scanner.nextInt();
            scanner.nextLine();

            switch (sub) {
                case 1:
                    tampilkanKonser();
                    break;
                case 2:
                    tampilkanTiket();
                    break;
                case 3:
                    tampilkanTransaksi();
```

```
                    break;
                case 4:
                    // kembali
                    break;
                default:
                    System.out.println(x:"Pilihan tidak valid!");
            }
        }

        private static void tampilkanKonser() throws IOException {
            try (BufferedReader br = new BufferedReader(new FileReader(fileName:"database.txt"))) {
                System.out.println(x:"\n| No |       Hari/Tanggal       |         Konser          |     Lokasi      | Tiket Tersedia, Deskripsi |");
                System.out.println(x:"-------------------------------------------------------------------------------------------");
                String line;
                int no = 1;
                while ((line = br.readLine()) != null) {
                    String[] f = line.split(regex:",", limit:6);
                    System.out.printf(format:"| %2d | %-12s/%-10s | %-15s | %-14s | %s |\n", no++, f[1], f[2], f[3], f[4], f[5]);
                }
            } catch (FileNotFoundException e) {
                System.err.println(x:"database.txt tidak ditemukan. Silakan tambah konser terlebih dahulu.");
            }
        }

        private static void tampilkanTiket() throws IOException {
            File file = new File(pathname:"tiket.txt");
            if (!file.exists()) {
                System.err.println(x:"Belum ada tiket yang tersedia.");
                return;
            }
            try (BufferedReader br = new BufferedReader(new FileReader(file))) {
                System.out.println(x:"\n| No | Tiket ID | Konser | Harga | Status |");
                System.out.println(x:"-----------------------------------------------");
                String line;
                int no = 1;
                while ((line = br.readLine()) != null) {
                    String[] f = line.split(regex:",", limit:4);
                    System.out.printf(format:"| %2d | %-8s | %-10s | %-5s | %s\n",
                            no++, f[0], f[1], f[2], f[3]);
```

```
120             }
121         }
122     }
123
124     private static void tampilkanTransaksi() throws IOException {
125         File file = new File(pathname:"transaksi.txt");
126         if (!file.exists()) {
127             System.err.println(x:"Belum ada transaksi yang tercatat.");
128             return;
129         }
130         try (BufferedReader br = new BufferedReader(new FileReader(file))) {
131             System.out.println(x:"\n| No | ID Transaksi | User | Konser | Jumlah | Total |");
132             System.out.println(x:"-----------------------------------------------------");
133             String line;
134             int no = 1;
135             while ((line = br.readLine()) != null) {
136                 String[] f = line.split(regex:",", limit:6);
137                 System.out.printf(format:"| %2d | %-12s | %-6s | %-6s | %-6s | %s\n",
138                         no++, f[0], f[1], f[2], f[3], f[4]);
139             }
140         }
141     }
142
143     private static void cariData() throws IOException{
144         try {
145             File file = new File(pathname:"database.txt");
146         } catch (Exception e){
147             System.err.println(x:"Konser Tidak ditemukan");
148             System.err.println(x:"Silahkan tambah konser terlebih dahulu");
149             return;
150         }
151
152         Scanner Input = new Scanner(System.in);
153         System.out.print(s:"Masukan konser yang ingin dicari : ");
154         String cariString = Input.nextLine();
155         String[] keywords = cariString.split(regex:"\\s+");
156
157         cekKonser(keywords);
158     }
```

```java
private static void cekKonser(String[] keywords) throws IOException{

    FileReader fileInput = new FileReader(fileName:"database.txt");
    BufferedReader bufferInput = new BufferedReader(fileInput);

    String data = bufferInput.readLine();
    boolean isExist;
    int nomorData = 0;
    int tiket = 0;
    System.out.println(x:"\n| No        |\tTanggal       |\tKonser        |\tTiket Tersedia     |\tDeskripsi          |\tLokasi     |");
    System.out.println(x:"-------------------------------------------------------------------------------------------------------------------");

    while(data != null){

        isExist = true;

        for(String keyword:keywords){
            isExist = isExist && data.toLowerCase().contains(keyword.toLowerCase());
        }

        if(isExist){
            nomorData++;
            tiket++;
            StringTokenizer stringToken = new StringTokenizer(data, delim:",");

            stringToken.nextToken();
            System.out.printf(format:"| %2d ", nomorData);
            System.out.printf(format:"|\t%-20s  ", stringToken.nextToken());
            System.out.printf(format:"|\t%-20s   ", stringToken.nextToken());
            System.out.printf(format:"|\t%-20s   ", stringToken.nextToken());
            System.out.printf(format:"| %2d ", tiket);
            System.out.printf(format:"|\t%s   ", stringToken.nextToken());
            System.out.print(s:"\n");
        }

        data = bufferInput.readLine();
    }
    System.out.println(x:"-------------------------------------------------------------------------------------------------------------------");

}
```

```java
private static void tambahData() throws IOException{

    FileWriter fileOutput = new FileWriter(fileName Loading... .txt",append:true);
    BufferedWriter bufferOutput = new BufferedWriter(fileOutput);

    Scanner Input = new Scanner(System.in);
    String nomor, tanggal, konser, lokasi, tiket, deskripsi;

    System.out.print(s:"masukkan nomor: ");
    nomor = Input.nextLine();
    tanggal = ambiltanggal();
    System.out.print(s:"masukkan konser: ");
    konser = Input.nextLine();
    System.out.print(s:"masukkan lokasi: ");
    lokasi = Input.nextLine();
    System.out.print(s:"masukkan tiket: ");
    tiket = Input.nextLine();
    System.out.print(s:"masukkan deskripsi konser: ");
    deskripsi = Input.nextLine();

    String[] keywords = {nomor + "," + tanggal + "," + konser + "," + lokasi + "," + tiket + "," + deskripsi};
    boolean isExist = false;
    try (BufferedReader reader = new BufferedReader(new FileReader(fileName:"database.txt"))) {
        String line;
    }

    if (!isExist){
        System.out.println(ambilEntryPertanggal(nomor, tanggal));
        String no = nomor.replaceAll(regex:"\\s+",replacement:"");
        System.out.println(x:"\nkonser yang akan anda masukan adalah");
        System.out.println(x:"----------------------------------------");
        System.out.println("nomor       : " + no);
        System.out.println("tanggal     : " + tanggal);
        System.out.println("konser      : " + konser);
        System.out.println("lokasi      : " + lokasi);
        System.out.println("tiket       : " + tiket);
        System.out.println("deskripsi   : " + deskripsi);

        boolean isTambah = getYesorNo(message:"Apakah akan ingin menambah data konser tersebut? ");
```

```
            if(isTambah){
                bufferOutput.write(no + "," + tanggal + ","+ konser +"," + lokasi + "," + tiket + "," + deskripsi);
                bufferOutput.newLine();
                bufferOutput.flush();
                System.out.println(x:"konser  dan terupdate");

            }

        } else {
            System.out.println(x:"konser yang anda akan masukan sudah tersedia di data konser:");
            cekKonser(keywords,isDisplay:true);
        }
        bufferOutput.close();
    }

    private static long ambilEntryPertanggal(String nomor, String tanggal) throws IOException {
        FileReader fileInput = new FileReader(fileName:"database.txt");
        BufferedReader bufferInput = new BufferedReader(fileInput);

        long entry = 0;
        String data = bufferInput.readLine();
        Scanner dataScanner;
        String no;

        while(data != null){
            dataScanner = new Scanner(data);
            dataScanner.useDelimiter(pattern:",");
            no = dataScanner.next();
            dataScanner = new Scanner(no);
            dataScanner.useDelimiter(pattern:"_");

            nomor = nomor.replaceAll(regex:"\\s+",replacement:"");

            if (nomor.equalsIgnoreCase(dataScanner.next()) && tanggal.equalsIgnoreCase(dataScanner.next()) ) {
                entry = dataScanner.nextInt();
            }
            data = bufferInput.readLine();
        }
        return entry;
    }
```

```
    private static boolean cekKonser(String[] keywords, boolean isDisplay) throws IOException{

        FileReader fileInput = new FileReader(fileName:"database.txt");
        BufferedReader bufferInput = new BufferedReader(fileInput);

        String data = bufferInput.readLine();
        boolean isExist = false;
        int nomorData = 0;
        int tiket = 0;

        if (isDisplay) {
            System.out.println(x:"\n| No |\tHari/Tanggal       |\tKonser        |\tLokasi       |\tTiket Tersedia     |\tDeskripsi         |");
            System.out.println(x:"-----------------------------------------------------------------------------------------------------------");
        }

        while(data != null){

            isExist = true;

            for(String keyword:keywords){
                isExist = isExist && data.toLowerCase().contains(keyword.toLowerCase());
            }

            if(isExist){
                if(isDisplay) {
                    nomorData++;
                    tiket++;
                    StringTokenizer stringToken = new StringTokenizer(data, delim:",");

                    stringToken.nextToken();
                    System.out.printf(format:"| %2d ", nomorData);
                    System.out.printf(format:"|\t%-20s  ", stringToken.nextToken());
                    System.out.printf(format:"|\t%-20s   ", stringToken.nextToken());
                    System.out.printf(format:"|\t%-20s   ", stringToken.nextToken());
                    System.out.printf(format:"| %2d ", tiket);
                    System.out.printf(format:"|\t%s   ", stringToken.nextToken());
                    System.out.print(s:"\n");
                } else {
                    break;
                }
```

```
            }

            data = bufferInput.readLine();
        }

        if (isDisplay){
            System.out.println(x:"-----------------------------------------------------------------------------------------------------------");
        }

        return isExist;
    }

    private static String ambiltanggal() {
        Scanner input = new Scanner(System.in);
        String tanggalInput;
        boolean tanggalValid = false;

        do {
            System.out.print(s:"Masukkan hari dan tanggal (Senin, 25/03/2025): ");
            tanggalInput = input.nextLine();

            if (tanggalInput.matches(regex:"\\w+,\\s\\d{2}/\\d{2}/\\d{4}")) {
                String[] splitTanggal = tanggalInput.split(regex:",\\s");
                String[] splitTgl = splitTanggal[1].split(regex:"/");
                int day = Integer.parseInt(splitTgl[0]);
                int month = Integer.parseInt(splitTgl[1]);

                if (month >= 1 && month <= 12 && day >= 1 && day <= 31) {
                    tanggalValid = true;
                } else {
                    System.out.println(x:"Tanggal tidak valid. Mohon masukkan tanggal yang benar.");
                }
            } else {
                System.out.println(x:"Format tanggal yang anda masukkan salah. Harus dalam format: Hari, DD/MM/YYYY");
            }
        } while (!tanggalValid);

        return tanggalInput;
    }
```

```
    private static void ubahData() throws IOException {
        File database = new File(pathname:"database.txt");
        FileReader fileInput = new FileReader(database);
        BufferedReader bufferedInput = new BufferedReader(fileInput);

        File tempDB = new File(pathname:"tempDB.txt");
        FileWriter fileOutput = new FileWriter(tempDB);
        BufferedWriter bufferedOutput = new BufferedWriter(fileOutput);

        System.out.println(x:"List Konser");
        tampilkanKonser();

        Scanner Input = new Scanner(System.in);
        System.out.print(s:"\nMasukan konser mana yang akan diupdate: ");
        int updateNum = Input.nextInt();

        String data = bufferedInput.readLine();
        int entryCounts = 0;

        while (data != null){
            entryCounts++;

            StringTokenizer st = new StringTokenizer(data,delim:",");

            if (updateNum == entryCounts){
                System.out.println(x:"\nKonser yang ingin di update adalah:");
                System.out.println(x:"--------------------------------------");
                System.out.println("Referensi   : " + st.nextToken());
                System.out.println("nomor       : " + st.nextToken());
                System.out.println("tanggal     : " + st.nextToken());
                System.out.println("konser      : " + st.nextToken());
                System.out.println("lokasi      : " + st.nextToken());
                System.out.println("tiket       : " + st.nextToken());
                System.out.println("deskripsi   : " + st.nextToken());

                String[] fieldData = {"nomor", "tanggal", "lokasi", "konser", "tiket", "deskripsi"};
                String[] tempData = new String[6];

                st = new StringTokenizer(data,delim:",");
                String originalData = st.nextToken();
```

```
403                 for(int i=0; i < fieldData.length ; i++) {
404                     boolean isUpdate = getYesorNo("apakah anda ingin merubah " + fieldData[i]);
405                     originalData = st.nextToken();
406                     if (isUpdate){
407 
408                         if (fieldData[i].equalsIgnoreCase(anotherString:"tanggal")){
409                             System.out.print(s:"masukan tanggal(hari, DD/MM/YYYY): ");
410                             tempData[i] = ambiltanggal();
411                         } else {
412                             Input = new Scanner(System.in);
413                             System.out.print("\nMasukan " + fieldData[i] + " baru: ");
414                             tempData[i] = Input.nextLine();
415                         }
416 
417                     } else {
418                         tempData[i] = originalData;
419                     }
420                 }
421 
422                 st = new StringTokenizer(data,delim:",");
423                 st.nextToken();
424                 System.out.println(x:"\nkonser terbaru adalah ");
425                 System.out.println(x:"--------------------------------------");
426                 System.out.println("nomor       : " + st.nextToken() + " --> " + tempData[0]);
427                 System.out.println("tanggal     : " + st.nextToken() + " --> " + tempData[1]);
428                 System.out.println("konser      : " + st.nextToken() + " --> " + tempData[2]);
429                 System.out.println("lokasi      : " + st.nextToken() + " --> " + tempData[3]);
430                 System.out.println("tiket       : " + st.nextToken() + " --> " + tempData[4]);
431                 System.out.println("deskripsi   : " + st.nextToken() + " --> " + tempData[5]);
432 
433                 boolean isUpdate = getYesorNo(message:"apakah sudah yakin ingin mengupdate?");
434 
435                 if (isUpdate){
436 
437                     boolean isExist = cekKonser(tempData,isDisplay:false);
438 
439                     if(isExist){
440                         System.err.println(x:"data konser sudah ada di ubah, proses update dibatalkan, \nsilahkan delete data yang bersangkutan");
441                         bufferedOutput.write(data);
```

```
443                     } else {
444 
445                         String nomor = tempData[0];
446                         String tanggal = tempData[1];
447                         String konser = tempData[2];
448                         String lokasi = tempData[3];
449                         String tiket = tempData[4];
450                         String deskripsi = tempData[5];
451 
452                         long nmrEntry = ambilEntryPertanggal(nomor, tanggal) + 1;
453 
454                         String Konser = nomor.replaceAll(regex:"\\s+",replacement:"");
455                         String no = Konser+"_"+tanggal+"_"+nmrEntry;
456 
457                         bufferedOutput.write(no + "," + nomor + ","+ tanggal +"," + konser + ","+ lokasi + "," + "," + tiket + deskripsi);
458                     }
459                 } else {
460                     bufferedOutput.write(data);
461                 }
462             } else {
463                 bufferedOutput.write(data);
464             }
465             bufferedOutput.newLine();
466 
467             data = bufferedInput.readLine();
468         }
469 
470         bufferedOutput.flush();
471         database.delete();
472         tempDB.renameTo(database);
473     }
474 
475     private static void hapusData() throws  IOException{
476         File database = new File(pathname:"database.txt");
477         FileReader fileInput = new FileReader(database);
478         BufferedReader bufferedInput = new BufferedReader(fileInput);
479 
480         File tempDB = new File(pathname:"tempDB.txt");
481         FileWriter fileOutput = new FileWriter(tempDB);
```

```
482         BufferedWriter bufferedOutput = new BufferedWriter(fileOutput);
483
484         System.out.println(x:"List Konser");
485         tampilkanKonser();
486
487         Scanner Input = new Scanner(System.in);
488         System.out.print(s:"\nMasukan nomor konser yang ingin dihapus: ");
489         int deleteNum = Input.nextInt();
490
491
492         boolean isFound = false;
493         int entryCounts = 0;
494
495         String data = bufferedInput.readLine();
496
497         while (data != null){
498             entryCounts++;
499             boolean isDelete = false;
500
501             StringTokenizer st = new StringTokenizer(data,delim:",");
502
503             if (deleteNum == entryCounts){
504                 System.out.println(x:"\nKonser yang ingin anda hapus adalah:");
505                 System.out.println(x:"-----------------------------------");
506                 System.out.println("Referensi       : " + st.nextToken());
507                 System.out.println("nomor           : " + st.nextToken());
508                 System.out.println("tanggal         : " + st.nextToken());
509                 System.out.println("konser          : " + st.nextToken());
510                 System.out.println("lokasi          : " + st.nextToken());
511                 System.out.println("tiket          : " + st.nextToken());
512                 System.out.println("deskripsi       : " + st.nextToken());
513
514                 isDelete = getYesorNo(message:"Apakah yakin ingin menghapus?");
515                 isFound = true;
516             }
517
518             if(isDelete){
519                 System.out.println(x:"Data konser berhasil dihapus");
520                 data = bufferedInput.readLine();
```

```
                continue;
            }

            bufferedOutput.write(data);
            bufferedOutput.newLine();
            data = bufferedInput.readLine();

        }

        if(!isFound){
            System.err.println(x:"konser tidak ditemukan");
        }

        bufferedOutput.flush();
        bufferedOutput.close();
        bufferedInput.close();

        database.delete();
        tempDB.renameTo(database);
    }

    private static boolean getYesorNo(String message){
        Scanner Input = new Scanner(System.in);
        System.out.print("\n"+message+" (y/n)? ");
        String pilihan = Input.next();

        while(!pilihan.equalsIgnoreCase(anotherString:"y") && !pilihan.equalsIgnoreCase(anotherString:"n")) {
            System.err.println(x:"Pilihan anda bukan y atau n?");
            System.out.print("\n"+message+" (y/n)? ");
            pilihan = Input.next();
        }

        return pilihan.equalsIgnoreCase(anotherString:"y");

    }

    private static void clearScreen(){
        try {
            if (System.getProperty(key:"os.name").contains(s:"Windows")){
                new ProcessBuilder(...command:"cmd","/c","cls").inheritIO().start().waitFor();
            } else {
                System.out.print(s:"\033\143");
            }
        } catch (Exception ex){
            System.err.println(x:"tidak bisa refresh");
        }
    }
}
```