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

USER

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

interface Authentication {
    void login(Scanner sc) throws Exception;
    void register(Scanner sc);
}

class ErrorUtil {
    public void handleError(Exception e) {
        System.err.println("[ERROR] " + e.getMessage());
    }

    public void handleError(String message) {
        System.err.println("[ERROR] " + message);
    }
}

public class App implements Authentication {
    public static final String APP_VERSION = "1.0.0";
    private static final ErrorUtil errorUtil = new ErrorUtil();

    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);
        }
```

```
42          public abstract void displayInfo();
43      }
44
45      static class AppAdmin extends User {
46          public AppAdmin(String username, String password) {
47              super(username, password, role:"admin");
48          }
49
50          public void accessPanel() throws Exception {
51              Admin.main(new String[]{});
52          }
53
54          @Override
55          public void displayInfo() {
56              System.out.println("[ADMIN] Username: " + getUsername());
57          }
58      }
59
60      public static class Customer extends User {
61          private int saldo;
62
63          public Customer(String username, String password, int saldo) {
64              super(username, password, role:"user");
65              this.saldo = saldo;
66          }
67
68          public int getSaldo() {
69              return saldo;
70          }
71
72          public void topUp(int amount) {
73              if (amount > 0) {
74                  saldo += amount;
75                  System.out.println("Top-up berhasil. Saldo: " + saldo);
76              }
77          }
78
79          public boolean kurangiSaldo(int amount) {
80              if (amount <= saldo) {
```

```
81                  saldo -= amount;
82                  return true;
83              }
84              return false;
85          }
86
87          @Override
88          public void displayInfo() {
89              System.out.println("[CUSTOMER] Username: " + getUsername() + ", Saldo: " + saldo);
90          }
91      }
92
93      private static final List<User> users = new ArrayList<>();
94
        Run main | Debug main | Run | Debug
95      public static void main(String[] args) {
96          Scanner sc = new Scanner(System.in);
97          users.add(new AppAdmin(username:"admin", password:"admin123"));
98          users.add(new Customer(username:"user", password:"user123", saldo:100_000));
99
100         App app = new App();
101         int choice = -1;
102         do {
103             System.out.println("\n=== SELAMAT DATANG DI PEMESANAN TIKET KONSER SAMARINDA v" + APP_VERSION + " ===");
104             System.out.println(x:"1. Login");
105             System.out.println(x:"2. Register (Customer)");
106             System.out.println(x:"3. Exit");
107
108             try {
109                 System.out.print(s:"Pilih: ");
110                 choice = sc.nextInt();
111                 sc.nextLine();
112             } catch (InputMismatchException ime) {
113                 errorUtil.handleError(message:"Input invalid, masukkan angka");
114                 sc.nextLine();
115                 continue;
116             }
```

```
            switch (choice) {
                case 1:
                    try {
                        app.login(sc);
                    } catch (Exception e) {
                        errorUtil.handleError(e);
                    }
                    break;
                case 2:
                    app.register(sc);
                    break;
                case 3:
                    System.out.println(x:"Keluar program.");
                    break;
                default:
                    errorUtil.handleError(message:"Pilihan invalid.");
            }
        } while (choice != 3);
        sc.close();
    }

    @Override
    public void login(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;
            }
        }
```

```
157         System.out.println(x:"Login gagal!");
158     }
159
160     @Override
161     public void register(Scanner sc) {
162         System.out.print(s:"Username baru: ");
163         String u = sc.nextLine();
164         System.out.print(s:"Password baru: ");
165         String p = sc.nextLine();
166         for (User usr : users) {
167             if (usr.getUsername().equals(u)) {
168                 System.out.println(x:"Username sudah ada!");
169                 return;
170             }
171         }
172         Customer c = new Customer(u, p, saldo:0);
173         users.add(c);
174         System.out.println("Registrasi sukses. Saldo awal: " + c.getSaldo());
175     }
176
177     private static void customerMenu(Scanner sc, Customer c) {
178         int m;
179         do {
180             System.out.println("\n-- MENU CUSTOMER (" + c.getUsername() + ") --");
181             System.out.println(x:"1. Top-up Saldo");
182             System.out.println(x:"2. Lihat Saldo");
183             System.out.println(x:"3. Lihat Konser yang Tersedia");
184             System.out.println(x:"4. Pesan Tiket Konser");
185             System.out.println(x:"5. Lihat Riwayat Transaksi");
186             System.out.println(x:"6. Lihat Tiket yang Sudah Dibeli");
187             System.out.println(x:"7. Logout");
188             System.out.print(s:"Pilih: ");
189             m = sc.nextInt();
190             sc.nextLine();
191
192             switch (m) {
193                 case 1:
194                     System.out.print(s:"Jumlah top-up: ");
195                     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, Customer c) {
        try (BufferedReader br = new BufferedReader(new FileReader(fileName:"tiket.txt"))) {
            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) {
```

```
                    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.InputMismatchException;
import java.util.NoSuchElementException;
import java.util.Scanner;
import java.util.StringTokenizer;

interface ErrorHandling {
    void handlingError(Exception e);
    void handlingError(String pesan);
}

class ErrorUtil implements ErrorHandling {
    public static final int MAX_ERRORS = 3;
    private static int hitungerror = 0;

    @Override
    public void handlingError(Exception e) {
        hitungerror++;
        System.err.println("[ERROR] " + e.getMessage());
        if (hitungerror >= MAX_ERRORS) {
            System.err.println(x:"Anda sudah melampaui error yang ditetapkan. Program terhenti.");
            System.exit(status:1);
        }
    }

    @Override
    public void handlingError(String pesan) {
        hitungerror++;
        System.err.println("[ERROR] " + pesan);
        if (hitungerror >= MAX_ERRORS) {
            System.err.println(x:"Anda sudah melampaui error yang ditetapkan. Program terhenti.");
            System.exit(status:1);
        }
    }
}

public class Admin {
    private static final String APP_NAME = "Tiket Konser Samarinda - Admin";
    private static final ErrorUtil errorUtil = new ErrorUtil();
    public static void main(String[] args) throws IOException {
        Scanner input = new Scanner(System.in);
```

```
42          int choice;
43          boolean lanjut = true;
44
45          do {
46              clearScreen();
47              System.out.println(x:"Selamat Datang Admin di penjualan Tiket Konser Samarinda");
48              System.out.println("==================== " + APP_NAME + " ====================");
49
50              System.out.println(x:"1. Menu Lihat");
51              System.out.println(x:"2. Cari Konser");
52              System.out.println(x:"3. Tambah Konser");
53              System.out.println(x:"4. Ubah Konser");
54              System.out.println(x:"5. Hapus Konser");
55              System.out.println(x:"6. Keluar");
56
57              try {
58                  System.out.print(s:"\nPilih menu: ");
59                  choice = input.nextInt();
60                  input.nextLine();
61              } catch (InputMismatchException ime) {
62                  errorUtil.handlingError(pesan:"Input invalid, masukkan kembali dari angka dalam menu.");
63                  input.nextLine();
64                  continue;
65              }
66
67              switch (choice) {
68                  case 1:
69                      try { showSubMenu(input); } catch (Exception e) { errorUtil.handlingError(e); }
70                      break;
71                  case 2:
72                      System.out.println("================== " + "CARI KONSER" + " ==================");
73                      try { cariData(); } catch (Exception e) { errorUtil.handlingError(e); }
74                      break;
75                  case 3:
76                      System.out.println("================ " + "TAMBAH KONSER" + " ================");
77                      try { tambahData(); } catch (Exception e) { errorUtil.handlingError(e); }
78                      break;
79                  case 4:
80                      System.out.println("================ " + "UBAH KONSER" + " ================");
```

```
                    try { ubahData(); } catch (Exception e) { errorUtil.handlingError(e); }
                    break;
                case 5:
                    System.out.println("=============== " + "HAPUS KONSER" + " ===============");
                    try { hapusData(); } catch (Exception e) { errorUtil.handlingError(e); }
                    break;
                case 6:
                    System.out.println(x:"Keluar dari Admin Panel.");
                    lanjut = false;
                    break;
                default:
                    errorUtil.handlingError(pesan:"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:
                break;
            default:
                errorUtil.handlingError(pesan:"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) {
            errorUtil.handlingError(pesan:"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("| %2d | %-8s | %-10s | %-5s | %s\n",
                        no++, f[0], f[1], f[2], f[3]);
            }
        }
    }

    private static void tampilkanTransaksi() throws IOException {
        File file = new File("transaksi.txt");
        if (!file.exists()) {
            System.err.println("Belum ada transaksi yang tercatat.");
            return;
        }
        try (BufferedReader br = new BufferedReader(new FileReader(file))) {
            System.out.println("\n| No | ID Transaksi | User | Konser | Jumlah | Total |");
            System.out.println("-----------------------------------------------------------");
            String line;
            int no = 1;
            while ((line = br.readLine()) != null) {
                String[] f = line.split(",", 6);
                System.out.printf("| %2d | %-12s | %-6s | %-6s | %-6s | %s\n",
                        no++, f[0], f[1], f[2], f[3], f[4]);
            }
        }
    }

    private static void cariData() throws IOException{
        try {
            File file = new File("database.txt");
        } catch (Exception e){
            errorUtil.handlingError("Konser Tidak ditemukan");
            errorUtil.handlingError("Silahkan tambah konser terlebih dahulu");
            return;
        }

        Scanner Input = new Scanner(System.in);
        System.out.print("Masukan konser yang ingin dicari : ");
        String cariString = Input.nextLine();
        String[] keywords = cariString.split("\\s+");
```

```
        cekKonser(keywords);
    }

    private static void cekKonser(String[] keywords) throws IOException{
        BufferedReader bufferInput = null;

        try {
            FileReader fileInput = new FileReader("database.txt");
            bufferInput = new BufferedReader(fileInput);
        } catch (IOException e) {
            errorUtil.handlingError(e);
            return;
        }

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

        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, ",");

                stringToken.nextToken();
                System.out.printf("| %2d ", nomorData);
                System.out.printf("|\t%-20s  ", stringToken.nextToken());
                System.out.printf("|\t%-20s  ", stringToken.nextToken());
                System.out.printf("|\t%-20s  ", stringToken.nextToken());
```

```
                System.out.printf("| %2d ", tiket);
                System.out.printf("|\t%s   ", stringToken.nextToken());
                System.out.print("\n");
            }

        }
        System.out.println("-------------------------------------------------------------------------------------------------------");

    }

    private static void tambahData() throws IOException {
        Scanner input = new Scanner(System.in);
        String nomor, tanggal, konser, lokasi, tiket, harga, deskripsi;

        System.out.print("Masukkan nomor konser: ");
        nomor = input.nextLine();

        tanggal = ambiltanggal();

        System.out.print("Masukkan nama konser: ");
        konser = input.nextLine();

        System.out.print("Masukkan lokasi konser: ");
        lokasi = input.nextLine();

        System.out.print("Masukkan jumlah tiket tersedia: ");
        tiket = input.nextLine();

        System.out.print("Masukkan harga tiket: ");
        harga = input.nextLine();

        System.out.print("Masukkan deskripsi konser: ");
        deskripsi = input.nextLine();

        String dataBaru = nomor + "," + tanggal + "," + konser + "," + lokasi + "," + tiket + "," + deskripsi;
        boolean isExist = false;

        try (BufferedReader reader = new BufferedReader(new FileReader("database.txt"))) {
            String line;
```

```
            while ((line = reader.readLine()) != null) {
                if (line.equalsIgnoreCase(dataBaru)) {
                    isExist = true;
                    break;
                }
            }
        } catch (FileNotFoundException e) { }

        if (!isExist) {
            System.out.println("\nKonser yang akan Anda masukkan adalah:");
            System.out.println("---------------------------------------");
            System.out.println("Nomor     : " + nomor);
            System.out.println("Tanggal   : " + tanggal);
            System.out.println("Konser    : " + konser);
            System.out.println("Lokasi    : " + lokasi);
            System.out.println("Tiket     : " + tiket);
            System.out.println("Harga     : " + harga);
            System.out.println("Deskripsi : " + deskripsi);

            boolean isTambah = getYesorNo("Apakah Anda ingin menambah data konser ini?");

            if (isTambah) {
                // Simpan ke database.txt
                try (BufferedWriter bufferOutput = new BufferedWriter(new FileWriter("database.txt", true))) {
                    bufferOutput.write(dataBaru);
                    bufferOutput.newLine();
                    bufferOutput.flush();
                }

                // Simpan tiket ke tiket.txt
                int jumlahTiket = Integer.parseInt(tiket);
                try (BufferedWriter tiketWriter = new BufferedWriter(new FileWriter("tiket.txt", true))) {
                    for (int i = 1; i <= jumlahTiket; i++) {
                        String idTiket = nomor.replaceAll("\\s+", "") + "_" + tanggal.replaceAll("\\s+", "") + "_T" + i;
                        tiketWriter.write(idTiket + "," + konser + "," + harga + ",tersedia");
                        tiketWriter.newLine();
                    }
                    System.out.println("Tiket berhasil ditambahkan ke tiket.txt sebanyak " + jumlahTiket);
                } catch (IOException e) {
```

```
315                     errorUtil.handlingError(e);
316                 }
317             }
318
319         } else {
320             System.out.println(x:"Konser yang Anda ingin tambahkan sudah ada dalam data.");
321             String[] keywords = {nomor, tanggal, konser, lokasi, tiket, deskripsi};
322             cekKonser(keywords, isDisplay:true);
323         }
324     }
325
326     private static long ambilEntryPertanggal(String nomor, String tanggal) throws IOException {
327         BufferedReader bufferInput = null;
328
329         try {
330             FileReader fileInput = new FileReader(fileName:"database.txt");
331             bufferInput = new BufferedReader(fileInput);
332         } catch (IOException e) {
333             errorUtil.handlingError(e);
334             return 0;
335         }
336
337         long entry = 0;
338         String data = bufferInput.readLine();
339         String nomorBersih = nomor.replaceAll(regex:"\\s+", replacement:"");
340
341         while(data != null){
342             Scanner dataScanner = new Scanner(data);
343             dataScanner.useDelimiter(pattern:"_");
344
345             if (dataScanner.hasNext()) {
346                 String no = dataScanner.next();
347
348                 Scanner refScanner = new Scanner(no);
349                 refScanner.useDelimiter(pattern:"_");
350
351                 try {
352                     String refNomor = refScanner.next();
353                     String refTanggal = refScanner.next();
```

```
                    String refEntry = refScanner.next();

                    if (nomorBersih.equalsIgnoreCase(refNomor) && tanggal.equalsIgnoreCase(refTanggal)) {
                        entry = Long.parseLong(refEntry);
                    }
                } catch (NoSuchElementException | NumberFormatException ex) { }
            }
            data = bufferInput.readLine();
        }
        bufferInput.close();
        return entry;
    }

    private static boolean cekKonser(String[] keywords, boolean isDisplay) throws IOException {
        BufferedReader bufferInput = null;

        try {
            FileReader fileInput = new FileReader(fileName:"database.txt");
            bufferInput = new BufferedReader(fileInput);
        } catch (IOException e) {
            errorUtil.handlingError(e);
            return false;
        }

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

        if (isDisplay) {
            System.out.println(x:"\n| No | Hari/Tanggal | Konser | Lokasi | Tiket Tersedia | Deskripsi |");
            System.out.println(x:"-------------------------------------------------------------------------------");
        }

        while (data != null) {
            boolean isMatch = true;

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

            if (isMatch) {
                found = true;

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

                    String id = st.nextToken();
                    String tanggal = st.nextToken();
                    String konser = st.nextToken();
                    String lokasi = st.nextToken();
                    String sisaTiket = st.nextToken();
                    String deskripsi = st.nextToken();

                    System.out.printf(format:"| %2d | %-14s | %-12s | %-10s | %-15s | %s\n",
                            nomorData, tanggal, konser, lokasi, sisaTiket, deskripsi);
                } else {
                    break;
                }
            }

            data = bufferInput.readLine();
        }

        if (isDisplay) {
            System.out.println(x:"-------------------------------------------------------------------------------");
        }
        bufferInput.close();
        return found;
    }

    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");
        File tempDB = new File(pathname:"tempDB.txt");

        BufferedReader bufferedInput = new BufferedReader(new FileReader(database));
        BufferedWriter bufferedOutput = new BufferedWriter(new FileWriter(tempDB));

        Scanner input = new Scanner(System.in);
        System.out.println(x:"List Konser");
        // tampilkanKonser();

        System.out.print(s:"\nMasukkan nomor konser yang ingin diubah: ");
        int updateNum = input.nextInt();
        input.nextLine();

        String data = bufferedInput.readLine();
```

```
471             int entryCounts = 0;
472
473             while (data != null) {
474                 entryCounts++;
475                 StringTokenizer st = new StringTokenizer(data, delim:",");
476
477                 if (updateNum == entryCounts) {
478                     String ref = st.nextToken();
479                     String nomor = st.nextToken();
480                     String tanggal = st.nextToken();
481                     String konser = st.nextToken();
482                     String lokasi = st.nextToken();
483                     String tiket = st.nextToken();
484                     String deskripsi = st.nextToken();
485
486                     System.out.println(x:"\nData konser yang ingin diubah:");
487                     System.out.println("Nomor     : " + nomor);
488                     System.out.println("Tanggal   : " + tanggal);
489                     System.out.println("Konser    : " + konser);
490                     System.out.println("Lokasi    : " + lokasi);
491                     System.out.println("Tiket     : " + tiket);
492                     System.out.println("Deskripsi : " + deskripsi);
493
494                     String[] updated = new String[6];
495                     String[] fields = {nomor, tanggal, konser, lokasi, tiket, deskripsi};
496                     String[] fieldNames = {"Nomor", "Tanggal", "Konser", "Lokasi", "Tiket", "Deskripsi"};
497
498                     for (int i = 0; i < fields.length; i++) {
499                         if (getYesorNo("Apakah ingin mengubah " + fieldNames[i] + "?")) {
500                             System.out.print("Masukkan " + fieldNames[i] + " baru: ");
501                             if (fieldNames[i].equalsIgnoreCase(anotherString:"Tanggal")) {
502                                 updated[i] = ambiltanggal();
503                             } else {
504                                 updated[i] = input.nextLine();
505                             }
506                         } else {
507                             updated[i] = fields[i];
508                         }
509                     }
```

```
511                 boolean isUpdate = getYesorNo(message:"Yakin ingin menyimpan perubahan?");
512                 if (isUpdate) {
513                     bufferedOutput.write(ref + "," + String.join(delimiter:",", updated));
514                     bufferedOutput.newLine();
515                     hapusTiketBerdasarkanKonser(konser);
516                     try (BufferedWriter tiketWriter = new BufferedWriter(new FileWriter(fileName:"tiket.txt", append:true))) {
517                         int jumlahTiket = Integer.parseInt(updated[4]);
518                         for (int i = 1; i <= jumlahTiket; i++) {
519                             String idTiket = updated[0].replaceAll(regex:"\\s+", replacement:"") + "_" + updated[1].replaceAll(regex:"\\s+", replacement:"") + "_T" + i;
520                             tiketWriter.write(idTiket + "," + updated[2] + ",50000,tersedia");
521                             tiketWriter.newLine();
522                         }
523                     }
524                     System.out.println(x:"Data konser dan tiket berhasil diperbarui.");
525                 } else {
526                     bufferedOutput.write(data);
527                     bufferedOutput.newLine();
528                 }
529             } else {
530                 bufferedOutput.write(data);
531                 bufferedOutput.newLine();
532             }
533             data = bufferedInput.readLine();
534         }
535
536         bufferedOutput.flush();
537         bufferedOutput.close();
538         bufferedInput.close();
539
540         database.delete();
541         tempDB.renameTo(database);
542     }
543
544     private static void hapusData() throws IOException {
545         File database = new File(pathname:"database.txt");
546         File tempDB = new File(pathname:"tempDB.txt");
547
548         BufferedReader bufferedInput = null;
549         BufferedWriter bufferedOutput = null;
```

```
551         try {
552             bufferedInput = new BufferedReader(new FileReader(database));
553             bufferedOutput = new BufferedWriter(new FileWriter(tempDB));
554         } catch (IOException e) {
555             errorUtil.handlingError(e);
556             return;
557         }
558 
559         Scanner input = new Scanner(System.in);
560         System.out.println(x:"List Konser");
561         // tampilkanKonser();
562 
563         System.out.print(s:"\nMasukkan nomor konser yang ingin dihapus: ");
564         int deleteNum = input.nextInt();
565         input.nextLine();
566 
567         boolean isFound = false;
568         int entryCounts = 0;
569         String data = bufferedInput.readLine();
570 
571         while (data != null) {
572             entryCounts++;
573             boolean isDelete = false;
574             StringTokenizer st = new StringTokenizer(data, delim:",");
575 
576             if (deleteNum == entryCounts) {
577                 System.out.println(x:"\nKonser yang ingin Anda hapus adalah:");
578                 System.out.println(x:"---------------------------------");
579                 String ref = st.nextToken();
580                 String nomor = st.nextToken();
581                 String tanggal = st.nextToken();
582                 String konser = st.nextToken();
583                 System.out.println("Referensi  : " + ref);
584                 System.out.println("Nomor      : " + nomor);
585                 System.out.println("Tanggal    : " + tanggal);
586                 System.out.println("Konser     : " + konser);
587                 System.out.println("Lokasi     : " + st.nextToken());
588                 System.out.println("Tiket      : " + st.nextToken());
589                 System.out.println("Deskripsi  : " + st.nextToken());
```

```
591                 isDelete = getYesorNo(message:"Apakah yakin ingin menghapus?");
592                 isFound = true;
593
594                 if (isDelete) {
595                     hapusTiketBerdasarkanKonser(konser);
596                     System.out.println(x:"Data konser dan tiket berhasil dihapus.");
597                     data = bufferedInput.readLine();
598                     continue;
599                 }
600             }
601
602             bufferedOutput.write(data);
603             bufferedOutput.newLine();
604             data = bufferedInput.readLine();
605         }
606
607         if (!isFound) {
608             System.err.println(x:"Konser tidak ditemukan.");
609         }
610
611         bufferedOutput.flush();
612         bufferedOutput.close();
613         bufferedInput.close();
614
615         database.delete();
616         tempDB.renameTo(database);
617     }
618
619     private static void hapusTiketBerdasarkanKonser(String namaKonser) throws IOException {
620         File tiketFile = new File(pathname:"tiket.txt");
621         File tempFile = new File(pathname:"temp_tiket.txt");
622
623         try (BufferedReader reader = new BufferedReader(new FileReader(tiketFile));
624              BufferedWriter writer = new BufferedWriter(new FileWriter(tempFile))) {
625             String line;
626             while ((line = reader.readLine()) != null) {
627                 String[] f = line.split(regex:",");
628                 if (f.length >= 2 && !f[1].equalsIgnoreCase(namaKonser)) {
629                     writer.write(line);
```

```
                    writer.newLine();
                }
            }
        }

        tiketFile.delete();
        tempFile.renameTo(tiketFile);
    }

    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");
        }
    }
}
```