```java
import java.util.ArrayList;
import java.util.InputMismatchException;
import java.util.Scanner;


final class Lapangan {
    private int id;
    private String nama;
    private double harga;

    public Lapangan(int id, String nama, double harga) {
        this.id = id;
        this.nama = nama;
        this.harga = harga;
    }

    public int getId() { return id; }
    public String getNama() { return nama; }
    public double getHarga() { return harga; }

    public void setNama(String nama) { this.nama = nama; }
    public void setHarga(double harga) { this.harga = harga; }


    public final void tampilkanInfo() {
        System.out.println("ID: " + id + ", Nama: " + nama + ", Harga: Rp" + harga + "/jam");
    }
}

// INTERFACE dengan 2 METHOD
interface Status {
    void cetakNota();
    double hitungBiaya();
}

// ABSTRACT CLASS
abstract class Reservasi implements Status {
    private int id;
    private final String namaPelanggan;
    private String tanggal;
    private int durasi;
    private Lapangan lapangan;

    public Reservasi(int id, String namaPelanggan, String tanggal, int durasi, Lapangan lapangan) {
        this.id = id;
        this.namaPelanggan = namaPelanggan;
        this.tanggal = tanggal;
        this.durasi = durasi;
        this.lapangan = lapangan;
    }

    public int getId() { return id; }
    public String getNamaPelanggan() { return namaPelanggan; }
    public String getTanggal() { return tanggal; }
    public int getDurasi() { return durasi; }
    public Lapangan getLapangan() { return lapangan; }

    public void setTanggal(String tanggal) { this.tanggal = tanggal; }
    public void setDurasi(int durasi) { this.durasi = durasi; }

    public void tampilkanInfo() {
        System.out.printf("ID Reservasi: " + id + ", Pelanggan: " + namaPelanggan +
                ", Tanggal: " + tanggal + ", Durasi: " + durasi + " jam, Lapangan: " + lapangan.getNama());
    }

    @Override
    public double hitungBiaya() {
        return durasi * lapangan.getHarga();
    }

    // abstract method dari interface
    @Override
    public abstract void cetakNota();
}

// SUBCLASS yang mengimplementasikan interface
class ReservasiBiasa extends Reservasi {
    public ReservasiBiasa(int id, String namaPelanggan, String tanggal, int durasi, Lapangan lapangan) {
        super(id, namaPelanggan, tanggal, durasi, lapangan);
    }

    @Override
    public void cetakNota() {
        tampilkanInfo();
        System.out.println("Total Biaya: Rp" + hitungBiaya());
    }
}

public class Main {
    static Scanner input = new Scanner(System.in);
    static ArrayList<Lapangan> daftarLapangan = new ArrayList<>();
    static ArrayList<Reservasi> daftarReservasi = new ArrayList<>();
    static int idReservasi = 1;

    // STATIC VARIABLE
    static int totalReservasi = 0;

    public static void main(String[] args) {
        daftarLapangan.add(new Lapangan(1, "Lapangan A", 50000));
        daftarLapangan.add(new Lapangan(2, "Lapangan B", 60000));

        while (true) {
            System.out.println("\n=== Sistem Reservasi Lapangan Futsal ===");
            System.out.println("1. Lihat Daftar Lapangan");
            System.out.println("2. Buat Reservasi");
            System.out.println("3. Lihat Reservasi Saya");
            System.out.println("4. Edit Reservasi");
            System.out.println("5. Hapus Reservasi");
            System.out.println("6. Keluar");
            System.out.print("Pilih menu: ");

            try {
                int pilihan = Integer.parseInt(input.nextLine());

                switch (pilihan) {
                    case 1 -> lihatLapangan();
                    case 2 -> tambahReservasi();
                    case 3 -> lihatReservasi();
                    case 4 -> editReservasi();
                    case 5 -> hapusReservasi();
                    case 6 -> {
                        System.out.println("Terima kasih! Total reservasi dibuat: " + totalReservasi);
                        return;
                    }
                    default -> System.out.println("Pilihan tidak valid.");
                }
            } catch (NumberFormatException e) {
                System.out.println("Input harus berupa angka! Coba lagi.");
            }
        }
    }

    static void lihatLapangan() {
        System.out.println("\nDaftar Lapangan:");
        for (Lapangan l : daftarLapangan) {
            l.tampilkanInfo();
        }
    }

    static void tambahReservasi() {
        try {
            System.out.print("Nama Pelanggan: ");
            String name = input.nextLine();
            System.out.print("Tanggal (YYYY-MM-DD): ");
            String tanggal = input.nextLine();
            System.out.print("Durasi (jam): ");
            int durasi = Integer.parseInt(input.nextLine());

            lihatLapangan();
            System.out.print("Pilih ID Lapangan: ");
            int idLap = Integer.parseInt(input.nextLine());

            Lapangan lapangan = cariLapangan(idLap);
            if (lapangan != null) {
                Reservasi reservasi = new ReservasiBiasa(idReservasi++, name, tanggal, durasi, lapangan);
                daftarReservasi.add(reservasi);
                totalReservasi++; // STATIC VAR digunakan
                System.out.println("Reservasi berhasil ditambahkan!");
            } else {
                System.out.println("Lapangan tidak ditemukan.");
            }
        } catch (NumberFormatException e) {
            System.out.println("Input angka tidak valid! Reservasi dibatalkan.");
        }
    }

    static void lihatReservasi() {
        if (daftarReservasi.isEmpty()) {
            System.out.println("Belum ada reservasi.");
        } else {
            System.out.println("\nDaftar Reservasi:");
            for (Reservasi r : daftarReservasi) {
                r.cetakNota();
                System.out.println("----------------------------------");
            }
        }
    }

    static void editReservasi() {
        lihatReservasi();
        if (daftarReservasi.isEmpty()) return;

        try {
            System.out.print("Masukkan ID reservasi yang ingin diedit: ");
            int id = Integer.parseInt(input.nextLine());

            Reservasi reservasi = cariReservasi(id);
            if (reservasi != null) {
                System.out.print("Tanggal Baru (YYYY-MM-DD): ");
                reservasi.setTanggal(input.nextLine());
                System.out.print("Durasi Baru (jam): ");
                reservasi.setDurasi(Integer.parseInt(input.nextLine()));
                System.out.println("Reservasi berhasil diperbarui!");
            } else {
                System.out.println("Reservasi tidak ditemukan.");
            }
        } catch (NumberFormatException e) {
            System.out.println("Input tidak valid!");
        }
    }

    static void hapusReservasi() {
        lihatReservasi();
        if (daftarReservasi.isEmpty) return;

        try {
            System.out.print("Masukkan ID reservasi yang ingin dihapus: ");
            int id = Integer.parseInt(input.nextLine());
            Reservasi reservasi = cariReservasi(id);
            if (reservasi != null) {
                daftarReservasi.remove(reservasi);
                System.out.println("Reservasi berhasil dihapus!");
            } else {
                System.out.println("Reservasi tidak ditemukan.");
            }
        } catch (NumberFormatException e) {
            System.out.println("Input tidak valid!");
        }
    }

    static Lapangan cariLapangan(int id) {
        for (Lapangan l : daftarLapangan) {
            if (l.getId() == id) return l;
        }
        return null;
    }

    static Reservasi cariReservasi(int id) {
        for (Reservasi r : daftarReservasi) {
            if (r.getId() == id) return r;
        }
        return null;
    }
}
```