

PACKAGE

```
package model;

public class userWarga extends
User {
    public userWarga(int id,
String username, String password
) {
        super
(id, username, password, "warga"
);
    }

    @Override
    public void displayRole() {
        System.out.println(
"Role: Warga - Melaporkan masala
h"
);
    }

    @Override
    public final void iniAbstract
() {
        System.out.println(
"gatau mo ngapain abstarct");
    }
}
```

```
package model;

public class userInstansi extends
    Instansi {
    private String departemen;

    public userInstansi(int id,
        String username, String password
        , String departemen) {
        super
        (id, username, password);
        this.departemen =
        departemen;
    }

    public String getDepartemen
    () {
        return departemen;
    }

    @Override
    public final void displayRole
    () {
        System.out.println(
        "Role: Instansi Pemerintah - Depa
        rtemen: "
        + departemen);
    }
}
```

```
package model;

public class userAdmin extends
User {
    public userAdmin(int id,
String username, String password
) {
        super
(id, username, password, "u");
    }

    @Override
    public void displayRole() {
        System.out.println(
"Role: u - Mengelola sistem");
    }

    @Override
    public void iniAbstract() {
        System.out.println(
"ndak tau mo ngapain");
    }
}
```

```
package model;

public abstract class User {
    private int id;
    private String username;
    private String password;
    private String role;

    public User(int id, String
username, String password, String
role) {
        this.id = id;
        this.username = username;
        this.password = password;
        this.role = role;
    }

    public int getId() {
        return id;
    }

    public String getUsername() {
        return username;
    }

    public String getPassword() {
        return password;
    }

    public String getRole() {
        return role;
    }

    public void displayRole() {
        System.out.println(
"Role: " + role);
    }

    public abstract void
iniAbstract();
}
```

```
package model;

public class LaporanPrioritas
    extends Laporan implements
    intLaporanPrioritas{
    private String prioritas;

    public LaporanPrioritas(
    int id, String deskripsi,
    String status, String
    kategori, int userId, String
    prioritas) {
        super
    (id, deskripsi, status, kateg
    ori, userId);
        this.prioritas =
    prioritas;
    }

    public String
    getPrioritas() {
        return prioritas;
    }

    @Override
    public void display() {
        super.display();
        System.out.println(
    "Prioritas: " + prioritas);
    }
}
```

```
package model;
```

```
public interface
```

```
intLaporanPrioritas {
```

```
    String getPrioritas();
```

```
    void display();
```

```
}
```

```

package model;

public class Laporan {
    private int id;
    private String deskripsi;
    private String status;
    private String kategori;
    private int userId;

    public Laporan(int id, String
deskripsi, String status, String
kategori, int userId) {
        this.id = id;
        this.deskripsi =
deskripsi;
        this.status = status;
        this.kategori = kategori;
        this.userId = userId;
    }

    public int getId() {
        return id;
    }

    public String getDeskripsi
() {
        return deskripsi;
    }

    public String getStatus() {
        return status;
    }

    public void setStatus(String
status) {
        this.status = status;
    }

    public String getKategori() {
        return kategori;
    }

    public int getUserId() {
        return userId;
    }

    public void display() {
        System.out.println(
            "ID: " + id +
            " | Deskripsi: " + deskripsi +
            " | Status: " + status +
            " | Kategori: " + kategori);
    }

    public void display(boolean
showUserId) {
        display();
        if (showUserId) {
            System.out.println(
                "User ID: " + userId);
        }
    }
}

```

```
package model;

public final class Komentar {
    private int idLaporan;
    private final int userId;
    private String isiKomentar;

    public Komentar(int idLaporan
, int userId, String isiKomentar
) {
        this.idLaporan =
idLaporan;
        this.userId = userId;
        this.isiKomentar =
isiKomentar;
    }

    public int getIdLaporan() {
        return idLaporan;
    }

    protected int getUserId() {
        return userId;
    }

    public String getIsiKomentar
() {
        return isiKomentar;
    }

    public void display() {
        System.out.println(
"Laporan ID: " + idLaporan +
" | User ID: " + userId +
" | Komentar: " + isiKomentar);
    }
}
```



```
package model;

public class Instansi extends
    User {
    public Instansi(int id,
String username, String password
    ) {
        super
(id, username, password,
"instansi");
    }

    @Override
    public void displayRole() {
        System.out.println(
"Role: Instansi - Menangani lapor
an"
);
    }

    @Override
    public void iniAbstract() {
        System.out.println(
"ndak tau mo ngapain");
    }
}
```

MAIN

```
private static void deleteReport
() {
    System.out.print(
    "Masukkan ID Laporan yang akan di
hapus: "
);
    int id = getIntInput();
    laporanList.removeIf
(laporan -> laporan.getId() ==
id);
    System.out.println(
    "Laporan berhasil dihapus!");
}

private static void
kelolaKomentar() {
    System.out.println(
    "Daftar Komentar:");
    for (Komentar komentar :
komentarList) {
        komentar.display();
    }
    System.out.println(
    "Kelola Komentar:");
    System.out.println(
    "1. Hapus Komentar");
    System.out.print(
    "Pilih menu: ");
    int pilihan = getIntInput
();
    if (pilihan == 1) {
        deleteComment();
    } else {
        System.out.println(
        "Pilihan tidak valid!");
    }
}

private static void
deleteComment() {
    System.out.print(
    "Masukkan ID Komentar yang akan d
ihapus: "
);
    int id = getIntInput();
    komentarList.removeIf
(komentar -> komentar.
getIdLaporan() == id);
    System.out.println(
    "Komentar berhasil dihapus!");
}
```

```

private static void
tambahKomentar() {
    System.out.print(
        "Masukkan ID Laporan yang dikomen-
        tari: "
    );
    int idLaporan = getIntInput(
    );
    System.out.print(
        "Masukkan Komentar: ");
    String isiKomentar = scanner.
        nextLine();

    komentarList.add(new Komentar
        (idLaporan, currentUser.getId
        (), isiKomentar));
    System.out.println(
        "Komentar berhasil ditambahkan!"
    );
}

private static void
kelolaPengguna() {
    System.out.println(
        "Daftar Pengguna:");
    for (User user : users) {
        System.out.println(
            "ID: " + user.
            getId() + " | Username: " + user.
            getUsername() + " | Role: " +
            user.getRole());
    }
    System.out.println(
        "Kelola Pengguna:");
    System.out.println(
        "1. Hapus Pengguna");
    System.out.print(
        "Pilih menu: ");
    int pilihan = getIntInput();
    if (pilihan == 1) {
        deleteUser();
    } else {
        System.out.println(
            "Pilihan tidak valid!");
    }
}

private static void deleteUser
() {
    System.out.print(
        "Masukkan ID Pengguna yang akan d
        ihapus: "
    );
    int id = getIntInput();
    users.removeIf(user -> user.
        getId() == id);
    System.out.println(
        "Pengguna berhasil dihapus!");
}

private static void kelolaLaporan
() {
    System.out.println(
        "Daftar Laporan:");
    for (Laporan laporan :
        laporanList) {
        laporan.display();
    }
    System.out.println(
        "Kelola Laporan:");
    System.out.println(
        "1. Hapus Laporan");
    System.out.print(
        "Pilih menu: ");
    int pilihan = getIntInput();
    if (pilihan == 1) {
        deleteReport();
    } else {
        System.out.println(
            "Pilihan tidak valid!");
    }
}

```

```

private static void
tampilkanLaporanSendiri() {
    for (Laporan laporan :
        laporanList) {
        if (laporan.getUserId()
            == currentUser.getId()) {
            laporan.display();
        }
    }
}

private static void
hapusLaporanSendiri() {
    System.out.print(
        "Masukkan ID Laporan yang akan di
        hapus: "
    );
    int id = getIntInput();
    laporanList.removeIf
    (laporan -> laporan.getId() ==
        id && laporan.getUserId() ==
        currentUser.getId());
    System.out.println(
        "Laporan berhasil dihapus!");
}

private static void
tampilkanLaporanByKategori() {
    System.out.print(
        "Masukkan Kategori: ");
    String kategori = scanner.
        nextLine();

    for (Laporan laporan :
        laporanList) {
        if (laporan.getKategori
            ().equalsIgnoreCase(kategori)) {
            laporan.display();
        }
    }
}

private static void
updateLaporanStatus() {
    System.out.print(
        "Masukkan ID Laporan yang akan di
        ubah: "
    );
    int id = getIntInput();
    System.out.print(
        "Masukkan Status baru: ");
    String status = scanner.
        nextLine();

    for (Laporan laporan :
        laporanList) {
        if (laporan.getId() ==
            id) {
            laporan.setStatus
            (status);
            System.out.println(
                "Status laporan berhasil diubah!");
        }
        return;
    }
    System.out.println(
        "Laporan tidak ditemukan!");
}

```

```

private static void
handleAdminMenu(int pilihan) {
    switch (pilihan) {
        case 1 -> kelolaPengguna
        ();
        case 2 -> kelolaLaporan
        ();
        case 3 -> kelolaKomentar
        ();
        default -> System.out.
        println("Pilihan tidak valid!");
    }
}

private static int getIntInput
() {
    while (true) {
        try {
            return Integer.
            parseInt(scanner.nextLine());
        } catch (
            NumberFormatException e) {
            System.out.print(
                "Input tidak valid, masukkan angk
                a: "
            );
        }
    }
}

private static void
tambahLaporan() {
    System.out.print(
        "Masukkan ID Laporan: ");
    int id = getIntInput();
    System.out.print(
        "Masukkan Deskripsi: ");
    String deskripsi = scanner.
        nextLine();
    System.out.print(
        "Masukkan Status: ");
    String status = scanner.
        nextLine();
    System.out.print(
        "Masukkan Kategori: ");
    String kategori = scanner.
        nextLine();

    laporanList.add(new Laporan
        (id, deskripsi, status, kategori,
        currentUser.getId()));
    System.out.println(
        "Laporan berhasil ditambahkan!");
}

private static void
tampilkanLaporan() {
    for (Laporan laporan :
        laporanList) {
        laporan.display();
        for (komentar komentar :
            komentarList) {
            if (komentar.
                getIdLaporan() == laporan.getId
                ()) {
                komentar.display
                ();
            }
        }
    }
}

```

```

private static void
displayInstansiMenu() {
    System.out.println(
        "1. Lihat Laporan Berdasarkan Kat
        egori"
    );
    System.out.println(
        "2. Ubah Status Laporan");
    System.out.println(
        "3. Tambah Komentar");
    System.out.println(
        "0. Logout");
    handleMenu(getIntInput());
}

private static void
displayAdminMenu() {
    System.out.println(
        "1. Kelola Pengguna");
    System.out.println(
        "2. Kelola Laporan");
    System.out.println(
        "3. Kelola Komentar");
    System.out.println(
        "0. Logout");
    handleMenu(getIntInput());
}

private static void handleMenu(
    int pilihan) {
    if (pilihan == 0) {
        currentUser = null;
        mainMenu();
        return;
    }

    switch (currentUser.getRole
    ()) {
        case "warga" ->
        handleWargaMenu(pilihan);
        case "instansi" ->
        handleInstansiMenu(pilihan);
        case "admin" ->
        handleAdminMenu(pilihan);
        default -> System.out.
        println("Pilihan tidak valid!");
    }
}

private static void
handleWargaMenu(int pilihan) {
    switch (pilihan) {
        case 1 -> tambahLaporan
        ();
        case 2 ->
        tampilkanLaporan();
        case 3 -> tambahKomentar
        ();
        case 4 ->
        tampilkanLaporanSendiri();
        case 5 ->
        hapusLaporanSendiri();
        default -> System.out.
        println("Pilihan tidak valid!");
    }
}

private static void
handleInstansiMenu(int pilihan) {
    switch (pilihan) {
        case 1 ->
        tampilkanLaporanByKategori();
        case 2 ->
        updateLaporanStatus();
        case 3 -> tambahKomentar
        ();
        default -> System.out.
        println("Pilihan tidak valid!");
    }
}

```

```

"Masukkan Role (warga/instansi/ad
min): "
);
    String role = scanner.
    nextLine();

    int id = users.size() + 1;
    switch (role.toLowerCase
    ()) {
        case "admin" -> users.
        add(new userAdmin
        (id, username, password));
        case "instansi" ->
        users.add(new Instansi
        (id, username, password));
        case "pemerintah" -> {
            System.out.print(
                "Masukkan Departemen: ");
            String departemen
            = scanner.nextLine();
            users.add(new
            userInstansi
            (id, username, password, departem
            en));
        }
        default -> users.add(
        new userWarga
        (id, username, password));
    }
    System.out.println(
        "Registrasi berhasil!");
}

private static void login() {
    System.out.print(
        "Masukkan Username: ");
    String username = scanner.
    nextLine();
    System.out.print(
        "Masukkan Password: ");
    String password = scanner.
    nextLine();

    for (User user : users) {
        if (user.getUsername
        ().equals(username) && user.
        getPassword().equals(password)) {
            currentUser =
            user;
            System.out.println
            ("Login berhasil sebagai " + user
            .getRole());
            menuUtama();
            return;
        }
    }
    System.out.println(
        "Login gagal! Username atau passw
        ord salah."
    );
}

private static void menuUtama
() {
    while (true) {
        System.out.println("\n
        Menu Utama:");
        switch (currentUser.
        getRole()) {
            case "warga" ->
            displayWargaMenu();
            case "instansi" ->

```

```

Komentar> komentarList = new
ArrayList<>();
private static final Scanner
scanner = new Scanner(System.in);
private static User
currentUser = null;

public static final void main
(String[] args) {
    seedUsers();
    mainMenu();
}

private static void seedUsers
() {
    users.add(new userAdmin(1
, "admin", "123"));
    users.add(new Instansi(2
, "instansi", "123"));
    users.add(new
userInstansi(3, "pemerintah",
"123", "Kesehatan"));
    users.add(new userWarga(4
, "warga", "123"));
}

private static void mainMenu
() {
    while (true) {
        System.out.println("
\nMenu Utama:");
        System.out.println(
"1. Login");
        System.out.println(
"2. Daftar");
        System.out.println(
"0. Keluar");
        System.out.print(
"Pilih menu: ");
        int pilihan =
getIntInput();
        switch (pilihan) {
            case 1 -> login
();
            case 2 ->
register();
            case 0 -> {
                System.out.
println(
"Terima kasih telah menggunakan a
plikasi!");
            };
            System.exit(0
);
        }
        default -> System
.out.println(
"Pilihan tidak valid!");
    }
}

private static void register
() {
    System.out.print(
"Masukkan Username: ");
    String username = scanner
.nextLine();
    System.out.print(
"Masukkan Password: ");
    String password = scanner
.nextLine();
    System.out.print(
"Masukkan Role (warga/instansi/ad
min): ");
    String role = scanner.
nextLine();

    int id = users.size() + 1
;
    switch (role.toLowerCase
()) {
        case "admin" -> users
.add(new userAdmin
(id, username, password));
        case "instansi" ->
users.add(new Instansi
(id, username, password));
        case "pemerintah" ->
{
            System.out.print(
"Masukkan Departemen: ");
            String departemen
= scanner.nextLine();
            users.add(new
userInstansi
(id, username, password, departem

```