

LAMPIRAN

Source Code Aplikasi

1. *Source Code EntityBase*

```
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;

public abstract class EntityBase {

    protected SessionFactory factory;
    protected Session session;
    protected Transaction transaction;

    public final void connect() {
        factory = new Configuration().configure().buildSessionFactory();
        session = factory.openSession();
        transaction = session.beginTransaction();
    }

    public final void disconnect() {
        transaction.commit();
        session.close();
    }
}
```

2. *Source Code IDataAccess*

```
import java.io.Serializable;
import java.util.List;
public interface IDataAccess<T> extends Serializable {
    public void insert(T obj);
    public void update(T obj);
    public void delete(T obj);
    public List<T> getAll();
    public List<T> searchAnd(String name, Object value, Object id);
    public T getById(String id);
    public List<T> getByProperty(String name, Object value);
    public List<T> search(String name, Object value);
}
```

3. *Source Code Login*

```
public String loginUkm() {
    for (Iterator it = akunList.iterator(); it.hasNext(); ) {
        akn = (akun) it.next();
        if ((akn.getUserName().equals(akun.getUserName())) &&
(akn.getPassword().equals(akun.getPassword()))) {
            if (akn.getLoginStatus().equals("dinkes")) {
                y = "bisa";
                b = true;
            }
        }
    }
}
```

```

        break;
    } else {
        y = "tdk";
        b = true;
        break;
    }
    } else {
        b = false;
    }
}
if (b == true) {
    if ("bisa".equals(y)) {
        isLogged = true;
        errorLogin = "";

FacesContext.getCurrentInstance().getExternalContext().getSessionMap().put("username",
akun.getUserName());
        return "inner/index.xhtml?faces-redirect=true";
    } else {
        isLogged = true;
        errorLogin = "";

FacesContext.getCurrentInstance().getExternalContext().getSessionMap().put("username",
akun.getUserName());
        return "innerUkm/index.xhtml?faces-redirect=true";
    }
} else {
    errorLogin = "Username dan password tidak sesuai, Harap cek kembali";
    return "LoginAdmin.xhtml";
}
}
}

```

4. *Source Code Logout*

```

public String logoutUkm() {

FacesContext.getCurrentInstance().getExternalContext().getSessionMap().remove("username");
    FacesContext.getCurrentInstance().getExternalContext().invalidateSession();
    akun = null;
    return "index.xhtml?faces-redirect=true";
}

```

5. *Source Code Inisialisasi*

```

@PostConstruct
public void init() {
    ukmList = new ArrayList<ukm>();
    ukmDaftar = new ArrayList<ukm>();
    ukmProfil = new ArrayList<ukm>();
    ukm = new ukm();
    dataAccess = new UkmDataAccess();
    ukmList = dataAccess.getAll();
    ukmProfil = dataAccess.getByProperty("userName", ukm.getUserName());
    lenght = "" + dataAccess.getByProperty("status ukm", "Proses").size(); }

```

6. *Source Code* Tambah Akun

```
public String addAkun() {
    for (Iterator it = akunList.iterator(); it.hasNext(); ) {
        akn = (akun) it.next();
        if (akn.getUserName().equals(akun.getUserName())) {
            akun.setUsernameError("Username sudah digunakan. Masukkan username yang lain.");
            return "RegistrasiUKM";
        }
    }
    akun.setLoginStatus("ukm");
    dataAccess.insert(akun);
    FacesMessage message = new FacesMessage("Data berhasil ditambahkan");
    message.setSeverity((FacesMessage.SEVERITY_INFO));
    FacesContext.getCurrentInstance().addMessage(null, message);
    akunList = dataAccess.getAll();
    return "DaftarUKM";
}
```

7. *Source Code* Ubah Password Admin

```
public String updateAkunDk() {
    akun.setLoginStatus("dinkes");
    dataAccess.update(akun);
    FacesMessage message = new FacesMessage("Data berhasil diubah");
    message.setSeverity((FacesMessage.SEVERITY_INFO));
    FacesContext.getCurrentInstance().addMessage(null, message);
    akunList = dataAccess.getByProperty("user_name", akun.getUserName());
    return "List";
}
```

8. *Source Code* Ubah Password UKM

```
public String updateAkun() {
    akun.setLoginStatus("ukm");
    dataAccess.update(akun);
    FacesMessage message = new FacesMessage("Data berhasil diubah");
    message.setSeverity((FacesMessage.SEVERITY_INFO));
    FacesContext.getCurrentInstance().addMessage(null, message);
    akunList = dataAccess.getByProperty("user_name", akun.getUserName());
    return "List";
}
```

9. *Source Code* Cek Username

```
public void listener(ValueChangeEvent vce) {
    ukm.setUserName(vce.getNewValue().toString());
    if (ukm.getUserName().equals("admindinkes")) {
        t = false;
        b = false;
        setErrorlogin("");
    } else {
        for (Iterator it = ukmList.iterator(); it.hasNext(); ) {
```

```

        uk = (ukm) it.next();
        if (uk.getUserName().equals(ukm.getUserName())) {
            if (uk.getStatusUkm().equals("Proses")) {
                b = false;
                setErrorlogin("Data " + uk.getUserName() + " belum terkonfirmasi");
                System.out.println("Data " + uk.getUserName() + " belum terkonfirmasi");
                t = true;
            } else {
                b = true;
                t = false;
                setErrorlogin("");
                ukm.setIdUkm(uk.getIdUkm());
            }
        }
    }
}
}
}

```

10. Source Code SetList Login

```

public void login() {
    if (false == t) {
        if (true == b) {
            ukmList = dataAccess.getByProperty("user_name", ukm.getUserName());
        } else {
            ukmList = dataAccess.getAll();
        }
    } else if (false == b) {
        setErrorlogin("Username dan password tidak sesuai, Harap cek kembali.");
    }
}
}
}

```

11. Source Code Konfirmasi UKM

```

public String konfirmUkm() {
    ukm.setStatusUkm("Terkonfirmasi");
    dataAccess.update(ukm);
    try {
        MailSender ms = new MailSender();
        ms.sendMailUpd(ukm.getEmail(), ukm.getUserName());
    } catch (Exception e) {
        setErrorText("Error gak bisa kirim gan " + e.toString());
    }
    FacesMessage message = new FacesMessage("Data berhasil diubah");
    message.setSeverity((FacesMessage.SEVERITY_INFO));
    FacesContext.getCurrentInstance().addMessage(null, message);
    ukmList = dataAccess.getAll();
    return "List";
}
}

```

12. Source Code Cek Email (Lupa Password)

```

public void forpassListen(ValueChangeEvent vce) {
    setErrorText("");
}

```

```

for (Iterator it = ukmList.iterator(); it.hasNext(); ) {
    uk = (ukm) it.next();
    if (uk.getEmail().equals(vce.getNewValue().toString())) {
        ukm.setUserName(uk.getUserName());
        l = false;
        setErrorText("");
        break;
    } else {
        l = true;
        setErrorText("Email tidak ditemukan");
    }
}
}
}

```

13. *Source Code Lupa Password*

```

public String forPass(String p) {
    try {
        for (Iterator it = ukmList.iterator(); it.hasNext(); ) {
            uk = (ukm) it.next();
            if (uk.getEmail().equals(ukm.getEmail())) {
                MailSender ms = new MailSender();
                ms.sendMailFP(uk.getUserName(), p, ukm.getEmail());
            }
        }
    } catch (Exception e) {
        System.out.println(e);
    }
    return "RegistrasiUKM";
}

```

14. *Source Code Tampilkan Data di Controller*

```

public String displayAll() {
    ukmList = dataAccess.getAll();
    return "List";
}

public String searchUkm() {
    ukmList = dataAccess.search(this.searchBy, this.searchValue);
    return "List";
}

public String detilUkm() {
    ukmList = dataAccess.getByProperty("user_name", name);
    return "Detil";
}

```

15. *Source Code Tampilkan Data di DataAccess*

```

@Override
public List<ukm> getAll() {
    connect();
    List<ukm> ukmList = session.createQuery("from ukm").list();
}

```

```

        disconnect();
        return ukmList;
    }
}
@Override
public List<ukm> getByProperty(String name, Object value) {
    connect();
    Query query = session.createQuery("from ukm where " + name + " =:value");
    query.setParameter("value", value);
    List<ukm> ukmList = query.list();
    disconnect();
    return ukmList;
}
}

```

16. Source Code Tampil Data Bulan ini

```

// DataAccess
public List<ukm> bulanan() {
    connect();
    Query query = session.createQuery("FROM ukm WHERE
CONCAT(YEAR(tanggalDaftar),'/',MONTH(tanggalDaftar))=CONCAT(YEAR(NOW()),'/',M
ONTH(NOW()))");
    List<ukm> ukmList = query.list();
    disconnect();
    return ukmList;
}
//Controller
public String displayBulan() {
    ukmList = dataAccess.bulanan();
    return "List";
}
}

```

17. Source Code Tampil Data Minggu ini

```

//DataAccess
public List<ukm> mingguan() {
    connect();
    Query query = session.createQuery("FROM ukm WHERE
YEARWEEK(tanggalDaftar)=YEARWEEK(NOW())");
    List<ukm> ukmList = query.list();
    disconnect();
    return ukmList;
}
//Controller
public String displayMinggu() {
    ukmList = dataAccess.mingguan();
    return "List";
}
}

```

18. Source Code Entity.ukm

```

package entity;
import java.util.Date;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

```

```

import javax.faces.application.FacesMessage;
import javax.faces.component.UIComponent;
import javax.faces.context.FacesContext;
import javax.faces.validator.ValidatorException;

/**
 *
 * @author anissa
 */
public class ukm {

    private int idUkm;
    private String noSpkp;
    private String namaLengkap;
    private String namaUsaha;
    private String jenisUsaha;
    private String komoditi;
    private String alamat;
    private int idKec;
    private int idKel;
    private String email;
    private String noHp;
    private String userName;
    private String statusUkm;
    private Date tanggalDaftar;

    public ukm() {

    }

    public ukm(int idUkm, String noSpkp, String namaLengkap, String namaUsaha, String
jenisUsaha, String komoditi, String alamat, int idKec, int idKel, String email, String noHp, String
userName, String statusUkm, Date tanggalDaftar) {
        this.idUkm = idUkm;
        this.noSpkp = noSpkp;
        this.namaLengkap = namaLengkap;
        this.namaUsaha = namaUsaha;
        this.jenisUsaha = jenisUsaha;
        this.komoditi = komoditi;
        this.alamat = alamat;
        this.idKec = idKec;
        this.idKel = idKel;
        this.email = email;
        this.noHp = noHp;
        this.userName = userName;
        this.statusUkm = statusUkm;
        this.tanggalDaftar = tanggalDaftar;
    }

    /**
     * @return the idUkm
     */
    public int getIdUkm() {
        return idUkm;
    }
}

```

```

/**
 * @param idUkm the idUkm to set
 */
public void setIdUkm(int idUkm) {
    this.idUkm = idUkm;
}

/**
 * @return the noSpkp
 */
public String getNoSpkp() {
    return noSpkp;
}

/**
 * @param noSpkp the noSpkp to set
 */
public void setNoSpkp(String noSpkp) {
    this.noSpkp = noSpkp;
}

/**
 * @return the namaLengkap
 */
public String getNamaLengkap() {
    return namaLengkap;
}

/**
 * @param namaLengkap the namaLengkap to set
 */
public void setNamaLengkap(String namaLengkap) {
    this.namaLengkap = namaLengkap;
}

/**
 * @return the namaUsaha
 */
public String getNamaUsaha() {
    return namaUsaha;
}

/**
 * @param namaUsaha the namaUsaha to set
 */
public void setNamaUsaha(String namaUsaha) {
    this.namaUsaha = namaUsaha;
}

/**
 * @return the jenisUsaha
 */
public String getJenisUsaha() {
    return jenisUsaha;
}

```

```

}

/**
 * @param jenisUsaha the jenisUsaha to set
 */
public void setJenisUsaha(String jenisUsaha) {
    this.jenisUsaha = jenisUsaha;
}

/**
 * @return the komoditi
 */
public String getKomoditi() {
    return komoditi;
}

/**
 * @param komoditi the komoditi to set
 */
public void setKomoditi(String komoditi) {
    this.komoditi = komoditi;
}

/**
 * @return the alamat
 */
public String getAlamat() {
    return alamat;
}

/**
 * @param alamat the alamat to set
 */
public void setAlamat(String alamat) {
    this.alamat = alamat;
}

/**
 * @return the idKec
 */
public int getIdKec() {
    return idKec;
}

/**
 * @param idKec the idKec to set
 */
public void setIdKec(int idKec) {
    this.idKec = idKec;
}

/**
 * @return the idKel
 */
public int getIdKel() {

```

```

    return idKel;
}

/**
 * @param idKel the idKel to set
 */
public void setIdKel(int idKel) {
    this.idKel = idKel;
}

/**
 * @return the email
 */
public String getEmail() {
    return email;
}

/**
 * @param email the email to set
 */
public void setEmail(String email) {
    this.email = email;
}

/**
 * @return the noHp
 */
public String getNoHp() {
    return noHp;
}

/**
 * @param noHp the noHp to set
 */
public void setNoHp(String noHp) {
    this.noHp = noHp;
}

/**
 * @return the userName
 */
public String getUserName() {
    return userName;
}

/**
 * @param userName the userName to set
 */
public void setUserName(String userName) {
    this.userName = userName;
}

@Override
public int hashCode() {
    int hash = 3;

```

```

        hash = 23 * hash + this.idUkm;
        return hash;
    }

    @Override
    public boolean equals(Object obj) {
        if (obj == null) {
            return false;
        }
        if (getClass() != obj.getClass()) {
            return false;
        }
        final ukm other = (ukm) obj;
        if (this.idUkm != other.idUkm) {
            return false;
        }
        return true;
    }

    /**
     * @return the statusUkm
     */
    public String getStatusUkm() {
        return statusUkm;
    }

    /**
     * @param statusUkm the statusUkm to set
     */
    public void setStatusUkm(String statusUkm) {
        this.statusUkm = statusUkm;
    }

    /**
     * @return the tanggalDaftar
     */
    public Date getTanggalDaftar() {
        return tanggalDaftar;
    }

    /**
     * @param tanggalDaftar the tanggalDaftar to set
     */
    public void setTanggalDaftar(Date tanggalDaftar) {
        this.tanggalDaftar = tanggalDaftar;
    }

    public void validateName(FacesContext fc, UIComponent c, Object value) {
        if (((String) value).contains("!") || ((String) value).contains("@") || ((String)
value).contains("#") || ((String) value).contains("$")
            || ((String) value).contains("%") || ((String) value).contains("^") || ((String)
value).contains("&") || ((String) value).contains("**")
            || ((String) value).contains("1") || ((String) value).contains("2") || ((String)
value).contains("3") || ((String) value).contains("4"))
    }

```

```

        || ((String) value).contains("5") || ((String) value).contains("6") || ((String)
value).contains("7") || ((String) value).contains("8")
        || ((String) value).contains("9") || ((String) value).contains("0")) {
            throw new ValidatorException(
                new FacesMessage("Nama tidak bisa berisi angka ataupun simbol"));
        }
    }

    public void validateEmail(FacesContext fc, UIComponent c, Object value) throws
ValidatorException {
        String email = (String) value;
        Pattern mask = null;
        mask = Pattern.compile(".*@.*[a-z]+");
        Matcher matcher = mask.matcher(email);
        if (!matcher.matches()) {
            FacesMessage message = new FacesMessage();
            message.setDetail("Invalid e-mail ID");
            message.setSummary("Invalid e-mail ID");
            throw new ValidatorException(message);
        }
    }
}

```

19. Source Code DataAccess.UkmDataAccess

```

package dataaccess;

/**
 *
 * @author anissa
 */
import abstracts.EntityBase;
import abstracts.IDataAccess;
import entity.ukm;
import java.util.List;
import org.hibernate.Query;

public class UkmDataAccess extends EntityBase implements IDataAccess<ukm> {

    public UkmDataAccess() {
    }

    @Override
    public void insert(ukm obj) {
        connect();
        session.save(obj);
        session.flush();
        disconnect();
    }

    @Override
    public void update(ukm obj) {
        connect();
        session.update(obj);
        session.flush();
    }
}

```

```

        disconnect();
    }

    @Override
    public void delete(ukm obj) {
        connect();
        session.delete(obj);
        session.flush();
        disconnect();
    }

    @Override
    public List<ukm> getAll() {
        connect();
        List<ukm> ukmList = session.createQuery("from ukm").list();
        disconnect();
        return ukmList;
    }

    @Override
    public ukm getById(String id) {
        connect();
        List<ukm> ukmList = getByProperty("idUkm", id);
        disconnect();
        if (ukmList != null && ukmList.size() > 0) {
            return ukmList.get(0);
        }
        return null;
    }

    @Override
    public List<ukm> getByProperty(String name, Object value) {
        connect();
        Query query = session.createQuery("from ukm where " + name + " =:value");
        query.setParameter("value", value);
        List<ukm> ukmList = query.list();
        disconnect();
        return ukmList;
    }

    @Override
    public List<ukm> search(String name, Object value) {
        connect();
        Query query = session.createQuery("from ukm where " + name + " like '%" + value + "%'");
        List<ukm> ukmList = query.list();
        disconnect();
        return ukmList;
    }

    @Override
    public List<ukm> searchAnd(String name, Object value, Object id) {
        connect();
        Query query = session.createQuery("from ukm where " + name + " like '%" + value + "%'
and idUkm =" + id);
        List<ukm> ukmList = query.list();
    }

```

```

    disconnect();
    return ukmList;
}

public List<ukm> bulanan() {
    connect();
    Query query = session.createQuery("FROM ukm WHERE
CONCAT(YEAR(tanggalDaftar),'/',MONTH(tanggalDaftar))=CONCAT(YEAR(NOW()),'/',M
ONTH(NOW()))");
    List<ukm> ukmList = query.list();
    disconnect();
    return ukmList;
}

public List<ukm> mingguan() {
    connect();
    Query query = session.createQuery("FROM ukm WHERE
YEARWEEK(tanggalDaftar)=YEARWEEK(NOW())");
    List<ukm> ukmList = query.list();
    disconnect();
    return ukmList;
}

public List<ukm> recent() {
    connect();
    Query query = session.createQuery("FROM ukm where statusUkm = 'Proses' ORDER BY
idUkm DESC");
    List<ukm> ukmList = query.list();
    disconnect();
    return ukmList;
}
}

```