

LAMPIRAN

Lampiran 1 Source Code service

a. *Source Code koneksi.*

```
package goodjob.rest.service;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
public class DBConnection {
    private static final String
DBCLASS =
"com.mysql.jdbc.Driver";
    private static final String DBURL =
"jdbc:mysql://localhost:3306/antrian?
allowMultiQueries=true";
    private static final String DBUSER
= "root";
    private static final String DBPWD =
"aku12345";
    @SuppressWarnings("finally")
    public static Connection
createConnection() throws Exception
{
    Connection con = null;
    try {
        Class.forName(DBCLASS);
        con =
DriverManager.getConnection(DBUR
L, DBUSER, DBPWD);
    } catch
(ClassNotFoundException |
SQLException e) {
        throw e;
    } finally {
        return con;
    }
}
    public static boolean
checkLogin(String uname, String
pwd) throws Exception {
    boolean isUserAvailable = false;
    Connection dbConn = null;
    try {
        try {
            dbConn =
DBConnection.createConnection();
```

```
        } catch (Exception e) {
        }
        Statement stmt =
dbConn.createStatement();
        String query = "SELECT *
FROM antrian.users WHERE
Username = '" + uname
+ "' AND Passwords = " +
"'" + pwd + "'";
        //System.out.println(query);
        ResultSet rs =
stmt.executeQuery(query);
        while (rs.next()) {
            //System.out.println(rs.getString(1) +
rs.getString(2) + rs.getString(3));
            isUserAvailable = true;
        }
    } catch (SQLException sqle) {
        throw sqle;
    } catch (Exception e) {
        // TODO Auto-generated catch
block
        if (dbConn != null) {
            dbConn.close();
        }
        throw e;
    } finally {
    } finally {
        if (dbConn != null) {
            dbConn.close();
        }
    }
    return isUserAvailable;
}
}
```

b. *source code service user*

```
package goodjob.rest.service.user;
import goodjob.rest.service.EntityBase;
import
goodjob.rest.service.user.UserService;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.UUID;
import javax.ws.rs.Consumes;
import javax.ws.rs.GET;
import javax.ws.rs.POST;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
import javax.ws.rs.Produces;
```

```

import javax.ws.rs.core.MediaType;
import
goodjob.rest.service.exceptions.NotFound
Exception;
import
goodjob.rest.service.pengguna.Pengguna
Service;
import java.net.URI;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.ws.rs.DELETE;
import javax.ws.rs.PUT;
import javax.ws.rs.core.Response;
import
org.springframework.transaction.annotati
on.Transactional;

@Path("/user")
@Produces(MediaType.APPLICATION_
JSON)
@Consumes(MediaType.APPLICATION
_JSON)
public class UserService extends
EntityBase implements
IUserDataAccess<User>{
    @POST
    @Transactional
    @Override
    public Response create(User entity) {
        try {
            con = connect();
            String query = "INSERT INTO
antrian.users (Name, DateOfBirth, Adress,
Gender, PhoneNumber, Username, Email,
Passwords, UserType, Gcm) VALUES (?,
?, ?, ?, ?, ?, ?, ?, ?)";
            PreparedStatement stmt =
con.prepareStatement(query);
            stmt.setString(1,
entity.getName());
            stmt.setString(2,
entity.getDateOfBirth());
            stmt.setString(3,
entity.getAddress());
            stmt.setString(4,
entity.getGender());
            stmt.setString(5,
entity.getPhoneNumber());
            stmt.setString(6,
entity.getUsername());

```

```

            stmt.setString(7,
entity.getEmail());
            stmt.setString(8,
entity.getPassword());
            stmt.setString(9,
entity.getUserType());
            stmt.setString(10,
entity.getGcm());
            //System.out.println(query);
            int rs = stmt.executeUpdate();
            if (rs != 0) {
                return
Response.created(URI.create(entity.getNa
me())).build();
            } else {
                return
Response.status(Response.Status.EXPEC
TATION_FAILED).build();
            }
        } catch (SQLException ex) {

            Logger.getLogger(PenggunaService.class
.getName()).log(Level.SEVERE, null,
ex);
            return
Response.status(Response.Status.NOT_F
OUND).build();
        }
    }
    @GET
    @Path("login/{username}/{usertype}")
    @Transactional
    @Override
    public User
findByUsername(@PathParam("usernam
e") String username,
@PathParam("usertype") String usertype)
    {
        User user = new User();
        try {
            con = connect();
            String query = "SELECT * FROM
antrian.users Where Username = ? and
UserType = ?";
            PreparedStatement stmt =
con.prepareStatement(query);
            stmt.setString(1, username);
            stmt.setString(2, usertype);
            ResultSet rs =
stmt.executeQuery();
            while (rs.next()) {
                user.setUserId(rs.getInt(1));
                user.setName(rs.getString(2));

```

```

user.setDateOfBirth(rs.getString(3));

user.setAddress(rs.getString(4));
    user.setGender(rs.getString(5));

user.setPhoneNumber(rs.getString(6));

user.setUsername(rs.getString(7));
    user.setEmail(rs.getString(8));

user.setPassword(rs.getString(9));

user.setUserType(rs.getString(10));
    user.setGcm(rs.getString(11));
    }
    } catch (SQLException ex) {

Logger.getLogger(UserService.class.getName()).log(Level.SEVERE, null, ex);
    }
    disconnect();
    return user;
    }
    @PUT
    @Path("/{userId}/{gcm}")
    @Transactional
    @Override
    public Response edit(@PathParam("userId") Integer userId, @PathParam("gcm") String gcm) {
        try {
            con = connect();
            String query = "UPDATE antrian.users SET Gcm = ? WHERE UserId = ?";
            PreparedStatement stmt = con.prepareStatement(query);
            stmt.setString(1, gcm);
            stmt.setInt(2, userId);
            //System.out.println(query);
            int rs = stmt.executeUpdate();
            if (rs != 0) {
                disconnect();
                return Response.created(URI.create("")).entity(rs).build();
            } else {
                disconnect();
                return Response.status(Response.Status.EXPECTATION_FAILED).build();
            }
        } catch (SQLException ex) {
            disconnect();

```

```

Logger.getLogger(UserService.class.getName()).log(Level.SEVERE, null, ex);
        return Response.status(Response.Status.NOT_FOUND).build();
    }
    }

    @Override
    public Response edit(User entity) {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
    }

    @Override
    public Response remove(Integer id) {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
    }

    @Override
    public String count() {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
    }

    @Override
    public User find(Integer id) {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
    }

    @Override
    public User findByEmail(String email)
    {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
    }
    }

```

c. *source code service antrian*

```
package goodjob.rest.service.queue;
import goodjob.rest.service.EntityBase;
import goodjob.rest.service.user.User;
import
goodjob.rest.service.user.UserService;
import goodjob.rest.service.user.User;
import java.net.URI;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.Calendar;
import java.sql.Date;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.time.LocalDate;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.ws.rs.Consumes;
import javax.ws.rs.DELETE;
import javax.ws.rs.GET;
import javax.ws.rs.POST;
import javax.ws.rs.PUT;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
import javax.ws.rs.Produces;
import javax.ws.rs.core.MediaType;
import javax.ws.rs.core.Response;
import
org.springframework.transaction.annotation.Transactional;
```

```
@Path("/queue")
@Produces(MediaType.APPLICATION_
JSON)
@Consumes(MediaType.APPLICATION
_JSON)
public class AntrianUserService extends
EntityBase implements
IAntrianUserService<AntrianUser> {

    public AntrianUserService() {
    }
    @POST
    @Transactional
    @Override
    public Response create(AntrianUser
entity) {
        try {
            con = connect();
            String query = "INSERT INTO
antrian.antrianuser (TanggalAntrian,
Statuss, NomorAntrian, UserId) VALUES
(?, ?,?,?)";
            PreparedStatement stmt =
con.prepareStatement(query,
Statement.RETURN_GENERATED_KE
YS);
            stmt.setString(1,
entity.getTanggalAntrian());
            stmt.setString(2,
entity.getStatuss());
            stmt.setInt(3, getLastAntrian());
            stmt.setInt(4, entity.getUserId());
            System.out.println(query);
            int rs = stmt.executeUpdate();
            if (rs != 0) {
                ResultSet result =
stmt.getGeneratedKeys();
```

```

        result.next();
        disconnect();
        return
Response.created(URI.create(String.value
Of(result.getInt(1)))));build();
    } else {
        disconnect();
        return
Response.status(Response.Status.EXPEC
TATION_FAILED).build();
    }
    } catch (SQLException ex) {
        disconnect();

Logger.getLogger(AntrianUserService.cla
ss.getName()).log(Level.SEVERE, null,
ex);

        return
Response.status(Response.Status.NOT_F
OUND).build();
    }
}
private int getLastQueueNumber() {
    int x = 0;
    try {
        con = connect();
        String query = "select
max(NomorAntrian) as NomorAntrian
from antrian.antrianuser";
        PreparedStatement stmt =
con.prepareStatement(query);
        ResultSet rs =
stmt.executeQuery();
        Date Tanggalsekarang = null;
        Calendar c =
Calendar.getInstance();
        c.setTime(Tanggalsekarang);

```

```

        c.add(Calendar.DATE, 1);
        Tanggalsekarang = new
java.sql.Date(c.getTimeInMillis());
        if (rs.next()) {
            x = rs.getInt("NomorAntrian");
            return x + 1;
        } else {
            return x = 1;
        }
    } catch (SQLException ex) {

Logger.getLogger(AntrianUserService.cla
ss.getName()).log(Level.SEVERE, null,
ex);
    }
    return x;
}
private int getLastAntrian() {
    int id=1;
    try {
        con = connect();
        String query = "select
NomorAntrian from antrian.antrianuser
WHERE TanggalAntrian =
DATE(NOW()) ORDER BY
NomorAntrian DESC LIMIT 1";
        PreparedStatement stmt =
con.prepareStatement(query);
        ResultSet resultSet =
stmt.executeQuery();
        while (resultSet.next()) {
            int a = resultSet.getInt(1)+1;
            if(a>1)
            {
                id = a;
            }
        }
    }
}

```

```

    } catch (SQLException ex) {

Logger.getLogger(AntrianUser.class.getName()).log(Level.SEVERE, null, ex);
    }
    disconnect();
    return id;
}
@Override
public Response edit(AntrianUser entity) {
    throw new
UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
}
@Override
public Response remove(Integer id) {
    throw new
UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
}
@GET
@Path("/{id}")
@Transactional
@Override
public AntrianUser find(@PathParam("id") Integer id) {
    AntrianUser queue = new
AntrianUser();
    try {
        con = connect();
        String query = "SELECT * FROM
antrian.antrianuser Where UserId = ?";

```

```

        PreparedStatement stmt =
con.prepareStatement(query);
        stmt.setInt(1, id);
        ResultSet rs =
stmt.executeQuery();
        while (rs.next()) {

queue.setAntrianId(rs.getInt(1));

queue.setTanggalAntrian(rs.getString(2));

queue.setStatuss(rs.getString(3));

queue.setNomorAntrian(rs.getInt(4));
            queue.setUserId(rs.getInt(5));
        }
    } catch (SQLException ex) {

Logger.getLogger(AntrianUser.class.getName()).log(Level.SEVERE, null, ex);
    }
    disconnect();
    return queue;
}
public List<AntrianUser> findAll() {
    return null;
}
@Override
public List<AntrianUser> findRange(Integer max, Integer first) {
    throw new
UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
}
@Override

```

```

public String count() {
    throw new
UnsupportedOperationException("Not
supported yet."); //To change body of
generated methods, choose Tools |
Templates.
}
@Override
public List<AntrianUser> find(Integer
doctorId, Date date) {
    throw new
UnsupportedOperationException("Not
supported yet."); //To change body of
generated methods, choose Tools |
Templates.
}
@Override
public AntrianUser findPatient(Integer
QueueNumber) {
    throw new
UnsupportedOperationException("Not
supported yet."); //To change body of
generated methods, choose Tools |
Templates.
}
@Override
public AntrianUser
findPatientQueue(Integer patientId,
Integer doctorId, Date datetaken) {
    throw new
UnsupportedOperationException("Not
supported yet."); //To change body of
generated methods, choose Tools |
Templates.
}
@GET
@Path("last/{TanggalAntrian}")

```

```

@Transactional
@Override
public AntrianUser
AmbilNomorAntrian(@PathParam("Tang
galAntrian") String tanggalAntrian) {
    AntrianUser queue = new
AntrianUser();
    try {
        con = connect();
        String query = "SELECT * FROM
antrian.antrianuser Where TanggalAntrian
>= ? ORDER BY NomorAntrian DESC
LIMIT 1";
        PreparedStatement stmt =
con.prepareStatement(query);
        stmt.setString(1, tanggalAntrian);
        ResultSet rs =
stmt.executeQuery();
        while (rs.next()) {
            queue.setAntrianId(rs.getInt(1));
            queue.setTanggalAntrian(rs.getString(2));
            queue.setStatuss(rs.getString(3));
            queue.setNomorAntrian(rs.getInt(4));
            queue.setUserId(rs.getInt(5));
        }
    } catch (SQLException ex) {
        Logger.getLogger(AntrianUserService.class.getName()).log(Level.SEVERE, null, ex);
    }
    disconnect();
    return queue;
}

```

```

    }
    @GET

    @Path("count/{TanggalAntrian}/{Statuss}")

    @Produces(MediaType.TEXT_PLAIN)
    @Transactional
    @Override
    public String
    count(@PathParam("TanggalAntrian")
    Date
    tanggalAntrian,@PathParam("Statuss")
    String statuss) {
        String count = "";
        try {
            con = connect();
            String query = "SELECT count(*)
FROM antrian.antrianuser Where
TanggalAntrian >= ? AND
TanggalAntrian <= ? AND Statuss = ?";
            PreparedStatement stmt =
con.prepareStatement(query);
            stmt.setObject(1, tanggalAntrian);
            Calendar c =
Calendar.getInstance();
            c.setTime(tanggalAntrian);
            c.add(Calendar.DATE, 1);
            tanggalAntrian= new
java.sql.Date(c.getTimeInMillis());
            stmt.setObject(2, tanggalAntrian);
            stmt.setString(3, statuss);
            ResultSet rs =
stmt.executeQuery();
            while (rs.next()) {
                count =
String.valueOf(rs.getInt(1));

```

```

    }
    } catch (SQLException ex) {

    Logger.getLogger(AntrianUserService.class.getName()).log(Level.SEVERE, null,
ex);
    }
    disconnect();
    return count;
    }
    @GET

    @Path("list/{TanggalAntrian}/{Statuss}")
    )
    @Transactional
    @Override
    public List<AntrianUser>
find(@PathParam("TanggalAntrian")
Date
tanggalAntrian,@PathParam("Statuss")
String statuss) {
        List<AntrianUser> listAntrian = new
ArrayList<>();
        try {
            con = connect();
            String query = "SELECT * FROM
antrian.antrianuser Where TanggalAntrian
>= ? AND TanggalAntrian <= ? AND
Statuss = ? ORDER BY TanggalAntrian
DESC";
            PreparedStatement stmt =
con.prepareStatement(query);
            stmt.setObject(1, tanggalAntrian);
            Calendar c =
Calendar.getInstance();
            c.setTime(tanggalAntrian);
            c.add(Calendar.DATE, 1);

```



```

        tanggalAntrian= new
        java.sql.Date(c.getTimeInMillis());
        stmt.setObject(2, tanggalAntrian);
        stmt.setString(3, status);
        ResultSet rs =
        stmt.executeQuery();
        while (rs.next()) {
            AntrianUser queue = new
            AntrianUser();

            queue.setAntrianId(rs.getInt(1));

            queue.setTanggalAntrian(rs.getString(2));

            queue.setStatus(rs.getString(3));

            queue.setNomorAntrian(rs.getInt(4));
            queue.setUserId(rs.getInt(5));
            listAntrian.add(queue);
        }
    } catch (SQLException ex) {

        Logger.getLogger(AntrianUser.class.getName()).log(Level.SEVERE, null, ex);
    }
    disconnect();
    return listAntrian;
}

@GET
@Path("count/{TanggalAntrian}")

@Produces(MediaType.TEXT_PLAIN)
@Transactional
@Override
public String
count(@PathParam("TanggalAntrian")
Date tanggalAntrian) {

```

```

        String count = "";
        try {
            con = connect();
            String query = "SELECT count(*)
            FROM antrian.antrianuser Where
            TanggalAntrian >= ? AND
            TanggalAntrian <= ?";
            PreparedStatement stmt =
            con.prepareStatement(query);
            stmt.setObject(1, tanggalAntrian);
            Calendar c =
            Calendar.getInstance();
            c.setTime(tanggalAntrian);
            c.add(Calendar.DATE, 1);
            tanggalAntrian= new
            java.sql.Date(c.getTimeInMillis());
            stmt.setObject(2, tanggalAntrian);
            ResultSet rs =
            stmt.executeQuery();
            while (rs.next()) {
                count =
                String.valueOf(rs.getInt(1));
            }
        } catch (SQLException ex) {

            Logger.getLogger(AntrianUserService.class.getName()).log(Level.SEVERE, null,
            ex);
        }
        disconnect();
        return count;
    }

    @GET
    @Path("count/{TanggalAntrian}/{Statuss
    }/{NomorAntrian}")

```

```

@Produces(MediaType.TEXT_PLAIN)
@Transactional
@Override
public String
count(@PathParam("TanggalAntrian")
Date
tanggalAntrian,@PathParam("Statuss")
String
statuss,@PathParam("NomorAntrian")
Integer nomorAntrian) {
    String count = "";
    try {
        con = connect();
        String query = "SELECT count(*)
FROM antrian.antrianuser Where
TanggalAntrian >= ? AND
TanggalAntrian <= ? AND Statuss = ?
AND NomorAntrian < ? ";
        PreparedStatement stmt =
con.prepareStatement(query);
        stmt.setObject(1, tanggalAntrian);
        Calendar c =
Calendar.getInstance();
        c.setTime(tanggalAntrian);
        c.add(Calendar.DATE, 1);
        tanggalAntrian= new
java.sql.Date(c.getTimeInMillis());
        stmt.setObject(2, tanggalAntrian);
        stmt.setString(3, statuss);
        stmt.setInt(4, nomorAntrian);
        ResultSet rs =
stmt.executeQuery();
        while (rs.next()) {
            count =
String.valueOf(rs.getInt(1));
        }
    }
}

```

```

    } catch (SQLException ex) {

Logger.getLogger(AntrianUser.class.getN
ame()).log(Level.SEVERE, null, ex);
    }
    disconnect();
    return count;
}
@GET
@Path("next/{TanggalAntrian}")
@Transactional
@Override
public AntrianUser
nextAntrian(@PathParam("TanggalAntria
n") Date tanggalAntrian) {
    AntrianUser queue = new
AntrianUser();
    try {
        con = connect();
        String query = "SELECT * FROM
antrian.antrianuser Where TanggalAntrian
= ? AND Statuss = ? Order By
NomorAntrian ASC LIMIT 1 ";
        PreparedStatement stmt =
con.prepareStatement(query);
        stmt.setObject(1, tanggalAntrian);
        Calendar c =
Calendar.getInstance();
        c.setTime(tanggalAntrian);
        c.add(Calendar.DATE, 1);
        Date date = new
java.sql.Date(c.getTimeInMillis());
        stmt.setString(2, "Menunggu");
        ResultSet rs =
stmt.executeQuery();
        while (rs.next()) {

```

```

queue.setAntrianId(rs.getInt(1));

queue.setTanggalAntrian(rs.getString(2));

queue.setStatus(rs.getString(3));

queue.setNomorAntrian(rs.getInt(4));
    queue.setUserid(rs.getInt(5));
    }
    } catch (SQLException ex) {

Logger.getLogger(AntrianUserService.class.getName()).log(Level.SEVERE, null, ex);
    }
    List<AntrianUser> listQueue =
find(tanggalAntrian, "Dilayani");
    if(!listQueue.isEmpty())

updateStatusUser(listQueue.get(0).getAntrianId(), "Selesai");
    if(queue.getAntrianId() != null)

updateStatusUser(queue.getAntrianId(),
"Dilayani");
    disconnect();
    return queue;
    }
    private Response
updateStatusUser(Integer antrianId, String
status) {
    try {
        con = connect();
        String query = "UPDATE
antrian.antrianuser SET Status = ?
WHERE AntrianId = ?";

```

```

PreparedStatement stmt =
con.prepareStatement(query);
    stmt.setString(1, status);
    stmt.setInt(2, antrianId);
    //System.out.println(query);
    int rs = stmt.executeUpdate();
    if (rs != 0) {
        disconnect();
        return
Response.created(URI.create(antrianId.to
String())).build();
    } else {
        disconnect();
        return
Response.status(Response.Status.EXPEC
TATION_FAILED).build();
    }
    } catch (SQLException ex) {
        disconnect();

Logger.getLogger(AntrianUserService.class.getName()).log(Level.SEVERE, null, ex);
        return
Response.status(Response.Status.NOT_F
OUND).build();
    }
    }
    @PUT
    @Path("batalAntrian/{id}")
    @Transactional
    @Override
    public Response
batalAntrian(@PathParam("id") Integer
userId)
    {
        try {

```

```

        con = connect();
        String query = "UPDATE
antrian.antrianuser SET Statuss = 'Batal'
WHERE UserId = ?";
        PreparedStatement stmt =
con.prepareStatement(query);
        // stmt.setString(1, "Batal");
        stmt.setInt(1, userId);
        //System.out.println(query);
        int rs = stmt.executeUpdate();
        if (rs != 0) {
            disconnect();
            return
Response.created(URI.create(userId.toStr
ing())).build();
        } else {
            disconnect();
            return
Response.status(Response.Status.EXPEC
TATION_FAILED).build();
        }
    } catch (SQLException ex) {
        disconnect();

Logger.getLogger(AntrianUserService.cla
ss.getName()).log(Level.SEVERE, null,
ex);

        return
Response.status(Response.Status.NOT_F
OUND).build();
    }
}
@Override
public AntrianUser
specialCallPatientQueue(Integer doctorId,
Date datetaken, Integer queueNumber) {

```

```

        throw new
UnsupportedOperationException("Not
supported yet."); //To change body of
generated methods, choose Tools |
Templates.
    }
    @GET
    @Path("data/{Statuss}")
    @Transactional
    @Override
    public AntrianUser
data(@PathParam("Statuss") String
Statuss) {
        AntrianUser queue = new
AntrianUser();
        try {
            con = connect();
            String query = "SELECT * FROM
antrian.antrianuser Where Statuss = ?";
            PreparedStatement stmt =
con.prepareStatement(query);

            stmt.setString(1, Statuss);
            ResultSet rs =
stmt.executeQuery();
            while (rs.next()) {

queue.setAntrianId(rs.getInt(1));

queue.setTanggalAntrian(rs.getString(2));

queue.setStatuss(rs.getString(3));

queue.setNomorAntrian(rs.getInt(4));
                queue.setUserId(rs.getInt(5));
            }
        } catch (SQLException ex) {

```

```

Logger.getLogger(UserService.class.getName()).log(Level.SEVERE, null, ex);
    }
    disconnect();
    return queue;
}
}
}
}

```

Lampiran 2 Source Code aplikasi

a. source code aactivity login

```

<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:orientation="vertical"

android:layout_width="match_parent"

android:layout_height="wrap_content"

android:id="@+id/parent_login_layout">
    <ScrollView
android:layout_width="match_parent"

android:layout_height="wrap_content">
        <LinearLayout
android:layout_width="match_parent"

android:layout_height="wrap_

```

```

content"

android:orientation="vertical"

android:weightSum="1">
    <FrameLayout
xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/doctor_profile_layout"

android:layout_width="match_parent"

android:layout_height="166dp">
    <LinearLayout
android:orientation="vertical"

android:layout_width="match_parent"

android:layout_height="225dp"

android:layout_gravity="left|top">
        <ImageView
android:id="@+id/imgBackground"

android:layout_width="match_parent"

android:layout_height="210dp"

android:scaleType="centerCrop"

android:src="@drawable/boediman"/>
    </LinearLayout>

```

```

        </FrameLayout>
        <LinearLayout

android:layout_width="match_
parent"

android:layout_height="match
parent"

android:elevation="2dp"

android:background="@color/c
olorBackground"

android:orientation="horizon
tal">
<LinearLayout

android:layout_width="0dp"

android:layout_weight="2.59"

android:layout_height="380dp
"

android:layout_marginTop="10
dp"

android:layout_marginBottom=
"50dp"

android:theme="@style/TextLa
rgeBlue">
<RelativeLayout

android:layout_width="fill_p
arent"

android:layout_height="wrap_
content"

android:layout_below="@+id/l
ogin_title"

android:layout_marginLeft="1
5dp"

android:layout_marginRight="
15dp"

android:layout_marginTop="20
dp"

```

```

android:background="#fff"

android:elevation="4dp"

android:orientation="vertica
l"

android:padding="20dp">
<LinearLayout

android:layout_width="fill_p
arent"

android:layout_height="wrap_
content"

android:orientation="vertica
l"

android:paddingTop="30dp">
<android.support.design.widg
et.TextInputLayout

android:layout_width="match_
parent"

android:layout_height="wrap_
content">
<EditText

android:layout_width="fill_p
arent"

android:layout_height="wrap_
content"

android:id="@+id/username"

android:drawableTint="#00000
0"

android:singleLine="true"

android:hint="Username"

android:inputType="textEmail
Address"

android:textSize="15dp"

android:textColor="#000000"
/>
</android.support.design.wid
get.TextInputLayout>

```

```

<android.support.design.widget.TextInputLayout

android:layout_width="match_
parent"

android:layout_height="wrap_
content">

<EditText

android:id="@+id/password"

android:layout_width="match_
parent"

android:layout_height="wrap_
content"

android:hint="@string/Passwo
rd"

android:imeActionId="@+id/lo
gin"

android:imeActionLabel="@str
ing/Password"

android:imeOptions="actionUn
specified"

android:inputType="textPassw
ord"

android:maxLines="1"

android:singleLine="true"

android:textSize="15dp"

android:textColor="#000000"
/>

</android.support.design.wid
get.TextInputLayout>

<CheckBox

android:id="@+id/checkBox1"

android:layout_width="wrap_c
ontent"

```

```

android:layout_height="wrap_
content"

android:layout_alignParentRi
ght="true"

android:layout_below="@+id/e
ditText1"

android:layout_marginRight="
20dp"

android:layout_marginTop="14
dp"

android:textSize="15dp"

android:text="Show Password"

android:textColor="#020202"
/>

<Button

android:id="@+id/btn_login"

android:layout_width="fill_p
arent"

android:layout_height="wrap_
content"

android:layout_margin="30dp"

android:background="@drawabl
e/button_primary"

android:text="Masuk"

android:textAllCaps="false"

android:textColor="#fff"

android:textSize="18sp"    />

<TextView

android:id="@+id/btnLinkToRe
gisterScreen"

android:layout_width="fill_p
arent"

android:layout_height="wrap_

```

```

content"

android:text="@string/link_t
o_reg"

android:gravity="center"

android:textSize="20sp"

android:textColor="#025f7c"

android:showAsAction="ifRoom
"/>

</LinearLayout>

</RelativeLayout>

</LinearLayout>
    </LinearLayout>
        </LinearLayout>
            </ScrollView>
        </LinearLayout>

```

b. source code activity menu

antrian

```

<LinearLayout

xmlns:android="http://schema
s.android.com/apk/res/androi
d"

xmlns:tools="http://schemas.
android.com/tools"

android:orientation="vertica
l"

android:layout_width="match_
parent"

android:layout_height="wrap_
content"

android:id="@+id/parentLayou
tQueue">
    <ScrollView

android:layout_width="match_
parent"

```

```

android:layout_height="wrap_
content"></ScrollView>
        <LinearLayout

android:layout_width="match_
parent"

android:layout_height="wrap_
content"

android:orientation="vertica
l">
            <FrameLayout

xmlns:android="http://schema
s.android.com/apk/res/androi
d"

xmlns:app="http://schemas.an
droid.com/apk/res-auto"

android:id="@+id/doctor_prof
ile_layout"

android:layout_width="match_
parent"

android:layout_height="343dp
">
                <LinearLayout

android:orientation="vertica
l"

android:layout_width="match_
parent"

android:layout_height="160dp
"

android:layout_gravity="left
|top">
                    <ImageView

android:id="@+id/imgBackgrou
nd"

android:layout_width="match_
parent"

android:layout_height="160dp
"

android:scaleType="centerCro
p"

```



```

android:src="@drawable/boedi
man" />
        </LinearLayout>
        <RelativeLayout
android:id="@+id/relativeLay
out"

android:layout_width="match_
parent"

android:layout_height="200dp
"

android:paddingBottom="1dp"

android:elevation="2dp"

android:background="@drawabl
e/layout_background"

android:layout_alignParentBo
ttom="true"

android:layout_alignParentLe
ft="true"

android:layout_alignParentSt
art="true"

android:layout_marginBottom=
"17dp"

android:layout_gravity="left
|bottom">
<LinearLayout

android:layout_width="match_
parent"

android:layout_height="600dp
"

android:orientation="vertica
l">
        <TextView

android:layout_width="match_
parent"

android:layout_height="wrap_
content"

android:background="@color/c

```

```

olorPrimary"

android:text="@string/Noantr
ian"

android:textColor="@android:
color/white"

android:gravity="center"

android:textStyle="bold"

style="@style/TextAppearance
.AppCompat.Medium"

android:textSize="30dp"

android:textIsSelectable="tr
ue" />
<LinearLayout

android:layout_width="match_
parent"

android:layout_height="50dp"

android:layout_weight="2"

android:layout_margin="5dp"

android:background="@color/c
olorBackground">
<TextView

android:id="@+id/nomorantria
n"

android:layout_width="match_
parent"

android:layout_height="match_
parent"

android:text="A009"

android:gravity="center"

android:textSize="49sp"

android:textIsSelectable="tr
ue" />
</LinearLayout>

```

```

<LinearLayout
    android:layout_width="match_
parent"

    android:layout_height="wrap_
content"

    android:orientation="horizon
tal">

<LinearLayout
    android:layout_width="0dp"

    android:layout_weight="2"

    android:layout_height="match
parent"

    android:layout_margin="5dp"

    android:elevation="2dp"

    android:background="@drawabl
e/layout_background"

    android:orientation="horizon
tal">

<LinearLayout
    android:layout_width="0dp"

    android:layout_weight="2"

    android:layout_height="match
parent"

    android:background="@color/c
olorPrimary">

<TextView
    android:layout_width="match_
parent"

    android:layout_height="wrap_
content"

    android:text="@string/label_
nomor_dilayani"

    android:layout_gravity="cent
er"

```

```

    android:gravity="center"

    android:textColor="@android:
color/white" />

</LinearLayout>

<LinearLayout
    android:layout_width="0dp"

    android:layout_weight="2"

    android:layout_height="match
parent"

    android:layout_marginTop="10
dp"

    android:layout_marginBottom=
"10dp"

    android:theme="@style/TextLa
rgeBlue">

<TextView
    android:id="@+id/nomordilaya
ni"

    android:padding="5dp"

    android:layout_width="match_
parent"

    android:layout_height="match
parent"

    android:text="001"

    android:layout_gravity="cent
er"

    android:gravity="center"/>

</LinearLayout>

</LinearLayout>

<LinearLayout
    android:layout_width="0dp"

    android:layout_weight="2"

    android:layout_height="match

```

```

_parent"

android:layout_marginBottom=
"16dp"

android:layout_margin="5dp"

android:elevation="2dp"

android:background="@drawabl
e/layout_background"

android:orientation="horizon
tal">

<LinearLayout

android:layout_width="0dp"

android:layout_weight="2"

android:layout_height="match
_parent"

android:background="@color/c
olorPrimary">

<TextView

android:layout_width="match_
parent"

android:layout_height="wrap_
content"

android:layout_gravity="cent
er"

android:gravity="center"

android:text="@string/label_
sis_a_ntrian"

android:textColor="@android:
color/white" />

</LinearLayout>

<LinearLayout

android:layout_width="0dp"

android:layout_weight="2"

android:layout_height="match
_parent"

```

```

android:layout_marginTop="10
dp"

android:layout_marginBottom=
"10dp"

android:theme="@style/TextLa
rgeBlue">

<TextView

android:id="@+id/sisaantrian
"

android:padding="5dp"

android:layout_width="match_
parent"

android:layout_height="wrap_
content"

android:layout_gravity="cent
er"

android:gravity="center"

android:text="020"/>

</LinearLayout>

</LinearLayout>

</LinearLayout>

</LinearLayout>

</RelativeLayout>

        <LinearLayout

android:orientation="horizon
tal"

android:layout_width="match_
parent"

android:layout_height="9dp"

android:layout_gravity="cent
er_horizontal|bottom"

android:background="@color/c
olorBackground">
        </LinearLayout>

```

```

        </FrameLayout>
        <LinearLayout
            android:layout_width="match_
parent"

            android:layout_height="wrap_
content"

            android:orientation="vertica
l">
            <RelativeLayout
                android:layout_width="match_
parent"

                android:layout_height="124dp
"

                android:paddingBottom="1dp"

                android:elevation="2dp"

                android:background="@drawabl
e/layout_background"

                android:layout_alignParentBo
ttom="true"

                android:layout_alignParentLe
ft="true"

                android:layout_alignParentSt
art="true"

                android:layout_marginBottom=
"17dp"

                android:layout_gravity="left
|bottom">
                <Space
                    android:id="@+id/spaceHolder
"

                    android:layout_width="10dp"

                    android:layout_height="0dp"

                    android:layout_centerHorizon
tal="true" />
                <Button
                    android:id="@+id/btn_cancel"

```

```

            android:layout_width="120dp"

            android:layout_height="50dp"

            android:paddingStart="10dp"

            android:paddingLeft="10dp"

            android:paddingRight="10dp"

            android:paddingEnd="10dp"

            android:background="@drawabl
e/button_secondary"

            android:textColor="@drawable
/button_secondary_text_color
"

            android:text="KELUAR"

            android:textSize="16sp"

            android:fontFamily="sans-
serif-medium"

            android:layout_centerVertica
l="true"

            android:layout_toLeftOf="@+i
d/spaceHolder"

            android:layout_toStartOf="@+
id/spaceHolder" />
            <Button
                android:id="@+id/Ambil"

                android:layout_width="120dp"

                android:layout_height="50dp"

                android:paddingStart="10dp"

                android:paddingLeft="10dp"

                android:paddingRight="10dp"

                android:paddingEnd="10dp"

                android:background="@drawabl
e/button_primary"

                android:textColor="@drawable
/button_primary_text_color"

```

```

android:text="AMBIL"

android:textSize="16sp"

android:fontFamily="sans-serif-medium"

android:layout_alignTop="@+id/btn_cancel"

android:layout_toRightOf="@+id/spaceHolder"

android:layout_toEndOf="@+id/spaceHolder" />

</RelativeLayout>
    </LinearLayout>
    </LinearLayout>
</LinearLayout>

```

c. source code notifikasi

```

package
com.example.faisal.antrian_b
oediman.gcm;

import
android.app.Notification;
import
android.app.NotificationManager;
import
android.app.PendingIntent;
import
android.content.Context;
import
android.content.Intent;
import
android.media.RingtoneManager;
import android.net.Uri;
import android.os.Bundle;
import
android.support.v4.app.NotificationCompat;
import
android.support.v4.content.LocalBroadcastManager;
import
android.text.TextUtils;
import android.util.Log;

//import
com.example.faisal.antrian_b

```

```

oediman.MainActivity;

import
com.example.faisal.antrian_b
oediman.MenuAntrian;
import
com.example.faisal.antrian_b
oediman.utils.SessionManager
;
import
com.example.faisal.antrian_b
oediman.utils.UserTakenUtility;
import
com.google.android.gms.gcm.GcmListenerService;

public class
GCMListenerService extends
GcmListenerService {

    private static final
String TAG =
GCMListenerService.class.getSimpleName();
    private
NotificationUtils
notificationUtils;
    UserTakenUtility
userTakenUtility;
    SessionManager
sessionManager;
    @Override
    public void
onMessageReceived(String
from, Bundle data) {
        userTakenUtility =
new UserTakenUtility();
        sessionManager = new
SessionManager(this);
        String title =
data.getString("title");
        String message =
data.getString("message");
        String image =
data.getString("image");
        String timestamp =
data.getString("created_at")
;
        Log.e(TAG, "From: "
+ from);
        Log.e(TAG, "Title: "
+ title);
        Log.e(TAG, "message:
" + message);
        Log.e(TAG, "image: "

```

```

+ image);
    Log.e(TAG,
"timestamp: " + timestamp);
    if
(!NotificationUtils.isAppIsInBackground(getApplicationCo
ntext())) {

userTakenUtility.clearTakeQue
ueData(this,
sessionManager.getIdUser()
);

NotificationUtils.clearNotif
ications(this);
    Intent
resultIntent = new
Intent(getApplicationContext
(), MenuAntrian.class);

resultIntent.putExtra("messa
ge", message);
    Intent
pushNotification = new
Intent(Config.PUSH_NOTIFICATION);

pushNotification.putExtra("m
essage", message);
LocalBroadcastManager.getIns
tance(this).sendBroadcast(pu
shNotification);
    if
(TextUtils.isEmpty(image)) {

showNotificationMessage(getA
pplicationContext(), title,
message, timestamp,
resultIntent);
    } else {
showNotificationMessageWithB
igImage(getApplicationContex
t(), title, message,
timestamp, resultIntent,
image);
    }
    } else {

    Intent
resultIntent = new
Intent(getApplicationContext
(), MenuAntrian.class);

resultIntent.putExtra("messa
ge", message);
    if

```

```

(TextUtils.isEmpty(image)) {

showNotificationMessage(getA
pplicationContext(), title,
message, timestamp,
resultIntent);
    } else {

showNotificationMessageWithB
igImage(getApplicationContex
t(), title, message,
timestamp, resultIntent,
image);
    }
    }

private void
sendNotification(String
message) {
    Intent intent = new
Intent(this,
MenuAntrian.class);

intent.addFlags(Intent.FLAG_
ACTIVITY_CLEAR_TOP);
    PendingIntent
pendingIntent =
PendingIntent.getActivity(th
is, 0, intent,
PendingIntent.FLAG_ONE_SHOT);
;
    Uri defaultSoundUri
=
RingtoneManager.getDefaultUr
i(RingtoneManager.TYPE_NOTIF
ICATION);

NotificationCompat.Builder
notificationBuilder = new
NotificationCompat.Builder(t
his)

.setContentTitle("Panggilan
Antrian")

.setContentText(message)

.setAutoCancel(true)

.setSound(defaultSoundUri)
.setContentIntent(pendingInt
ent).setPriority(Notification
n.PRIORITY_HIGH);
    NotificationManager

```

```

notificationManager =
(NotificationManager)
getSystemService(Context.NOT
IFICATION_SERVICE);

notificationManager.notify(C
onfig.NOTIFICATION_ID,
notificationBuilder.build())
;
    }
    /**
     * Showing notification
     with text only
     */
    private void
showNotificationMessage(Cont
ext context, String title,
String message, String
timeStamp, Intent intent) {
        notificationUtils =
new
NotificationUtils(context);

intent.setFlags(Intent.FLAG_
ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_T
ASK);

notificationUtils.showNotifi
cationMessage(title,
message, timeStamp, intent);
    }

    /**
     * Showing notification
     with text and image
     */
    private void
showNotificationMessageWithB
igImage(Context context,
String title, String
message, String timeStamp,
Intent intent, String
imageUrl) {
        notificationUtils =
new
NotificationUtils(context);

intent.setFlags(Intent.FLAG_
ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_T
ASK);

notificationUtils.showNotifi
cationMessage(title,
message, timeStamp, intent,

```

```

imageUrl);
    }
}

```