

## Jurnal MODUL 8

### anggota:

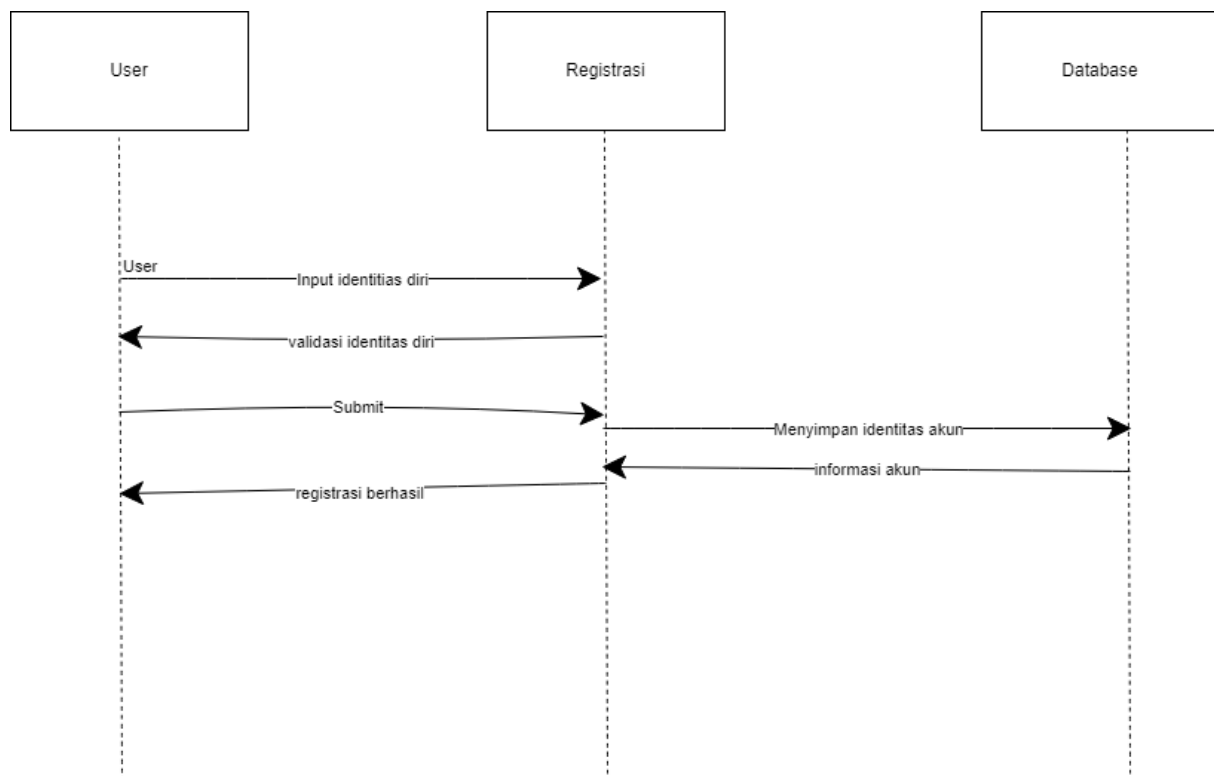
Afifuddin Mawardi - 1301194113

Muhammad Satria Pradananta - 1301190243

Muhamad Fikry Saputra - 1301194300

Muhammad Rifky Anami - 1301190357

### Sequence Diagram Register



## Sequence Register Register

```
1 package com.example.shashank.login;
2
3 import android.content.Context;
4 import android.content.Intent;
5 import android.database.Cursor;
6 import android.database.sqlite.SQLiteDatabase;
7 import android.support.v7.app.AppCompatActivity;
8 import android.os.Bundle;
9 import android.text.TextUtils;
10 import android.view.View;
11 import android.widget.Button;
12 import android.widget.EditText;
13 import android.widget.Toast;
14
15 public class RegisterActivity extends AppCompatActivity {
16
17     EditText username, Password, Name ;
18     Button Register;
19     String NameHolder, usernameHolder, PasswordHolder;
20     Boolean EditTextEmptyHolder;
21     SQLiteDatabase sqLiteDatabaseObj;
22     String SQLiteDataBaseQueryHolder ;
23     SQLiteHelper sqLiteHelper;
24     Cursor cursor;
25     String F_Result = "Not_Found";
26
27     @Override
28     protected void onCreate(Bundle savedInstanceState) {
29         super.onCreate(savedInstanceState);
30         setContentView(R.layout.activity_register);
31
32         Register = (Button)findViewById(R.id.buttonRegister);
33
34         username = (EditText)findViewById(R.id.editEmail);
35         Password = (EditText)findViewById(R.id.editPassword);
36         Name = (EditText)findViewById(R.id.editName);
37
38         sqLiteHelper = new SQLiteHelper( context: this);
39
40         Register.setOnClickListener((view) -> {
41             SQLiteDatabaseBuild();
42             SQLiteTableBuild();
43             CheckEditTextStatus();
44             CheckingusernameAlreadyExistsOrNot();
45             EmptyEditTextAfterDataInsert();
46         });
47     }
48 }
49
50
51
52
```

```

53 public void SQLiteDataBaseBuild(){
54     sqLiteDatabaseObj = openOrCreateDatabase(SQLiteHelper.DATABASE_NAME, Context.MODE_PRIVATE, factory: null);
55 }
56
57 public void SQLiteTableBuild() {
58     sqLiteDatabaseObj.execSQL("CREATE TABLE IF NOT EXISTS " + SQLiteHelper.TABLE_NAME + "(" + SQLiteHelper.Table_Colu
59 )
60
61 public void InsertDataIntoSQLiteDatabase(){
62     if(EditTextEmptyHolder == true)
63     {
64         SQLiteDataBaseQueryHolder = "INSERT INTO "+SQLiteHelper.TABLE_NAME+" (name,username,password) VALUES('"+NameH
65         sqLiteDatabaseObj.execSQL(SQLiteDataBaseQueryHolder);
66         sqLiteDatabaseObj.close();
67         Toast.makeText( context: RegisterActivity.this, text: "Register Sukses", Toast.LENGTH_LONG).show();
68     }
69     else {
70         Toast.makeText( context: RegisterActivity.this, text: "Text harus diisi semua !", Toast.LENGTH_LONG).show();
71     }
72 }
73
74
75
76 public void EmptyEditTextAfterDataInsert(){
77
78     Name.getText().clear();
79
80     username.getText().clear();
81
82     Password.getText().clear();
83
84 }
85

```

```

76 public void EmptyEditTextAfterDataInsert(){
77     Name.getText().clear();
78     username.getText().clear();
79     Password.getText().clear();
80 }
81
82 public void CheckEditTextStatus(){
83
84     NameHolder = Name.getText().toString() ;
85     usernameHolder = username.getText().toString();
86     PasswordHolder = Password.getText().toString();
87
88     if(TextUtils.isEmpty(NameHolder) || TextUtils.isEmpty(usernameHolder) || TextUtils.isEmpty>PasswordHolder)){
89         EditTextEmptyHolder = false ;
90     }
91     else {
92         EditTextEmptyHolder = true ;
93     }
94 }
95
96 public void CheckingusernameAlreadyExistsOrNot(){
97     sqLiteDatabaseObj = sqLiteHelper.getWritableDatabase();
98     cursor = sqLiteDatabaseObj.query(SQLiteHelper.TABLE_NAME, columns: null, selection: " " + SQLiteHelper.Table_Column_2_u
99     while (cursor.moveToNext()) {
100         if (cursor.isFirst()) {
101             cursor.moveToFirst();
102             F_Result = "username Found";
103             cursor.close();
104         }
105     }
106     CheckFinalResult();
107 }

```

```


110 public void CheckFinalResult(){
111     if(F_Result.equalsIgnoreCase( anotherString: "username Found"))
112     {
113         Toast.makeText( context: RegisterActivity.this, text: "username Already Exists", Toast.LENGTH_LONG).show();
114     }
115     else {
116         InsertDataIntoSQLiteDatabase();
117     }
118     F_Result = "Not_Found" ;
119 }
120 }

```

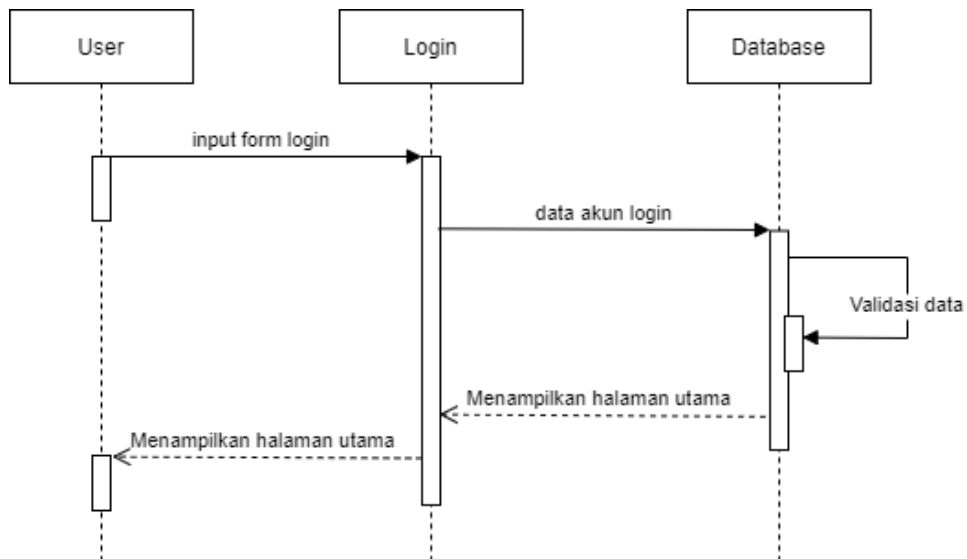
## UI

### Sign Up

DAFTAR



## Sequence Login



## CRUD - Create Username Account, terhubung dengan Database

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    LoginButton = (Button)findViewById(R.id.buttonLogin);
    RegisterButton = (Button)findViewById(R.id.buttonRegister);
    username = (EditText)findViewById(R.id.editEmail);
    Password = (EditText)findViewById(R.id.editPassword);

    sqliteHelper = new SQLiteHelper( context: this);
    LoginButton.setOnClickListener((view) -> {
        // Calling EditText is empty or no method.
        CheckEditTextStatus();
        // Calling login method.
        LoginFunction();
    });

    RegisterButton.setOnClickListener((view) -> {
        // Opening new user registration activity using intent on button click.
        Intent intent = new Intent( packageContext: MainActivity.this, RegisterActivity.class);
        startActivity(intent);
    });
}

```

## Fungsi Login

```
public void LoginFunction(){
    if(EditTextEmptyHolder) {
        SQLiteDatabaseObj = sqliteHelper.getWritableDatabase();
        cursor = SQLiteDatabaseObj.query(SQLiteHelper.TABLE_NAME, columns: null, selection: " " + SQLiteHelper.Table_Column_2_username +
        while (cursor.moveToNext()) {
            if (cursor.isFirst()) {
                cursor.moveToFirst();
                TempPassword = cursor.getString(cursor.getColumnIndex(SQLiteHelper.Table_Column_3_Password));
                cursor.close();
            }
        }
        CheckFinalResult();
    }
    else {
        Toast.makeText(context: MainActivity.this, text: "username dan password harus diisi", Toast.LENGTH_LONG).show();
    }
}
```

## Login Error Handle

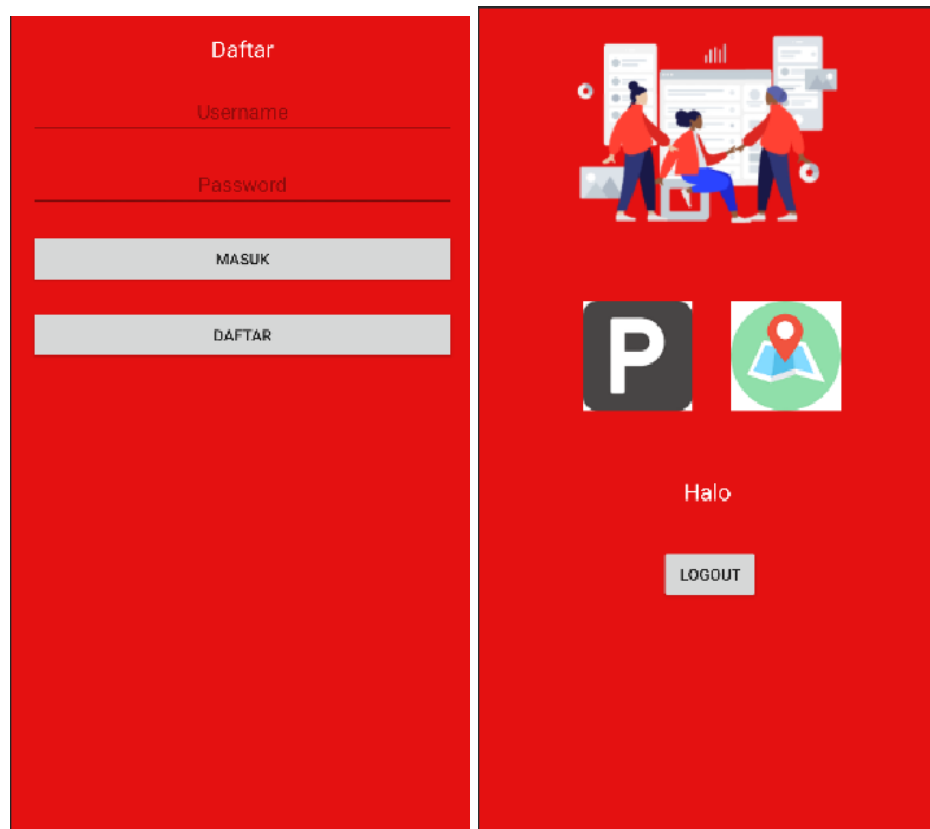
```
public void CheckEditTextStatus(){

    usernameHolder = username.getText().toString();
    PasswordHolder = Password.getText().toString();

    if( TextUtils.isEmpty(usernameHolder) || TextUtils.isEmpty>PasswordHolder)){
        EditTextEmptyHolder = false ;
    }
    else {
        EditTextEmptyHolder = true ;
    }
}
```

```
public void CheckFinalResult(){
    if(TempPassword.equalsIgnoreCase>PasswordHolder))
    {
        Toast.makeText(context: MainActivity.this, text: "Login Sukses", Toast.LENGTH_LONG).show();
        // Going to Dashboard activity after login success message.
        Intent intent = new Intent(packageContext: MainActivity.this, DashboardActivity.class);
        // Sending username to Dashboard Activity using intent.
        intent.putExtra(Userusername, usernameHolder);
        startActivity(intent);
    }
    else {
        Toast.makeText(context: MainActivity.this, text: "UserName atau Password salah", Toast.LENGTH_LONG).show();
    }
    TempPassword = "NOT_FOUND" ;
}
}
```

## UI



Link GitHub

[https://github.com/fikrysaputra/RPL-Desain-dan-Implementasi\\_TParking](https://github.com/fikrysaputra/RPL-Desain-dan-Implementasi_TParking)