

Nama : John Isaac Witness

NIM : 00000088626

Mobile Application Programming W9

LINK GITHUB: https://github.com/akunjone/LAB_WEEK_09

Commit No. 1 – Building Jetpack Compose UI

```
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        setContent {
            ...
        }
    }

    @Preview(showBackground = true)
    @Composable
    fun PreviewHome() {
        Home(items = listOf("Tanu", "Tina", "Tono"))
    }
}

@Composable
fun Home(items: List<String>) {
    LazyColumn {
        item {
            Column {
                modifier = Modifier.padding(all = 16.dp).fillMaxSize(),
                horizontalAlignment = Alignment.CenterHorizontally
            }
        }
    }
}
```

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the `MainActivity.kt` file, which contains Jetpack Compose code to create a list of names. A preview window on the right shows a UI with an input field and a list of names: Tanu, Tina, and Tono.

Commit No. 2 – State and Event Handler

Sebelum dimasukkan nama-nama baru:

```
surface {
    fun Home(
        //items: List<String>
    ) {
        val listData = remember {
            mutableStateListOf(
                Student(name = "Tanu"),
                Student(name = "Tina"),
                Student(name = "Tono")
            )
        }

        var inputField = remember { mutableStateOf(value = Student(name = "")) }

        HomeContent(
            listData,
            inputField = inputField.value,
            onInputValueChange = { input -> inputField.value = input.copy(name = input.name) },
            onButtonClick = {
                if(inputField.value.name.isNotBlank()){
                    listData.add(inputField.value)
                    inputField.value = Student(name = "")
                }
            }
        )
    }
}
```

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the `MainActivity.kt` file, which contains Jetpack Compose code with state management using `remember` and event handling using `onButtonClick`. A preview window on the right shows a UI with an input field and a list of names: Tanu, Tina, and Tono.

Ketika nama diinput:

Commit No. 1 - Building Jetpack Compose UI

```
fun Home( //items: List<String>, ){ val listData = remember{ mutableStateListOf( Student( name = "Tanu"), Student( name = "Tina"), Student( name = "Tono") ) } var inputField = remember{ mutableStateOf( value = Student( name = "")) } HomeContent( listData, inputField = inputField.value, onInputValueChange = {input -> inputField.value = inputField.value.copy( name = input) }, onClick = { if(inputField.value.name.isNotBlank()){ listData.add(inputField.value) inputField.value = Student( name = "") } } ) LazyColumn { item{ Column ( modifier = Modifier.padding(16.dp).fillMaxSize(), horizontalAlignment = Alignment.CenterHorizontally ){ Text(text = strRes) } } } }
```

Enter a name
John
Submit
Tanu
Tina
Tono

Setelah klik submit:

Commit No. 1 - Building Jetpack Compose UI

```
fun Home( //items: List<String>, ){ val listData = remember{ mutableStateListOf( Student( name = "Tanu"), Student( name = "Tina"), Student( name = "Tono") ) } var inputField = remember{ mutableStateOf( value = Student( name = "")) } HomeContent( listData, inputField = inputField.value, onInputValueChange = {input -> inputField.value = inputField.value.copy( name = input) }, onClick = { if(inputField.value.name.isNotBlank()){ listData.add(inputField.value) inputField.value = Student( name = "") } } ) LazyColumn { item{ Column ( modifier = Modifier.padding(16.dp).fillMaxSize(), horizontalAlignment = Alignment.CenterHorizontally ){ Text(text = strRes) } } } }
```

Enter a name
Submit
Tanu
Tina
Tono
John

Commit No. 3 – UI Element and Theme

The screenshot shows the Android Studio interface with the project `LAB_WEEK_09` open. The code editor displays `MainActivity.kt` which contains Kotlin code for a LazyColumn. The preview window on the right shows a mobile application interface with a text input field labeled "Enter a name", a "Submit" button, and a list of names: Tanu, Tina, Tono. The bottom status bar indicates the build was successful at 08/11/2025 15:16.

```
fun HomeContent(
    LazyColumn {
        item {
            // Button(onClick = { onButtonClick() })
            Text(text = stringResource(R.string.button_click))
        }
    }
    items(items = listData){ item ->
        Column (
            modifier = Modifier.padding(vertical = 4.dp).fillMaxSize(),
            horizontalAlignment = Alignment.CenterHorizontally
        ){
            //call OnBackgroundItemText UI Element
            OnBackgroundTitleText(text = item.name)
            //Text(text = item.name)
        }
    }
}
```

Commit No. 4 – Navigation

The screenshot shows the Android Studio interface with the project `LAB_WEEK_09` open. The code editor displays `MainActivity.kt` which has been modified to include imports for navigation components like `NavController` and `NavigationGraph`. The preview window on the right shows a mobile application interface with a text input field labeled "Enter a name", a "Submit" button, and a list of names: Tanu, Tina, Tono. The bottom status bar indicates the build was successful at 08/11/2025 15:46.

```
package com.example.lab_week_09

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Row
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.items
import androidx.compose.foundation.text.KeyboardOptions
import androidx.compose.material3.Button
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Scaffold
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.material3.TextField
import androidx.compose.runtime.Composable
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.snapshots.SnapshotStateList
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.res.stringResource
```

The screenshot shows the Android Studio interface with the project 'LAB_WEEK_09' open. The code editor displays the file 'MainActivity.kt'. In the top right corner, there is a floating window titled 'Enter a name' containing a text input field and two buttons: 'Submit' and 'Finish'. Below this window, a list of student names is displayed: Tanu, Tina, Tono, and John. The bottom right corner of the screen shows the text 't:1'.

```
package com.example.lab_week_09

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Row
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.items
import androidx.compose.foundation.text.KeyboardOptions
import androidx.compose.material3.Button
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Scaffold
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.material3.TextField
```

Setelah klik finish:

The screenshot shows the same Android Studio environment as before, but now the floating window has disappeared. The list of student names at the bottom of the screen now includes the new entry 'Student(name=John)'. The bottom right corner still shows 't:1'.

```
[Student(name=Tanu),  
 Student(name=Tina),  
 Student(name=Tono),  
 Student(name=John)]
```

ASSIGNMENT 1

User tetap akan bisa mengetik Enter atau Spasi, tapi saat disubmit, akan ditrim:

The screenshot shows the Android Studio interface with the code editor open to `MainActivity.kt`. The code implements a state management logic for adding students to a list. A floating input dialog titled "Enter a name" is displayed, showing the text "budi". The list view on the right shows three items: Tanu, Tina, and Tono.

```
fun Home() {
    val listData = remember {
        Student(name = "Tina"),
        Student(name = "Tono")
    }
    var inputField = remember { mutableStateOf(value = Student(name = "")) }
    HomeContent(
        listData,
        inputField = inputField.value,
        onInputValueChange = { input ->
            inputField.value = inputField.value.copy(name = input)
        },
        onButtonClick = {
            if(inputField.value.name.trim().isNotBlank()) {
                // replace enter/newline dengan space, lalu trim
                inputField.value = inputField.value.copy(name = inputField.value.name.replace( oldValue = ""))
                listData.add(inputField.value)
                inputField.value = Student(name = "")
            }
        },
        navigateFromHomeToResult = { navigateFromHomeToResult(listData.toList().toString()) }
    )
}
```

This screenshot is identical to the one above, except the floating input dialog now contains the text "budi". The list view on the right still shows Tanu, Tina, and Tono.

```
fun Home() {
    val listData = remember {
        Student(name = "Tina"),
        Student(name = "Tono")
    }
    var inputField = remember { mutableStateOf(value = Student(name = "")) }
    HomeContent(
        listData,
        inputField = inputField.value,
        onInputValueChange = { input ->
            inputField.value = inputField.value.copy(name = input)
        },
        onButtonClick = {
            if(inputField.value.name.trim().isNotBlank()) {
                // replace enter/newline dengan space, lalu trim
                inputField.value = inputField.value.copy(name = inputField.value.name.replace( oldValue = ""))
                listData.add(inputField.value)
                inputField.value = Student(name = "")
            }
        },
        navigateFromHomeToResult = { navigateFromHomeToResult(listData.toList().toString()) }
    )
}
```

atau jika menekan tombol enter akan ditrim juga:

The screenshot shows the Android Studio interface with the code editor open to `MainActivity.kt`. The code implements a state management logic for an input field and a list of students. A floating window displays an input field with placeholder "Enter a name" and a list of names: Tanu, Tina, Tono, budi. The list includes the user input "juniiiii".

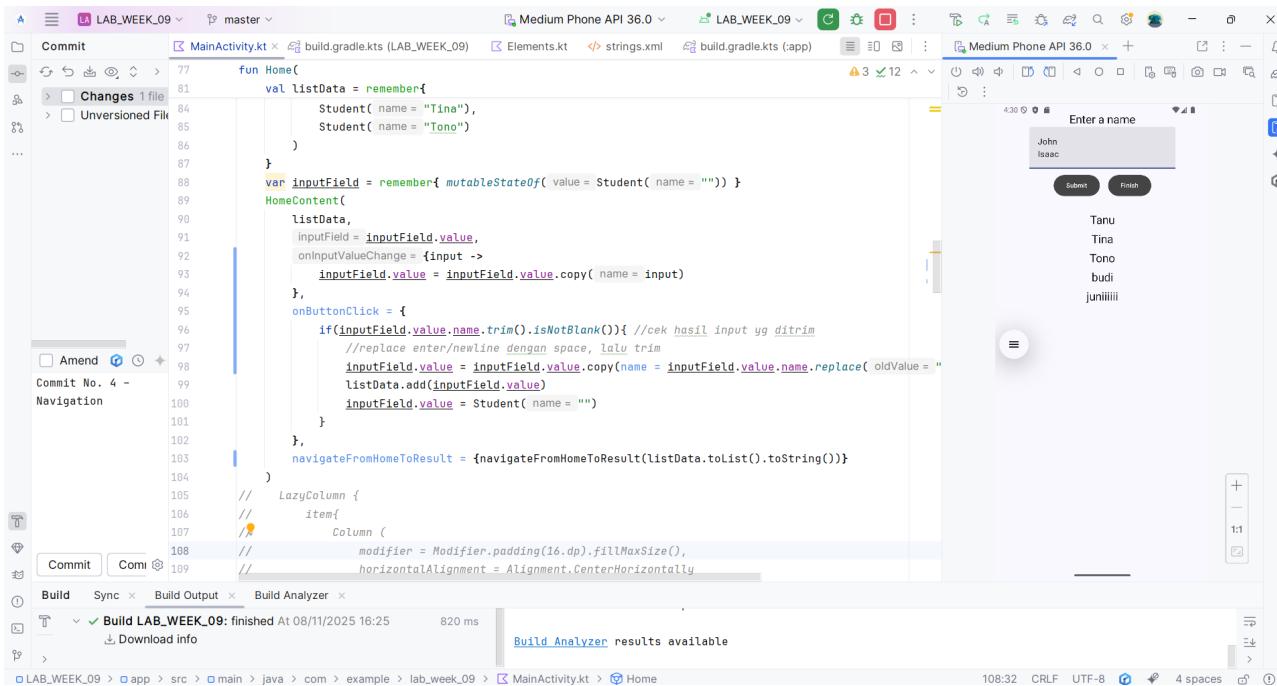
```
fun Home() {
    val listData = remember {
        Student(name = "Tina"),
        Student(name = "Tono")
    }
    var inputField = remember { mutableStateOf(value = Student(name = "")) }
    HomeContent(
        listData,
        inputField = inputField.value,
        onInputValueChange = { input ->
            inputField.value = inputField.value.copy(name = input)
        },
        onButtonClick = {
            if(inputField.value.name.trim().isNotBlank()){//cek hasil input yg ditrim
                //replace enter/newline dengan space, lalu trim
                inputField.value = inputField.value.copy(name = inputField.value.name.replace(" ", ""))
                listData.add(inputField.value)
                inputField.value = Student(name = "")
            }
        },
        navigateFromHomeToResult = {navigateFromHomeToResult(listData.toList().toString())}
    )
}
```

The screenshot shows the same setup as the first one, but the floating window now displays the user input "juniiiii" instead of "juniiii". The list of names remains the same: Tanu, Tina, Tono, budi.

```
fun Home() {
    val listData = remember {
        Student(name = "Tina"),
        Student(name = "Tono")
    }
    var inputField = remember { mutableStateOf(value = Student(name = "")) }
    HomeContent(
        listData,
        inputField = inputField.value,
        onInputValueChange = { input ->
            inputField.value = inputField.value.copy(name = input)
        },
        onButtonClick = {
            if(inputField.value.name.trim().isNotBlank()){//cek hasil input yg ditrim
                //replace enter/newline dengan space, lalu trim
                inputField.value = inputField.value.copy(name = inputField.value.name.replace(" ", ""))
                listData.add(inputField.value)
                inputField.value = Student(name = "")
            }
        },
        navigateFromHomeToResult = {navigateFromHomeToResult(listData.toList().toString())}
    )
}
```

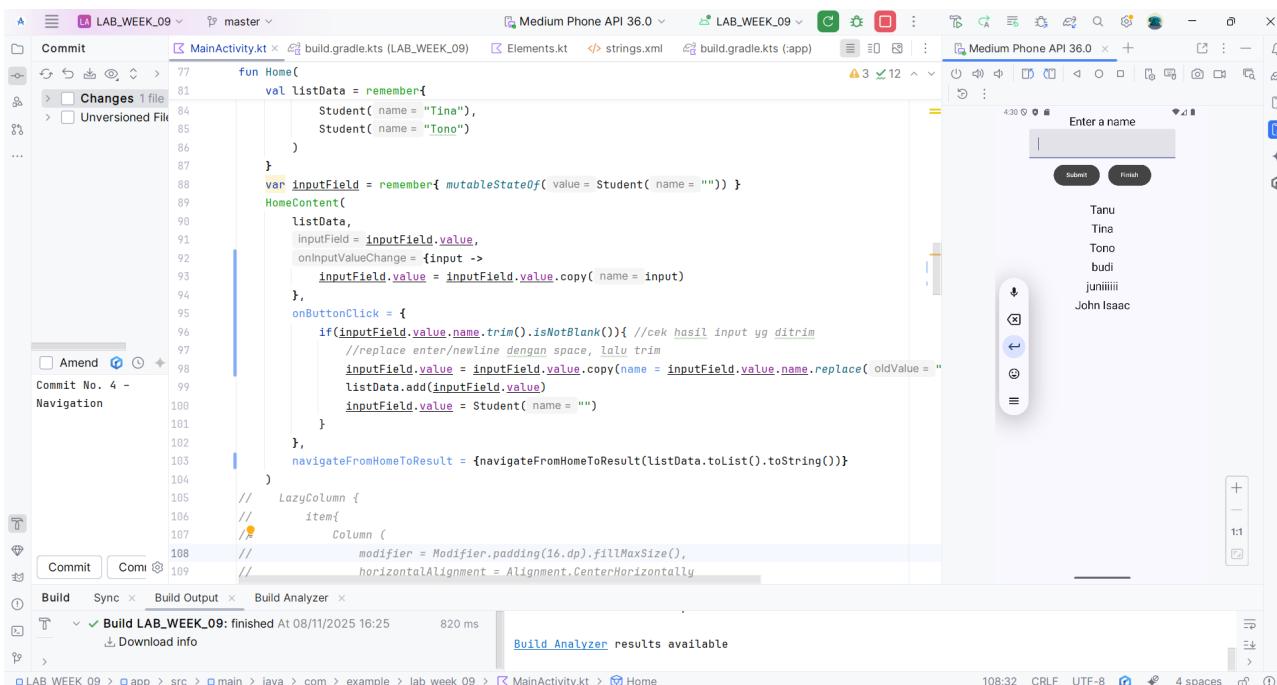
Jika user menekan enter atau spasi, tidak akan tersubmit.

Dan jika user menekan tombol enter ditengah tengah input data, akan diganti dengan spasi:



```
fun Home() {
    val listData = remember {
        Student(name = "Tina"),
        Student(name = "Tono")
    }
    var inputField = remember { mutableStateOf(value = Student(name = "")) }
    HomeContent(
        listData,
        inputField = inputField.value,
        onInputValueChange = { input ->
            inputField.value = inputField.value.copy(name = input)
        },
        onButtonClick = {
            if(inputField.value.name.trim().isNotBlank()){//cek hasil input yg ditrim
                //replace enter/newline dengan space, lalu trim
                inputField.value = inputField.value.copy(name = inputField.value.name.replace(" ", ""))
                listData.add(inputField.value)
                inputField.value = Student(name = "")
            }
        },
        navigateFromHomeToResult = {navigateFromHomeToResult(listData.toList().toString())}
    )
}
```

Build LAB_WEEK_09: finished At 08/11/2025 16:25



```
fun Home() {
    val listData = remember {
        Student(name = "Tina"),
        Student(name = "Tono")
    }
    var inputField = remember { mutableStateOf(value = Student(name = "")) }
    HomeContent(
        listData,
        inputField = inputField.value,
        onInputValueChange = { input ->
            inputField.value = inputField.value.copy(name = input)
        },
        onButtonClick = {
            if(inputField.value.name.trim().isNotBlank()){//cek hasil input yg ditrim
                //replace enter/newline dengan space, lalu trim
                inputField.value = inputField.value.copy(name = inputField.value.name.replace(" ", ""))
                listData.add(inputField.value)
                inputField.value = Student(name = "")
            }
        },
        navigateFromHomeToResult = {navigateFromHomeToResult(listData.toList().toString())}
    )
}
```

Build LAB_WEEK_09: finished At 08/11/2025 16:25

Dan itu juga tidak akan mengganggu json ketika klik tombol finish:

The screenshot shows the Android Studio interface with the code editor open to `MainActivity.kt`. The code is a Kotlin function for the `Home` component. It uses `remember` to handle state changes, including adding a new student to a list and updating an input field. A preview window on the right shows a list of students: [Student(name=Tanu), Student(name=Tina), Student(name=Tono), Student(name=budi), Student(name=juniiii), Student(name=John Isaac)]. Below the code editor, the build output shows a successful build at 16:25. The bottom status bar indicates the file is 820 ms old.

```
fun Home() {
    val listData = remember {
        Student(name = "Tina"),
        Student(name = "Tono")
    }
    var inputField = remember { mutableStateOf(Student(name = "")) }
    HomeContent(
        listData,
        inputField = inputField.value,
        onInputValueChange = { input ->
            inputField.value = inputField.value.copy(name = input)
        },
        onButtonClick = {
            if(inputField.value.name.trim().isNotBlank()) {
                // replace enter/newline dengan space, lalu trim
                inputField.value = inputField.value.copy(name = inputField.value.name.replace(" ", ""))
                listData.add(inputField.value)
                inputField.value = Student(name = "")
            }
        },
        navigateFromHomeToResult = { navigateFromHomeToResult(listData.toList().toString()) }
    )
}
```

ASSIGNMENT 2

Data mahasiswa dimasukkan kedalam json, json akan diconvert ke LazyColumn:

The screenshot shows the Android Studio interface with the code editor open to `MainActivity.kt`. The code defines a `ResultContent` function that takes a string of JSON and converts it into a list of students using Moshi. It then displays this list in a `LazyColumn`. A preview window on the right shows a user interface with an input field labeled "Enter a name" and three buttons labeled "Submit", "Finish", and "Tono". The bottom status bar indicates the file is 829 ms old.

```
@Composable
fun ResultContent(listData: String) {
    val students = try {
        MoshiInstance.studentListAdapter.fromJson(string = listData)
    } catch (e: Exception) {
        emptyList()
    }
    LazyColumn {
        modifier = Modifier
            .padding(vertical = 4.dp)
            .fillMaxSize(),
        horizontalAlignment = Alignment.CenterHorizontally
    } {
        item {
            OnBackgroundTitleText(
                text = "Student List: ${students.size} items"
            )
        }
        items(items = students){ student ->
            Column(
                modifier = Modifier
                    .padding(vertical = 8.dp, horizontal = 16.dp)
                    .fillMaxWidth(),
                horizontalAlignment = Alignment.CenterHorizontally
            ) {
                OnBackgroundTitleText(text = student.name)
            }
        }
    }
}
```

ketika kita mengklik Finish:

The screenshot shows the Android Studio interface with the code editor open to `MainActivity.kt`. The code defines a `ResultContent` function that returns a `LazyColumn` containing a title and a list of student names. The list currently shows three items: Tanu, Tina, and Tono. The code editor has syntax highlighting and code completion features.

```
1 Usage
@Composable
fun ResultContent(listData: String) {
    val students = try {
        MoshiInstance.studentListAdapter.fromJson(string = listData)
    } catch (e: Exception) {
        emptyList()
    }
    LazyColumn(
        modifier = Modifier
            .padding(vertical = 4.dp)
            .fillMaxSize(),
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        item {
            OnBackgroundTitleText(
                text = "Student List: (${students.size} items)"
            )
        }
        items(items = students){ student ->
            Column(
                modifier = Modifier
                    .padding(vertical = 8.dp, horizontal = 16.dp)
                    .fillMaxWidth(),
                horizontalAlignment = Alignment.CenterHorizontally
            ) {
                OnBackgroundTitleText(text = student.name)
            }
        }
    }
}
```

hasilnya akan berupa text LazyColumn.

Dan jika ditambahkan student baru:

The screenshot shows the Android Studio interface with the code editor open to `MainActivity.kt`. The code is identical to the previous one, but the list now contains four items: Tanu, Tina, Tono, and John Isaac. The code editor highlights the new student name.

```
1 Usage
@Composable
fun ResultContent(listData: String) {
    val students = try {
        MoshiInstance.studentListAdapter.fromJson(string = listData)
    } catch (e: Exception) {
        emptyList()
    }
    LazyColumn(
        modifier = Modifier
            .padding(vertical = 4.dp)
            .fillMaxSize(),
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        item {
            OnBackgroundTitleText(
                text = "Student List: (${students.size} items)"
            )
        }
        items(items = students){ student ->
            Column(
                modifier = Modifier
                    .padding(vertical = 8.dp, horizontal = 16.dp)
                    .fillMaxWidth(),
                horizontalAlignment = Alignment.CenterHorizontally
            ) {
                OnBackgroundTitleText(text = student.name)
            }
        }
    }
}
```

akan tetap bertambah.

Terima kasih.