1. Buat project baru, beri nama **ProfilUser**.
2. Buka `build.gradle.kts (Module :app)`, scroll ke blok `dependencies`, lalu tambahkan dependensi berikut untuk `ViewModel`. Dependensi ini digunakan untuk menambahkan viewmodel berbasis siklus proses ke aplikasi compose Anda.

```
dependencies {
// other dependencies

    implementation("androidx.lifecycle:lifecycle-viewmodel-
compose:2.6.1")
//...
}
```

Klik Sync Now

1. Dalam paket com.example.profiluser buat paket baru bernama **data**
2. Dalam paket **data**, buat class/file Kotlin bernama **UserUiState** dengan pilihan Data class

```
package com.example.profiluser.data

data class UserUiState(
    val nama : String = "Siti Munawaroh",
    val umur : Int = 21
)
```

3. Dalam paket **com.example.profiluser** buat paket baru bernama **model**.
4. Dalam paket **model** buat class/file Kotlin bernama **UserViewModel** dengan kode seperti berikut.

```
import androidx.lifecycle.ViewModel
import com.example.profiluser.data.UserUiState
import kotlinx.coroutines.flow.MutableStateFlow
import kotlinx.coroutines.flow.StateFlow
import kotlinx.coroutines.flow.asStateFlow


class UserViewModel : ViewModel() {
    private val _userUiState = MutableStateFlow(UserUiState())
    val userUiState: StateFlow<UserUiState> = _userUiState.asStateFlow()

    fun updateUser(nama: String, umur: Int) {
        _userUiState.value = _userUiState.value.copy(nama = nama, umur =
umur)
    }
}
```

5. Ubah **MainActivity** menjadi seperti berikut.

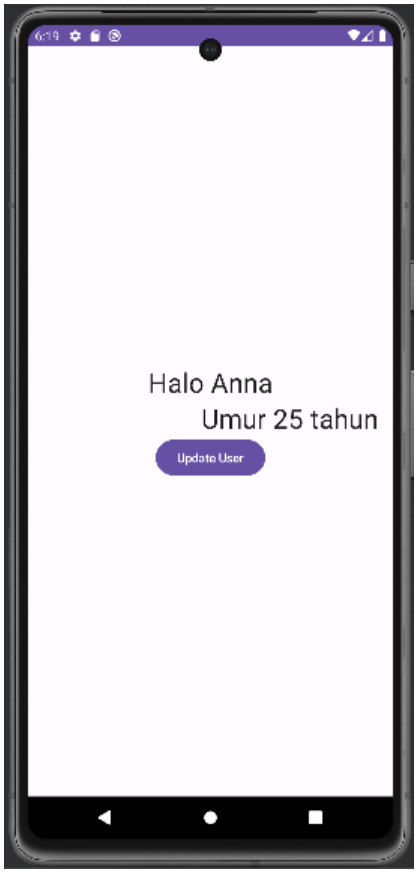
```
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Arrangement
```

```
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.Button
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.collectAsState
import androidx.compose.runtime.getValue
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.profiluser.model.UserViewModel
import com.example.profiluser.ui.theme.ProfilUserTheme
import androidx.lifecycle.viewmodel.compose.viewModel

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            ProfilUserTheme {
                // A surface container using the 'background' color from
the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    ProfilUserApp()
                }
            }
        }
    }
}


@Composable
fun ProfilUserApp(viewModel: UserViewModel = viewModel(),
                  modifier: Modifier = Modifier,
                  onUpDateUser: () -> Unit =
{viewModel.updateUser("Anna",25)}
){
    val user by viewModel.userUiState.collectAsState()
    Column(
        verticalArrangement = Arrangement.Center,
        modifier = modifier.padding(16.dp)

    ) {
        Text(
            text = "Halo ${user.nama}",
            fontSize = 30.sp,
```

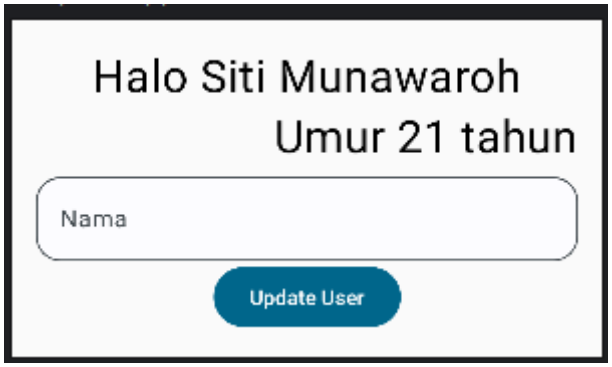
```
            modifier = Modifier.align(alignment =
Alignment.CenterHorizontally)
        )
        Text(
            text = "Umur ${user.umur} tahun",
            fontSize = 30.sp,
            modifier = Modifier.align(alignment = Alignment.End)
        )
        Button(
            modifier = Modifier.align(alignment =
Alignment.CenterHorizontally),
            onClick = onUpDateUser
        ) {
            Text("Update User")
        }

    }
}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    ProfilUserTheme {
        ProfilUserApp()
    }
}
```

Hasil:

Jika button Update User diklik akan menghasilkan:



6. Kita bisa menambahkan OutlinedTextField sehingga user dapat mengisi nama:



7. Tambahkan ke file UserViewModel.kt :

```
// jika isian nama diubah akan terjadi rekomposisi (penampilan ulang)
var isianNama by mutableStateOf("")
    private set
. . .

fun updateNama(namaUser: String){
    isianNama = namaUser
}
```

Jangan lupa mengimport:

```
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.setValue
```

8. Kedalam MainActivity, tambahkan kode berikut tepat di atas Button:

```
OutlinedTextField(
    value = viewModel.isianNama,  // isian awal TextField
    singleLine = true,
    shape = MaterialTheme.shapes.large,
    modifier = Modifier.fillMaxWidth(),
    colors = TextFieldDefaults.colors(
        focusedContainerColor = MaterialTheme.colorScheme.surface,
        unfocusedContainerColor = MaterialTheme.colorScheme.surface,
        disabledContainerColor = MaterialTheme.colorScheme.surface,
    ),
    onValueChange = { },  // jika user mengetik akan mengerjakan kode
ini
    label = {
            Text(text="Nama")
    },
    keyboardOptions = KeyboardOptions.Default.copy(
        imeAction = ImeAction.Done
    ),
    keyboardActions = KeyboardActions(
        onDone = { } // jika tombol <enter> diklik akan menjalankan kode
ini
    )
)
```

Jangan lupa menambahkan sejumlah import:

```
import androidx.compose.material3.OutlinedTextField
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.material3.TextFieldDefaults
import androidx.compose.foundation.text.KeyboardOptions
import androidx.compose.ui.text.input.ImeAction
import androidx.compose.foundation.text.KeyboardActions
```

9. Jalankan aplikasi, coba masukkan nama ke **OutlinedTextField**. Sampai di sini user masih belum bisa memasukkan nama. Untuk mengerjakan itu parameter **onValueChange** perlu ditambahkan kode untuk mengisinya. Ubahlah kode :

```
    onValueChange = { },  // jika user mengetik akan mengerjakan kode
ini
```

menjadi:

```
onValueChange = {viewModel.updateNama(it)}, // jika user mengetik akan
mengerjakan kode ini
```

10. Ubahlah juga di

```
    keyboardActions = KeyboardActions(
        onDone = { } // jika tombol <enter> diklik akan menjalankan kode
ini
    )
```

Menjadi

```
 keyboardActions = KeyboardActions(
    onDone = { onUpDateUser() } // jika tombol <enter> diklik akan
menjalankan kode ini
)
```

11. Jalankan aplikasi, coba masukkan nama ke OutlinedTextField, seharusnya sudah bisa memasukkan nama.

Selanjutnya, nama yang sudah diisi oleh user dapat digunakan untuk mengganti nama yang ditampilkan pada saat button diklik.

12. Tambahkan fungsi berikut di kelas UserViewModel:

```
fun updateUser2() {
    _userUiState.value = _userUiState.value.copy(nama = isianNama)
}
```

13. Dalam MainActivity, fungsi ResponsiApp(), ubahlah parameter:

```
onUpDateUser: () -> Unit = {viewModel.updateUser("Anna",25)}
```

menjadi:

```
onUpDateUser: () -> Unit = {viewModel.updateUser2()}
```

14. Jalankan aplikasi, ketikkan nama ke **OutlinedTextField**, kemudian klik button Update User.

Tambahkan ke dalam aplikasi untuk mengisi umur, sehingga jika button diklik akan menampilkan nama dan umur dari masukan user.

15. Di bawah **OutlinedTextField** untuk isian nama, tambahkan composable **OutlinedTextField** untuk isisan umur.

```
OutlinedTextField(  // untuk mengisi umur
    value = viewModel.isianUmur,  // isian awal TextField
    singleLine = true,
    shape = MaterialTheme.shapes.large,
    modifier = Modifier.fillMaxWidth(),
    colors = TextFieldDefaults.colors(
        focusedContainerColor = MaterialTheme.colorScheme.surface,
        unfocusedContainerColor = MaterialTheme.colorScheme.surface,
        disabledContainerColor = MaterialTheme.colorScheme.surface,
    ),
    onValueChange = { viewModel.updateUmur(it) },  // jika user mengetik
akan mengerjakan kode ini
    label = {
        Text(text="Umur")
    },
    keyboardOptions = KeyboardOptions.Default.copy(
        imeAction = ImeAction.Done
    ),
    keyboardActions = KeyboardActions(
        onDone = { onUpDateUser() } // jika tombol <enter> diklik akan
menjalankan kode ini
    )
)
```

16. Tambahkan ke file UserViewModel.kt kode berikut untuk isian umur:

```
var isianUmur by mutableStateOf("")
    private set
. . .
fun updateUmur(umurUser: String){
    isianUmur = umurUser
}

fun updateUser3() {
    var umur1 : Int? = isianUmur.toIntOrNull()
    if(umur1 == null) umur1 = 0
    _userUiState.value = _userUiState.value.copy(nama = isianNama, umur
= umur1)
}
```

17. Dalam MainActivity, fungsi ResponsiApp(), ubahlah parameter:

```
onUpDateUser: () -> Unit = {viewModel.updateUser2()}
```

menjadi:

```
onUpDateUser: () -> Unit = {viewModel.updateUser3()}
```