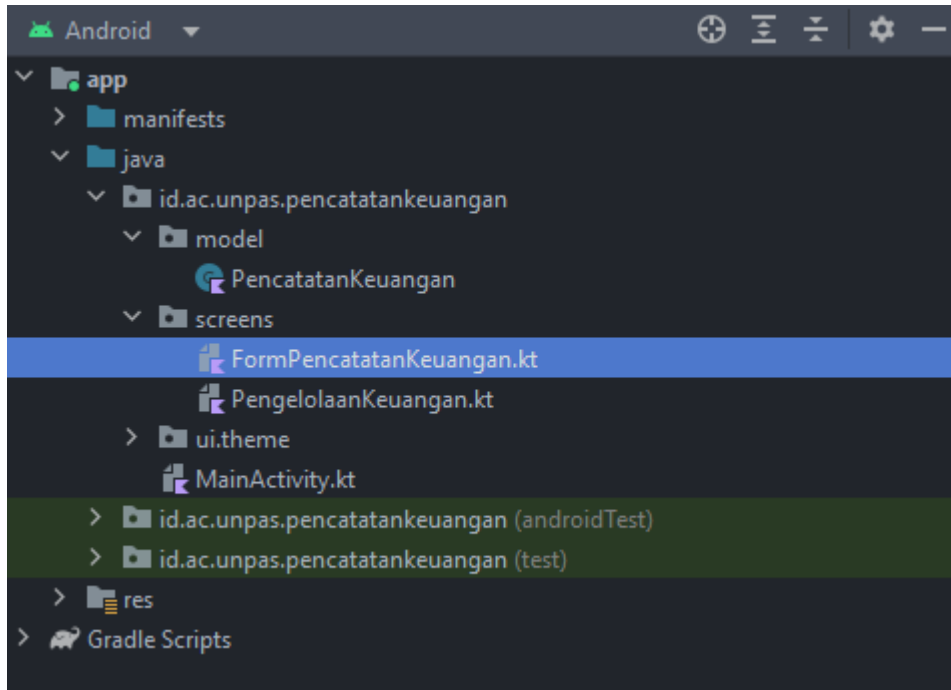




FAKULTAS TEKNIK UNIVERSITAS PASUNDAN SEMESTER GENAP 2022

Npm : 203040098
Nama : Agam Ramdhan Kamil Atmaja
Matakuliah : Praktikum Pemograman Mobile - A

Struktur Folder



Source Code

Model/PencatatanKeuangan

```
package id.ac.unpas.pencatatankeuangan.model

data class PencatatanKeuangan(
    val tanggal: String,
    val keterangan: String,
    val pemasukan: String,
    val pengeluaran: String
)
```

Screens/FormPencatatanKeuangan.Kt

```
package id.ac.unpas.pencatatankeuangan.screens

import android.widget.Toast
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.text.KeyboardOptions
import androidx.compose.material.*
import androidx.compose.runtime.Composable
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
```



FAKULTAS TEKNIK

UNIVERSITAS PASUNDAN

SEMESTER GENAP 2022

```
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.text.TextStyle
import androidx.compose.ui.text.input.KeyboardCapitalization
import androidx.compose.ui.text.input.KeyboardType
import androidx.compose.ui.text.input.TextFieldValue
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import id.ac.unpas.pencatatankeuangan.model.PencatatanKeuangan
import id.ac.unpas.pencatatankeuangan.ui.theme.Purple700
import id.ac.unpas.pencatatankeuangan.ui.theme.Teal200

@Composable
fun FormPencatatanKeuangan(onSimpan: (PencatatanKeuangan) -> Unit) {
    val context = LocalContext.current
    val tanggal = remember { mutableStateOf(TextFieldValue("")) }
    val keterangan = remember { mutableStateOf(TextFieldValue("")) }
    val pemasukan = remember { mutableStateOf(TextFieldValue("")) }
    val pengeluaran = remember { mutableStateOf(TextFieldValue("")) }

    Column(modifier = Modifier
        .padding(10.dp)
        .fillMaxWidth()) {
        OutlinedTextField(
            label = { Text(text = "Tanggal") },
            value = tanggal.value,
            onValueChange = {
                tanggal.value = it
            },
            modifier = Modifier.padding(4.dp).fillMaxWidth(),
            placeholder = { Text(text = "yyyy-mm-dd") }
        )
        OutlinedTextField(
            label = { Text(text = "Keterangan") },
            value = keterangan.value,
            onValueChange = {
                keterangan.value = it
            },
            modifier = Modifier.padding(4.dp).fillMaxWidth(),
            keyboardOptions = KeyboardOptions(capitalization =
                KeyboardCapitalization.Characters, keyboardType = KeyboardType.Text),
            placeholder = { Text(text = "XXXXXX") }
        )
        OutlinedTextField(
            label = { Text(text = "Pemasukan") },
            value = pemasukan.value,
            onValueChange = {
                pemasukan.value = it
            },
            modifier = Modifier.padding(4.dp).fillMaxWidth(),
            keyboardOptions = KeyboardOptions(keyboardType =
                KeyboardType.Decimal),
            placeholder = { Text(text = "5.000.000") }
        )
        OutlinedTextField(
            label = { Text(text = "Pengeluaran") },
            value = pengeluaran.value,
            onValueChange = {
```



FAKULTAS TEKNIK

UNIVERSITAS PASUNDAN

SEMESTER GENAP 2022

```
        pengeluaran.value = it
    },
    modifier = Modifier.padding(4.dp).fillMaxWidth(),
    keyboardOptions = KeyboardOptions(keyboardType =
    KeyboardType.Decimal),
    placeholder = { Text(text = "5.000.000") }
)
val loginButtonColors = ButtonDefaults.buttonColors(
    backgroundColor = Purple700,
    contentColor = Teal200
)
val resetButtonColors = ButtonDefaults.buttonColors(
    backgroundColor = Teal200,
    contentColor = Purple700
)
Row(modifier = Modifier.padding(4.dp).fillMaxWidth()) {
    Button(modifier = Modifier.weight(5f), onClick = {
        val tanggalValue = tanggal.value.text
        val keteranganValue = keterangan.value.text
        val pemasukanValue = pemasukan.value.text
        val pengeluaranValue = pengeluaran.value.text

        // validasi inputan tanggal dan keterangan
        if (tanggalValue.isBlank() || keteranganValue.isBlank()) {
            Toast.makeText(context, "Tanggal dan keterangan harus diisi",
            Toast.LENGTH_SHORT).show()
            return@Button
        }

        // validasi inputan pemasukan dan pengeluaran
        if (pemasukanValue.isBlank() && pengeluaranValue.isBlank()) {
            Toast.makeText(context, "Pemasukan atau pengeluaran harus diisi
            salah satu", Toast.LENGTH_SHORT).show()
            return@Button
        }

        val item = PencatatanKeuangan(tanggalValue, keteranganValue,
        pemasukanValue, pengeluaranValue)
        onSimpan(item)
        tanggal.value = TextFieldValue("")
        keterangan.value = TextFieldValue("")
        pemasukan.value = TextFieldValue("")
        pengeluaran.value = TextFieldValue("")
    }, colors = loginButtonColors) {
        Text(
            text = "Simpan",
            style = TextStyle(
                color = Color.White,
                fontSize = 18.sp
            ), modifier = Modifier.padding(8.dp)
        )
    }
    Button(modifier = Modifier.weight(5f), onClick = {
        tanggal.value = TextFieldValue("")
        keterangan.value = TextFieldValue("")
        pemasukan.value = TextFieldValue("")
        pengeluaran.value = TextFieldValue("")
    }, colors = resetButtonColors) {
```



FAKULTAS TEKNIK

UNIVERSITAS PASUNDAN

SEMESTER GENAP 2022

```
        Text(
            text = "Reset",
            style = TextStyle(
                color = Color.White,
                fontSize = 18.sp
            ), modifier = Modifier.padding(8.dp)
        )
    }
}
```

Screens/PengelolaanKeuangan.Kt

```
package id.ac.unpas.pencatatankeuangan.screens

import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Row
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.items
import androidx.compose.material.Divider
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.collectAsState
import androidx.compose.runtime.getValue
import androidx.compose.runtime.remember
import androidx.compose.ui.Modifier
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import id.ac.unpas.pencatatankeuangan.model.PencatatanKeuangan
import kotlinx.coroutines.flow.MutableStateFlow

@Composable
fun PengelolaanKeuanganScreen() {
    val _list = remember { MutableStateFlow(listOf<PencatatanKeuangan>()) }
    val list by remember { _list }.collectAsState()
    Column(modifier = Modifier.fillMaxWidth()) {
        FormPencatatanKeuangan { item ->
            val newList = ArrayList(list)
            newList.add(item)
            _list.value = newList
        }
        LazyColumn(modifier = Modifier.fillMaxWidth()) {
            items(items = list, itemContent = { item ->
                Row(modifier = Modifier
                    .padding(15.dp)
                    .fillMaxWidth()) {

                    Column(modifier = Modifier.weight(3f)) {
                        Text(text = "Tanggal", fontSize = 14.sp)
                        Text(text = item.tanggal, fontSize = 16.sp,
                            fontWeight = FontWeight.Bold)
                    }
                }
            })
        }
    }
}
```



FAKULTAS TEKNIK UNIVERSITAS PASUNDAN SEMESTER GENAP 2022

```
Column(modifier = Modifier.weight(3f)) {
    Text(text = "Keterangan", fontSize = 14.sp)
    Text(text = item.keterangan, fontSize = 16.sp, fontWeight =
    FontWeight.Bold)
}
Column(modifier = Modifier.weight(3f)) {
    Text(text = "Pemasukan", fontSize = 14.sp)
    Text(text = "${item.pemasukan}", fontSize = 16.sp,
    fontWeight = FontWeight.Bold)
}
Column(modifier = Modifier.weight(3f)) {
    Text(text = "Pengeluaran", fontSize = 14.sp)
    Text(text = "${item.pengeluaran}", fontSize = 16.sp,
    fontWeight = FontWeight.Bold)
}
}
Divider(modifier = Modifier.fillMaxWidth())
})
}
}
```

Hasil

