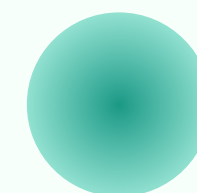
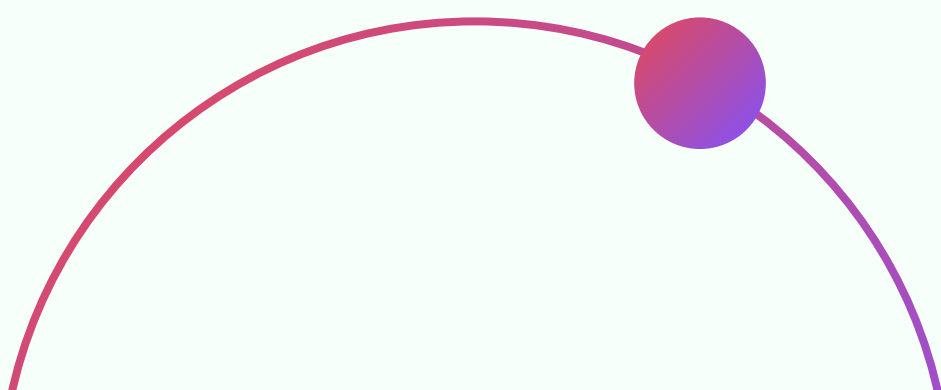




# MOBILE PROGRAMMING

**SCROLLVIEW, SPINNER, AND ARRAYADAPTER IN ANDROID VIEW  
RADIOBUTTON, CHECKBOX, EXPOSEDDROPDOWNMENUBOX IN  
JETPACK COMPOSE**

**ROBBY TAN**



**ScrollView**

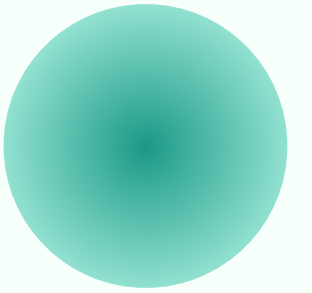




# ScrollView Class

- Sebuah *class* yang digunakan untuk membuat tampilan layar dapat di-*scroll*.
- ScrollView hanya dapat memiliki 1 buah komponen di dalamnya.
  - ScrollView → **EditText, EditText, Button**
  - ScrollView → **Container** (LinearLayout, ConstraintLayout, etc.)
- Digunakan untuk *container* yang memiliki isi cukup banyak, kemudian dilakukan perubahan orientasi (*portrait* → *landscape* atau sebaliknya).

```
01. <?xml version="1.0" encoding="utf-8"?>
02. <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
03.     xmlns:tools="http://schemas.android.com/tools"
04.     android:layout_width="match_parent"
05.     android:layout_height="match_parent"
06.     android:padding="@dimen/padding"
07.     tools:context=".MainActivity">
08.
09.     <LinearLayout
10.         android:id="@+id/ll_root"
11.         android:layout_width="match_parent"
12.         android:layout_height="wrap_content"
13.         android:orientation="vertical">
14.
15.         <EditText
16.             android:id="@+id/et_first_name"
17.             android:layout_width="match_parent"
18.             android:layout_height="wrap_content"
19.             android:hint="@string/text_first_name"
20.             android:inputType="textPersonName" />
21.
22.         <EditText
23.             android:id="@+id/et_last_name"
24.             android:layout_width="match_parent"
25.             android:layout_height="wrap_content"
26.             android:hint="@string/text_last_name"
27.             android:inputType="textPersonName" />
28.
29.     </LinearLayout>
30. </ScrollView>
```

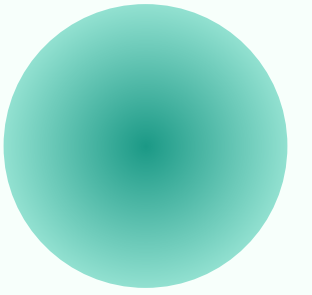


# There is no ScrollView in Jetpack Compose

- Jetpack Compose menghilangkan *class* ScrollView dan menggantinya dengan **Modifier.verticalScroll()** dan **Modifier.horizontalScroll()**.
- Kedua *modifier* ini biasa diletakkan pada *container* (**Column** atau **Row**).
- Contoh implementasi terdapat pada baris ke-22.

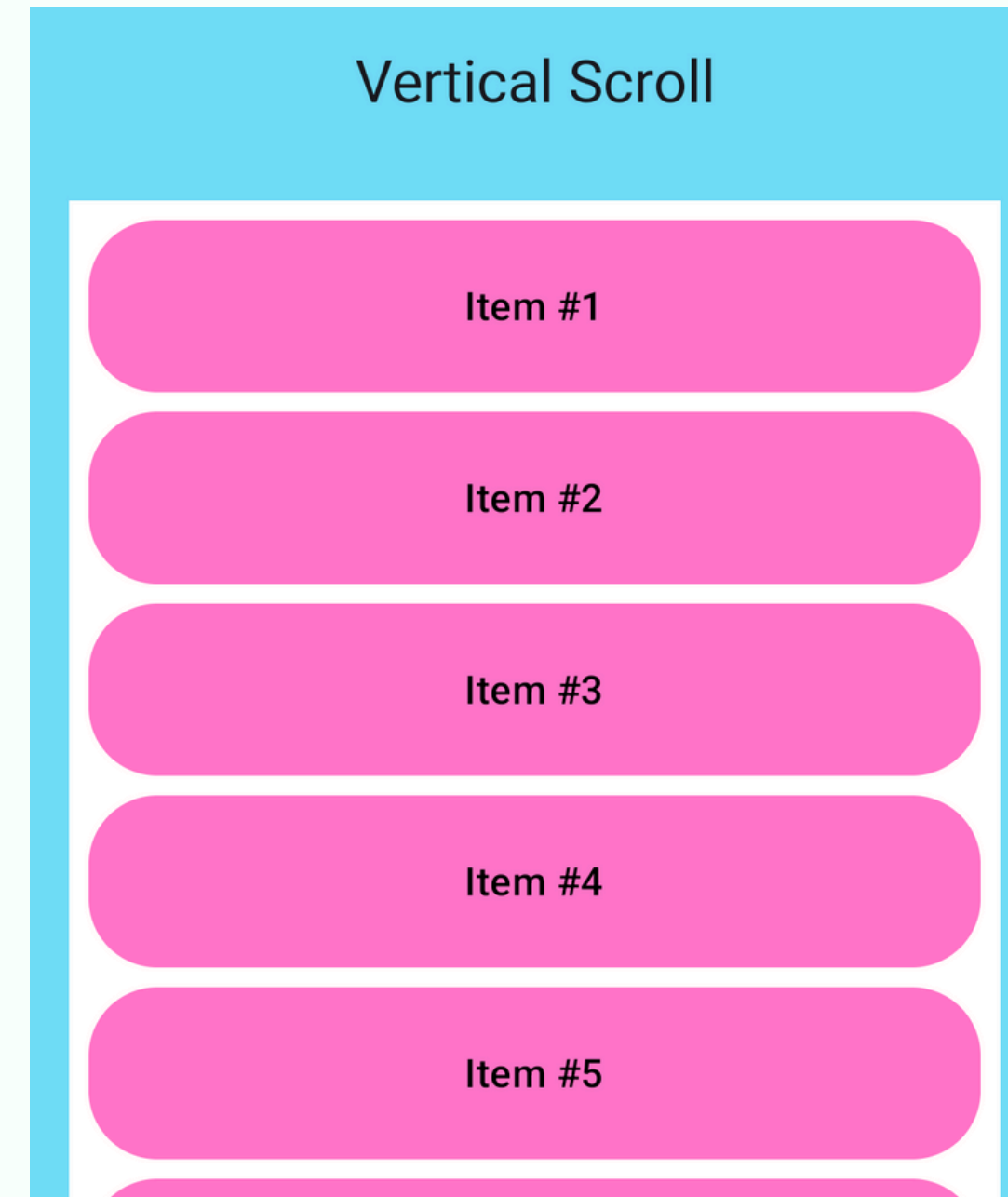
```
1. @Composable
2. fun Card(index: Int) {
3.     Box(
4.         modifier = Modifier
5.             .fillMaxWidth()
6.             .height(70.dp)
7.             .background(Color(0xFFFF76C9), shape = MaterialTheme.shapes.extraLarge),
8.         contentAlignment = Alignment.Center
9.     ) {
10.        Text(
11.            text = "Item #${index}",
12.            style = MaterialTheme.typography.titleMedium,
13.            color = Color.Black
14.        )
15.    }
16. }
```

```
1. @Composable
2. fun VerticalScrollExample(modifier: Modifier = Modifier) {
3.     val scrollState = rememberScrollState()
4.
5.     Column(
6.         modifier = modifier
7.             .fillMaxSize()
8.             .background(Color(0xFF70DCF6))
9.             .padding(16.dp),
10.        horizontalAlignment = Alignment.CenterHorizontally
11.    ) {
12.        Text(
13.            text = "Vertical Scroll",
14.            style = MaterialTheme.typography.headlineSmall,
15.            modifier = Modifier.padding(bottom = 16.dp)
16.        )
17.        Spacer(modifier = Modifier.height(16.dp))
18.        Column(
19.            modifier = Modifier
20.                .fillMaxWidth()
21.                .background(Color.White)
22.                .verticalScroll(scrollState)
23.                .padding(8.dp),
24.            horizontalAlignment = Alignment.CenterHorizontally
25.        ) {
26.            repeat(30) { index ->
27.                Card(index = index + 1)
28.                Spacer(modifier = Modifier.height(8.dp))
29.            }
30.        }
31.    }
32. }
```



# Scroll Example in Jetpack Compose

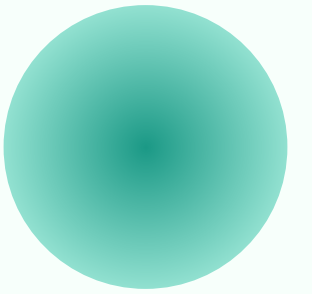
```
1. @Composable
2. fun VerticalScrollExample(modifier: Modifier = Modifier) {
3.     val scrollState = rememberScrollState()
4.
5.     Column(
6.         modifier = modifier
7.         .fillMaxSize()
8.         .background(Color(0xFF70DCF6))
9.         .padding(16.dp),
10.        horizontalAlignment = Alignment.CenterHorizontally
11.    ) {
12.        Text(
13.            text = "Vertical Scroll",
14.            style = MaterialTheme.typography.headlineSmall,
15.            modifier = Modifier.padding(bottom = 16.dp)
16.        )
17.        Spacer(modifier = Modifier.height(16.dp))
18.        Column(
19.            modifier = Modifier
20.            .fillMaxWidth()
21.            .background(Color.White)
22.            .verticalScroll(scrollState)
23.            .padding(8.dp),
24.            horizontalAlignment = Alignment.CenterHorizontally
25.        ) {
26.            repeat(30) { index ->
27.                Card(index = index + 1)
28.                Spacer(modifier = Modifier.height(8.dp))
29.            }
30.        }
31.    }
32. }
```



# **Spinner, CheckBox, RadioButton**







# RadioButton dan CheckBox Class

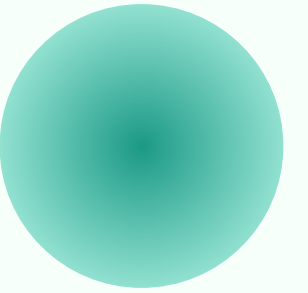
- *Class* yang memiliki fungsi untuk menampilkan pilihan bagi pengguna.
- **RadioButton** → pengguna hanya diizinkan memilih satu pilihan yang disediakan. Penggunaan *class* ini disertai dengan **RadioGroup**.
- **CheckBox** → Pilihan yang dipilih dapat berjumlah lebih dari 1.

```
01. <?xml version="1.0" encoding="utf-8"?>
02. <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
03.     xmlns:tools="http://schemas.android.com/tools"
04.     android:id="@+id/sv_root"
05.     android:layout_width="match_parent"
06.     android:layout_height="match_parent"
07.     android:padding="@dimen/app_padding"
08.     tools:context=".MainActivity">
09.
10.     <LinearLayout
11.         android:layout_width="match_parent"
12.         android:layout_height="wrap_content"
13.         android:orientation="vertical">
14.
15.         <CheckBox
16.             android:id="@+id/cb_disability"
17.             android:layout_width="match_parent"
18.             android:layout_height="wrap_content"
19.             android:text="@string/text_disability" />
20.
21.         <RadioGroup
22.             android:layout_width="wrap_content"
23.             android:layout_height="wrap_content"
24.             android:orientation="horizontal">
25.
26.             <RadioButton
27.                 android:id="@+id/rb_male"
28.                 android:layout_width="match_parent"
29.                 android:layout_height="wrap_content"
30.                 android:checked="true"
31.                 android:text="@string/text_male" />
32.
33.             <RadioButton
34.                 android:id="@+id/rb_female"
35.                 android:layout_width="match_parent"
36.                 android:layout_height="wrap_content"
37.                 android:text="@string/text_female" />
38.         </RadioGroup>
39.     </LinearLayout>
40. </ScrollView>
```



# Spinner Class

- Sebuah *class* yang menampilkan daftar item.
- Padanan dalam HTML adalah `<select>-<option>`.
- Sumber data:
  - String resources dalam bentuk array (**<string-array>**)
  - Kotlin Collection (List)
- Data akan dimasukkan menjadi objek **ArrayAdapter** yang kemudian diset menjadi bagian dari Spinner



10:50

NRP (e.g. 1772999)

Your Name

Your Address

Your Email (e.g. someone@email.com)

☐ Disability

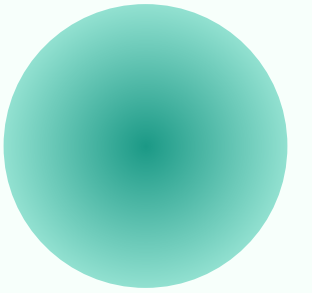
☒ Male ☐ Female

Scholarship BC Foundation ▼

S1 Kedokteran ▼

SUBMIT

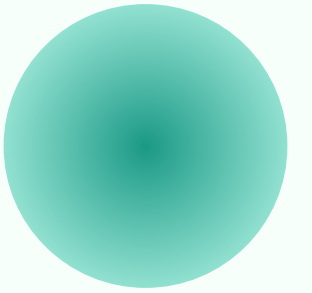




# Simple Project Class Source Code (1/2)

```
01. <?xml version="1.0" encoding="utf-8"?>
02. <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
03.     xmlns:tools="http://schemas.android.com/tools"
04.     android:id="@+id/sv_root"
05.     android:layout_width="match_parent"
06.     android:layout_height="match_parent"
07.     android:padding="@dimen/app_padding"
08.     tools:context=".MainActivity">
09.
10.     <LinearLayout
11.         android:layout_width="match_parent"
12.         android:layout_height="wrap_content"
13.         android:orientation="vertical">
14.
15.         <EditText
16.             android:id="@+id/et_nrp"
17.             android:layout_width="match_parent"
18.             android:layout_height="wrap_content"
19.             android:autofillHints=""
20.             android:hint="@string/text_nrp"
21.             android:inputType="number" />
22.
23.         <EditText
24.             android:id="@+id/et_name"
25.             android:layout_width="match_parent"
26.             android:layout_height="wrap_content"
27.             android:autofillHints="name"
28.             android:hint="@string/text_name"
29.             android:inputType="textPersonName" />
30.
31.         <EditText
32.             android:id="@+id/et_address"
33.             android:layout_width="match_parent"
34.             android:layout_height="wrap_content"
35.             android:autofillHints="postalAddress"
36.             android:hint="@string/text_address"
37.             android:inputType="textPostalAddress|textMultiline" />
38.
39.         <EditText
40.             android:id="@+id/et_email"
41.             android:layout_width="match_parent"
42.             android:layout_height="wrap_content"
43.             android:autofillHints="emailAddress"
44.             android:hint="@string/text_email"
45.             android:inputType="textEmailAddress" />
46.
```

```
47.         <CheckBox
48.             android:id="@+id/cb_disability"
49.             android:layout_width="match_parent"
50.             android:layout_height="wrap_content"
51.             android:text="@string/text_disability" />
52.
53.         <RadioGroup
54.             android:layout_width="wrap_content"
55.             android:layout_height="wrap_content"
56.             android:orientation="horizontal">
57.
58.             <RadioButton
59.                 android:id="@+id/rb_male"
60.                 android:layout_width="match_parent"
61.                 android:layout_height="wrap_content"
62.                 android:checked="true"
63.                 android:text="@string/text_male" />
64.
65.             <RadioButton
66.                 android:id="@+id/rb_female"
67.                 android:layout_width="match_parent"
68.                 android:layout_height="wrap_content"
69.                 android:text="@string/text_female" />
70.         </RadioGroup>
71.
72.         <Spinner
73.             android:id="@+id/spin_scholarship"
74.             android:layout_width="match_parent"
75.             android:layout_height="wrap_content" />
76.
77.         <Spinner
78.             android:id="@+id/spin_departments"
79.             android:layout_width="match_parent"
80.             android:layout_height="wrap_content" />
81.
82.         <Button
83.             android:id="@+id/btn_submit"
84.             android:layout_width="match_parent"
85.             android:layout_height="wrap_content"
86.             android:text="@string/action_submit" />
87.
88.     </LinearLayout>
89.
90. </ScrollView>
```



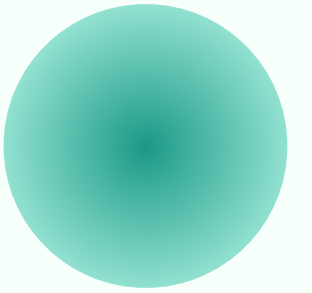
# Simple Project Class Source Code (2/2)

```
1. class MainActivity : AppCompatActivity() {
2.
3.     private lateinit var binding: ActivityMainBinding
4.     private var departmentArrayAdapter: ArrayAdapter<Department>? = null
5.     private var scholarshipArrayAdapter: ArrayAdapter<CharSequence>? = null
6.
7.     override fun onCreate(savedInstanceState: Bundle?) {
8.         super.onCreate(savedInstanceState)
9.         binding = ActivityMainBinding.inflate(layoutInflater)
10.        val view: View? = binding.getRoot()
11.        setContentView(view)
12.        binding.spinDepartments.setAdapter(getDepartmentArrayAdapter())
13.        binding.spinScholarship.setAdapter(getScholarshipArrayAdapter())
14.        binding.btnSubmit.setOnClickListener { v ->
15.            this.submitAction()
16.        }
17.    }
18.
19.    private fun submitAction() {
20.        val context = applicationContext
21.        val nrp: String? = binding.etNrp.getText().toString()
22.        val name: String? = binding.etName.getText().toString()
23.        val address: String? = binding.etAddress.getText().toString()
24.        val email: String? = binding.etEmail.getText().toString()
25.        val disability: Boolean = binding.cbDisability.isChecked
26.        val gender: String = if (binding.rbMale.isSelected) {
27.            binding.rbMale.text.toString()
28.        } else {
29.            binding.rbFemale.getText().toString()
30.        }
31.        val department: Department = binding.spinDepartments.getSelectedItem() as Department
32.        Toast.makeText(
33.            context,
34.            "$name ($nrp) $address $email $disability $gender $department",
35.            Toast.LENGTH_LONG
36.        ).show()
37.    }
38.
39.    fun getDepartments(): List<Department> {
40.        return listOf(
41.            Department("10", "S1 Kedokteran"),
42.            Department("72", "S1 Teknik Informatika"),
43.            Department("73", "S1 Sistem Informasi"),
44.            Department("74", "S2 Ilmu Komputer")
45.        )
46.    }
```

```
47.
48.    fun getDepartmentArrayAdapter(): ArrayAdapter<Department> {
49.        if (departmentArrayAdapter == null) {
50.            departmentArrayAdapter = ArrayAdapter(
51.                this,
52.                android.R.layout.simple_spinner_dropdown_item,
53.                getDepartments()
54.            )
55.        }
56.        return departmentArrayAdapter!!
57.    }
58.
59.    fun getScholarshipArrayAdapter(): ArrayAdapter<CharSequence>? {
60.        if (scholarshipArrayAdapter == null) {
61.            scholarshipArrayAdapter = ArrayAdapter.createFromResource(
62.                this,
63.                R.array.scholarships_array,
64.                android.R.layout.simple_spinner_item
65.            )
66.            scholarshipArrayAdapter?.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
67.        }
68.        return scholarshipArrayAdapter
69.    }
70. }
```

**RadioButton, CheckBox,  
ExposableDropDownMenu**





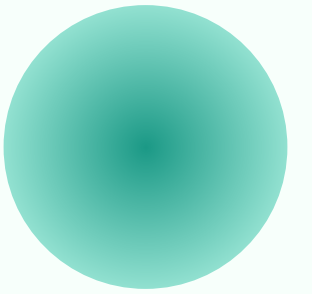
# RadioButton & CheckBox (Jetpack Compose)

```
1. @Composable
2. fun RadioCheckExample(modifier: Modifier = Modifier) {
3.     var name by remember { mutableStateOf("") }
4.     var isDisabilityChecked by remember { mutableStateOf(false) }
5.     var selectedGender by remember { mutableStateOf("Male") }
6.     val context = LocalContext.current
7.
8.     Column(
9.         modifier = modifier
10.        .fillMaxSize()
11.        .padding(16.dp),
12.        verticalArrangement = Arrangement.Top
13.    ) {
14.
15.        TextField(
16.            value = name,
17.            onValueChange = { name = it },
18.            label = { Text("Name") },
19.            keyboardOptions = KeyboardOptions(
20.                keyboardType = KeyboardType.Text,
21.                imeAction = ImeAction.Next
22.            ),
23.            modifier = Modifier.fillMaxWidth()
24.        )
25.
26.        Row(verticalAlignment = Alignment.CenterVertically) {
27.            Checkbox(
28.                checked = isDisabilityChecked,
29.                onCheckedChange = { isDisabilityChecked = it }
30.            )
31.            Text("Disability")
32.        }
33.
34.        Text("Gender")
35.        Row {
36.            Row(verticalAlignment = Alignment.CenterVertically) {
37.                RadioButton(
38.                    selected = selectedGender == "Male",
39.                    onClick = { selectedGender = "Male" }
40.                )
41.                Text("Male")
42.            }
43.            Row(verticalAlignment = Alignment.CenterVertically) {
44.                RadioButton(
45.                    selected = selectedGender == "Female",
46.                    onClick = { selectedGender = "Female" }
47.                )
48.                Text("Female")
49.            }
50.        }
51.    }
```

```
52.        Button(
53.            onClick = {
54.                Toast.makeText(
55.                    context,
56.                    "Name: $name. Gender: $selectedGender. Disability Status: ${if (isDisabilityChecked) "Yes" else "No"}",
57.                    Toast.LENGTH_LONG
58.                ).show()
59.            },
60.            modifier = Modifier.fillMaxWidth()
61.        ) {
62.            Text("Submit")
63.        }
64.    }
65. }
```

The UI mockup displays a form with a light purple background. It features a text input field for 'Name', a checkbox for 'Disability' (which is currently unchecked), and two radio buttons for 'Gender' (with 'Male' selected). At the bottom is a dark blue rounded button labeled 'Submit'.

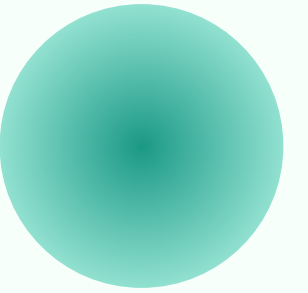




# ExposableDropDownMenuBox (1/2)

- Composable pengganti Spinner
- Tidak menggunakan ArrayAdapter sebagai sumber data penyimpanan.

```
1. @OptIn(ExperimentalMaterial3Api::class)
2. @Composable
3. fun SimpleExposedDropDownMenuExample(modifier: Modifier = Modifier) {
4.     val languages = listOf("Kotlin", "Java", "Python", "Swift", "Dart", "JavaScript")
5.     var expanded by remember { mutableStateOf(false) }
6.     var selectedText by remember { mutableStateOf(languages[0]) }
7.
8.     Column(
9.         modifier = modifier
10.        .fillMaxSize()
11.        .padding(16.dp)
12.        .background(Color(0xFFE2FCC1)),
13.        horizontalAlignment = Alignment.CenterHorizontally,
14.        verticalArrangement = Arrangement.Center
15.    ) {
16.        Text(
17.            text = "Choose your programming language",
18.            style = MaterialTheme.typography.titleMedium
19.        )
20.        Spacer(modifier = Modifier.height(16.dp))
21.
22.        ExposedDropDownMenuBox(
23.            expanded = expanded,
24.            onExpandedChange = {
25.                expanded = !expanded
26.            },
27.            modifier = Modifier.width(250.dp)
28.        ) {
29.            OutlinedTextField(
30.                modifier = Modifier
31.                    .menuAnchor(type = MenuAnchorType.PrimaryNotEditable)
32.                    .fillMaxWidth(),
33.                readOnly = true,
34.                value = selectedText,
35.                onValueChange = {},
36.                label = { Text("Programming Language") },
37.                trailingIcon = { ExposedDropDownMenuDefaults.TrailingIcon(expanded = expanded) },
38.                colors = ExposedDropDownMenuDefaults.outlinedTextFieldColors(),
39.            )
40.        }
```



# ExposableDropDownMenuBox (2/2)

```
42.         ExposedDropDownMenu(  
43.             expanded = expanded,  
44.             onDismissRequest = {  
45.                 expanded = false  
46.             }  
47.         ) {  
48.             languages.forEach { selectionOption ->  
49.                 DropdownMenuItem(  
50.                     text = { Text(selectionOption) },  
51.                     onClick = {  
52.                         selectedText = selectionOption  
53.                         expanded = false  
54.                     },  
55.                 )  
56.             }  
57.         }  
58.     }  
59.  
60.     Spacer(modifier = Modifier.height(32.dp))  
61.  
62.     Text(text = "Your choice: $selectedText", style = MaterialTheme.typography.bodyLarge)  
63. }  
64. }
```

Choose your programming language

Programming Language

Kotlin



Your choice: Kotlin

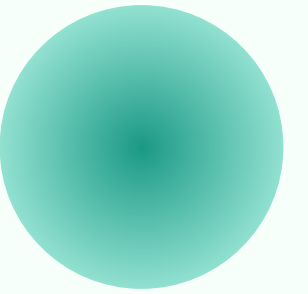


# Exercise



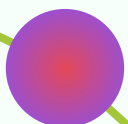


# Exercise



Buatlah sebuah aplikasi Android dengan ketentuan sebagai berikut:

1. Menggunakan seluruh Composable *class* yang sudah dipelajari.
2. Aplikasi terdiri dari 2 halaman
  - a. Halaman Utama (terdapat minimal 2 tombol)
  - b. Halaman input data
  - c. Halaman yang menampilkan data
3. Contoh studi kasus yang dapat dibuat (pilih salah satu)
  - a. Data Guru dan Siswa
  - b. Data Pasien
  - c. Data Mobil/ Motor





# Thank You

ROBBY.TAN@IT.MARANATHA.EDU