



Laporan Praktikum

Mata pelajaran Konsentrasi Keahlian RPL

Materi Ajar :

KOTLIN & JETPACK COMPOSE: MEMBUAT APLIKASI PROFIL DIRI

No.	Nama	NIS	Kelas
1.	AVIF MUHAMAD FADILAH	20356	XI RPL

Pengembangan Perangkat Lunak dan Gim (PPLG)

SMK Negeri 1 Bawang

2026

JOBSHEET PRAKTIKUM KOTLIN & JETPACK COMPOSE

Mata Pelajaran: Konsentrasi Keahlian RPL

Materi Pokok: Membuat Aplikasi Profil Diri dengan Jetpack Compose (Kotlin)

Kelas / Semester: XI RPL / Ganjil

Alokasi Waktu: 2 x 45 Menit (1 Kali Pertemuan)

I. KOMPETENSI DASAR & INDIKATOR KETERCAPAIAN

- **Kompetensi Dasar:** Menerapkan pembuatan antarmuka (UI) aplikasi Android menggunakan Jetpack Compose berbasis bahasa Kotlin.
- **Indikator:**
 1. Siswa mampu membuat proyek baru Android Studio dengan template Empty Activity (Jetpack Compose).
 2. Siswa mampu menambahkan Vector Asset (ikon) ke dalam resource proyek.
 3. Siswa mampu menyusun komponen UI (Column, Icon, Text, Button) menjadi halaman profil diri.
 4. Siswa mampu mengidentifikasi dan memperbaiki error pada saat proses build hingga aplikasi berjalan sukses.

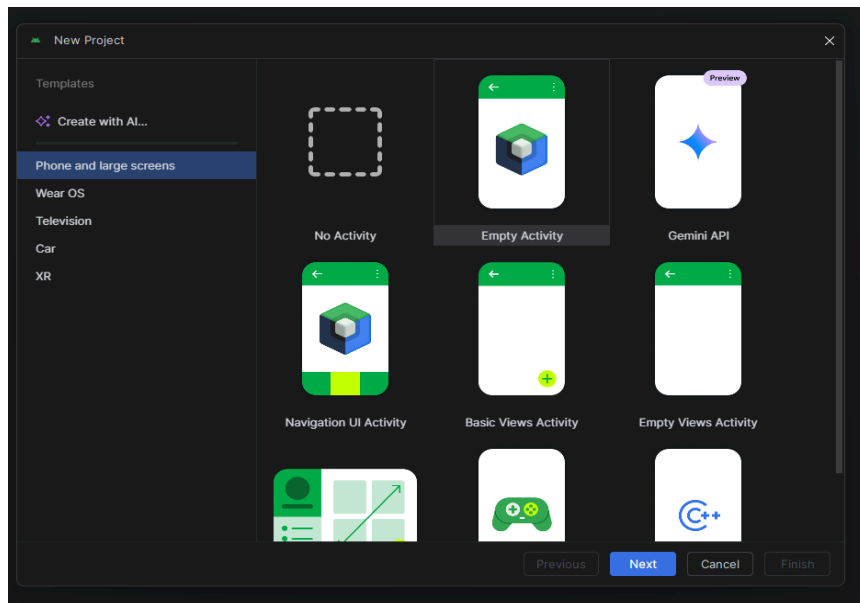
II. ALAT DAN BAHAN

1. Komputer/Laptop dengan Sistem Operasi Windows/Linux/macOS.
2. Android Studio (Meerkat/Narwhal atau versi terbaru) dengan plugin Kotlin & Jetpack Compose.
3. Android Virtual Device (AVD) / Emulator, contoh: **Medium Phone API 29**.
4. Berkas aset gambar (SVG) untuk dijadikan ikon aplikasi.
5. Aset figma yang sudah dibuat.

III. LANGKAH KERJA & DOKUMENTASI

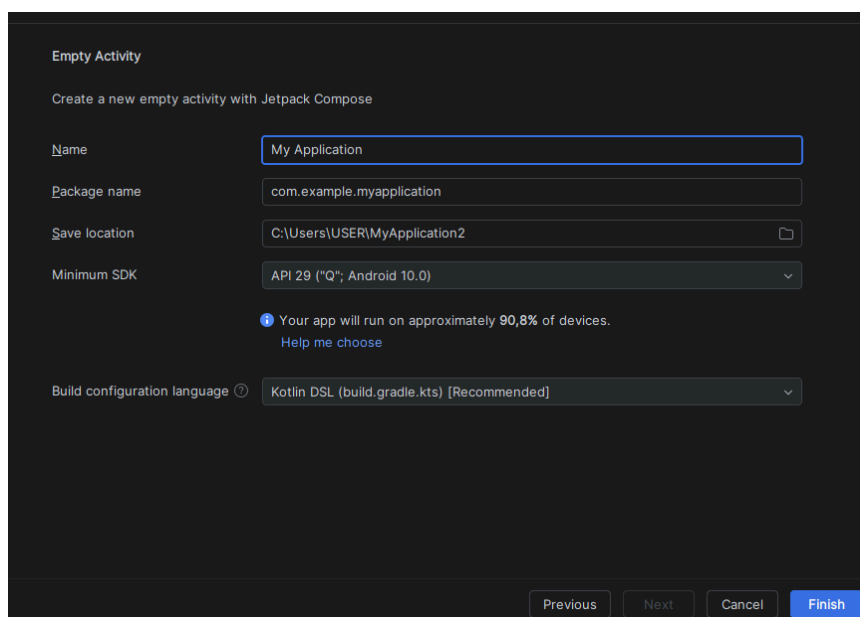
Langkah 1: Membuat Proyek Baru

1. Buka Android Studio, pilih **File > New > New Project**.
2. Pada kategori **Phone and large screens**, pilih template **Empty Activity**, lalu klik **Next**.



Gambar 1. Memilih template Empty Activity pada New Project

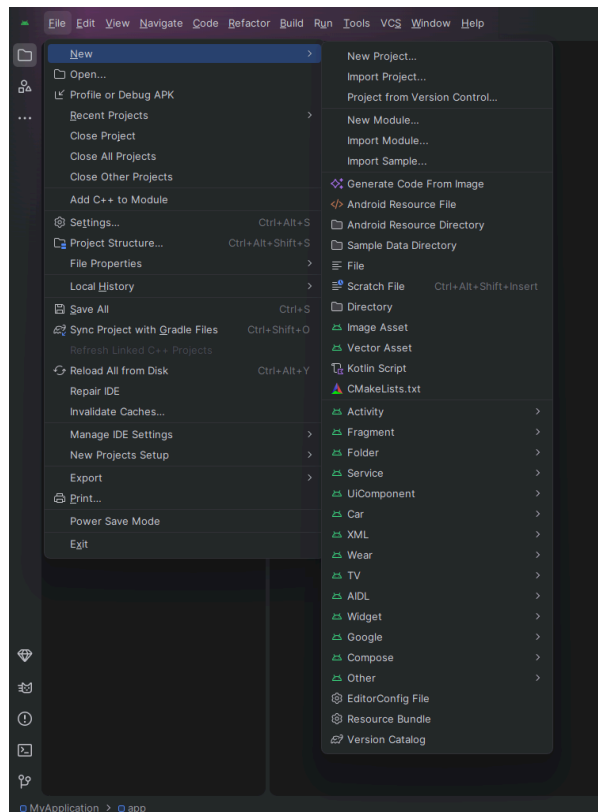
- Isi konfigurasi proyek: **Name = My Application**, **Package name = com.example.myapplication**, **Minimum SDK = API 29 (Android 10.0)**, dan bahasa konfigurasi build **Kotlin DSL (build.gradle.kts)**. Klik **Finish**.



Gambar 2. Konfigurasi nama, package, dan minimum SDK proyek

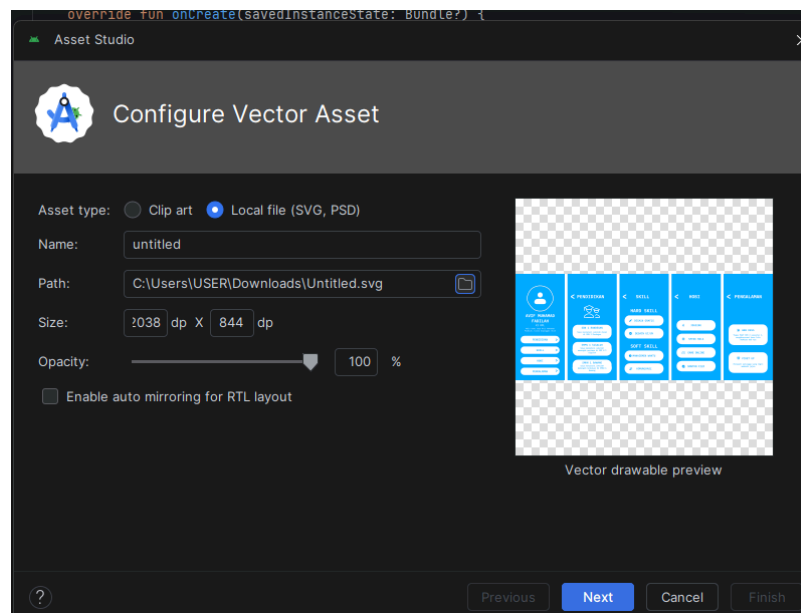
Langkah 2: Menambahkan Vector Asset (Ikon)

- Klik kanan pada folder **res**, pilih **New > Vector Asset** (dapat juga diakses lewat menu **File > New**).



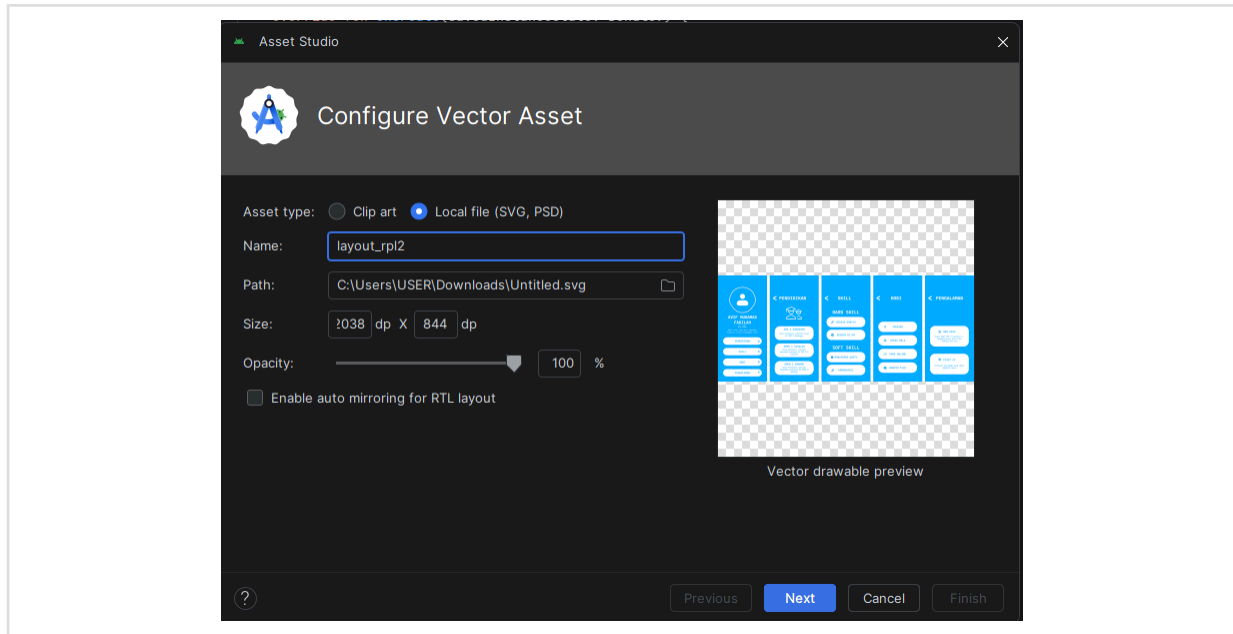
Gambar 3. Mengakses menu New untuk membuat resource/asset baru

2. Pilih opsi **Local file (SVG, PSD)**, kemudian arahkan **Path** ke berkas SVG ikon yang telah disiapkan.



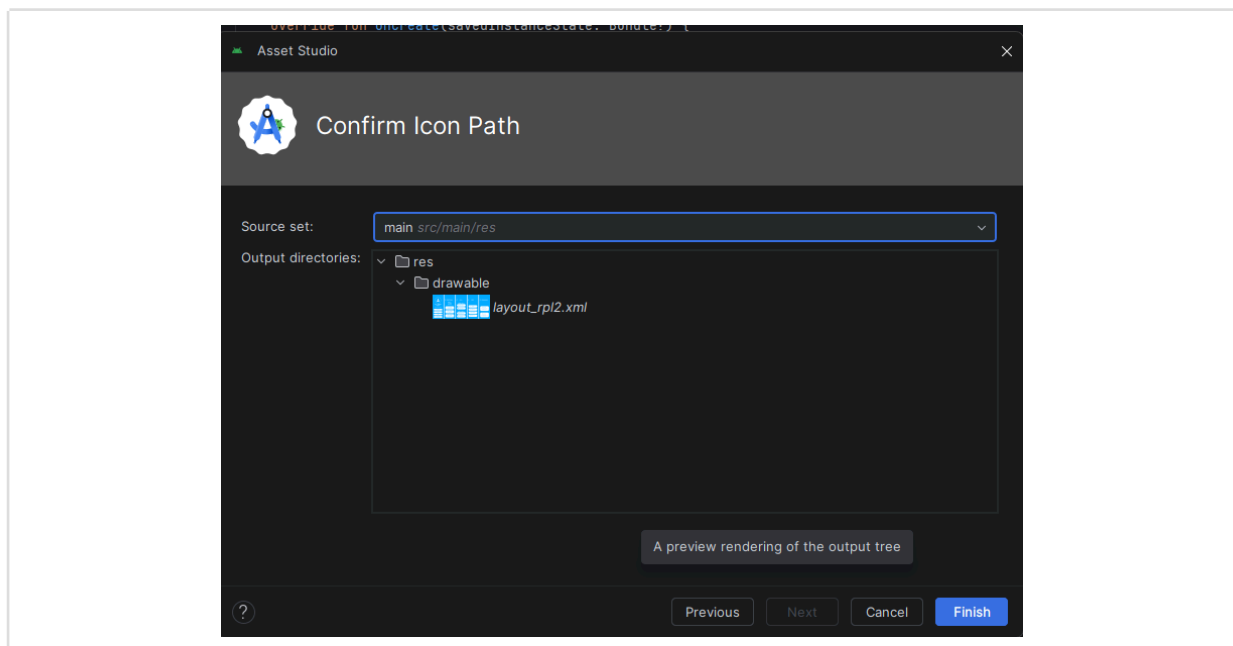
Gambar 4. Memilih berkas SVG lokal sebagai Vector Asset

3. Ubah **Name** aset menjadi **layout_rp12** agar mudah dikenali, lalu klik **Next**.



Gambar 5. Penamaan Vector Asset sebelum disimpan

4. Pada jendela **Confirm Icon Path**, pastikan output tersimpan di **res/drawable/layout_rpl2.xml**, lalu klik **Finish**.



Gambar 6. Konfirmasi lokasi penyimpanan Vector Asset

Langkah 3: Menulis Kode Tampilan (UI) pada MainActivity.kt

Buat komponen Composable untuk menyusun halaman profil, contoh potongan kode halaman “Pendidikan”:

```
@Composable
fun HalamanPendidikan(navController: NavController) {
    Column(
        modifier = Modifier
            .fillMaxSize()
            .background(Color(0xFF00A2FF))
    )
```

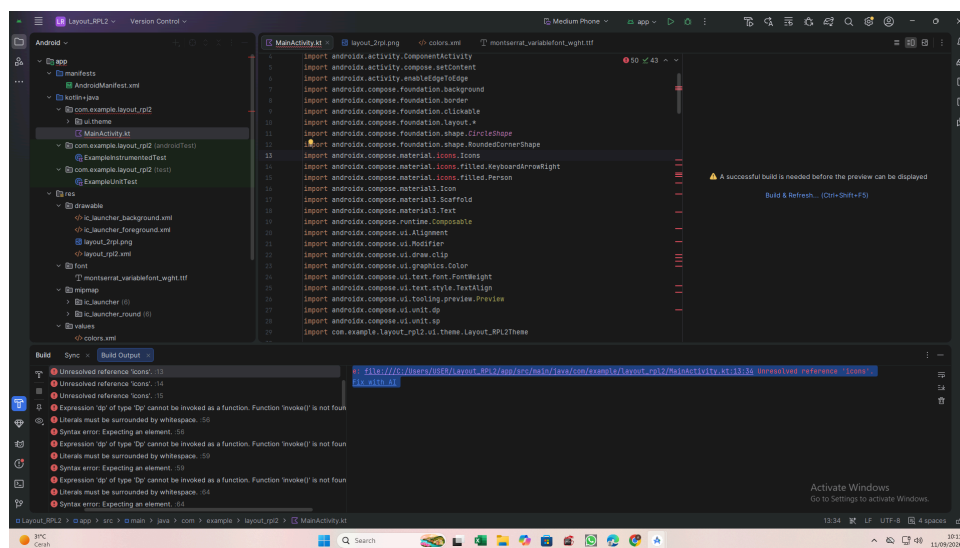
```

        .padding(horizontal = 24.dp)
        .verticalScroll(state = rememberScrollState()),
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        SubPageHeader(title = "PENDIDIKAN", navController = navController)
        Spacer(modifier = Modifier.height(16.dp))
        Icon(
            imageVector = Icons.Default.School,
            contentDescription = null,
            tint = Color.White,
            modifier = Modifier.size(80.dp)
        )
        Spacer(modifier = Modifier.height(32.dp))
        InfoCard(title = "SDN 1 RANDEGAN", description = "...")
        InfoCard(title = "SMPN 1 SIGALUH", description = "...")
        InfoCard(title = "SMKN 1 BAWANG", description = "...")
    }
}

```

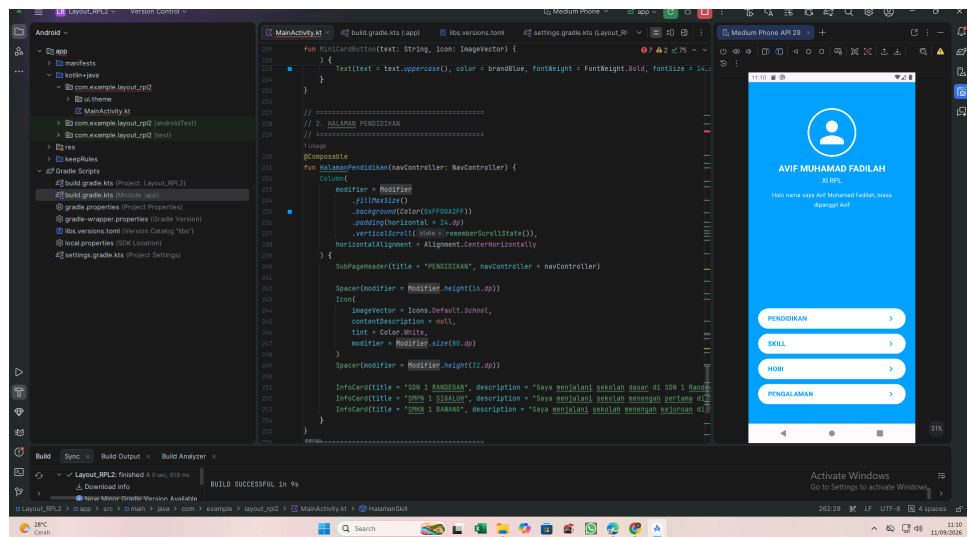
Langkah 4: Menjalankan Build & Menangani Error

1. Jalankan **Build > Rebuild Project**. Pada percobaan pertama, build gagal karena **import** library ikon (**androidx.compose.material.icons.Icons**) belum lengkap serta ada kesalahan penulisan pada nilai **dp** (kurang spasi), sehingga muncul beberapa **Unresolved reference** dan **Syntax error**.



Gambar 7. Log error pada Build Output sebelum diperbaiki

2. Perbaiki dengan melengkapi **import androidx.compose.material.icons.filled.*** dan menambahkan spasi pada setiap nilai satuan (**dp**), kemudian lakukan **Sync Project with Gradle Files** dan **Rebuild Project** kembali.
3. Jalankan aplikasi pada emulator **Medium Phone API 29**. Build berhasil (**BUILD SUCCESSFUL**) tanpa error, dan tampilan halaman profil (Pendidikan, Skill, Hobi, Pengalaman) tampil dengan benar pada preview.



Gambar 8. Build berhasil (BUILD SUCCESSFUL) dan tampilan aplikasi berjalan sesuai rancangan