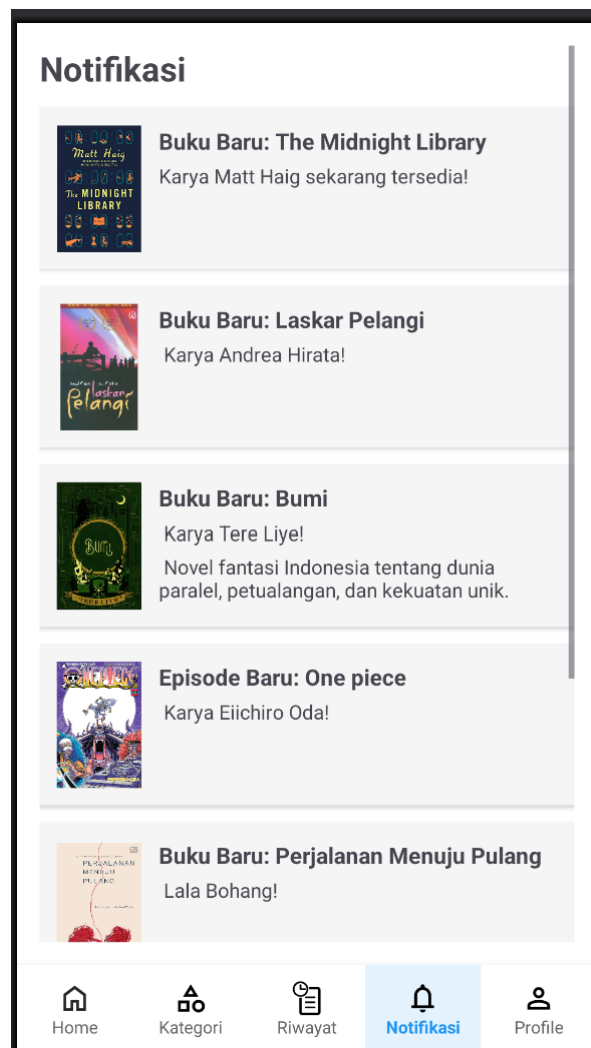


Nama : Defri Liyadi

NPM : 23312244

Kelas : IF 23 FX

1.



Pada tampilan ini, saya membuat layout halaman **Notifikasi** untuk aplikasi LiterKu. Layout ini dirancang agar pengguna dapat dengan mudah melihat informasi terbaru terkait buku atau episode baru yang tersedia.

Beberapa poin penting dari hasil kerja saya

1. **Judul Halaman**

Di bagian atas terdapat judul "Notifikasi" sebagai identitas halaman agar pengguna tahu konteks dari informasi yang ditampilkan.

2. **Desain Kartu Notifikasi**

Setiap notifikasi ditampilkan dalam bentuk kartu berisi:

- Gambar sampul buku atau episode
- Judul notifikasi (misalnya "Buku Baru: The Midnight Library")
- Penulis atau deskripsi singkat dari buku/episode tersebut

3. **Informasi yang Bervariasi**

Notifikasi mencakup berbagai jenis konten, seperti:

- Buku baru dari berbagai penulis (Matt Haig, Andrea Hirata, Tere Liye, Lala Bohang)
- Episode baru dari komik seperti "One Piece"

4. **Scrolling**

Saya membuat beberapa notifikasi dan juga agar layarnya bisa discrol agar memudahkan para pengguna melihat notifikasi yang lebih awal

5. **Navigasi Bawah (Bottom Navigation Bar)**

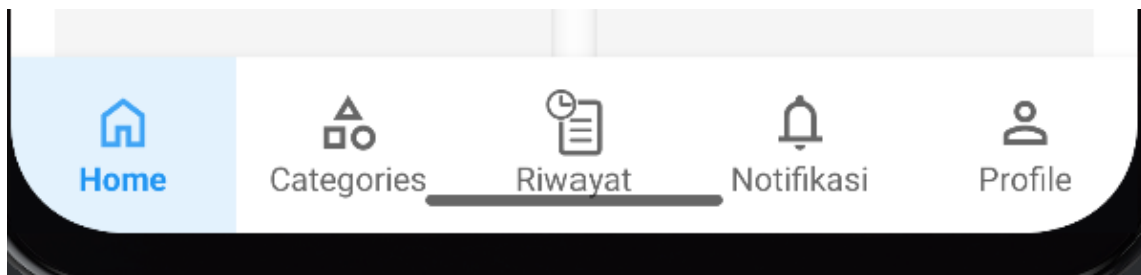
Saya juga menyertakan navbar bawah agar pengguna bisa dengan mudah berpindah antar halaman, seperti Home, Kategori, Riwayat, Notifikasi, dan Profil.

Tampilan ini dibuat dengan pendekatan user-friendly, informatif, dan visual yang menarik agar pengguna tetap up-to-date dengan koleksi terbaru dari aplikasi.

2. Saya membuat codingan binding, agar untuk ke halaman atau layout yang lain hanya tinggal mengklik navbar yang di bawah saja

```
5 class HomeActivity : AppCompatActivity() {
6     override fun onCreate(savedInstanceState: Bundle?) {
7         super.onCreate(savedInstanceState)
8         binding = ActivityHomeBinding.inflate(layoutInflater)
9         setContentView(binding.root)
10        updateNavbarSelection(R.id.layout_home)
11
12        // Home
13        binding.bottomNavigation.findViewById<LinearLayout>(R.id.layout_home).setOnClickListener {
14            updateNavbarSelection(R.id.layout_home)
15            binding.mainScrollView.visibility = View.VISIBLE
16            supportFragmentManager.popBackStack(
17                name: null,
18                androidx.fragment.app.FragmentManager.POP_BACK_STACK_INCLUSIVE
19            )
20        }
21
22        // Kategori
23        binding.bottomNavigation.findViewById<LinearLayout>(R.id.layout_categories).setOnClickListener {
24            updateNavbarSelection(R.id.layout_categories)
25            binding.mainScrollView.visibility = View.GONE
26
27            // Hapus semua fragment sebelum berpindah
28            supportFragmentManager.popBackStack(
29                name: null,
30                androidx.fragment.app.FragmentManager.POP_BACK_STACK_INCLUSIVE
31            )
32
33            val fragment = KategoriFragment()
34            supportFragmentManager.beginTransaction()
35                .replace(R.id.fragment_container, fragment)
36                .addToBackStack(name: null)
37                .commit()
38        }
39    }
40}
```

Seperti ini lah bentukan navbarnya



3. Kemudian saya membuat codingan logika seperti dibawah

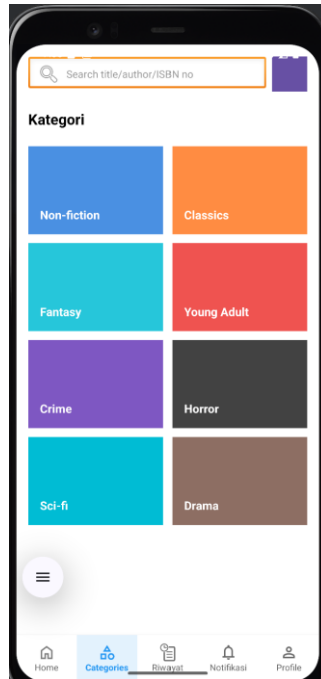
```
        .commit()
    }
    // Kembali otomatis ke home jika backstack kosong
    supportFragmentManager.addOnBackStackChangeListener {
        if (supportFragmentManager.backStackEntryCount == 0) {
            binding.mainScrollView.visibility = View.VISIBLE
            updateNavbarSelection(R.id.layout_home)
        }
    }
}
```

Tujuannya agar Ketika kita berada di halaman atau layout selain di halaman home, maka Ketika kita mengklik tombol back dia akan Kembali ke halaman home, bukan ke halaman sebelumnya

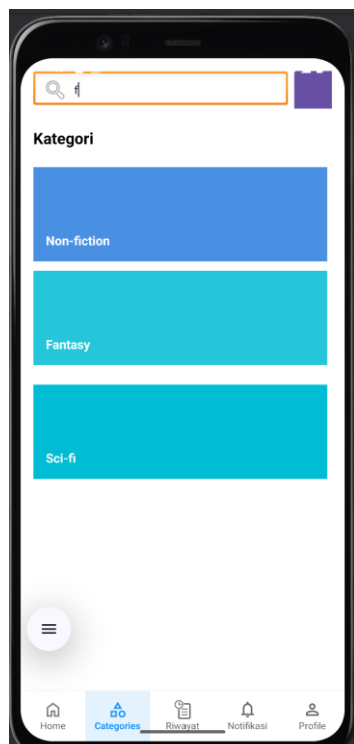
Contoh

Awal kita berada di halaman home, kemudian kita mengklik halaman kategori di bagian navbar, otomatis kita akan berada di halaman kategori, kemudian kita klik lagi di bagian navbar yaitu Riwayat, maka kita akan berada di halaman Riwayat, dan selanjutnya kita akan mengklik tombol back, dengan codingan di atas kita akan langsung dibawa Kembali ke halaman home, bukan ke halaman sebelumnya yaitu halaman kategori.

4. Kemudian saya membuat Di halaman Kategori Buku, saya menambahkan fitur pencarian yang memungkinkan pengguna untuk mencari kategori buku secara lebih cepat dan efisien. Dengan fitur ini, pengguna cukup mengetikkan kata kunci pada kolom pencarian, lalu sistem akan secara otomatis menampilkan kategori yang sesuai dengan input tersebut. Fitur ini dirancang untuk memudahkan navigasi dan mempercepat proses menemukan kategori buku yang diinginkan tanpa harus menggulir halaman secara manual.

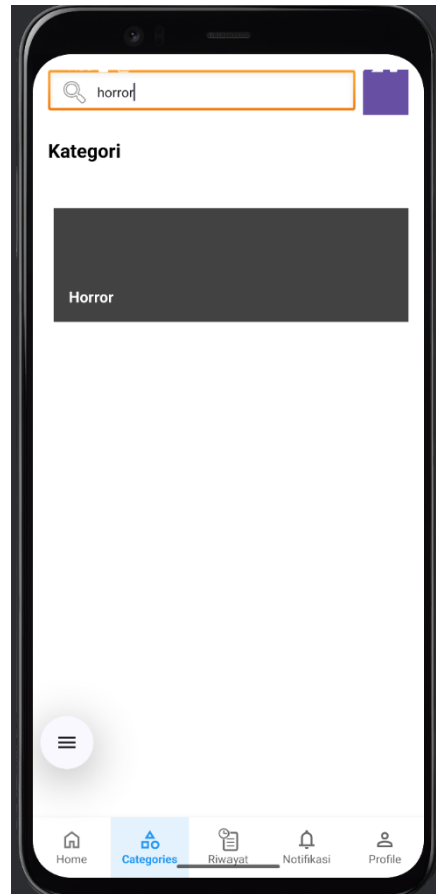
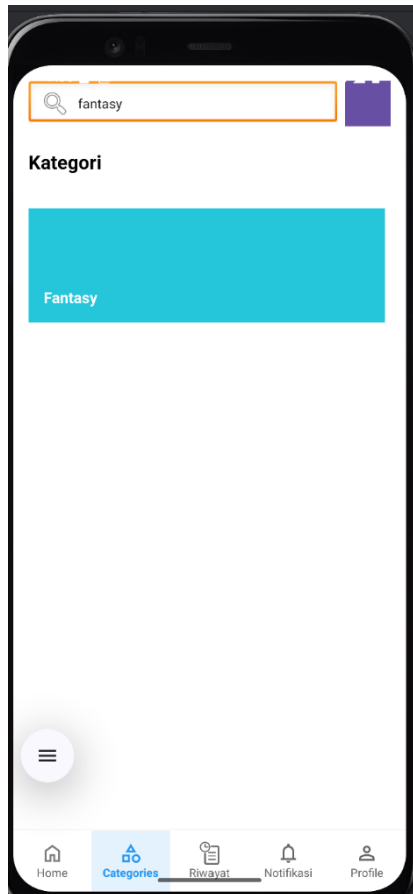


Tampilan di atas adalah halaman kategori buku, dan apabila kita ingin mencari kategori buku tinggal mengklik tombol pencarian di atas



Itu contoh Ketika kita mencari kategori dengan 1 huruf, maka nama kategori yg berkaitan dengan huruf tersebut akan muncul, dan yang tidak ada ikatan akan tidak muncul

Kemudian apabila kita mengetik lebih spesifik maka dia akan muncul sesuai yang kita ketik contoh seperti di bawah



Adapun codingan untuk logika pencarian di atas adalah sebagai berikut

```
3 class KategoriFragment : Fragment() {
4     override fun onCreateView(
5     )
6     override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
7         super.onViewCreated(view, savedInstanceState)
8
9         // Deteksi teks yang diketik di kolom pencarian
10        binding.searchEditText.addTextChangedListener(object : TextWatcher {
11            override fun beforeTextChanged(s: CharSequence?, start: Int, count: Int, after: Int) {}
12            override fun afterTextChanged(s: Editable?) {}
13
14            override fun onTextChanged(s: CharSequence?, start: Int, before: Int, count: Int) {
15                filterKategori(s.toString())
16            }
17        })
18    }
19
20    private fun filterKategori(query: String) {
21        val q = query.lowercase()
22
23        val kategoriList = listOf(
24            Pair(binding.kategoriNonfiction, "non-fiction"),
25            Pair(binding.kategoriClassics, "classics"),
26            Pair(binding.kategoriFantasy, "fantasy"),
27            Pair(binding.kategoriYoungAdult, "young adult"),
28            Pair(binding.kategoriCrime, "crime"),
29            Pair(binding.kategoriHorror, "horror"),
30            Pair(binding.kategoriScifi, "sci-fi"),
31            Pair(binding.kategoriDrama, "drama")
32        )
33
34        for ((view, keyword) in kategoriList) {
35            view.visibility = if (keyword.contains(q, ignoreCase = true)) View.VISIBLE else View.GONE
36        }
37    }
38 }
```

5. Saya juga membuat background dari navbar menjadi warna biru dan teks nya menjadi warna biru serta bold Ketika memasuki dari masing-masing halaman tersebut

