**LAPORAN PRAKTIKUM**

**PEMROGRAMAN PERANGKAT BERGERAK**

****

**JUDUL :**

**SQLITE CARD VIEW**

Disusun oleh:

Usman Pamungkas (20102198)

**TEKNIK INFORMATIKA**

**INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

**BANYUMAS, JAWA TENGAH**

**2023**

**Pembahasan SQLite Card View**

SQLite adalah sistem manajemen basis data relasional yang dapat diintegrasikan dengan mudah ke dalam aplikasi Android. Dalam Kotlin, kita dapat menggunakan SQLite untuk menyimpan dan mengambil data dari aplikasi kita. Salah satu komponen yang sering digunakan untuk menampilkan data dalam aplikasi Android adalah Card View. Card View adalah komponen yang memungkinkan kita membuat tampilan kartu yang estetis dengan bayangan, sudut melengkung, dan elemen dekoratif lainnya.

Untuk menghubungkan data dari SQLite ke Card View dalam aplikasi Kotlin, kita dapat menggunakan RecyclerView dan Adapter. RecyclerView adalah komponen yang digunakan untuk menampilkan daftar data yang dapat digulirkan dalam aplikasi Android. Dalam konteks Card View, kita dapat menggunakan RecyclerView untuk menampilkan daftar kartu yang diambil dari data dalam basis data SQLite. Adapter, seperti RecyclerView.Adapter, digunakan untuk menghubungkan data dari SQLite ke tampilan Card View di dalam RecyclerView.

Dalam penggunaan Card View pada aplikasi Kotlin, kita dapat menyesuaikan tampilan kartu dengan informasi seperti judul, deskripsi, dan gambar yang diambil dari data dalam SQLite. Kita dapat menggunakan perpustakaan pihak ketiga seperti Glide atau Picasso untuk memuat gambar dari URL atau basis data SQLite ke dalam tampilan kartu. Dengan demikian, penggunaan SQLite Card View pada aplikasi Kotlin dapat membantu kita membuat aplikasi yang interaktif dan menarik dengan mudah dan efisien.

**Langkah-Langkah Praktikum**

|  |  |
| --- | --- |
| **Langkah Praktikum** | **Pembahasan** |
|  | Pengguna dapat membuat project baru terlebih dahulu, dan sesuaikan nama serta tempat penyimpanan file project yang akan dibuat |
|  | Pada bagian pemilihan template pengguna dapat memilih Empty Activity |
|  | Selanjutnya pada activity\_main.xml buat layout form untuk Registrasi Data dengan 4 kolom inputan dan 2 button seperti gambar disamping. |
|  | Selanjutnya buatlah Layout Resource File baru dengan nama item\_dataku |
|  | Lalu buatlah Values Resource File baru dengan nama style |
|  | Pada style.xml buatlah 2 style yaiitu edit\_text dan btn\_Style |
|  | Selanjutnya pada item\_dataku buatlah layout seperti gambar di samping dengan tujuan untuk menampilkan data yang pengguna inputkan disertai dengan tombol delete dan update |
|  | Selanjutnya buatlah Empty Activity baru dengan nama UpdateActivity serta layout file nya juga dengan nama activity\_update |
|  | Lalu pada activity\_update.xml buat layout untuk update data seperti gambar disamping. |
|  | Selanjutnya buatlah kotlin class baru dengan nama DBModel untuk inisialisasi Variable |
|  | Selanjutnya buatlah DBInfo untuk menampung objek tabel. |
|  | Selanjutnya buatlah kotlin class baru dengan nama DBAdapter |
|  | Lalu buat juga kotlin class baru dengan nama DBHelper untuk membantu menampung variable yang telah di update |
|  | Selanjutnya buatlah Empty Activity baru dengan nama RvDbActivity beserta dengan layout nya dengan nama activity\_rv\_db yang berfungsi untuk recyclerview |
|  | Lalu buatlah Empty Activity baru dengan nama MainActivity2 beserta dengan layout nya dengan nama activity\_main2 |
|  | Gambar di samping merupakan hasil pada bagian input data |
|  | Gambar di samping merupakan hasil data yang telah di inputkan |
|  | Gambar di samping merupakan tampilan update data |

**Pembahasan Login With SQLite Shared Preference**

Login with SQLite Shared Preference adalah pendekatan yang umum digunakan dalam pengembangan aplikasi Android untuk mengelola proses login pengguna. SQLite digunakan sebagai sistem manajemen basis data untuk menyimpan informasi pengguna, seperti email dan password, dalam tabel. Informasi ini dapat diakses dan diverifikasi saat pengguna melakukan login. Selain itu, Shared Preference digunakan untuk menyimpan status login pengguna, sehingga pengguna tetap masuk setelah aplikasi ditutup dan dibuka kembali. Dengan menggabungkan SQLite dan Shared Preference, aplikasi dapat mengamankan informasi pengguna dan mempertahankan status login.

Dalam implementasi "Login with SQLite Shared Preference", tabel SQLite digunakan untuk menyimpan informasi pengguna, biasanya berisi kolom seperti email, password, dan mungkin juga nama pengguna. Ketika pengguna mencoba untuk login, aplikasi akan memeriksa informasi yang diinputkan dengan data yang tersimpan di tabel SQLite. Jika informasi yang dimasukkan cocok, maka login berhasil dan status login disimpan dalam Shared Preference. Jika tidak cocok, maka login gagal. Dengan menggunakan SQLite, informasi pengguna dapat dengan aman disimpan dan diakses dengan query SQL.

Shared Preference digunakan dalam "Login with SQLite Shared Preference" untuk menyimpan status login pengguna. Ketika pengguna berhasil login, status login disimpan dalam Shared Preference dan digunakan untuk mengecek apakah pengguna sudah masuk atau belum saat membuka kembali aplikasi. Dengan cara ini, pengguna tidak perlu melakukan login ulang setiap kali membuka aplikasi. Selain itu, Shared Preference juga dapat digunakan untuk menyimpan data lain yang relevan dengan sesi login, seperti preferensi pengguna atau pengaturan aplikasi yang disesuaikan. Dengan menggabungkan SQLite dan Shared Preference, aplikasi dapat memberikan pengalaman pengguna yang nyaman dan aman dalam proses login.  
**Langkah-Langkah Praktikum**

|  |  |
| --- | --- |
| **Langkah Praktikum** | **Pembahasan** |
|  | Pengguna dapat membuat project baru terlebih dahulu, dan sesuaikan nama serta tempat penyimpanan file project yang akan dibuat |
|  | Pada bagian pemilihan template pengguna dapat memilih Empty Activity |
|  | Buatlah empty activity baru dengan nama LoginActivity sertakan juga untuk layout nya |
|  | Selanjutnya buatlah layout sepert gambar di samping, terdapat 2 kolom inputan yaitu untuk Email dan Password serta terdapat 2 tombol yaitu untuk login dan register |
|  | Lalu buatlah empty activity baru dengan nama RegisterActivity sertakan juga untuk layout nya |
|  | Gambar di samping merupakan tampilan dari menu register dengan adanya 3 kolom inputan dan 2 tombol |
|  | Gambar di samping merupakan tampilan disaat pengguna telah login, hanya terdapat Text Hello World dan tombol untuk Logout |
|  | Selanjutnya buatlah Empty Activity baru sebagai loading screen |
|  | Lalu buatlah Package baru dengan nama DB |
|  | Pada Package DB buatlah ketiga class baru seperti gambar di samping, ada DataModelUser, DBHelper dan DBInfo |
|  | Pada onCreate(), layout yang terkait dengan SplashActivity ditentukan dengan menggunakan setContentView(). Kemudian, menggunakan Handler().postDelayed(), diatur delay selama 2000 milidetik (2 detik).  Setelah delay selesai, Intent digunakan untuk berpindah dari SplashActivity ke LoginActivity. Fungsi startActivity() digunakan untuk memulai aktivitas baru, yaitu LoginActivity. Selain itu, pemanggilan finish() digunakan untuk mengakhiri SplashActivity sehingga pengguna tidak dapat kembali ke halaman splash setelah berpindah ke halaman login. |
|  | Pada bagian login Ketika tombol login ditekan, aplikasi akan memeriksa apakah username dan password yang dimasukkan pengguna cocok dengan data yang tersimpan dalam database SQLite menggunakan fungsi cekLogin() dari DBHelper. Jika cocok, data login (email, password, dan status) akan disimpan dalam SharedPreferences menggunakan objek editSavedLogin. Selanjutnya, pengguna akan diarahkan ke MainActivity.  Jika username dan password tidak cocok, akan ditampilkan pesan "Gagal Login" menggunakan Toast. Jika tombol register ditekan, pengguna akan diarahkan ke RegisterActivity untuk mendaftar. |
|  | ditentukan dengan menggunakan setContentView(). Selanjutnya, inisialisasi elemen-elemen yang digunakan dalam proses registrasi, seperti EditText untuk email, password, dan nama lengkap, serta tombol register dan tombol cancel. Selain itu, DBHelper juga diinisialisasi untuk mengakses dan mengelola database SQLite.  Fungsi registerme() digunakan ketika tombol register ditekan. Pada fungsi ini, nilai yang diinput oleh pengguna diambil dari EditText. Selanjutnya, dilakukan pengecekan apakah email yang diinput sudah ada dalam database menggunakan fungsi cekUser() dari DBHelper. Jika email belum terdaftar, pengguna baru akan didaftarkan dengan menggunakan fungsi RegisterUser() dari DBHelper. Setelah itu, pengguna akan diarahkan ke halaman login dengan menggunakan Intent dan startActivity(). |
|  | Pada MainActivity ditentukan dengan menggunakan setContentView(). Selanjutnya, dilakukan inisialisasi tombol btnLogout yang digunakan untuk proses logout.  Ketika tombol btnLogout ditekan, dilakukan penghapusan data login yang disimpan dalam SharedPreferences menggunakan objek editSavedLogin. Data email dan password dihapus dengan memberikan nilai null, dan status login diubah menjadi "Off". Setelah itu, dilakukan pemanggilan commit() untuk menyimpan perubahan yang dilakukan. |
|  | Gambar di samping merupakan loading screen sebelum memasuki bagian login atau register |
|  | Gambar di samping merupakan hasil tampilan dari menu login |
|  | Gambar di samping merupakan hasil dari tampilan menu register |
|  | Gambar di samping adalah tampilan menu utama apabila pengguna telah register dan login |
|  | Gambar di samping merupakan data pengguna yang berhasil Register dan pastinya sudah bisa login |

LINK GITHUB :