

MOBILE FINAL PROJECT – KEPO

Nama : Muhamad Arief Liman

NIM : 2301876972

Emulator : Pixel 4

Folder dan File dalam Project

1. Model

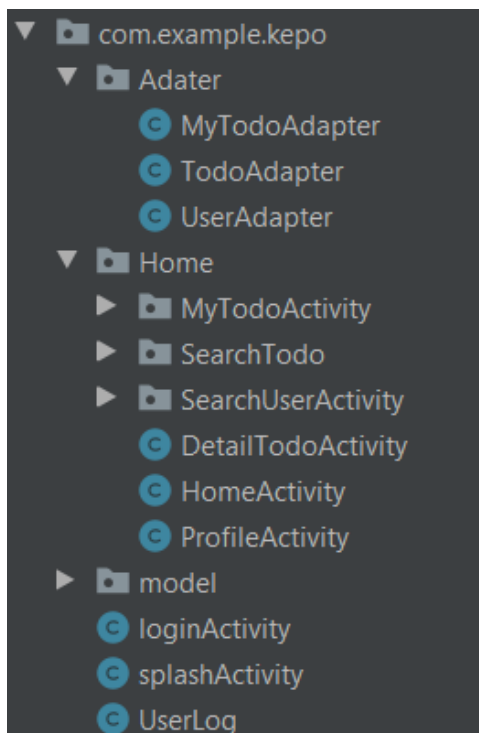
1. User : Menjadi Base Model untuk menyimpan attribute user
2. Todo : Menjadi Base Model untuk menyimpan attribute Todos

2. Adapter

1. MyTodoAdapter : Adapter yang digunakan untuk menampilkan My Todo (Checkbox).
2. TodoAdapter : Adapter yang digunakan untuk menampilkan semua Todo yang ada.
3. UserAdapter : Adapter yang digunakan untuk menampilkan user yang ada.

3. Home

4. UserLog : Merupakan SharedPreferences yang digunakan untuk menyimpan User yang Sedang Login.



Screen / Activity

1. Splash Screen
2. Login Screen
3. Home Screen
4. My Todo Screen (CRUD Todos)
5. Search Todo Screen (By User dan By Todos)
6. Search User Screen
7. Profile Screens

- Splash Screen



Pada Splash Screen, Saya Menggunakan Handler dan Menggunakan method postDelayed untuk mendapatkan delay 1,5 detik. Didalam Method tersebut ada 2 Params (Runnable, Delay). Pada Runnable terdapat Method run, Dimana dalam method tersebut saya melakukan validasi untuk user yang sudah login atau belum menggunakan UserLog.

```

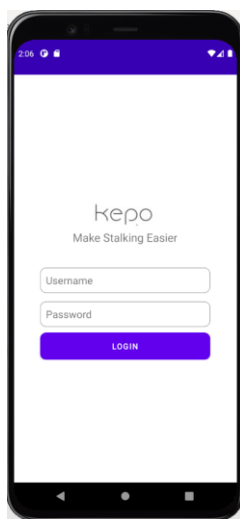
14     private Handler waitingHandler = new Handler();
15     private UserLog userLog;
16
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_splash);
21
22         // untuk melakukan Delayed pada screen.
23         waitingHandler.postDelayed(new Runnable() {
24             @Override
25             public void run() {
26                 try{
27                     // Mengambil Data User Yang Sudah Login
28                     userLog = new UserLog( context: splashActivity.this);
29                     User user = userLog.loadLog();
30
31                     Intent getIntent;
32                     if(user.getUser_id().equals("")){
33                         // Jika Belum Login Ke Login Activity
34                         getIntent = new Intent( packageContext: splashActivity.this, loginActivity.class);
35                     }else{
36                         // Jika Sudah Login Langsung Diarahkan Ke Home Activity
37                         getIntent = new Intent( packageContext: splashActivity.this, HomeActivity.class);
38                     }
39                     startActivity(getIntent);
40                     finish();
41                 }catch (Exception e){
42                     e.printStackTrace();
43                 }
44             }
45             // delay -> 1500 -> 1,5 Detik
46         }, delayMillis: 1500);
47     }
  
```

- Login Screen

Pada Login Screen, Saya Menggunakan 2 Edit Text untuk Menerima username dan Password serta 1 button untuk mengSubmit.

Pada Class Java Login Activity nya saya menggunakan 'Volley' untuk mengirim dan menerima Response dari API It Div. Setelah Memasukan Username dan Password secara Benar user tersebut akan disimpan di UserLog dan dilanjutkan Ke Screen Home.

Jika Memasukan Username dan Password yang salah ataupun Tidak Mengisi Password dan Username tersebut, Akan muncul Pesan Error. Pada Pesan Error disini saya membuat nya dengan AlertDialog yang diberi gravity bottom.



```
private void LoginPost(){
    User user = new User();

    // Mengambil Username dan password yang sudah di Input
    user.setUsername(binding.etUsername.getText().toString());
    user.setPassword(binding.etPassword.getText().toString());

    // Jika Username atau Password Kosong
    if(user.getUsername().equals("") || user.getPassword().equals("")) {

        // Membentuk AlertDialog
        AlertDialog.Builder builder = new AlertDialog.Builder( context LoginActivity.this);

        // Setting Title dan Message dan Apa yang terjadi ketika di Click
        builder.setTitle("Error")
            .setMessage("Please Input Username or Password")
            .setPositiveButton( text: "Oke", new DialogInterface.OnClickListener() {

                @Override
                public void onClick(DialogInterface dialog, int which) {
                    // Field pada edit text akan dihilangkan
                    binding.etPassword.setText("");
                    binding.etUsername.setText("");
                    dialog.cancel();
                }
            })

        AlertDialog ad = builder.create();
        // Membuat AlertDialog berada pada Bottom of Activity
        ad.getWindow().setGravity(Gravity.BOTTOM);
        ad.show();
    }else{
```

```
}else{
    // Jika Tidak Kosong dan membuat visible Progress Bar
    binding.pbReqMeth.setVisibility(View.VISIBLE);
    // Melakukan Request Volley dengan Method Post dan Body
    JSONObjectRequest jsonObjectRequest = new JSONObjectRequest(
        Request.Method.POST,
        BASE_URL,
        getBodyJson(user),
        new Response.Listener<JSONObject>() {

            @Override
            // Success Response
            public void onResponse(JSONObject response) {
                // Progress Bar dihilangkan dan Mengambil data berupa JSONObject response
                binding.pbReqMeth.setVisibility(View.GONE);
                try{
                    int status = response.getInt( name: "status");
                    String msg = response.getString( name: "message");
                    JSONObject data = response.getJSONObject("data");
                    if(status == 200 && msg.equals("Login success")){
                        String user_id = data.getString( name: "user_id");
                        String name = data.getString( name: "name");

                        // Memasukan Data yang didapat kedalam Class Model User dan UserLog
                        user.setUser_id(user_id);
                        user.setName(name);

                        userLog.saveLog(user);

                        gotoHomeAct();
                    }else{
```

```
new Response.ErrorListener() {

            @Override
            // Error Response
            public void onErrorResponse(VolleyError error) {
                // Menampilkan Error Message dengan AlertDialog
                AlertDialog.Builder builder = new AlertDialog.Builder( context LoginActivity.this);

                builder.setTitle("Error")
                    .setMessage(error.toString())
                    .setPositiveButton( text: "Oke", new DialogInterface.OnClickListener() {

                        @Override
                        public void onClick(DialogInterface dialog, int which) {
                            binding.etPassword.setText("");
                            binding.etUsername.setText("");
                            dialog.cancel();
                        }
                    })

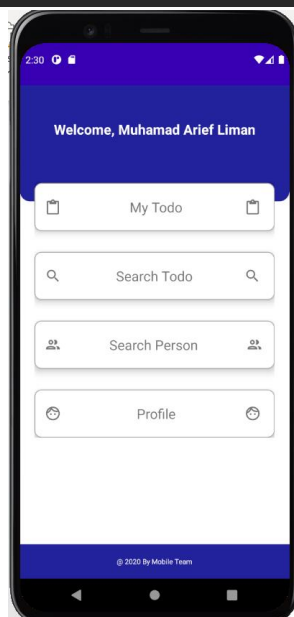
                AlertDialog ad = builder.create();
                ad.getWindow().setGravity(Gravity.BOTTOM);
                ad.show();
            }
        });

    // Dimasukan Kedalam Queue
    RequestQueue queue = Volley.newRequestQueue( context this);
    queue.add(jsonObjectRequest);
```

- Home Screen

Pada Home Screen ini, terdapat Text dimana menyambut pengguna “Welcome, {name}” dan juga terdapat 4 Section untuk memilih yaitu My Todo, Search Todo, Search User dan Profile.

```
// My Todo Action
binding.llMyTodo.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent getIntent = new Intent( packageContext: HomeActivity.this, MyTodoActivity.class);
        startActivity(getIntent);
    }
});
// Search Todo action
binding.llSearchTodo.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent getIntent = new Intent( packageContext: HomeActivity.this, SearchTodoActivity.class);
        startActivity(getIntent);
    }
});
// Search Person Action
binding.llSearchPerson.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent getIntent = new Intent( packageContext: HomeActivity.this, SearchUserActivity.class);
        startActivity(getIntent);
    }
});
// Profile Action
binding.llProfile.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent getIntent = new Intent( packageContext: HomeActivity.this, ProfileActivity.class);
        startActivity(getIntent);
    }
});
});
```



- My Todo Screen

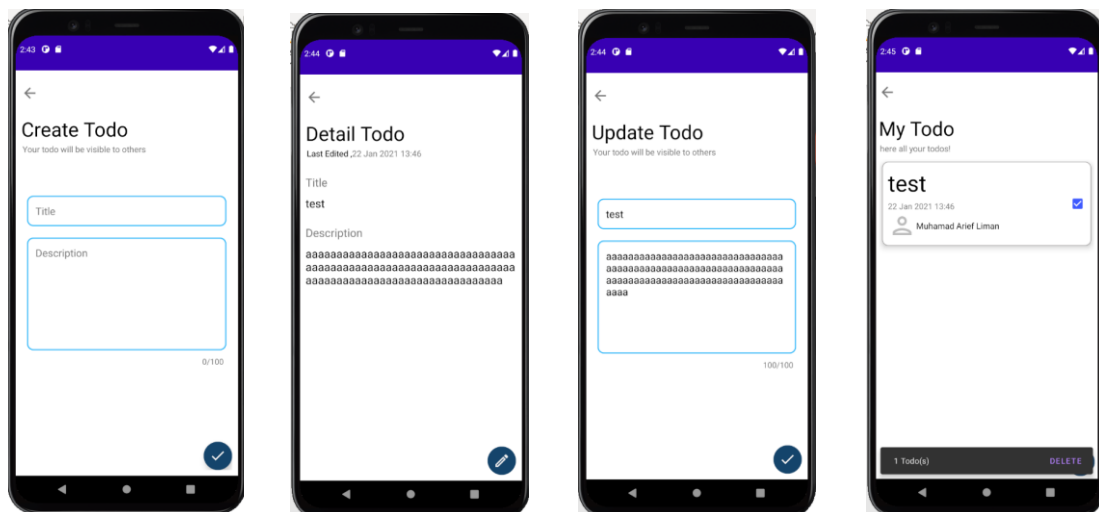
Pada Activity ini kita dapat Membuat, melihat, mengupdate serta mendelete Todo(s) yang telah User buat. untuk membuat dapat menekan button pada pojok kanan bawah dan nanti akan diarahkan ke Screen Create Todo setelah mengisi Kelengkapan seperti Title dan Description dari Todo(s) akan Muncul Alert Dialog untuk meyakinkan user membuat Todo tersebut. Jika Title dan Description tidak diisi akan memunculkan Pesan Error.

User Dapat Melihat Secara Detail dengan menekan Box Todo. Setelah diarahkan ke Activity Detail User Juga dapat mengupdate Todo Tersebut dengan Menekan button pojok kanan bawah. Tampilan Pada Update Sangat Mirip dengan Create akan tetapi pada Update Field nya sudah Terisi dengan Todos nya.

Disini saya Menggunakan TextWatcher() pada Create dan Update untuk menghitung penambahan dan penghilangan huruf. Jika Description Melebihi Batas yang ditentukan maka akan muncul Error Message.

User juga dapat melakukan Delete kepada Todo yang sudah tidak dipakai, Dengan Menekan Checkbox pada box todos tersebut, setelah itu akan muncul Snackbar dan menunjukkan banyaknya Todos yang telah dipilih, Setelah itu tekan Button Delete dan nantinya akan muncul AlertDialog untuk meyakinkan User menghapus todo yang telah dipilih.

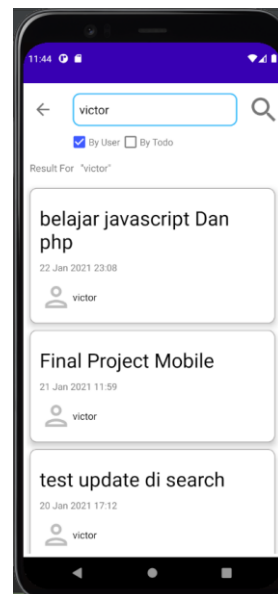
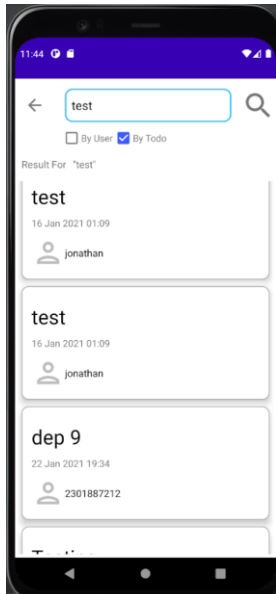
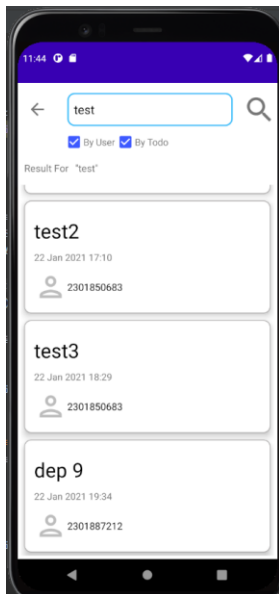
Untuk CRUD Todo disini saya Menggunakan Volley untuk merequest ke API, Snackbar untuk menunjukan item yang telah dipilih dan AlertDialog serta assets untuk tampilan Button Create dan Update



- Search Todo Screen

Pada Search Todo dapat dicari berdasarkan 2 jenis pencarian, By User dan By Todo. Jika User memilih By User maka akan ditampilkan Todo yang dimiliki oleh user yang dicari. Jika Mencari By Todo maka akan menampilkan semua Todo yang memiliki Hubungan dengan keywords yang di cari baik dari Title dan Description. Dan User dapat Memilih Keduanya dan akan menghasilkan semua Todo yang berkaitan dengan Keywords tersebut.

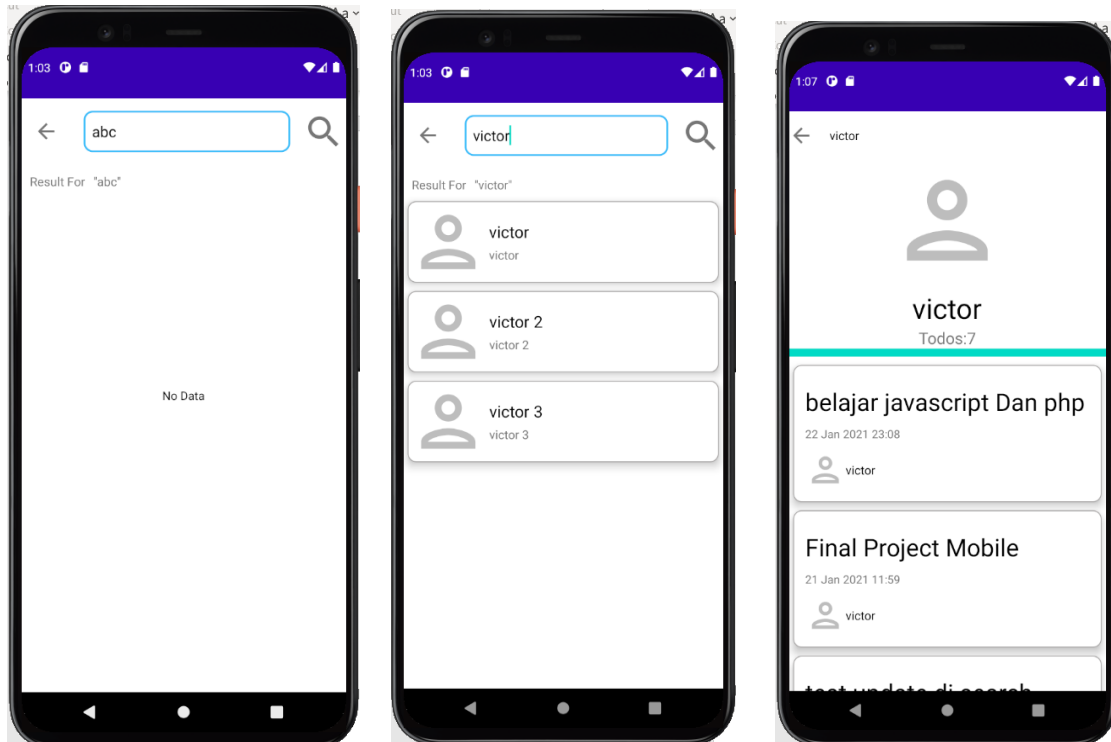
Disini Saya Menggunakan Volley dan Recycler View serta adapter untuk menampilkan hasil todo yang telah didapat melalui API.



- Search User

Pada Activity ini, User dapat mencari user lain dan melihat detail dari user tersebut dan melihat banyak Todo yang dimiliki oleh User tersebut Dan Dapat Melihat Detail dari Todo tersebut. Jika tidak ada User yang di temukan maka akan diberi pesan “No Data”.

Disini Saya Menggunakan Volley dan memasukan data tersebut ke dalam adapter dan Recycler View untuk menampilkan data data dari API.



- Profile Screen

Di Screen Ini, User dapat melihat Full Name dan Username dan diberikan pilihan untuk Logout dan Jika User Log out maka user akan menuju Splash Screen dan masuk kembali Ke menu login dengan begitu UserLog juga telah terhapus.

