

Nama : ALIF AHSANIL SATRIA
NPM : 1606882540

Requirement metode & teknologi:

1. Menerapkan seluruh stack Android Framework standard

- a. Menerapkan activity

Activity yang saya buat terdiri dari 5 :



LoginActivity : menampilkan halaman login

MainActivity : menampilkan halaman utama

RegisterActivity : menampilkan halaman register

QuizActivity : menampilkan halaman kuis

ProfileActivity : menampilkan halaman user tepat setelah login

- b. Menggunakan Service dan pemanggilan Remote Method (di luar aplikasi)

Saya menerapkan pemanggilan remote method untuk menampilkan notifikasi ketika waktu menjawab pertanyaan sudah habis. Jadi, 1 pertanyaan memiliki batas waktu 30 detik untuk menjawab dan apabila waktu menjawab sudah habis, akan ada notifikasi :

```
buttonConfirmText.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        if (!answered) {  
            if (rb1.isChecked() || rb2.isChecked() || rb3.isChecked()) {  
                checkAnswer();  
            } else {  
                Toast.makeText(context, QuizActivity.this, "Please select an answer", Toast.LENGTH_SHORT).show();  
            }  
        } else {  
            showNextQuestion();  
            displayNotification();  
        }  
    }  
});  
}  
  
private void displayNotification() {  
    NotificationCompat.Builder builder = new NotificationCompat.Builder(context, CHANNEL_ID);  
    builder.setSmallIcon(R.drawable.ic_warning_notification);  
    builder.setContentTitle("Timeout Notification");  
    builder.setContentText("Your time to answer this question is out");  
    builder.setPriority(NotificationCompat.PRIORITY_DEFAULT);  
  
    NotificationManagerCompat notificationManagerCompat = NotificationManagerCompat.from(this);  
    notificationManagerCompat.notify(NOTIFICATION_ID, builder.build());  
}
```

- c. Memanfaatkan ContentProvider

Saya memanfaatkan ContentProvider untuk menyimpan gambar hasil tangkapan kamera ke dalam direktori android.com.example.quiz di hp kita dengan cara sebagai berikut :

Bagian method onCreate :

```
FloatingActionButton fab = findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startActivityForResult(getPickImageChooserIntent(), IMAGE_RESULT);
    }
});

permissions.add(CAMERA);
permissions.add(WRITE_EXTERNAL_STORAGE);
permissions.add(READ_EXTERNAL_STORAGE);
permissionsToRequest = findUnAskedPermissions(permissions);

if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
    if (permissionsToRequest.size() > 0)
        requestPermissions(permissionsToRequest.toArray(new String[permissionsToRequest.size()]), ALL_PERMISSIONS_RESULT);
}

loadHighscore();

Button buttonStartQuiz = findViewById(R.id.button_start_quiz);
buttonStartQuiz.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) { startQuiz(); }
});

IntentFilter filter = new IntentFilter( action: "berhasil_login");
registerReceiver(broadcastReceiver, filter);
```

```
public Intent getPickImageChooserIntent() {

    Uri outputFileUri = getCaptureImageOutputUri();

    List<Intent> allIntents = new ArrayList<>();
    PackageManager packageManager = getPackageManager();

    Intent captureIntent = new Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
    List<ResolveInfo> listCam = packageManager.queryIntentActivities(captureIntent, 0);
    for (ResolveInfo res : listCam) {
        Intent intent = new Intent(captureIntent);
        intent.setComponent(new ComponentName(res.activityInfo.packageName, res.activityInfo.name));
        intent.setPackage(res.activityInfo.packageName);
        if (outputFileUri != null) {
            intent.putExtra(MediaStore.EXTRA_OUTPUT, outputFileUri);
        }
        allIntents.add(intent);
    }

    Intent galleryIntent = new Intent(Intent.ACTION_GET_CONTENT);
    galleryIntent.setType("image/*");
    List<ResolveInfo> listGallery = packageManager.queryIntentActivities(galleryIntent, 0);
    for (ResolveInfo res : listGallery) {
        Intent intent = new Intent(galleryIntent);
        intent.setComponent(new ComponentName(res.activityInfo.packageName, res.activityInfo.name));
        intent.setPackage(res.activityInfo.packageName);
        allIntents.add(intent);
    }
}
```

```

private String getImageFromFilePath(Intent data) {
    boolean isCamera = data == null || data.getData() == null;

    if (isCamera) return getCaptureImageOutputUri().getPath();
    else return getPathFromURI(data.getData());
}

public String getImageFilePath(Intent data) {
    return getImageFromFilePath(data);
}

private String getPathFromURI(Uri contentUri) {
    String[] proj = {MediaStore.Audio.Media.DATA};
    Cursor cursor = getContentResolver().query(contentUri, proj, null, null, null);
    int column_index = cursor.getColumnIndexOrThrow(MediaStore.Audio.Media.DATA);
    cursor.moveToFirst();
    return cursor.getString(column_index);
}

```

Untuk bagian lebih detail, bisa dilihat pada source code (karena terlalu banyak jika semuanya ditampilkan di sini)

d. Menerapkan BroadcastReceiver

Saya mengimplementasikan broadcast yang menampilkan “Berhasil Login” di halaman ProfileActivity apabila loginnya berhasil :

Pada method onCreate :

```
IntentFilter filter = new IntentFilter( action: "berhasil_login");
registerReceiver(broadcastReceiver, filter);

Intent intentNotification = new Intent( action: "berhasil_login");
sendBroadcast(intentNotification);
```

Pada method onDestroy :

```
unregisterReceiver(broadcastReceiver);
```

Membuat anonymous class BroadcastReceiver :

```
private BroadcastReceiver broadcastReceiver = (context, intent) -> {
    Toast.makeText(context, text: "Berhasil Login", Toast.LENGTH_SHORT).show();
};
```

e. Menerapkan Background Process (cth: AsyncTask) yang tidak “mati” ketika activity tidak aktif

Saya menerapkan ini pada CountdownTimer yg menyatakan sisa waktu menjawab soal kuis :

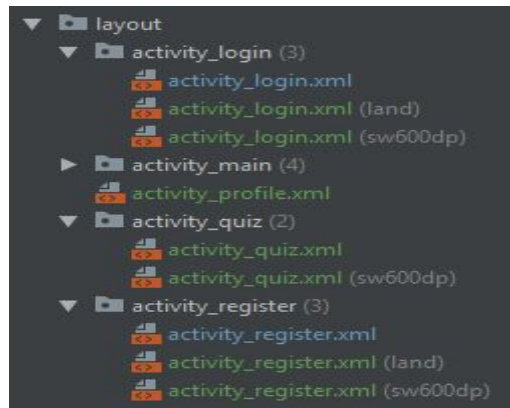
```
private void startCountDown() {
    CountdownTimer = new CountdownTimer(timeLeftInMillis, countdownInterval: 1000) {
        @Override
        public void onTick(long millisUntilFinished) {
            timeLeftInMillis = millisUntilFinished;
            updateCountDownText();
        }

        @Override
        public void onFinish() {
            timeLeftInMillis = 0;
            updateCountDownText();
            checkAnswer();
        }
    }.start();
}
```

2. Menerapkan multi environment

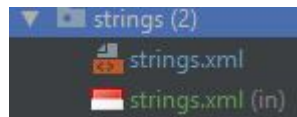
a. Multi Layout

Bisa dilihat pada direktori ini :

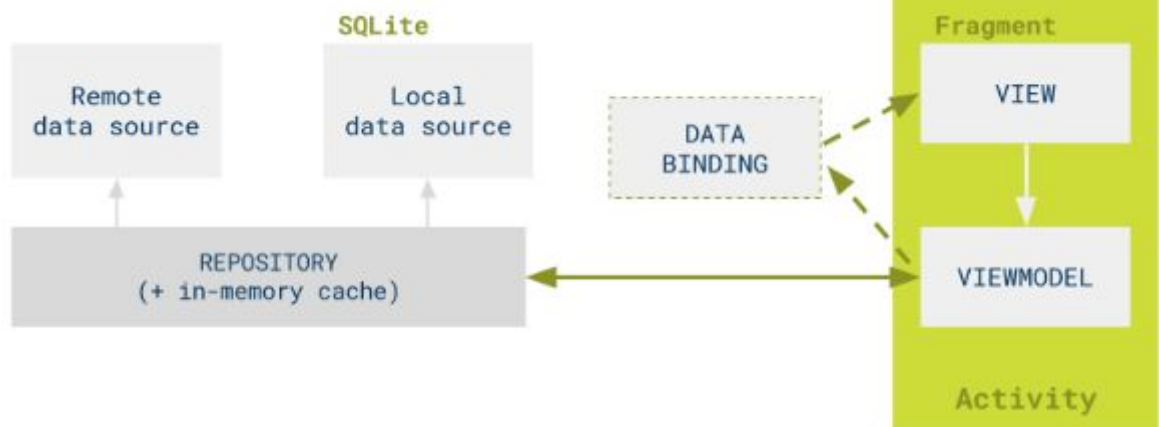


b. Multi Language

Bisa dilihat pada direktori ini :



3. Menerapkan Design Pattern MVVM & Background Task



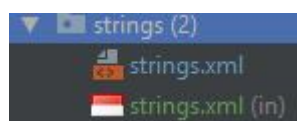
a. Untuk local data source, saya menggunakan room database persistence. Entity direpresentasikan oleh class User, Question, LoginUser. Dao, untuk mengakses entity di database direpresentasikan oleh UserDao, QuestionDao, LoginUserDao.

b. Untuk repository, direpresentasikan oleh QuizRepository

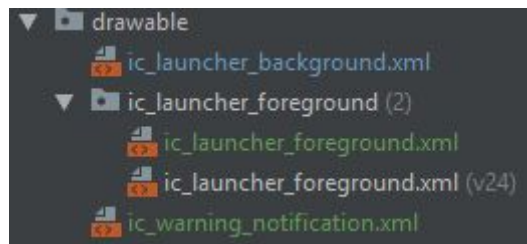
c. Untuk view, direpresentasikan oleh MainActivity, LoginActivity, RegisterActivity, ProfileActivity, dan QuizActivity

d. Untuk viewmodel, direpresentasikan oleh QuizViewModel dan akan melakukan binding dengan view

4. a. bisa dilihat di direktori ini :



b. bisa dilihat di direktori ini :



c. Sudah dijelaskan dan diberikan potongan kodenya pada bagian 1c

5. Menerapkan data persistence

Entity yang digunakan : User, Question, LoginUser. User merepresentasikan data pengguna, Question merepresentasikan daftar pertanyaan, dan LoginUser sebagai representasi state apakah user sudah login atau belum menggunakan LiveData.