

Dokumentasi

Tugas Tengah dan Tugas Akhir Semester

TKTPL Gasal 2020/2021



Nama: Aulia Rosyida

NPM: 1706025346

Tanggal Pengumpulan:

30 Januari 2021

Link Github:

[aulia-rosyida/DietSkuy \(github.com\)](https://github.com/aulia-rosyida/DietSkuy)

Fakultas Ilmu Komputer

Universitas Indonesia

2021

Deskripsi Tugas

Membuat aplikasi sederhana yang memanfaatkan **seluruh requirement, metode & teknologi** yang ditentukan di dalam spesifikasi tugas. Contoh: Menampilkan lokasi GPS saat ini, IP Address, Operator Telekomunikasi yang sedang digunakan, jumlah Contact yang tersimpan di dalam HP, kapasitas baterai yang masih tersedia, dll.

Dokumentasi

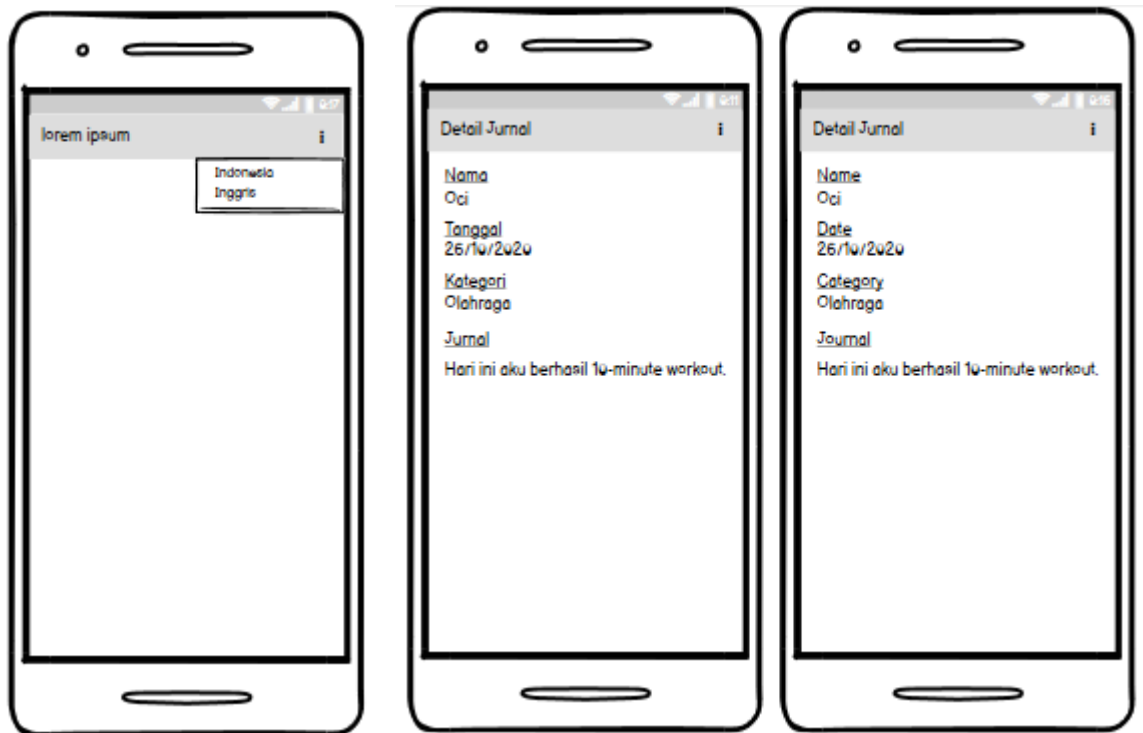
I. Aplikasi: DietSkuy

DietSkuy merupakan sebuah aplikasi untuk jurnal mengenai hal-hal yang dilakukan untuk menunjang diet untuk keluarga. User dapat mengisi nama, kategori penunjang diet, misal: olahraga, makan buah, dsb. Dengan aplikasi ini, pengguna dapat tracking kegiatan apa saja yang sudah dilakukan untuk membantu target diet anggota keluarga yang dilakukan. Akan ada stopwatch yang menghitung mundur untuk kegiatan 10-minutes workout.

II. Rancangan UI/UX (Wireframe)







III. penerapan memenuhi 11 requirement & teknologi dari tugas akhir.

Stack Android Framework Standard

1. Menerapkan activity:

- Main Activity: sebagai base layout untuk main fragment, berisi 1 fragment untuk menu-menu utama, yakni 10-menit-workout, buat jurnal, dan lihat jurnal.

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main_activity);  
        if (savedInstanceState == null) {  
            getSupportFragmentManager().beginTransaction()  
                .replace(R.id.container, MainFragment.newInstance())  
                .commitNow();  
        }  
    }  
}
```

Pada file MainActivity.java dapat dilihat bagian untuk memanggil MainFragment.

- Workout Activity: berisi activity untuk handle timer 10 menit workout.

```
public class WorkoutActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_workout);  
    }  
}
```

- c. List Jurnal Activity: berisi activity untuk handle list jurnal diet

```
public class ListJurnalActivity extends AppCompatActivity {
    private TaskListFragment taskListFragment = new TaskListFragment();

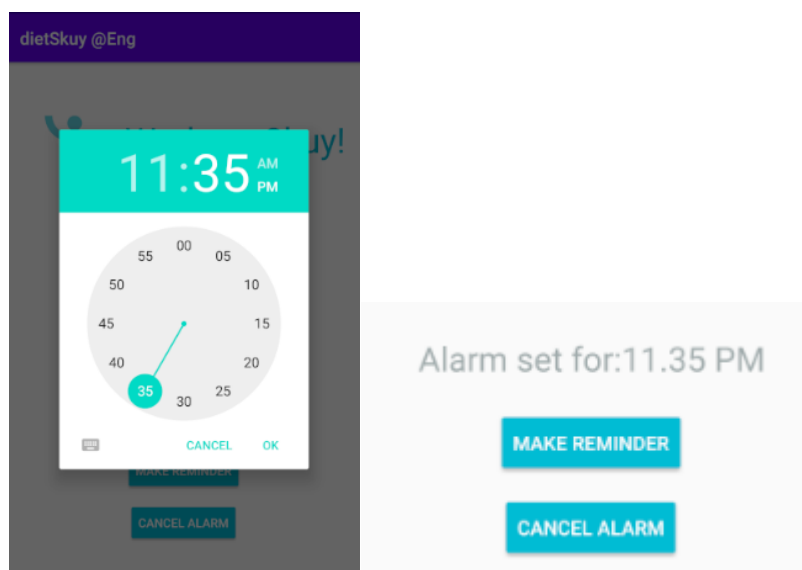
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_list_jurnal);
        getSupportFragmentManager().beginTransaction()
            .add(R.id.container, taskListFragment)
            .addToBackStack("task_list")
            .commit();
    }
}
```

2. Menggunakan Service : Saya ingin memanfaatkan alarm manager sebagai service untuk pengingat.

WorkoutActivity.java

```
private void startAlarm(Calendar c) {
    AlarmManager alarmManager = (AlarmManager) getSystemService(Context.ALARM_SERVICE);
    Intent intent = new Intent( packageContext: this, AlertReceiver.class);
    PendingIntent pendingIntent = PendingIntent.getBroadcast( context: this, requestCode: 1, intent, flags: 0);
    if (c.before(Calendar.getInstance())) {
        c.add(Calendar.DATE, 1);
    }
    alarmManager.setExact(AlarmManager.RTC_WAKEUP, c.getTimeInMillis(), pendingIntent);
    Toast.makeText( context: this, text: "Reminder Set!", Toast.LENGTH_SHORT).show();
}
```

ALARM_SERVICE saya gunakan dengan getSystemService untuk melakukan retrieve sebuah AlarmManager ketika mendapat intents pada waktu yang dipilih. RTC_WAKEUP akan membangunkan device untuk menyalakan pending intent pada spesifik waktu yang sudah di set ketika klik button “Make Reminder” menggunakan DialogTimePicker.



Ketika memilih waktu yang diinginkan, maka setelah klik “OK” akan memunculkan tulisan alarm set untuk jam yang sudah dipilih.

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
    alarmManager.setExactAndAllowWhileIdle(AlarmManager.RTC_WAKEUP, wakeTime, pendingIntent);
} else if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {
    alarmManager.setExact(AlarmManager.RTC_WAKEUP, wakeTime, pendingIntent);
} else {
    alarmManager.set(AlarmManager.RTC_WAKEUP, wakeTime, pendingIntent);
}
```

alarmManager kemudian akan membangunkan sesuai dengan waketime, yakni c.getTimeInMillis() untuk melakukan pendingIntent.

```
Intent intent = new Intent( packageContext this, AlertReceiver.class);
PendingIntent pendingIntent = PendingIntent.getBroadcast( context this, requestCode 1, intent, flags 0);
```

pendingIntent akan mendapatkan broadcast yang akan diterima oleh implementasi dari BroadcastReceiver, yakni AlertReceiver.

3. Memanfaatkan Content Provider: Memanfaatkan ContentProvider untuk keperluan penyimpanan data. Disini saya membuat custom content family provider.

FamilyContentProvider.java

```
@Override
public Cursor query(Uri uri, String[] projection, String selection,
                    String[] selectionArgs, String sortOrder) {
    Cursor cursor = null;

    final SQLiteDatabase db = famHelper.getReadableDatabase();

    switch (famUriMatcher.match(uri)){
        case FAMILY:
            cursor = db.query(FamilyDbContract.FamTable.TABLE_NAME, projection, selection,
                            selectionArgs, null, null, sortOrder);
            break;
    }
    return cursor;
}
```

Beberapa file lainnya yang mendukung penggunaan:

Family.java	A
FamilyContentProvider.java	M
FamilyDbActivity.java	A
FamilyDbContract.java	A
FamilyDbHelper.java	A

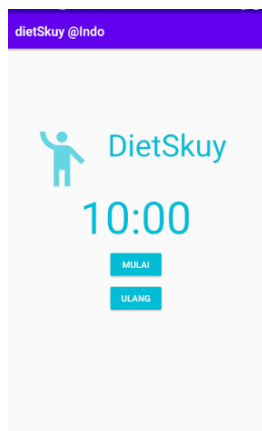
4. Menerapkan BroadcastReceiver: saya menggunakan implementasinya untuk menangkap permintaan broadcast yang dikirimkan via Intent ketika sudah memilih waktu untuk reminder. Berikut adalah potongan code yang mengimplementasikan broadcast receiver:

AlertReceiver.java

```
public class AlertReceiver extends BroadcastReceiver {  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        NotificationHelper notificationHelper = new NotificationHelper(context);  
        NotificationCompat.Builder nb = notificationHelper.getChannelNotification();  
        notificationHelper.getManager().notify( id: 1, nb.build());  
    }  
}
```

Jadi, ketika AlertReceiver dipanggil, maka saya memanfaatkan objek NotificationHelper untuk melakukan tugasnya.

5. Menerapkan Background Process yang tidak “mati” ketika activity tidak aktif. Saya memanfaatkan background process pada CountdownTimer untuk fitur 10-minute workout.



Saya menggunakan CountdownTimer dengan membuat 3 method, yakni startTimer(), stopTimer() dan updateTimer(). Di dalam startTimer() dan stopTimer() terdapat Boolean untuk mengecek apakah timerRunning (waktu timer) sedang berjalan atau tidak. Sehingga ketika aplikasi sedang kehilangan fokus, timer tetap berjalan.

WorkoutActivity.java

```
public class WorkoutActivity extends AppCompatActivity {  
    private TextView countdownText, reminderText;  
    private Button countdownButton, resetButton, pickerButton, cancelButton;  
  
    private CountdownTimer countDownTimer;  
    private long timeLeftInMilliseconds = 600000; //10 menit  
    private boolean timerRunning;  
  
    int reminderHour, reminderMinute;  
}
```

Inisialisasi Boolean timerRunning untuk menjaga agar timer tetap menyala ketika aplikasi sedang *lost focus*. Boolean tersebut digunakan pada 2 method berikut:

startTimer()

```
public void startTimer() {  
    countdownTimer = new CountdownTimer(timeLeftInMilliseconds, countDownInterval: 1000) {  
        @Override  
        public void onTick(long l) {  
            timeLeftInMilliseconds = l;  
            updateTime();  
        }  
  
        @Override  
        public void onFinish() {  
        }  
    }.start();  
  
    countdownButton.setText("Pause");  
    timerRunning = true;  
}
```

stopTimer()

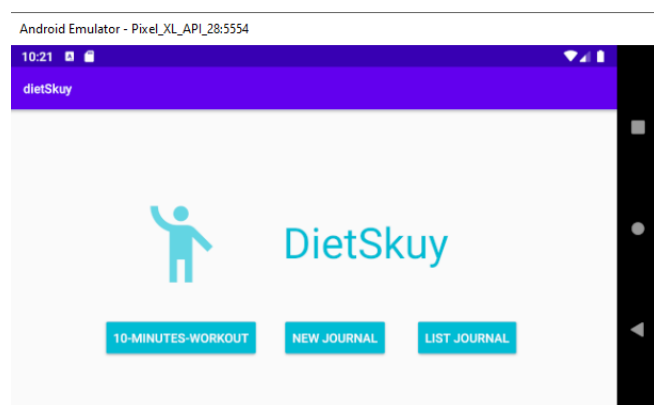
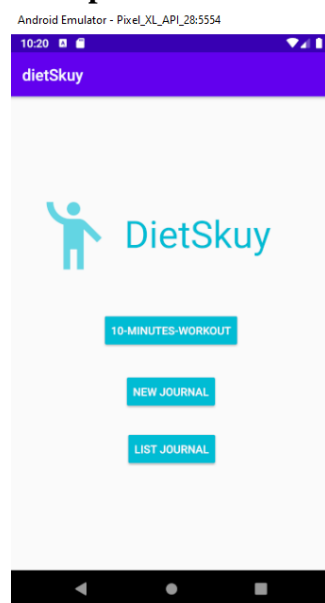
```
public void stopTimer() {  
    countdownTimer.cancel();  
    countdownButton.setText("Start");  
    timerRunning = false;  
}
```

Menerapkan Multi Environment dan Asset dengan Benar

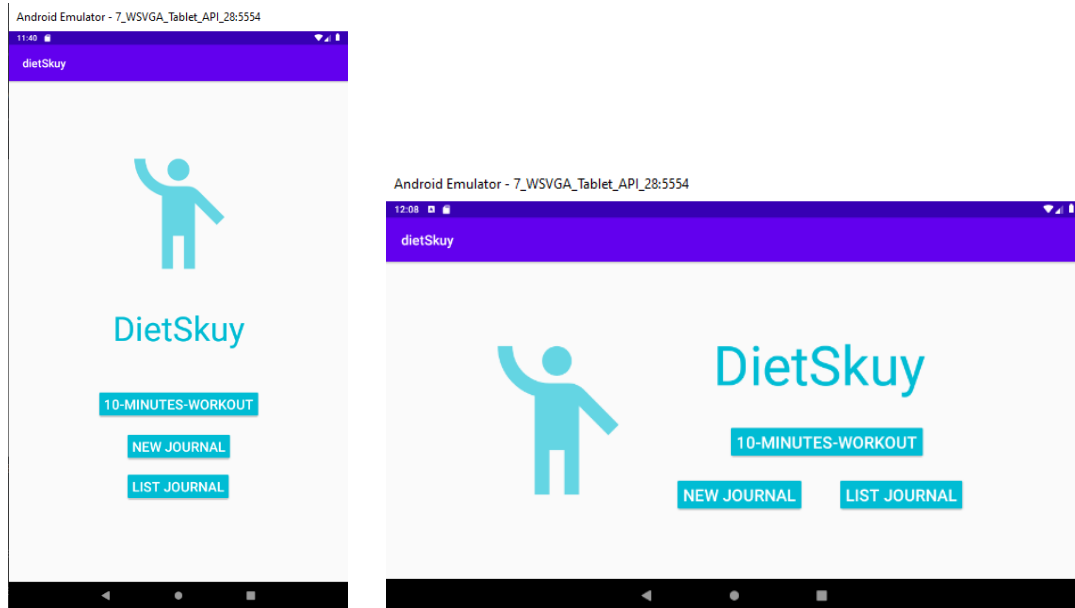
1. Multi Layout

Saya akan menerapkan multi layout ketika portrait, maka button dan icon akan berjejer kebawah, sedangkan ketika landscape button akan berjejer ke samping baik pada tablet maupun handphone.

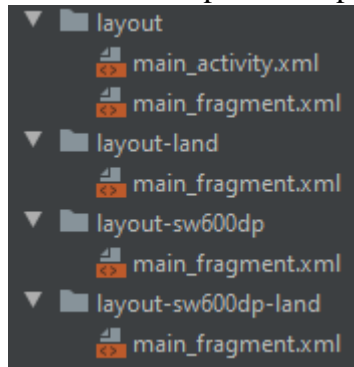
Handphone



Tablet

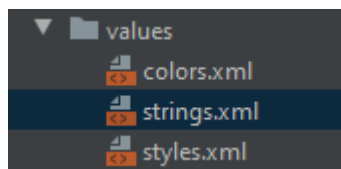


Source code dapat di cek pada file berikut:

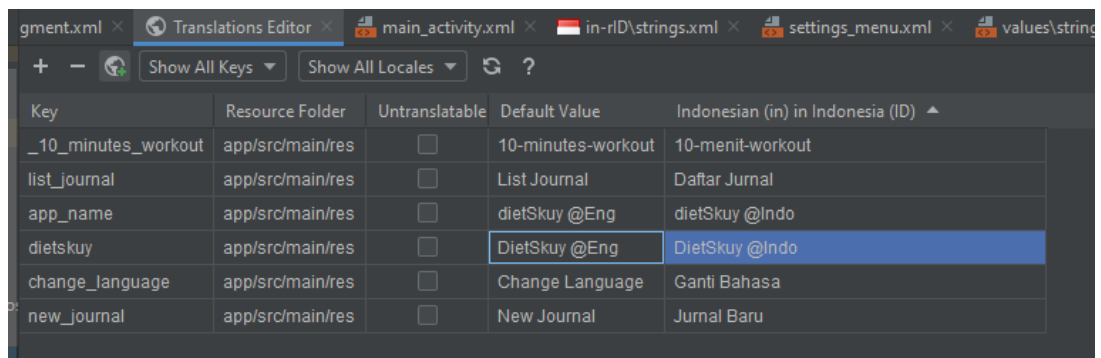


Dapat terlihat bahwa layout dan layout-land dibuat untuk android mobile, sedangkan layout-sw600dp dan layout-sw600dp-land dibuat untuk android tablet.

2. Multi Language (i18n): Akan terdapat 2 bahasa, yaitu Bahasa Indonesia dan Inggris. Pertama, perlu membuat semua string di dalam strings.xml. File dapat ditemukan pada folder res -> values -> strings.xml

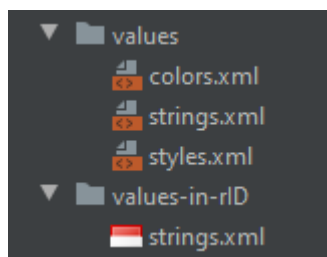


Kemudian, menambahkan strings dengan jenis bahasa baru, dapat otomatis apabila menambahkan dengan “Translations Editor”

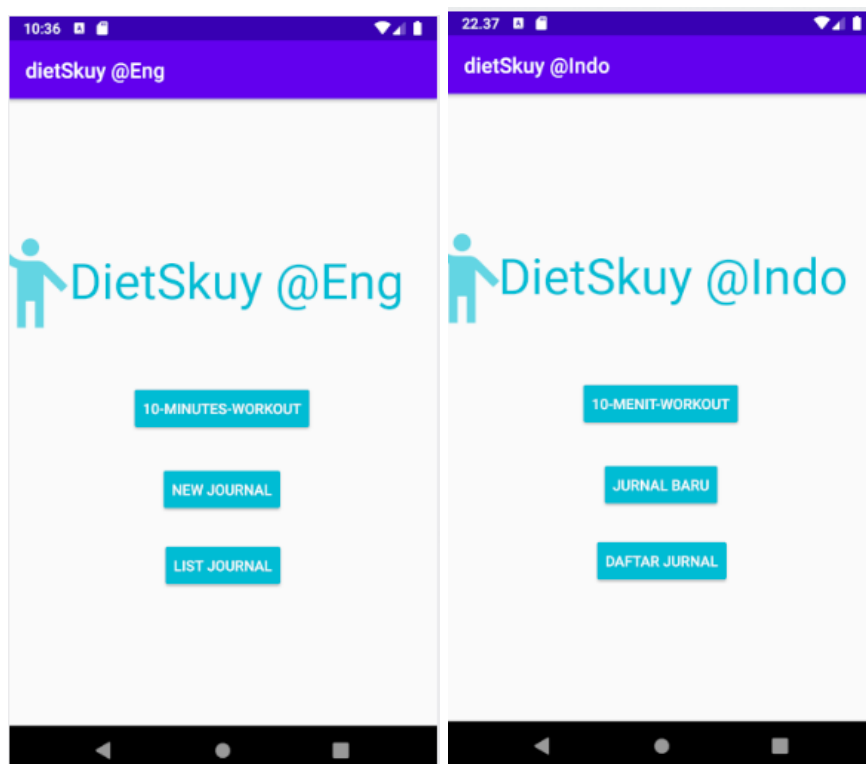


Key	Resource Folder	Untranslatable	Default Value	Indonesian (in) in Indonesia (ID)
_10_minutes_workout	app/src/main/res	<input type="checkbox"/>	10-minutes-workout	10-menit-workout
list_journal	app/src/main/res	<input type="checkbox"/>	List Journal	Daftar Jurnal
app_name	app/src/main/res	<input type="checkbox"/>	dietSkuy @Eng	dietSkuy @Indo
dietskuy	app/src/main/res	<input type="checkbox"/>	DietSkuy @Eng	DietSkuy @Indo
change_language	app/src/main/res	<input type="checkbox"/>	Change Language	Ganti Bahasa
new_journal	app/src/main/res	<input type="checkbox"/>	New Journal	Jurnal Baru

Dapat terlihat ada tambahan folder value disini:



Berikut hasil screenshot perbandingan antara penggunaan bahasa Indonesia dan Inggris:

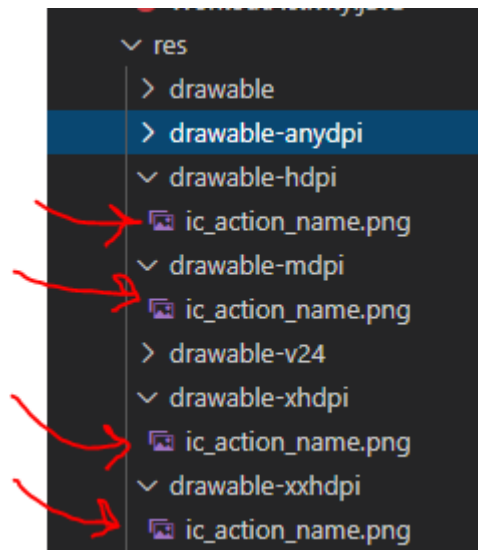


Aslinya saya berencana agar dapat membuat tombol yang langsung dapat memilih bahasa antara bahasa Inggris dan Indonesia dari dalam aplikasi, tapi ternyata ketika saya coba implementasi bahasanya tidak berubah sehingga keinginan saya tercover dengan mengubah bahasanya secara langsung pada settings handphone.

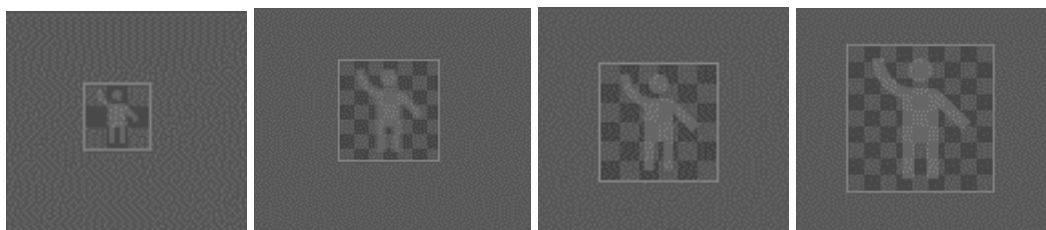
3. Mempersiapkan drawable resource untuk 2 (dua) screen size (& beda resolusi).

Menyiapkan icon dengan resolusi berbeda untuk beberapa resolusi device.

Icon yang saya pilih dapat di cek pada screenshot di bawah ini:



Ada 4 jenis ukuran icon berbeda yang saya buat. Yakni hdpi, mdpi, xhdpi, dan xxhdpi. Rasionya yakni sebagai berikut: mdpi adalah ukuran asli, hdpi memiliki rasio 1.5x, xhdpi memiliki rasio 2.0x, dan xxhdpi memiliki rasio 3x.



Dari kiri ke kanan: mdpi – hdpi – xhdpi – xxhdpi

Menerapkan Design Pattern MVVM & Background Task

1. Melakukan implementasi ViewModel sehingga behaviour dari komponen tersebut memiliki sifat *disjoint* dengan komponen lainnya (relatif lebih independen)

MVVM adalah salah satu cara dalam membuat struktur kode. Kelebihan yang didapat ketika menggunakan paradigma ini adalah komponen UI dipisah dari business logic, business logic tidak disatukan dengan operasi database, sehingga dapat lebih mudah dimengerti dan dibaca sehingga lebih mudah untuk mengatur life cycle events dan aplikasinya akan berada di posisi state yang sama ketika ditinggalkan oleh user. Berikut implementasinya:

SharedViewModel.java

```

public class SharedViewModel extends ViewModel {
    private MutableLiveData<Item> selected = new MutableLiveData<>();

    public void setSelected(Item item){
        selected.setValue(item);
    }

    public MutableLiveData<Item> getSelected() {
        return selected;
    }
}

```

ViewModel tersebut akan digunakan untuk menghubungkan antara list jurnal dengan detail jurnal.

2. Melakukan Binding antara ViewModel dengan Model

Binding data dengan model. View yang memanipulasi data. Viewmodel yang handle data yang berubah ketika kategori dan nama ditambahkan.

TaskListFragment.java

```

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    binding = DataBindingUtil.inflate(inflater, R.layout.fragment_task_list, container, false);
    return binding.getRoot();
}

@Override
public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
    super.onViewCreated(view, savedInstanceState);
    SharedViewModel viewModel = new ViewModelProvider(requireActivity()).get(SharedViewModel.class);
    int count = 3;
    List<Item> list = new ArrayList<Item>();
    list.add(new Item("Selasa", "Proyek MPOS", "Fixing Bug"));
    list.add(new Item("Senin", "Training Lab RSE", "Belajar tentang ABS"));
    list.add(new Item("Rabu", "Beres-beres", "Beresin meja, kamar tidur, dan lemari"));

    if(list.size() == count) {
        RecyclerView adapter = new RecyclerView.Adapter(list);
        binding.recyclerView.setAdapter(adapter);
        adapter.setOnClickListener((v, position) -> {
            viewModel.setSelected(adapter.getItemAt(position));
            getParentFragmentManager().beginTransaction()
                .replace(R.id.container, detailsFragment)
                .addToBackStack(null)
                .commit();
        });
    }
}

```

Dapat terlihat bahwa data yang ditambahkan pada list diatas akan diproses pada RecyclerViewAdapter yang kemudian akan dilakukan binding di dalamnya. Begitu pula dengan detail jurnal yang juga diimplementasikan binding di dalamnya:

DetailsFragment.java

```
public class DetailsFragment extends Fragment {
    private FragmentDetailsBinding binding;

    public DetailsFragment() {
    }

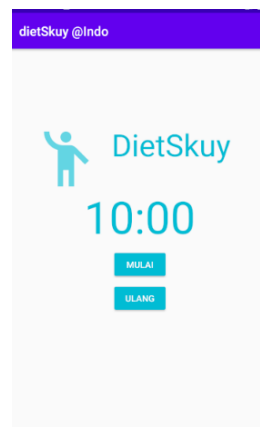
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        binding = DataBindingUtil.inflate(inflater, R.layout.fragment_details, container, false);
        return binding.getRoot();
    }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
        SharedViewModel viewModel = new ViewModelProvider(requireActivity()).get(SharedViewModel.class);
        viewModel.getSelected().observe(getViewLifecycleOwner(), item -> {
            binding.day.setText(item.getDay());
            binding.task.setText(item.getTask());
            binding.details.setText(item.getDetails());
        });
    }
}
```

Implementasinya seperti diatas, jadi view Model akan melakukan observasi dan kemudian melakukan binding untuk menampilkan pada layout fragment details sesuai dengan hari, task, dan detail dari list yang dipilih.

3. Background task

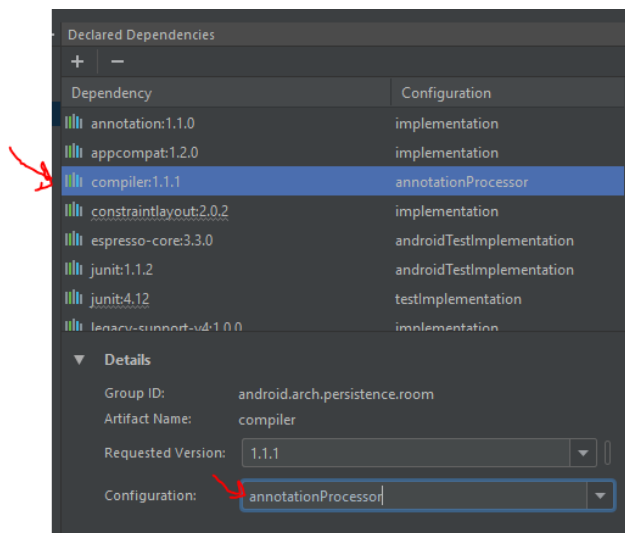
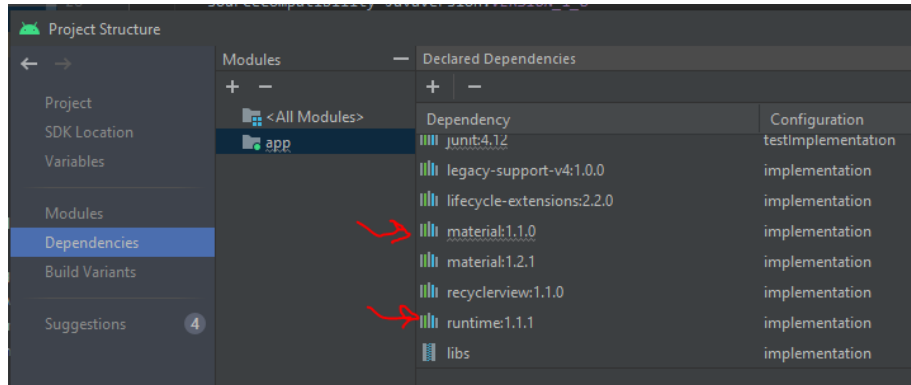
Timer akan tetap berjalan apabila keluar dari aplikasi.



Saya menggunakan CountdownTimer dengan membuat 3 method, yakni startTimer(), stopTimer() dan updateTimer(), di dalamnya terdapat Boolean untuk mengecek apakah timerRunning (waktu timer) sedang berjalan atau tidak. Sehingga ketika aplikasi sedang kehilangan fokus, timer tetap berjalan.

Menerapkan Data Persistence

1. Menggunakan **Room** Persistence Library untuk melakukan penyimpanan data di dalam perangkat. Data yang disimpan paling tidak terdiri dari 3 Entity. Terdapat 3 entity yaitu: user, jurnal, dan kategori yang ditambahkan. Pertama, tambahkan dependencies yang diperlukan:



MainAdapter.java

```
public class MainAdapter extends RecyclerView.Adapter<MainAdapter.ViewHolder> {
    private List<MainData> dataList;
    private Activity context;
    private RoomDB database;

    //konstruktor
    public MainAdapter(Activity context, List<MainData> dataList){
        this.context = context;
        this.dataList = dataList;
        notifyDataSetChanged();
    }

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        //inisialisasi view
        View view = LayoutInflater.from(parent.getContext())
            .inflate(R.layout.list_row_main, parent, false);
        return new ViewHolder(view);
    }
}
```

Digunakan untuk mengecek ketika ada dataset yang berubah, baik hapus, update, maupun create data yang terbaru. Menggunakan ViewHolder yang ada pada RecyclerView.Adapter

MainDao.java

```
@Dao
public interface MainDao {
    //insert query
    @Insert(onConflict = REPLACE)
    void insert(MainData mainData);

    //hapus query
    @Delete
    void delete(MainData mainData);

    //hapus semua query
    @Delete
    void reset(List<MainData> mainData);

    //update query
    @Query("UPDATE table_name SET text = :sText WHERE ID = :sID")
    void update(int sID, String sText);

    //dapatkan semua query data
    @Query("SELECT * FROM table_name")
    List<MainData> getAll();
}
```

Melakukan implementasi dao untuk query insert, delete, reset, update, dan getAll()

MainData.java

```
//definisikan nama tabel
@Entity(tableName = "table_name")
public class MainData implements Serializable {

    //create id column
    @PrimaryKey(autoGenerate = true)
    private int ID;

    //create text column
    @ColumnInfo(name = "text")
    private String text;

    public void setID(int ID) {
        this.ID = ID;
    }

    public void setText(String text) {
        this.text = text;
    }

    public int getID() {
        return ID;
    }

    public String getText() {
        return text;
    }
}
```

Melakukan implementasi untuk pembuatan entitas. Dimana terdapat id dan text untuk category data.

RoomDatabaseCategory.java

```
//inisialisasi adapter
adapter = new MainAdapter(RoomDatabaseCategory.this, dataList);
recyclerView.setAdapter(adapter);
```

Melakukan inisialisasi adapter terlebih dahulu seperti screenshot diatas, kemudian untuk menambah data atau mereset data, implementasinya berikut ini:

```
btAdd.setOnClickListener(view -> {
    //ambil string dari edit teks
    String sText = editText.getText().toString().trim();

    //cek kondisi
    if(!sText.equals("")){
        //ketika teksnya tidak kosong, inisialisasi MainData
        MainData data = new MainData();
        data.setText(sText);
        //melakukan insert teks pada database
        database.mainDao().insert(data);

        //hapus field edit text
        editText.setText("");

        //beri notifikasi ketika ada data yang di insert
        dataList.clear();
        dataList.addAll(database.mainDao().getAll());
        adapter.notifyDataSetChanged();
    }
});
```

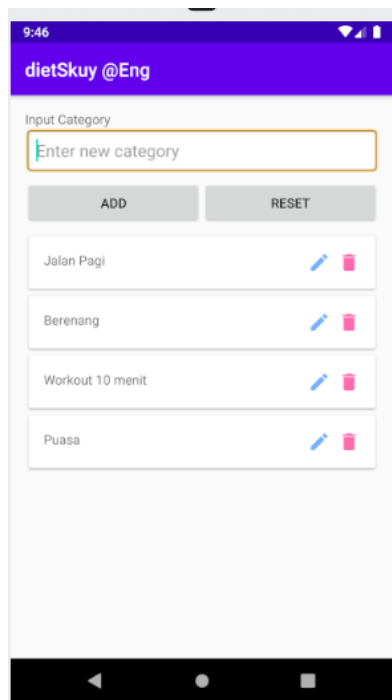
Memanfaatkan Dao dan adapter dalam melakukan perubahan pada room database. Begitupula dengan button reset:

```
btReset.setOnClickListener(view -> {
    //hapus semua data dari database
    database.mainDao().reset(dataList);

    //memberi notifikasi ketika semua data sudah selesai dihapus
    dataList.clear();
    dataList.addAll(database.mainDao().getAll());
    adapter.notifyDataSetChanged();
});
```

Adapter kemudian akan memberikan notifikasi lagi ketika dataset berubah.

Tampilan pada aplikasi seperti berikut ini:



Category dapat diubah-ubah sesuai keperluan.

Menerapkan Runtime Permission

1. Saya mendeklarasikan izin di manifest, kemudian aplikasi harus dibuat untuk meminta konfirmasi dari pengguna apakah akan memberikan izin atas resources (hardware maupun data). Ketika pengguna menyetujui akses maka aplikasi akan melanjutkan proses, namun apabila pengguna tidak menyetujui akses maka pengguna harus diarahkan ke penjelasan yang berisi “mengapa aplikasi membutuhkan izin atas resource tersebut”. Aplikasi Dietskuy dimenggunakan Image, maka persetujuan runtime permission nya baru dimunculkan ketika resource akan diakses. Berikut implementasinya:

Saya menggunakan implementasi untuk handle runtime permission `READ_EXTERNAL_STORAGE`, permission untuk hal ini pertama dideklarasikan pada *AndroidManifest.xml*:

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

Kemudian menggunakan `Intent.ACTION_PICK` dan melakukan setting pada intent untuk type image untuk dipilih melalui gallery:

RoomDatabaseCategory.java

```
private void pickImageFromGallery() {  
    //menggunakan intent untuk pick image  
    Intent intent = new Intent(Intent.ACTION_PICK);  
    intent.setType("image/*");  
    startActivityForResult(intent, IMAGE_PICK_CODE);  
}
```

```
@Override  
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  
    super.onActivityResult(requestCode, resultCode, data);  
    if (resultCode == RESULT_OK && requestCode == IMAGE_PICK_CODE) {  
        //set image to image view  
        mImageView.setImageURI(data.getData());  
    }  
}
```

Mengecek terlebih dahulu juga apakah request Runtime permission sudah diberikan atau belum:

RoomDatabaseCategory.java

```
if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.M){  
    if(checkSelfPermission(Manifest.permission.READ_EXTERNAL_STORAGE)  
        == PackageManager.PERMISSION_DENIED){  
        //izin belum diberikan, meminta request  
        String[] permissions = {Manifest.permission.READ_EXTERNAL_STORAGE};  
        //memberi pop-up untuk runtime permission  
        requestPermissions(permissions, PERMISSION_CODE);  
    }  
    else{ //izin sudah diberikan  
        pickImageFromGallery();  
    }  
}
```

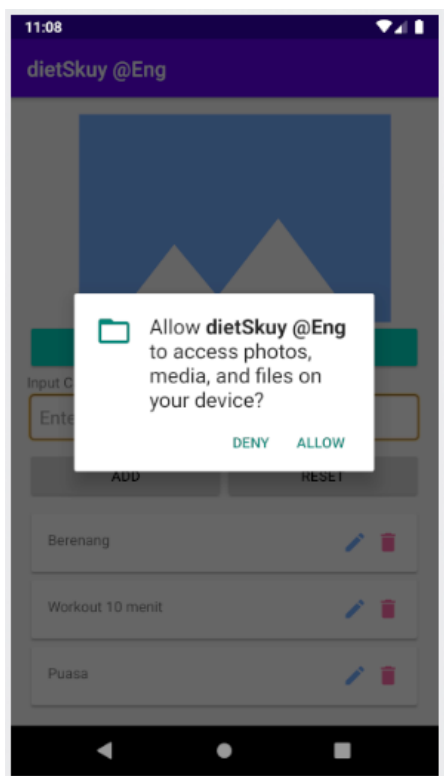
apabila belum, maka akan mengeluarkan pop-up untuk dapat memilih gambar melalui gallery. Sekarang melakukan handle hasil dari runtime permission. Jika pengguna klik ALLOW maka akan melanjutkan dalam memilih gambar, sedangkan jika klik DENY maka akan memberikan penjelasan mengapa perlu memberi izin:

RoomDatabaseCategory.java

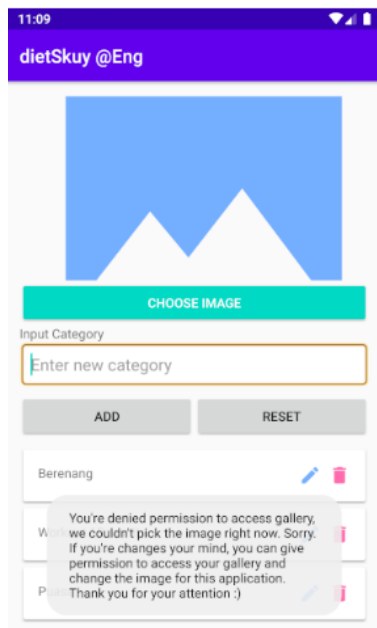
```
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    switch (requestCode){
        case PERMISSION_CODE:{
            if(grantResults.length > 0 && grantResults[0] ==
                PackageManager.PERMISSION_GRANTED){
                //izin sudah diberikan
                pickImageFromGallery();
            }
            else{//izin tidak diberikan
                Toast.makeText( context: this, R.string.explanation_denied_permission, Toast.LENGTH_LONG).show();
            }
        }
    }
}
```

Hal tersebut saya lakukan pada method `onRequestPermissionsResult`.

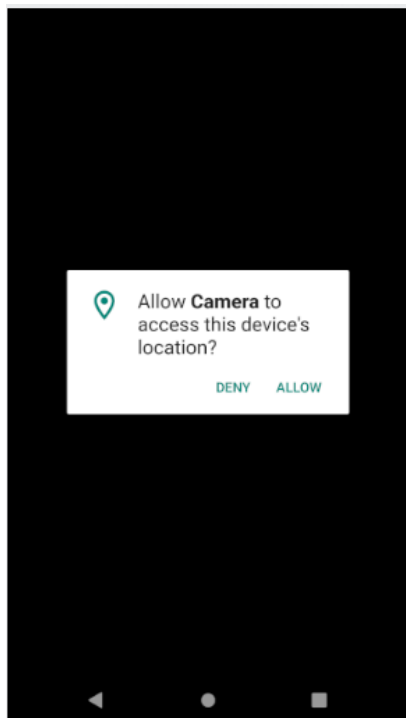
Berikut hasil tampilannya: Ketika pengguna melakukan klik pada *button* “choose image”, maka akan muncul *top-up* untuk menanyakan apakah pengguna ingin memberikan izin kepada aplikasi ini atau tidak.



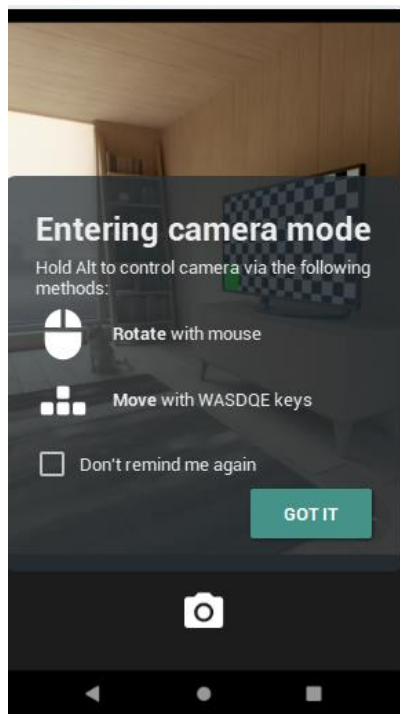
Apabila Pengguna klik *deny*, maka akan muncul toast dan beserta penjelasan apabila tidak memberikan izin kepada aplikasi ini:



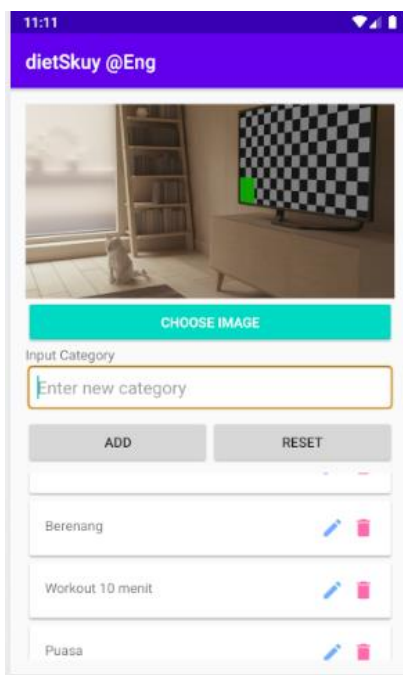
Apabila pengguna klik allow, maka akan mengecek terlebih dahulu ketersediaan gambar pada gallery. Apabila tidak ada gambar, maka akan diarahkan untuk mengambil gambar terlebih dahulu. Dibawah ini adalah jenis runtime permission dimana camera meminta akses lokasi device.



Tetapi pengguna diberikan opsi apabila tidak ingin lokasinya diketahui, tetap dapat mengakses kamera, seperti gambar dibawah ini:



Berikut hasil gambar yang sudah diambil dan diletakkan pada aplikasi:

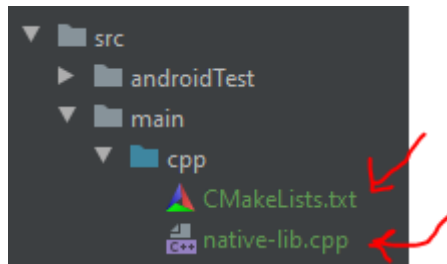


Gambar saya pasang sebesar 200dp x 400dp dan gambarnya akan menyesuaikan.

Memanfaatkan JNI (Java Native Interface)

1. Memanfaatkan fungsi library native C

Saya membuat implementasinya pada struktur dibawah ini:



2. Membuat sendiri sebuah library yang mengandung fungsi yg dibutuhkan kemudian di-compile dari source code native C

Native-lib.cpp

```
extern "C"
JNIEXPORT jint JNICALL
Java_id_ac_ui_cs_mobileprogramming_aulia_1rosyida_dietskuy_NewJurnalActivity_getTxtLen(JNIEnv *env,
                                                jobject NewJurnalActivity thiz,
                                                jstring txt) {

    // TODO: implement getTxtLen()
    const char* txt2 = env->GetStringUTFChars(txt, isCopy: 0);
    int len = strlen(txt2);

    return len;
}
```

Saya membuat Fungsi getTxtLen() yang menghitung jumlah character pada string. Hal ini saya gunakan untuk menghitung jumlah character pada quotes yang diinput oleh pengguna.

3. Melakukan pemanggilan library dengan JNI

Untuk melakukan pemanggilan library, melakukan definisi *CMakeLists.txt*

```
add_library( # Sets the name of the library.
            native-lib

            # Sets the library as a shared library.
            SHARED

            # Provides a relative path to your source file(s).
            native-lib.cpp)
```

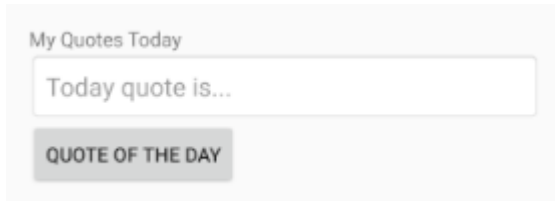
NewJurnalActivity.java

```
public native int getTxtLen(String txt);

private void enterText(){
    String txt2 = editText.getText().toString();
    int txtLen;
    txtLen = getTxtLen(txt2);
    textView.setText("text len: " + txtLen + " " + txt2);
}

@Override
public void onClick(View view) {
    if(view == b2){
        enterText();
    }
}
```

Setelah mendefinisikan `getTxtLen(String txt)`, kemudian diproses, sehingga ketika user melakukan klik pada button `b2`, yakni `Quotes Of The Day`, maka proses akan dilakukan oleh method `enterText()`. Dilakukan pemanggilan `getTxtLen()` untuk mendapatkan jumlah karakter teks yang kemudian akan ditampilkan pada halaman *activity_new_jurnal.xml*



Screenshot diatas adalah layout yang menggunakan JNI untuk menghitung jumlah karakter quote teks yang diinput oleh pengguna.

Menampilkan animasi dengan OpenGL

1. Aplikasi harus menerapkan sebuah halaman animasi dengan OpenGL. Animasi ini diperbolehkan untuk tidak ada hubungannya dengan fungsi utama aplikasi, Berikut implementasi saya:

AndroidManifest.xml

```
<uses-feature android:glEsVersion="0x00020000" android:required="true"/>
```

Membuat tag `uses-feature` untuk dapat menggunakan `glEsVersion` dengan `required true`.

OpenGLRenderer.java

```
public class OpenGLRenderer implements GLSurfaceView.Renderer {
    private double redValue = 1f;
    private static final double FLASH_DURATION = 1000;
    @Override
    public void onSurfaceCreated(GL10 gl10, EGLConfig eglConfig) {
        GLES20.glClearColor((float)redValue, 0, 0, 1f);
    }

    @Override
    public void onSurfaceChanged(GL10 gl10, int i, int i1) {
    }

    @Override
    public void onDrawFrame(GL10 gl10) {
        GLES20.glClearColor((float) redValue, 0,0, 1f);
        GLES20.glClear(GLES20.GL_COLOR_BUFFER_BIT);

        redValue = ((Math.sin(System.currentTimeMillis() * 2 * Math.PI / FLASH_DURATION) * 0.5) + 0.5);
    }
}
```

Implementasi `onSurfaceCreated` untuk inialisasi warna merah pada kotak, kemudian membuatnya jadi seolah berkedip menggunakan `onDrawFrame` dimana mengubah `redValue` agar semakin lama semakin gelap dan berubah kembali seperti awal menggunakan `currentTimeMillis()` agar perubahan hexa dapat berubah seiring waktu.

OpenGLView.java

```
public class OpenGLView extends GLSurfaceView {  
    public OpenGLView(Context context) {  
        super(context);  
        init();  
    }  
  
    public OpenGLView(Context context, AttributeSet attrs) {  
        super(context, attrs);  
        init();  
    }  
  
    private void init() {  
        setEGLContextClientVersion(2);  
        setPreserveEGLContextOnPause(true);  
        setRenderer(new OpenGLRenderer());  
    }  
}
```

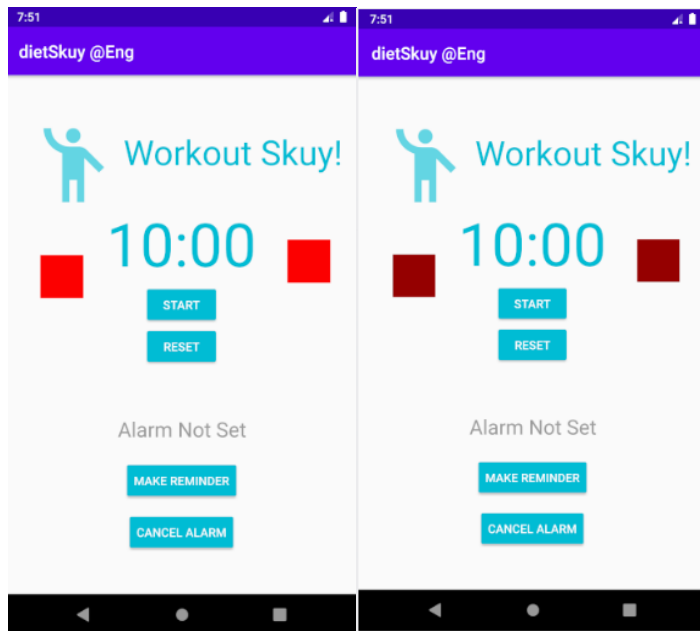
Setelah mengatur perubahan warna, maka OpenGLView melakukan extend GLSurfaceView yang akan mengurus bagaimana ia akan ditampilkan pada layout nantinya.

WorkoutActivity.java

```
openGLView = (OpenGLView) findViewById(R.id.openGLView);  
openGLView2 = (OpenGLView) findViewById(R.id.openGLView2);
```

```
@Override  
protected void onResume() {  
    super.onResume();  
    openGLView.onResume();  
    openGLView2.onResume();  
}  
  
@Override  
protected void onPause() {  
    super.onPause();  
    openGLView.onPause();  
    openGLView2.onPause();  
}
```

Mengatur proses berkedip dengan memanfaatkan activityLifecycle onResume() dan onPause() dimana animasi ini untuk menghemat memori akan pause dan resume bersamaan dengan fokus halaman workout. Animasi yang saya buat adalah animasi kotak berkedip sebagai penghias pada halaman workout. Animasi ini diharapkan dapat menjadi penyemangat saat pengguna sedang melakukan workout.



Warna kotak berkedipnya dimulai dari warna merah, kemudian berangsur-angsur menjadi lebih tua.

Memanfaatkan ConnectivityManager

Memanfaatkan ConnectivityManager untuk mengetahui status konektifitas yang dihubungkan dengan fitur tertentu. Berikut implementasi di dalam aplikasi:

Pada *AndroidManifest.xml* menambahkan akses permission INTERNET dan ACCESS_NETWORK_STATE:

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

Lalu mengimplementasikan ConnectivityManager di dalam activity yang diinginkan, pada aplikasi DietSkuy, saya menggunakannya pada list jurnal.

ListJurnalActivity.java

```
private ConnectivityManager mCManager;
private ConnectivityManager.NetworkCallback mCallback;
private AlertDialog mDialog;
```

Melakukan inisialisasi ConnectivityManager, NetworkCallback, dan AlertDialog yang akan digunakan untuk memunculkan pop-up ketika sedang tidak ada koneksi internet.

```
mCallback = new ConnectivityManager.NetworkCallback(){
    @Override
    public void onLost(@NonNull Network network) {
        //ketika koneksi network sedang lost, showing alert dialog
        AlertDialog.Builder builder = new AlertDialog.Builder(ListJurnalActivity.this, R.style.Theme
        .setTitle(R.string.no_internet_connection)
        .setMessage(R.string.check_internet_connection)
        .setPositiveButton(R.string.ok, new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                //closing app
                //methodnya dapat berjalan dengan minimum API 21
                finishAndRemoveTask();
            }
        });
        //memunculkan dialog
        mDialog = builder.show();
    }

    @Override
    public void onAvailable(Network network) {
        //menutup dialog pop-up jika networknya kembali sebelum user accept alert dialog
        if(mDialog!=null){
            mDialog.dismiss();
        }
    }
}
```

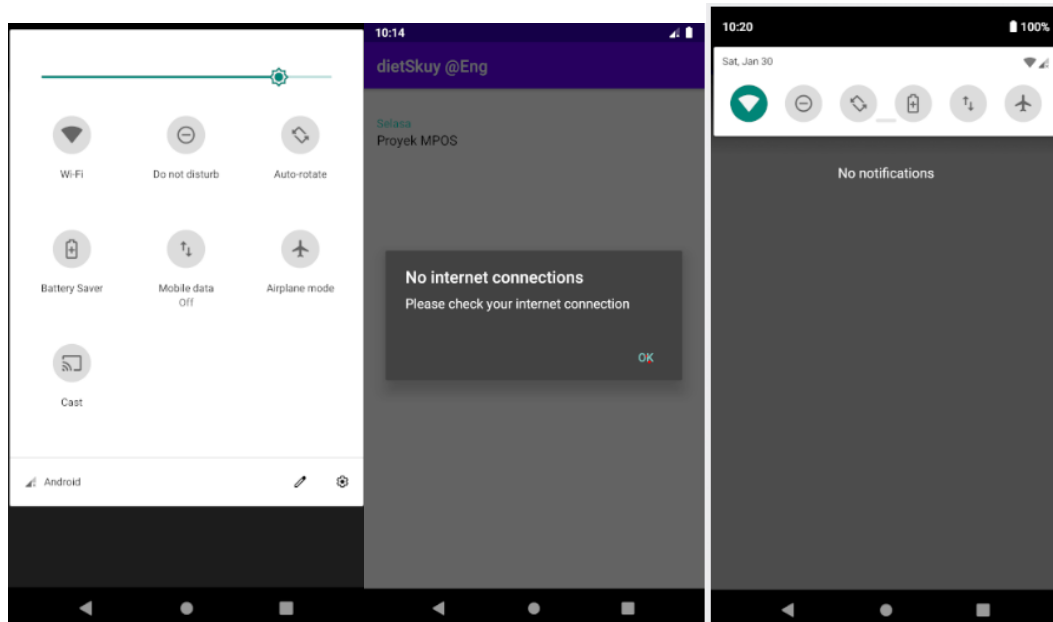
Di dalamnya melakukan builder AlertDialog yang akan muncul seperti pop-up. Lalu lakukan callback dan request network pada ConnectivityManager.

```
//melakukan callback dan network request
mCManager.registerNetworkCallback(request, mCallback);
```

Melakukan override method onStop() sehingga callback akan di unregister ketika halamannya di close.

```
@Override
protected void onStop() {
    //unregister callback ketika closing list jurnal activity
    mCManager.unregisterNetworkCallback(mCallback);
    super.onStop();
}
```

Ketika list jurnal activity diakses, ketika ia tidak ada koneksi internet, maka akan memunculkan pop-up seperti gambar dibawah ini:



Ketika wifi/mobile data dinyalakan kembali, maka pop-up otomatis sudah tertutup.

Menerapkan service background yang berhubungan dengan tampilan / aplikasi utama.

Menerapkan service background yang berhubungan dengan tampilan / aplikasi utama.
Cth: Service pemutar lagu tetap berjalan walaupun tampilan/aplikasi utama sedang lost focus.

Menerapkan Notifikasi atas event penting yang terjadi saat aplikasi dalam keadaan lost focus.

Menerapkan notifikasi atas event penting yang terjadi saat aplikasi dalam keadaan lost focus. Yakni menampilkan notifikasi mengenai reminder untuk melakukan workout dengan waktu yang dipilih sesuai dengan TimePicker. Berikut implementasinya:

NotificationHelper.java

```
public NotificationHelper(Context base) {  
    super(base);  
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {  
        createChannel();  
    }  
}
```

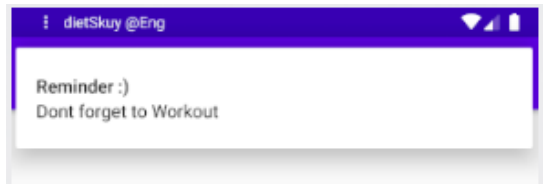
Ketika notificationHelper dipanggil, ia akan memanggil method createChannel(). Nah,

```
private void createChannel() {  
    NotificationChannel channel = new NotificationChannel(channelID, channelName, NotificationManager.IMPORTANCE_HIGH);  
    getManager().createNotificationChannel(channel);  
}
```

Di dalamnya memanggil getManager(),

```
}  
public NotificationManager getManager() {  
    if (mManager == null) {  
        mManager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);  
    }  
    return mManager;  
}
```

getManager() adalah method yang dipanggil pada AlertReceiver untuk mentrigger method ChannelNotification() yang akan melakukan return NotificationCompat.Builder. Hasilnya akan memunculkan kotak dialog seperti berikut:

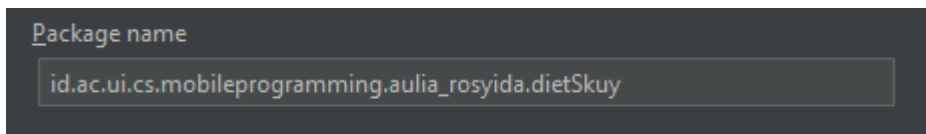


Notifikasi untuk workout berhasil diimplementasi.

Requirement Umum

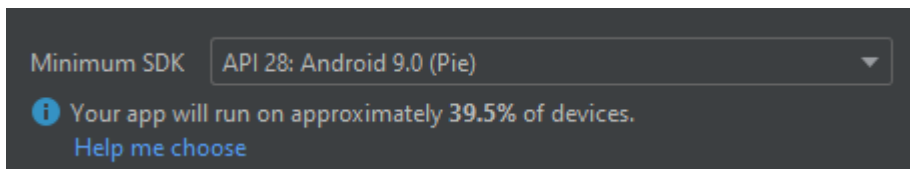
i. Package name

id.ac.ui.cs.mobileprogramming.aulia_rosyida.dietSkuy



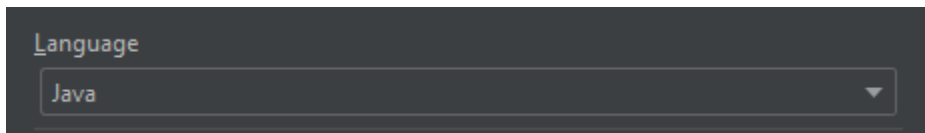
ii. API Android

API 28

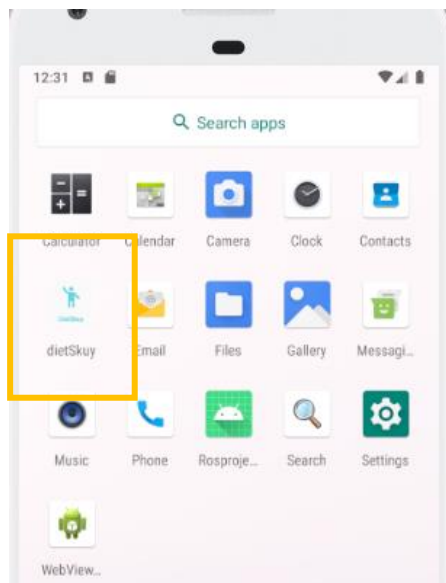


iii. Bahasa Pemrograman

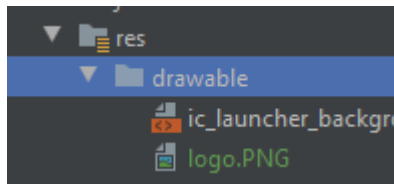
Java



iv. logo aplikasi Anda sendiri



Sebelumnya letakkan terlebih dahulu gambar logo pada folder res/drawable:



Implementasinya pada AndroidManifest.xml baris 7-9 dibawah ini:

