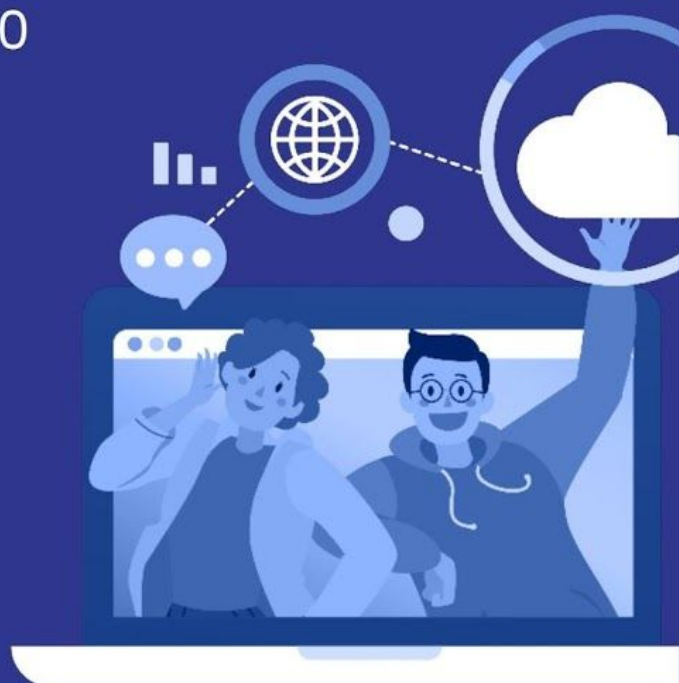


Modul Pelatihan

Vocational School Graduate Academy

Tema

Digital Talent Scholarship 2020



KATA PENGANTAR

Era Digitalisasi pada Industri 4.0 di Indonesia saat ini dihadapkan pada tantangan hadirnya permintaan dan penawaran talenta digital dalam mendukung perkembangan ekosistem industri teknologi. Tantangan tersebut perlu dihadapi salah satunya melalui kegiatan inovasi dan inisiasi dari berbagai pihak dalam memajukan talenta digital Indonesia, baik dari pemerintah maupun mitra kerja pemerintah yang dapat menyiapkan angkatan kerja muda sebagai talenta digital Indonesia. Kementerian Komunikasi dan Informatika melalui Badan Penelitian dan Pengembangan Sumber Daya Manusia sejak tahun 2018-2019, telah menginisiasi Program Digital Talent Scholarship yang telah berhasil dianugerahkan kepada 26.000 penerima pelatihan di bidang teknologi informasi dan komunikasi. Program Digital Talent Scholarship ini ditujukan untuk memberikan pelatihan dan sertifikasi tema-tema bidang teknologi informasi dan komunikasi, diharapkan menjadi bagian untuk memenuhi permintaan dan penawaran talenta digital Indonesia.

Tahun ini, Program Digital Talent Scholarship menargetkan pelatihan peningkatan kompetensi bagi 60.000 peserta yang bertujuan untuk meningkatkan keterampilan dan daya saing SDM bidang teknologi informasi dan komunikasi sebagai bagian dari program pembangunan prioritas nasional. Program pelatihan DTS 2020 ditujukan untuk meningkatkan keterampilan, keahlian angkatan kerja muda Indonesia, masyarakat umum dan aparatur sipil negara di bidang teknologi informasi dan komunikasi sehingga dapat meningkatkan produktivitas dan daya saing bangsa di era Industri 4.0.

Program DTS 2020 secara garis besar dibagi menjadi Tujuh akademi, yaitu: Fresh Graduate Academy (FGA), Program pelatihan berbasis kompetensi bersama GlobalTech yang ditujukan kepada peserta pelatihan bagi lulusan S1 bidang TIK dan MIPA, dan terbuka bagi penyandang disabilitas; Vocational School Graduate Academy (VSGA), Program pelatihan berbasis kompetensi nasional yang ditujukan kepada peserta pelatihan bagi lulusan SMK dan Pendidikan Vokasi bidang TI, Telekomunikasi, Desain, dan Multimedia; Coding Teacher Academy (CTA), Program pelatihan merupakan program pelatihan pengembangan sumberdaya manusia yang ditujukan kepada peserta pelatihan bagi Guru setingkat SMA/SMK/MA/SMP/SD di bidang pemrograman. Online Academy (OA), Program pelatihan OA merupakan program pelatihan Online di bidang Teknologi Informasi yang ditujukan kepada peserta pelatihan bagi Masyarakat umum, ASN, mahasiswa, dan pelaku industri; Thematic Academy (TA), Program pelatihan TA merupakan program pelatihan multisektor bagi pengembangan sumberdaya manusia yang ditujukan kepada peserta pelatihan dari jenjang dan multidisiplin yang berbeda; Regional Development Academy (RDA), Program pelatihan RDA merupakan program pelatihan pengembangan sumberdaya manusia yang ditujukan untuk meningkatkan kompetensi ASN di Kawasan Prioritas Pariwisata dan 122 Kabupaten Prioritas Pembangunan. Digital Entrepreneurship Academy (DEA), Program pelatihan DEA merupakan program pelatihan pengembangan sumberdaya manusia yang ditujukan kepada talenta digital di bidang Usaha Mikro, Kecil, dan Menengah (UMKM).

Selamat mengikuti Pelatihan Digital Talent Scholarship, mari persiapkan diri kita menjadi talenta digital Indonesia.

Jakarta, 2020

Kepala Badan Penelitian dan Pengembangan Sumber Daya Manusia
Kementerian Komunikasi dan Informatika Republik Indonesia
Dr. Ir. Basuki Yusuf Iskandar, MA

Pendahuluan

Unit kompetensi ini berhubungan dengan pengetahuan, keterampilan, dan sikap kerja yang dibutuhkan dalam membuat aplikasi berbasis mobile Android.

A. Tujuan Umum

Setelah mempelajari modul ini peserta latih diharapkan mampu membuat aplikasi berbasis mobile sederhana.

B. Tujuan Khusus

Adapun tujuan mempelajari unit kompetensi melalui buku informasi Pelatihan Mobile Programming ini guna memfasilitasi peserta latih sehingga pada akhir pelatihan diharapkan memiliki kemampuan dalam Merancang Database dan Data Persistence pada Mobile Data.

Latar belakang

Unit kompetensi ini dinilai berdasarkan tingkat kemampuan dalam merancang mobile programming. Adapun penilaian dilakukan dengan menggabungkan serangkaian metode untuk menilai kemampuan dan penerapan pengetahuan pendukung penting. Penilaian dilakukan dengan mengacu kepada Kriteria Unjuk Kerja (KUK) dan dilaksanakan di Tempat Uji Kompetensi (TUK), ruang simulasi atau workshop dengan cara:

- 1.1 Lisan
- 1.2 Wawancara
- 1.3 Tes tertulis
- 1.4 Demonstrasi
- 1.5 Metode lain yang relevan.

Deskripsi Pelatihan

Topik ini akan membahas tentang mendesain internal storage pada aplikasi berbasis mobile dan mendesain eksternal storage pada aplikasi berbasis mobile

Tujuan Pembelajaran

Setelah pertemuan selesai peserta pelatihan mampu:

- a. menjelaskan konseptual internal dan external storage pada aplikasi berbasis mobile
- b. menjelaskan internal dan external storage berdasarkan spesifikasi device mobile dan operating system berbasis mobile
- c. merancang media penyimpanan internal dan external dengan proses create, read, update, delete dan data berjalan dibangun sesuai kebutuhan.

Kompetensi Dasar

Mampu mendesain internal storage pada aplikasi berbasis mobile dan mendesain eksternal storage pada aplikasi berbasis mobile.

Indikator Hasil Belajar

- a. menjelaskan konseptual internal dan external storage pada aplikasi berbasis mobile
- b. menjelaskan internal dan external storage berdasarkan spesifikasi device mobile dan operating system berbasis mobile
- c. merancang media penyimpanan internal dan external dengan proses create, read, update, delete dan data berjalan dibangun sesuai kebutuhan.

INFORMASI PELATIHAN

Akademi	Vocational School Graduate Academy
Mitra Pelatihan	Perguruan Tinggi
Tema Pelatihan	Mobile Programmimg
Sertifikasi	<i>Sertifikasi Kompetensi Junior Mobile Programming dari BNSP</i>

Persyaratan Sarana Peserta/spesifikasi device Tools/media ajar yang akan digunakan	<p>Laptop dengan spesifikasi:</p> <ol style="list-style-type: none"> 1. RAM minimal 4 GB (disarankan 8 GB) 2. Laptop dengan 32/64-bit processor 3. Laptop dengan Operating System Windows 7,8,10, Linux, atau MAC OSX 4. Laptop dengan konektivitas WiFi dan memiliki webcam <p>Akses Internet Dedicated 126 kbps per peserta per perangkat</p>
Aplikasi yang akan di gunakan selamat pelatihan	<ol style="list-style-type: none"> 1. JDK (Java Development Kit) 2. Android Studio 3. Pencil 4. Sublime Text 5. XAMPP
Tim Penyusun	<ul style="list-style-type: none"> • Alif Akbar Fitrawan, S.Pd, M. Kom (Politeknik Negeri Banyuwangi); • Anwar, S.Si, MCs. (Politeknik Negeri Lhokseumawe); • Eddo Fajar Nugroho (BPPTIK Cikarang); • Eddy Tungadi, S.T., M.T. (Politeknik Negeri Ujung Pandang); • Fitri Wibowo (Politeknik Negeri Pontianak); • Ghifari Munawar (Politeknik Negeri Bandung); • Hetty Meileni, S.Kom., M.T. (Politeknik Negeri Sriwijaya) ; • I Wayan Candra Winetra, S.Kom., M.Kom (Politeknik Negeri Bali) ; • Irkham Huda (Vokasi UGM) ; • Josseano Amakora Koli Parera, S.Kom., M.T. (Politeknik Negeri Ambon) ; • I Komang Sugiarta, S.Kom., MMSI (Universitas Gunadarma) ;

	<ul style="list-style-type: none"> • Lucia Sri Istiyowati, M.Kom (Institut Perbanas) ; • Maksy Sendiang,ST,MIT (Politeknik Negeri Manado) ; • Medi Noviana (Universitas Gunadarma) ; • Muhammad Nashrullah (Politeknik Negeri Batam) ; • Nat. I Made Wiryana, S.Si., S.Kom., M.Sc. (Universitas Gunadarma) ; • Rika Idmayanti, ST, M.Kom (Politeknik Negeri Padang) ; • Rizky Yuniar Hakkun (Politeknik Elektronik Negeri Surabaya) ; • Robinson A.Wadu,ST.,MT (Politeknik Negeri Kupang) ; • Roslina. M.IT (Politeknik Negeri Medan) ; • Sukanto, SKom., MT. (Politeknik Negeri Semarang) ; • Syamsi Dwi Cahya, M.Kom. (Politeknik Negeri Jakarta) ; • Syamsul Arifin, S.Kom, M.Cs (Politeknik Negeri Jember) ; • Usmanudin (Universitas Gunadarma) ; • Wandy Alifha Saputra (Politeknik Negeri Banjarmasin) ;
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

INFORMASI PEMBELAJARAN

Unit Kompetensi	Materi pembelajaran	Kegiatan pembelajaran	Durasi Pelatihan	Rasio Praktek : Teori	Sumber pembelajaran
Kebutuhan dalam membuat aplikasi berbasis mobile Android	Modul dan Slide dalam Merancang Database dan Data Persistence pada Mobile Data	Daring/Online	Live Class 2 JP LMS 4 JP @ 45 Menit	60:40	

Materi Pokok

Merancang database dan data persistence pada mobile data.

Sub Materi Pokok

Mendesain internal storage pada aplikasi berbasis mobile
Mendesain eksternal storage pada aplikasi berbasis mobile
Mendesain sqlite database pada aplikasi berbasis mobile
Mendesain database mobile dengan model layer

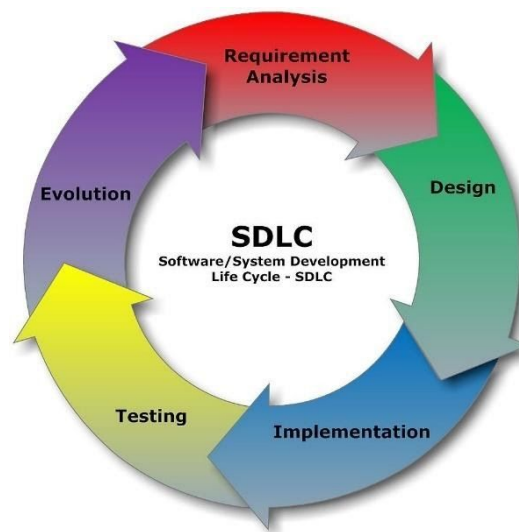
MERANCANG DATABASE DAN DATA PERSISTENCE PADA MOBILE DATA

A. Pengetahuan yang diperlukan dalam Merancang Database Dan Data Persistence Pada Mobile Data

Pengenalan SDLC

Sebelum membahas lebih jauh mengenai implementasi data pada aplikasi mobile, ada baiknya kita overview sekilas mengenai tahapan SDLC (Software Development Life Cycle) dalam pengembangan aplikasi. SDLC merupakan suatu tahapan utuh dalam pengembangan aplikasi mulai dari

tahap analisa kebutuhan, desain, implementasi, pengujian, sampai dengan tahapan evolusi (*continuous improvement*). Tahapan ini dilakukan secara sekuensial, artinya masing-masing tahapan harus sudah “selesai” terlebih dahulu sebelum lanjut ke tahap berikutnya. Umumnya SDLC dilakukan secara berkala, selama produk tersebut masih digunakan oleh user. Kenapa hal ini dilakukan ? Karena kebutuhan user itu bertumbuh, termasuk kebutuhan dalam lingkup aplikasi.



Gambar 1. Tahapan Software Development Life Cycle (SDLC)

Tahapan SDLC adalah sebagai berikut :

- Requirement Analysis : Penggalan kebutuhan aplikasi termasuk kebutuhan fungsional dan non fungsionalnya.
- Design : Perancangan aplikasi termasuk perancangan data, perancangan kelas, desain UI, dlsb.
- Implementasi : Dapat juga dikatakan tahap coding / pembuatan source code aplikasi
- Testing : Tahapan untuk menguji kesiapan aplikasi
- Evolution : Tahap pengembangan aplikasi yang berkelanjutan termasuk penambahan fitur, bug fixing, penyesuaian kebutuhan, dlsb.

Data Storage

Penyimpanan data pada android terdiri dari dua jenis data, yaitu : (1) data temporary dan (2) data persistence. Data temporary umumnya diproses melalui internal storage dalam bentuk cache. Sedangkan data persistence disimpan secara permanen dalam suatu file. Di bawah ini terdapat beberapa opsi data storage yaitu :

- Shared Preferences : menyimpan data primitive berupa key-value pairs.
- Internal Storage : menyimpan data private pada device memory.
- External Storage : menyimpan data public pada shared external storage.
- Database : menyimpan data struktur pada sebuah database
- Network Connection : menyimpan data pada web server
- Bundle Class : memetakan data string dalam bentuk map
- File IO : menyimpan data berupa file pada internal storage

Mengapa Data Persistence di Perlukan?

Android dapat di-shutdown dan di-restart kapanpun dibutuhkan, namun bagaimana ketika :

- merotasi screen,
- mengubah bahasa,
- aplikasi berjalan secara background process dan disimpan dalam short-in memory dan menekan tombol Back.

Apakah yang akan terjadi ? data hilang !

Lalu bagaimana solusinya?,

Android menyediakan beberapa solusi untuk menangani data persistence, tergantung dari sifatnya apakah private atau public. Beberapa solusi yang dipilih harus sesuai dengan kebutuhan spesifiknya, seperti apakah data yang disimpan akan bersifat private atau boleh diakses secara bebas (public) serta berapa banyak space yang

dibutuhkannya. Android juga menyediakan cara untuk membuka data privat ke aplikasi lain (apabila diperlukan), melalui Content Provider.

Android File System

Penyimpanan android terbagi menjadi 2 bagian yaitu :

- Internal Storage – direktori private yang hanya diakses pada aplikasi itu sendiri. Karakteristik dari internal storage : selalu tersedia (dapat diakses), menggunakan file system dari device, hanya bisa diakses pada aplikasi itu sendiri, kecuali jika secara eksplisit diatur agar dapat dibaca / ditulis dan ketika aplikasi di-uninstall, sistem akan menghapus semua file aplikasi dari internal storage. Internal Storage paling baik digunakan jika tidak ada yang boleh mengakses file selain aplikasi itu sendiri. Internal Storage bersifat private, aplikasi akan selalu meminta izin untuk read/write data, direktori penyimpanan persistence – **getFilesDir()** dan direktori penyimpanan temporary – **getCacheDir()**.
- External Storage – direktori public dapat diakses secara lebih luas pada aplikasi itu sendiri dan aplikasi lainnya. Karakteristik dari external storage : Tidak selalu tersedia (dapat dihapus), menggunakan file system dari device atau dari penyimpanan fisik eksternal seperti SD Card, World-readable, semua aplikasi dapat membaca file tersebut dan ketika aplikasi di-uninstall, system tidak akan menghapus file private dari aplikasi tersebut. External storage paling baik digunakan jika File tidak memerlukan batasan akses (semua public), File dapat di-share ke aplikasi lain dan membolehkan pengguna untuk mengakses file melalui komputer (via USB Connection).

Contoh Pembuatan File :

Di bawah ini adalah contoh *Instance Object* serta penggunaan parameter `getFileDir` menggunakan bahasa pemrograman java.

```
File file = new File(context.getFilesDir(), filename);
```

Untuk fungsi-fungsi pengelolaan filenya, menggunakan standard java.io file operator atau stream.

1. Mendesain internal storage pada aplikasi berbasis mobile

a. Internal Storage

Langkah menulis File Private ke Internal Storage

Untuk membuat dan menuliskan file private ke internal storage :

1. Panggil `openFileOutput()` dengan nama file dan mode pengoperasiannya. Proses ini mengembalikan `FileOutputStream`.
2. Tulis pada file melalui method `write()`.
3. Tutup aliran (*stream*) melalui method `close()`.

Contoh Kode Program untuk Penulisan File :

```
String filename = "myfile";  
String fileContents = "Hello world!";  
  
FileOutputStream outputStream;  
  
try {  
    outputStream=openFileOutput(filename, Context.MODE_PRIVATE);  
    outputStream.write(fileContents.getBytes());  
    outputStream.close();  
} catch (Exception e) {  
    e.printStackTrace();  
}
```

`MODE_PRIVATE` akan membuat file (atau mengganti / replace file dengan nama yang sama) dan membuatnya private untuk aplikasi anda. Mode lain yang tersedia antara lain :

- MODE_APPEND
- MODE_WORLD_READABLE *
- MODE_WORLD_WRITEABLE *

Catatan : * sudah tidak digunakan lagi sejak API level 17 (dimulai dari android N) karena penggunaannya akan berdampak pada SecurityException.

Langkah membaca File Private ke Internal Storage

Untuk membaca file *private* dari *internal storage* :

1. Panggil `openFileInput()` dan memberikan nama file yang ingin dibaca.
Proses ini mengembalikan `FileInputStream`.
2. Baca byte dari file dengan method `read()`
3. Tutup aliran (stream) dengan method `close()`

```
String FILENAME = "hello_file.txt";
FileInputStream fin = openFileInput(FILENAME);

StringBuffer fileContent = new StringBuffer("");

byte[] buffer = new byte[1024];

while((n=fin.read(buffer))!=-1){

fileContent.append(new String(buffer,0,n));

}

fin.close();
```

Untuk penyimpanan file cache bisa dilakukan dengan 2 cara yaitu :

1. `getCacheDir()`– digunakan untuk membuka file pada direktori internal storage.
2. `getCacheExternalDir()`– digunakan untuk membuka cache pada external storage

Contoh kode untuk menuliskan file Cache :

`createTempFile()` akan membuat file pada direktori cache yang tersimpan secara privat pada aplikasi.

```
Private File getTempFile(Context context, String url) { File file;  
try {  
  
    String fileName = Uri.parse(url).getLastPathSegment();  
  
    file = File.createTempFile(fileName, null, context.getCacheDir());  
  
} catch (IOException e) {  
  
    // Error while creating file  
  
}  
  
return file;  
  
}
```

Beberapa method penting lainnya :

1. `getFilesDir()` – mendapatkan path absolut ke direktori file pada internal storage.
2. `getDir()` – membuat (atau membuka yang sudah ada) direktori pada internal storage.
3. `deleteFile()` – menghapus file yang disimpan pada internal storage.
4. `fileList()` – mengembalikan daftar file yang saat ini telah disimpan dalam *internal storage*.

Mendesain eksternal storage pada aplikasi berbasis mobile

a. External Storage

Android menyediakan metode penyimpanan file external pada ponsel atau tablet yang terbagi berdasarkan beberapa jenis, yaitu:

1) Untuk ponsel Android generasi awal biasanya hanya dibagi menjadi dua bagian, yaitu:

- Internal storage
- SD card

2) Untuk Smartphone yang tidak memiliki slot SD card juga dibagi menjadi dua bagian, yaitu:

- Internal/System storage,
- Phone/USB Storage.

3) Untuk smartphone dan tablet Android saat ini pada umumnya dibagi menjadi tiga bagian, yaitu:

- Internal/System storage
- Phone/USB Storage
- SD card.

b. Konfigurasi Pada AndroidManifest.xml

Agar bisa membaca atau menulis file pada penyimpanan eksternal, aplikasi harus memperoleh READ_EXTERNAL_STORAGE atau izin sistem WRITE_EXTERNAL_STORAGE yang di daftarkan pada **AndroidManifest.xml**. Misalnya seperti baris kode dibawah :

```
<manifest>

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

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

</manifest>
```

c. Fungsi Penting dalam Penyimpanan Eksternal

- `getExternalStorageState()` : untuk memeriksa apakah media tersebut tersedia.
- `getExternalStorageDirectory()` : nilai dari fungsi ini menghasilkan lokasi direktori penyimpanan eksternal / eksternal utama.
- `getExternalStoragePublicDirectory(String type)` : nilai dari fungsi ini menghasilkan lokasi direktori penyimpanan eksternal / eksternal utama sesuai dari isi Argumen type yang ada pada fungsi tersebut.

d. External Storage State

Dibawah ini ada beberapa contoh baris kode penggunaan External Storage State :

```
/* Periksa apakah penyimpanan eksternal
tersedia untuk membaca dan menulis */

public boolean isExternalStorageWritable() {
    String state = Environment.getExternalStorageState();
    if (Environment.MEDIA_MOUNTED.equals(state)) {
        return true;
    }
    return false;
}
```

```
/* Periksa apakah penyimpanan eksternal
setidaknya tersedia untuk dibaca */

public boolean isExternalStorageReadable() {
    String state = Environment.getExternalStorageState();
    if (Environment.MEDIA_MOUNTED.equals(state) ||
        Environment.MEDIA_MOUNTED_READ_ONLY.equals(state)) {
        return true;
    }
    return false;
}
```

Jenis-jenis status nilai balik (return value) dari fungsi `getExternalStorageState` adalah sebagai berikut :

- **MEDIA_UNKNOWN** : kartu SD Card tidak dapat dikenali
- **MEDIA_REMOVED** : status tidak ada kartu SD Card yang tersedia
- **MEDIA_UNMOUNTED** : Status penyimpanan jika media ada tetapi tidak dipasang.
- **MEDIA_MOUNTED** : Status penyimpanan jika media ada dan terpasang dengan akses baca / tulis.
- **MEDIA_MOUNTED_READ_ONLY** : Status penyimpanan jika media ada dan terpasang dengan akses hanya baca.

Contoh penyimpanan pada external file directory :

```
File file = new File(Environment.getExternalStorageDirectory(), "DemoFile.txt");
```

Contoh penyimpanan pada external storage public directory :

```
File file = new
File(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_
DOCUMENTS), "DemoFile.txt");
```

e. Jenis-jenis Direktori Publik

Adapun nilai-nilai statik yang sering digunakan dari fungsi `getExternalStoragePublicDirectory(String type)` di antaranya :

- **DIRECTORY_DOCUMENTS** : Direktori standar tempat meletakkan dokumen yang telah dibuat oleh pengguna.
- **DIRECTORY_DOWNLOADS** : Direktori standar tempat meletakkan file yang telah diunduh oleh pengguna.
- **DIRECTORY_MOVIES** : Direktori standar untuk menempatkan film yang tersedia bagi pengguna.
- **DIRECTORY_PICTURES** : Direktori standar tempat meletakkan gambar yang tersedia bagi pengguna.

f. Membuat fungsi Create, Update, Read, dan Delete

- Contoh kode implementasi untuk membuat file (Create) :

```
void buatFile() {
    String isiFile = "Coba Isi Data File Text";
    String state = Environment.getExternalStorageState();

    if (!Environment.MEDIA_MOUNTED.equals(state)) {
        return;
    }
    File file = new File(Environment.getExternalStorageDirectory(),
        FILENAME);

    FileOutputStream outputStream = null;
    try {
        file.createNewFile();
        outputStream = new FileOutputStream(file, true);
        outputStream.write(isiFile.getBytes());
        outputStream.flush();
        outputStream.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```


- Contoh kode implementasi untuk mengupdate file (Update) :

```
void ubahFile() {
    String ubah = "Update Isi Data File Text";
    String state = Environment.getExternalStorageState();

    if (!Environment.MEDIA_MOUNTED.equals(state)) {
        return;
    }
    File file = new File(Environment.getExternalStorageDirectory(),
FILENAME);

    FileOutputStream outputStream = null;
    try {
        file.createNewFile();
        outputStream = new FileOutputStream(file, false);
        outputStream.write(ubah.getBytes());
        outputStream.flush();
        outputStream.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

- Contoh kode untuk membaca file (Read) :

```
public void bacaFile() {
    File sdcard = Environment.getExternalStorageDirectory();
    File file = new File(sdcard, FILENAME);

    if (file.exists()) {

        StringBuilder text = new StringBuilder();

        try {
            BufferedReader br = new BufferedReader(new FileReader(file));

            String line = br.readLine();

            while (line != null) {
                text.append(line);
                line = br.readLine();
            }
            br.close();
        } catch (IOException e) {
            System.out.println("Error " + e.getMessage());
        }
    }
}
```

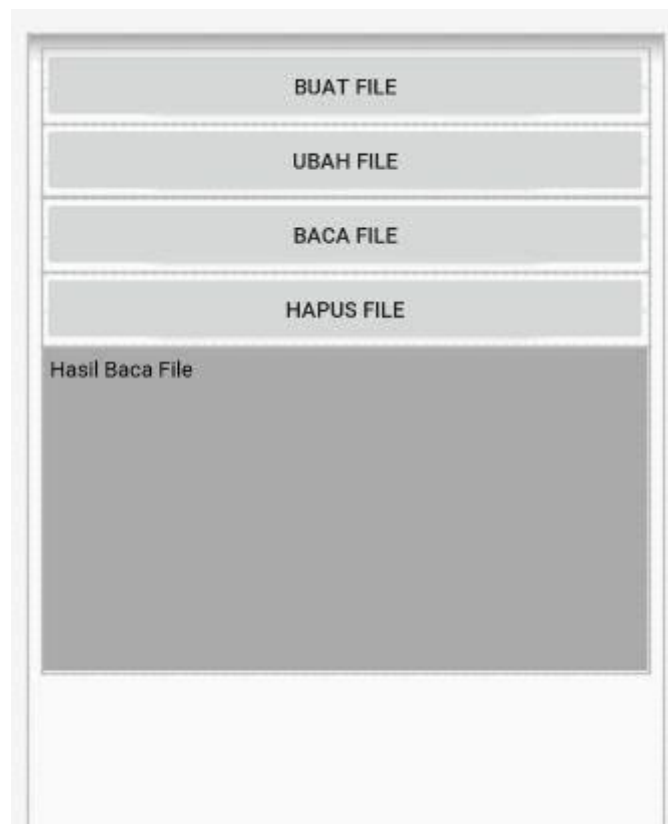
```
    }  
    textBaca.setText(text.toString());  
  }  
}
```

- Contoh kode untuk menghapus file (Delete) :

```
void hapusFile() {  
    File file = new File(Environment.getExternalStorageDirectory(),  
        FILENAME);  
    if (file.exists()) {  
        file.delete();  
    }  
}
```

LATIHAN DATA PERSISTENCE

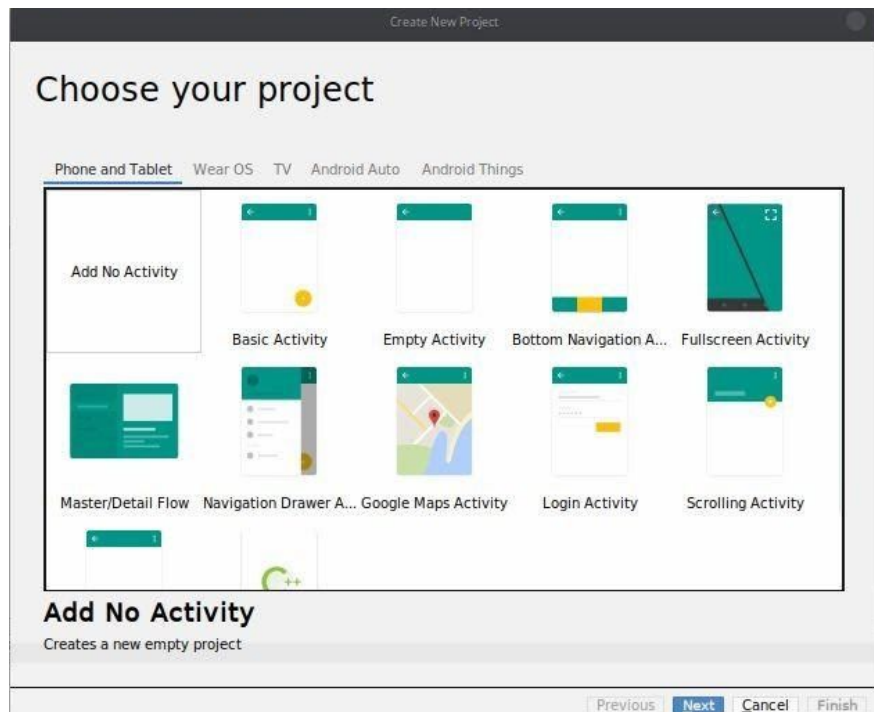
Saat ini kita akan coba implementasikan data persistence pada sebuah aplikasi sederhana. Aplikasi ini memanfaatkan penyimpanan file berupa *internal storage*. Berikut adalah contoh tampilan akhir dari aplikasi yang diharapkan. Aplikasi memiliki fungsi untuk membuat file, mengubah file, membaca file dan menghapus file.



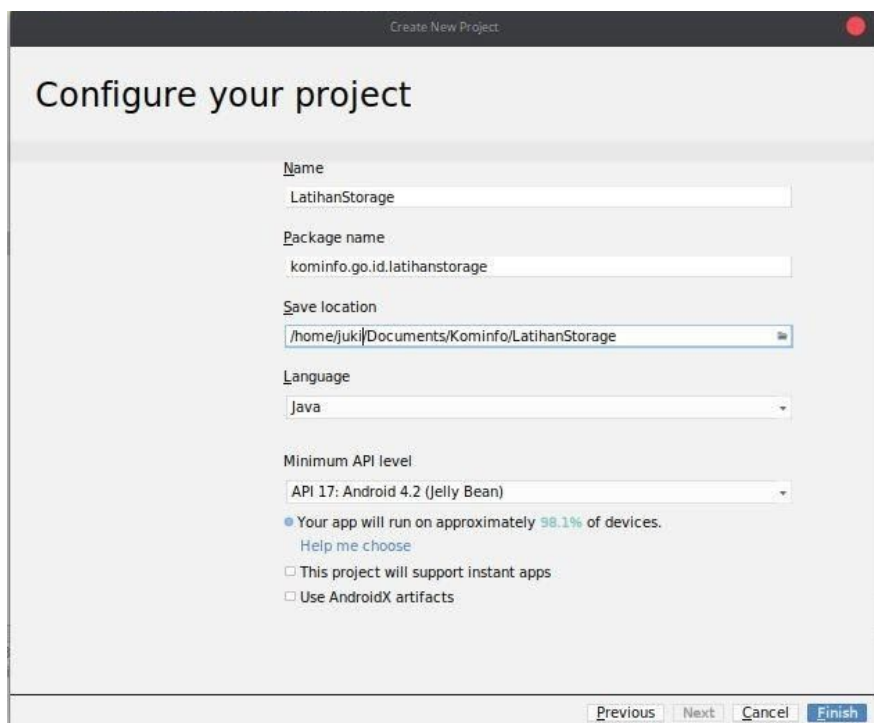
Gambar 2. Tampilan Akhir Aplikasi Internal Storage

Untuk membuat aplikasi tersebut silahkan ikuti langkah berikut :

1. Buatlah sebuah proyek baru pada Android Studio dan beri nama proyeknya "LatihanStorage".



Gambar 3. Buatlah Sebuah Proyek Baru pada Android Studio



Gambar 4. Berikan Nama Proyek “LatihanStorage”

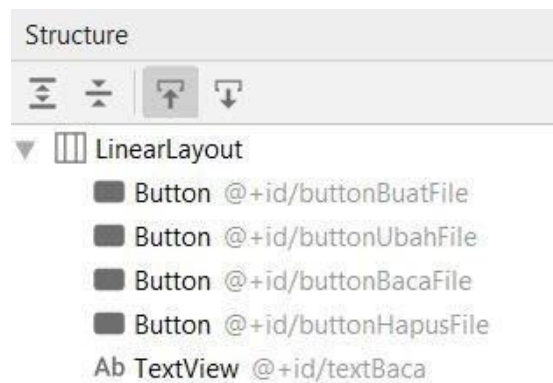
2. Buatlah sebuah Activity, beri nama kelas InternalActivity.java dan sebuah Layout dengan nama **activity_internal.xml**

Gambar 5. Tambahkan InternalActivity

3. Membuat sebuah view di dalam layout **activity_internal.xml** (sesuaikan dengan tampilan berikut ini)



Gambar 6. Tampilan View pada file activity_internal.xml



Gambar 7. Struktur Layout pada file activity_internal.xml

Berikut ini adalah kode XML pada file **activity_internal.xml** :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="10dp"
    android:orientation="vertical">
```

```

<Button
    android:id="@+id/buttonBuatFile"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Buat File" />

<Button
    android:id="@+id/buttonUbahFile"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Ubah File" />

<Button
    android:id="@+id/buttonBacaFile"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Baca File" />

<Button
    android:id="@+id/buttonHapusFile"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Hapus File" />

<TextView
    android:id="@+id/textBaca"
    android:layout_width="match_parent"
    android:layout_height="210dp"
    android:background="@android:color/darker_gray"
    android:ems="10"
    android:gravity="top|left"
    android:padding="5dp"
    android:textColorHint="#000"
    android:hint="Hasil Baca File"
    android:inputType="textMultiLine" />

</LinearLayout>

```

4. Pada kelas InternalActivity.java tambahkan baris fungsi CREATE, READ, UPDATE, dan DELETE.

```
Package kominfo.go.id.latihanstorage;
```

```
import android.os.Environment;
```

```
import android.support.v7.app.AppCompatActivity;
```

```

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.IOException;

public class InternalActivity extends AppCompatActivity implements
View.OnClickListener {
    public static final String FILENAME = "namafile.txt";
    Button buatFile, ubahFile, bacaFile, deleteFile;
    TextView textBaca;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_internal);
        buatFile = findViewById(R.id.buttonBuatFile);
        ubahFile = findViewById(R.id.buttonUbahFile);
        bacaFile = findViewById(R.id.buttonBacaFile);
        deleteFile = findViewById(R.id.buttonHapusFile);
        textBaca = findViewById(R.id.textBaca);

        buatFile.setOnClickListener(this);
        ubahFile.setOnClickListener(this);
        bacaFile.setOnClickListener(this);
        deleteFile.setOnClickListener(this);
    }

    void buatFile() {
        String isiFile = "Coba Isi Data File Text";
        File file = new File(getFilesDir(), FILENAME);

        FileOutputStream outputStream = null;
        try {
            file.createNewFile();
            outputStream = new FileOutputStream(file, true);
            outputStream.write(isiFile.getBytes());
            outputStream.flush();
            outputStream.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

    }
}

void ubahFile() {
    String ubah = "Update Isi Data File Text";

    File file = new File(getFilesDir(), FILENAME);

    FileOutputStream outputStream = null;
    try {
        file.createNewFile();
        outputStream = new FileOutputStream(file, false);
        outputStream.write(ubah.getBytes());
        outputStream.flush();
        outputStream.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

void bacaFile() {
    File sdcard = getFilesDir();
    File file = new File(sdcard, FILENAME);

    if(file.exists()) {

        StringBuilder text = new StringBuilder();

        try {
            BufferedReader br = new BufferedReader(new FileReader(file));

            String line = br.readLine();

            while (line != null) {
                text.append(line);
                line = br.readLine();
            }
            br.close();
        } catch (IOException e) {
            System.out.println("Error " + e.getMessage());
        }
        textBaca.setText(text.toString());
    }
}

void hapusFile() {
    File file = new File(getFilesDir(), FILENAME);
    if (file.exists()) {

```



```

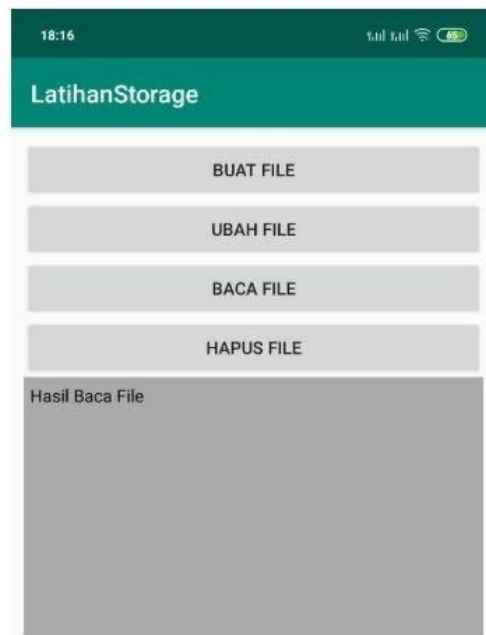
        file.delete();
    }
}

@Override
public void onClick(View v) {
    jalankanPerintah(v.getId());
}

public void jalankanPerintah(int id) {
    switch (id) {
        case R.id.buttonBuatFile:
            buatFile();
            break;
        case R.id.buttonBacaFile:
            bacaFile();
            break;
        case R.id.buttonUbahFile:
            ubahFile();
            break;
        case R.id.buttonHapusFile:
            hapusFile();
            break;
    }
}
}

```

5. Jalankan Aplikasi pada Device / Emulator.



Gambar 8. Tampilan aplikasi saat menekan tombol BUAT FILE

Device File Explorer				
Xiaomi Mi-4c Android 7.1.2, API 25				
Name	Permissions	Date	Size	
▶ id.ac.mercubuana.prod.joinumb	drwxrwx--x	1970-01-01 19:11	4 KB	
▶ iflix.play	drwxrwx--x	1970-01-01 19:11	4 KB	
▶ in.nitishverma.nmaptutorial	drwxrwx--x	1970-01-01 19:11	4 KB	
▼ kominfo.go.id.latihanstorage	drwxrwx--x	1970-01-01 19:11	4 KB	
▶ cache	drwxrwx--x	2019-04-27 19:44	4 KB	
▶ code_cache	drwxrwx--x	2019-04-27 19:44	4 KB	
▼ files	drwxrwx--x	2019-04-27 19:44	4 KB	
namafile.txt	-rw-----	2019-04-27 19:50	25 B	
▶ me.twrp.twrpapp	drwxrwx--x	1970-01-01 19:11	4 KB	

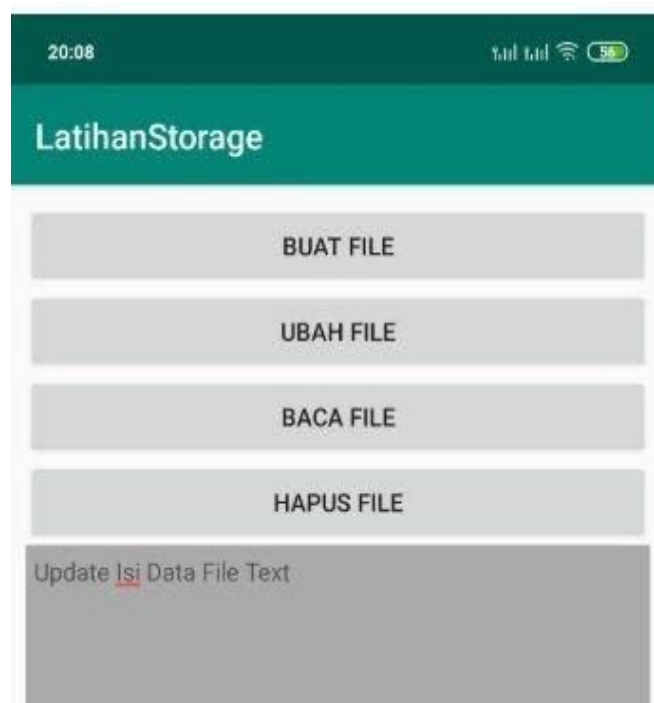
Gambar 9. File yang telah berhasil dibuat

Path location pada *internal storage* berada didalam data/data/[package aplikasi]/files/namafile.txt

Catatan : apabila file tidak ditemukan silahkan restart terlebih dahulu (atau klik kanan pada device explorer lalu *synchronize*)



Gambar 10. Contoh tampilan saat menekan tombol BACA FILE



Gambar 11. Contoh tampilan saat menekan tombol UBAH FILE

Device File Explorer			
Xiaomi Redmi Note 7 Android 9, API 28			
Name	Permissions	Date	Size
▶ com.xiaomi.xmsf	drwxrwx--x	1970-01-01 22:02	4 KB
▶ id.co.babe	drwxrwx--x	1970-01-01 22:02	4 KB
▼ kominfo.go.id.latihanstorage	drwxrwx--x	1970-01-01 22:02	4 KB
▶ cache	drwxrws--x	2019-04-27 20:06	4 KB
▶ code_cache	drwxrws--x	2019-04-27 20:06	4 KB
▶ files	drwxrwx--x	2019-04-27 20:12	4 KB
▶ org.codeaurora.ims	drwxrwx--x	1970-01-01 22:02	4 KB
▶ org.telegram.messenger	drwxrwx--x	1970-01-01 22:02	4 KB

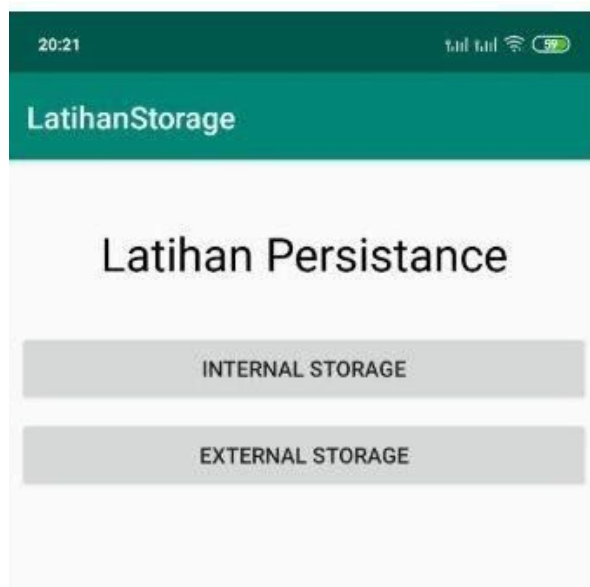
Gambar 12. File “namafile.txt” akan hilang saat menekan tombol HAPUS FILE

LATIHAN MANDIRI

Berdasarkan latihan yang telah dilakukan sebelumnya (pada BAB III), penyimpanan masih dilakukan secara *internal storage*, selanjutnya silahkan tambahkan fitur agar aplikasi dapat melakukan penyimpanan juga melalui *external storage*.

Langkah 1

Tambahkan sebuah activity yang memuat 2 *button* yang berfungsi untuk menyimpan data secara *internal storage* maupun *external storage* sebagaimana tampilan berikut dengan nama **MainActivity.java**



Gambar 13. Tampilan MainActivity.java

Langkah 2

Sesuaikan dengan fungsi *button* yang telah dibuat pada **Langkah 1**, yakni apabila dipilih tombol *internal storage* maka akan memunculkan **InternalActivity.java**, sedangkan apabila dipilih tombol *external storage* akan memunculkan halaman baru yang memiliki tampilan dan fungsi yang sama dengan *internal storage* (yang membedakan hanya lokasi penyimpanan menggunakan *external storage*).

Mendesain sqlite database pada aplikasi berbasis mobile

a. Overview SQLite

SQLite merupakan basis data tertanam (*embedded*) yang awalnya dikembangkan untuk menyimpan informasi/data pada perangkat sistem kendali missile. Dikembangkan pertama oleh D. Richard Hipp tahun 2000 Tujuan utama dari SQLite adalah untuk memudahkan program basis data berjalan tanpa perlu adanya instalasi basis data.

Tahun 2000 versi 1.0 SQLite direlease, menggunakan dasar GDBM. Versi 2.0 tahun 2001 dan versi 3.0 tahun 2004 (terakhir adalah versi 3.28.0 release tahun 2019) dengan ada banyak penambahan fitur misal internasionalisasi

b. Pengguna SQLite

SQLite banyak digunakan sebagai penyimpanan oleh beberapa aplikasi / vendor terkenal.

- Adobe – menggunakan SQLite di aplikasi Photoshop dan Acrobat Reader
- Apple – beberapa fitur apple menggunakannya (Apple mail, Safari web browser)
- Mozilla – Web browser nya menggunakan SQLite
- Google – Google Desktop dan Google Gears menggunakannya.
- McAfee – menggunakan SQLite di beberapa program antivirusnya
- PHP – PHP menyisipkan built in SQLite 2 dan SQLite 3 di librarinya
- Python – SQLite di paketkan dalam Bahasa pemrograman Phyton

c. Pembuatan Database dan tabel SQLite

Pembuatan Database SQLite dapat dilakukan dengan beberapa cara antara lain dengan Perangkat Database editor atau Pemrograman berbasis mobile.

1) Menggunakan Database Editor

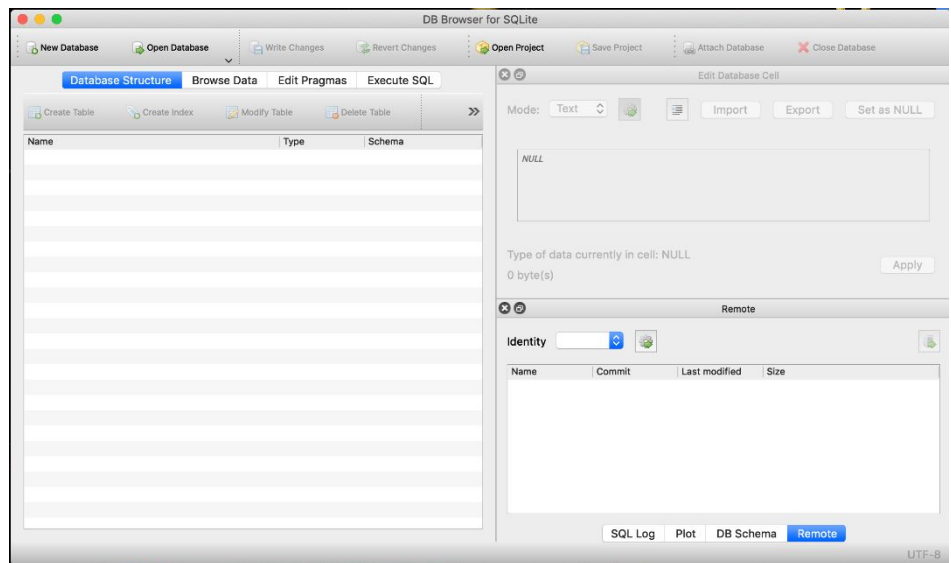
Langkah pembuatan database dengan menggunakan database editor:

- Langkah 1 Unduh DBBrowserForSQLite

Unduh perangkat lunak dan pilih sesuai dengan sistem operasi yang sesuai di <https://sqlitebrowser.org/dl/>

- Langkah 2 Instalasi DBBrowserForSQLite

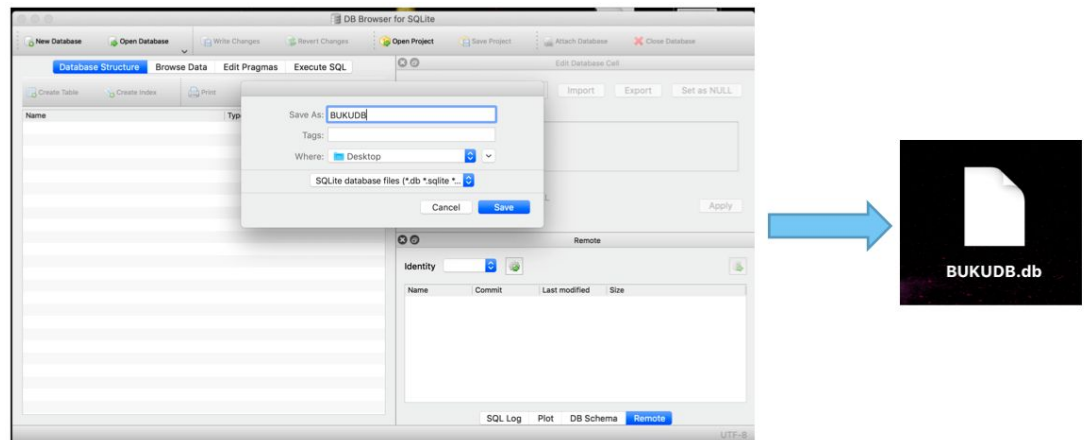
Install perangkat lunak setelah selesai diunduh. Kemudian jalankan aplikasinya.



- Langkah 3

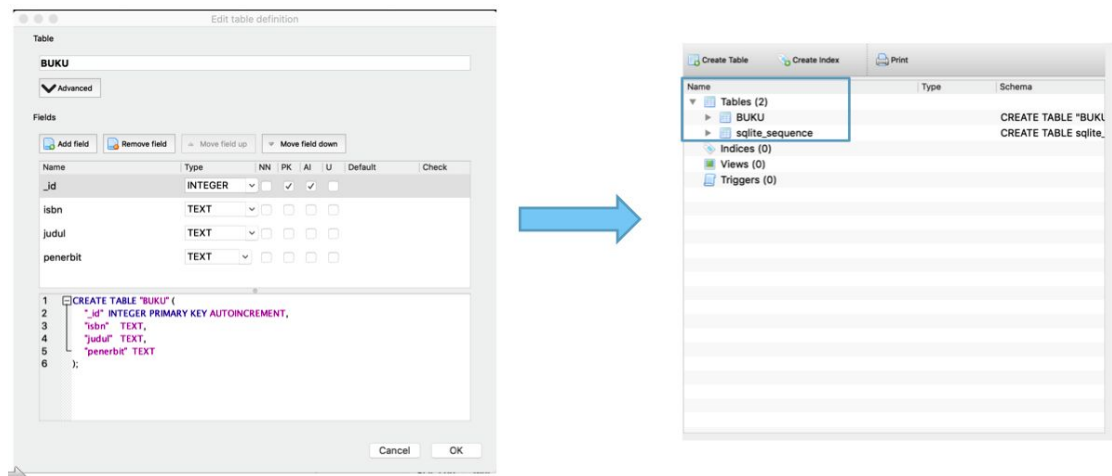
Membuat Database (jika belum ada)

- a) Pilih dan tekan tombol "New Database" di pojok kiri atas.
- b) Kemudian beri nama database sesuai kebutuhan.
- c) Contoh untuk materi ini adalah "BUKUDB".
- d) Pilih tempat penyimpanan database tersebut, misal Desktop
- e) File akan terbentuk di tempat penyimpanan yang ditentukan

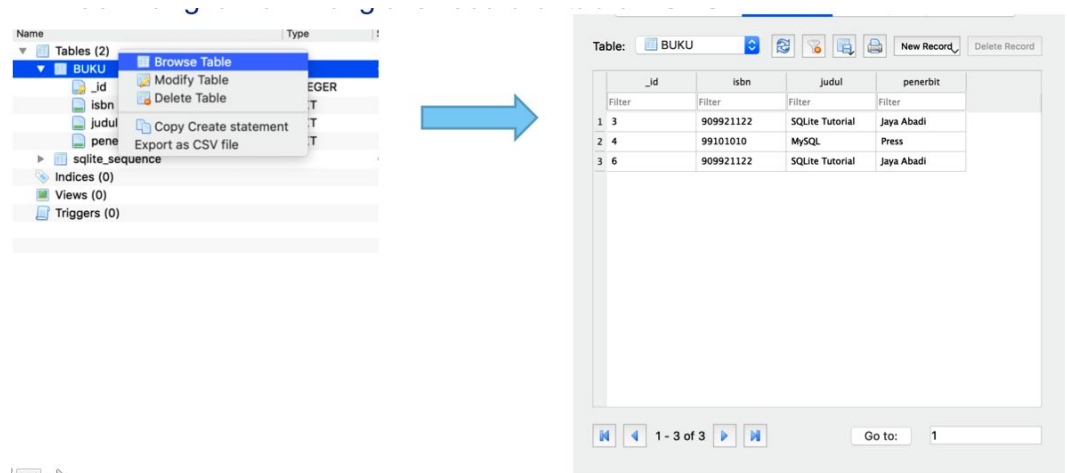


Membuka Database (jika sudah ada)

- a) Pilih dan tekan tombol “Open Database” di pojok kiri atas.
 - b) Kemudian pilih file database. Misal “BUKUDB”.
 - c) Klik Open
- Langkah 4 Membuat Tabel
 - a) Setelah database terbentuk, maka kita bisa membuat table di database tersebut.
 - b) Pilih tombol “Create Table” di sebelah kiri atas. Akan muncul form untuk memasukkan informasi table yang akan dibuat.
 - c) Beri nama table. Di materi ini akan dibuat contoh table BUKU untuk menyimpan data buku.
 - d) Kemudian tambahkan field sesuai kebutuhan. Untuk contoh table BUKU, field yang dibutuhkan :
 - `_id`, tipe int autoincrement
 - `title`, tipe text
 - `isbn`, tipe text
 - `publisher`, tipe text
 - e) Pada kolom bawah akan tampak definisi dari SQL yang akan dieksekusi
 - f) Tekan tombol OK untuk membuat table

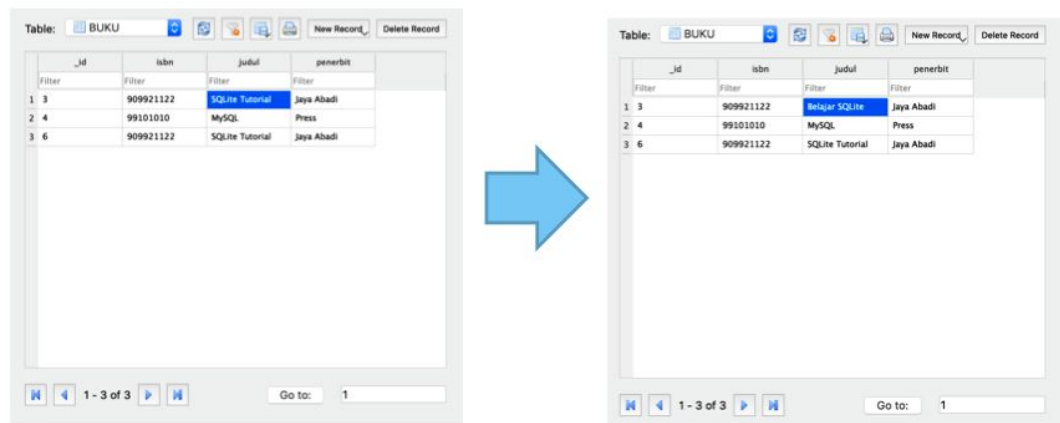


- Langkah 5 Melihat dan menambah data di Table
 - a) Setelah tabel terbentuk, maka kita bisa melihat dan menambahkan record ke dalam table.
 - b) Pilih table yang diinginkan, dalam contoh ini adalah “BUKU”, yang ada di daftar tabel
 - c) Kemudian klik kanan pada nama table tersebut. Pilih “Browse Table”.
 - d) Tampak data yang tersimpan dalam table tersebut.
 - e) Masing – masing kolom bisa langsung diisi nilai sesuai kebutuhan. Contoh untuk materi ini :
 - Kolom _id : dikosongi (karena diisi otomatis oleh sistem)
 - Kolom isbn : 909921129
 - Kolom title : Belajar SQLite
 - Kolom publisher : Jaya Abadi
 - f) Kemudian tekan tombol “New Record” di bagian atas.
 - g) Langkah diatas bisa dilakukan berulang untuk menambah data yang berbeda
 - h) Isi record pada tabel BUKU

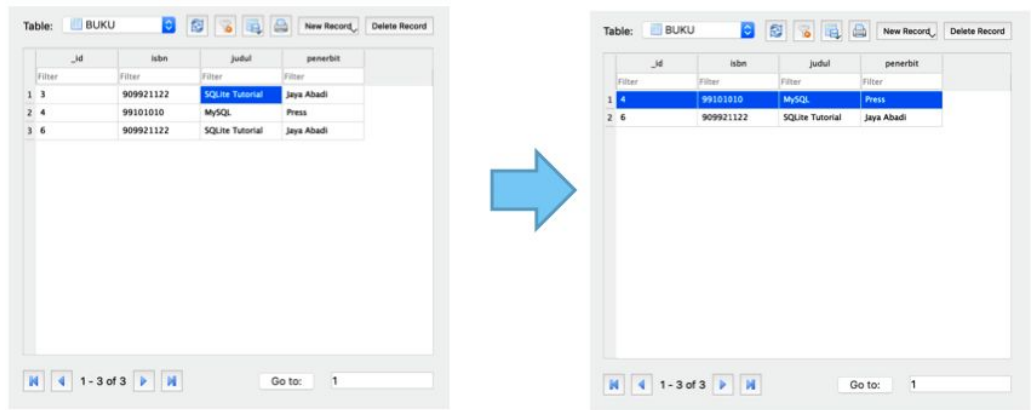


Tips: Jika data baru yang dimasukkan pada table tidak Nampak pada saat berpindah – pindah table, lakukan : Close Database -> Open Database -> pilih file database. Hal ini disebabkan karena ada kelemahan di perangkat lunak.

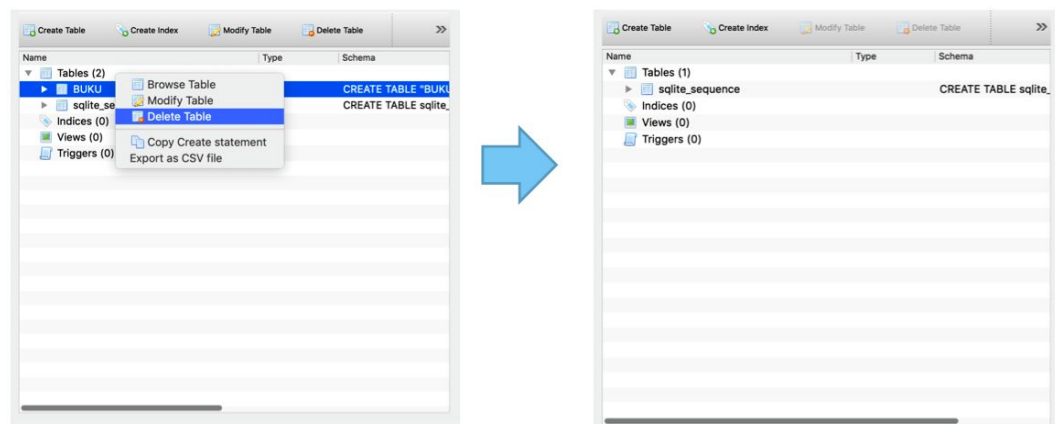
- Langkah 6 (Optional) menghapus data di Tabel
 - a) Untuk mengupdate informasi data pada table dapat langsung dilakukan pada baris yang dimaksud.
 - b) Setelah merubah pastikan tekan tombol “Write Changes” di bagian atas.



- Langkah 7 (Optional) menghapus Tabel
 - a) Untuk menghapus informasi record pada table dapat langsung dilakukan pada baris yang dimaksud. Kemudian tekan tombol “Delete Record”
 - b) Setelah merubah pastikan tekan tombol “Write Changes” di bagian atas.

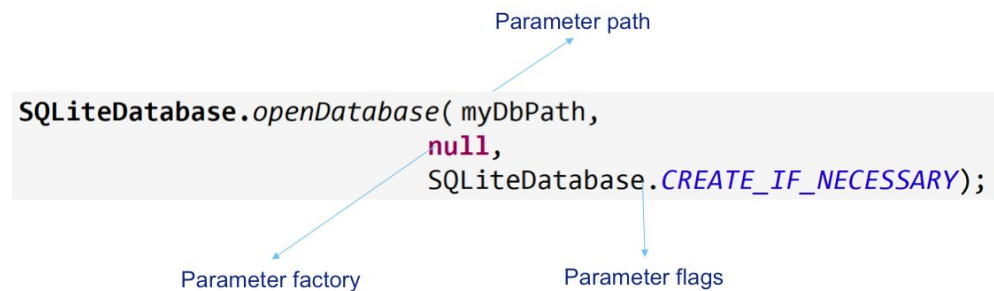


- Langkah 8 (pilihan)
 - c) Untuk menghapus Tabel, dapat dilakukan dengan memilih table yang dimaksud pada daftar table. Kemudian klik kanan pilih “Delete Table”.
 - d) Jika yakin tekan tombol “Yes” pada kotak dialog
 - e) Setelah merubah pastikan tekan tombol “Write Changes” di bagian atas



2) Menggunakan Pemrograman Berbasis Mobile

Di pemrograman Android, untuk membuka atau membuat database SQLite menggunakan fungsi berikut :



Jika database tidak ada, maka akan dibuat baru. Jika ada, maka akan dibuka database yang ada, dengan flag berikut : OPEN_READWRITE, OPEN_READONLY, CREATE_IF_NECESSARY .

Parameters

path : file database yang akan dibuka atau dibuat

factory : an optional factory class that is called to instantiate a cursor when query is called, or null for default

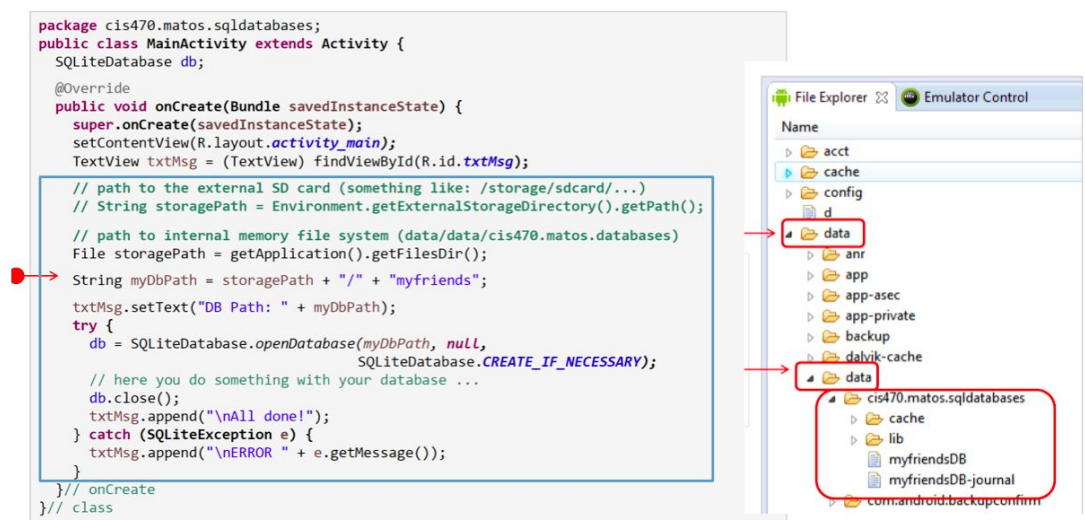
flags : to control database access mode

Returns the newly opened database

Throws SQLException if the database cannot be opened

Databases

Contoh pembuatan database program, dimana database akan dibuat dan disimpan pada internal memory file sistem:



Alternatif lain untuk membuat / membuka SQLite dengan menggunakan fungsi berikut :

```
SQLiteDatabase db = this.openOrCreateDatabase(
    "myfriendsDB",
    MODE_PRIVATE,
    null);
```

The diagram shows the code snippet above with three blue arrows pointing to specific parts: one from 'Parameter nama db' to the string 'myfriendsDB', one from 'Parameter factory' to the 'this' keyword, and one from 'Parameter mode' to the 'MODE_PRIVATE' constant.

MODE bernilai : MODE_PRIVATE, MODE_WORLD_READABLE, dan MODE_WORLD_WRITEABLE

Parameter **factory** diabaikan jika tidak digunakan (diisi null)

3) Structured Query Language di SQLite

Operasi yang paling umum untuk mengelola database adalah **Action queries** dan **Retrieval queries**.

- A retrieval query biasanya berupa perintah SQL-Select dimana hasil balik berupa table yang menyimpan sejumlah kolom dan baris sebagai hasil dari query.
- An action query biasanya digunakan untuk pekerjaan maintenance dan administrative, seperti manipulasi table, user dan lingkungan

Membuat dan mengisi table menggunakan action queries

Fungsi execSQL()

Contoh: membuat table yang memiliki 3 field, recID, name dan phone. Kemudian setelah table dibuat, lakukan operasi insert ke table sebanyak 3 data

```

db.execSQL("create table tblAMIGO ("
    + " recID integer PRIMARY KEY autoincrement, "
    + " name text, "
    + " phone text ); " );

db.execSQL( "insert into tblAMIGO(name, phone) values ('AAA', '555-1111');" );
db.execSQL( "insert into tblAMIGO(name, phone) values ('BBB', '555-2222');" );
db.execSQL( "insert into tblAMIGO(name, phone) values ('CCC', '555-3333');" );

```

Untuk menjalankan query SELECT, fungsi `rawQuery()` bisa digunakan dengan beberapa pendekatan :

- Tanpa parameter

Untuk menjalankan query SELECT dengan menggunakan fungsi `rawQuery()` tanpa parameter.

```

Cursor c1 = db.rawQuery("select * from tblAMIGO", null);

```

Parameter SQL Statement

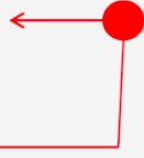


- ✓ Pada contoh `rawQuery` diatas berisi pernyataan Select yang akan mengambil semua baris dan kolom dari table `tblAMIGO`. Hasil query ditampung oleh obyek Cursor `c1`
- ✓ Cursor `c1` akan digunakan untuk mengiterasi seluruh baris hasil query
- ✓ Metode untuk mengiterasi baris per baris hasil query melalui obyek `c1` dibahas berikutnya.

- Menggunakan parameter

Untuk menjalankan query SELECT dengan menggunakan fungsi `rawQuery()` dengan parameter

```
String mySQL = "select count(*) as Total "  
              + " from tblAmigo "  
              + " where recID > ? "  
              + "    and name  = ? ";  
  
String[] args = {"1", "BBB"};  
  
Cursor c1 = db.rawQuery(mySQL, args);
```



Simbol “?” di pernyataan SQL merupakan penanda pengganti. Ketika fungsi dipanggil, maka sistem otomatis akan mengganti symbol dengan nilai argument.

SQL Cursor

Cursor digunakan untuk melakukan akses sekuensial dan acak pada table yang dihasilkan dari pernyataan SQL Select.

Cursor hanya mampu beroperasi pada satu baris pada satu waktu di table. Cursor memiliki beberapa tipe operator seperti:

- ✓ Pengenali Posisi: `isFirst()`, `isLast()`, `isBeforeFirst()`, `isAfterLast()`.
- ✓ Navigasi Record: `moveToFirst()`, `moveToLast()`, `moveToNext()`, `moveToPrevious()`, `move(n)`.
- ✓ Pengambilan Informasi Field: `getInt`, `getString`, `getFloat`, `getBlob`, `getDouble`, etc.
- ✓ Inspeksi Skema: `getColumnName()`, `getColumnNames()`, `getColumnIndex()`, `getColumnCount()`, `getCount()`.

Untuk mengiterasi hasil query obyek Cursor dapat dilakukan dengan contoh berikut

```

1 → String sql = "select * from tblAmigo";
    Cursor c1 = db.rawQuery(sql, null);

2 → c1.moveToPosition(-1);

    while ( c1.moveToNext() ){

        int recId = c1.getInt(0);
        String name = c1.getString(1);
        String phone = c1.getString(c1.getColumnIndex("phone"));

        // do something with the record here...

    }

```

Persiapkan rawQuery dengan melemparkan pernyataan SQL Select di parameteranya.

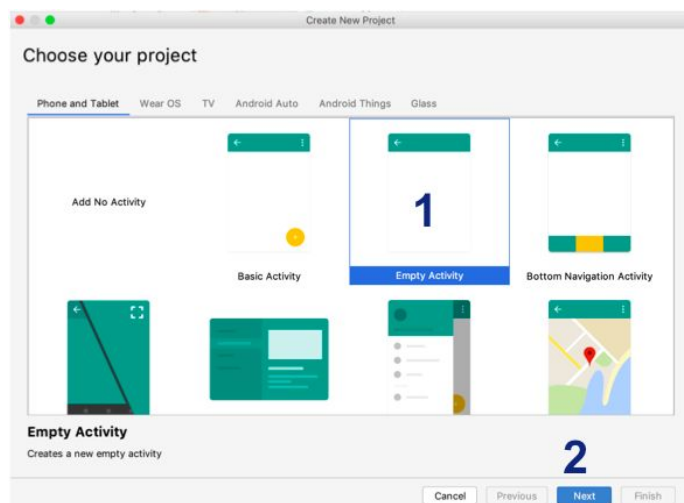
Pindahkan posisi penanda ke posisi baris pertama file. Nilai valid antara $-1 \leq \text{posisi} \leq \text{jumlahbaris}$

Kemudian gunakan fungsi moveToNext untuk berpindah ke baris berikutnya

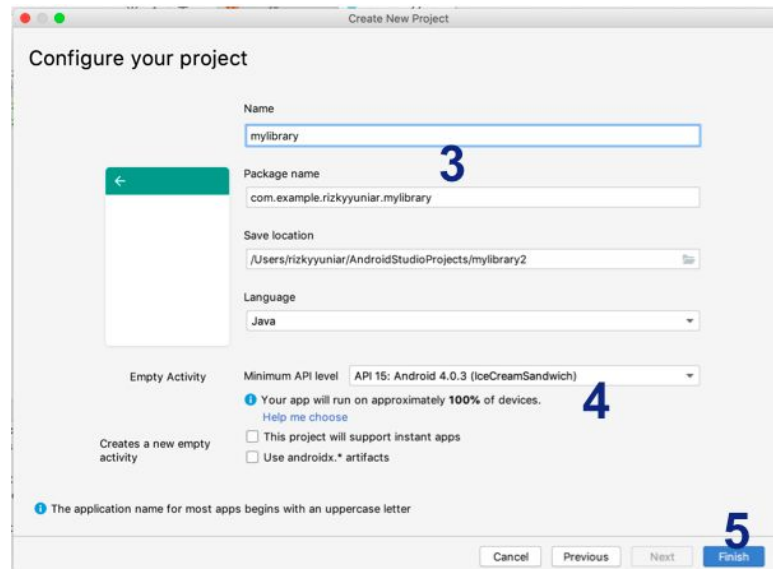
Mendesain database mobile dengan model layer

Secara umum langkah pembuatan database dengan database editor:

- Langkah 1 Buat project Android baru
 - 1) Pilih empty project
 - 2) Tekan Nex

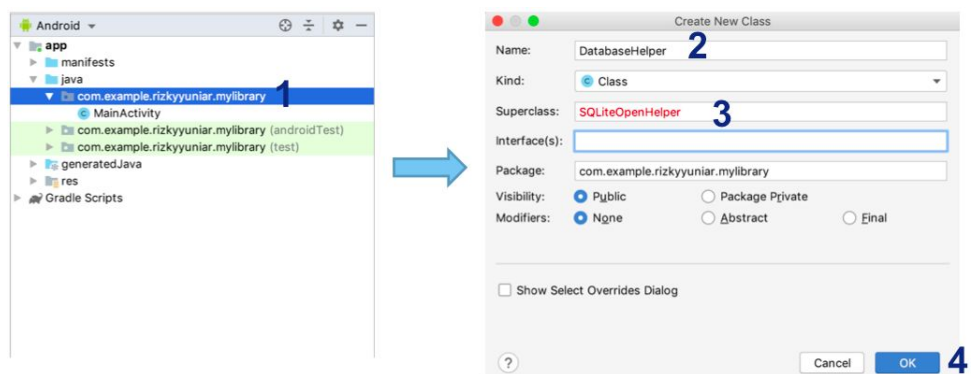


- 3) Beri nama project dan tempat penyimpanan project
- 4) Pastikan minimum API level
- 5) Tekan Finish untuk membuat project baru



- Langkah 2 Buat kelas DatabaseHelper

- 1) Klik kanan pada package, pilih New -> Java Class
- 2) Beri nama kelas DatabaseHelper
- 3) Beri nama superclass SQLiteOpenHelper (case sensitive)
- 4) Tekan OK untuk membuat kelas



- 5) Buka kelas DatabaseHelper kemudian modifikasi program sesuai dengan contoh program yang diberikan.

Kelas DatabaseHelper digunakan untuk membungkus kemampuan membuat, membuka dan memanipulasi database SQLite.

- 6) Tambahkan library – library yang dibutuhkan

```
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.util.Log;
import java.util.ArrayList;
```

7) Buat beberapa variable yang dibutuhkan kelas untuk definisi dari database SQLite

- o Nama Tabel : students
- o Field table : id, firstname
- o Nama database : students_database

```
public static String DATABASE_NAME = "student_database";
private static final int DATABASE_VERSION = 1;
private static final String TABLE_STUDENTS = "students";
private static final String KEY_ID = "id";
private static final String KEY_FIRSTNAME = "name";

private static final String CREATE_TABLE_STUDENTS = "CREATE TABLE "
+ TABLE_STUDENTS + "(" + KEY_ID
+ " INTEGER PRIMARY KEY AUTOINCREMENT," + KEY_FIRSTNAME + " TEXT );";
```

8) modifikasi program sesuai dengan contoh program yang diberikan. Buat constructor kelas

```
public DatabaseHelper(Context context) {
    super(context, DATABASE_NAME, factory: null, DATABASE_VERSION);

    Log.d( tag: "table", CREATE_TABLE_STUDENTS);
}
```

9) Tambahkan fungsi override onCreate dan onUpgrade

```
@Override
public void onCreate(SQLiteDatabase db) {
    db.execSQL(CREATE_TABLE_STUDENTS);
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("DROP TABLE IF EXISTS '" + TABLE_STUDENTS + "'");
    onCreate(db);
}
```

10) Buat fungsi addStudentDetail

```

public long addStudentDetail(String student) {
    SQLiteDatabase db = this.getWritableDatabase();
    // Creating content values
    ContentValues values = new ContentValues();
    values.put(KEY_FIRSTNAME, student);
    // insert row in students table
    long insert = db.insert(TABLE_STUDENTS,
        nullColumnHack: null, values);

    return insert;
}

```

11) Buat fungsi getAllStudentList

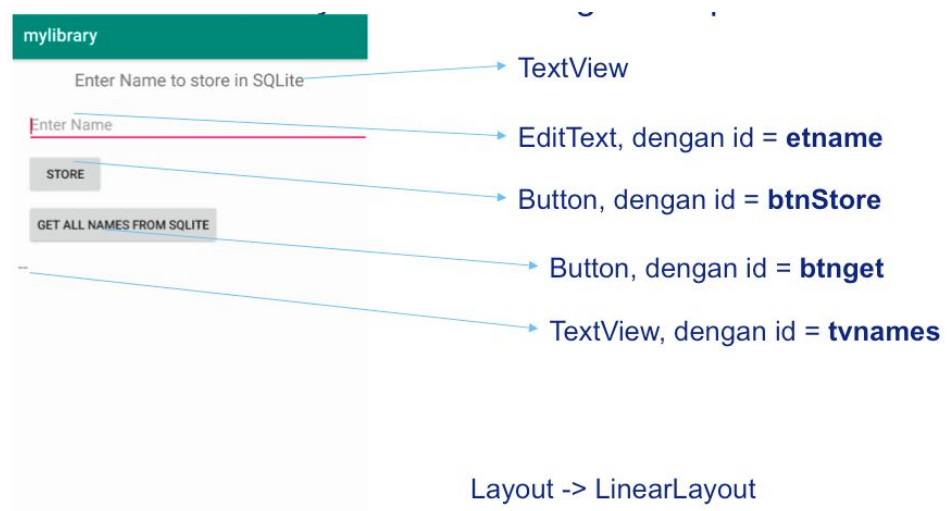
```

public ArrayList<String> getAllStudentsList() {
    ArrayList<String> studentsArrayList = new ArrayList<String>();
    String name="";
    String selectQuery = "SELECT * FROM " + TABLE_STUDENTS;
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor c = db.rawQuery(selectQuery, selectionArgs: null);
    // looping through all rows and adding to list
    if (c.moveToFirst()) {
        do {
            name = c.getString(c.getColumnIndex(KEY_FIRSTNAME));
            // adding to Students list
            studentsArrayList.add(name);
        } while (c.moveToNext());
        Log.d( tag: "array", studentsArrayList.toString());
    }
    return studentsArrayList;
}

```

- Langkah 3 Buat antarmuka layer input

Modifikasi layout sesuai dengan tampilan berikut



Source activity_main.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context=".MainActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textColor="#fff"
        android:textSize="20sp"
        android:layout_marginTop="10dp"
        android:text="Enter Name to store in SQLite" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="40dp"
        android:id="@+id/etname"
        android:background="#fff"
        android:layout_marginTop="20dp"
        android:layout_marginLeft="20dp"
        android:hint="Enter Name" />

    <Button
        android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
android:id="@+id/btnstore"
android:layout_marginLeft="20dp"
android:layout_marginTop="10dp"
android:text="Store"/>
```

<Button

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/btnget"
android:layout_marginLeft="20dp"
android:layout_marginTop="10dp"
android:text="Get All Names from SQLite" />
```

<TextView

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/tvnames"
android:layout_marginLeft="10dp"
android:textColor="#fff"
android:textSize="20sp"
android:layout_marginTop="10dp"
android:text="Enter Name to store in SQLite" />
```

</LinearLayout>

- Langkah 4 Modifikasi kelas MainActivity untuk koneksi ke antarmuka dan database

Buka kelas MainActivity kemudian modifikasi beberapa baris program.

1. Import beberapa library yang dibutuhkan
2. Tambahkan beberapa variable untuk menampung beberapa view komponen

Full source :

```
package com.example.rizkyyuniar.mylibrary;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    private Button btnStore, btnGetall;
    private EditText etname;
    private DatabaseHelper databaseHelper;
    private TextView tvnames;
    private ArrayList<String> arrayList;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        databaseHelper = new DatabaseHelper(this);
        tvnames = (TextView) findViewById(R.id.tvnames);

        btnStore = (Button) findViewById(R.id.btnstore);
        btnGetall = (Button) findViewById(R.id.btnget);
        etname = (EditText) findViewById(R.id.etname);
```

```

btnStore.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

databaseHelper.addStudentDetail(etname.getText().toString());
        etname.setText("");
        Toast.makeText(MainActivity.this,
            "Stored Successfully!",
            Toast.LENGTH_SHORT).show();
    }
});

btnGetall.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        arrayList = databaseHelper.getAllStudentsList();
        tvnames.setText("");
        for (int i = 0; i < arrayList.size(); i++){
            tvnames.setText(tvnames.getText().toString()+"",
            "+arrayList.get(i));
        }
    }
});
}
}

```

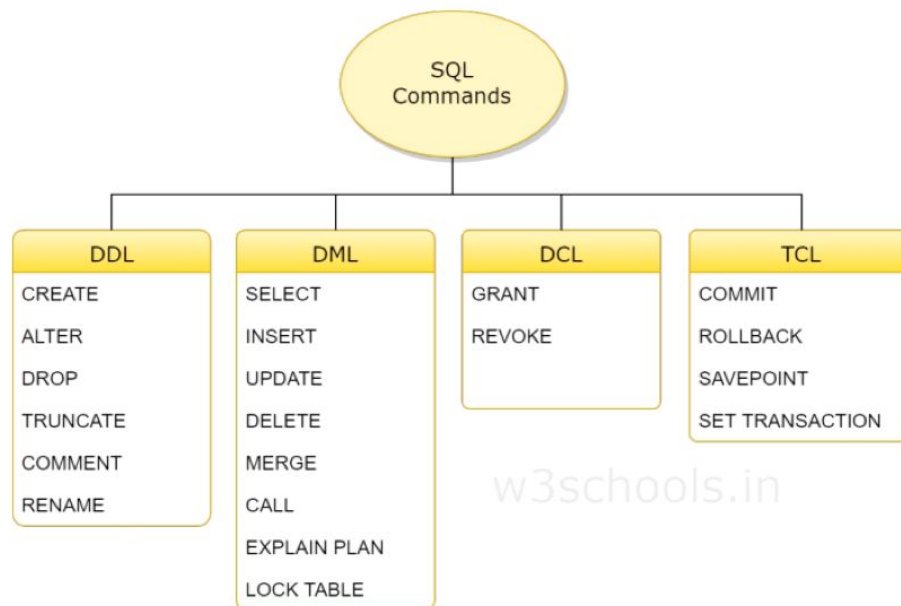
- Langkah 5 Jalankan program

Jika berhasil, program akan menampilkan input, tombol store dan tombol get all names.

- o Masukkan nama
- o Tekan store untuk menyimpan
- o Tekan untuk melihat data yang telah tersimpan
- o Nama yang disimpan akan terlihat

Pengenalan SQL

Structured Query Language (SQL) adalah bahasa standar untuk menyimpan, memanipulasi dan menerima data di basis data. Memiliki 4 kategori fungsi Data Definition Language (DDL), Data Manipulation Language (DML), Data Control Language (DCL) dan Transaction Control Language (TCL)



1) Data Definition Language (DDL)

DDL berhubungan dengan database schemas dan deskripsi, bagaimana data disimpan dalam database.

- o CREATE – membuat database dan object (table, index, views, store procedure, function, and triggers)

o

sintaks dasar untuk membuat table di basis data

```
CREATE TABLE table_name
(
    column_name column_type,
    [...]
);
```


contoh :

```
CREATE TABLE "BUKU" (  
  "_id" INTEGER PRIMARY KEY AUTOINCREMENT,  
  "isbn" TEXT,  
  "title" TEXT,  
  "publisher" TEXT  
);
```

Tipe column:

- ✓ NULL : A NULL is considered its own distinct type. A NULL type does not hold a value. Literal NULLs are represented by the keyword NULL.
- ✓ Integer
- ✓ Float
- ✓ Text
- ✓ BLOB : A length of raw bytes, copied exactly as provided

- o ALTER – mengubah struktur dari database
sintaks dasar untuk mengubah table di basis data :

```
ALTER TABLE database_name.table_name RENAME TO new_table_name;  
ALTER TABLE database_name.table_name ADD COLUMN column_def...;
```

- o DROP – menghapus obyek dari database

Sintaks dasar:

```
DROP TABLE table_name;
```

2) Data Manipulation Language (DML)

DML berhubungan dengan manipulasi data, SQL statements yang sering digunakan SELECT, INSERT, UPDATE, DELETE, dan sebagainya.

- o SELECT – retrieve data from a database

sintaks umum perintah SELECT di basis data SQLite

```
SELECT [DISTINCT] select_heading
FROM source_tables
WHERE filter_expression
GROUP BY grouping_expressions
HAVING filter_expression
ORDER BY ordering_expressions
LIMIT count
OFFSET count
```

- FROM *source_tables* : Designates one or more source tables and combines them together into one large working table
- WHERE *filter_expression* : Filters specific rows out of the working table.
- GROUP BY *grouping_expressions* : Groups sets of rows in the working table based off similar values.
- SELECT *select_heading* : Defines the result set columns and (if applicable) grouping aggregates.
- HAVING *filter_expression* : Filters specific rows out of the grouped table. Requires a GROUP BY.
- DISTINCT : Eliminates duplicate rows.
- ORDER BY *ordering_expressions* : Sorts the rows of the result set.
- OFFSET *count* : Skips over rows at the beginning of the result set. Requires a LIMIT.
- LIMIT *count* : Limits the result set output to a specific number of rows.

o INSERT – insert data into a table

adalah sintaks dasar untuk menyisipkan baris pada table di basis data

```
INSERT INTO table_name (column_name [, ...]) VALUES (new_value [, ...]);
```

```
INSERT INTO table_name VALUES (new_value [, ...]);
```

contoh:

```
INSERT INTO parts ( name, stock, status ) VALUES ( 'Widget', 17, 'IN STOCK' );
```

- o UPDATE – updates existing data within a table

```
UPDATE table_name SET column_name=new_value [, ...] WHERE expression
```

· Contoh Update SQL :

```
-- Update the price and stock of part_id 454:  
UPDATE parts SET price = 4.25, stock = 75 WHERE part_id = 454;
```

- o DELETE – Delete all records from a database table

```
DELETE FROM table_name WHERE expression;
```

Contoh Delete SQL :

```
-- Delete the row with rowid 385:  
DELETE FROM parts WHERE part_id = 385;  
  
-- Delete all rows with a rowid greater than or equal to 43  
-- and less than or equal to 246:  
DELETE FROM parts WHERE part_id >= 43 AND part_id <= 246;
```

3) Data Control Language (DCL)

Perintah dalam DCL umumnya berhubungan dengan hak,ijin dan pengawasan dalam database .

- o GRANT – allow users access privileges to the database
- o REVOKE – withdraw users access privileges given by using the GRANT command

4) Transaction Control Language (TCL)

TCL transaksi dalam sebuah database.

- o COMMIT – commits a Transaction
- o ROLLBACK – rollback a transaction in case of any error occurs
- o SAVEPOINT – to rollback the transaction making points within groups
- o SET TRANSACTION – specify characteristics of the transaction

B. Keterampilan yang diperlukan dalam merancang database dan data persistence pada mobile data

1. Membaca, meneliti, menggali dan mengolah data informasi
2. Berkomunikasi dengan para pihak

A. Sikap yang diperlukan dalam merancang database dan data persistence pada mobile data

1. Ketelitian
2. Kesabaran
3. Sopan

Tugas Dan Proyek Pelatihan
1. Kuis 8 dan 10 Mobile Programming

Link Referensi Modul Pertama
1. Video Pembelajaran 2. E-book 3. Link Youtube/Website rujukan

Link Pertanyaan Modul Pertama
https://app.sli.do/ (bisa menggunakan aplikasi ini)

Bahan Tayang
Bisa berupa Link/ Screen Capture Slide pelatihan

Link room Pelatihan dan Jadwal live sesi bersama instruktur
Zoom, Blue Jeans, Meets

Penilaian
Komposisi penilaian Kuis 8 Mobile Programming: Nilai 10 (Range 0 -10) Komposisi penilaian Kuis 10 Mobile Programming: Nilai 10 (Range 0 -10)

Target Penyelesaian Modul Pertama
1hari/sampai 6JP

REFERENSI

1. Dhingra, M. et al, Wireless Network Security Threats and Their Solutions: a short study, International Journal of Smart Sensors and Ad Hoc Networks (IJSSAN), ISSN No. 2248-9738 (Print), Vol-2, Iss-1,2, 2012
2. Efraim Turban, D. K.-P. (2012). Electronic commerce (Seventh Edition ed.). Pearson.
3. Gottapu Sasibhushana Rao, 2013. Mobile Cellular Communication (First Edition). Pearson.
4. [Guowang Miao](#), Jens Zander, Ki Won Sung, and Ben Slimane, Fundamentals of Mobile Data Networks, Cambridge University Press, [ISBN 1107143217](#), 2016.
5. Herbert Schildt, *Java2 : A beginner's Guide*, Second Edition, McGraw-Hill/Osborne
6. Larsson, J. and Waller, Ida. Security in wireless networks: Vulnerabilities and countermeasures, Department of Software Engineering and Computer Science Blekinge Institute of Technology, Springer, 2003
7. Matthew Mathias, Swift Programming, 2nd edition, Big Nerd Ranch
8. <https://developer.android.com/reference/android/os/Environment.html>
9. <https://developer.android.com/reference/java/io/File.html>
10. <https://developer.android.com/guide/topics/data/data-storage?hl=id>
11. <https://developer.android.com/training/data-storage/files#InternalVsExternalStorage>
12. Owens, Michael (2006). The Definitive Guide to SQLite.
<https://www.sqlite.org/chronology.html>
13. <https://developer.android.com>
14. <https://www.quora.com/How-does-the-algorithm-of-Google-Maps-work>
15. <https://tips-droid.blogspot.com/2013/07/mengenal-android-sensor.html>
16. <https://mainthebest.com/smartphones/sensor-smartphone-android/#accelerometer>

17. PROFESSIONAL Android™ Sensor Programming Greg Milette Adam Strou
18. Irwanto, A. (2013, September 7). area teknik. Retrieved Oktober 25, 2017, from <http://area-teknik.blogspot.co.id/2013/09/perbedaan-teknologi-1g-2g-25g-3g-35g-4g.html>
19. Wulan. (2015, November 2015). Srydari. Retrieved Oktober 25, 2017, from <http://srydari.blogspot.co.id/2015/11/perkembangan-teknologi-dari-1g-sampai-5g.html>
20. “ “, Java™ Programming Language, Oracle America
21. <https://developer.apple.com/library/archive/referencelibrary/GettingStarted/DevelopiOSAppsSwift/index.html/>
22. <https://developer.android.com/topic/libraries/architecture>
23. <https://www.oracle.com/technetwork/java/javase/overview/index.html>
24. Android Cook Book, McGraw-Hill/Osborne, 2013



DIGITAL
TALENT
SCHOLARSHIP



digitalent.kominfo.go.id

 [DTS_kominfo](#)

 [digitalent.kominfo](#)

 [digitalent.kominfo](#)

   [Digital Talent Scholarship](#)



Badan Penelitian dan Pengembangan Sumber Daya Manusia
Kementerian Komunikasi dan Informatika
Republik Indonesia