

Mata Kuliah	:	IF330324-IF430324 Pembangunan Aplikasi Mobile
Semester	:	2 (Genap)
Topik	:	Pengenalan <i>Content Provider & Broadcast Receiver</i>
Tanggal Sesi	:	21 Februari 2017
Minggu ke-/sesi	:	3/3
Waktu pengerjaan	:	2x50 menit
Aktifitas	:	Mahasiswa mengimplementasikan <i>Content Provider & Broadcast Receiver</i> pada Aplikasi Android
Setoran	:	<i>Folder</i> proyek hasil pengerjaan Bagian II yang telah dikompres menjadi .rar. Nama file dengan format: pam_w03s03_ NIM .rar. NIM diganti dengan NIM Anda.
Batas penyerahan	:	25 Februari 2017, sebelum 21.30 WIB via <i>e-course</i>
Tujuan praktikum	:	Mahasiswa mampu memahami dan menggunakan <i>content provider & Broadcast Receiver</i> pada Android
Sarana	:	Android Studio, Genymotion
Jenis kegiatan	:	Praktikum

Mulai minggu ke-2 sesi 2, mahasiswa akan melakukan praktikum PAM yang dibagi ke dalam dua bagian, yaitu:

1. Mengikuti langkah-langkah praktikum untuk mengulas atau memperkuat kemampuan teknis mahasiswa yang didasarkan pada materi yang diajarkan pada kelas teori atau materi yang menjadi pelengkap atau tambahan dari materi yang telah disampaikan pada kelas teori.
2. Mengerjakan bagian *deliverables* pada dokumen praktikum untuk kemudian disetor via Moodle. Tugas yang diberikan tersebut harus dikerjakan dan disetor sesuai dengan waktu deadline yang diberikan. Perbedaan bagian ini dengan praktikum yang telah dilakukan sebelumnya adalah mahasiswa diharapkan akan menghasilkan sebuah aplikasi yang utuh yang akan dikumpulkan pada minggu ke-14 masa perkuliahan PAM. Deskripsi aplikasi yang akan dibangun dijelaskan pada bagian *deliverables*. Mahasiswa diharapkan akan membangun komponen atau bagian-bagian dari aplikasi tersebut secara bertahap setiap minggu, sesuai dengan topik praktikum yang sedang diberikan pada setiap sesi.

BAGIAN I: LANGKAH-LANGKAH PRAKTIKUM

Content Provider digunakan untuk mengolah dan berbagi data. Data dapat disimpan dalam *file* sistem, *database* SQLite, atau dengan cara lain yang pada prinsipnya sama. Dengan adanya *Content Provider* memungkinkan antar aplikasi untuk saling berbagi data. Komponen ini sangat berguna ketika sebuah aplikasi membutuhkan data dari aplikasi lain. Praktikum kali ini Anda akan menerapkan *content provider* pada aplikasi anda untuk mengakses data dari aplikasi *Contacts/People* yang sudah tersedia di perangkat anda. Data yang dapat diakses pada aplikasi *Contacts/People* adalah nama, nomor telepon, alamat *e-mail*, foto, dan informasi pribadi lainnya.

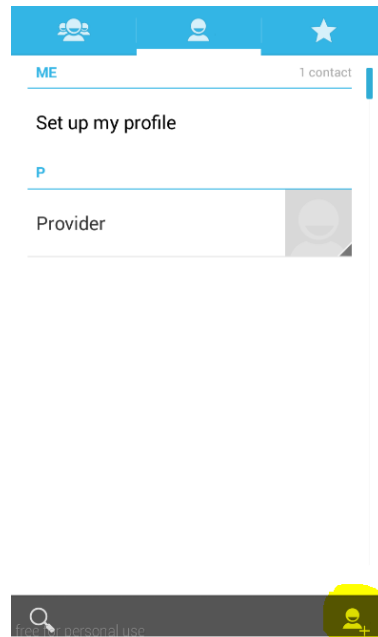
Broadcast Receiver adalah komponen aplikasi Android yang dapat digunakan untuk menanggapi *system-wide broadcast announcements* terkait *event* tertentu yang diberikan oleh sistem Android. Contoh *broadcast* yang dikirimkan oleh sistem operasi Android misalkan notifikasi adanya aplikasi baru ter-*install*/ter-*update*, layar mati, *booting* sistem operasi selesai, ada SMS atau telepon masuk,

kondisi baterai dan sebagainya. *Broadcast Receiver* yang akan anda implementasikan pada praktikum kali ini adalah menampilkan persentase baterai perangkat anda.

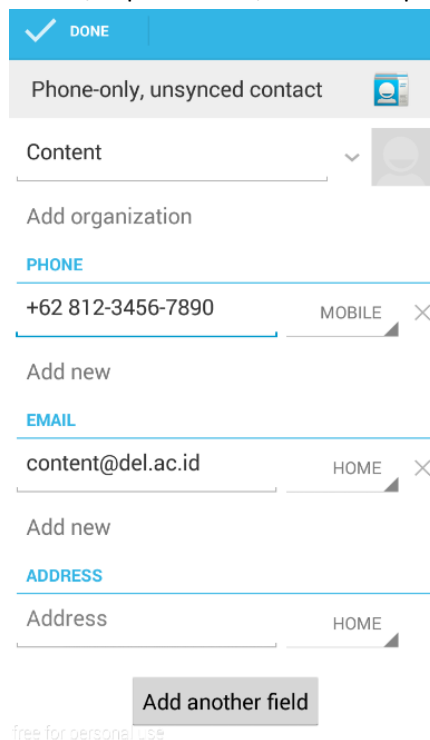
CONTENT PROVIDER

1. Jika anda menggunakan emulator bawaan android atau menggunakan *Genymotion* yang belum mempunyai kontak, maka terlebih dahulu tambahkan kontak dari aplikasi *Contacts/People* yang sudah tersedia.

Bukalah aplikasi *Contacts/People* kemudian pilih menu *Add Contact*.



2. Kemudian isilah data yang diperlukan, seperti nama, nomor telepon, *email*, dll.



3. Setelah selesai pilih DONE. Kemudian tambahkan beberapa kontak lagi.

CONTOH PENGGUNAAN CONTENT PROVIDER

1. Buatlah sebuah project beri nama **ContentProvider_nimlengkap**, pilih *Empty Activity* beri nama **MainActivity**.
2. Pada *file* `activity_main.xml` tambahkan satu *textview* untuk menampilkan informasi kontak.

```
<TextView
    android:id="@+id/contact"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true" />
```

3. Kemudian pada *file* `MainActivity.java` ubah kode menjadi seperti di bawah ini. Pada kode tersebut anda akan mengakses data pada aplikasi *Contact/People* yaitu nama dan nomor telepon dan akan ditampilkan pada *textview* `Contact`.

```
import android.database.Cursor;
import android.net.Uri;
import android.provider.ContactsContract;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        TextView contact = (TextView) findViewById(R.id.contact);
        Cursor cursor = getContacts();
        while (cursor.moveToNext()) {
            /*Get contact information*/
            String contactID =
cursor.getString(cursor.getColumnIndex(ContactsContract.Contacts._ID));
            String displayName =
cursor.getString(cursor.getColumnIndex(ContactsContract.Data.DISPLAY_NAME));
            String displayPhone = "";
            int hasPhone =
cursor.getInt(cursor.getColumnIndex(ContactsContract.Contacts.HAS_PHONE_NUMBER))
;

            /*Check if phone number exist*/
            if (hasPhone == 1) {
                Cursor phoneCursor =
getContentResolver().query(ContactsContract.CommonDataKinds.Phone.CONTENT_URI,
null,
                ContactsContract.CommonDataKinds.Phone.CONTACT_ID + " =
" + contactID, null, null);
                while (phoneCursor.moveToNext()) {
                    Log.v("Content Providers", displayPhone =
phoneCursor.getString(phoneCursor.getColumnIndex(
                ContactsContract.CommonDataKinds.Phone.NUMBER)));
                }
                phoneCursor.close();
            }

            /*Display contact information*/
            contact.append(displayName + "\n" + displayPhone);
        }
    }
}
```

```

        contact.append("\n===== \n");
    }

    private Cursor getContacts() {
        /* Run query */
        Uri uri = ContactsContract.Contacts.CONTENT_URI;
        String[] projection = new String[]{ContactsContract.Contacts._ID,
            ContactsContract.Contacts.DISPLAY_NAME,
            ContactsContract.Contacts.HAS_PHONE_NUMBER};
        String selection = ContactsContract.Contacts.IN_VISIBLE_GROUP + " = '1' +
        ("1") + "'";
        String[] selectionArgs = null;
        String sortOrder = ContactsContract.Contacts.DISPLAY_NAME + " COLLATE
        LOCALIZED ASC";
        return managedQuery(uri, projection, selection, selectionArgs,
        sortOrder);
    }
}

```

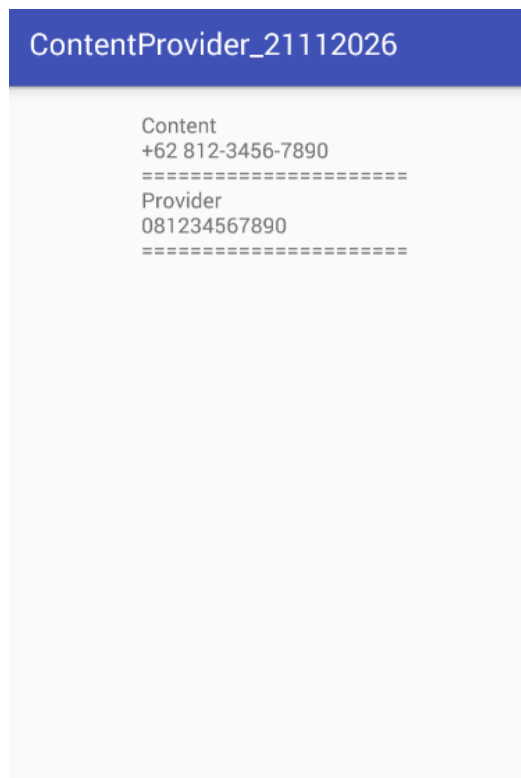
4. Untuk mengakses *content provider* pada *Contacts/People* memerlukan sebuah *permission*. Buka *file* *AndroidManifest.xml* kemudian tambahkan kode berikut :

```

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

```

5. Jalankan aplikasi anda pada emulator kesayangan anda, hasil yang diharapkan seperti gambar berikut.



CONTOH PENGGUNAAN BROADCAST RECEIVER

1. Buat sebuah aplikasi dengan nama `BroadcastReceiver_NimLengkap`, pilih *Empty Activity* beri nama **MainActivity**.
2. Pada `activity_main.xml` buatlah sebuah *textview* dan *progressbar* yang akan memvisualisasikan persentase baterai perangkat anda.

```
<TextView
    android:id="@+id/batteryLevel"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Battery Level : " />

<ProgressBar
    android:id="@+id/progressBar"
    style="?android:attr/progressBarStyleHorizontal"
    android:max="100"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/batteryLevel"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />
```

3. Pada `MainActivity.java` anda akan menambahkan *broadcast receiver* untuk mengakses persentase baterai perangkat anda kemudian menampilkannya pada *textview* dan *progressbar*. Jika baterai perangkat anda dalam keadaan 100% maka akan muncul *pop-up/toast* 'Battery Full.'.

```
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    private TextView batteryLevel;
    private ProgressBar mBatteryLevelProgress;

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

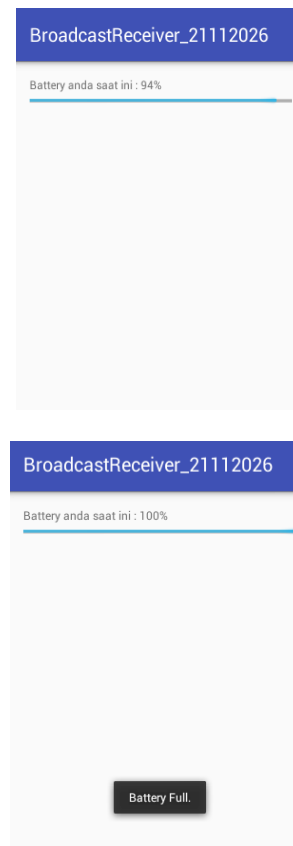
        batteryLevel = (TextView) findViewById(R.id.batteryLevel);
        mBatteryLevelProgress = (ProgressBar) findViewById(R.id.progressBar);

        //Kode untuk meregister event yang akan ditangkap oleh BroadcastReceiver
        //Bisa juga diset di AndroidManifest.xml file
        this.registerReceiver(this.myBatteryReceiver, new
        IntentFilter(Intent.ACTION_BATTERY_CHANGED));
    }

    private BroadcastReceiver myBatteryReceiver = new BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {
            int lvl = intent.getIntExtra("level", 0);
            batteryLevel.setText("Battery anda saat ini : " +
```

```
String.valueOf(lvl) + "%");
    mBatteryLevelProgress.setProgress(lvl);
    if (lvl == 100) {
        Toast.makeText(context, "Battery Full.",
            Toast.LENGTH_LONG).show();
    }
}
};
}
```

4. Hasil yang diharapkan seperti gambar di bawah ini.



Tugas

Pahami lebih baik lagi tentang Content Provider dan Broadcast Receiver pada Android dengan:

1. Eksplorasi website Android Developer menu *Content Providers* serta *Broadcast Receivers* dengan alamat url <http://developer.android.com/guide/index.html>.
2. Eksplorasi contoh berikut ini untuk lebih memahami topik praktikum ini dengan membuka website:
 - a. Android – Content Providers:
http://www.tutorialspoint.com/android/android_content_providers.htm.
 - b. Android – Broadcast Receivers:
http://www.tutorialspoint.com/android/android_broadcast_receivers.htm.

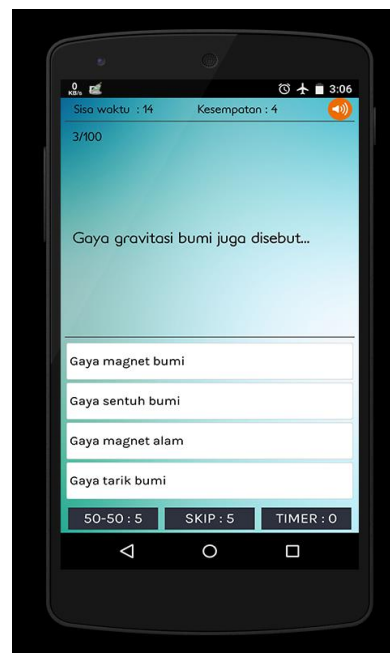
BAGIAN II: DELIVERABLES

Pada minggu ke-14, Anda diharapkan menyelesaikan pembuatan aplikasi **Adu Jago Kuis**. Aplikasi ini Anda bangun dengan menggunakan keterangan aplikasi “Asah Otak Kuis” karya Spirit Mandiri¹ pada Google Play Store sebagai referensi. Tujuan kegiatan ini adalah Anda dapat memahami komponen-komponen dan detail teknis dari sebuah aplikasi populer yang sudah di-publish ke pasar aplikasi *mobile*.

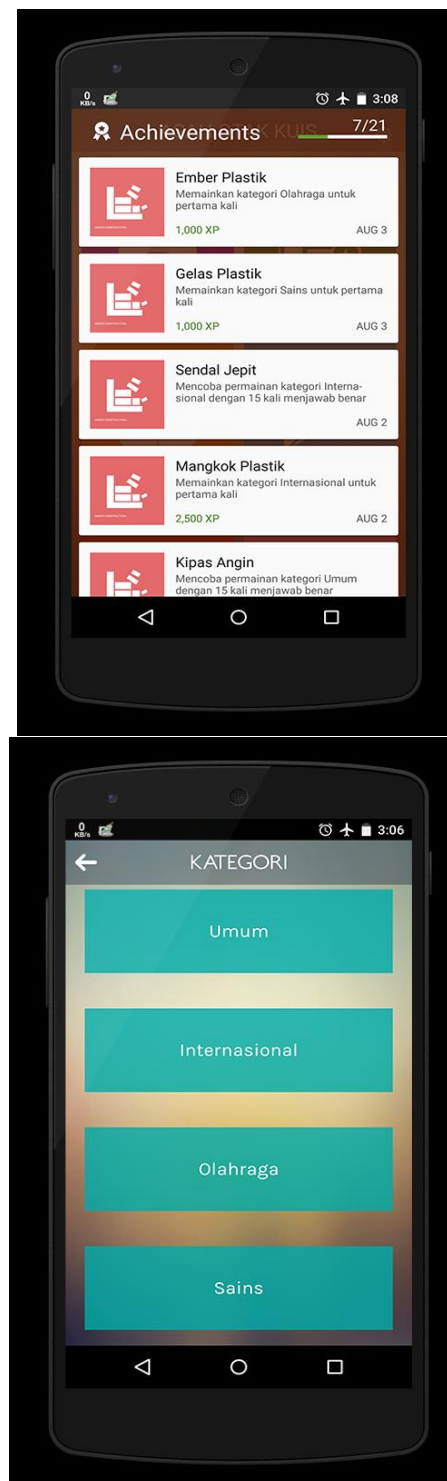
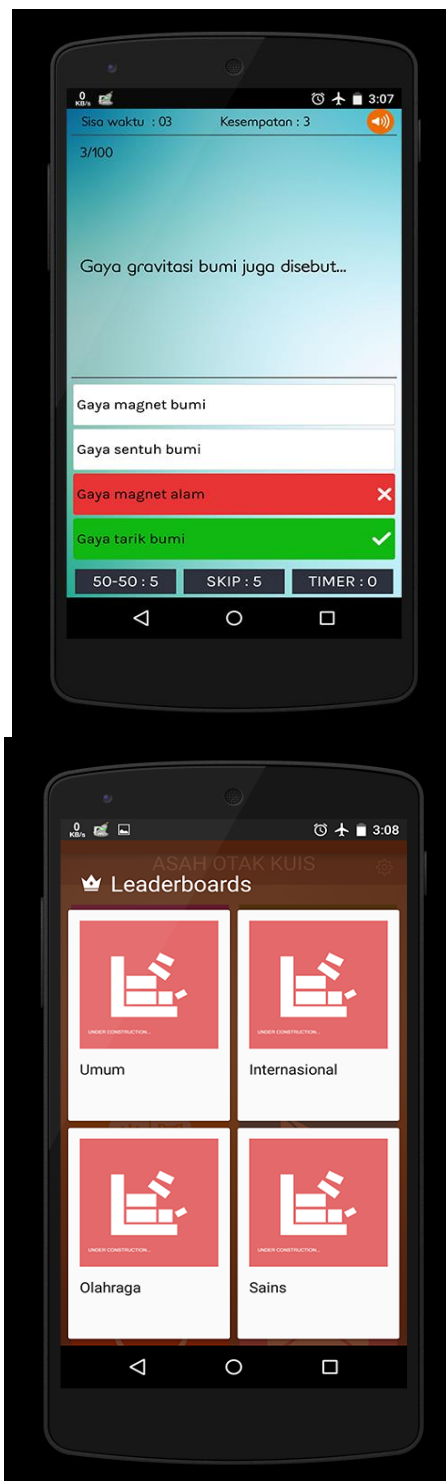
Dengan demikian, Anda juga nantinya dapat membangun aplikasi sejenis dan menjadi potfolio Anda. Sebagai informasi, jumlah *install* aplikasi ini per tanggal 14 Februari 2016 berada di kisaran 500,000 - 1,000,000 *install* (update terakhir aplikasi ini, yaitu versi 6.0.5 adalah 24 Juli 2016).

Deskripsi Umum Aplikasi

Aplikasi **Adu Jago Kuis** ini bertujuan untuk adu kemampuan dalam menjawab kuis bagi para pesertanya. Untuk pembuatan aplikasi ini, Anda dibebaskan untuk menyediakan data soal dan jawaban kuis yang Anda inginkan, termasuk gambar, teks dan *resources* yang lainnya. Kumpulan gambar berikut ini menunjukkan struktur umum aplikasi “Asah Otak Kuis” karya Spirit Mandiri. Desain struktur aplikasi pada praktikum ini diserahkan kepada Anda, tapi diharapkan tidak terlalu kompleks jika dibandingkan dengan aplikasi “Asah Otak Kuis” karya Spirit Mandiri.



¹ Referensi: <https://play.google.com/store/apps/details?id=com.spiritmandiri.asahotakkuis>

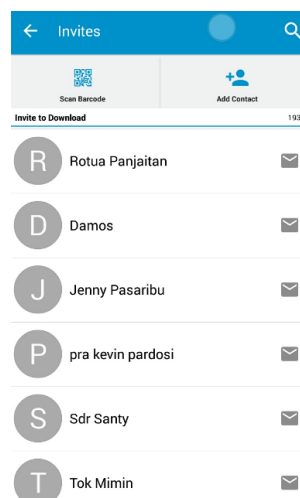




Pada praktikum-praktikum di minggu-minggu mendatang, Anda diharapkan untuk mulai membangun komponen-komponen aplikasi tersebut. Contohnya, Anda akan membuat komponen `Activity` aplikasi **Adu Jago Kuis** buatan Anda pada praktikum minggu ke-2 sesi-2 ini (topik pada praktikum ini adalah `Activity`). Hasil pembuatan komponen tersebut yang akan menjadi Deliverables Anda setiap sesi praktikum. Di minggu ke-14, Anda diharapkan menyeteror aplikasi Adu Jago Kuis secara lengkap dan tanpa error.

Deliverables Sesi Praktikum Ini

1. Layaknya aplikasi BBM yang memungkinkan penggunaanya mengundang teman untuk mengunduh aplikasi BBM tersebut, aplikasi Adu Jago Kuis yang telah anda buat juga diharapkan untuk memiliki fungsi 'Undang Teman'. Untuk sementara, fungsi ini hanya akan menampilkan data pada aplikasi kontak anda. Anda tidak perlu mengimplementasikan fungsi kirim sms/email untuk mengirim undangan. Desain untuk aplikasi ini diserahkan kepada Anda. Berikut gambar contoh undang teman pada aplikasi BBM.



2. Implementasikanlah *Broadcast Receiver* pada aplikasi Adu Jago Kuis Anda. *Broadcast Receiver* yang akan Anda implementasikan adalah *persentase* baterai seperti pada contoh praktikum. Namun Anda tidak perlu menampilkan *progress bar* baterai karena *progress bar* baterai sendiri sudah ditampilkan pada perangkat anda oleh sistem Android. Anda hanya akan memunculkan *Toast* dengan kondisi:
- Jika kondisi baterai 100% maka muncul "Baterai penuh",
 - Jika baterai dalam kondisi <5% maka muncul pesan "Baterai lemah".

Contoh penggunaan *Toast*:

```
Context context = getApplicationContext();  
String text = "Message saved as draft.";  
int duration = Toast.LENGTH_SHORT;  
  
Toast toast = Toast.makeText(context, text, duration);  
toast.show();
```

Output:

