LAPORAN PEMROGRAMAN ANDROID

JOB SHEET MINGGU 6

Intent

Oleh:

DYAH PUTRI AMBARSARI NIM. 1731710107

Kelas MI-2F



PROGRAM STUDI MANAJEMEN INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

**20 MARET 2019**

**Praktikum 1 - Mengirim Data antar Activity(Explicit Intent)**

|  |  |
| --- | --- |
| Langkah | Praktikum |
|  | Buka aplikasi Android Studio dan buat sebuah project baru. Nama aplikasi bisa diisikan sesuai dengan keinginan.  Pilih Empty Activity untuk template project |
|  | Pilihlah minimum sdk sesuai kebutuhan, misalnya pada praktikum ini kita gunakan API 15: Android 4.0.3. Terima nama aktivitas default (MainActivity). Pastikan kotak Generate Layout file dicentang, kemudian klik Finish |
|  | Buka res/layout/activity\_main.xml. Pada Layout Editor, klik tab Design di bagian bawah layar. Ubah layout yang digunakan menjadi **RelativeLayout** |
|  | Hapus TextView yang berbunyi "Hello World" |
|  | Tambahkan sebuah Plain Text. Beralih ke XML Editor (klik tab Text) dan modifikasi atribut Plain Text    Jika terdapat kesalahan, abaikan sementara kesalahan yang muncul tersebut. Masalah ini akan kita atasi di langkah selanjutnya. |
|  | Letakkan kursor pada kata "Ketikkan pesan". Tekan **Alt+Enter** (**Option+Enter** di Mac), dan pilih **Extract string resources**. |
|  | Ubah nama Sumber Daya menjadi editText\_main dan klik OK.    Proses ini membuat sumber daya string dalam file values/res/string.xml, dan string dalam kode Anda digantikan dengan referensi ke sumber daya string tersebut. |
|  | Tambahkan sebuah Button ke layout dalam posisi apa pun. Beralih ke XML Editor (klik tab Text) dan modifikasi atribut Button sebagai berikut: |
|  |  |
|  | Letakkan kursor pada kata "Send". Tekan **Alt+Enter** (**Option+Enter** di Mac), dan pilih **Extract string resources**. |
|  | Ubah nama Sumber Daya menjadi button\_main dan klik OK. |
|  | Kode Program secara keseluruhan untuk activity\_main.xml seperti gambar di bawah ini. |
|  | Dalam Editor XML, letakkan kursor pada kata "launchSecondActivity". Tekan **Alt+Enter** (**Option+Enter** di Mac) dan pilih **Create 'launchSecondActivity(View)' in 'MainActivity**. |
|  | Dalam launchSecondActivity, tambahkan pernyataan log yang bernama "Button diklik!" |
|  | Tempatkan kursor pada kata "Log" dan tekan **Alt+Enter** (**Option+Enter** di Mac). Android Studio menambahkan pernyataan impor untuk |
|  | Pada bagian atas kelas, tambahkan konstanta untuk variabel LOG\_TAG: |
|  | Kode Program secara keseluruhan untuk MainActivity.java seperti gambar di bawah ini. |

**Membuat Activity kedua**

|  |  |
| --- | --- |
| Langkah | Praktikum |
|  | Klik folder app untuk proyek Anda dan pilih **File > New > Activity > Empty Activity** |
|  | Namakan aktivitas baru "SecondActivity." Pastikan Generate Layout File dicentang, dan nama layout akan diisi dengan activity\_second. |
|  | Klik Finish. Android Studio menambahkan layout aktivitas baru (activity\_second) dan file Java baru (SecondActivity) ke proyek Anda untuk aktivitas baru tersebut. Ini juga akan memperbarui manifes Android untuk menyertakan aktivitas baru |
|  | Buka manifests/AndroidManifest.xml. |
|  | Temukan elemen <activity> yang dibuat Android Studio untuk aktivitas kedua. |
|  | Tambahkan atribut ini ke elemen <activity>: |
|  | Tempatkan kursor di "Second Activity" dan tekan **Alt+Enter** (**Option+Enter** di Mac). Tempatkan kursor di "Second Activity" dan tekan **Alt+Enter** (**Option+Enter** di Mac). |
|  | Pilih **Extract string resource**, namakan sumber daya "activity2\_name", dan klik OK. Android Studio menambahkan sumber daya string untuk label aktivitas. |
|  | Tambahkan elemen <meta-data> element inside the <activity> di dalam elemen untuk aktivitas kedua : |
|  | Kode Program secara keseluruhan untuk AndroidManifest.xml seperti gambar di bawah ini. |
|  |  |
|  | Buka res/layout/activity\_second.xml. Ubah layout yang digunakan menjadi **RelativeLayout** |
|  | Tambahkan TextView ("Plain Textview" dalam Layout Editor). Berikan atribut berikut pada TextView: |
|  | Ekstrak string "Pesan Diterima" ke sumber daya bernama text\_header. |
|  | Tambahkan TextView yang kedua (Plain Textview dalam Layout Editor). Berikan atribut berikut pada TextView:    android:layout\_below "@+id/text\_header" android:textSize "?android:attr/textAppearanceMedium" |
|  | Keseluruhan koding : |

Menambah Intent :

|  |  |
| --- | --- |
| Langkah | Praktikum |
|  | Buka file MainActivity.java |
|  | Buat intent baru dalam metode launchSecondActivity().  Konstruktor intent memerlukan dua argumen untuk intent eksplisit: konteks Aplikasi dan komponen spesifik yang akan menerima intent tersebut. Di sini kita harus menggunakan this sebagai konteksmya, dan SecondActivity.class sebagai kelas spesifiknya. |
|  | Tempatkan kursor di intent dan tekan **Alt+Enter** (**Option+Enter** di Mac) untuk menambahkan impor ke kelas intent |
|  | Panggil metode startActivity() dengan intent baru sebagai argumennya.  startActivity(intent); |
|  | Tambahkan konstanta publik di bagian atas kelas untuk mendefinisikan kunci untuk ekstra intent:  public static final String EXTRA\_MESSAGE = "com.exmple.asus.intent.extra.MESSAGE"; |
|  | Tambahkan variabel privat di bagian atas kelas untuk menampung objek EditText. Mengimpor kelas EditText.  private EditText mMessageEditText; |
|  | Dalam metode onCreate(), gunakan **findViewByID** untuk mendapatkan referensi tentang contoh EditText dan menetapkannya ke variabel privat tersebut:  mMessageEditText = (EditText) findViewById(R.id.editText\_main); |
|  | Dalam metode launchSecondActivity(), di bawah intent baru, dapatkan teks dari EditText sebagai string:  String message = mMessageEditText.getText().toString(); |
|  | Tambahkan string tersebut ke intent sebagai sebuah ekstra dengan konstanta EXTRA\_MESSAGE sebagai kunci dan string-nya sebagai nilai  :  intent.putExtra(EXTRA\_MESSAGE, message); |
|  | Hasil koding : |
|  | Buka file SecondActivity.java |
|  |  |
|  | Dalam metode onCreate(), dapatkan intent yang mengaktifkan aktivitas berikut ini:  Intent intent = getIntent(); |
|  | Dapatkan string berisi pesan dari ekstra intent menggunakan variabel statis MainActivity.EXTRA\_MESSAGE sebagai kuncinya:  String message = intent.getStringExtra(MainActivity.EXTRA\_MESSAGE); |
|  | Gunakan findViewByID untuk mendapatkan referensi ke TextView untuk pesan dari layout (Anda mungkin perlu mengimpor kelas TextView):  TextView textView = (TextView) findViewById(R.id.text\_message); |
|  | Atur teks TextView tersebut ke string dari ekstra intent:  textView.setText(message); |
|  | Hasil akhir koding : |
|  | Hasil :  D:\KULIAH\SEMESTER 4\Project Mobile\Intent\ss\Screenshot_2019-03-20-12-12-14-233_com.dyahputri.android.intent.png D:\KULIAH\SEMESTER 4\Project Mobile\Intent\ss\Screenshot_2019-03-20-12-12-18-542_com.dyahputri.android.intent.png |

## Praktikum 2 - Memanfaatkan Defined Action (Implicit Intent)

Membuat Layout

|  |  |
| --- | --- |
| Langkah | Praktikum |
|  | Buka aplikasi Android Studio dan buat sebuah project baru. Nama aplikasi bisa diisikan sesuai dengan keinginan. Pilih Empty Activity untuk template project |
|  | Pilihlah minimum sdk sesuai kebutuhan, misalnya pada praktikum ini kita gunakan API 15: Android 4.0.3. Terima nama aktivitas default (MainActivity). Pastikan kotak Generate Layout file dicentang, kemudian klik Finish |
|  | Buka file res/values/strings.xml dan tambahkan sumber daya string berikut ini:  http://www.polinema.ac.id Open Website Politeknik Negeri Malang Open Location |
|  | Ubah RelativeLayout default ke LinearLayout. Tambahkan atribut android:orientation dan berikan nilai "vertical.". |
|  | Buka res/layout/activity\_main.xml. Pada Layout Editor, klik tab Design di bagian bawah layar. Ubah layout yang digunakan menjadi **LinearLayout** |
|  | Atur atribut dari Edit Text dan Button tersebut seperti contoh berikut. |
|  | Koding secara keseluruhan : |

#### **Mengimplementasikan Buka Situs Website**

|  |  |
| --- | --- |
| Langkah | Praktikum |
|  | Kita akan mengimplementasikan metode handler on-click untuk tombol pertama dalam layout ("Open Website.") Tindakan ini menggunakan intent implisit untuk mengirimkan URI yang diberikan ke aktivitas yang dapat menangani intent implisit tersebut (seperti browser web).   1. Buka file MainActivity.java. |
|  | Tambahkan variabel privat di bagian atas kelas untuk menampung objek EditText untuk URI situs web. |
|  | Dalam metode onCreate(), gunakan findViewByID() untuk mendapatkan referensi tentang instance EditText dan menetapkannya ke variabel privat tersebut |
|  |  |
|  | Dalam metode onCreate(), gunakan findViewByID() untuk mendapatkan referensi tentang instance EditText dan menetapkannya ke variabel privat tersebut |
|  | Buat metode baru bernama openWebsite()  Dapatkan nilai string EditText dan Enkode dan parse string ke dalam objek Uri |
|  | Dapatkan nilai string EditText |
|  | Buat intent baru dengan Intent.ACTION\_VIEW sebagai tindakan dan URI sebagai data |
|  | Gunakan resolveActivity() dan pengelola paket Android untuk menemukan aktivitas yang dapat menangani intent implisit. Periksa untuk memastikan permintaan berhasil diatasi.  Dalam pernyataan if, panggil startActivity() untuk mengirimkan intent.  Tambahkan blok else untuk mencetak pesan log jika intent tidak bisa diatasi. |
|  | Kode program secara keseluruhan : |

#### **Mengimplementasikan Buka Lokasi**

|  |  |
| --- | --- |
| Langkah | Praktikum |
|  | Buka file MainActivity.java. |
|  | Tambahkan variabel privat di bagian atas kelas untuk menampung objek EditText bagi URI lokasi. |
|  | Dalam metode onCreate(), gunakan findViewByID() untuk mendapatkan referensi tentang instance EditText dan menetapkannya ke variabel privat tersebut |
|  | Buat metode baru bernama openLocation untuk digunakan sebagai metode onClick untuk tombol Open Location |
|  | Dapatkan nilai string EditText mLocationEditText. |
|  | Parse string itu ke dalam objek Uri dengan kueri penelusuran geo: |
|  | Buat intent baru dengan Intent.ACTION\_VIEW sebagai tindakan dan loc sebagai datanya. |
|  | Pecahkan intent dan periksa untuk memastikan intent berhasil diatasi. Jika demikian, startActivity(), jika tidak catat log pesan kesalahan |
|  | Kode Program secara keseluruhan untuk MainActivity.java seperti gambar di bawah ini. |
|  | Hasil :  D:\KULIAH\SEMESTER 4\Project Mobile\Intent\ss\Screenshot_2019-03-20-13-58-27-454_com.dyahputri.android.implicit_intent.png  D:\KULIAH\SEMESTER 4\Project Mobile\Intent\ss\Screenshot_2019-03-20-13-58-43-164_com.google.android.apps.maps.png D:\KULIAH\SEMESTER 4\Project Mobile\Intent\ss\Screenshot_2019-03-20-13-58-35-728_com.android.browser.png |

**Latihan :**

Buatlah sebuah aplikasi Android untuk proses Registrasi yang terdiri dari dua activity. Pada activity pertama, terdapat 3 Edit Text (Nama, Alamat, dan No. Hp) dan 1 Button (Register). Pada saat Button diklik, activity kedua akan muncul dengan menampilkan Nama dan Alamat sesuai isian sebelumnya. Selain itu, pada activity kedua juga terdapat sebuah Button (Share). Apabila Button tersebut diklik, maka data Nama dan Alamat akan di-share.

|  |  |
| --- | --- |
| Langkah | Praktikum |
|  | Buka aplikasi Android Studio dan buat sebuah project baru. Nama aplikasi bisa diisikan sesuai dengan keinginan. Pilih Empty Activity untuk template project |
|  | Pilihlah minimum sdk sesuai kebutuhan, misalnya pada praktikum ini kita gunakan API 15: Android 4.0.3. Terima nama aktivitas default (MainActivity). Pastikan kotak Generate Layout file dicentang, kemudian klik Finish |
|  | Buka res/layout/activity\_main.xml. Pada Layout Editor, klik tab Design di bagian bawah layar. Ubah layout yang digunakan menjadi **RelativeLayout dan tambahkan juga constraint layout** |
|  | Hapus TextView yang berbunyi "Hello World" |
|  | Tambahkan sebuah 3 buah Plain Text . Beralih ke XML Editor (klik tab Text) dan modifikasi atribut Plain Text  Seperti dibawah :  Untuk pengisian nomor telepon    Untuk pengisian nama    Untuk pengisian alamat |
|  | Letakkan kursor pada kata "nama". Tekan **Alt+Enter** (**Option+Enter** di Mac), dan pilih **Extract string resources**. |
|  | Ubah nama Sumber Daya menjadi namaText dan klik OK.    Proses ini membuat sumber daya string dalam file values/res/string.xml, dan string dalam kode Anda digantikan dengan referensi ke sumber daya string tersebut. |
|  | Tambahkan sebuah Button ke layout dalam posisi apa pun. Beralih ke XML Editor (klik tab Text) dan modifikasi atribut Button sebagai berikut: |
|  | Letakkan kursor pada kata "login". Tekan **Alt+Enter** (**Option+Enter** di Mac), dan pilih **Extract string resources**. |
|  | Ubah nama Sumber Daya menjadi button dan klik OK. |
|  | Dalam Editor XML, letakkan kursor pada kata "launchSecondActivity". Tekan **Alt+Enter** (**Option+Enter** di Mac) dan pilih **Create 'launchSecondActivity(View)' in 'MainActivity**. |
|  | Dalam launchSecondActivity, tambahkan pernyataan log yang bernama "Login" |
|  | Tempatkan kursor pada kata "Log" dan tekan **Alt+Enter** (**Option+Enter** di Mac). Android Studio menambahkan pernyataan impor untuk |
|  | Pada bagian atas kelas, tambahkan konstanta untuk variabel LOG\_TAG: |
|  | Kode Program secara keseluruhan untuk MainActivity.java seperti gambar di bawah ini. |

**Membuat Activity kedua**

|  |  |
| --- | --- |
| Langkah | Praktikum |
|  | Klik folder app untuk proyek Anda dan pilih **File > New > Activity > Empty Activity** |
|  | Namakan aktivitas baru "SecondActivity." Pastikan Generate Layout File dicentang, dan nama layout akan diisi dengan activity\_second. |
|  | Klik Finish. Android Studio menambahkan layout aktivitas baru (activity\_second) dan file Java baru (SecondActivity) ke proyek Anda untuk aktivitas baru tersebut. Ini juga akan memperbarui manifes Android untuk menyertakan aktivitas baru |
|  | Buka manifests/AndroidManifest.xml. |
|  | Temukan elemen <activity> yang dibuat Android Studio untuk aktivitas kedua. |
|  | Tambahkan atribut ini ke elemen <activity>: |
|  | Tempatkan kursor di "Second Activity" dan tekan **Alt+Enter** (**Option+Enter** di Mac). Tempatkan kursor di "Second Activity" dan tekan **Alt+Enter** (**Option+Enter** di Mac). |
|  | Pilih **Extract string resource**, namakan sumber daya "activity2\_name", dan klik OK. Android Studio menambahkan sumber daya string untuk label aktivitas. |
|  | Tambahkan elemen <meta-data> element inside the <activity> di dalam elemen untuk aktivitas kedua : |
|  | Kode Program secara keseluruhan untuk AndroidManifest.xml seperti gambar di bawah ini. |
|  | Beralih ke activity\_second |
|  | Buka res/layout/activity\_second.xml. Ubah layout yang digunakan menjadi **RelativeLayout** |
|  | Tambahkan TextView ("Plain Textview" dalam Layout Editor). Berikan atribut berikut pada TextView: |
|  | Ekstrak string "Registrasi di terima" ke sumber daya bernama text\_header. |
|  | Tambahkan TextView yang kedua (Plain Textview dalam Layout Editor). Berikan atribut berikut pada TextView:    android:layout\_below "@+id/text\_header" android:textSize "?android:attr/textAppearanceMedium" |
|  | Keseluruhan koding : |

Menambah Intent :

|  |  |
| --- | --- |
| Langkah | Praktikum |
|  | Buka file MainActivity.java |
|  | Buat intent baru dalam metode launchSecondActivity().  Konstruktor intent memerlukan dua argumen untuk intent eksplisit: konteks Aplikasi dan komponen spesifik yang akan menerima intent tersebut. Di sini kita harus menggunakan this sebagai konteksmya, dan SecondActivity.class sebagai kelas spesifiknya. |
|  | Tempatkan kursor di intent dan tekan **Alt+Enter** (**Option+Enter** di Mac) untuk menambahkan impor ke kelas intent |
|  | Panggil metode startActivity() dengan intent baru sebagai argumennya.  startActivity(intent); |
|  | Tambahkan konstanta publik di bagian atas kelas untuk mendefinisikan kunci untuk ekstra intent: |
|  | Tambahkan variabel privat di bagian atas kelas untuk menampung objek EditText. Mengimpor kelas EditText. Kita deklarasikan menjadi 3 karena kita akan menampilkan nama, telepon, dan alamaat |
|  | Dalam metode onCreate(), gunakan **findViewByID** untuk mendapatkan referensi tentang contoh EditText dan menetapkannya ke variabel privat tersebut: |
|  | Dalam metode launchSecondActivity(), di bawah intent baru, dapatkan teks dari EditText sebagai string: |
|  | Tambahkan string tersebut ke intent sebagai sebuah ekstra dengan konstanta EXTRA\_MESSAGE sebagai kunci dan string-nya sebagai nilai terdiri dari nama, telephon dan alamat |
|  | Hasil koding : |
|  | Buka file SecondActivity.java |
|  | Tempatkan kursor di intent dan tekan **Alt+Enter** (**Option+Enter** di Mac) untuk menambahkan impor ke kelas intent. |
|  | Dalam metode onCreate(), dapatkan intent yang mengaktifkan aktivitas berikut ini: |
|  | Dapatkan string berisi pesan dari ekstra intent menggunakan variabel statis MainActivity.EXTRA\_MESSAGE sebagai kuncinya: |
|  | Gunakan findViewByID untuk mendapatkan referensi ke TextView untuk pesan dari layout (Anda mungkin perlu mengimpor kelas TextView): |
|  | Atur teks TextView tersebut ke string dari ekstra intent: |
|  | Untuk dapat mengshare informasi kita tambahkan |
|  | Serta method    Yang terhubung dengan button di activity\_second.xml |
|  | Koding scara keseluruhan |
|  | Hasil nya :  D:\KULIAH\SEMESTER 4\Project Mobile\Intent\ss\Screenshot_2019-03-24-23-22-10-277_com.example.malcuisine.png  D:\KULIAH\SEMESTER 4\Project Mobile\Intent\ss\Screenshot_2019-03-24-23-22-16-126_com.example.malcuisine.png D:\KULIAH\SEMESTER 4\Project Mobile\Intent\ss\Screenshot_2019-03-24-23-22-21-134_android.png  C:\Users\Putri\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_2019-03-26-04-55-50-114_com.whatsapp.png |