

Modul Perkuliahan

**Pemrograman *Mobile*:
*User Experience***

Dr. Habibullah Akbar

Session 5

2019

SESSION OUTCOMES

1. Mahasiswa melakukan *refreshment* mengenai sesi sebelumnya, yaitu pembahasan aktifitas dan *intent*
2. Mahasiswa dapat menjelaskan perbedaan *user interface* (UI) dan *user experience* (UX)
3. Mahasiswa dapat memahami tujuan dan cara menggunakan komponen UI yang interaktif
4. Mahasiswa memahami dan menggunakan menu navigasi untuk memberikan UX yang baik bagi user

OUTLINE MATERI:

1. Pendahuluan
2. Perbedaan *User Interface* dan *User Experience*
3. Komponen-komponen UI
4. Menu-menu Navigasi
5. Latihan membuat Aplikasi menggunakan Komponen Interaktif dan Menu Navigasi

ISI MATERI

1. Pendahuluan

Pada sesi sebelumnya, anda telah belajar mengenai konsep aktifitas dan *intent*. Setiap aplikasi biasanya memiliki lebih dari 1 aktifitas. Untuk berpindah dari 1 aktifitas kepada aktifitas lainnya yang masih dalam 1 aplikasi yang sama, anda dapat menggunakan *intent* eksplisit. Adapun untuk berpindah kepada aktifitas yang berada pada aplikasi yang berbeda, anda dapat menggunakan *intent* implisit.

Pada sesi ini, anda akan belajar mengenai komponen UI dan menu navigasi untuk menghasilkan aplikasi yang dapat memberikan pengalaman yang menyenangkan bagi *user*.

2. Perbedaan User Interface (UI) dan User Experience (UX)

User Interface (UI) merupakan layout yang digunakan oleh *user* untuk berinteraksi dengan aplikasi, lebih tepatnya dengan aktifitas. Gampangnya, UI adalah bagaimana tampilan dan tata letak dari aplikasi yang dibuat. Android menyediakan berbagai macam komponen-komponen *View* yang dapat digunakan untuk membuat UI seperti *EditText*, *CheckBox*, dan *Spinner*. Selain itu, android juga menyediakan modul UI lainnya seperti menu navigasi.

Desain UI haruslah mudah bagi *user* sehingga memberikan pengalaman yang menyenangkan saat menggunakan aplikasi. Untuk itu, anda perlu fokus pada penataan komponen-komponen UI yang serasi.

Berbeda dengan UI, *user experience* (UX) merupakan pengalaman total yang dirasakan oleh *user* saat berinteraksi dengan aplikasi anda. Anda bertanggung-jawab atas kenyamanan user dengan pemilihan menu navigasi yang tepat sesuai kebutuhan *user*. Jika *user* mendapatkan kepuasan dalam menggunakan aplikasi maka *user* akan kembali bahkan dapat merekomendasikan aplikasi anda kepada orang lain.

3. Komponen-komponen UI

Pada pertemuan sebelumnya sebenarnya anda telah mempelajari beberapa komponen UI yaitu *Button* dan *EditText*. *User* biasanya akan sering berinteraksi dengan *Buttons*. *Buttons* itu sendiri dapat berbentuk

- Teks, seperti yang ditunjukkan di sisi kiri gambar dibawah.
- *Icon*, seperti yang ditunjukkan di tengah gambar dibawah.
- Kedua-duanya dari *icon* dan teks, seperti yang ditunjukkan di sisi kanan gambar dibawah ini.



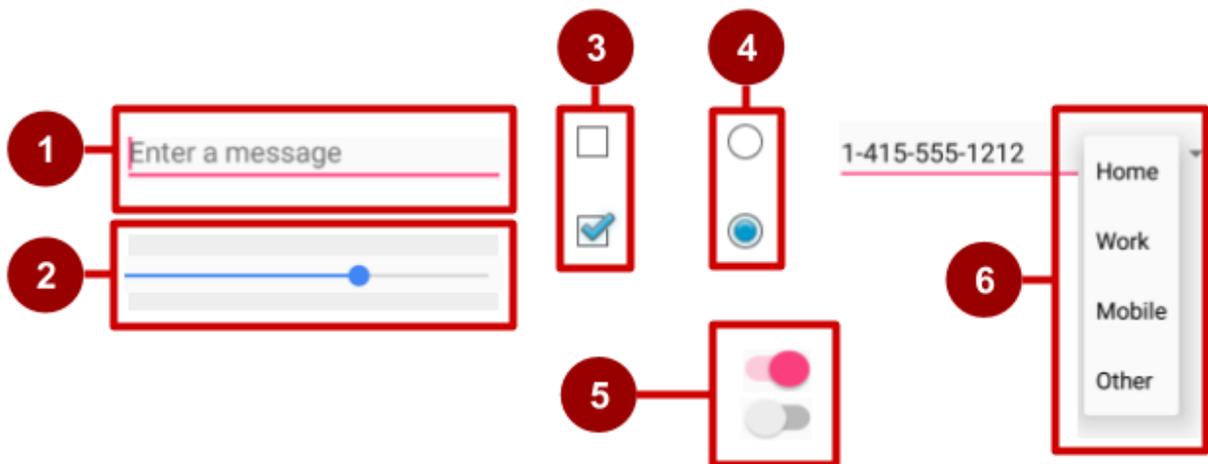
Untuk membuat tombol yang memiliki *icon* dan teks sekaligus pada layout XML anda, tambahkan atribut `drawableLeft` untuk menampilkan *icon* seperti yang ditunjukkan pada gambar di bawah ini:

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_text"
    android:drawableLeft="@drawable/button_icon"
    <!-- more attributes ... -->
/>
```

Note: untuk memilih dan menambahkan *icon* pada folder *drawable*, pelajari lebih lanjut di <https://google-developer-training.github.io/android-developer-fundamentals-course-concepts-v2/unit-2-user-experience/lesson-4-user-interaction/4-1-c-buttons-and-clickable-images/4-1-c-buttons-and-clickable-images.html>.

Selain *Button*, komponen lainnya yang penting untuk membuat UI yang baik adalah kontrol input. Kontrol input adalah elemen interaktif yang menerima masukan dari *user*. *User* memberikan masukan dengan cara memasukkan teks/angka ataupun dengan pilihan-pilihan pada views lainnya seperti *checkboxes*, *radio buttons*, and menu *drop-down*.

Android memberikan berbagai macam kontrol input seperti pada gambar dibawah



1. EditText, digunakan agar user dapat memasukkan teks menggunakan keyboard/keypad.
2. SeekBar digunakannya dengan menggeser ke sisi kiri ataupun sisi kanan untuk mengatur suatu nilai.
3. CheckBox adalah sebuah tombol yang setiap pilihannya dapat di-check ataupun uncheck sekaligus.
4. RadioButton adalah tombol dimana user hanya men-check ataupun uncheck, namun tidak bisa keduanya sekaligus. *User* harus memilih satu pilihan saja.
5. Switch digunakan untuk menghidupkan ataupun mematikan suatu pilihan.

- Spinner digunakan untuk memberikan menu *dropdown* agar *user* dapat memilih salah satu nilai dari daftar pilihan.

Jika aplikasi anda perlu mendapatkan masukan dari *user*, jadikan prosesnya semudah mungkin misalnya dengan mengantisipasi sumber data dan meminimalkan jumlah gerakan *user*.

Salah satu tips lainnya adalah dengan menggunakan kontrol input seperti aplikasi-aplikasi lain yang semisal. Misalnya, pengguna mengharapkan *Spinner* untuk menampilkan menu *drop-down*.

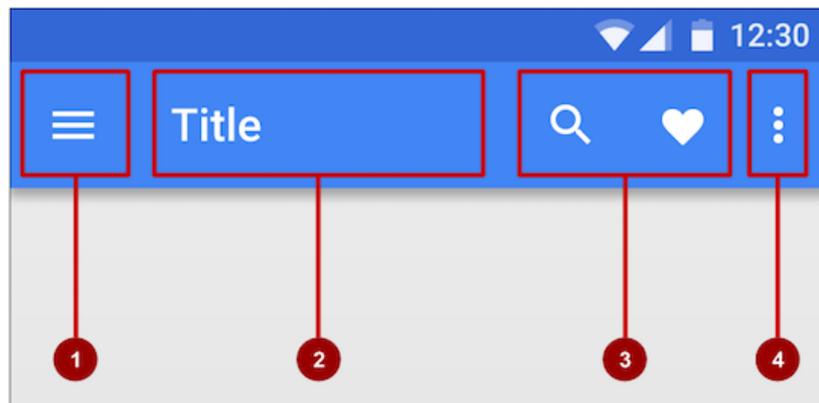
4. Menu-menu Navigasi

Sebelum kita membahas menu-menu navigasi, ada baiknya kita membahas bagian atas pada layout standar. Setiap anda membuat aktifitas seperti pada sesi sebelumnya maka android secara otomatis akan menyertakan bilah aplikasi (*app bar*). Bilah atas ini kadang juga disebut sebagai

- bilah aksi (*action bar*)
- *toolbar*

Bilah aplikasi secara *default* akan menampilkan judul aplikasi, sebagaimana yang telah ditentukan pada atribut `android:label` di file `AndroidManifest.xml`. Bilah aplikasi juga akan menyertakan tombol balik untuk navigasi ke aktivitas induk (saat anda berada pada aktifitas anak).

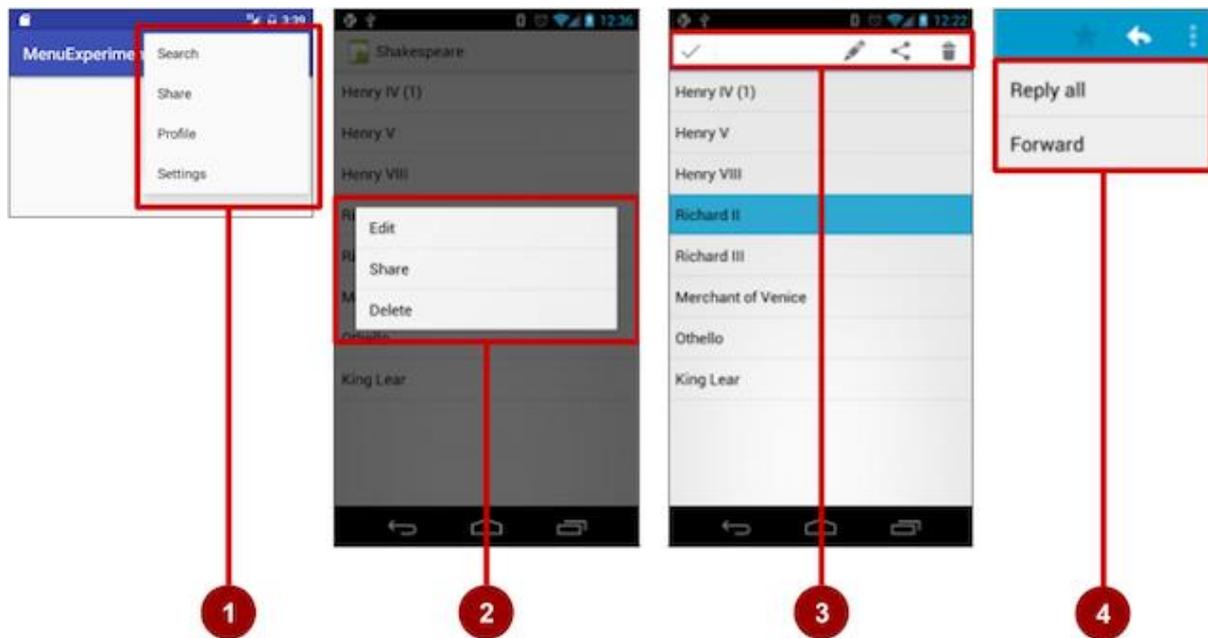
Bilah aplikasi biasanya dibagi menjadi empat area fungsional, seperti yang ditunjukkan pada gambar di bawah ini.



1. Tombol navigasi or tombol balik (*up button*).
2. Judul (*title*), digunakan untuk akan menampilkan nama nama aplikasi ataupun nama dari aktifitas.
3. *Action icons*, digunakan untuk membuka aktifitas secara langsung.
4. Menu *overflow*, digunakan untuk menyembunyikan item-item dari menu yang sifatnya opsional.

Menu digunakan untuk menghasilkan pengalaman interaksi *user* (UX) yang konsisten dimana user dapat memilih suatu fungsi khusus yang disediakan aplikasi. Misalnya untuk mencari informasi, menyimpan informasi, mengedit informasi, ataupun navigasi ke aktifitas yang lain.

Gambar dibawah ini menunjukkan jenis-jenis menu yang dapat anda gunakan dalam mengembangkan aplikasi Android.



Jenis-jenis menu

1. Menu Pilihan.

Menu pilihan akan muncul pada bilah aplikasi untuk menyediakan opsi utama dalam penggunaan aplikasi. Menu pilihan biasanya akan muncul di sisi kiri bilah aplikasi. Contoh menu pilihan adalah pencarian adalah *Search* untuk melakukan pencarian, *Share* untuk membagikan tautan, dan *Settings* untuk melakukan pengaturan tertentu (lihat gambar diatas).

2. Menu Kontekstual

Menu kontekstual berbeda dengan menu pilihan dimana menu kontekstual hanya akan muncul apabila *user* melakukan ketukan yang panjang (*long tap*) pada elemen di layar. Contoh pilihan pada menu kontekstual adalah *Edit* untuk mengubah elemen dan *Delete* untuk menghapus elemen tersebut.

3. Bilah Aksi Kontekstual.

Bilah aksi kontekstual fungsinya mirip dengan menu kontekstual, namun bilah aksi akan menutupi bilah aplikasi pada bagian atas layar. Contoh pilihan pada bilah aksi kontekstual sama dengan pilihan pada menu kontekstual yaitu *Edit* dan *Delete*.

4. Menu *Popup*

Menu *popup* akan muncul menutupi view yang diklik *user* ataupun dibawahnya, dan akan hilang jika *user* mengklik bagian di luar area menu *popup*. Contoh menu *popup* misal pada aplikasi *Gmail* adalah *Reply All* dan *Forward*.

5. Latihan membuat Aplikasi menggunakan Komponen Interaktif dan Menu

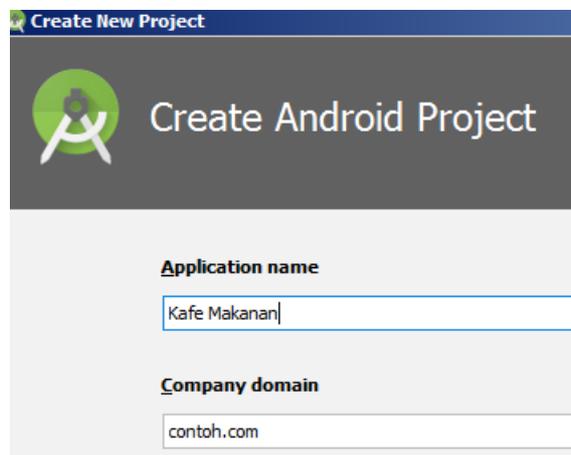
Pada latihan ini anda akan belajar untuk membuat sebuah aplikasi untuk pemesanan makanan yang menggunakan gambar sebagai komponen interaktif.

Note: anda dapat menggunakan metode `onClick()` untuk menampilkan sebuah pesan setelah user mengklik gambar tersebut ataupun berpindah pada aktifitas lainnya.

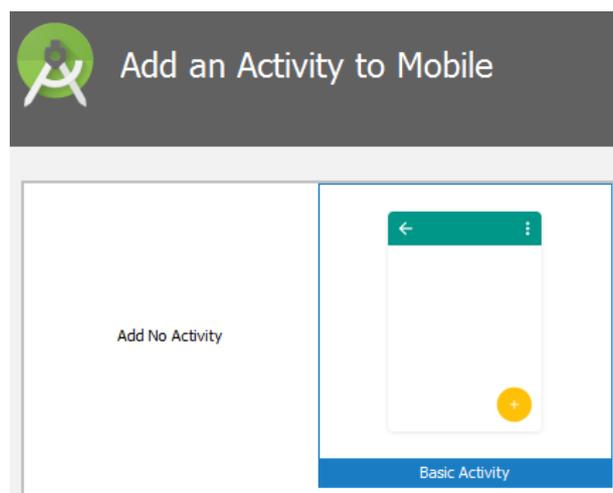
Aplikasi ini akan dibuat berdasarkan dua tahapan berikut:

5.1 Membuat Aktifitas Pemesanan

Pertama-tama, buatlah Android Studio anda dan buat sebuah proyek baru kemudian beri nama Kafe Makanan.



Lalu, pilih templat Basic Activity, klik *Next* dan *Finish*.



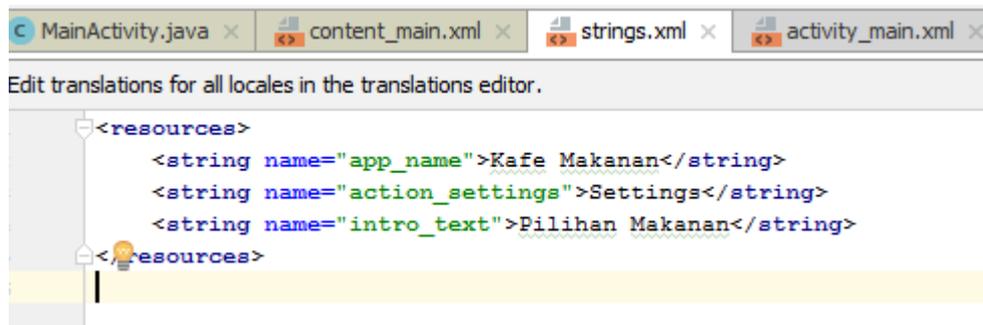
Tempat tersebut akan menghasilkan 2 layout yaitu

- `activity_main.xml` yang nantinya akan digunakan untuk membuat pada bagian 5.2
- `content_main.xml` untuk komponen interaktif.

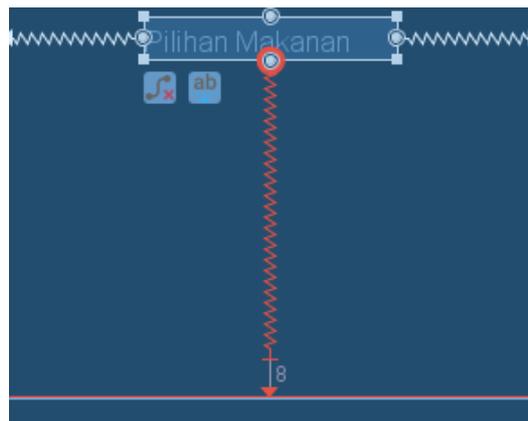
Bukalah `content_main.xml` dan pada tab Text, gantilah atribut `TextView` yang berisi nilai Hello World menjadi seperti berikut

```
<TextView
    android:id="@+id/textintro"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:text="@string/intro_text"
    android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

Seperti pada pelajaran sebelumnya anda telah belajar untuk membuat string global pada file `strings.xml` untuk digunakan pada `views`. Isi `intro_text` menjadi sebagai berikut



Kemudian *delete*-lah constraint dibawah dari `TextView` tersebut dengan mengklik bulatan pada bagian bawahnya seperti gambar dibawah. Atur juga top margin menjadi 8dp.



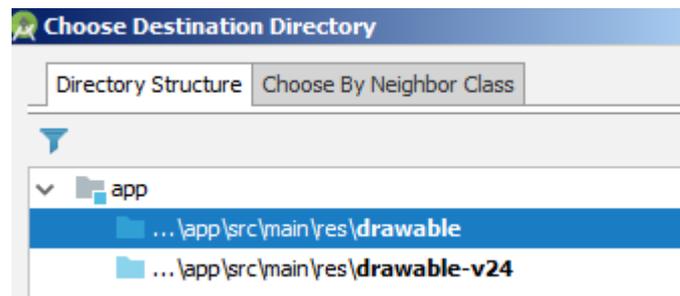
Berikutnya kita akan memasukkan pilihan-pilihan makanan berupa gambar. Pada contoh ini, kita akan menggunakan gambar dari halaman web berikut https://github.com/google-developer-training/android-fundamentals-apps-v2/blob/master/DroidCafe/app/src/main/res/drawable/donut_circle.png.

Gambar yang akan digunakan adalah

- donut_circle.png
- froyo_circle.png
- icecream_circle.png

note: anda dapat mengganti gambar yang anda sukai (gunakan file PNG), namun pada contoh ini harus berukuran sekitar 113 x 113 piksel.

Untuk menambahkan gambar, *Copy* gambar-gambar tersebut (bisa juga dengan Ctrl + C), lalu pilih folder drawable pada proyek (app > res > drawable), kemudian *Paste* (bisa juga dengan Ctrl + V). Setelah itu maka anda akan diminta memilih *directory*. Pilihlah drawable (bukan drawable-v24) kemudian klik OK.



Berikutnya bukalah `content_main.xml` dan klik tab *Design*. Tambahkan `ImageView` pada layout dan pilihlah gambar donut_circle lalu klik OK. Kemudian, tambahkan constraint terhadap `TextView` dan terhadap sisi kiri layout dengan masing-masing margin 24dp. Lalu set nilai atribut berikut

Attribute field	Enter the following:
ID	donut
contentDescription	Donuts are glazed and sprinkled with candy. (You can copy/paste the text into the field.)

Tambahkan 2 gambar lagi, yaitu gambar `ice_cream_circle` dan gambar `froyo_circle` dan letakkan berurutan dibawah gambar sebelumnya dengan margin keatas 24dp. Kemudian setingan kedua gambar tersebut dengan atribut berikut.

Attribute field	Enter the following:
ID	ice_cream
contentDescription	Ice cream sandwiches have chocolate wafers and vanilla filling. (You can copy/paste the text into the field.)
ID	froyo
contentDescription	FroYo is premium self-serve frozen yogurt. (You can copy/paste the text into the field.)

Setelah itu, anda perlu menjadikan setiap `contentDescription`. Bukalah tab `Text` dari layout tersebut maka anda akan mendapatkan warning.

```

<ImageView
    android:id="@+id/donut"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:contentDescription="Donuts are glazed and sprinkled with candy."
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textintro"
    app:srcCompat="@drawable/donut_circle" />

```

Hardcoded string "Donuts are glazed and sprinkled with candy.", should use @string resource [more...](#) (Ctrl+F1)

Oleh karena itu jadikanlah setiap string pada masing-masing contentDescription menjadi variable string global pada file strings.xml menjadi seperti berikut

```

<resources>
    <string name="app_name">Kafe Makanan</string>
    <string name="action_settings">Settings</string>
    <string name="intro_text">Pilihan Makanan</string>
    <string name="donuts">Donuts are glazed and sprinkled with candy.</string>
    <string name="ice_cream_sandwiches">Ice cream sandwiches have chocolate wafers and vanilla filling.</string>
    <string name="froyo">FroYo is premium self-serve frozen yogurt.</string>
</resources>

```

Dan ubah nilai contentDescription menjadi berikut

```

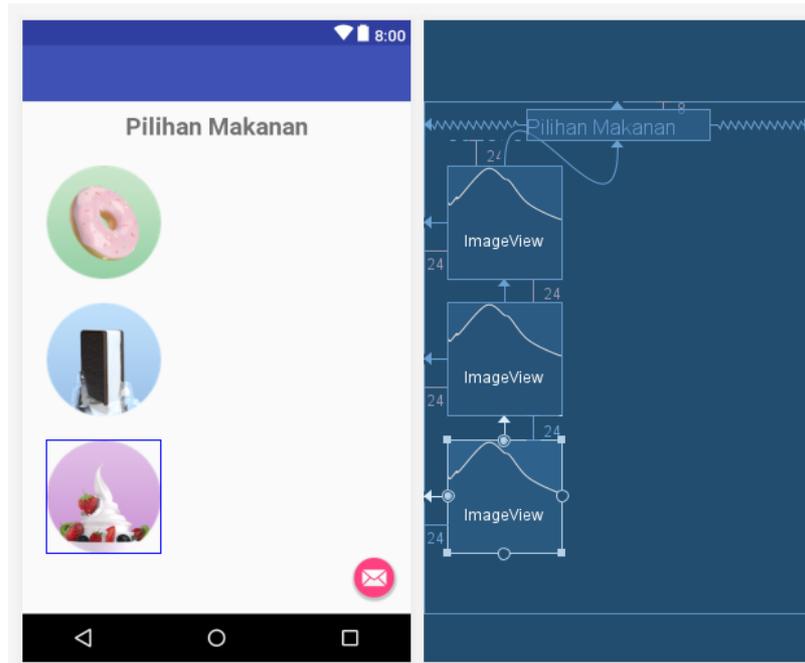
<ImageView
    android:id="@+id/donut"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:contentDescription="@string/donuts"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textintro"
    app:srcCompat="@drawable/donut_circle" />

<ImageView
    android:id="@+id/ice_cream"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:contentDescription="@string/ice_cream_sandwiches"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/donut"
    app:srcCompat="@drawable/icecream_circle" />

<ImageView
    android:id="@+id/froyo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:contentDescription="@string/froyo"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/ice_cream"
    app:srcCompat="@drawable/froyo_circle" />

```

Layout anda akan berbentuk seperti gambar dibawah ini



Berikutnya, anda perlu menambahkan TextView untuk memberikan deskripsi pada setiap ketiga pilihan makanan di atas.

Langkah-langkah menambahkan deskripsi

1. Tarik sebuah TextView pada layout
2. Tambahkan *constraint* TextView tersebut pada sisi kanan layout dengan menariknya dari sisi kanan TextView.
3. Tambahkan juga *constraint* pada sisi kiri TextView terhadap sisi kanan ImageView donut
4. Kemudian beri juga constraint sisi kanan terhadap layout. Atur ketiga constraints tersebut dengan margin 24dp.
5. Berikutnya beri nama atribut ID menjadi donut_description.

Lalu, pada bagian kanan jendela tab Design, set nilai atribut text sesuai string global yang telah dibuat, yaitu @string/donuts. Agar deskripsi teks tampak rapi, atur atribut layout_width menjadi 0dp.

Lakukan hal yang sama dengan menambahkan TextView untuk masing-masing gambar. Adapun untuk masing-masing ID aturlah menjadi ice_cream dan froyo agar menjadi seperti berikut ini

```

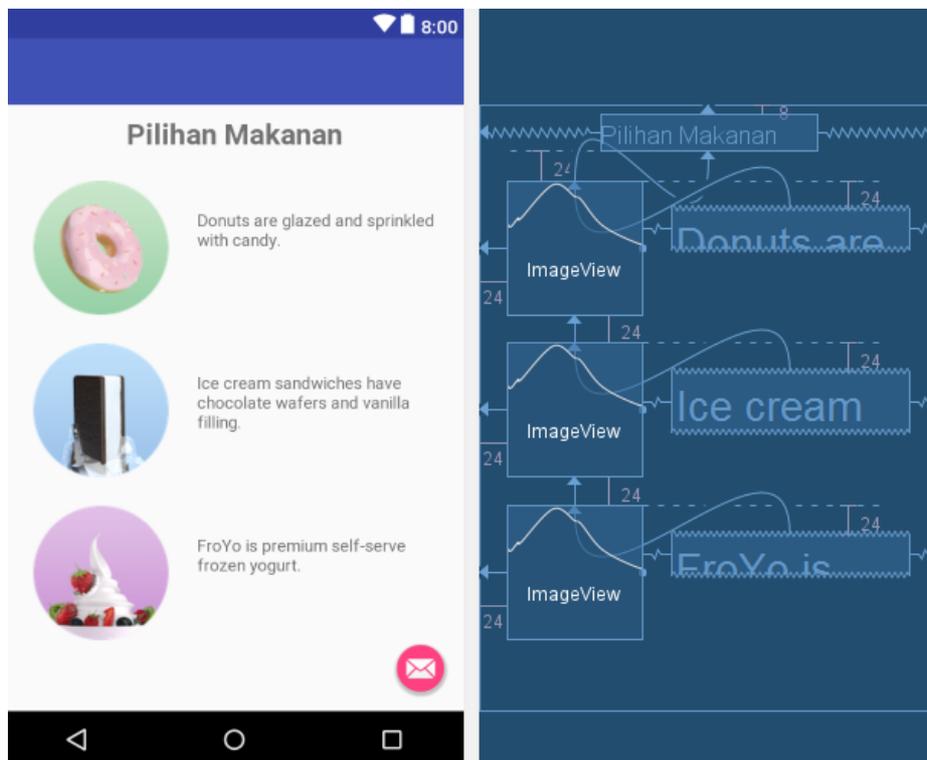
<TextView
    android:id="@+id/donut_description"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginEnd="24dp"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:text="@string/donuts"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/donut"
    app:layout_constraintTop_toTopOf="@+id/donut" />

<TextView
    android:id="@+id/ice_cream_description"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginEnd="24dp"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:text="@string/ice_cream_sandwiches"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/ice_cream"
    app:layout_constraintTop_toTopOf="@+id/ice_cream" />

<TextView
    android:id="@+id/froyo_description"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginEnd="24dp"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:text="@string/froyo"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/froyo"
    app:layout_constraintTop_toTopOf="@+id/froyo" />

```

Hasil pengaturan TextView pada langkah diatas akan menghasilkan layout sebagai berikut



Anda akan menambahkan aksi tertentu ketika *user* mengklik gambar makanan dengan menggunakan metode `onClick`. Misal untuk menampilkan sebuah pesan ataupun berpindah pada aktifitas lainnya. Untuk tujuan tersebut silahkan anda mencoba sendiri seperti pada <https://codelabs.developers.google.com/codelabs/android-training-clickable-images/index.html?index=.%2F.%2Fandroid-training#3>.

5.2 Membuat Menu pada Bagian Atas (Bilah Aplikasi)

Setelah anda membuat UI utama dari aplikasi Pilihan Makanan, maka anda perlu menambahkan bilah aplikasi yang biasanya berada pada bagian atas.

Pertama-tama bukalah file `activity_main.xml` lalu pilih tab `Text`. Anda akan melihat bahwasanya layout utama menggunakan layout `CoordinatorLayout` beserta layout `AppBarLayout`. `AppBarLayout` berfungsi seperti `LinearLayout` vertikal, yang menggunakan kelas `Toolbar` untuk menampilkan bilah aplikasi.

Pada file `activity_main.xml` juga pertanyaan `include layout` untuk memanggil layout lainnya seperti `layout_content_main.xml`. Pemisahan kedua layout ini bertujuan memudahkan anda untuk merubah isi layout tersebut tanpa mengganggu layout utama.

Jalankan aplikasi dan coba klik tombol menu *overflow* (tiga titik vertikal) di sisi kanan. menu, maka anda akan mendapatkan hanya 1 pilihan menu yaitu `Settings` seperti gambar berikut



Bukalah file AndroidManifest.xml, maka kita dapat melihat atribut android:theme secara default bernilai NoActionBar. Tema ini didefinisikan di file styles.xml (coba cek di **app > res > values > styles.xml**).

Untuk menambah item pada Menu Pilihan, bukalah file menu_main.xml yang berada pada **app > res > menu**. Ubahlah nilai atributnya menjadi

Attribute	Value
android:id	"@+id/action_contact"
android:title	"Contact"
app:showAsAction	"never"

Ganti string "Contact" menjadi variabel string global pada file strings.xml. Kemudian tambahkan beberapa item yaitu Order, Status dan Favorites sesuai setingan berikut ini

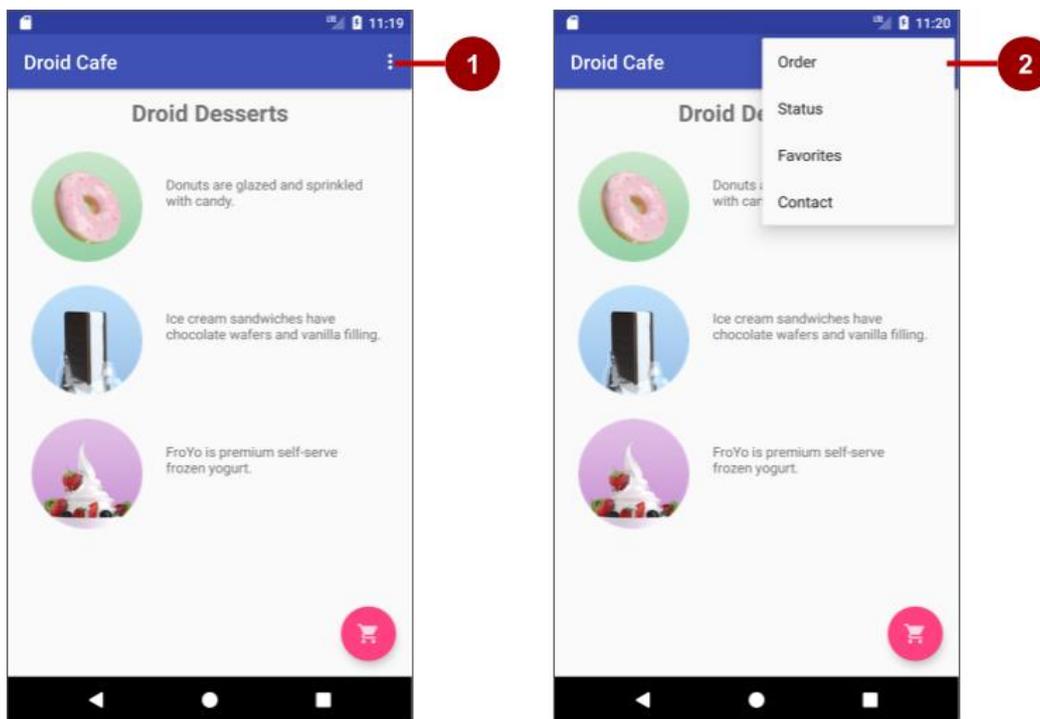
Attribute	Value
android:id	"@+id/action_order"
android:orderInCategory	"10"
android:title	"Order"
app:showAsAction	"never"

Status item attribute	Value
android:id	"@+id/action_status"
android:orderInCategory	"20"
android:title	"Status"
app:showAsAction	"never"

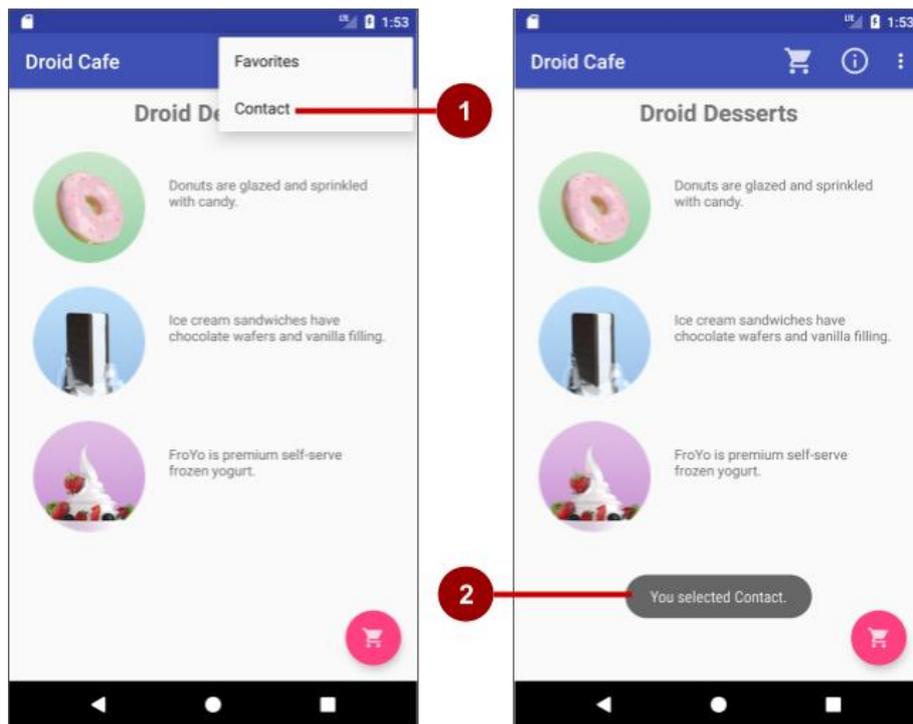
Favorites item attribute	Value
android:id	"@+id/action_favorites"
android:orderInCategory	"30"
android:title	"Favorites"
app:showAsAction	"never"

Bukalah file MainActivity.java dan gantilah pernyataan pada metode `onOptionsItemSelected()` dari `id action_settings` menjadi `id action_order`.

Jalankan aplikasi tersebut kemudian tap pada icon menu overflow pada bagian kanan atas. Hasilnya yaitu Menu Pilihan akan muncul pada sisi kanan atas tersebut seperti pada gambar berikut



Berikutnya anda dapat menambahkan *icon* pada item-item pilihan tersebut dan menampilkannya pada bilah aplikasi seperti berikut ini



Silahkan anda coba sendiri seperti pada tutorial berikut ini

<https://codelabs.developers.google.com/codelabs/android-training-menus-and-pickers/index.html?index=..%2F..%2Fandroid-training#3>

Simpulan

Pada sesi ini anda telah belajar mengenai konsep UI dan UX. Komponen-komponen UI yang biasa digunakan user untuk berinteraksi dengan aplikasi adalah EditText dan ImageView. Pada latihan membuat aplikasi diatas, ImageView tersebut dapat dibuat menjadi seperti *Button* dengan menambahkan metode `onClick` padanya. Selain itu, anda juga telah belajar cara menampilkan Menu Pilihan pada bilah aplikasi pada bagian atas layout. Hal tersebut merupakan modal anda untuk membuat UX yang baik bagi user. Namun demikian, anda perlu mencoba komponen lainnya seperti SeekBar, CheckBox, Spinner dan juga menu navigasi lainnya secara mandiri.

Pada sesi berikutnya, anda akan mempelajari bagaimana membuat desain tematik untuk menghasilkan aplikasi yang lebih konsisten dan sesuai dengan harapan *user*.

Referensi

1. <https://developer.android.com/courses/fundamentals-training/overview-v2>
2. <https://medium.com/@chunchuanlin/android-vs-ios-compare-20-ui-components-patterns-part-1-ad33c2418b45>