**Jobsheet 6 – Layout dan Navigasi**

Disusun untuk memenuhi Tugas Mata Kuliah Praktikum Pemrograman Mobile

Dosen Pembimbing : Ade Ismail, S.Kom., M.TI.



Disusun oleh

Liya Novitasari 2241760006

Kelas 3E

|  |
| --- |
|  |
|  |

**Jurusan Teknologi Informasi**

**Program Studi D-4 Sistem Informasi Bisnis**

**POLITEKNIK NEGERI MALANG**

**MALANG**

**2024**

**Praktikum**

**Praktikum 1: Membangun Layout di Flutter**

|  |  |
| --- | --- |
| **Langkah** | **Jawaban/Deskripsi** |
| 1 | Buatlah sebuah project flutter baru dengan nama **layout\_flutter**. Atau sesuaikan style laporan praktikum yang Anda buat. |
| 2 | Buka file main.dart lalu ganti dengan kode berikut. Isi nama dan NIM Anda di text title. |
| 3 | Langkah pertama adalah memecah tata letak menjadi elemen dasarnya:   * Identifikasi baris dan kolom. * Apakah tata letaknya menyertakan kisi-kisi (grid)? * Apakah ada elemen yang tumpang tindih? * Apakah UI memerlukan tab? * Perhatikan area yang memerlukan alignment, padding, atau borders.   Pertama, identifikasi elemen yang lebih besar. Dalam contoh ini, empat elemen disusun menjadi sebuah kolom: sebuah gambar, dua baris, dan satu blok teks.    Selanjutnya, buat diagram setiap baris. Baris pertama, disebut bagian Judul, memiliki 3 anak: kolom teks, ikon bintang, dan angka. Anak pertamanya, kolom, berisi 2 baris teks. Kolom pertama itu memakan banyak ruang, sehingga harus dibungkus dengan widget yang Diperluas.    Baris kedua, disebut bagian Tombol, juga memiliki 3 anak: setiap anak merupakan kolom yang berisi ikon dan teks.    Setelah tata letak telah dibuat diagramnya, cara termudah adalah dengan menerapkan pendekatan bottom-up. Untuk meminimalkan kebingungan visual dari kode tata letak yang banyak bertumpuk, tempatkan beberapa implementasi dalam variabel dan fungsi. |
| 4 | Pertama, Anda akan membuat kolom bagian kiri pada judul. Tambahkan kode berikut di bagian atas metode build() di dalam kelas MyApp:    **/\* soal 1 \*/** Letakkan widget Column di dalam widget Expanded agar menyesuaikan ruang yang tersisa di dalam widget Row. Tambahkan properti crossAxisAlignment ke CrossAxisAlignment.start sehingga posisi kolom berada di awal baris.  **/\* soal 2 \*/** Letakkan baris pertama teks di dalam Container sehingga memungkinkan Anda untuk menambahkan padding = 8. Teks ‘Batu, Malang, Indonesia' di dalam Column, set warna menjadi abu-abu.  **/\* soal 3 \*/** Dua item terakhir di baris judul adalah ikon bintang, set dengan warna merah, dan teks "41". Seluruh baris ada di dalam Container dan beri padding di sepanjang setiap tepinya sebesar 32 piksel. Kemudian ganti isi body text ‘Hello World' dengan variabel titleSection seperti berikut: |

**Praktikum 2: Implementasi button row**

|  |  |
| --- | --- |
| **Langkah** | **Jawaban/Deskripsi** |
| 1 | Bagian tombol berisi 3 kolom yang menggunakan tata letak yang sama—sebuah ikon di atas baris teks. Kolom pada baris ini diberi jarak yang sama, dan teks serta ikon diberi warna primer.  Karena kode untuk membangun setiap kolom hampir sama, buatlah metode pembantu pribadi bernama buildButtonColumn(), yang mempunyai parameter warna, Icon dan Text, sehingga dapat mengembalikan kolom dengan widgetnya sesuai dengan warna tertentu.  **lib/main.dart (\_buildButtonColumn)** |
| 2 | Buat Fungsi untuk menambahkan ikon langsung ke kolom. Teks berada di dalam Container dengan margin hanya di bagian atas, yang memisahkan teks dari ikon.  Bangun baris yang berisi kolom-kolom ini dengan memanggil fungsi dan set warna, Icon, dan teks khusus melalui parameter ke kolom tersebut. Sejajarkan kolom di sepanjang sumbu utama menggunakan MainAxisAlignment.spaceEvenly untuk mengatur ruang kosong secara merata sebelum, di antara, dan setelah setiap kolom. Tambahkan kode berikut tepat di bawah deklarasi titleSection di dalam metode build():  **lib/main.dart (buttonSection)** |
| 3 | Tambahkan variabel buttonSection ke dalam body seperti berikut: |

**Praktikum 3: Implementasi text section**

|  |  |
| --- | --- |
| **Langkah** | **Jawaban/Deskripsi** |
| 1 | Tentukan bagian teks sebagai variabel. Masukkan teks ke dalam Container dan tambahkan padding di sepanjang setiap tepinya. Tambahkan kode berikut tepat di bawah deklarasi buttonSection:    Dengan memberi nilai softWrap = true, baris teks akan memenuhi lebar kolom sebelum membungkusnya pada batas kata. |
| 2 | Tambahkan widget variabel textSection ke dalam body seperti berikut: |

**Praktikum 4: Implementasi image section**

|  |  |
| --- | --- |
| **Langkah** | **Jawaban/Deskripsi** |
| 1 | Anda dapat mencari gambar di internet yang ingin ditampilkan. Buatlah folder images di root project **layout\_flutter**. Masukkan file gambar tersebut ke folder images, lalu set nama file tersebut ke file pubspec.yaml seperti berikut:    Contoh nama file gambar di atas adalah lake.jpg |
| 2 | Tambahkan aset gambar ke dalam body seperti berikut:    BoxFit.cover memberi tahu kerangka kerja bahwa gambar harus sekecil mungkin tetapi menutupi seluruh kotak rendernya. |
| 3 | Pada langkah terakhir ini, atur semua elemen dalam ListView, bukan Column, karena ListView mendukung scroll yang dinamis saat aplikasi dijalankan pada perangkat yang resolusinya lebih kecil. |
|  |  |

**Praktikum 5: Membangun Navigasi di Flutter**

|  |  |
| --- | --- |
| **Langkah** | **Jawaban/Deskripsi** |
| 1 | Sebelum melanjutkan praktikum, buatlah sebuah project baru Flutter dengan nama **belanja**dan susunan folder seperti pada gambar berikut. Penyusunan ini dimaksudkan untuk mengorganisasi kode dan widget yang lebih mudah. |
| 2 | Buatlah dua buah file dart dengan nama home\_page.dart dan item\_page.dart pada folder **pages**. Untuk masing-masing file, deklarasikan class HomePage pada file home\_page.dart dan ItemPage pada item\_page.dart. Turunkan class dari StatelessWidget. Gambaran potongan kode dapat anda lihat sebagai berikut. |
| 3 | Setelah kedua halaman telah dibuat dan didefinisikan, bukalah file main.dart. Pada langkah ini anda akan mendefinisikan **Route**untuk kedua halaman tersebut. Definisi penamaan **route**harus bersifat **unique**. Halaman **HomePage**didefinisikan sebagai **/**. Dan halaman **ItemPage**didefinisikan sebagai **/item**. Untuk mendefinisikan halaman awal, anda dapat menggunakan named argument initialRoute. Gambaran tahapan ini, dapat anda lihat pada potongan kode berikut. |
| 4 | Sebelum melakukan perpindahan halaman dari HomePage ke ItemPage, dibutuhkan proses pemodelan data. Pada desain mockup, dibutuhkan dua informasi yaitu nama dan harga. Untuk menangani hal ini, buatlah sebuah file dengan nama item.dart dan letakkan pada folder **models**. Pada file ini didefinisikan pemodelan data yang dibutuhkan. Ilustrasi kode yang dibutuhkan, dapat anda lihat pada potongan kode berikut. |
| 5 | Pada halaman HomePage terdapat ListView widget. Sumber data ListView diambil dari model List dari object Item. Gambaran kode yang dibutuhkan untuk melakukan definisi model dapat anda lihat sebagai berikut. |
| 6 | Untuk menampilkan ListView pada praktikum ini digunakan itemBuilder. Data diambil dari definisi model yang telah dibuat sebelumnya. Untuk menunjukkan batas data satu dan berikutnya digunakan widget Card. Kode yang telah umum pada bagian ini tidak ditampilkan. Gambaran kode yang dibutuhkan dapat anda lihat sebagai berikut.  Jalankan aplikasi pada emulator atau pada device anda. |
| 7 | Item pada ListView saat ini ketika ditekan masih belum memberikan aksi tertentu. Untuk menambahkan aksi pada ListView dapat digunakan widget InkWell atau GestureDetector. Perbedaan utamanya InkWell merupakan material widget yang memberikan efek ketika ditekan. Sedangkan GestureDetector bersifat umum dan bisa juga digunakan untuk gesture lain selain sentuhan. Pada praktikum ini akan digunakan widget InkWell.  Untuk menambahkan sentuhan, letakkan cursor pada widget pembuka Card. Kemudian gunakan shortcut quick fix dari VSCode (**Ctrl + .** pada Windows atau **Cmd + .** pada MacOS). Sorot menu wrap with widget... Ubah nilai widget menjadi InkWell serta tambahkan named argument onTap yang berisi fungsi untuk berpindah ke halaman ItemPage. Ilustrasi potongan kode dapat anda lihat pada potongan berikut. |
|  |  |