

Komponen dalam Mendesain Aplikasi

TI2137 - Pengembangan Aplikasi Mobil Front-End

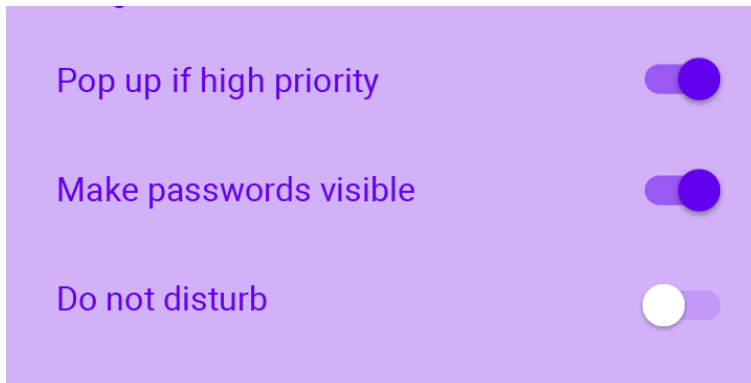


Team Teaching

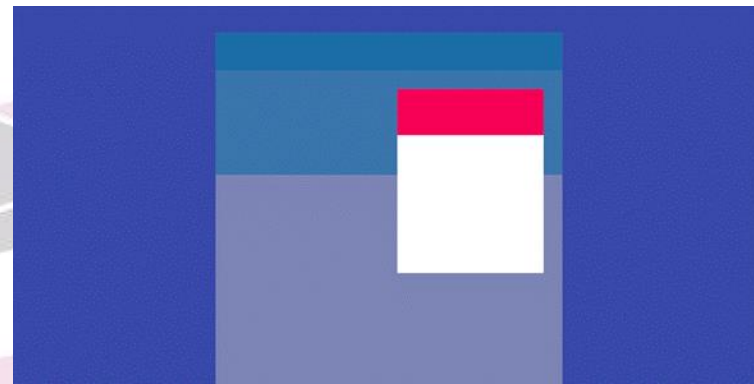
- Sunaryo Winardi, S.Kom., M.Tl.
- Sio Jurnal Pipin, S.Kom., M.Kom.
- M. Taufiq Hidayat Pohan, S.Kom.
- Richy Rotuahta Saragih, S.Kom.



Komponen dalam Mendesain Aplikasi Part. 3

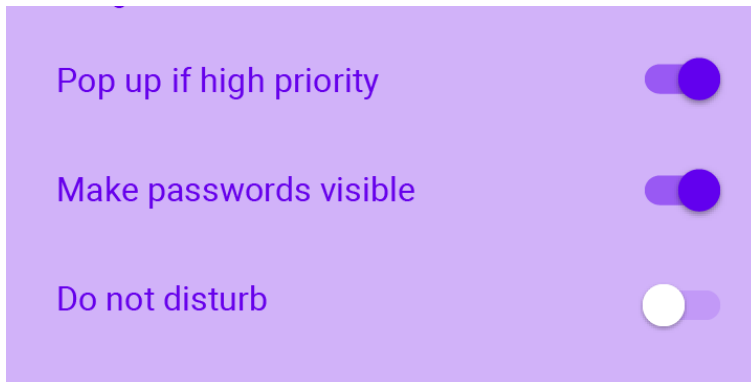


Switch

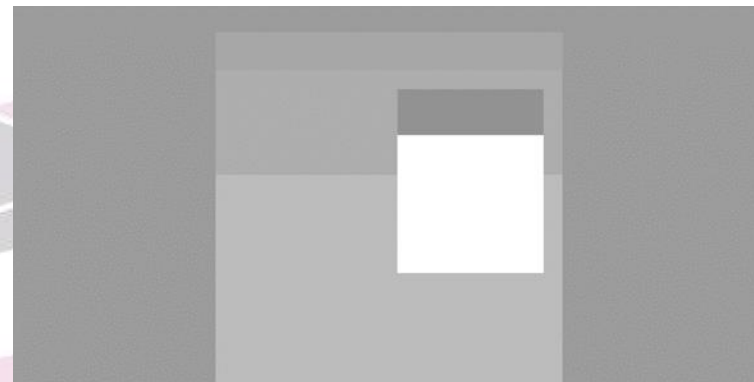


Dropdown Button

Komponen dalam Mendesain Aplikasi Part. 3

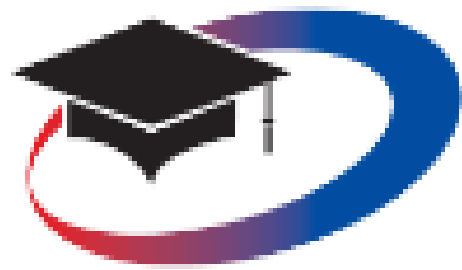


Switch



Dropdown Button

Switch

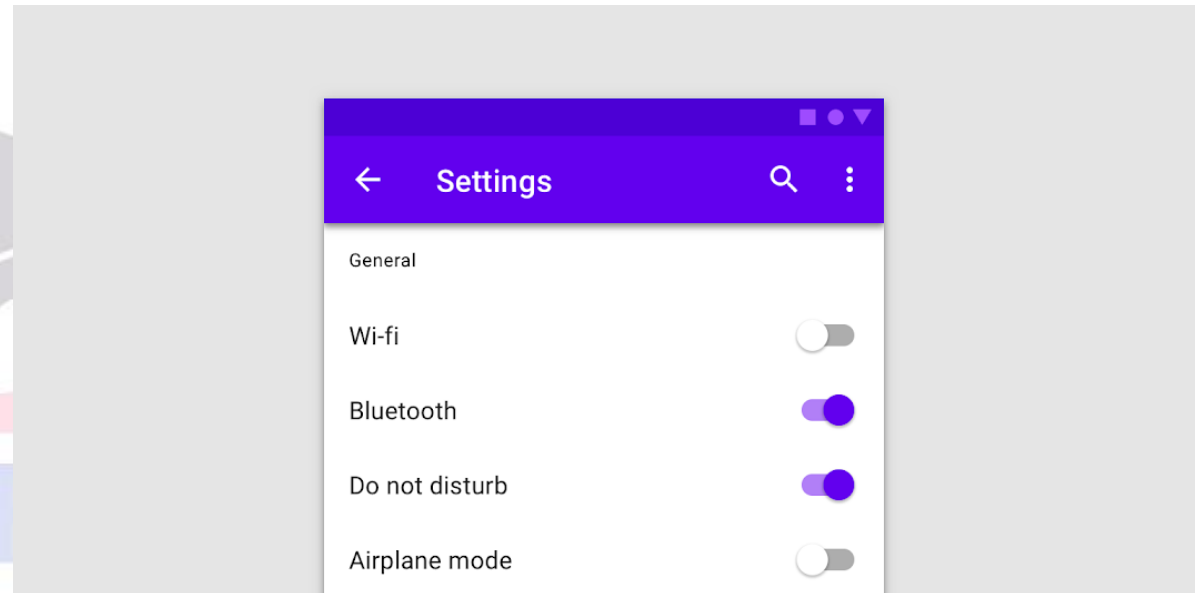


UNIVERSITAS
MIKROSKIL

PRODI. TEKNIK INFORMATIKA (S-1)

Switch adalah...

- Elemen yang memiliki pengaturan status on/off
- User dapat memilih satu status on antar **True** atau **False**
- Cara yang lebih tepat untuk menyesuaikan pengaturan

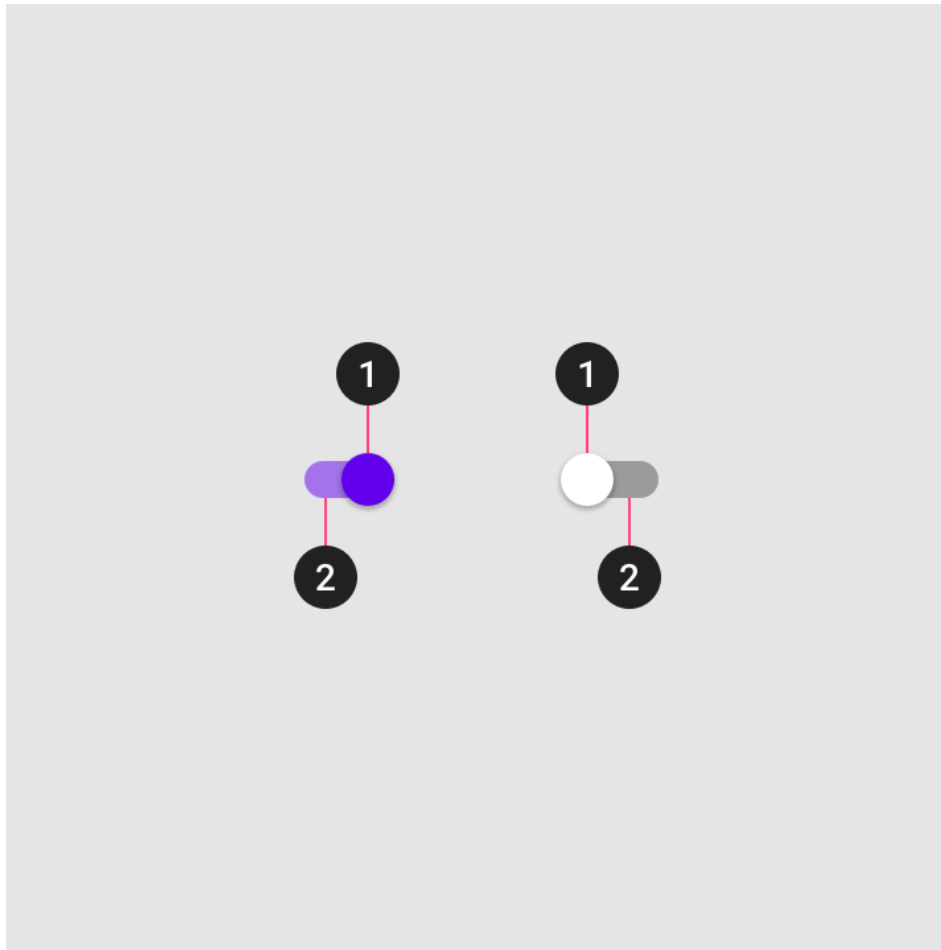


Gunakan checkbox ketika

- Aktifkan atau nonaktifkan satu item, di ponsel dan tablet
- Mengaktifkan atau nonaktifkan sesuatu dengan segera



Anatomi pada Switch



1. Thumb button

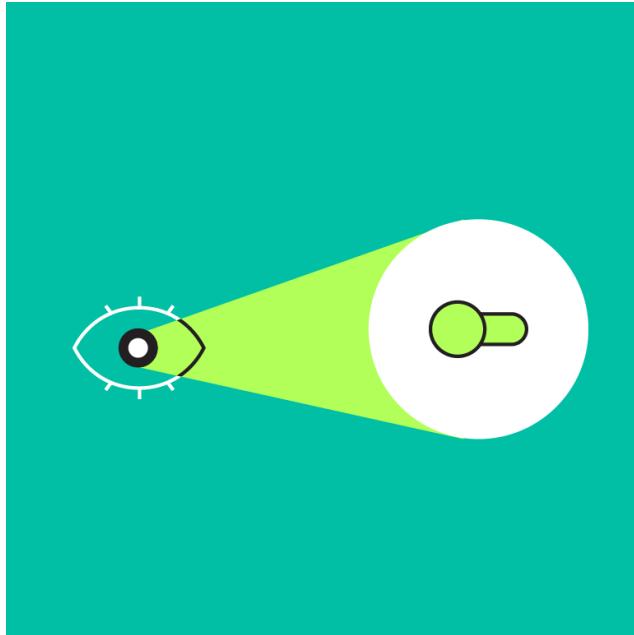
- A. Active Color (Optional)
- C. Inactive Color (Optional)

2. Track button

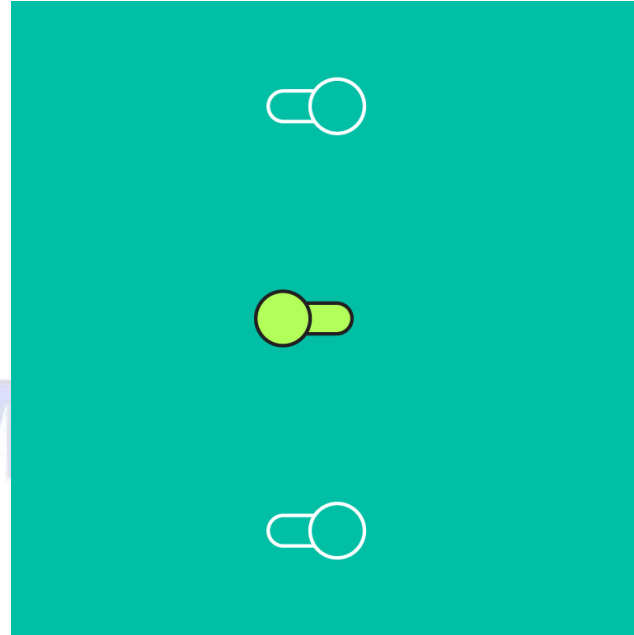
- A. Active Color (Optional)
- B. Inactive Color (Optional)

Prinsiple of Switches

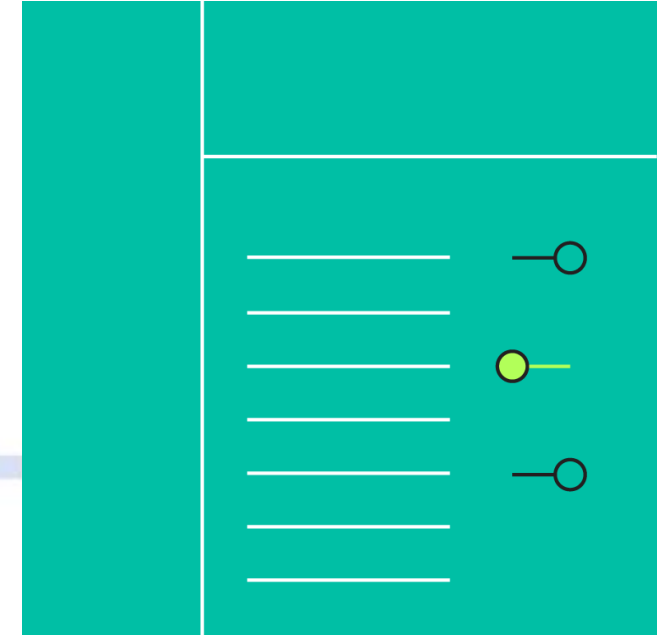
Familiar



Scannable

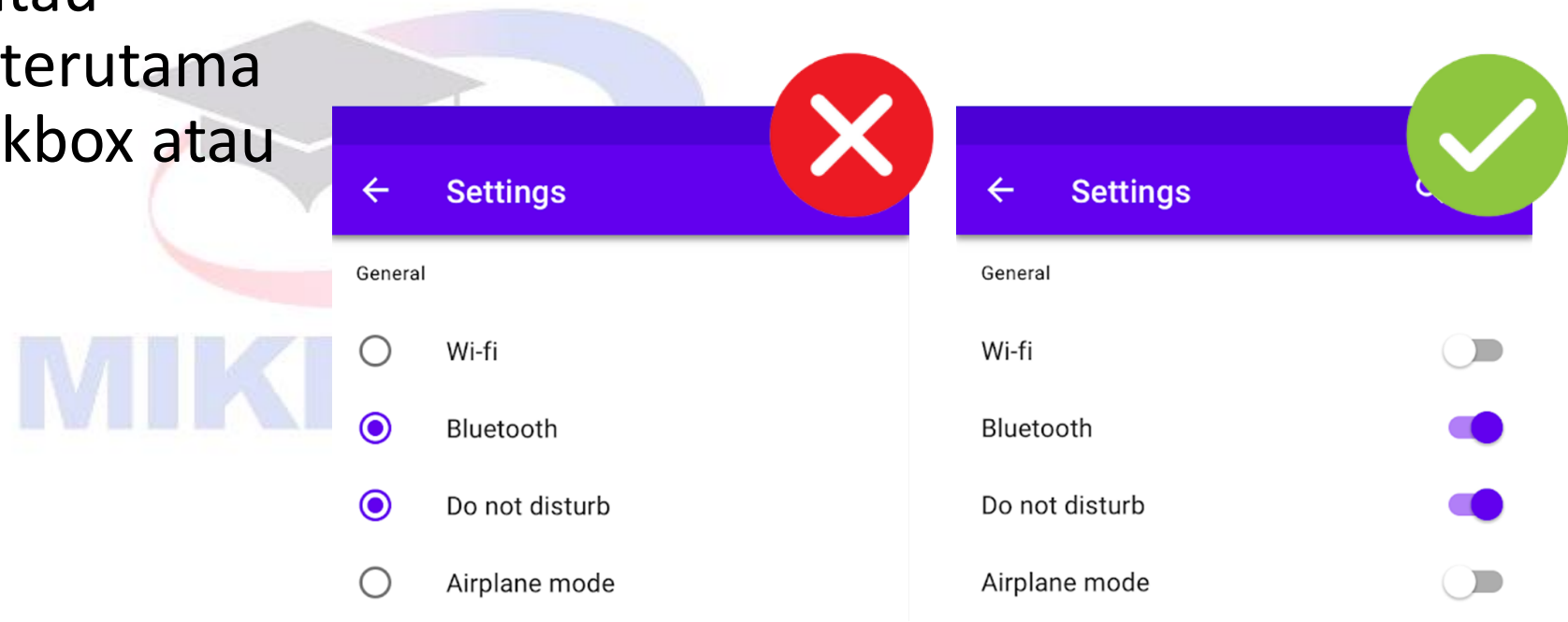


Efficient



Kapan sebaiknya menggunakan switch?

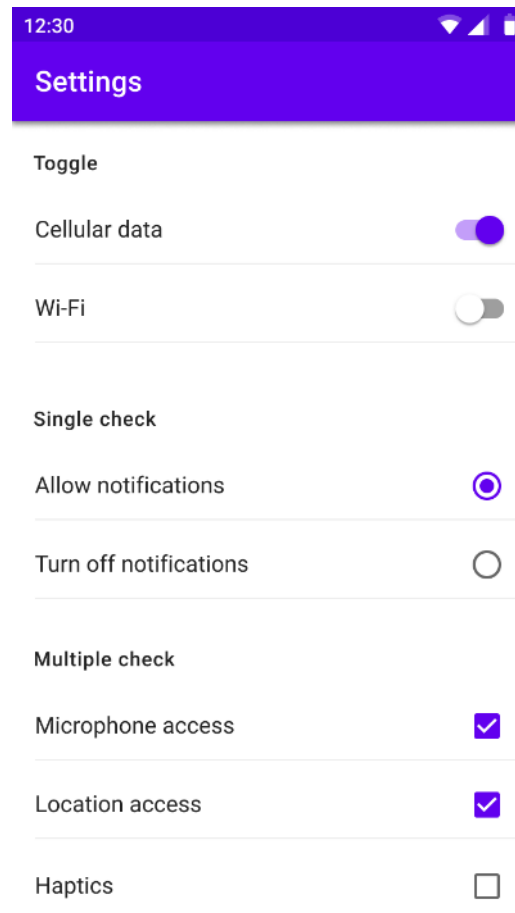
Gunakan Switch Ketika pengguna dapat mengaktifkan atau menonaktifkan item, terutama di seluler, bukan checkbox atau radio button



Platform Differences

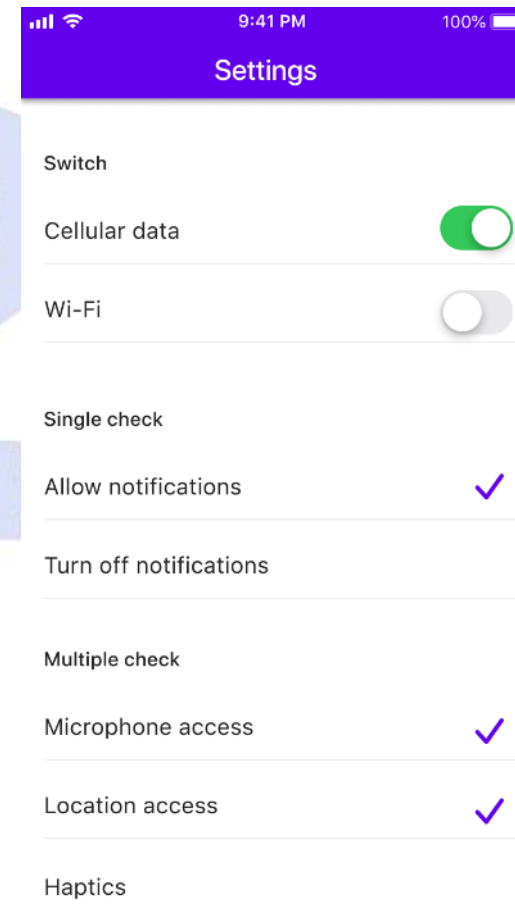
Android

Gunakan Material switches

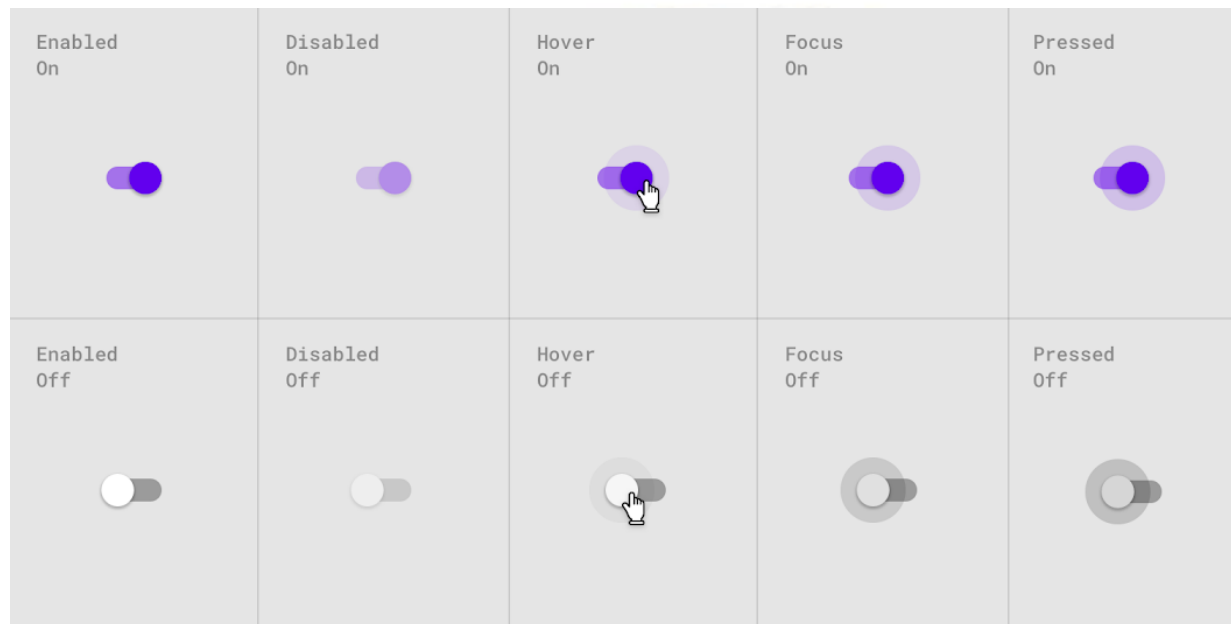


iOS

Material Switch ataupun Cupertino

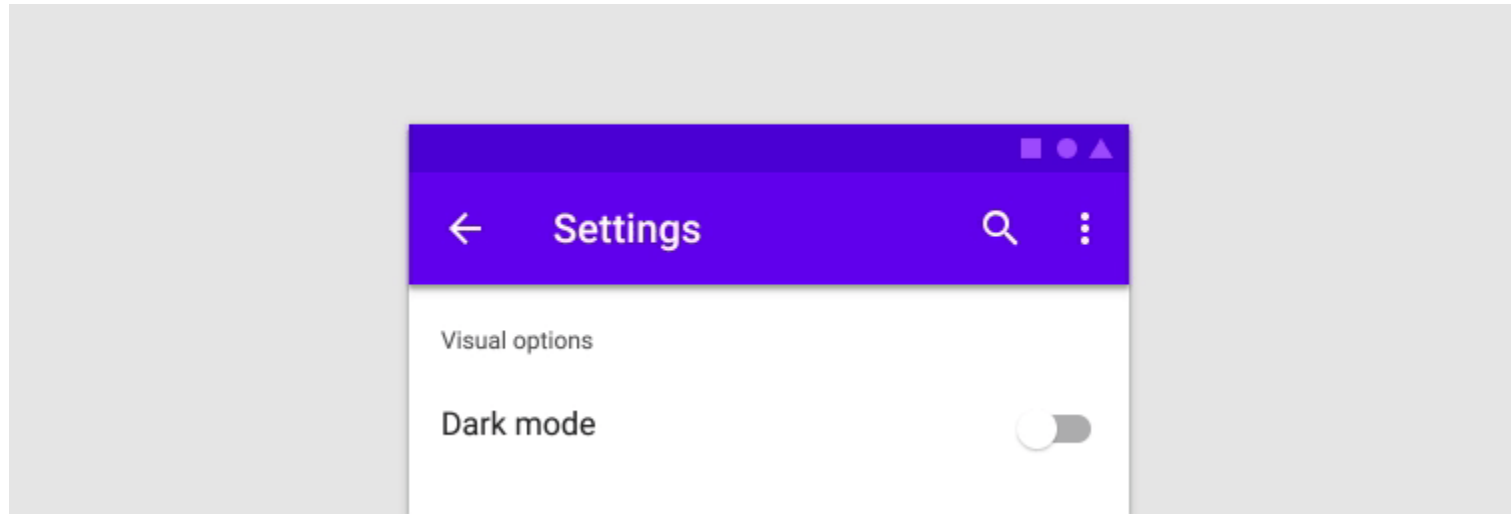


Status di Switch



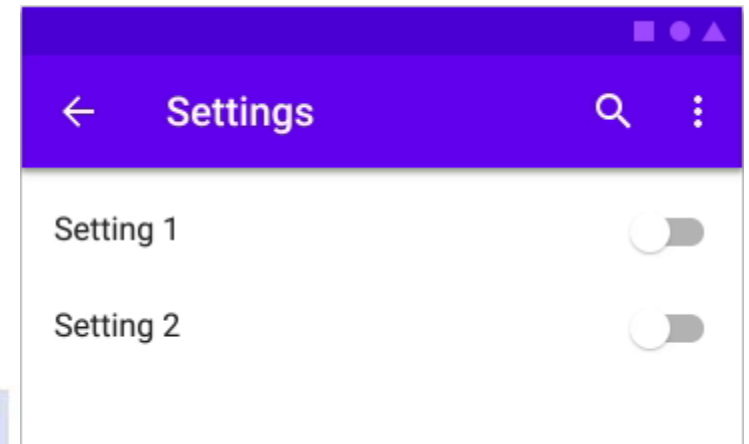
Behavior

Menggunakan sakelar untuk menghidupkan dan mematikan opsi



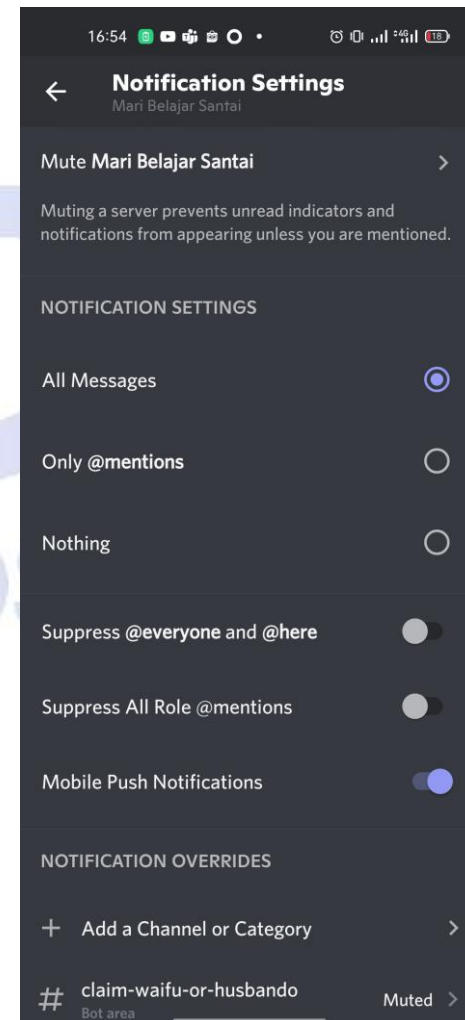
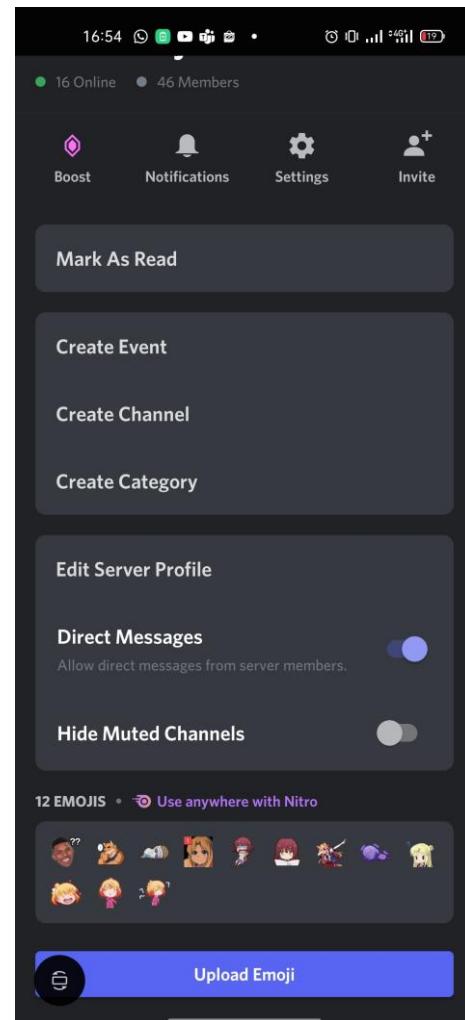
Menampilkan status saat proses

- Switch menunjukkan status sebenarnya dari suatu proses, terkadang ada **penundaan** dalam perubahan statusnya.
- **Animasi** atau **Progress Indicator** dapat digunakan untuk menampilkan status proses tersebut.
- **Jangan** meletakkan animasi status pada **Thumb** switch



Lebih lengkap : [Codesinsider.com](https://codesinsider.com)

Contoh Implementasi Switch

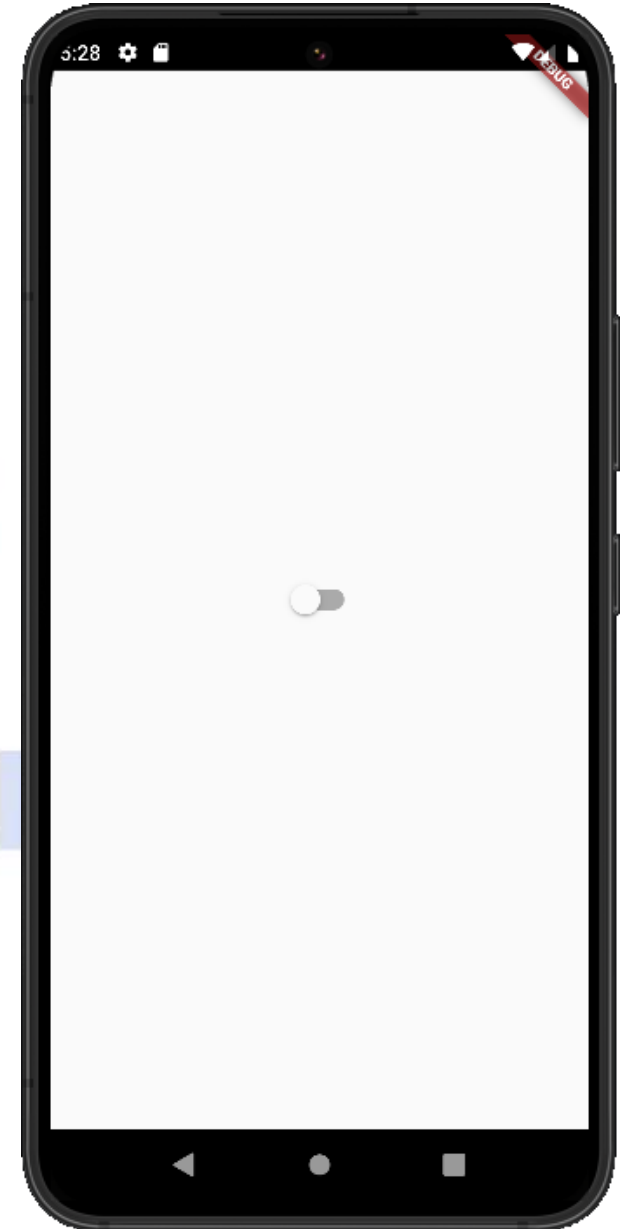


Penerapan di Flutter



Implementasi di Flutter

```
Switch(  
  value: switchVal,  
  onChanged: (bool val) {  
    setState(() {}),  
  },  
)
```



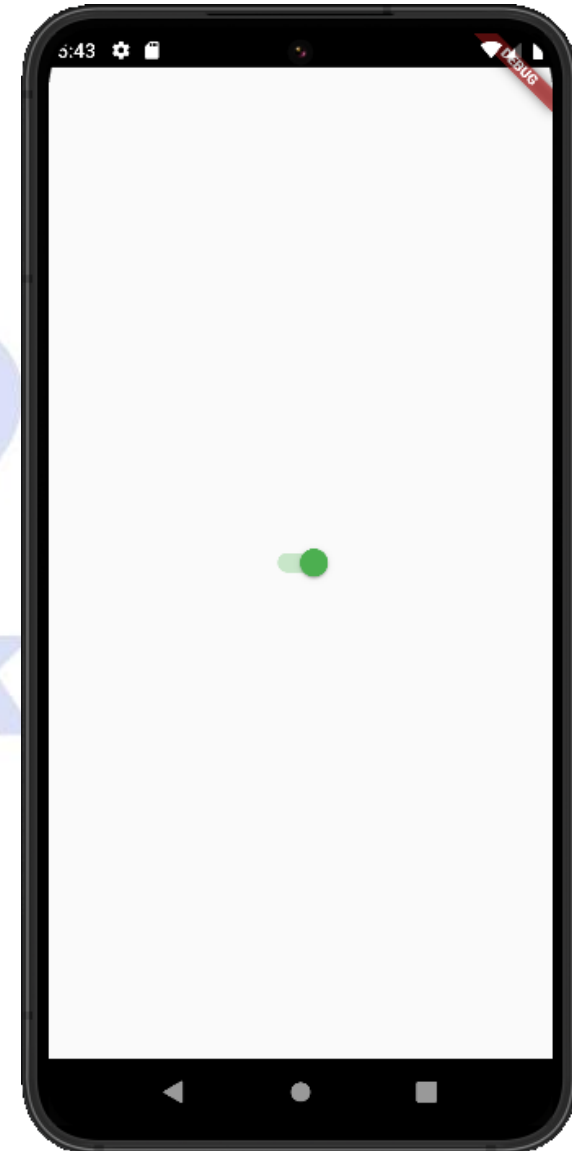
Property di Switch

Property	Class/DataType	Deskripsi
onChanged	ValueChanged<bool>	Fungsi yang dipanggil setiap kali pengguna mengetuk Switch
value	bool	Berisi nilai Boolean benar atau salah untuk mengontrol fungsi Switch ON atau OFF
activeColor	Color	Menentukan warna bola bulat Switch saat ON
activeTrackColor	Color	Menentukan warna bilah track Switch
inactiveThumbColor	Color	Menentukan warna bola bulat Switch saat OFF
inactiveTrackColor	Color	Menentukan warna bilah track sakelar saat OFF
dragStartBehavior	DragStartBehavior	Menangani perilaku drag start. Jika kita mengaturnya sebagai DragStartBehavior.start, drag akan memindahkan sakelar dari aktif ke nonaktif

Property lain dapat dilihat [di sini](#)

Switch

```
class _Apps extends State<Apps> {  
  bool switchVal = false;  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      body: Center(  
        child: Switch(  
          value: switchVal,  
          activeColor: Colors.green,  
          activeTrackColor: Colors.green[100],  
          onChanged: (bool val) {  
            setState(() {  
              switchVal = val;  
              print(switchVal);  
            });  
          },  
        ),  
      ),  
    );  
  }  
}
```



SwitchListTile



Flutter

SwitchListTile

```
class _Apps extends State<Apps> {  
  bool _light = false;  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        backgroundColor: Colors.green,  
        title: const Text("SwitchListTile"),  
        centerTitle: true,  
      ),  
      body: SafeArea(  
        child: SwitchListTile(  
          secondary: Icon(  
            Icons.lightbulb_outline,  
            color: _light ? Colors.green : Colors.grey,  
          ),  
          title: const Text('Lights'),  
          value: _light,  
          activeColor: Colors.green,  
          activeTrackColor: Colors.green[100],  
          onChanged: (bool val) {  
            setState(() {  
              _light = val;  
            });  
          },  
        ),  
      ),  
    );  
  }  
}
```



Property di SwitchListTile

Property	Class/DataType	Deskripsi
value	boolean	Menentukan nilai dari switch (true === checked)
onChanged	ValueChanged<bool>	Mentrigger function Ketika value switch berubah
activeColor	Color	Menentukan warna ketika switch tercentang
checkColor	Color	Menentukan warna pada icon centang
title	Widget	Menambahkan widget ke dalam title dari SwitchListTile (umumnya Text Widget)
selected	boolean	Mengatur apakah kita akan menampilkan warna icon atau text menggunakan activeColor.
subtitle	Widget	Menambahkan widget yang ditampilkan di bawah dari title
secondary	Widget	Menambahkan widget yang posisinya berlawanan dari posisi Switch. (Kalau Switch di sebelah kiri, maka widget secondary berada di sebelah kanan, begitu sebaliknya).

Property lain dapat dilihat [di sini](#) dan [di sini](#)

Contoh menggunakan property title

```
SwitchListTile(  
  title: const Text('Lights'),  
  value: _light,  
  activeColor: Colors.green,  
  activeTrackColor: Colors.green[100],  
  onChanged: (bool val) {  
    setState(() {  
      _light = val;  
    });  
  },  
)
```



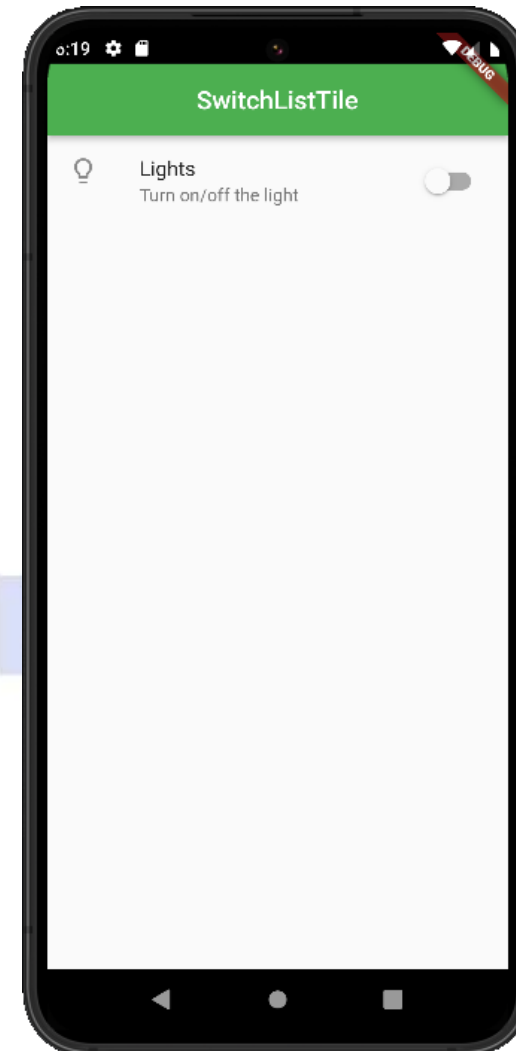
Contoh menggunakan property subtitle

```
SwitchListTile(  
  title: const Text('Lights'),  
  subtitle: const Text("Turn on/off the light"),  
  value: _light,  
  activeColor: Colors.green,  
  activeTrackColor: Colors.green[100],  
  onChanged: (bool val) {  
    setState(() {  
      _light = val;  
    });  
  },  
),
```



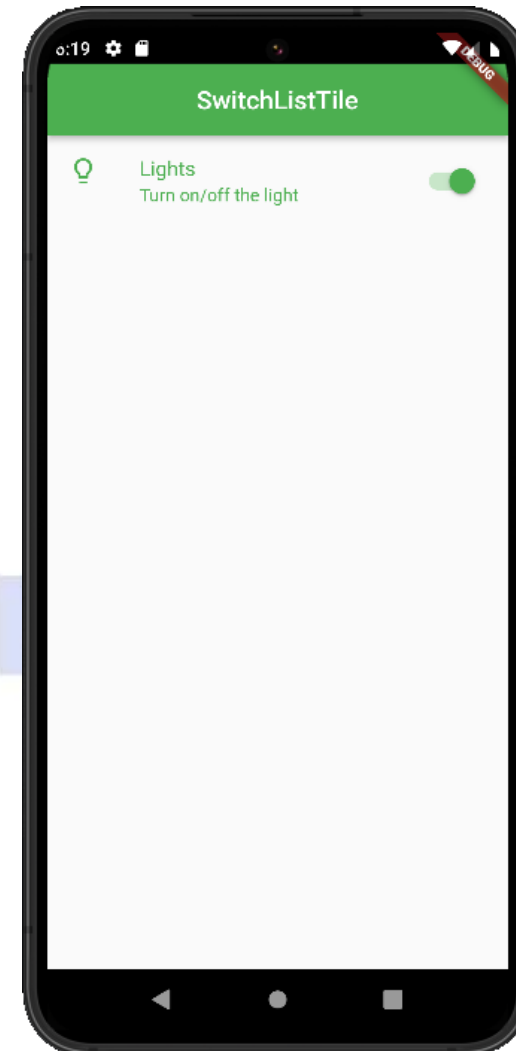
Contoh menggunakan property secondary

```
SwitchListTile(  
  secondary: const Icon(Icons.lightbulb_outline),  
  title: const Text('Lights'),  
  subtitle: const Text("Turn on/off the light"),  
  value: _light,  
  activeColor: Colors.green,  
  activeTrackColor: Colors.green[100],  
  onChanged: (bool val) {  
    setState(() {  
      _light = val;  
    });  
  },  
)
```



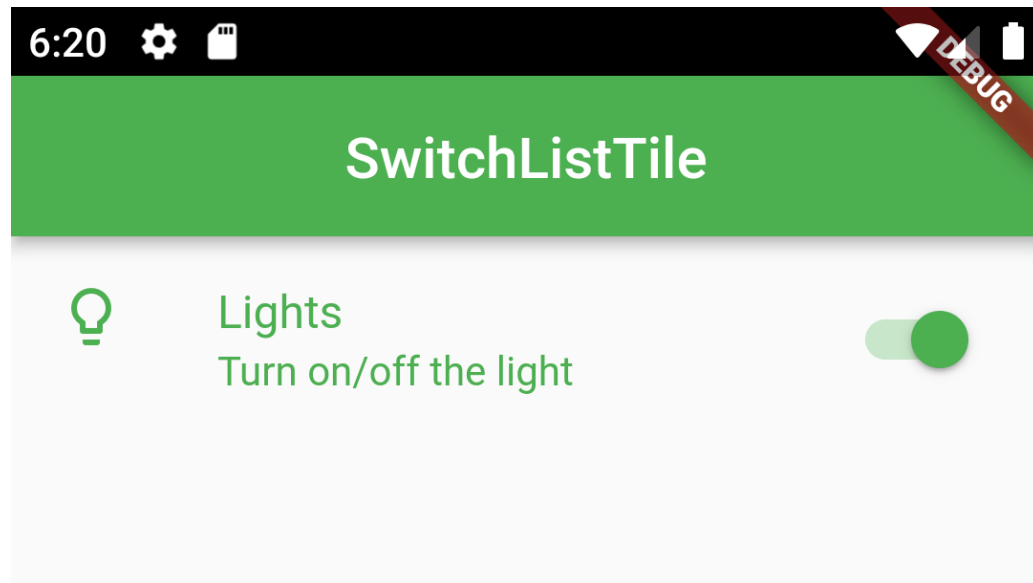
Contoh menggunakan property selected

```
SwitchListTile(  
  secondary: const Icon(Icons.lightbulb_outline),  
  title: const Text('Lights'),  
  subtitle: const Text("Turn on/off the light"),  
  value: _light,  
  selected: _light,  
  activeColor: Colors.green,  
  activeTrackColor: Colors.green[100],  
  onChanged: (bool val) {  
    setState(() {  
      _light = val;  
    });  
  },  
)
```

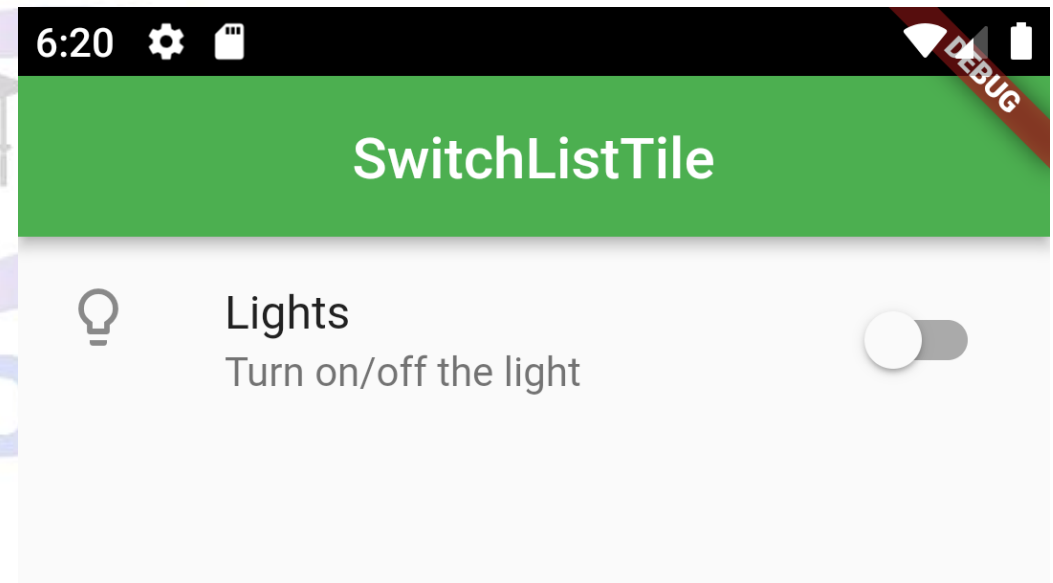


Contoh menggunakan property selected

Ketika switch on

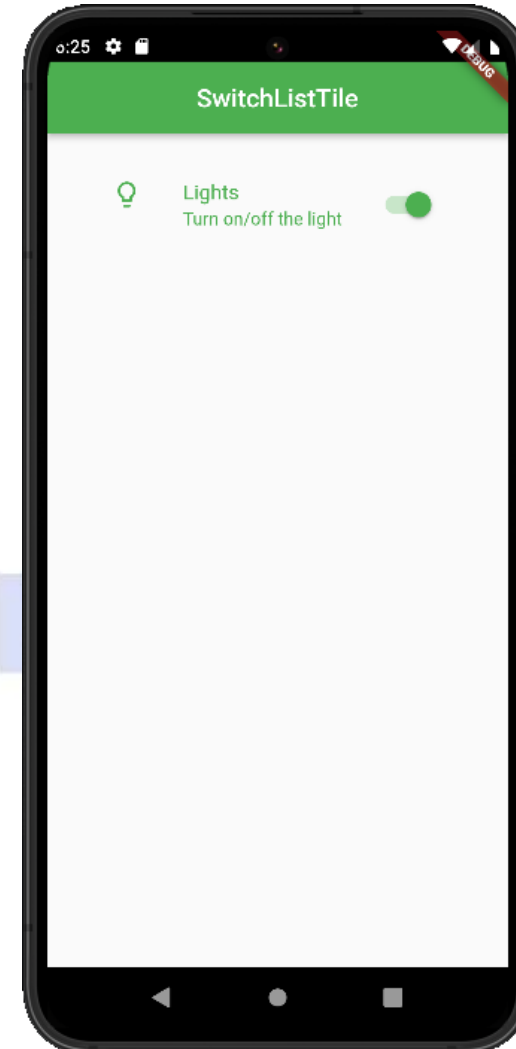


Ketika switch off



Contoh menggunakan property **contentPadding**

```
SwitchListTile(  
  contentPadding: const EdgeInsets.symmetric(  
    vertical: 20,  
    horizontal: 50,  
  ),  
  secondary: const Icon(Icons.lightbulb_outline),  
  title: const Text('Lights'),  
  subtitle: const Text("Turn on/off the light"),  
  value: _light,  
  selected: _light,  
  activeColor: Colors.green,  
  activeTrackColor: Colors.green[100],  
  onChanged: (bool val) {  
    setState(() {  
      _light = val;  
    });  
  },  
)
```



controlAffinity

controlAffinity digunakan untuk mengontrol posisi **Switch** terhadap teks. Dibutuhkan sebuah kelas **ListTileControlAffinity** yang memiliki 3 constants, yaitu:

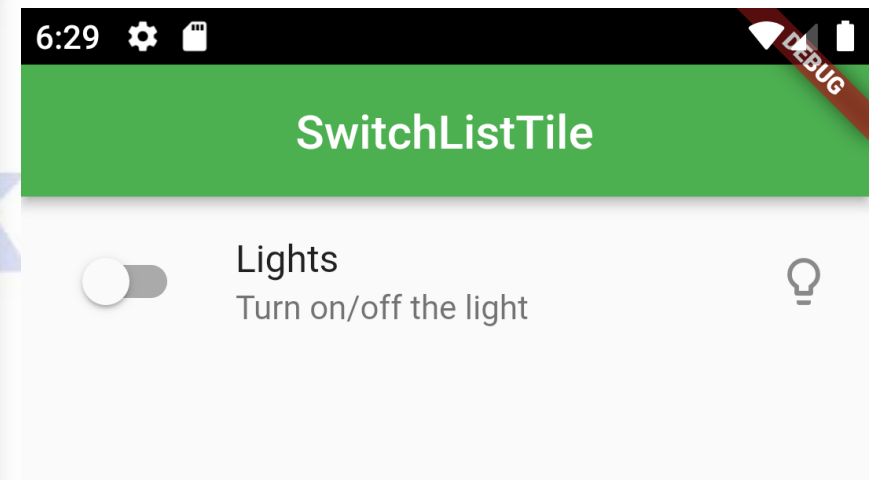
- **leading**
- **trailing**
- **Platform** (Menyesuaikan standard dari platform os)



ListTileControlAffinity.leading

Posisi Switch di sebelah kiri, teks di tengah dan *secondary* berada di sebelah kanan.

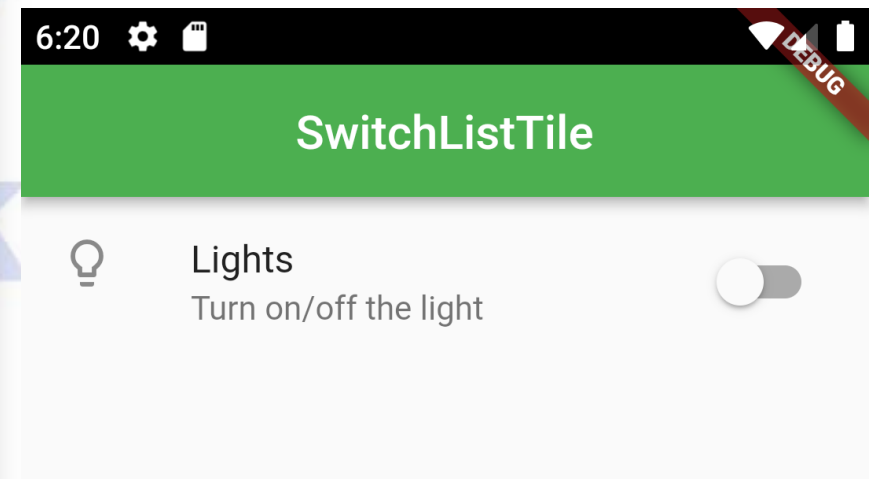
```
SwitchListTile(  
  controlAffinity: ListTileControlAffinity.leading,  
  secondary: const Icon(Icons.lightbulb_outline),  
  title: const Text('Lights'),  
  subtitle: const Text("Turn on/off the light"),  
  value: _light,  
  selected: _light,  
  activeColor: Colors.green,  
  activeTrackColor: Colors.green[100],  
  onChanged: (bool val) {  
    setState(() {  
      _light = val;  
    });  
  },  
)
```



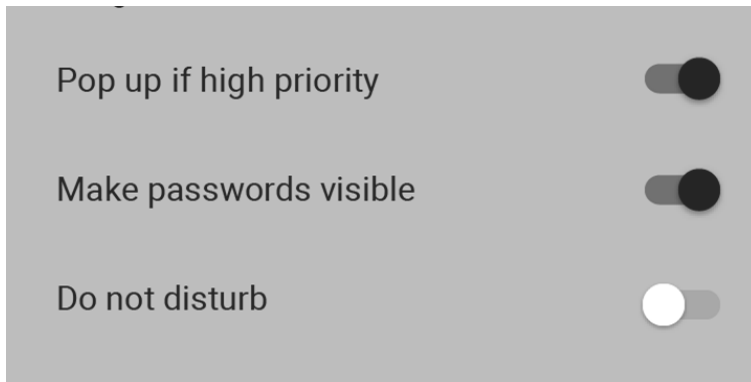
ListTileControlAffinity.trailing

Posisi Switch di sebelah kanan, teks di tengah dan *secondary* berada di sebelah kiri.

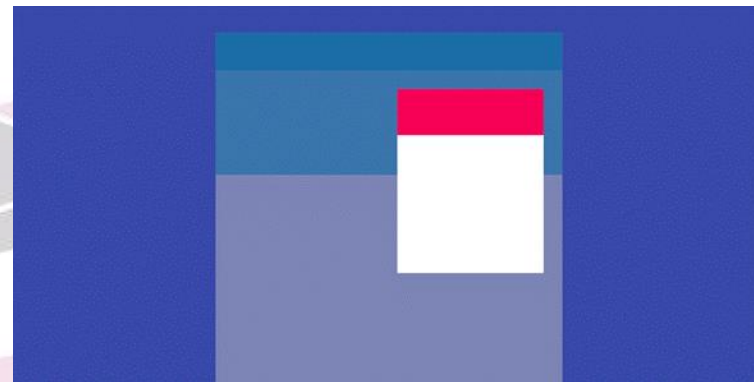
```
SwitchListTile(  
  controlAffinity: ListTileControlAffinity.trailing,  
  secondary: const Icon(Icons.lightbulb_outline),  
  title: const Text('Lights'),  
  subtitle: const Text("Turn on/off the light"),  
  value: _light,  
  selected: _light,  
  activeColor: Colors.green,  
  activeTrackColor: Colors.green[100],  
  onChanged: (bool val) {  
    setState(() {  
      _light = val;  
    });  
  },  
)
```



Komponen dalam Mendesain Aplikasi Part. 3



Switch



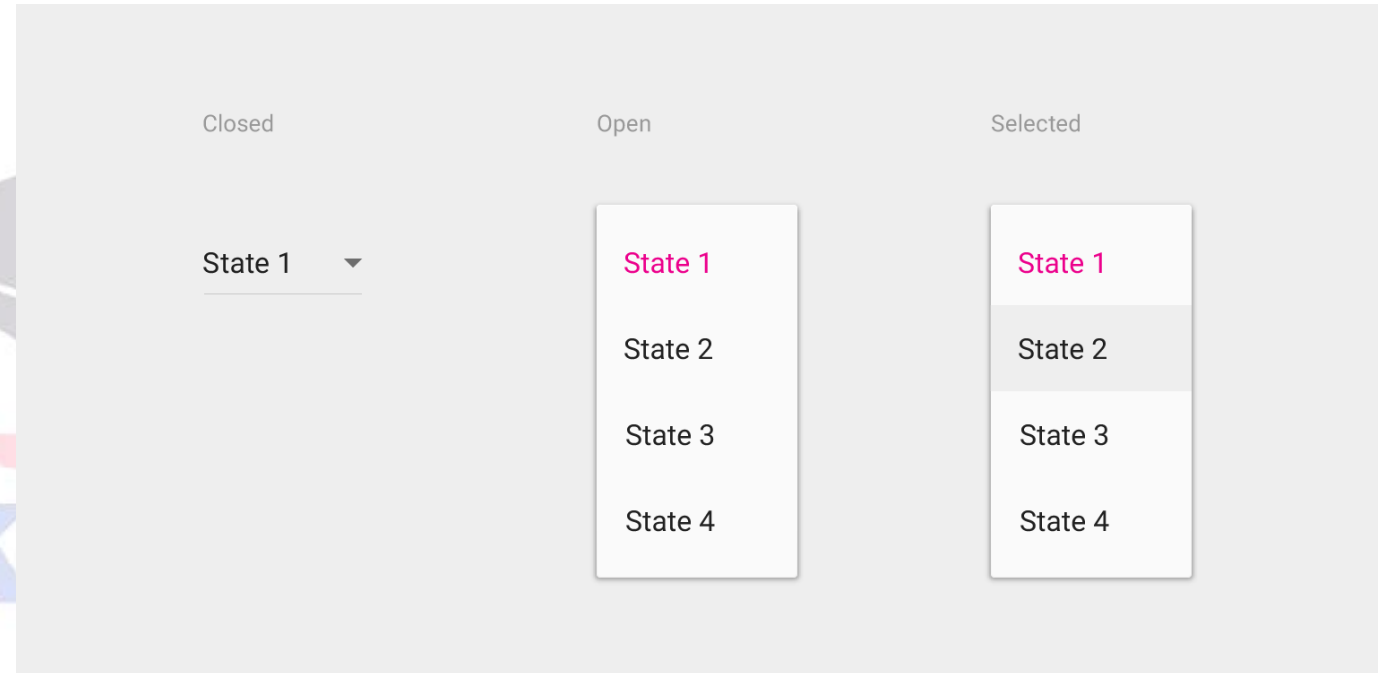
Dropdown Button

Dropdown Button



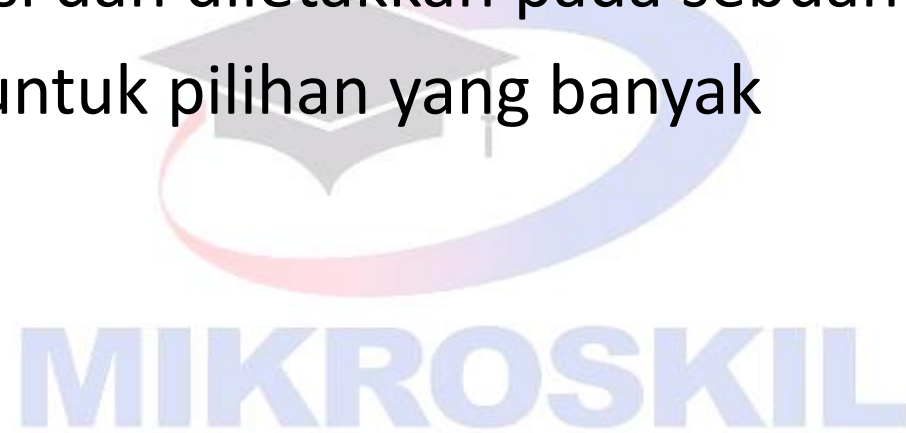
Dropdown Button adalah...

- Tombol yang memiliki sekumpulan opsi unik yang dapat dipilih salah satu
- Menampilkan icon panah bawah
- Mirip seperti Select pada HTML

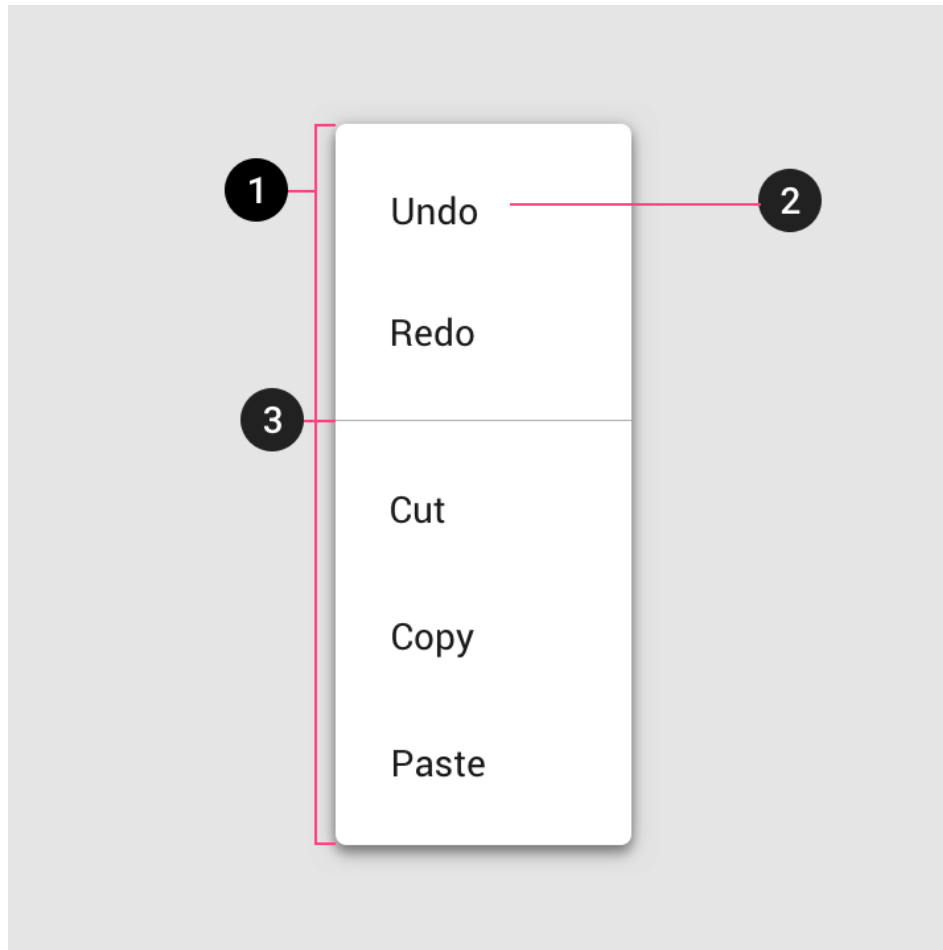


Gunakan Dropdown Button ketika

- User dapat memilih satu opsi dari sebuah list
- Memiliki banyak opsi dan diletakkan pada sebuah button
- Mengurangi space untuk pilihan yang banyak



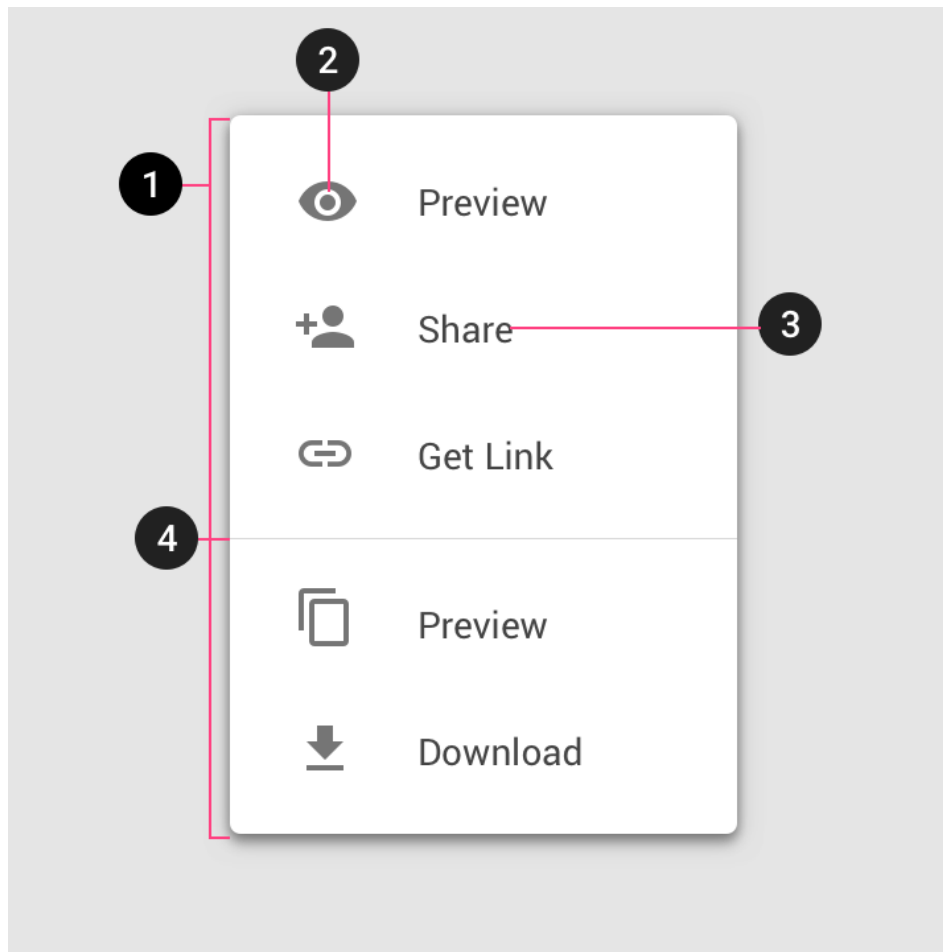
Anatomi pada Dropdown Button



Text list

1. Container
2. Text
3. Divider

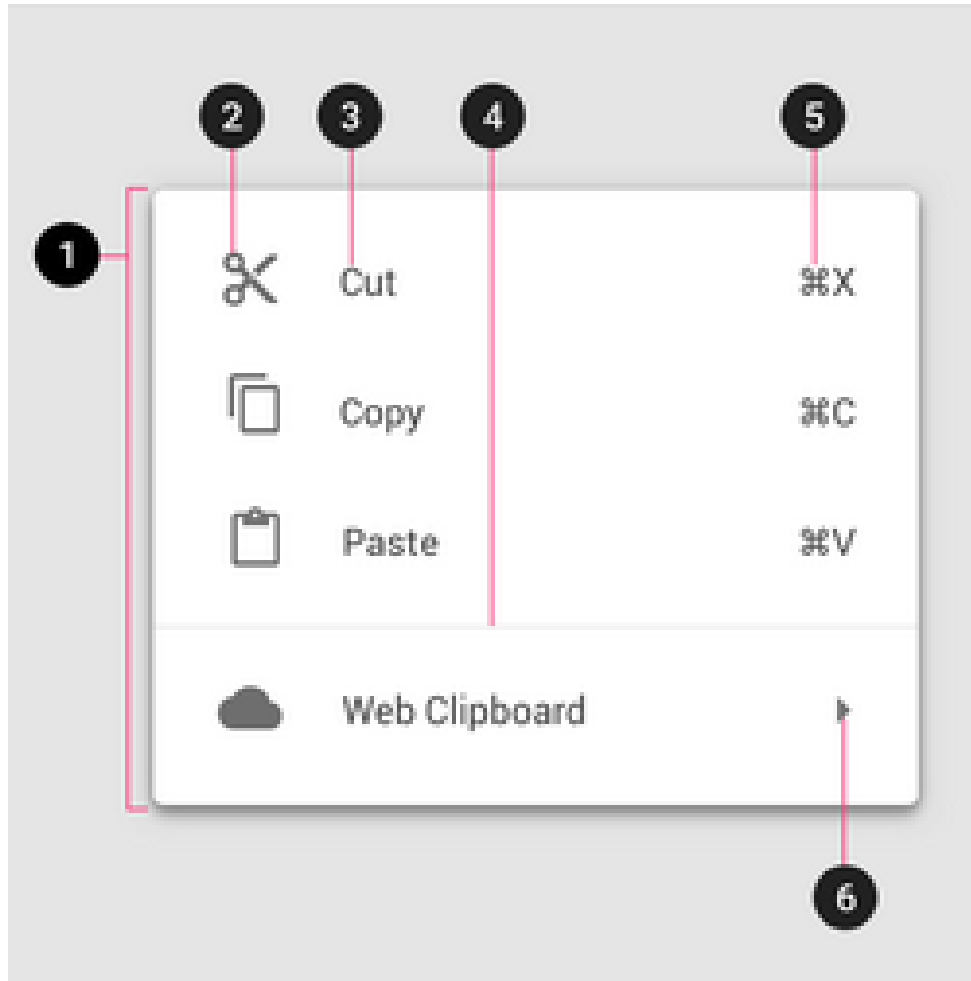
Anatomi pada Dropdown Button



Text and icon list

1. Container
2. Leading icon
3. Text
4. Divider

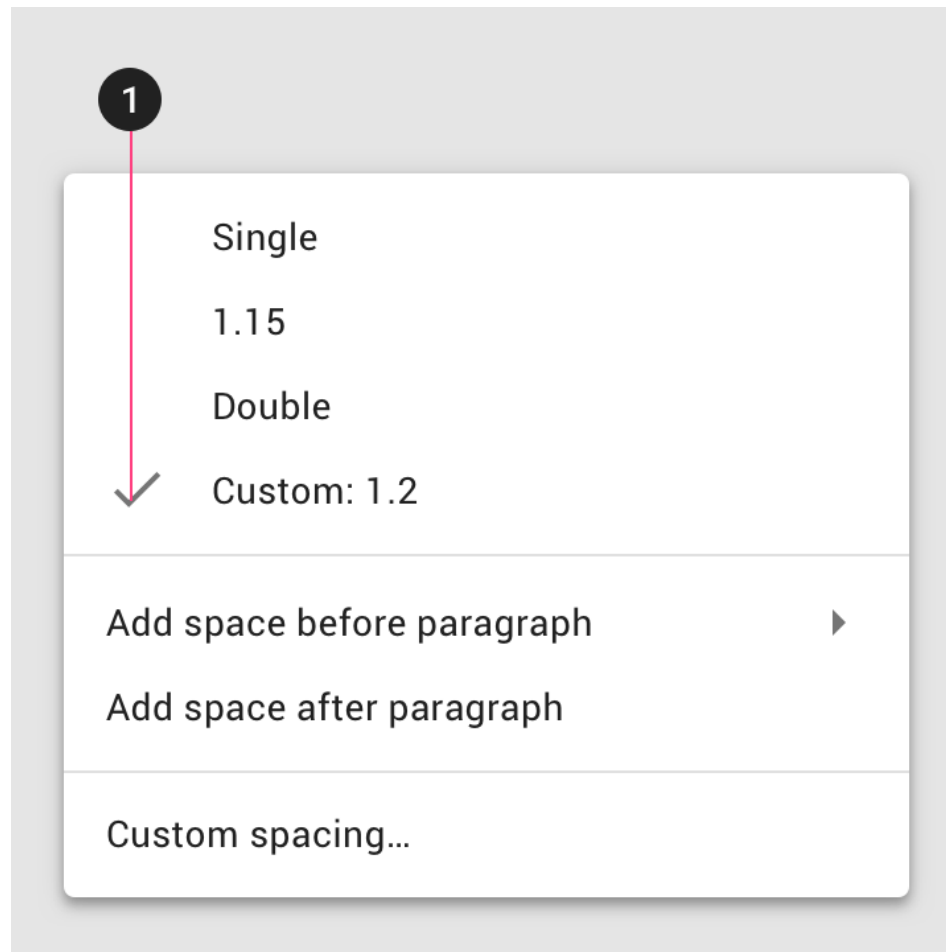
Anatomi pada Dropdown Button



Text, icon, and keyboard command list

1. Container
2. Leading icon
3. Text
4. Divider
5. Command
6. Cascading menu indicator

Anatomi pada Dropdown Button



Text with selection state list

1. Selection state

Action pada Menu atau Dropdown Button

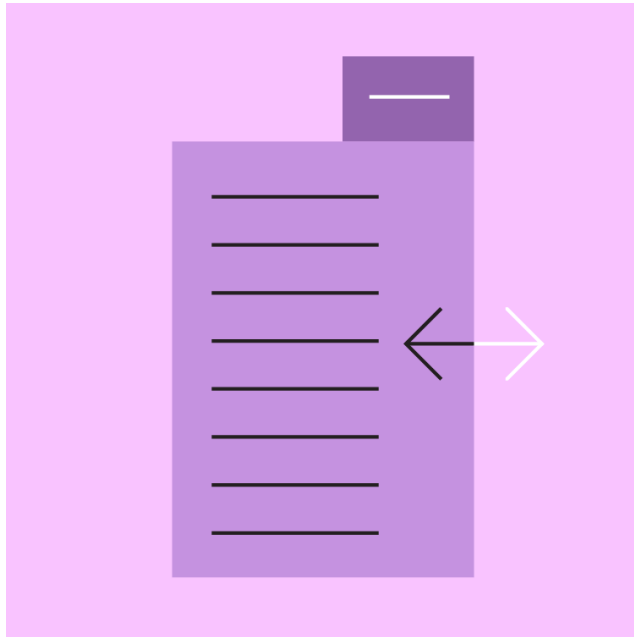


Disabled

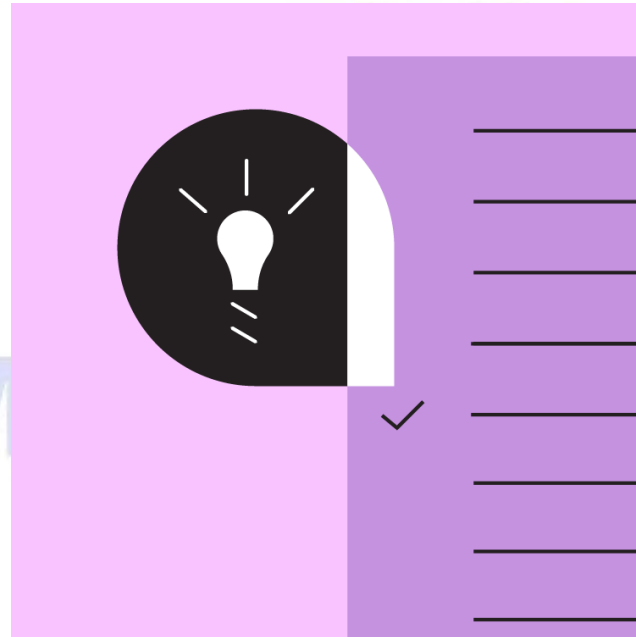
- Tampilan list dari dropdown dapat dinonaktifkan (disabled) dan digunakan pada kondisi tertentu.
- List tersebut biasanya harus ditampilkan secara disable daripada dihapus.

Prinsiple of Dropdown button

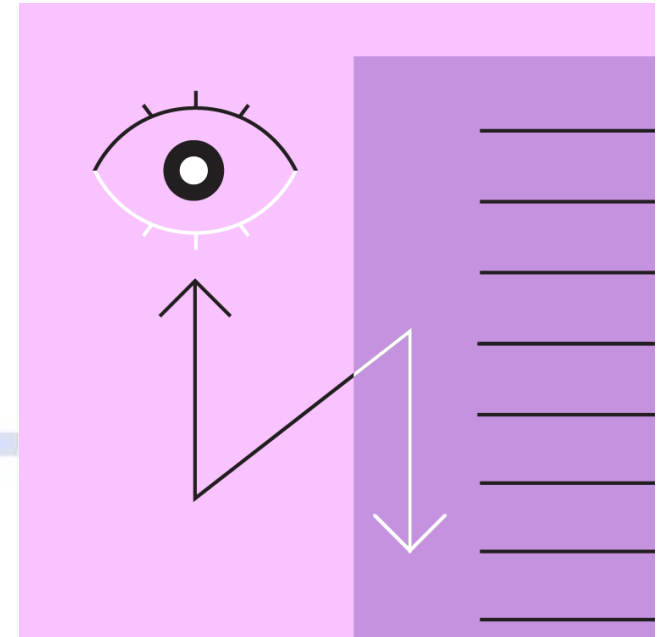
Nimble



Contextual



Scannable



Kapan sebaiknya menggunakan dropdown button?

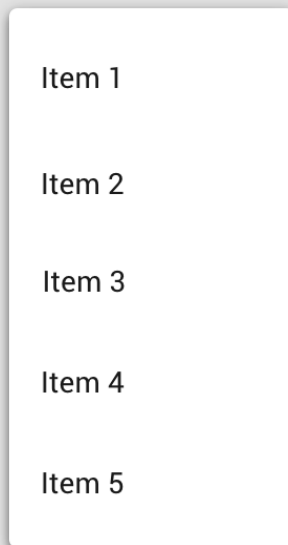
Sebuah menu yang menampilkan daftar pilihan pada tampilan sementara.

Tampilan list muncul saat pengguna berinteraksi dengan tombol, tindakan, atau kontrol lainnya.

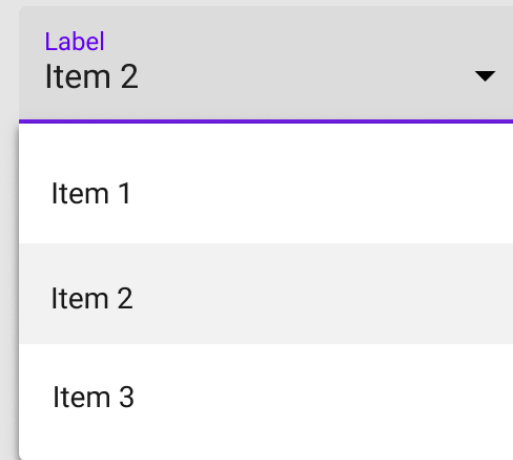


Jenis Menu

Dropdown Menus

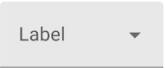
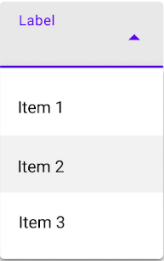
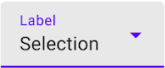
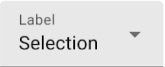
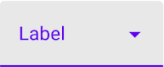
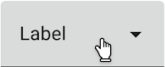

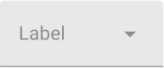
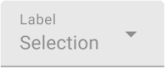


Dropdown Button


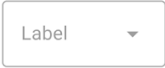



States di Dropdown Button

Filled Dropdown Button

Inactive	Activated	Populated active
		
Populated inactive	Focused	Hover
		
Error	Disabled	Populated disabled
		

Outlined Dropdown Button

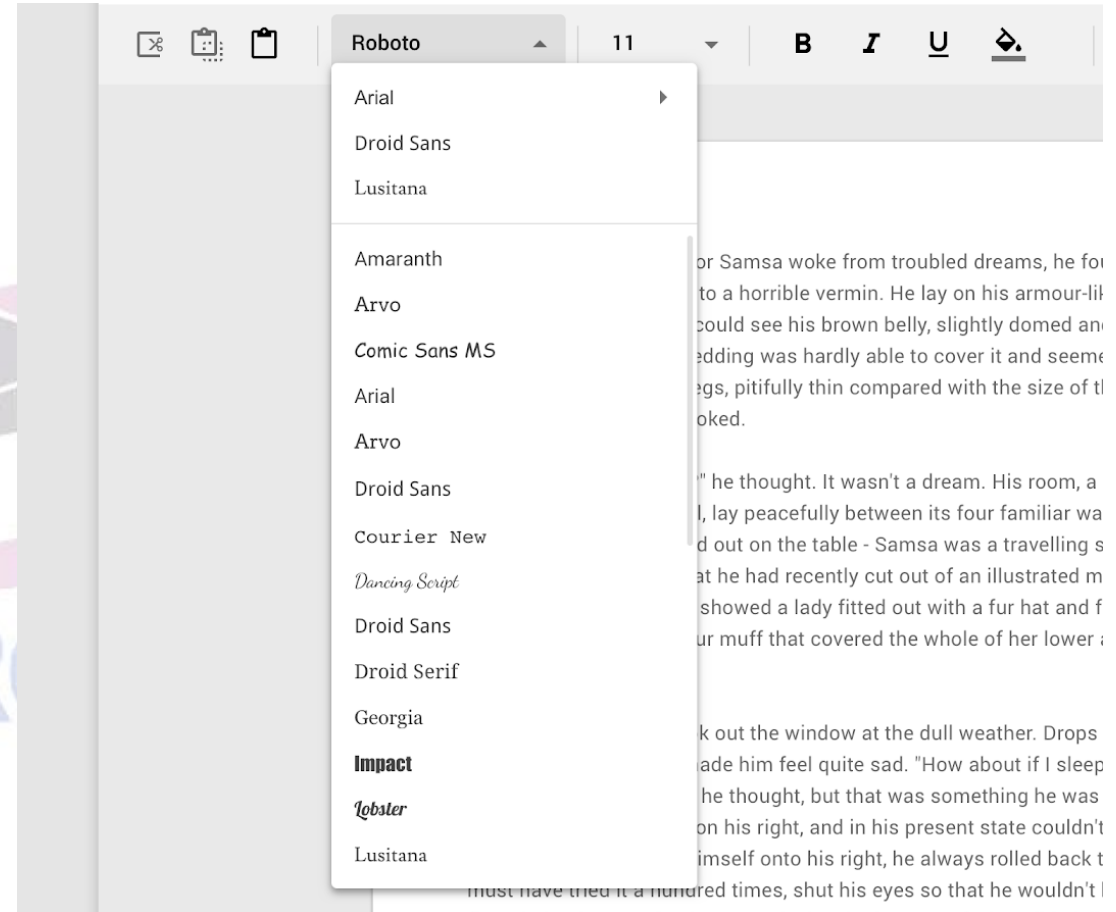
Inactive	Activated	Populated active
		
Populated inactive	Focused	Hover
		
Error	Disabled	Populated disabled
		

Behavior

Scrolling

Jika tidak semua item menu ditampilkan sekaligus, menu dapat scroll.

MIKRO

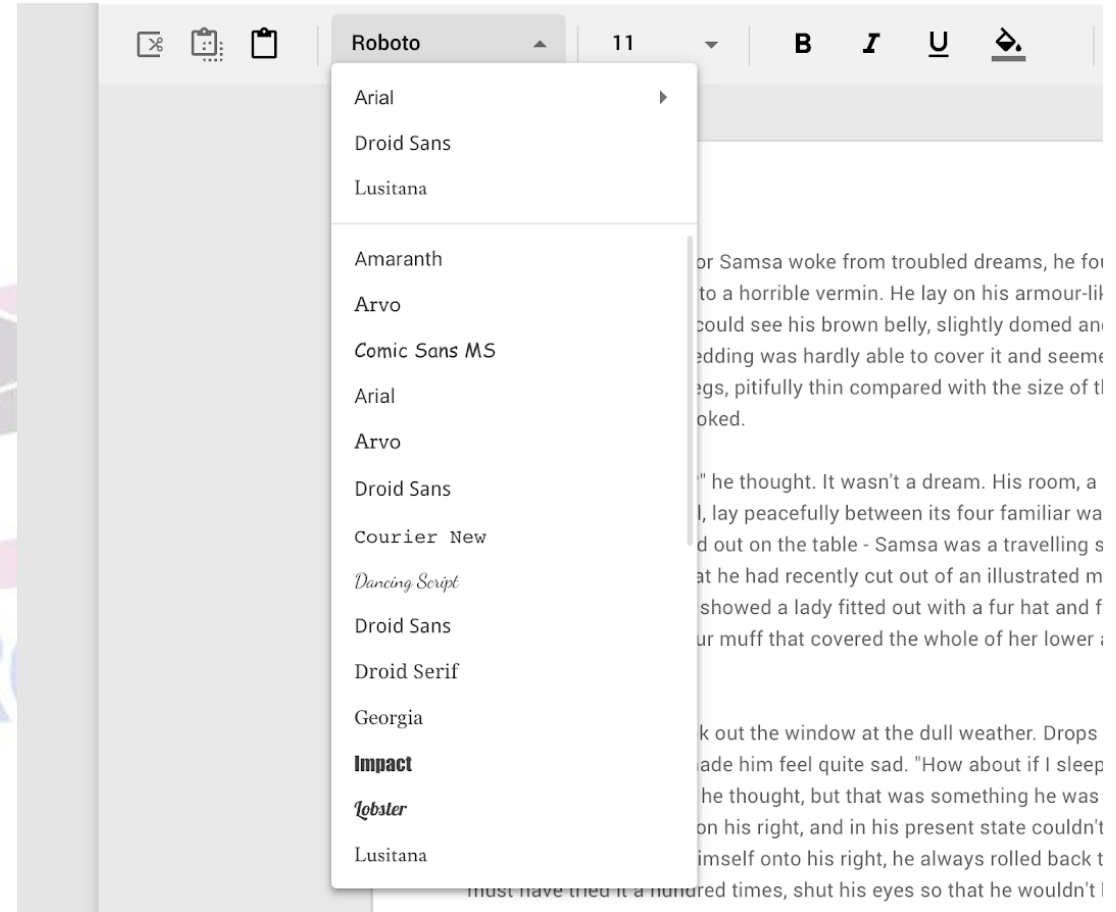


Behavior

Transitions

Menu menggunakan pola transisi pudar untuk masuk dan keluar layar.

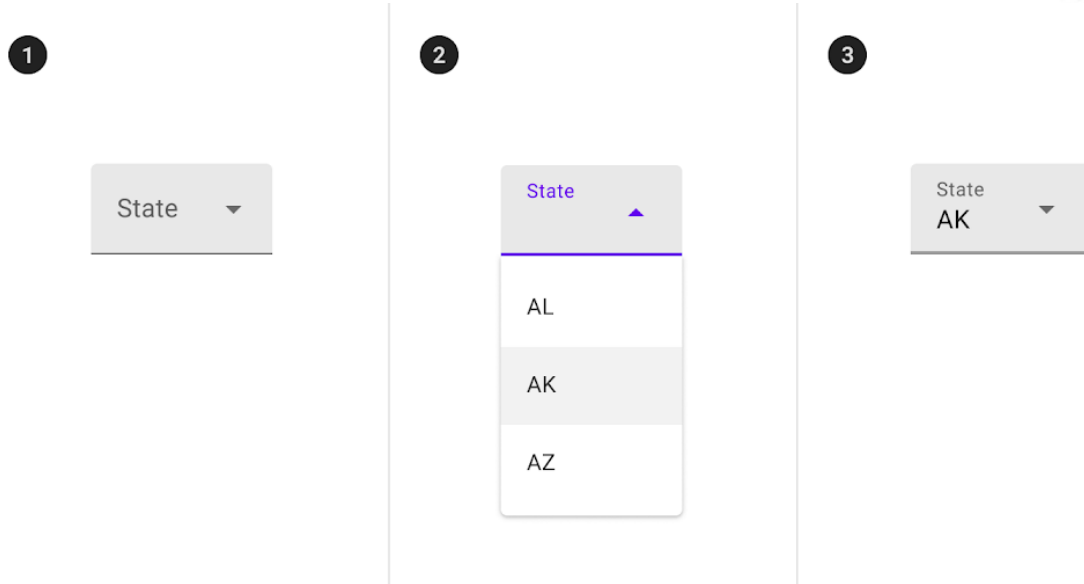
Animasi menciptakan hubungan antara menu dan tindakan yang menghasilkan menu.



Behavior pada Dropdown Button

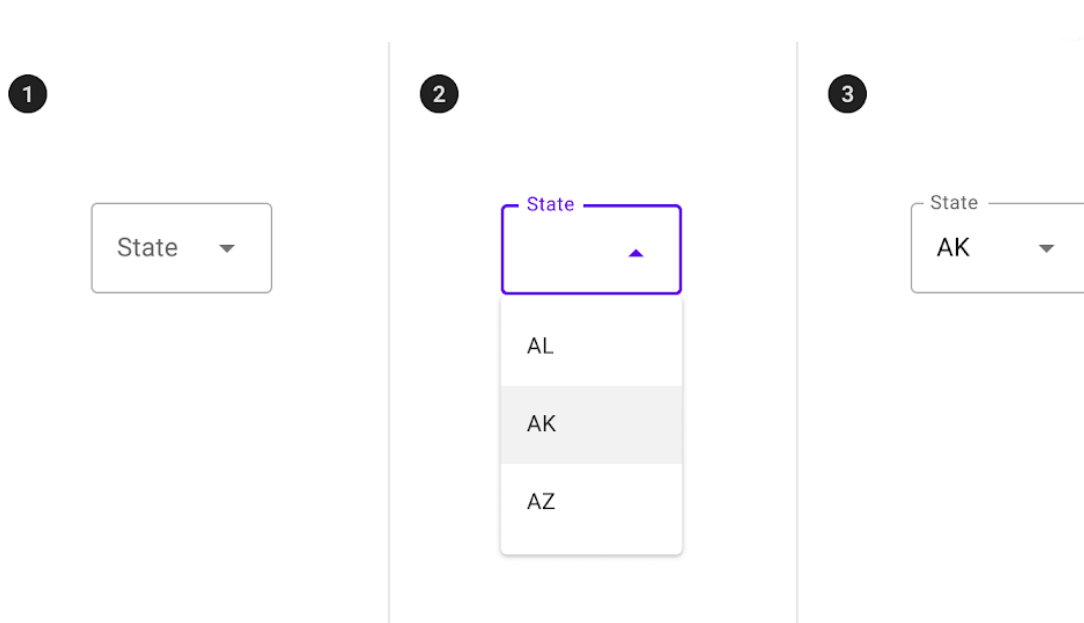
Filled Dropdown Button

1. Inactive
2. Activated
3. Inactive and populated



Behavior pada Dropdown Button

Outlined Dropdown Button

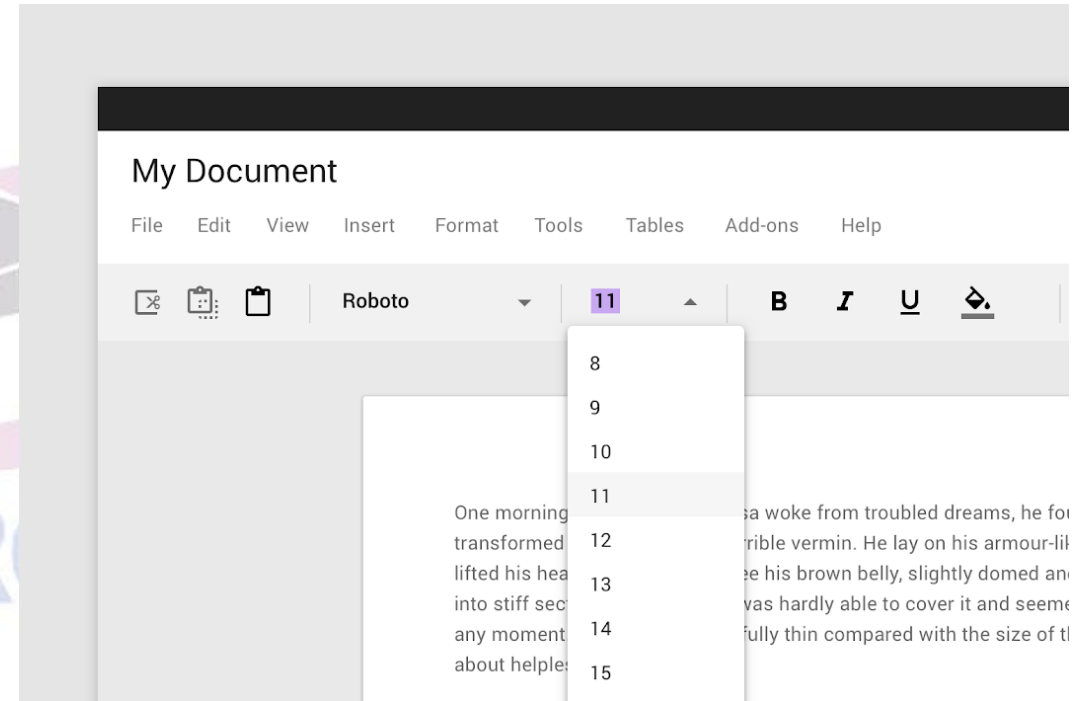


1. Inactive
2. Activated
3. Inactive and populated

OSKIL

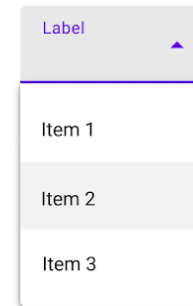
Editable Dropdown Button

- Menu **dropdown** yang terpilih dapat diedit ditampilkan di atas.
- Pengguna dapat memasukkan nilai yang tidak tercantum dalam menu.
- Nilai input dapat dibatasi dengan type data tertentu.



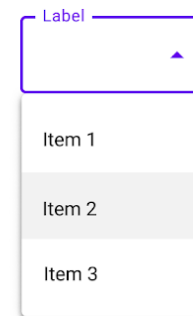
Filled Dropdown Button

- Terlihat seperti bidang teks yang terisi
- Menampilkan item yang dipilih di atas
- Pilihan untuk user terbatas sesuai dengan yang tersedia

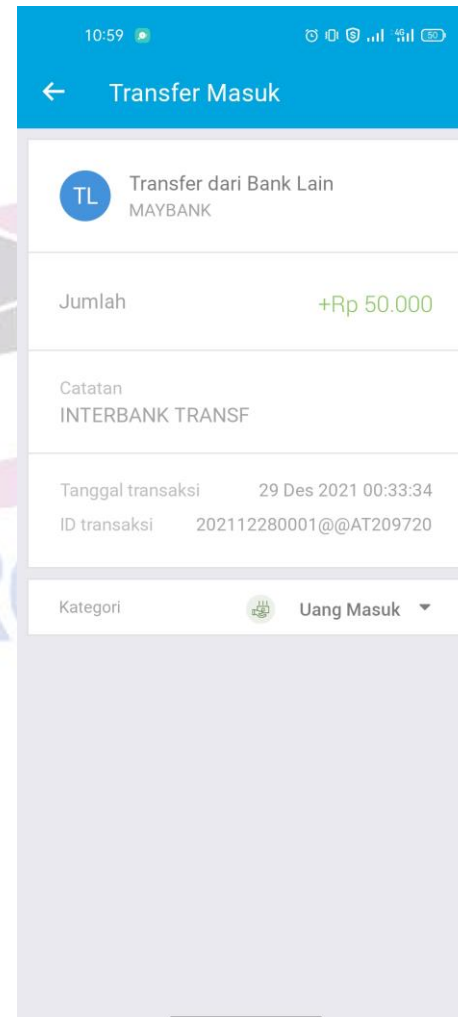
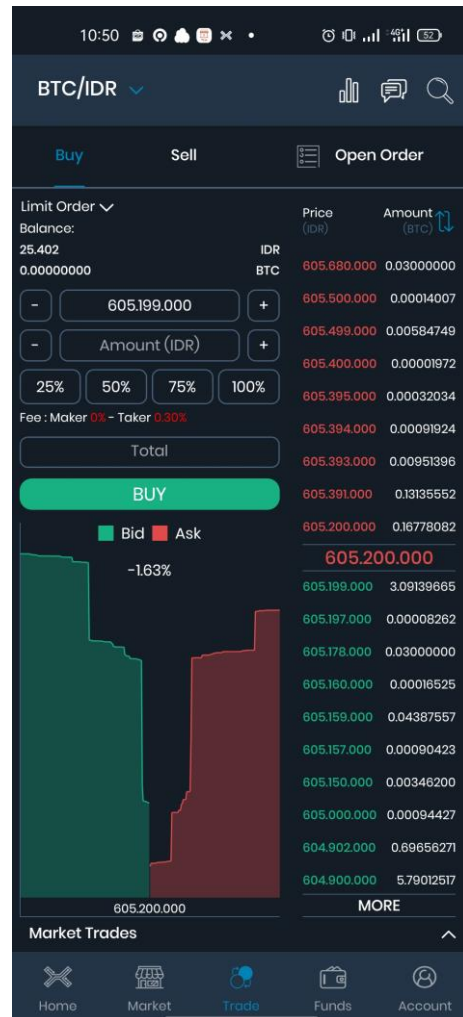


Outlined Dropdown Button

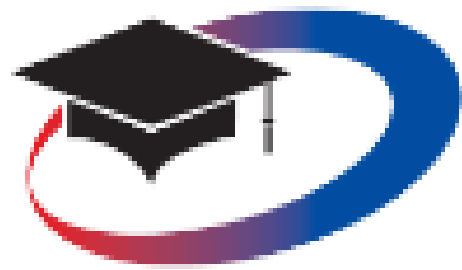
- Tampilan seperti bidang teks kosong dan memiliki border sekelilingnya
- Menampilkan item yang dipilih di atas
- Pilihan untuk user terbatas sesuai dengan yang tersedia



Contoh Implementasi Dropdown Button



Penerapan di Flutter

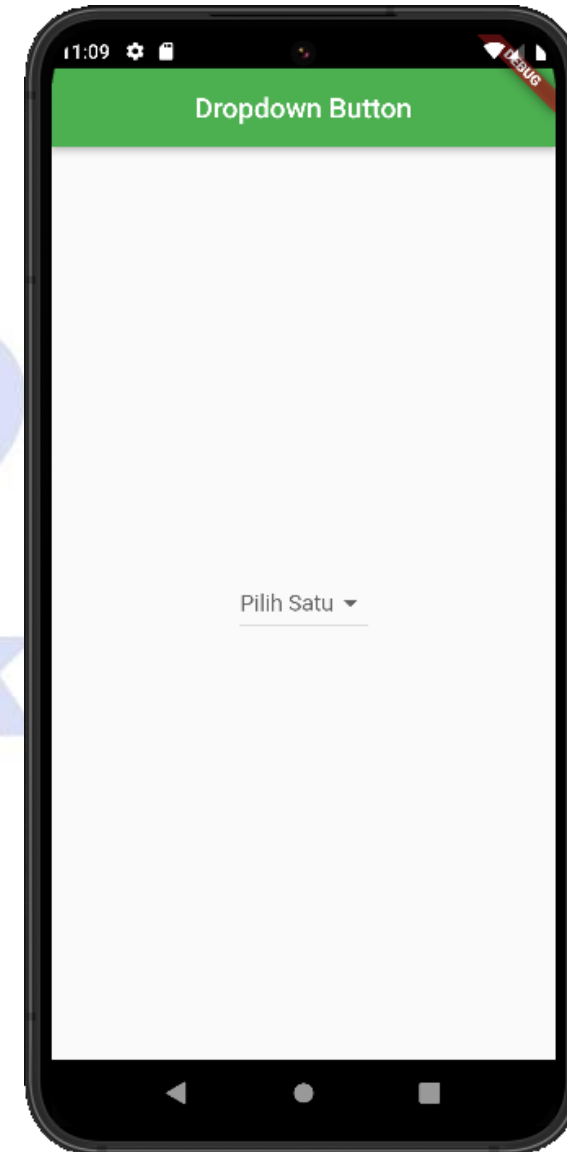


UNIVERSITAS
MIKROSKIL

PRODI. TEKNIK INFORMATIKA (S-1)

Dropdown

```
DropdownButton(  
  value: _dropdownValue,  
  hint: const Text("Pilih Satu"),  
  items: const [  
    DropdownMenuItem(  
      value: 'Option 1',  
      child: Text('Option 1'),  
    ),  
    DropdownMenuItem(  
      value: 'Option 2',  
      child: Text('Option 2'),  
    ),  
    DropdownMenuItem(  
      value: 'Option 3',  
      child: Text('Option 3'),  
    ),  
  ],  
  onChanged: (value) {  
    setState(() {  
      _dropdownValue = value;  
    });  
  },  
),
```



Property di DropdownButton

Property	Class/DataType	Deskripsi
items	List<DropDownMenuItem <T>>	Berisi list item yang dapat dipilih oleh user
onChanged	ValueChanged<T?>	Fungsi yang dipanggil ketika terjadi perubahan nilai
value	T	Nilai awal atau yang dipilih oleh user
alignment	AlignmentGeometry	Menentukan hint atau item yang dipilih diposisikan di dalam button
borderRadius	BorderRadius	Menentukan border radius dari button
disabledHint	Widget	Menampilkan widget saat dropdown button disabled
dropdownColor	Color	Menentukan warna background dropdown
hint	Widget	Menampilkan widget saat dropdown belum dipilih
icon	Widget	Menampilkan icon pada button

Property lain dapat dilihat [di sini](#)

Terima Kasih

