**13. Number Picker**

Untuk membuat number picker kita perlu dependency

https://pub.dev/packages/numberpicker​, install seperti biasa, jika sudah kita ketikkan kode berikut :

import​ ​ 'package​ :flutter/material.dart';

import​ ​ 'dart​ :async';

import​ ​ 'package​ :numberpicker/numberpicker.dart';

class​ ​ PageNumberPicker​ ​ extends​ ​ StatefulWidget​ {

​ @override

\_PageNumberPickerState createState() => \_PageNumberPickerState();

}

class​ ​ \_PageNumberPickerState​ ​ extends​ ​ State<PageNumberPicker>​ {

int currentIntValue = ​ 10​ ;

double currentDoubleValue = ​ 3.0​ ;

​ NumberPicker​ intNumberPicker, decNumberPicker;

​ *// method untuk menghandle value ketika berubah*

handleValueChanged(num value) {

​ if​ (value != ​ null​ ) {

​ if​ (value is int) {

setState(() {

currentIntValue = value;

});

} ​ else​ {

setState(() {

currentDoubleValue = value;

});

}

}

}

​ *// method untuk menghandle value ketika berubah dari luar*

handleValueChangedExternally(num value) {

​ if​ (value != ​ null​ ) {

​ if​ (value is int) {

setState(() {

currentIntValue = value;

intNumberPicker.animateInt(value);

});

} ​ else​ {

setState(() {

currentDoubleValue = value;

decNumberPicker.animateDecimal(currentDoubleValue.toInt());

});

}

}

}

​ @override

​ Widget​ build(​ BuildContext​ context) {

intNumberPicker = ​ new​ ​ NumberPicker​ .integer(

initialValue: currentIntValue,

minValue: ​ 0​ ,

maxValue: ​ 100​ ,

step: ​ 10​ ,

onChanged: handleValueChanged);

decNumberPicker = ​ new​ ​ NumberPicker​ .decimal(

initialValue: currentDoubleValue,

minValue: ​ 1​ ,

maxValue: ​ 5​ ,

decimalPlaces: ​ 2​ , ​ *// mengatur 2 angka di belakang koma*

onChanged: handleValueChanged);

​ return​ ​ Scaffold​ (

appBar: ​ AppBar​ (

title: ​ Text​ (​ 'Page​ ​ Number​ ​ Picker​ '),

backgroundColor: ​ Colors​ .brown,

),

body: ​ Center​ (

child: ​ Column​ (

mainAxisAlignment: ​ MainAxisAlignment​ .spaceAround,

children: <​ Widget​ >[

intNumberPicker,

​ RaisedButton​ (

child: ​ Text​ (​ 'Current​ int value : $currentIntValue'),

color: ​ Colors​ .green,

onPressed: () {

showDialogInteger();

},

),

decNumberPicker,

​ RaisedButton​ (

child: ​ Text​ (​ 'Current​ ​ Decimal​ ​ Value​ :

$currentDoubleValue'),

onPressed: () {

showDialogDouble();

},

)

],

),

));

}

*// saat menekan tombol Current int value*

​ Future​ showDialogInteger() async {

await showDialog<int>(

context: context,

builder: (​ BuildContext​ context) {

​ return​ ​ NumberPickerDialog​ .integer(

minValue: ​ 0​ ,

maxValue: ​ 100​ ,

initialIntegerValue: currentIntValue,

);

}).then(handleValueChangedExternally);

}

​ *// saat menekan tombol Current dec value*

​ Future​ showDialogDouble() async {

await showDialog<double>(

context: context,

builder: (​ BuildContext​ context) {

​ return​ ​ NumberPickerDialog​ .decimal(

minValue: ​ 1​ ,

maxValue: ​ 5​ ,

initialDoubleValue: currentDoubleValue,

decimalPlaces: ​ 2​ ,

title: ​ Text​ (​ "Silahkan pilih bilangan desimal"​ ),

);

});

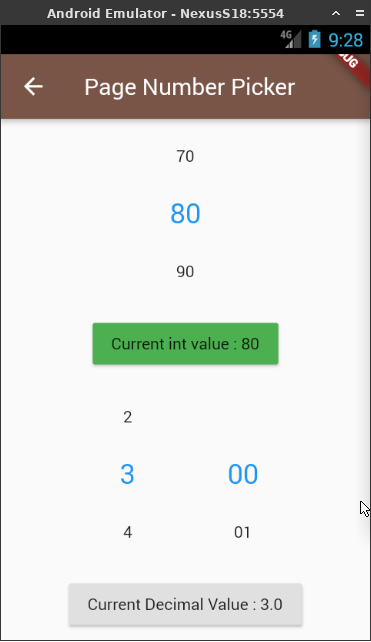
}

}

Penjelasan :

1. handleValueChanged(num value), yaitu fungsi yang dijalankan saat kita menggeser nilai angkanya
2. handleValueChangedExternally(num value), yaitu fungsi yang dijalankan saat kita menggeser nilai dari dialog yang muncul
3. showDialogInteger() = menampilkan dialog numberpicker integer
4. showDialogDouble() = menampilkan dialog numberpicker decimal

hasilnya :



**14. Membuat Search**

Untuk membuat halaman search kita ketikkan kode berikut :

import​ ​ 'package​ :flutter/material.dart';

class​ ​ SearchListPage​ ​ extends​ ​ StatefulWidget​ {

​ @override

\_SearchListPageState createState() => \_SearchListPageState();

}

class​ ​ \_SearchListPageState​ ​ extends​ ​ State<SearchListPage>​ {

​ var​ etSearce = ​ new​ ​ TextEditingController​ ();

bool isSearch = ​ true​ ;

​ String​ query = ​ ""​ ;

​ List​ <​ String​ > dataList;

​ List​ <​ String​ > filterList;

​ @override

void initState() {

​ *//* ​ *TODO:*​ *implement initState*

​ super​ .initState();

dataList = ​ new​ ​ List​ <​ String​ >();

dataList = [​ "Snake"​ , ​ "Elephant"​ , ​ "cats"​ , ​ "dog"​ , ​ "orion"​ ,

"boomerang"​ ,

​ "pelican"​ , ​ "ghost"​ , ​ "eagle"​ , ​ "horse head"​ , ​ "elephant trunk"​ ,

"butterfly"​ ];

dataList.sort();

}

\_SearchListPageState() {

etSearce.addListener((){

​ if​ (etSearce.text.isEmpty) {

setState(() {

isSearch = ​ false​ ;

query = ​ ""​ ;

});

} ​ else​ {

setState(() {

isSearch = ​ true​ ;

query = etSearce.text;

});

}

});

}

​ @override

​ Widget​ build(​ BuildContext​ context) {

​ return​ ​ Scaffold​ (

appBar: ​ AppBar​ (

title: ​ Text​ (​ 'Page​ ​ Search​ ​ Listview​ '),

backgroundColor: ​ Colors​ .brown,

),

body: ​ Container​ (

margin: ​ EdgeInsets​ .all(​ 10.0​ ),

child: ​ Column​ (

children: <​ Widget​ >[

\_createSearchView(),

isSearch ? \_performSearch() : \_createSearchView(),

],

),

),

);

}

​ *// membuat form search*

​ Widget​ \_createSearchView () {

​ return​ ​ Container​ (

decoration: ​ BoxDecoration​ (

border: ​ Border​ .all(width: ​ 1.0​ )

),

child: ​ TextField​ (

controller: etSearce,

decoration: ​ InputDecoration​ (

hintText: ​ "Search"​ ,

hintStyle: ​ TextStyle​ (color: ​ Colors​ .green)

),

textAlign: ​ TextAlign​ .center,

),

);

}

​ *// membuat form*

​ Widget​ \_createListView() {

​ return​ ​ Flexible​ (child: ​ ListView​ .builder(

itemCount: dataList.length,

itemBuilder: (​ BuildContext​ context, int index){

​ return​ ​ Card​ (

child: ​ Container​ (margin: ​ EdgeInsets​ .all(​ 10.0​ )),

color: ​ Colors​ .white,

elevation: ​ 5.0​ ,);

},

),);

}

​ *// lakukan search dengan fungsi contains*

​ Widget​ \_performSearch() {

filterList = ​ new​ ​ List​ <​ String​ >();

​ for​ (int i = ​ 0​ ; i < dataList.length; i++ ) {

var​ item = dataList[i];

​ if​ (item.toLowerCase().contains(query.toLowerCase())){

filterList.add(item);

}

}

​ return​ \_createFilteredListView();

}

*// tampilkan daftar hasil pencarian*

​ Widget​ \_createFilteredListView() {

​ return​ ​ Flexible​ (

child: ​ ListView​ .builder(

itemCount: filterList.length,

itemBuilder: (​ BuildContext​ context, int index) {

​ return​ ​ Card​ (

color: ​ Colors​ .white,

elevation: ​ 5.0​ ,

child: ​ Container​ (

margin: ​ EdgeInsets​ .all(​ 10.0​ ),

child: ​ Text​ ('${filterList[index]}'),

),

);

},),

);

}

}

Hasilnya :

