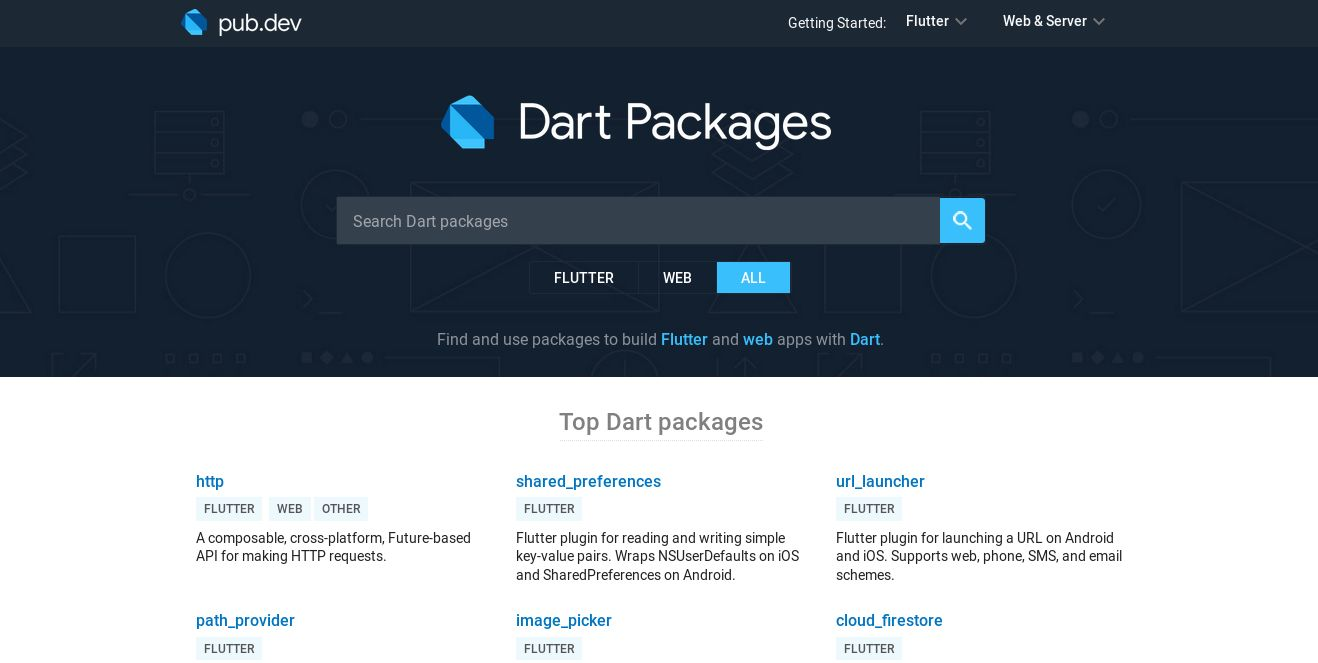
**9. Notification Widget**

Kali ini kita akan membuat notification widget, yaitu Toast, AlertDialog, dan SnackBar, untuk memakai nya kita perlu menginstall library yang bisa dilihat melalui website ​https://pub.dev/

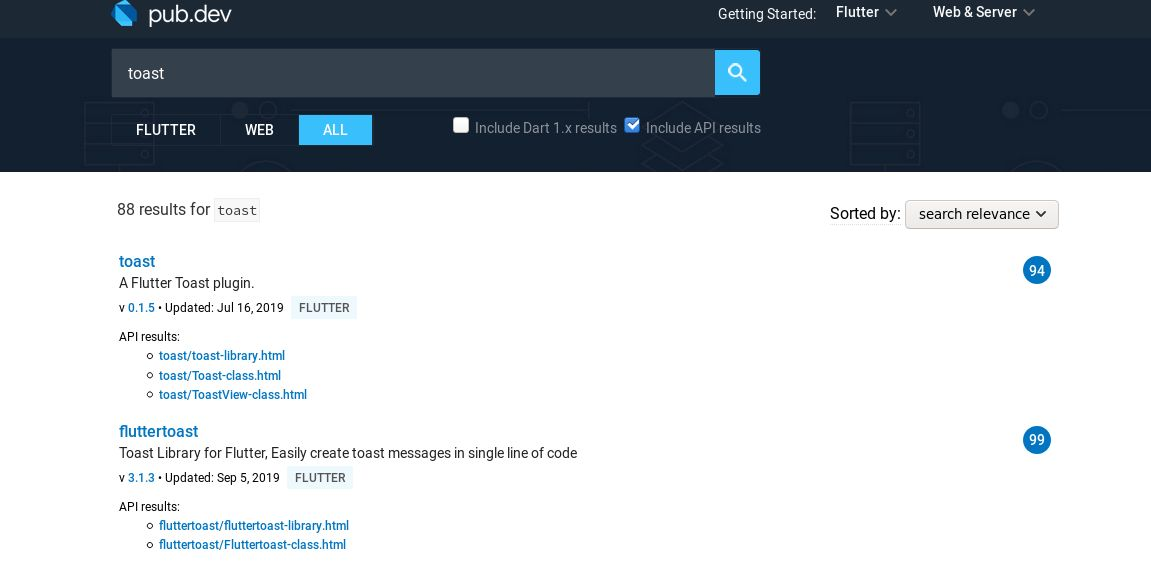


**A. Toast**

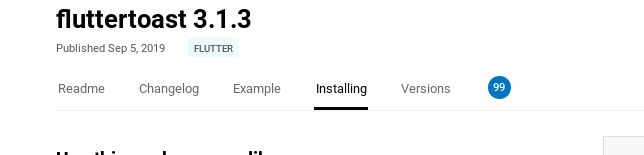
Untuk menggunakan toast, ada dua package yaitu toast, dan fluttertoast, bedanya toast lebih

sederhana sedangkan fluttertoast ada konfigurasi yang kita bisa atur:

1. Cari flutter :



2. Coba kita gunakan yang fluttertoast, cara installnya terdapat pada tab Installing



3. taruh ​ **fluttertoast: ^3.1.3​** , pada pubspec.yaml seperti berikut :

Pastikan posisinya sejajar dengan flutter.



lalu jalankan ​**flutter pub get**

4. import package-nya



lalu kita isikan codenya sesuai dengan panduannya.

body:​ Container(

color:​ Colors.red,

child:​ MaterialButton(

child:​ Text(​ "Toast"​ ),

textColor:​ Colors.white,

onPressed:​ () {

Fluttertoast.showToast(

msg:​ ​ "This is Center Short Toast"​ ,

toastLength:​ Toast.LENGTH\_SHORT,

gravity:​ ToastGravity.CENTER,

timeInSecForIos:​ ​ 1​ ,

backgroundColor:​ Colors.green,

textColor:​ Colors.white,

fontSize:​ ​ 16.0

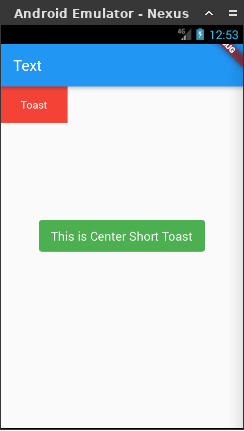
);

},

),

)

Hasilnya :

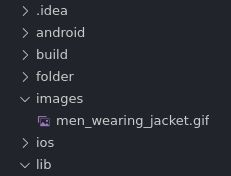


**B. Alert Dialog**

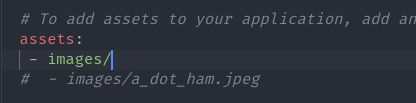
Untuk alert dialog kita gunakan giffy\_dialog, caranya :

* 1. buka ​ https://pub.dev/packages/giffy\_dialog​ , lalu taruh di pubspec.yaml dan import.
  2. Setelah itu coba download gambar ini

https://github.com/xsahil03x/giffy\_dialog/blob/master/example/assets/men\_wearing\_jacket.gif​ , lalu buat folder images dan taruh gambarnya seperti berikut :



* 1. Daftarkan asset images di pubspec.yaml



4. lalu masukan kode berikut :

Container(

​ color​ : Colors.red,

child: MaterialButton(

​ child​ : Text(​ "Alert Dialog"​ ),

textColor: Colors.white,

onPressed: () {

showDialog(

​ context​ : context,builder: (​ \_​ ) =>

NetworkGiffyDialog(

​ image​ :

Image.asset('assets/men\_wearing\_jacket.gif'),

title: Text('Men Wearing Jackets',

style: TextStyle(

​ fontSize​ : ​ 22.0​ , fontWeight:

FontWeight.w600),

),

description: Text('This is a men wearing

jackets',

textAlign: TextAlign.center,

style: TextStyle(),

),

onOkButtonPressed: () {},

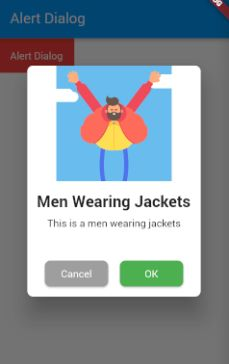
) )​ *;*

},

),

)

hasilnya :



**C. SnackBar**

Untuk membuat snackbar kita perlu menambahkan Globalkey, caranya seperti berikut :

class​ ​ \_MyHomePageState​ ​ extends​ ​ State<MyHomePage>​ {

​ GlobalKey​ <​ ScaffoldState​ > \_key = ​ GlobalKey​ <​ ScaffoldState​ >();

​ @override

​ Widget​ build(​ BuildContext​ context) {

​ return​ ​ Scaffold​ (

key: \_key,

appBar: ​ AppBar​ (title: ​ Text​ (​ "SnackBar"​ ),),

body: ​ Container​ (

color: ​ Colors​ .red,

child: ​ MaterialButton​ (

child: ​ Text​ (​ "Snack Bar"​ ),

textColor: ​ Colors​ .white,

onPressed: () {

\_key.currentState.showSnackBar(​ SnackBar​ (content:

Text​ (​ 'Snack​ bar'),));

},

),

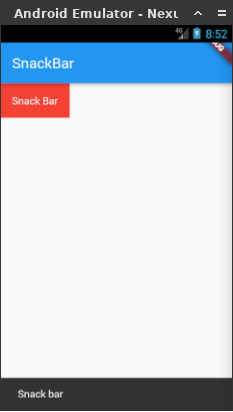
),

);

}

}

Lalu hasilnya :



**10. Retrieve Data**

**A. Passing dengan TextEditingController**

Untuk mengambil isi data pada sebuah form text kita membutuhkan TextEditingController  dan kita lakukan setState, lalu kita tampilkan dibawah seperti berikut :

class​ ​ SimpleLogin​ ​ extends​ ​ StatefulWidget​ {

​ @override

\_SimpleLoginState createState() => \_SimpleLoginState();

}

class​ ​ \_SimpleLoginState​ ​ extends​ ​ State<SimpleLogin>​ {

​ final​ ​ GlobalKey​ <​ ScaffoldState​ > \_key = ​ GlobalKey​ <​ ScaffoldState​ >();

​ TextEditingController​ usernameController = ​ TextEditingController​ ();

​ TextEditingController​ passwordController = ​ TextEditingController​ ();

​ String​ username = ​ ""​ , password = ​ ""​ ;

​ @override

​ Widget​ build(​ BuildContext​ context) {

​ return​ ​ Scaffold​ (

key: \_key,

appBar: ​ AppBar​ (

backgroundColor: ​ Colors​ .green,

title: ​ Text​ (​ 'Logi​ n'),

),

body: ​ Padding​ (

padding: const ​ EdgeInsets​ .all(​ 8.0​ ),

child: ​ Center​ (

child: ​ Column​ (

children: <​ Widget​ >[

​ TextFormField​ (

controller: usernameController,

decoration: ​ InputDecoration​ (

fillColor: ​ Colors​ .greenAccent,

hintText: ​ 'usernam​ e'

),

),

​ SizedBox​ (height: ​ 8.0​ ,),

​ TextFormField​ (

controller: passwordController,

Hasilnya :

obscureText: ​ true​ ,

decoration: ​ InputDecoration​ (

fillColor: ​ Colors​ .greenAccent,

hintText: ​ 'passwor​ d'

),

),

​ SizedBox​ (height: ​ 8.0​ ,),

​ Container​ (

color: ​ Colors​ .green,

child: ​ MaterialButton​ (

child: ​ Text​ (​ "submit"​ ),

textColor: ​ Colors​ .white,

onPressed: () {

setState(() {

username = usernameController.text;

password = passwordController.text;

});

},

),

),

​ SizedBox​ (height: ​ 16.0​ ,),

​ Text​ (username),

​ Text​ (password),

],

)

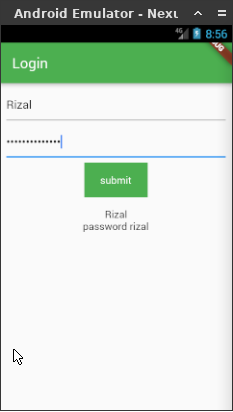
),

)

);

}

}



**B. Passing dengan Constructor**

Untuk membuat constructor dengan parameter, kita, cukup membuatnya seperti ini :

class​ ​ PageTwo​ ​ extends​ ​ StatelessWidget​ {

​ String​ msg;

// wajib diisi

​ PageTwo​ (​ this​ .msg);

​ @override

​ Widget​ build(​ BuildContext​ context) {

​ return​ ​ Scaffold​ (

appBar: ​ AppBar​ (

title: ​ Text​ (​ "Page Two"​ ),

),

body: ​ Column​ (

children: <​ Widget​ >[

​ Text​ (msg)

],

)

);

}

}

atau seperti ini artinya opsional

class​ ​ PageTwo​ ​ extends​ ​ StatelessWidget​ {

​ String​ msg = ​ ""​ ;

​ PageTwo​ (​ {this​ .msg​ }​ );

...

Cara memanggilnya seperti ini

PageTwo​ (​ "Pesan terkirim"​ )

untuk yang opsional seperti ini

PageTwo​ (msg: ​ "Pesan terkirim"​ )

hasilnya apabila dipanggil :

