

'PreferredSize' digunakan untuk menentukan ukuran yang diinginkan untuk deskripsi

'bottom' merupakan properti yang digunakan untuk menambahkan deskripsi ke 'AppBar'

'Size.fromHeight' akan membuat objek 'Size' dengan lebar default dan tinggi sesuai yang diinginkan

Dalam kasus ini, 'PreferredSize' digunakan untuk menentukan ukuran yang diinginkan untuk deskripsi di bawah judul 'AppBar'

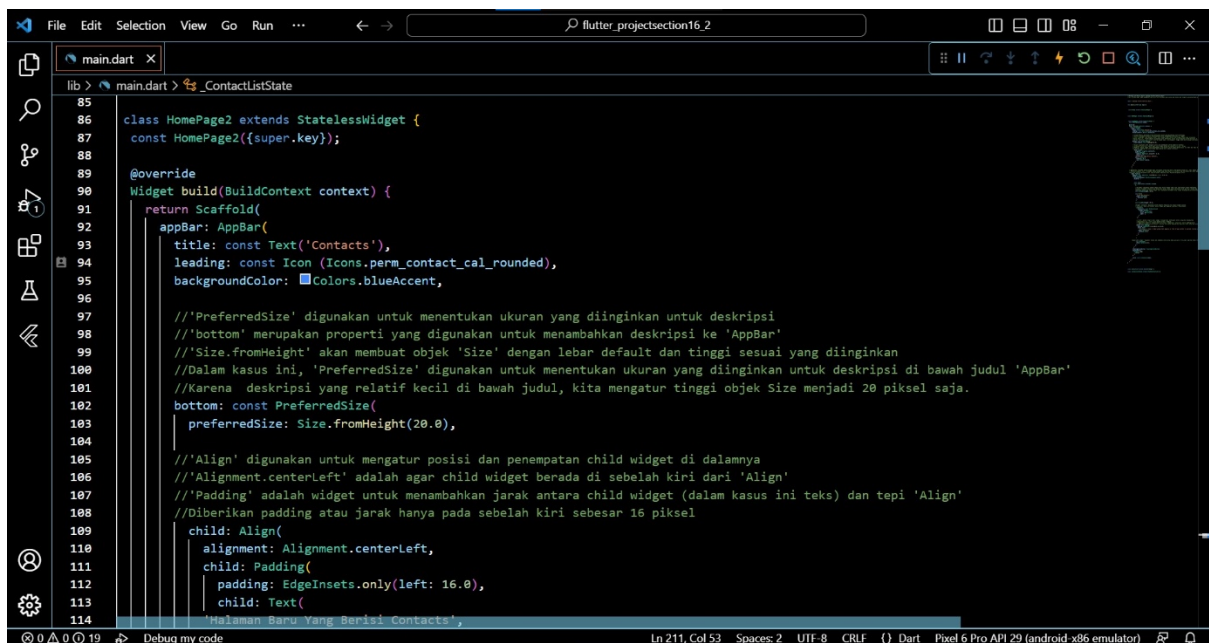
Karena deskripsi yang relatif kecil di bawah judul, kita mengatur tinggi objek Size menjadi 20 piksel saja.

'Align' digunakan untuk mengatur posisi dan penempatan child widget di dalamnya

'Alignment.centerLeft' adalah agar child widget berada di sebelah kiri dari 'Align'

'Padding' adalah widget untuk menambahkan jarak antara child widget (dalam kasus ini teks) dan tepi 'Align'

Diberikan padding atau jarak hanya pada sebelah kiri sebesar 16 piksel



```
85
86 class HomePage2 extends StatelessWidget {
87   const HomePage2({super.key});
88
89   @override
90   Widget build(BuildContext context) {
91     return Scaffold(
92       appBar: AppBar(
93         title: const Text('Contacts'),
94         leading: const Icon(Icons.perm_contact_cal_rounded),
95         backgroundColor: Colors.blueAccent,
96
97         // 'PreferredSize' digunakan untuk menentukan ukuran yang diinginkan untuk deskripsi
98         // 'bottom' merupakan properti yang digunakan untuk menambahkan deskripsi ke 'AppBar'
99         // 'Size.fromHeight' akan membuat objek 'Size' dengan lebar default dan tinggi sesuai yang diinginkan
100        // Dalam kasus ini, 'PreferredSize' digunakan untuk menentukan ukuran yang diinginkan untuk deskripsi di bawah judul 'AppBar'
101        // Karena deskripsi yang relatif kecil di bawah judul, kita mengatur tinggi objek Size menjadi 20 piksel saja.
102        bottom: const PreferredSize(
103          preferredSize: Size.fromHeight(20.0),
104
105          // 'Align' digunakan untuk mengatur posisi dan penempatan child widget di dalamnya
106          // 'Alignment.centerLeft' adalah agar child widget berada di sebelah kiri dari 'Align'
107          // 'Padding' adalah widget untuk menambahkan jarak antara child widget (dalam kasus ini teks) dan tepi 'Align'
108          // Diberikan padding atau jarak hanya pada sebelah kiri sebesar 16 piksel
109          child: Align(
110            alignment: Alignment.centerLeft,
111            child: Padding(
112              padding: EdgeInsets.only(left: 16.0),
113              child: Text(
114                'Halaman Baru Yang Berisi Contacts',
115              ),
116            ),
117          ),
118        ),
119     );
120   }
121 }
```

'EdgeInsets.fromLTRB' adalah widget agar kita dapat custom mau kasih jarak berapa untuk kiri, atas, kanan, dan bawah dari child widget

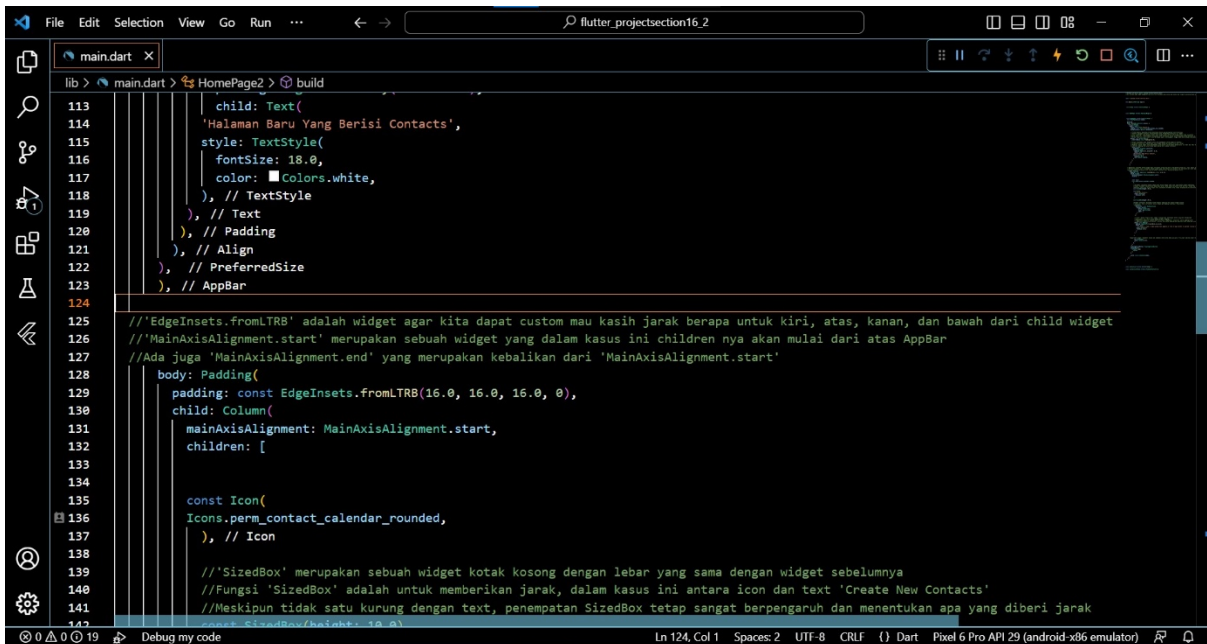
'MainAxisAlignment.start' merupakan sebuah widget yang dalam kasus ini children nya akan mulai dari atas AppBar

Ada juga 'MainAxisAlignment.end' yang merupakan kebalikan dari 'MainAxisAlignment.start'

'SizedBox' merupakan sebuah widget kotak kosong dengan lebar yang sama dengan widget sebelumnya

Fungsi 'SizedBox' adalah untuk memberikan jarak, dalam kasus ini antara icon dan text 'Create New Contacts'

Meskipun tidak satu kurung dengan text, penempatan SizedBox tetap sangat berpengaruh dan menentukan apa yang diberi jarak



```
lib > main.dart > HomePage2 > build
113 | | | | | child: Text(
114 | | | | | 'Halaman Baru Yang Berisi Contacts',
115 | | | | | style: TextStyle(
116 | | | | |   fontSize: 18.0,
117 | | | | |   color: Colors.white,
118 | | | | | ), // TextStyle
119 | | | | | ), // Text
120 | | | | | ), // Padding
121 | | | | | ), // Align
122 | | | | | ), // PreferredSize
123 | | | | | ), // AppBar
124 | | | | |
125 | | | | | // 'EdgeInsets.fromLTRB' adalah widget agar kita dapat custom mau kasih jarak berapa untuk kiri, atas, kanan, dan bawah dari child widget
126 | | | | | // 'MainAxisAlignment.start' merupakan sebuah widget yang dalam kasus ini children nya akan mulai dari atas AppBar
127 | | | | | // Ada juga 'MainAxisAlignment.end' yang merupakan kebalikan dari 'MainAxisAlignment.start'
128 | | | | | body: Padding(
129 | | | | |   padding: const EdgeInsets.fromLTRB(16.0, 16.0, 16.0, 0),
130 | | | | |   child: Column(
131 | | | | |     mainAxisAlignment: MainAxisAlignment.start,
132 | | | | |     children: [
133 | | | | |
134 | | | | |
135 | | | | |     const Icon(
136 | | | | |       Icons.perm_contact_calendar_rounded,
137 | | | | |     ), // Icon
138 | | | | |
139 | | | | |     // 'SizedBox' merupakan sebuah widget kotak kosong dengan lebar yang sama dengan widget sebelumnya
140 | | | | |     // Fungsi 'SizedBox' adalah untuk memberikan jarak, dalam kasus ini antara icon dan text 'Create New Contacts'
141 | | | | |     // Meskipun tidak satu kurung dengan text, penempatan SizedBox tetap sangat berpengaruh dan menentukan apa yang diberi jarak
142 | | | | |     const SizedBox(height: 10.0)
```

Widget 'Container' digunakan untuk mengatur tampilan atau layout widget lainnya

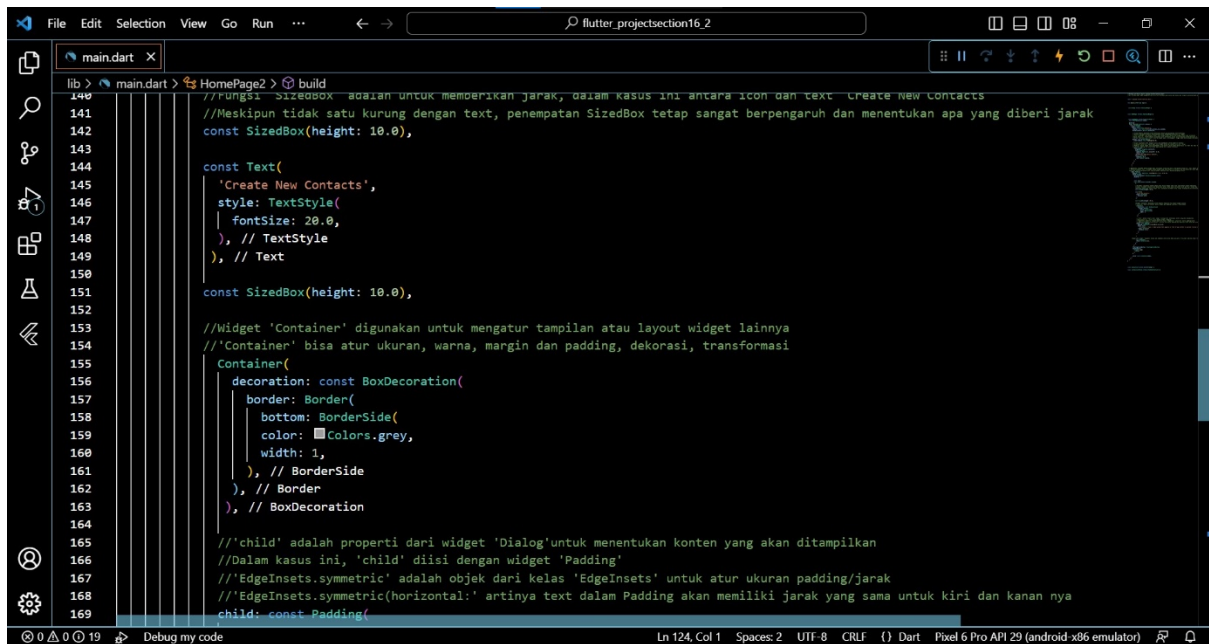
'Container' bisa atur ukuran, warna, margin dan padding, dekorasi, transformasi

'child' adalah properti dari widget 'Dialog' untuk menentukan konten yang akan ditampilkan

Dalam kasus ini, 'child' diisi dengan widget 'Padding'

'EdgeInsets.symmetric' adalah objek dari kelas 'EdgeInsets' untuk atur ukuran padding/jarak

'EdgeInsets.symmetric(horizontal:)' artinya text dalam Padding akan memiliki jarak yang sama untuk kiri dan kanan nya



```
lib > main.dart > HomePage2 > build
//fungsi: SizedBox adalah untuk memberikan jarak, dalam kasus ini antara icon dan text 'Create New Contacts'
//Meskipun tidak satu kurung dengan text, penempatan SizedBox tetap sangat berpengaruh dan menentukan apa yang diberi jarak
const SizedBox(height: 10.0),

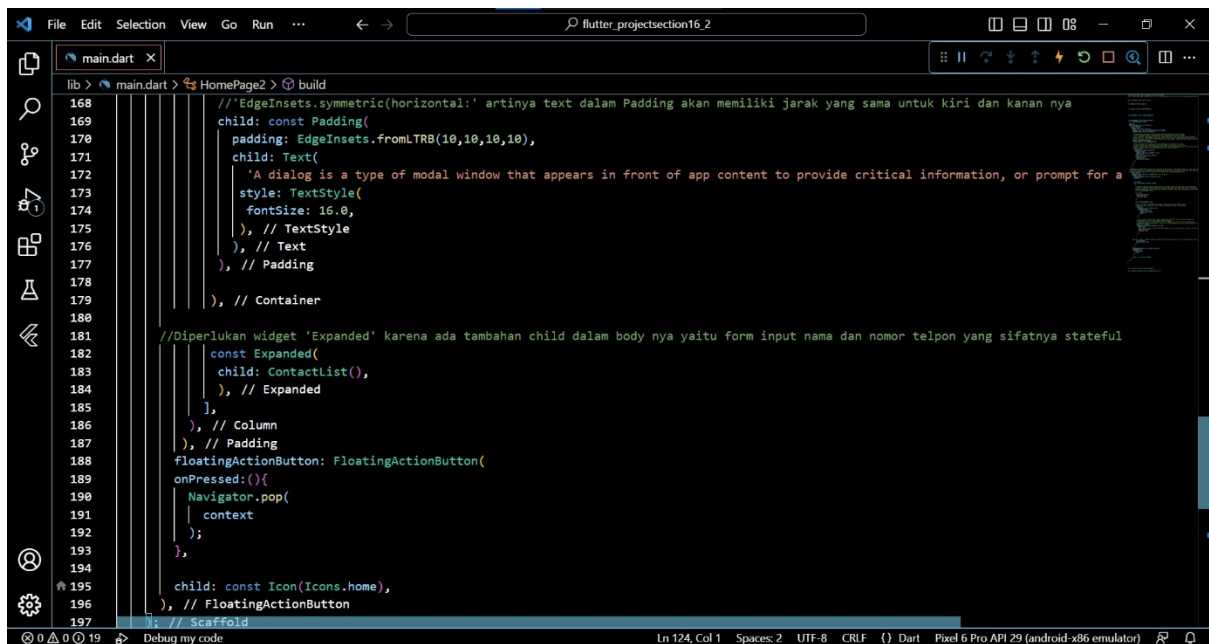
const Text(
  'Create New Contacts',
  style: TextStyle(
    |   fontSize: 20.0,
    |   ), // TextStyle
  ), // Text

const SizedBox(height: 10.0),

//Widget 'Container' digunakan untuk mengatur tampilan atau layout widget lainnya
//'Container' bisa atur ukuran, warna, margin dan padding, dekorasi, transformasi
Container(
  decoration: const BoxDecoration(
    border: Border(
      bottom: BorderSide(
        color: Colors.grey,
        width: 1,
      ), // BorderSide
    ), // Border
  ), // BoxDecoration

  //child' adalah properti dari widget 'Dialog' untuk menentukan konten yang akan ditampilkan
  //Dalam kasus ini, 'child' diisi dengan widget 'Padding'
  //EdgeInsets.symmetric' adalah objek dari kelas 'EdgeInsets' untuk atur ukuran padding/jarak
  //EdgeInsets.symmetric(horizontal: ' artinya text dalam Padding akan memiliki jarak yang sama untuk kiri dan kanan nya
  child: const Padding(
```

Diperlukan widget 'Expanded' karena ada tambahan child dalam body nya yaitu form input nama dan nomor telpon yang sifatnya stateful



```
lib > main.dart > HomePage2 > build
//EdgeInsets.symmetric(horizontal: ' artinya text dalam Padding akan memiliki jarak yang sama untuk kiri dan kanan nya
child: const Padding(
  padding: EdgeInsets.fromLTRB(10,10,10,10),
  child: Text(
    'A dialog is a type of modal window that appears in front of app content to provide critical information, or prompt for a
    style: TextStyle(
      fontSize: 16.0,
    ), // TextStyle
  ), // Text
), // Padding
), // Container

//Diperlukan widget 'Expanded' karena ada tambahan child dalam body nya yaitu form input nama dan nomor telpon yang sifatnya stateful
const Expanded(
  child: ContactList(),
), // Expanded
), // Column
), // Padding
floatingActionButton: FloatingActionButton(
  onPressed: (){
    Navigator.pop(
      |   context
    );
  },
  child: const Icon(Icons.home),
), // FloatingActionButton
); // Scaffold
```

'TextEditingController' akan mengontrol nilai yang diinputkan ke dalam 'TextField'

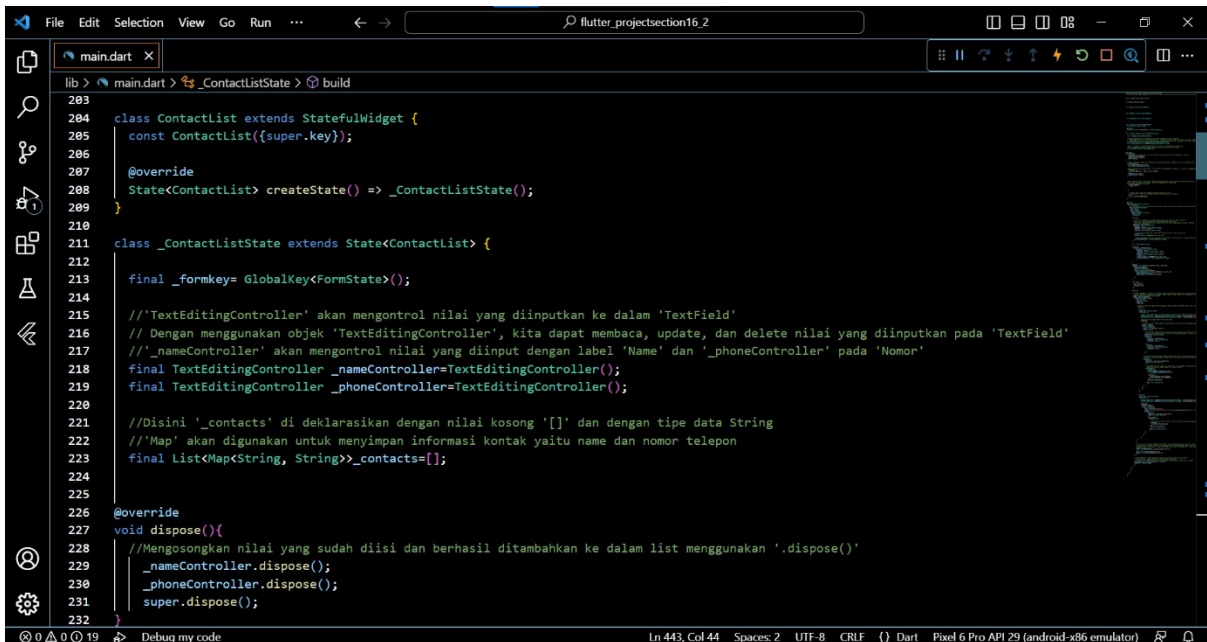
Dengan menggunakan objek 'TextEditingController', kita dapat membaca, update, dan delete nilai yang diinputkan pada 'TextField'

'_nameController' akan mengontrol nilai yang diinput dengan label 'Name' dan '_phoneController' pada 'Nomor'

Disini '_contacts' di deklarasikan dengan nilai kosong '[]' dan dengan tipe data String

'Map' akan digunakan untuk menyimpan informasi kontak yaitu name dan nomor telepon

Mengosongkan nilai yang sudah diisi dan berhasil ditambahkan ke dalam list menggunakan '.dispose()'



```
lib > main.dart > _ContactListState > build
203
204 class ContactList extends StatefulWidget {
205   const ContactList({super.key});
206
207   @override
208   State<ContactList> createState() => _ContactListState();
209 }
210
211 class _ContactListState extends State<ContactList> {
212
213   final _formkey= GlobalKey<FormState>();
214
215   //'TextEditingController' akan mengontrol nilai yang diinputkan ke dalam 'TextField'
216   // Dengan menggunakan objek 'TextEditingController', kita dapat membaca, update, dan delete nilai yang diinputkan pada 'TextField'
217   //'_nameController' akan mengontrol nilai yang diinput dengan label 'Name' dan '_phoneController' pada 'Nomor'
218   final TextEditingController _nameController=TextEditingController();
219   final TextEditingController _phoneController=TextEditingController();
220
221   //Disini 'contacts' di deklarasikan dengan nilai kosong '[]' dan dengan tipe data String
222   //'Map' akan digunakan untuk menyimpan informasi kontak yaitu name dan nomor telepon
223   final List<Map<String, String>>_contacts=[];
224
225
226   @override
227   void dispose(){
228     //Mengosongkan nilai yang sudah diisi dan berhasil ditambahkan ke dalam list menggunakan '.dispose()'
229     _nameController.dispose();
230     _phoneController.dispose();
231     super.dispose();
232   }
233 }
```

Fungsi 'addContact' adalah untuk menerima nilai yang diinput dalam '_nameController' dan '_phoneController' yang berupa text

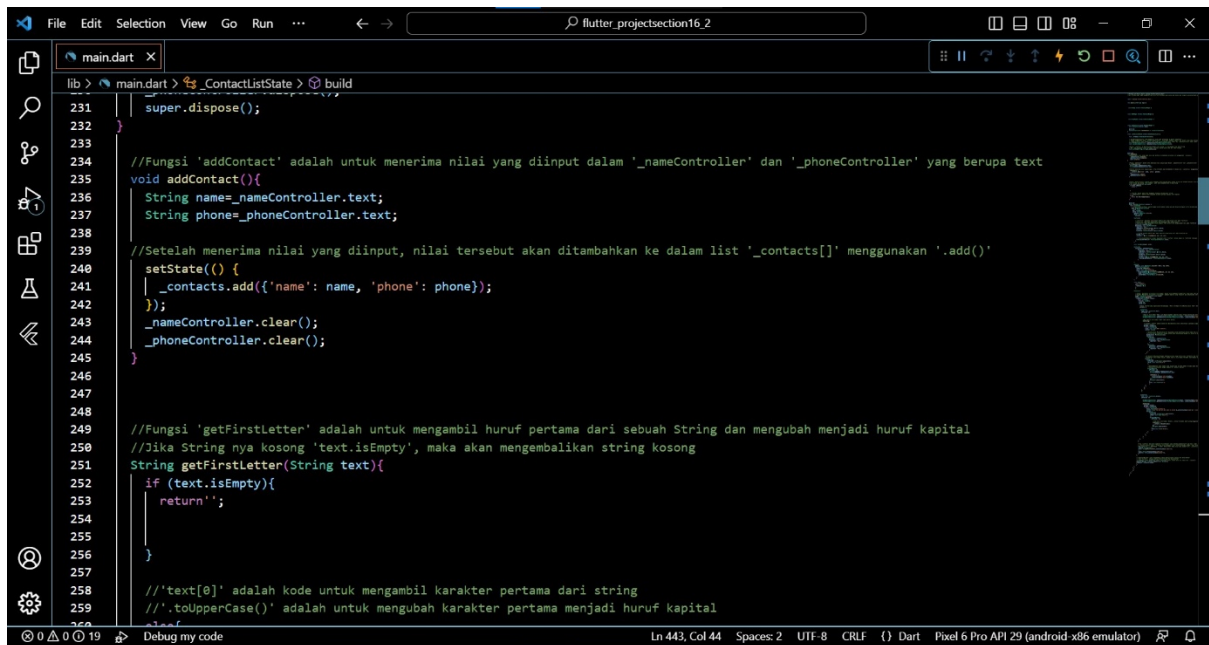
Setelah menerima nilai yang diinput, nilai tersebut akan ditambahkan ke dalam list '_contacts[]' menggunakan '.add()'

Fungsi 'getFirstLetter' adalah untuk mengambil huruf pertama dari sebuah String dan mengubah menjadi huruf kapital

Jika String nya kosong 'text.isEmpty', maka akan mengembalikan string kosong

'text[0]' adalah kode untuk mengambil karakter pertama dari string

'.toUpperCase()' adalah untuk mengubah karakter pertama menjadi huruf kapital



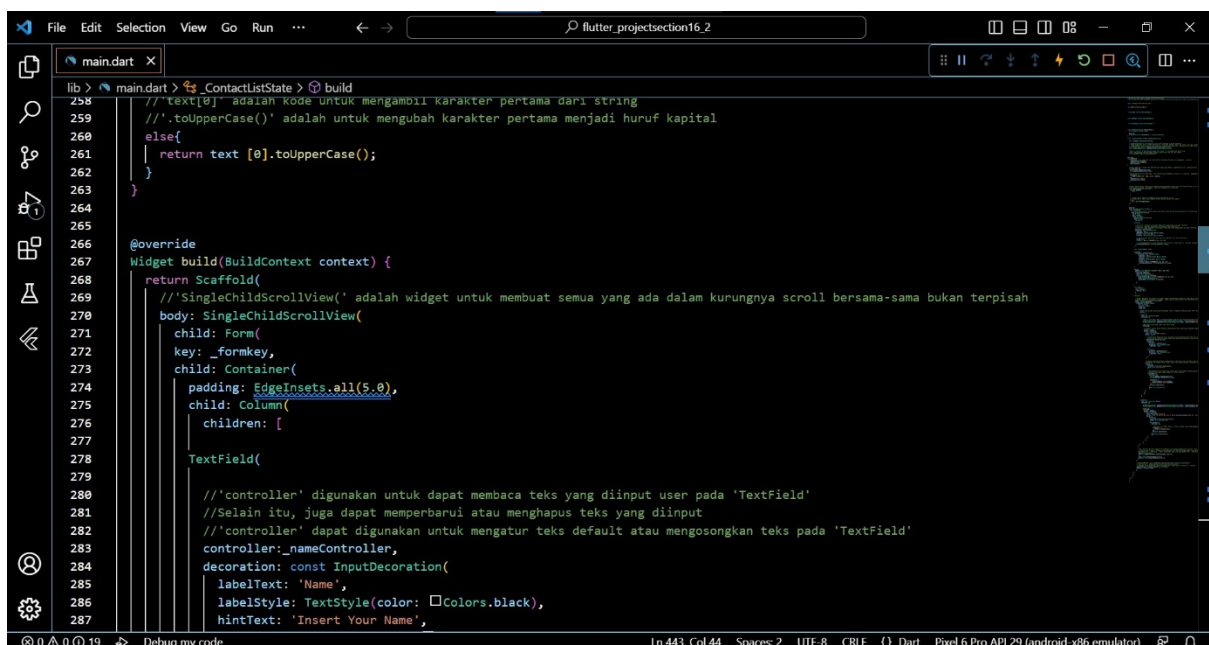
```
lib > main.dart > ContactListState > build
231 | super.dispose();
232 | }
233 |
234 | //Fungsi 'addContact' adalah untuk menerima nilai yang diinput dalam '_nameController' dan '_phoneController' yang berupa text
235 | void addContact(){
236 |   String name=_nameController.text;
237 |   String phone=_phoneController.text;
238 |
239 |   //Setelah menerima nilai yang diinput, nilai tersebut akan ditambahkan ke dalam list '_contacts[]' menggunakan '.add()'
240 |   setState() {
241 |     _contacts.add({'name': name, 'phone': phone});
242 |   });
243 |   _nameController.clear();
244 |   _phoneController.clear();
245 | }
246 |
247 |
248 |
249 | //Fungsi 'getFirstLetter' adalah untuk mengambil huruf pertama dari sebuah String dan mengubah menjadi huruf kapital
250 | //Jika String nya kosong 'text.isEmpty()', maka akan mengembalikan string kosong
251 | String getFirstLetter(String text){
252 |   if (text.isEmpty){
253 |     return '';
254 |   }
255 |
256 | }
257 |
258 | // 'text[0]' adalah kode untuk mengambil karakter pertama dari string
259 | // '.toUpperCase()' adalah untuk mengubah karakter pertama menjadi huruf kapital
260 | }
```

'SingleChildScrollView(' adalah widget untuk membuat semua yang ada dalam kurungnya scroll bersama-sama bukan terpisah

'controller' digunakan untuk dapat membaca teks yang diinput user pada 'TextField'

Selain itu, juga dapat memperbarui atau menghapus teks yang diinput

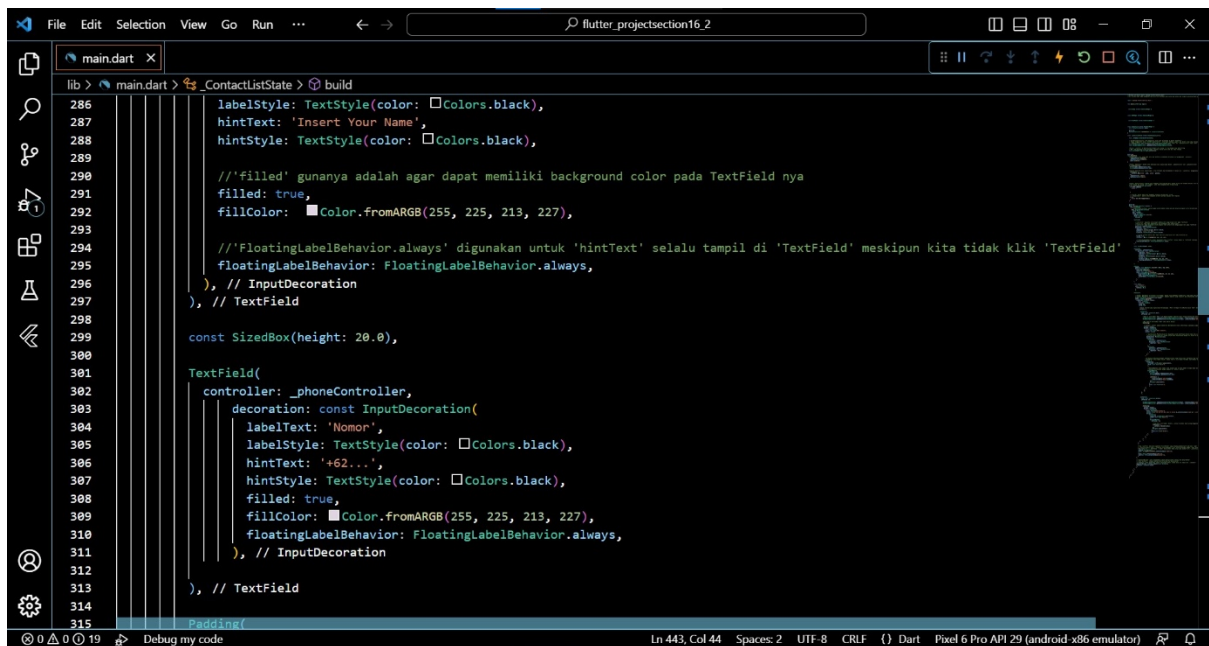
'controller' dapat digunakan untuk mengatur teks default atau mengosongkan teks pada 'TextField'



```
lib > main.dart > ContactListState > build
258 | // 'text[0]' adalah kode untuk mengambil karakter pertama dari string
259 | // '.toUpperCase()' adalah untuk mengubah karakter pertama menjadi huruf kapital
260 | else{
261 |   return text [0].toUpperCase();
262 | }
263 |
264 | }
265 |
266 | @override
267 | Widget build(BuildContext context) {
268 |   return Scaffold(
269 |     // 'SingleChildScrollView(' adalah widget untuk membuat semua yang ada dalam kurungnya scroll bersama-sama bukan terpisah
270 |     body: SingleChildScrollView(
271 |       child: Form(
272 |         key: _formkey,
273 |         child: Container(
274 |           padding: EdgeInsets.all(5.0),
275 |           child: Column(
276 |             children: [
277 |
278 |               TextField(
279 |
280 |                 // 'controller' digunakan untuk dapat membaca teks yang diinput user pada 'TextField'
281 |                 // Selain itu, juga dapat memperbarui atau menghapus teks yang diinput
282 |                 // 'controller' dapat digunakan untuk mengatur teks default atau mengosongkan teks pada 'TextField'
283 |                 controller: _nameController,
284 |                 decoration: const InputDecoration(
285 |                   labelText: 'Name',
286 |                   labelStyle: TextStyle(color: Colors.black),
287 |                   hintText: 'Insert Your Name',
288 |                 ),
289 |               ),
290 |             ],
291 |           ),
292 |         ),
293 |       ),
294 |     ),
295 |   );
296 | }
```

'filled' gunanya adalah agar dapat memiliki background color pada TextField nya

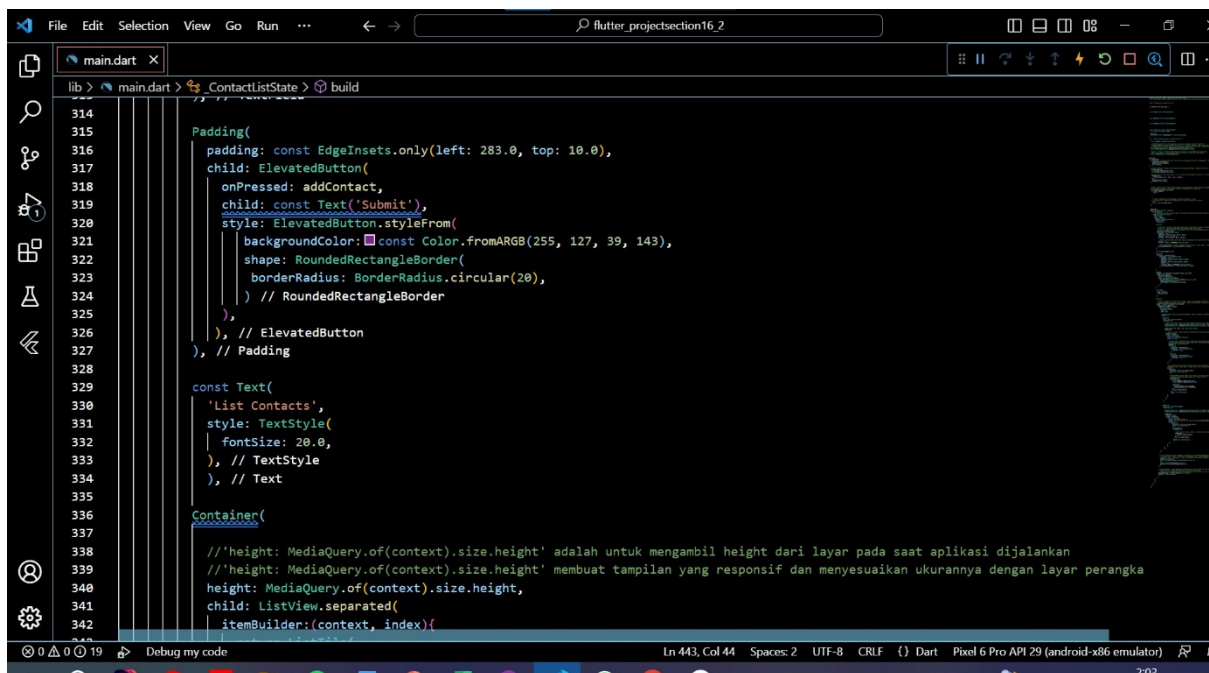
'FloatingLabelBehavior.always' digunakan untuk 'hintText' selalu tampil di 'TextField' meskipun kita tidak klik 'TextField'



```
lib > main.dart > ContactListState > build
286 |       labelStyle: TextStyle(color: Colors.black),
287 |       hintText: 'Insert Your Name',
288 |       hintStyle: TextStyle(color: Colors.black),
289 |
290 |       // 'filled' gunanya adalah agar dapat memiliki background color pada TextField nya
291 |       filled: true,
292 |       fillColor: Color.fromARGB(255, 225, 213, 227),
293 |
294 |       // 'FloatingLabelBehavior.always' digunakan untuk 'hintText' selalu tampil di 'TextField' meskipun kita tidak klik 'TextField'
295 |       floatingLabelBehavior: FloatingLabelBehavior.always,
296 |     ), // InputDecoration
297 |   ), // TextField
298 |
299 |   const SizedBox(height: 20.0),
300 |
301 |   TextField(
302 |     controller: _phoneController,
303 |     decoration: const InputDecoration(
304 |       labelText: 'Nomor',
305 |       labelStyle: TextStyle(color: Colors.black),
306 |       hintText: '+62...',
307 |       hintStyle: TextStyle(color: Colors.black),
308 |       filled: true,
309 |       fillColor: Color.fromARGB(255, 225, 213, 227),
310 |       floatingLabelBehavior: FloatingLabelBehavior.always,
311 |     ), // InputDecoration
312 |   ), // TextField
313 |
314 |   Padding(
315 |
```

'height: MediaQuery.of(context).size.height' adalah untuk mengambil height dari layar pada saat aplikasi dijalankan

'height: MediaQuery.of(context).size.height' membuat tampilan yang responsif dan menyesuaikan ukurannya dengan layar perangkat



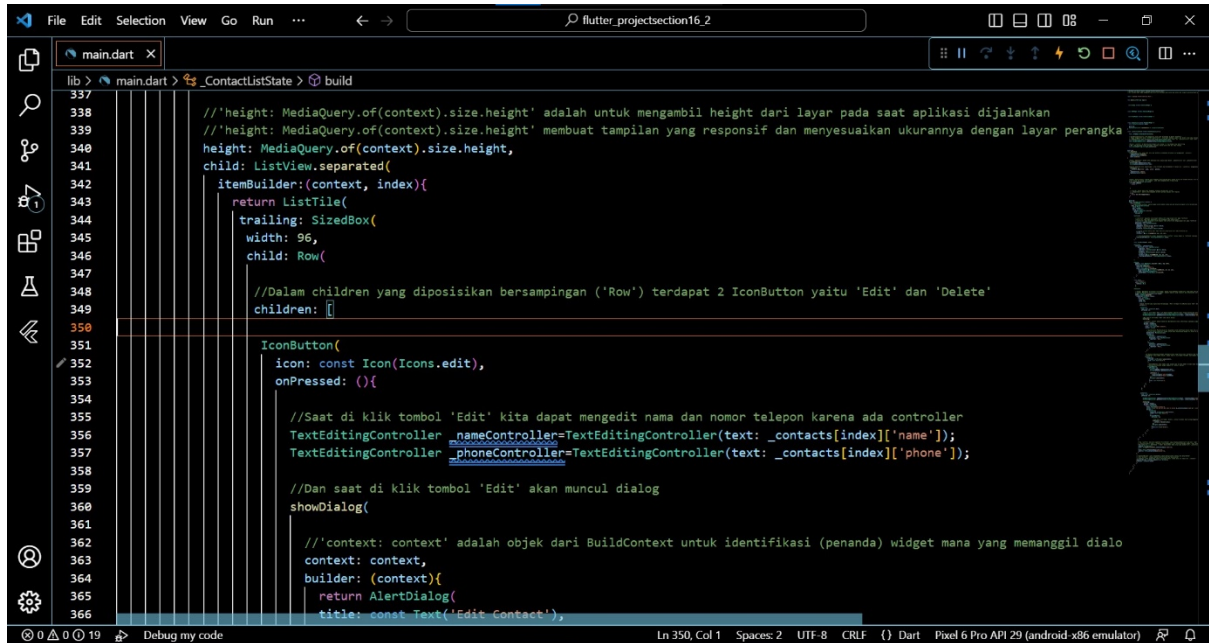
```
lib > main.dart > ContactListState > build
314 |   Padding(
315 |     padding: const EdgeInsets.only(left: 283.0, top: 10.0),
316 |     child: ElevatedButton(
317 |       onPressed: addContact,
318 |       child: const Text('Submit'),
319 |       style: ElevatedButton.styleFrom(
320 |         backgroundColor: const Color.fromARGB(255, 127, 39, 143),
321 |         shape: RoundedRectangleBorder(
322 |           borderRadius: BorderRadius.circular(20),
323 |         ) // RoundedRectangleBorder
324 |       ),
325 |     ), // ElevatedButton
326 |   ), // Padding
327 |
328 |   const Text(
329 |     'List Contacts',
330 |     style: TextStyle(
331 |       fontSize: 20.0,
332 |     ), // TextStyle
333 |   ), // Text
334 |
335 |   Container(
336 |
337 |     // 'height: MediaQuery.of(context).size.height' adalah untuk mengambil height dari layar pada saat aplikasi dijalankan
338 |     // 'height: MediaQuery.of(context).size.height' membuat tampilan yang responsif dan menyesuaikan ukurannya dengan layar perangkat
339 |     height: MediaQuery.of(context).size.height,
340 |     child: ListView.separated(
341 |       itemBuilder: (context, index){
342 |
```

Dalam children yang diposisikan bersampingan ('Row') terdapat 2 IconButton yaitu 'Edit' dan 'Delete'

Saat di klik tombol 'Edit' kita dapat mengedit nama dan nomor telepon karena ada controller

Dan saat di klik tombol 'Edit' akan muncul dialog

'context: context' adalah objek dari BuildContext untuk identifikasi (penanda) widget mana yang memanggil dialog



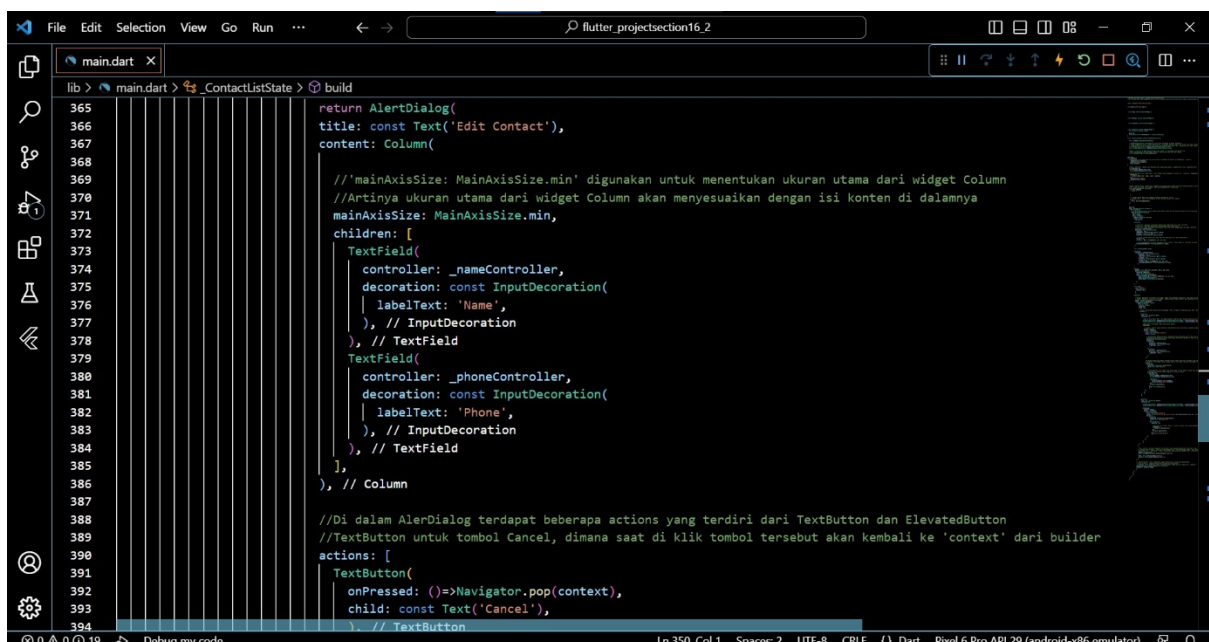
```
lib > main.dart > _ContactListState > build
337
338 // 'height: MediaQuery.of(context).size.height' adalah untuk mengambil height dari layar pada saat aplikasi dijalankan
339 // 'height: MediaQuery.of(context).size.height' membuat tampilan yang responsif dan menyesuaikan ukurannya dengan layar perangkat
340 height: MediaQuery.of(context).size.height,
341 child: ListView.separated(
342   itemBuilder: (context, index){
343     return ListTile(
344       trailing: SizedBox(
345         width: 96,
346         child: Row(
347
348           // Dalam children yang diposisikan bersampingan ('Row') terdapat 2 IconButton yaitu 'Edit' dan 'Delete'
349           children: [
350
351             IconButton(
352               icon: const Icon(Icons.edit),
353               onPressed: (){
354
355                 // Saat di klik tombol 'Edit' kita dapat mengedit nama dan nomor telepon karena ada controller
356                 TextEditingController _nameController = TextEditingController(text: _contacts[index]['name']);
357                 TextEditingController _phoneController = TextEditingController(text: _contacts[index]['phone']);
358
359                 // Dan saat di klik tombol 'Edit' akan muncul dialog
360                 showDialog(
361
362                   // 'context: context' adalah objek dari BuildContext untuk identifikasi (penanda) widget mana yang memanggil dialog
363                   context: context,
364                   builder: (context){
365                     return AlertDialog(
366                       title: const Text('Edit Contact'),
```

'mainAxisSize: MainAxisSize.min' digunakan untuk menentukan ukuran utama dari widget Column

Artinya ukuran utama dari widget Column akan menyesuaikan dengan isi konten di dalamnya

Di dalam AlertDialog terdapat beberapa actions yang terdiri dari TextButton dan ElevatedButton

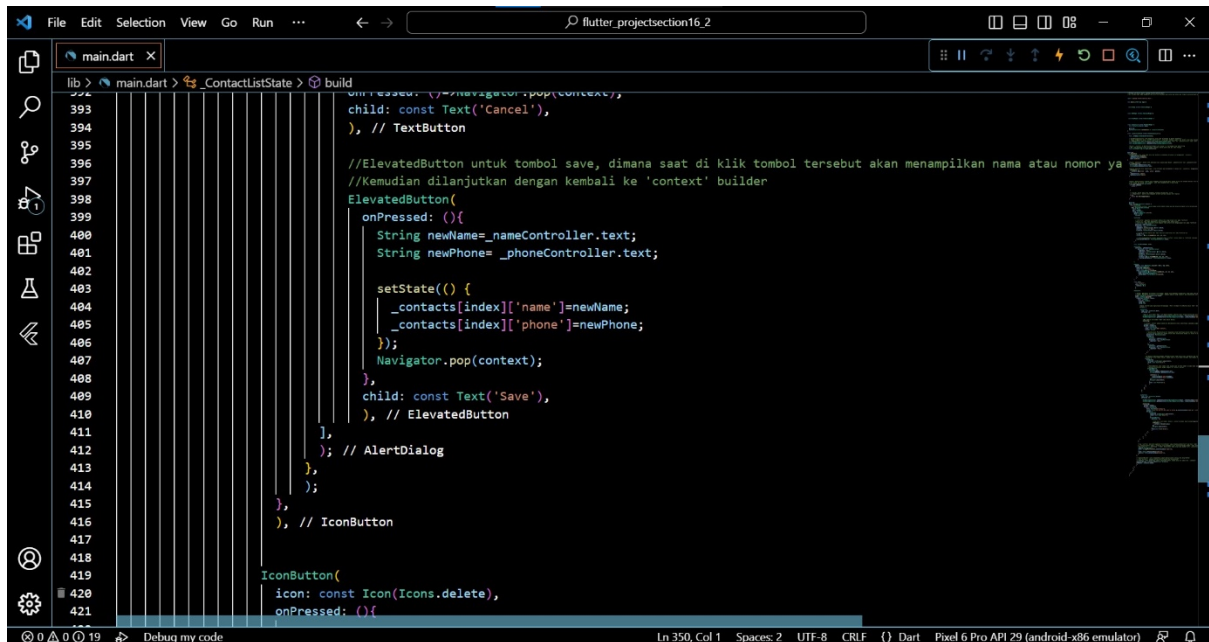
TextButton untuk tombol Cancel, dimana saat di klik tombol tersebut akan kembali ke 'context' dari builder



```
365 return AlertDialog(
366   title: const Text('Edit Contact'),
367   content: Column(
368
369     // 'mainAxisSize: MainAxisSize.min' digunakan untuk menentukan ukuran utama dari widget Column
370     // Artinya ukuran utama dari widget Column akan menyesuaikan dengan isi konten di dalamnya
371     mainAxisSize: MainAxisSize.min,
372     children: [
373       TextField(
374         controller: _nameController,
375         decoration: const InputDecoration(
376           labelText: 'Name',
377         ), // InputDecoration
378       ), // TextField
379       TextField(
380         controller: _phoneController,
381         decoration: const InputDecoration(
382           labelText: 'Phone',
383         ), // InputDecoration
384       ), // TextField
385     ],
386   ), // Column
387
388   // Di dalam AlertDialog terdapat beberapa actions yang terdiri dari TextButton dan ElevatedButton
389   // TextButton untuk tombol Cancel, dimana saat di klik tombol tersebut akan kembali ke 'context' dari builder
390   actions: [
391     TextButton(
392       onPressed: () => Navigator.pop(context),
393       child: const Text('Cancel'),
394     ), // TextButton
```

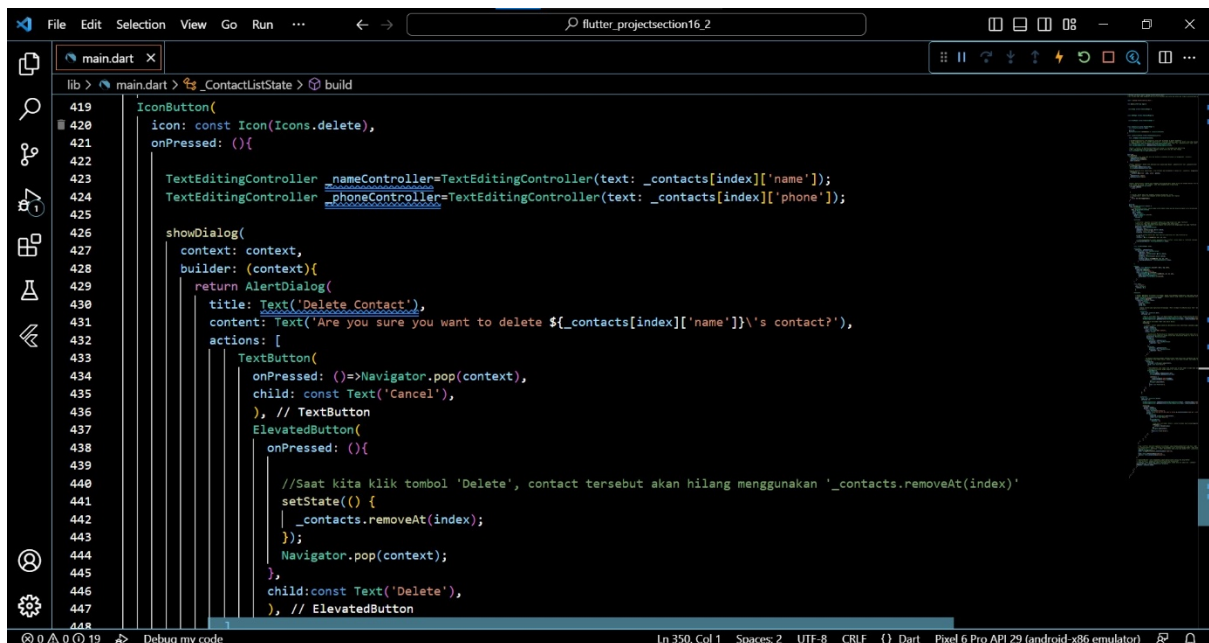
ElevatedButton untuk tombol save, dimana saat di klik tombol tersebut akan menampilkan nama atau nomor yang diedit

Kemudian dilanjutkan dengan kembali ke 'context' builder



```
lib > main.dart > _ContactListState > build
393       onPressed: ()=>Navigator.pop(context),
394       child: const Text("Cancel"),
395     ), // TextButton
396
397     //ElevatedButton untuk tombol save, dimana saat di klik tombol tersebut akan menampilkan nama atau nomor ya
398     //Kemudian dilanjutkan dengan kembali ke 'context' builder
399     ElevatedButton(
400       onPressed: (){
401         String newName=_nameController.text;
402         String newPhone= _phoneController.text;
403
404         setState(() {
405           _contacts[index]['name']=newName;
406           _contacts[index]['phone']=newPhone;
407         });
408         Navigator.pop(context);
409       },
410       child: const Text('Save'),
411     ), // ElevatedButton
412   ), // AlertDialog
413 ),
414 );
415 },
416 ), // IconButton
417
418
419 IconButton(
420   icon: const Icon(Icons.delete),
421   onPressed: (){
```

Saat kita klik tombol 'Delete', contact tersebut akan hilang menggunakan '_contacts.removeAt(index)'



```
419   IconButton(
420     icon: const Icon(Icons.delete),
421     onPressed: (){
422
423       TextEditingController _nameController=TextEditingController(text: _contacts[index]['name']);
424       TextEditingController _phoneController=TextEditingController(text: _contacts[index]['phone']);
425
426       showDialog(
427         context: context,
428         builder: (context){
429           return AlertDialog(
430             title: Text("Delete Contact"),
431             content: Text('Are you sure you want to delete ${_contacts[index]['name']}'s contact?'),
432             actions: [
433               TextButton(
434                 onPressed: ()=>Navigator.pop(context),
435                 child: const Text("Cancel"),
436               ), // TextButton
437               ElevatedButton(
438                 onPressed: (){
439
440                   //Saat kita klik tombol 'Delete', contact tersebut akan hilang menggunakan '_contacts.removeAt(index)'
441                   setState(() {
442                     _contacts.removeAt(index);
443                   });
444                   Navigator.pop(context);
445                 },
446                 child:const Text('Delete'),
447               ), // ElevatedButton
448             ],
449           );
450         },
451       );
452     },
453   ),
454 );
```

Tiap 'ListTile' akan ada lingkaran/'CircleAvatar' yang dalamnya adalah huruf awal dari 'name'

'_contacts[index]['name']' fungsinya untuk mengambil nilai yang berhubungan dengan 'name' dalam Map

'itemCount' untuk menentukan jumlah item dalam daftar, dalam kasus ini semua list '_contacts'

File Edit Selection View Go Run ... flutter_projectsection16.2

main.dart x

lib > main.dart > ContactListState > build

```

447         ), // ElevatedButton
448     ],
449 ); // AlertDialog
450 }
451 );
452 },
453 ), // IconButton
454 ],
455 ), // Row
456 ), // SizedBox
457
458 //Tiap 'ListTile' akan ada lingkaran/'CircleAvatar' yang dalamnya adalah huruf awal dari 'name'
459 //'_contacts[index]['name']' fungsinya untuk mengambil nilai yang berhubungan dengan 'name' dalam Map
460 //Ada tanda seru '!' dibelakang ['name'] menunjukkan bahwa nilai yang diambil dari '_contacts[index]' tidak null
461 leading: CircleAvatar(
462   child: Text(getFirstLetter(_contacts[index]['name']!)),
463 ), // CircleAvatar
464 title: Text(_contacts[index]['name']!),
465 subtitle: Text(_contacts[index]['phone']!),
466 ); // ListTile
467 },
468
469 //separatorBuilder' untuk menambahkan widget pemisah antara setiap item dalam daftar
470 //const Divider() sebagai pemisah antara setiap item dalam daftar
471 //itemCount' untuk menentukan jumlah item dalam daftar, dalam kasus ini semua list '_contacts'
472 separatorBuilder: (context,index)>const Divider(),
473 itemCount: _contacts.length,
474 ), // ListView.separated
475 ), // Container
476

```

Ln 457, Col 1, Spaces: 2, UTF-8, CRLF { Dart Pixel 6 Pro API 29 (android-x86 emulator)

```

459 //'_contacts[index]['name']' fungsinya untuk mengambil nilai yang berhubungan dengan 'name' dalam Map
460 //Ada tanda seru '!' dibelakang '['name']' menunjukkan bahwa nilai yang diambil dari '_contacts[index]' tidak null
461 leading: CircleAvatar(
462   child: Text(getFirstLetter(_contacts[index]['name']!)),
463 ), // CircleAvatar
464 title: Text(_contacts[index]['name']!),
465 subtitle: Text(_contacts[index]['phone']!),
466 ); // ListTile
467
468
469 //'separatorBuilder' untuk menambahkan widget pemisah antara setiap item dalam daftar
470 //'const Divider()' sebagai pemisah antara setiap item dalam daftar
471 //'itemCount' untuk menentukan jumlah item dalam daftar, dalam kasus ini semua list '_contacts'
472 separatorBuilder: (context, index)=>const Divider(),
473 itemCount: _contacts.length,
474 ), // ListView.separated
475 ), // Container
476 ],
477 ), // Column
478 ], // Container
479 ), // Form
480 ), // SingleChildScrollView
481 ); // Scaffold
482 }
483 }

```

