

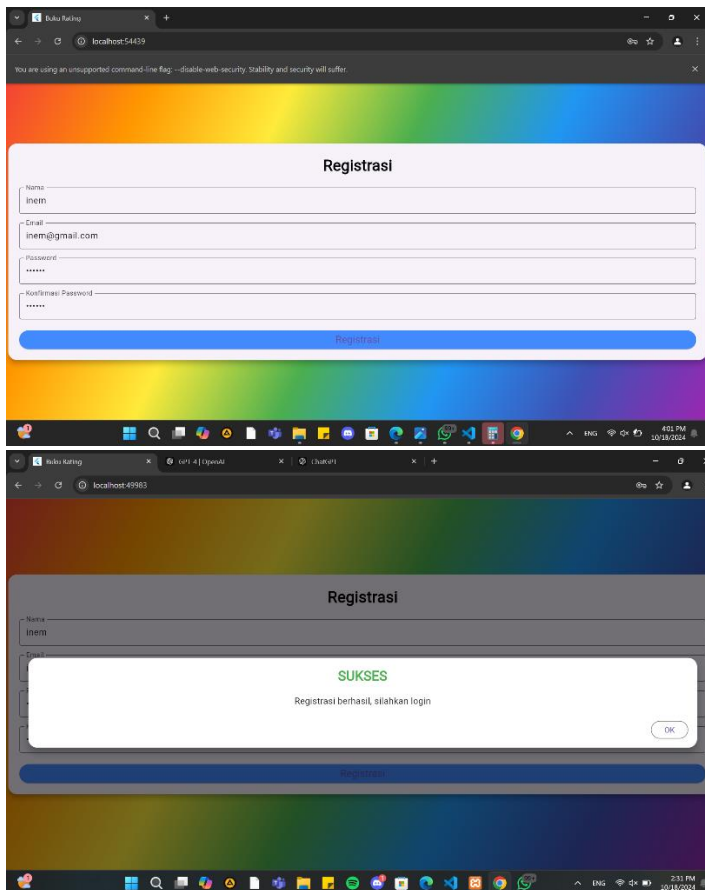
Nama: Syifa Rahmadani Hemi S

NIM: H1D022096

Responsi Praktikum Mobile. Paket Buku Table Rating

## 1. Registrasi

```
void _submit() {
  _formKey.currentState!.save();
  setState(() {
    _isLoading = true;
  });
  RegistrasiBloc.registrasi(
    nama: _namaTextboxController.text,
    email: _emailTextboxController.text,
    password: _passwordTextboxController.text)
    .then((value) {
      showDialog(
        context: context,
        barrierDismissible: false,
        builder: (BuildContext context) => SuccessDialog(
          description: "Registrasi berhasil, silahkan login",
          onClick: () {
            Navigator.pop(context);
          },
        ),
      );
    }, onError: (error) {
      showDialog(
        context: context,
        barrierDismissible: false,
        builder: (BuildContext context) => const WarningDialog(
          description: "Registrasi gagal, silahkan coba lagi",
        ),
      );
    }).whenComplete(() {
      setState(() {
        _isLoading = false;
      });
    });
}
```



### Penjelasan:

- Fungsi `_submit()` dipanggil saat tombol registrasi ditekan.
- `_formKey.currentState!.save()` menyimpan state form saat ini.
- `setState(() { _isLoading = true; })` mengubah state untuk menampilkan indikator loading.
- `RegistrasiBloc.registrasi()` dipanggil dengan data dari form.
- Kalau registrasi berhasil, dialog pesan sukses dimunculin.
- Kalau registrasi gagal, dialog peringatan dimunculin.
- State loading diubah menjadi false setelah proses selesai.

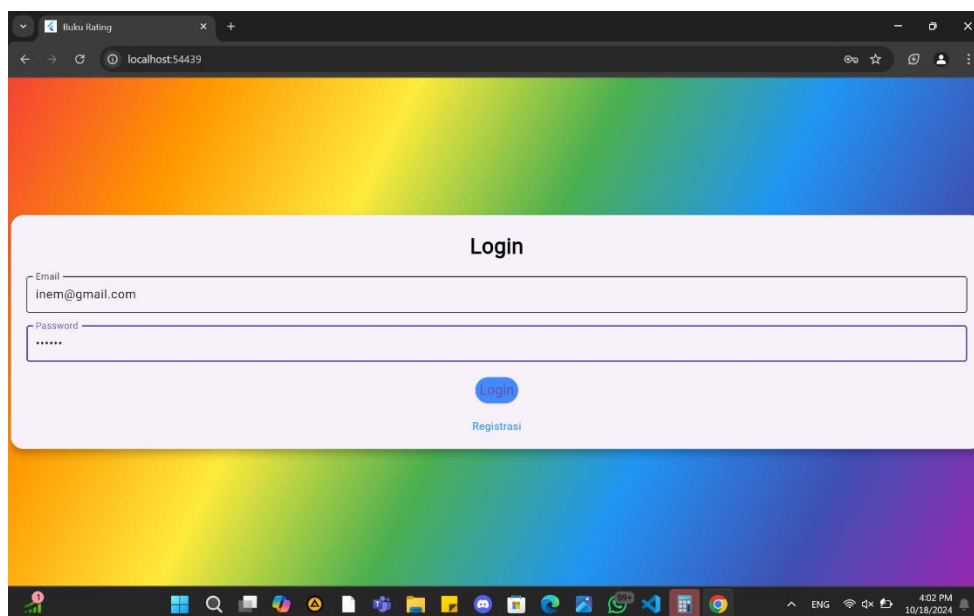
### 2. Login

```
void _submit() {
  _formKey.currentState!.save();
  setState(() {
    _isLoading = true;
  });
  LoginBloc.login(
    email: _emailTextboxController.text,
    password: _passwordTextboxController.text
  ).then((value) async {
    if (value.code == 200) {
      print(value.userID);
      await UserInfo().setToken(value.token!);
      await UserInfo().setUserID(value.userID!);
      Navigator.pushReplacement(context,
```

```

        MaterialPageRoute(builder: (context) => const RatingPage()));
    } else {
      showDialog(
        context: context,
        barrierDismissible: false,
        builder: (BuildContext context) => const WarningDialog(
          description: "Login gagal, silahkan coba lagi",
        )),
    }
  }, onError: (error) {
    print(error);
    showDialog(
      context: context,
      barrierDismissible: false,
      builder: (BuildContext context) => const WarningDialog(
        description: "Login gagal, silahkan coba lagi",
      )),
  }).whenComplete(() {
    setState(() {
      _isLoading = false;
    });
  });
}
}

```



## Penjelasan

- Fungsi `_submit()` dipanggil saat menekan tombol login.
- `_formKey.currentState!.save()` buat menyimpan state form saat ini.
- `setState(() { _isLoading = true; })` buat mengubah state untuk menampilkan indikator loading.
- `LoginBloc.login()` ini dipanggil dengan email dan password dari controller.

- Kalau loginnya berhasil (kode 200), token dan userID disimpan, lalu navigasi ke halaman produk.
- Kalau loginnya gagal, dialog peringatan dimunculin.
- Lalu yang terakhir, state loading diubah kembali menjadi false baik loginnya berhasil ataupun tidak.

### 3. Read

```
class _RatingDetailState extends State<RatingDetail> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: [
              Colors.red,
              Colors.orange,
              Colors.yellow,
              Colors.green,
              Colors.blue,
              Colors.indigo,
              Colors.purple,
            ],
            begin: Alignment.topLeft,
            end: Alignment.bottomRight,
          ),
        ),
        child: Center(
          child: SingleChildScrollView(
            child: Card(
              elevation: 8,
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(16),
              ),
              margin: const EdgeInsets.all(16.0),
              child: Padding(
                padding: const EdgeInsets.all(20.0),
                child: Column(
                  mainAxisAlignment: MainAxisAlignment.min,
                  children: [
                    Text(
                      'Detail Rating',
                      style: TextStyle(
                        fontSize: 24,
                        fontWeight: FontWeight.bold,
                        color: Colors.black,
                      ),
                    ),
                  ],
                ),
              ),
            ),
          ),
        ),
      ),
    );
  }
}
```

```

        const SizedBox(height: 20),
        _buildDetailText(
          "Average Rating: ${widget.rating!.averageRating}",
          20.0),
        _buildDetailText(
          "Total Reviews: ${widget.rating!.totalRating}", 18.0),
        _buildDetailText(
          "Best Seller Rank: ${widget.rating!.bestRating}", 18.0),
        const SizedBox(height: 20),
        _tombolHapusEdit(),
      ],
    ),
  ),
),
),
),
),
),
),
);
}

```

```

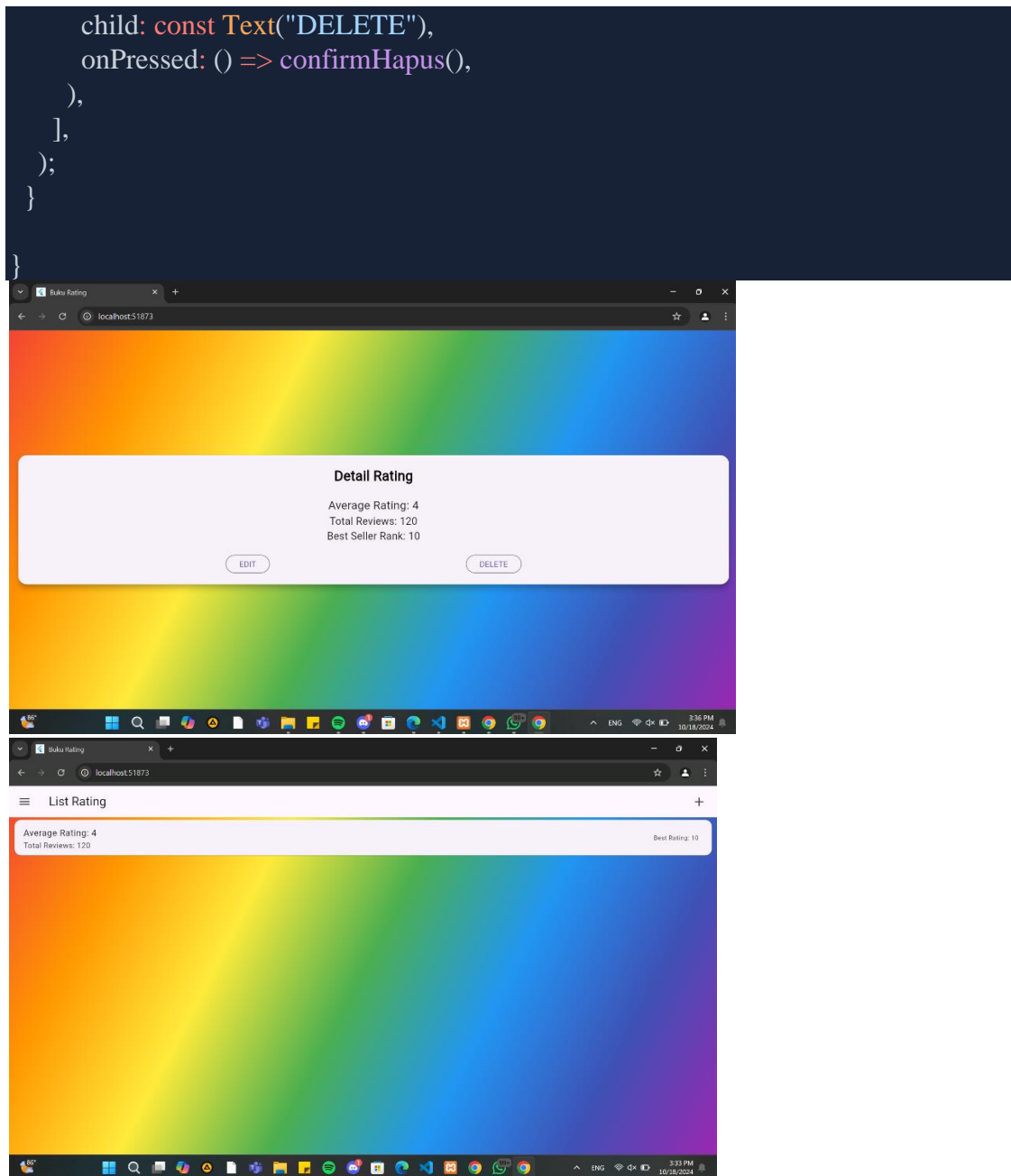
Widget _buildDetailText(String text, double fontSize) {
  return Text(
    text,
    style: TextStyle(fontSize: fontSize),
    textAlign: TextAlign.center,
  );
}

```

```

Widget _tombolHapusEdit() {
  return Row(
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,
    children: [
      // Tombol Edit
      OutlinedButton(
        child: const Text("EDIT"),
        onPressed: () {
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => RatingForm(
                rating: widget.rating!,
              ),
            ),
          );
        },
      ),
      // Tombol Hapus
      OutlinedButton(

```



## Penjelasan

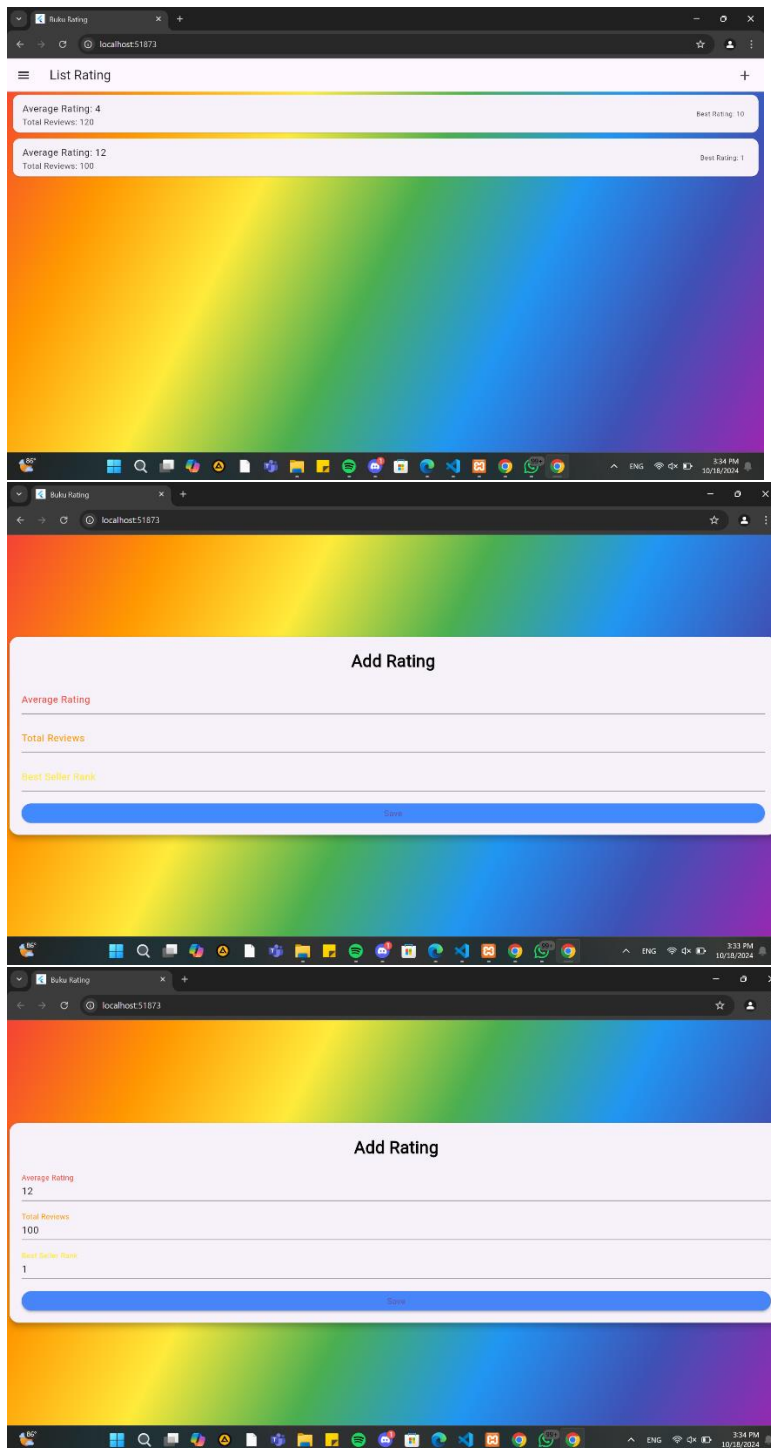
- RatingDetail adalah widget stateful yang menerima objek Rating.
- build method membuat tampilan dengan Scaffold yang memiliki AppBar dengan title "Detail Rating".
- Body Scaffold berisi Column yang menampilkan:
  - Average Produk dengan font size 20.0
  - Total Reviews dengan font size 18.0
  - Best Seller Rank dengan font size 18.0
- \_tombolHapusEdit() menambahkan tombol untuk mengedit atau menghapus rating.
- Halaman detail mengakses data rating melalui widget.rating dan menampilkannya.
- Data ditampilkan langsung tanpa request jaringan tambahan, karena semua info sudah ada dalam objek Rating yang diteruskan.

#### 4. Create

```
void simpan() {
  setState(() {
    _isLoading = true;
  });

  Rating createRating = Rating(id: null);
  createRating.averageRating =
    int.tryParse(_averageRatingTextboxController.text) ?? 0;
  createRating.totalRating =
    int.tryParse(_totalRatingTextboxController.text) ?? 0;
  createRating.bestRating =
    int.tryParse(_bestRatingTextboxController.text) ?? 0;

  RatingBloc.addRating(rating: createRating).then((value) {
    Navigator.of(context).push(MaterialPageRoute(
      builder: (BuildContext context) => const RatingPage(),
    ));
  }, onError: (error) {
    showDialog(
      context: context,
      builder: (BuildContext context) => const WarningDialog(
        description: "Simpan gagal, silahkan coba lagi",
      ),
    );
  }).whenComplete(() {
    setState(() {
      _isLoading = false;
    });
  });
}
```



## Penjelasan

- Fungsi `simpan()` dipanggil saat menekan tombol simpan.
- State loading diubah jadi `true`.
- Objek Rating baru dibuat dengan data dari text controller.
- `RatingBloc.addRating()` dipanggil buat nyimpen rating baru.
- Kalau berhasil, pindah ke halaman list rating.

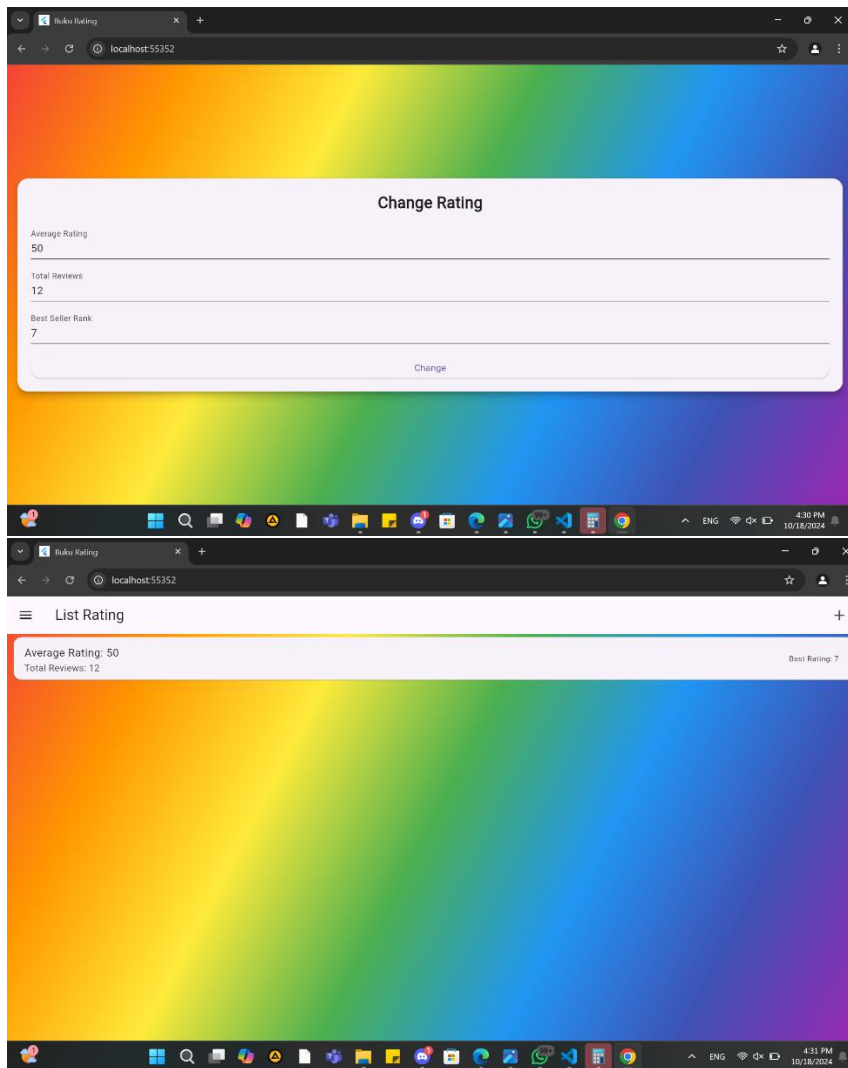


- Kalau gagal, dialog peringatan dimunculin.
- Terakhir, state loading diubah lagi menjadi false.

## 5. Update

```
void ubah() {
  setState(() {
    _isLoading = true;
  });
  Rating updateRating = Rating(id: widget.rating!.id!);
  updateRating.averageRating =
    int.parse(_averageRatingTextboxController.text);
  updateRating.totalRating = int.parse(_totalRatingTextboxController.text);
  updateRating.bestRating = int.parse(_bestRatingTextboxController.text);

  RatingBloc.updateRating(rating: updateRating).then((value) {
    Navigator.of(context).push(MaterialPageRoute(
      builder: (BuildContext context) => const RatingPage(),
    ));
  }, onError: (error) {
    showDialog(
      context: context,
      builder: (BuildContext context) => const WarningDialog(
        description: "Permintaan ubah data gagal, silahkan coba lagi",
      ),
    );
  }).whenComplete(() {
    setState(() {
      _isLoading = false;
    });
  });
}
```



### Penjelasan:

- Fungsi `ubah()` dipanggil saat menekan tombol ubah.
- State `loading` diubah menjadi `true`.
- Objek `Rating` dibuat dengan ID yang sudah ada dan data baru dari text controller.
- `RatingBloc.updateRating()` akan dipanggil buat memperbarui rating.
- Kalau berhasil, pindah navigasi ke halaman list rating.
- Kalau gagal, dialog peringatan dimunculin.
- Terakhir, state `loading` diubah lagi jadi `false`.

### 6. Delete

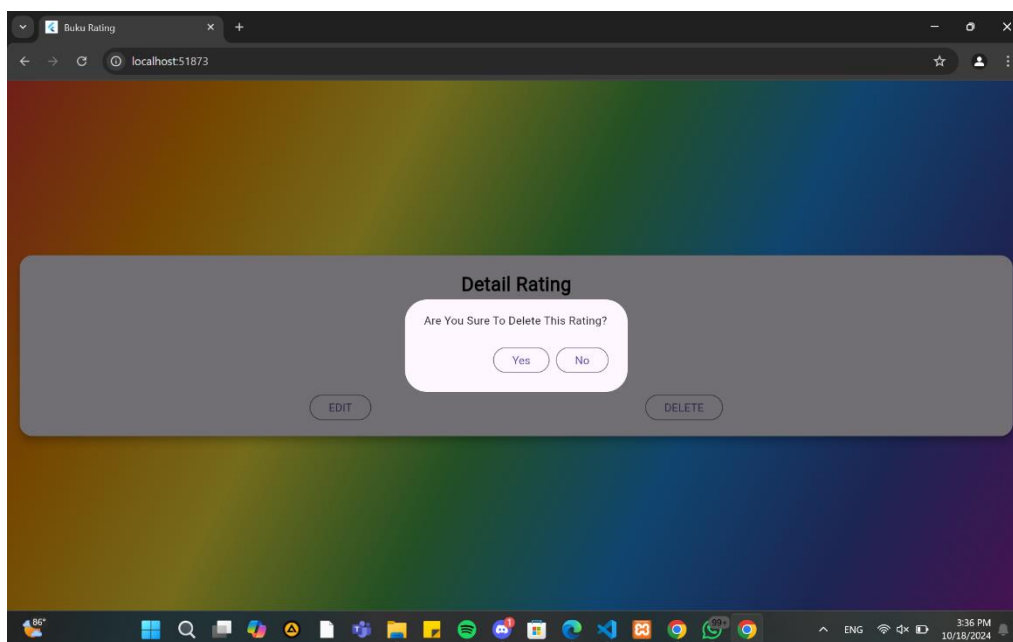
```
void confirmHapus() {
  AlertDialog alertDialog = AlertDialog(
    content: const Text("Are You Sure To Delete This Rating?"),
    actions: [
      // Tombol hapus
      OutlinedButton(
```

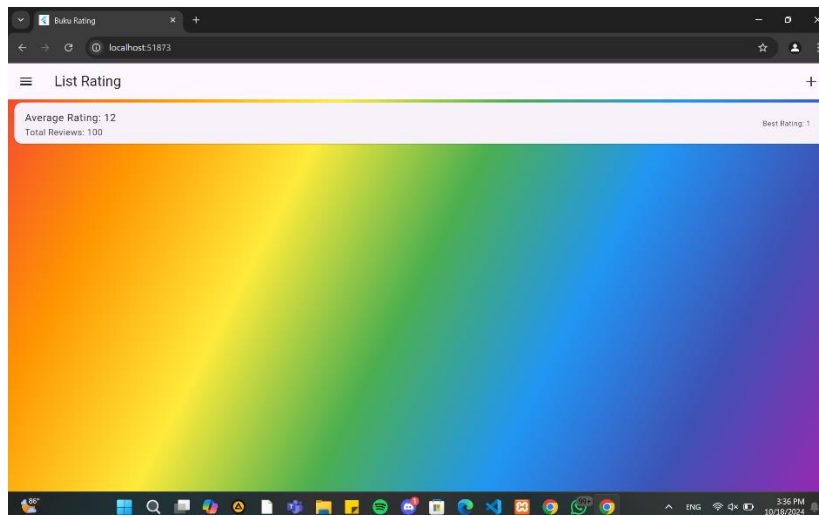
```

child: const Text("Yes"),
onPressed: () {
  RatingBloc.deleteRating(id: widget.rating!.id!).then((value) {
    Navigator.of(context).push(MaterialPageRoute(
      builder: (context) => const RatingPage(),
    ));
  }, onError: (error) {
    showDialog(
      context: context,
      builder: (BuildContext context) => const WarningDialog(
        description: "Hapus gagal, silahkan coba lagi",
      ),
    );
  });
},
),
// Tombol batal
OutlinedButton(
  child: const Text("No"),
  onPressed: () => Navigator.pop(context),
),
],
);

showDialog(builder: (context) => alertDialog, context: context);
}
}

```





### Penjelasan:

Kode ini buat menghapus rating menampilkan dialog konfirmasi sebelum melakukan penghapusan. Jika dikonfirmasi, `RatingBloc.deleteRating()` akan dipanggil buat menghapus rating dari database