

TUGAS AKHIR PEMROGRAMAN MOBILE II
APLIKASI KELOLA GUDANG BERBASIS MOBILE

Mata Kuliah : Pemrograman Mobile II

Kelas : 7P51



Disusun oleh:

21.230.0060 Muhammad Bangga Pramudya

21.230.0062 Hafid Firman Febrian

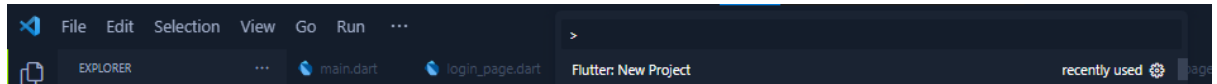
21.230.0064 Adnan Rahmawan

INSTITUT WIDYA PRATAMA

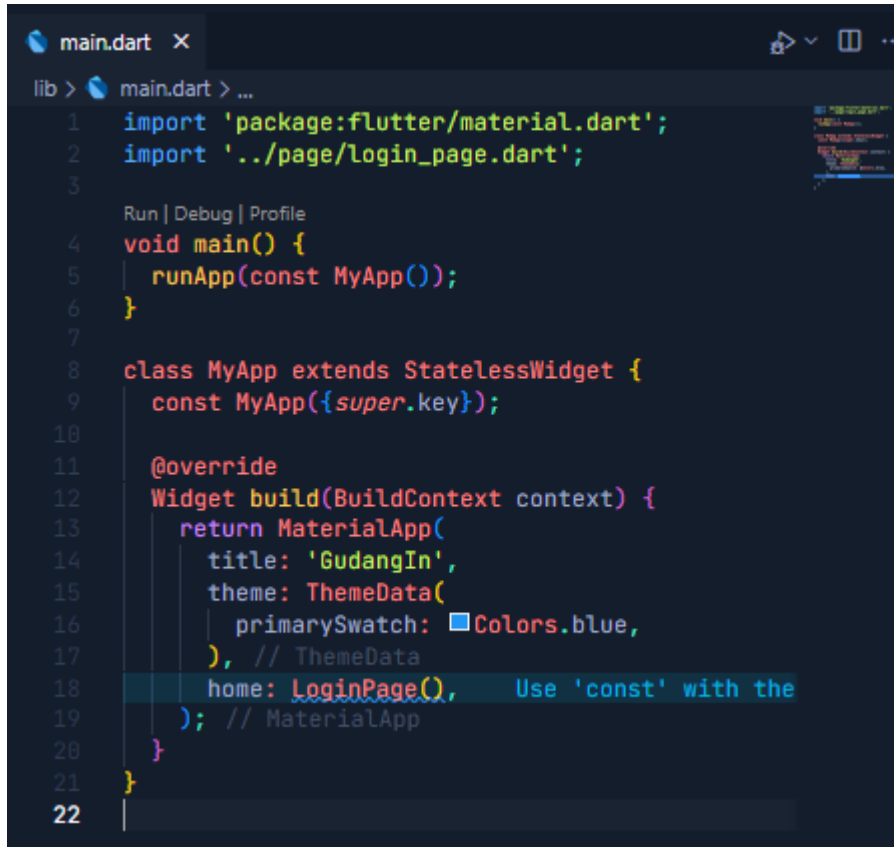
2024

Langkah – langkah :

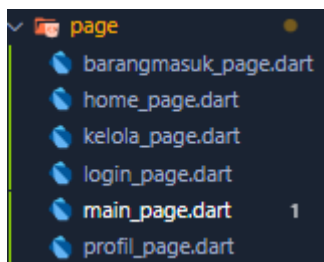
1. Membuat proyek flutter baru dengan cara masuk kedalam VS Code, klik menu view, pilih command pallette, lalu pilih opsi flutter : new project



2. Setelah project baru terbentuk sesuaikan file main.dart menjadi yang diinginkan



3. Membuat file-file yang akan digunakan yaitu file home, barang masuk, kelola, main page, dan profil



4. Membuat file main page yang digunakan untuk menempatkan navigasi dalam hal ini kelompok kami menggunakan bottom navigation

```

class _MainPageState extends Type: List<Widget>
  int _selectedIndex = 0;
  package:ta_mobile/page/main_page.dart

  static final List<Widget> _pages = <Widget>[
    HomePage(), Use 'const' with the constructor to improve performa
    BarangmasukPage(), Use 'const' with the constructor to improve p
    ProfilPage(),
  ]; // <Widget>[]

  void _onItemTapped(int index) {
    setState(() {
      _selectedIndex = index;
    });
  }

```

- **_selectedIndex** digunakan untuk menentukan halaman utama yang akan diakses saat aplikasi pertama kali dibuka
- Pada bagian static final list berisi nama kelas dari tampilan yang akan dipanggil secara berurutan, dengan urutan Homepage(), BarangMasuk(), dan ProfilPage()

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Center(
      child: _pages.elementAt(_selectedIndex),
    ), // Center
    bottomNavigationBar: BottomNavigationBar(
      items: const <BottomNavigationBarItem>[
        BottomNavigationBarItem(
          icon: Icon(Icons.home),
          label: 'Home',
        ), // BottomNavigationBarItem
        BottomNavigationBarItem(
          icon: Icon(Icons.add_business_rounded),
          label: 'Masuk',
        ), // BottomNavigationBarItem
        BottomNavigationBarItem(
          icon: Icon(Icons.person),
          label: 'Profil',
        ), // BottomNavigationBarItem
      ], // <BottomNavigationBarItem>[]
      currentIndex: _selectedIndex,
      selectedItemColor: Colors.blue[800],
      onTap: _onItemTapped,
    ), // BottomNavigationBar
  ); // Scaffold
}

```

- Widget Bottom Navigation Bar diisi dengan 3 menu navigasi yaitu navigasi tampilan, Home, Barang Masuk, dan Profil
- Dalam Widget Bottom navigation Bar terdapat widget ikon yang berisi ikon dari masing, masing navigasi untuk mempertegas dan memperjelas masing-masing menu.

5. Membuat tampilan home_page.dart

```

import 'package:auto_animated/auto_animated.dart';
import 'package:flutter/material.dart';
import '../helper/api_service.dart';
import '../model/barang.dart';

class HomePage extends StatefulWidget {
  const HomePage({super.key});

  @override
  _HomePageState createState() => _HomePageState(); Invalid use of a
}

class _HomePageState extends State<HomePage> {
  final ApiService apiService = ApiService();
  late Future<List<Barang>> barangList;

  @override
  void initState() {
    super.initState();
    barangList = apiService.fetchBarang();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Stok Barang'),
        backgroundColor: Colors.blue.shade300,
        foregroundColor: Colors.white,
      ), // AppBar
      body: FutureBuilder<List<Barang>>(
        future: barangList,
        builder: (context, snapshot) {
          if (snapshot.connectionState == ConnectionState.waiting) {
            return const Center(child: CircularProgressIndicator());
          }
          if (snapshot.hasError) {
            return Center(
              child: Text('Error: ${snapshot.error}'),
            ); // Center

```

[illegible]

```

81     backgroundColor: Colors.blueAccent,
82     child: Text(
83       barang.namaBarang[0]
84         .toUpperCase(), // Inisial Bar
85       style: const TextStyle(
86         color: Colors.white,
87         fontWeight: FontWeight.bold,
88       ), // TextStyle
89     ), // Text
90   ), // CircleAvatar
91   title: Text(
92     barang.namaBarang,
93     style: const TextStyle(
94       fontSize: 16, // Ukuran lebih bes
95       fontWeight:
96         FontWeight.w600, // Bold ring
97       color: Colors.black87,
98     ), // TextStyle
99   ), // Text
100   subtitle: Text(
101     'Stok: ${barang.stok}',
102     style: TextStyle(
103       fontSize:
104         14, // Ukuran lebih kecil dar
105       color: Colors.grey.shade700, //
106     ), // TextStyle
107   ), // Text
108 ), // ListTile
109 ), // Card
110 ), // Padding
111 if (index < barangList.length - 1)
112   const Padding(
113     padding: EdgeInsets.symmetric(horizonta
114     child: Divider(
115       thickness: 1.2,
116       color: Colors.grey, // Warna abu ur
117     ), // Divider
118   ), // Padding
119 ],
120 ), // Column

```

```
lib > page > home_page.dart > _HomePageState > build
123     },
124     ), // LiveList.options
125     Positioned(
126       bottom: 16.0,
127       right: 16.0,
128       child: FloatingActionButton(
129         onPressed: () {
130           // =====
131         },
132         backgroundColor: Colors.blue.shade800,
133         foregroundColor: Colors.white,
134         child: Icon(Icons.add), // Use 'const' with the con
135       ), // FloatingActionButton
136     ), // Positioned
137   ],
138 ); // Stack
139 } else {
140   return const Center(child: Text('No data available'));
141 }
142 },
143 ), // FutureBuilder
144 ); // Scaffold
145 }
146 }
147 }
```

- Membuat tampilan dengan kerangka widget Scaffold sebagai dasar
- Widget appBar digunakan sebagai header atau judul aplikasi
- Widget future builder digunakan sebagai tempat widget list nantinya
untuk meload data dari database
- Widget padding digunakan untuk mengatur tampilan dari widget ListTile
- barang.namabarang digunakan untuk menampilkan nama barang dari field
pada database
- \$ {barang.stok} digunakan untuk menampilkan data stok dari field stok
database

- Elevated Button digunakan untuk membuat tombol untuk mengakses halaman input

6. Membuat tampilan barangmasuk_page.dart

```

7. import 'package:auto_animated/auto_animated.dart';
8. import 'package:flutter/material.dart';
9. import '../helper/api_service.dart';
10. import '../model/barang.dart';
11.
12. class BarangmasukPage extends StatefulWidget {
13.   const BarangmasukPage({super.key});
14.
15.   @override
16.   _BarangmasukPageState createState() => _BarangmasukPageState();
17. }
18.
19. class _BarangmasukPageState extends State<BarangmasukPage> {
20.   final ApiService apiService = ApiService();
21.   late Future<List<BarangMasuk>> barangMasukList;
22.
23.   @override
24.   void initState() {
25.     super.initState();
26.     barangMasukList = apiService.fetchMasuk();
27.   }
28.
29.   @override
30.   Widget build(BuildContext context) {
31.     return Scaffold(
32.       appBar: AppBar(
33.         title: const Text('Stok Barang Masuk'),
34.         backgroundColor: Colors.blue.shade300,
35.         foregroundColor: Colors.white,
36.       ),
37.       body: FutureBuilder<List<BarangMasuk>>(
38.         future: barangMasukList,
39.         builder: (context, snapshot) {
40.           if (snapshot.connectionState == ConnectionState.waiting) {
41.             return const Center(child: CircularProgressIndicator());
42.           }
43.           if (snapshot.hasError) {
44.             return Center(
45.               child: Text('Error: ${snapshot.error}'),
46.             );
47.           }
48.           if (snapshot.hasData) {
49.             final barangMasukList = snapshot.data!;

```



```

98.         barang.namaBarang,
99.         style: const TextStyle(
100.             fontSize: 16, // Ukuran lebih besar
101.             fontWeight:
102.                 FontWeight.w600, // Bold ringan
103.             color: Colors.black87,
104.         ),
105.     ),
106.     subtitle: Text(
107.         'Stok ${barang.jumlah}, ${barang.tanggal}',
108.         style: TextStyle(
109.             fontSize:
110.                 14, // Ukuran lebih kecil dari nama
111.             color: Colors.grey.shade700, // Warna soft
112.         ),
113.     ),
114. ),
115. ),
116. ),
117. if (index < barangMasukList.length - 1)
118.     const Padding(
119.         padding: EdgeInsets.symmetric(horizontal: 20),
120.         child: Divider(
121.             thickness: 1.2,
122.             color: Colors.grey, // Warna abu untuk pemisah
123.         ),
124.     ),
125. ],
126. ),
127. ),
128. );
129. },
130. ),
131. Positioned(
132.     bottom: 16.0,
133.     right: 16.0,
134.     child: FloatingActionButton(
135.         onPressed: () {
136.             // =====
137.         },
138.         backgroundColor: Colors.blue.shade800,
139.         foregroundColor: Colors.white,
140.         child: Icon(Icons.add),
141.     ),
142. ),
143. ],
144. );
145. } else {

```

```

146.         return const Center(child: Text('No data available'));
147.     }
148. },
149. ),
150. );
151. }
152. }
153.

```

- Membuat tampilan dengan kerangka widget Scaffold sebagai dasar
- Widget appBar digunakan sebagai header atau judul aplikasi
- Widget future builder digunakan sebagai tempat widget list nantinya untuk meload data dari database
- Widget padding digunakan untuk mengatur tampilan dari widget ListTile
- barang.namabarang digunakan untuk menampilkan nama barang dari field pada database
- \$ {barang.stok} digunakan untuk menampilkan data stok dari field stok database
- Elevated Button digunakan untuk membuat tombol untuk mengakses halaman input

7. Membuat tampilan profil_page.dart

```

8. import 'package:flutter/material.dart';
9.
10. class ProfilPage extends StatelessWidget {
11.   @override
12.   Widget build(BuildContext context) {
13.     return Scaffold(
14.       appBar: AppBar(
15.         title: Text("Profil"),
16.         backgroundColor: Colors.blue.shade300,
17.       ),
18.       body: Center(
19.         child: Column(
20.           mainAxisAlignment: MainAxisAlignment.center,
21.           crossAxisAlignment: CrossAxisAlignment.center,
22.           children: [

```

```

23. // Foto Profil
24. CircleAvatar(
25.   radius: 50,
26.   backgroundImage: AssetImage(
27.     'assets/profile.jpg'), // Ganti dengan gambar yang sesuai
28. ),
29. SizedBox(height: 12),
30. // Nama & Email
31. Text(
32.   "User",
33.   style: TextStyle(fontSize: 22, fontWeight: FontWeight.bold),
34. ),
35. Text(
36.   "xxx@email.com",
37.   style: TextStyle(fontSize: 16, color: Colors.grey),
38. ),
39. ],
40. ),
41. ),
42. );
43. }
44.
45. // Widget untuk menampilkan informasi profil dengan ikon
46. }
47.

```

- Membuat tampilan dengan kerangka widget Scaffold sebagai dasar
- Widget appBar digunakan sebagai header atau judul aplikasi
- Widget Column digunakan untuk mengatur tata letak tampilan menjadi rata tengah / center dengan mainAxisAlignment dan crossAxisAlignment
- Widget circleavatar digunakan untuk memberikan foto profil berupa avatar yang berasal dari folder assets

8. Membuat login_page.dart

```

9. import 'package:flutter/material.dart';
10. import '../helper/api_service.dart';
11. import 'main_page.dart';
12.
13. class LoginPage extends StatefulWidget {
14.   const LoginPage({super.key});
15.
16.   @override

```

```

17. LoginPageState createState() => LoginPageState();
18. }
19.
20. class LoginPageState extends State<LoginPage> {
21.   final _usernameController = TextEditingController();
22.   final _passwordController = TextEditingController();
23.
24.   final ApiService apiService = ApiService();
25.
26.   Future<void> login() async {
27.     try {
28.       final data = await apiService.login(
29.         _usernameController.text,
30.         _passwordController.text,
31.       );
32.       if (data['status'] == true) {
33.         Navigator.pushReplacement(
34.           context,
35.           MaterialPageRoute(builder: (context) => const MainPage()),
36.         );
37.         ScaffoldMessenger.of(context).showSnackBar(SnackBar(
38.           content: Text('Login sukses: ${data['data']['username']}'),
39.           backgroundColor: Colors.green,
40.         ));
41.       }
42.       if (data['status'] == false) {
43.         ScaffoldMessenger.of(context).showSnackBar(SnackBar(
44.           content: Text('Error: ${data['message']}'),
45.           backgroundColor: Colors.red,
46.         ));
47.       }
48.     } catch (error) {
49.       ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
50.         content: Text('Error: gagal terhubung ke server'),
51.         backgroundColor: Colors.red,
52.       ));
53.     }
54.   }
55.
56.   @override
57.   Widget build(BuildContext context) {
58.     return Scaffold(
59.       body: Padding(
60.         padding: EdgeInsets.all(16.0),
61.         child: Column(
62.           mainAxisAlignment: MainAxisAlignment.center,
63.           crossAxisAlignment: CrossAxisAlignment.center,
64.           children: [

```

```

65.     TextField(
66.         controller: _usernameController,
67.         decoration: InputDecoration(
68.             border: OutlineInputBorder(
69.                 borderRadius: BorderRadius.circular(12.0),
70.             ),
71.             labelText: 'Username'),
72.     ),
73.     SizedBox(
74.         height: 10,
75.     ),
76.     TextField(
77.         controller: _passwordController,
78.         decoration: InputDecoration(
79.             border: OutlineInputBorder(
80.                 borderRadius: BorderRadius.circular(12.0),
81.             ),
82.             labelText: 'Password'),
83.         obscureText: true,
84.     ),
85.     SizedBox(height: 20),
86.     ElevatedButton(
87.         style: ElevatedButton.styleFrom(
88.             minimumSize: Size.fromHeight(40),
89.             backgroundColor: Colors.blue,
90.             foregroundColor: Colors.white),
91.         onPressed: login,
92.         child: Text('Login'),
93.     ),
94. ],
95. ),
96. ),
97. );
98. }
99. }
100.

```

- Membuat tampilan dengan kerangka widget Scaffold sebagai dasar
- Widget appBar digunakan sebagai header atau judul aplikasi
- Widget Column digunakan untuk mengatur tata letak tampilan menjadi rata tengah / center dengan mainAxisAlignment dan crossAxisAlignment
- TextField adalah form yang digunakan untuk menampung input yang didalamnya terdapat _usernameController, dan _passwordController dan mengirimnya untuk dilakukan verifikasi

- Widget SizedBox digunakan untuk mengatur jarak antara widget
- Widget ElevatedButton adalah bentuk tombol yang digunakan untuk mengeksekusi fungsi login.

Langkah-langkah pembuatan API

1. Route

```
 9 // login
10 $routes->post('/api/login', 'AuthController::login');
11
12 // user
13 $routes->post('/api/user', 'AuthController::getProfile');
14
15 $routes->post('/api/barang/add', 'BarangController::barangAdd');
16 $routes->delete('/api/barang/delete/(:num)', 'BarangController::hapusBarang/$1');
17 // barang - stok
18 $routes->get('/api/stok', 'BarangController::showStok');
19 $routes->post('/api/stok/satuan', 'BarangController::stokSatuan');
20
21 // barang - masuk
22 $routes->get('/api/b_masuk', 'BarangController::barangMasuk');
23 $routes->post('/api/b_masuk/add', 'BarangController::barangMasukAdd');
24
25 // barang - keluar
26 $routes->get('/api/b_keluar', 'BarangController::barangKeluar');
27 $routes->post('/api/b_keluar/add', 'BarangController::barangKeluarAdd');
28
```

2. Model

User

```
 6
 7 class UserModel extends Model
 8 {
 9     protected $table = 'users';
10     protected $primaryKey = 'id';
11     protected $allowedFields = ['username', 'password', 'created_at'];
12
13     public function getUserByUsername($username)
14     {
15         return $this->where('username', $username)->first();
16     }
17 }
18
```

stok

```
6
7  class StokModel extends Model
8  {
9      protected $table = 'stok';
10     protected $primaryKey = 'id_barang';
11     protected $allowedFields = ['nama_barang', 'stok'];
12
13     public function getId($id)
14     {
15         return $this->where('id_barang', $id)->first();
16     }
17 }
18
```

Barang Masuk

```
6
7  class MasukModel extends Model
8  {
9      protected $table = 'barang_masuk';
10     protected $primaryKey = 'id_masuk';
11     protected $allowedFields = ['id_barang', 'jumlah', 'tanggal'];
12
13     public function getAllBarang()
14     {
15         return $this->join('stok', 'stok.id_barang = barang_masuk.id_barang')
16             ->select('barang_masuk.*')
17             ->select('stok.nama_barang as nama_barang')
18             ->findAll();
19     }
20 }
21
```

Barang Keluar

```

6
7 class KeluarModel extends Model
8 {
9     protected $table = 'barang_keluar';
10    protected $primaryKey = 'id_keluar';
11    protected $allowedFields = ['id_barang', 'jumlah', 'tanggal'];
12
13    public function getAllBarang()
14    {
15        return $this->join('stok', 'stok.id_barang = barang_keluar.id_barang')
16            ->select('barang_keluar.*')
17            ->select('stok.nama_barang as nama_barang')
18            ->findAll();
19    }
20 }
21

```

3. Controller

Untuk mengambil semua data barang

```

public function showStok()
{
    $allbarang = $this->stokModel->findAll();

    $barang = [];
    foreach ($allbarang as $a) {
        $barang[] = [
            'id_barang' => $a['id_barang'],
            'nama_barang' => $a['nama_barang'],
            'stok' => $a['stok'],
        ];
    }

    return $this->respond($barang, 200);
}

```

Untuk menampilkan data barang secara satuan

```
public function stokSatuan()
{
    $input = $this->request->getJSON();
    $id = $input->id_barang;

    if (!$input || !$id) {
        return $this->respond([
            'status' => false,
            'message' => 'id_barang harus diisi'
        ], 400);
    }

    $model = $this->stokModel->where('id_barang', $id)->first();

    $barang = [
        'id_barang' => $model['id_barang'],
        'nama_barang' => $model['nama_barang'],
        'stok' => $model['stok'],
    ];

    return $this->respond($barang, 200);
}
```

Untuk menampilkan barang masuk

```
public function barangMasuk()
{
    $allbarang = $this->masukModel->getAllBarang();

    $barang = [];

    foreach ($allbarang as $a) {
        $barang[] = [
            'id_masuk' => $a['id_masuk'],
            'nama_barang' => $a['nama_barang'],
            'jumlah' => $a['jumlah'],
            'tanggal' => $a['tanggal'],
        ];
    }

    return $this->respond($barang, 200);
}
```

Untuk menambah barang

```
public function barangAdd()
{
    $input = $this->request->getJSON();
    $nama_barang = $input->nama_barang;

    if (!$input || !$nama_barang) {
        return $this->respond([
            'status' => false,
            'message' => 'nama barang harus diisi'
        ], 400);
    }

    $model = $this->stokModel->where('nama_barang', $nama_barang)->first();

    if ($model) {
        return $this->respond([
            'status' => false,
            'message' => 'nama barang sudah ada'
        ], 400);
    }

    $this->stokModel->insert([
        'nama_barang' => $nama_barang,
    ]);

    return $this->respond([
        'status' => true,
        'message' => 'Barang berhasil ditambahkan',
        'nama_barang' => $nama_barang,
    ], 200);
}
```

Untuk menghapus barang

```
public function hapusBarang($id)
{
    $model = $this->stokModel->where('id_barang', $id)->first();

    if (!$model) {
        return $this->respond([
            'status' => false,
            'message' => 'barang tidak ditemukan'
        ], 404);
    }

    $this->stokModel->delete($id);

    return $this->respond([
        'status' => true,
        'message' => 'Barang berhasil dihapus',
        'nama_barang' => $model['nama_barang'],
    ], 200);
}
```

Untuk menambah barang masuk

```
public function barangMasukAdd()
{
    $input = $this->request->getJSON();
    $id = $input->id_barang;
    $jumlah = $input->jumlah;

    if (!$input || !$id || !$jumlah) {
        return $this->respond([
            'status' => false,
            'message' => 'id_barang dan jumlah harus diisi'
        ], 400);
    }

    $model = $this->stokModel->where('id_barang', $id)->first();

    if (!$model) {
        return $this->respond([
            'status' => false,
            'message' => 'id_barang tidak ditemukan'
        ], 404);
    }

    $stok = $model['stok'] + $jumlah;

    $this->stokModel->update($id, ['stok' => $stok]);

    $this->masukModel->save([
        'id_barang' => $id,
        'jumlah' => $jumlah,
        'tanggal' => date('Y-m-d H:i:s')
    ]);

    return $this->respond([
        'status' => true,
        'message' => 'Barang berhasil ditambahkan',
        'id_barang' => $model['id_barang'],
        'nama_barang' => $model['nama_barang'],
        'stok' => $model['stok'],
    ], 200);
}
```

Untuk menampilkan barang keluar

```
public function barangKeluar()
{
    $allbarang = $this->keluarModel->getAllBarang();

    $barang = [];

    foreach ($allbarang as $a) {
        $barang[] = [
            'id_keluar' => $a['id_keluar'],
            'nama_barang' => $a['nama_barang'],
            'jumlah' => $a['jumlah'],
            'tanggal' => $a['tanggal'],
        ];
    }

    return $this->respond($barang, 200);
}
```

Untuk menambah barang keluar

```
public function barangKeluarAdd()
{
    $input = $this->request->getJSON();
    $id = $input->id_barang;
    $jumlah = $input->jumlah;

    if (!$input || !$id || !$jumlah) {
        return $this->respond([
            'status' => false,
            'message' => 'id_barang dan jumlah harus diisi'
        ], 400);
    }

    $model = $this->stokModel->where('id_barang', $id)->first();

    if (!$model) {
        return $this->respond([
            'status' => false,
            'message' => 'id_barang tidak ditemukan'
        ], 404);
    }

    $stok = $model['stok'] - $jumlah;

    $this->stokModel->update($id, ['stok' => $stok]);

    $this->keluarModel->save([
        'id_barang' => $id,
        'jumlah' => $jumlah,
        'tanggal' => date('Y-m-d H:i:s')
    ]);

    return $this->respond([
        'status' => true,
        'message' => 'Barang berhasil ditambahkan',
        'id_barang' => $model['id_barang'],
        'nama_barang' => $model['nama_barang'],
        'stok' => $model['stok'],
    ], 200);
}
```

Table	Action	Rows	Type
<input type="checkbox"/> barang_keluar	Browse Structure Search Insert Empty Drop	3	Int
<input type="checkbox"/> barang_masuk	Browse Structure Search Insert Empty Drop	5	Int
<input type="checkbox"/> stok	Browse Structure Search Insert Empty Drop	8	Int
<input type="checkbox"/> users	Browse Structure Search Insert Empty Drop	2	Int
4 tables	Sum	18	Int