

**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 projek flutter baru dengan cara masuk kedalam VS Code, klik menu view, pilih command palette, lalu pilih opsi flutter : new project

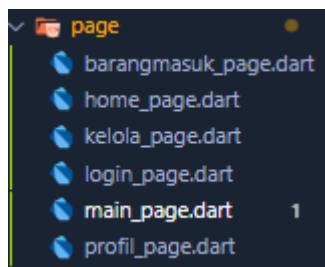


2. Setelah project baru terbentuk sesuaikan file main.dart menjadi yang diinginkan

A screenshot of the VS Code interface showing the 'main.dart' file. The code defines a main function that runs an MyApp widget. The MyApp widget extends StatelessWidget and uses MaterialApp to define the app's title, theme, and home page as LoginPage().

```
1 import 'package:flutter/material.dart';
2 import '../page/login_page.dart';
3
4 void main() {
5   runApp(const MyApp());
6 }
7
8 class MyApp extends StatelessWidget {
9   const MyApp({super.key});
10
11   @override
12   Widget build(BuildContext context) {
13     return MaterialApp(
14       title: 'GudangIn',
15       theme: ThemeData(
16         primarySwatch: Colors.blue,
17       ), // ThemeData
18       home: LoginPage(),    Use 'const' with the
19     ); // MaterialApp
20   }
21 }
22
```

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 State<MainPage> {
    int _selectedIndex = 0;
    static final List<Widget> _pages = <Widget>[
        HomePage(), // Use 'const' with the constructor to improve performance
        BarangmasukPage(), // Use 'const' with the constructor to improve performance
        ProfilPage(),
    ]; // <Widget>[]

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

- `_selected index` 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
                    }
                    else {
                        return ListView.builder(
                            itemCount: barangList.length,
                            itemBuilder: (context, index) {
                                return Card(
                                    elevation: 2,
                                    margin: const EdgeInsets.all(10),
                                    child: Column(
                                        mainAxisSize: MainAxisSize.min,
                                        children: [
                                            Row(
                                                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                                                children: [
                                                    Text(barangList[index].id.toString()),
                                                    Text(barangList[index].name),
                                                    Text(barangList[index].stock.toString())
                                                ],
                                            ),
                                            Row(
                                                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                                                children: [
                                                    Text('Edit'),
                                                    Text('Delete')
                                                ],
                                            )
                                        ],
                                    ),
                                );
                            },
                        );
                    }
                }
            )
        );
    }
}
```

① 29 ⌂ 0 ⌂

Spaces: 2 UTF-8 LF () Dart ⌂ Go Live Chrome (web-javascript)

∅ Prettier ⌂

```
42     if (snapshot.hasData) {
43         final barangList = snapshot.data!;
44
45         return Stack(
46             children: [
47                 LiveList.options(
48                     options: const LiveOptions(
49                         delay: Duration(milliseconds: 100),
50                         showItemInterval: Duration(milliseconds: 100),
51                         showItemDuration: Duration(milliseconds: 300),
52                         reAnimateOnVisibility: true,
53                     ), // LiveOptions
54                     itemCount: barangList.length,
55                     itemBuilder: (context, index, animation) {
56                         final barang = barangList[index];
57                         return FadeTransition(
58                             opacity: animation,
59                             child: SlideTransition(
60                                 position: Tween<Offset>(
61                                     begin: const Offset(0.1, 0), // Muncul dari samping
62                                     end: Offset.zero,
63                                 ).animate(animation), // Tween
64                                 child: Column(
65                                     children: [
66                                         Padding(
67                                             padding: const EdgeInsets.symmetric(
68                                                 horizontal: 10, vertical: 5), // Edge
69                                             child: Card(
70                                                 elevation: 3,
71                                                 shape: RoundedRectangleBorder(
72                                                     borderRadius: BorderRadius.circular(1
73                                                 ), // RoundedRectangleBorder
74                                                 child: ListTile(
75                                                     contentPadding: const EdgeInsets.symmetric(
76                                                         horizontal: 16,
77                                                         vertical: 8,
78                                                 ), // EdgeInsets.symmetric
79                                                     leading: CircleAvatar(
80                                                     backgroundColor: Colors.blueAccent
81                                             )
82                                         )
83                                     )
84                                 )
85                             )
86                         );
87                     );
88                 );
89             ],
90         );
91     }
92 }
```

```
81     backgroundColor: Colors.blueAccent,
82     child: Text(
83       barang.namaBarang[0]
84         .toUpperCase(), // Inisial Belanja
85       style: const TextStyle(
86         color: Colors.white,
87         fontWeight: FontWeight.bold,
88       ), // TextStyle
89     ), // CircleAvatar
90   title: Text(
91     barang.namaBarang,
92     style: const TextStyle(
93       fontSize: 16, // Ukuran lebih besar
94       fontWeight:
95         FontWeight.w600, // Bold ringan
96       color: Colors.black87,
97     ), // TextStyle
98   subtitle: Text(
99     'Stok: ${barang.stok}',
100    style: TextStyle(
101      fontSize:
102        14, // Ukuran lebih kecil daripada title
103      color: Colors.grey.shade700, //
104    ), // TextStyle
105  ), // Text
106  ), // ListTile
107  ), // Card
108  ), // Padding
109  if (index < barangList.length - 1)
110  const Padding(
111    padding: EdgeInsets.symmetric(horizontal: 10),
112    child: Divider(
113      thickness: 1.2,
114      color: Colors.grey, // Warna abu-abu
115    ), // Divider
116    ), // Padding
117  ],
118  ), // Column
```

```
lib > page > home_page.dart > _HomePageState > build
123           },
124           ),
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 cor
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!;
```

```
50.
51.     return Stack(
52.       children: [
53.         LiveList.options(
54.           options: const LiveOptions(
55.             delay: Duration(milliseconds: 100),
56.             showItemInterval: Duration(milliseconds: 100),
57.             showItemDuration: Duration(milliseconds: 300),
58.             reAnimateOnVisibility: true,
59.           ),
60.           itemCount: barangMasukList.length,
61.           itemBuilder: (context, index, animation) {
62.             final barang = barangMasukList[index];
63.             return FadeTransition(
64.               opacity: animation,
65.               child: SlideTransition(
66.                 position: Tween<Offset>(
67.                   begin:
68.                     const Offset(0.1, 0), // Muncul dari samping kanan
69.                     end: Offset.zero,
70.                   ).animate(animation),
71.                   child: Column(
72.                     children: [
73.                       Padding(
74.                         padding: const EdgeInsets.symmetric(
75.                           horizontal: 10, vertical: 5),
76.                         child: Card(
77.                           elevation: 3,
78.                           shape: RoundedRectangleBorder(
79.                             borderRadius: BorderRadius.circular(12),
80.                           ),
81.                           child: ListTile(
82.                             contentPadding: const EdgeInsets.symmetric(
83.                               horizontal: 16,
84.                               vertical: 8,
85.                             ),
86.                             leading: CircleAvatar(
87.                               backgroundColor: Colors.blueAccent.shade100,
88.                               child: Text(
89.                                 barang.namaBarang[0]
90.                                   .toUpperCase(), // Inisial Barang
91.                                   style: const TextStyle(
92.                                     color: Colors.white,
93.                                     fontWeight: FontWeight.bold,
94.                                   ),
95.                                   ),
96.                                   ),
97.                                   title: Text(
```



```
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');
```

### 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 $allowedFields = ['nama_barang', 'stok'];
12
13    public function getById($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 banrang 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	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	3	Innodb	
<input type="checkbox"/> barang_masuk	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	5	Innodb	
<input type="checkbox"/> stok	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	8	Innodb	
<input type="checkbox"/> users	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	2	Innodb	
<b>4 tables</b>	<b>Sum</b>			<b>18 Innodb</b>