

SISTEM MANAJEMEN JASA TENAGA KERJA BANGUNAN

BERBASIS MOBILE

(STUDI KASUS: SDM PEMALANG)

ABSTRAK

Tukang bangunan atau biasa disebut dengan kuli bangunan adalah sebuah profesi dalam bidang rancang bangunan yang memiliki keahlian masing-masing di dalam perancangan bangunan seperti: ruko, rumah, dan bangunan lainnya. Dan profesi ini merupakan salah satu bidang yang banyak dicari di Pemalang. Dengan meningkatnya populasi penduduk, sehingga menjadi banyak yang membutuhkan tempat tinggal. Meski saat ini banyak masyarakat yang berprofesi sebagai tukang bangunan. Namun masyarakat masih banyak yang mengalami kesulitan dalam mencari jasa yang ahli dalam bidang ini. Karena minimnya informasi yang diperoleh mengenai penyedia jasa tenaga kerja bangunan. Selain itu masyarakat juga tidak jarang yang merasa harga jasa tersebut terlalu mahal. Karena kurangnya informasi jasa dan kinerja dari tukang tersebut.

Dengan adanya aplikasi manajemen jasa tenaga kerja bangunan berbasis web ini dapat memberikan informasi mengenai jasa tukang dan memajemen kinerja tukang bangunan. Sehingga dalam aplikasi ini bagi customer bisa mengupload informasi yang bahwa beliau sedang membutuhkan tenaga kerja. Dan untuk pekerja juga dapat melihat informasi dari customer posting dan memilih jenis pekerjaan yang cocok untuk dia ambil sesuai dengan keahliannya.

Selain itu customer juga bisa memilih pekerja yang menurutnya sesuai dengan yang dibutuhkan dengan melihat rating dan pengalaman pekerja yang telah di publikasi. Aplikasi ini juga dapat mempermudah masyarakat Pemalang untuk negosiasi harga antara customer dan pekerja sehingga dapat lebih efisien dan mendapat kesepakatan yang jelas sesuai dengan ketentuan.

Kata Kunci: Aplikasi, Tukang, Customer, Pencari Jasa, Penyedia Jasa, Manajemen

N a m a : 1. Abdul Haris As'ari
Kewarganegaraan : Indonesia
Alamat : Rimpak RT 02/RW 06, Kec. Belik, Kab. Pemalang,
Jawa Tengah
Kode Pos : 52356

N a m a : 2. Agus Suhendar.S.T,.M.Eng
Kewarganegaraan : Indonesia
Alamat : Degan II, 082/041, Banjararum, Kalibawang, Kulon Progo,
Yogyakarta
Kode Pos : 55672

Daftar Nama Pengusul

N a m a : 1. Abdul Haris As'ari
Kewarganegaraan : Indonesia
Alamat : Rimpak RT 02 RW 06, Kec. Belik, Kab. Pemalang,
Jawa Tengah
Kode Pos : 52356
No Hp : 088225236570
Alamat email : abdulharisasari@gmail.com

N a m a : 2. Agus Suhendar,S.T.,M.Eng
Kewarganegaraan : Indonesia
Alamat : Degan II, 082/041, Banjararum, Kalibawang,
Kulon Progo, Yogyakarta
Kode Pos : 55672
No Hp : 081578888331
Alamat email : agus.suhendar@staff.uty.ac.id

PROVINSI JAWA TENGAH
KABUPATEN PEMALANG

NIK : 3327030107010242

Nama : ABDUL HARIS AS'ARI
Tempat/Tgl Lahir : PEMALANG, 01-07-2001
Jenis Kelamin : LAKI-LAKI Gol. Darah : -
Alamat : DK RIMPAK
RT/RW : 002/006
Kel/Desa : BELIK
Kecamatan : BELIK
Agama : ISLAM
Status Perkawinan : BELUM KAWIN
Pekerjaan : PELAJAR/MAHASISWA
Kewarganegaraan : WNI
Bertaku Hingga : SEUMUR HIDUP



PEMALANG
07-03-2019

PROVINSI DAERAH ISTIMEWA YOGYAKARTA
KABUPATEN KULON PROGO

NIK : 3401122007870001

Nama : AGUS SUHENDAR, S.T. M.Eng.
Tempat/Tgl Lahir : KOTABUMI, 20-07-1987
Jenis kelamin : LAKI-LAKI Gol. Darah : B
Alamat : DEGAN II
RT/RW : 082/041
Kel/Desa : BANJARARUM
Kecamatan : KALIBAWANG
Agama : ISLAM
Status Perkawinan : KAWIN
Pekerjaan : DOSEN
Kewarganegaraan : WNI
Berlaku Hingga : SEUMUR HIDUP



KULON PROGO
21-05-2018

A handwritten signature in black ink, consisting of a large, stylized letter 'A' followed by several horizontal strokes.

MANUAL SISTEM

SISTEM MANAJEMEN JASA TENAGA KERJA BANGUNAN

BERBASIS MOBILE

(Studi Kasus: SDM Pernalang)



OLEH

1. Agus Suhendar.S.T.,M.Eng (NIDN. 110919178)
2. Abdul Haris As'ari (NIM. 5200411275)

UNIVERSITAS TEKNOLOGI YOGYAKARTA

TAHUN 2023

DAFTAR ISI

JUDUL.....	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR.....	ii
1. TAMPILAN DAN KODE PROGRAM	1
1.1 Halaman Splash Screen.....	1
1.1.1 Tampilan Splash Screen.....	1
1.1.2 Source Code Halaman Splash Screen	1
1.2 Halaman Login.....	3
1.2.1. Tampilan Login.....	3
1.2.2. Source Code Halaman Login	3
1.3 Halaman Home	7
1.3.1. Tampilan Halaman Home	7
1.3.2. Source Code Halaman Home.....	7
1.4 Halaman Profile	12
1.4.1 Tampilan Halaman Profile	12
1.4.2 Source Code Halaman Profile.....	12
1.5 Halaman Messages.....	21
1.5.1 Tampilan Halaman Messages	21
1.5.2 Source Code Halaman Messages	21
1.7 Halaman Hitung Gaji	35
1.6.1 Tampilan Hitung Gaji	35
1.6.2 Source Halaman Hitung Gaji	35
2. PENGGUNAAN PROGRAM.....	42
2.1 Melakukan Login	42
2.2 Melakukan Pencarian Tukang.....	43

2.3	Melakukan Hitung Perkiraan Gaji	43
2.4	Melakukan Negosiasi Gaji.....	44
2.5	Melakukan Edit Profile	44

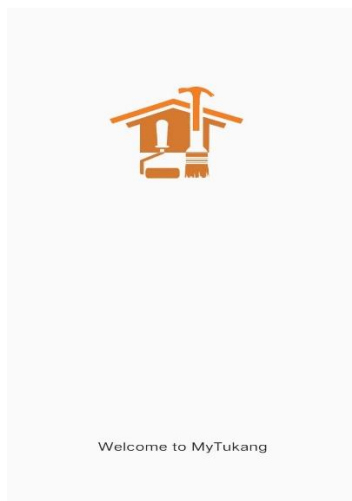
DAFTAR GAMBAR

Gambar 1 - Halamanan Splash Screen.....	1
Gambar 2 - Halaman Login	3
Gambar 3 - Halaman Home	7
Gambar 4 - Halaman Profile	12
Gambar 5 - Halaman Messages	21
Gambar 6 - Halaman Hitung Gaji.....	35
Gambar 7 - Alur Login	42
Gambar 8 - Alur pencarian tukang.....	43
Gambar 9 - Alur Hitung Perkiraan Gaji.....	43
Gambar 10 - Alur Negosiasi	44
Gambar 11 - Alur Edit profile.....	44

1. TAMPILAN DAN KODE PROGRAM

1.1 Halaman Splash Screen

1.1.1 Tampilan Splash Screen



Gambar 1- Halamanan Splash Screen

1.1.2 Source Code Halaman Splash Screen

a. slash_screen.dart

```
import 'dart:developer';

import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:t1/mytukang/screens/home_page.dart';

import '../main.dart';
import '../api/apis.dart';
import 'auth/login_screen.dart';

//splash screen
class SplashScreen extends StatefulWidget {
  const SplashScreen({super.key});

  @override
  State<SplashScreen> createState() => _SplashScreenState();
}

class _SplashScreenState extends State<SplashScreen> {
  @override
  void initState() {
    super.initState();
```

```

Future.delayed(const Duration(seconds: 2), () {
  //exit full-screen
  SystemChrome.setEnabledSystemUIMode(SystemUiMode.edgeToEdge);
  SystemChrome.setSystemUIOverlayStyle(const SystemUiOverlayStyle(
    systemNavigationBarColor: Colors.white,
    statusBarColor: Colors.white));

  if (APIs.auth.currentUser != null) {
    log('\nUser: ${APIs.auth.currentUser}');
    //navigate to home screen
    Navigator.pushReplacement(
      context, MaterialPageRoute(builder: (_) => const HomePage1()));
  } else {
    //navigate to login screen
    Navigator.pushReplacement(
      context, MaterialPageRoute(builder: (_) => const LoginScreen()));
  }
});
}

@override
Widget build(BuildContext context) {
  //initializing media query (for getting device screen size)
  mq = MediaQuery.of(context).size;

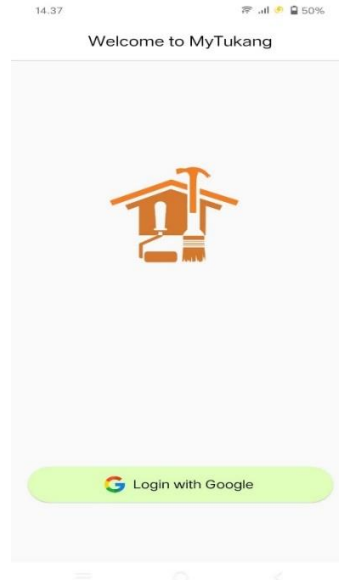
  return Scaffold(
    //body
    body: Stack(children: [
      //app logo
      Positioned(
        top: mq.height * .15,
        right: mq.width * .25,
        width: mq.width * .5,
        child: Image.asset('images/icon.png')),

      //google login button
      Positioned(
        bottom: mq.height * .15,
        width: mq.width,
        child: const Text('Welcome to MyTukang',
          textAlign: TextAlign.center,
          style: TextStyle(
            fontSize: 16, color: Colors.black87, letterSpacing: .5))),
    ]),
  );
}
}

```

1.2 Halaman Login

1.2.1. Tampilan Login



Gambar 2 - Halaman Login

1.2.2. Source Code Halaman Login

a. Login_screen.dart

```
import 'dart:developer';
import 'dart:io';

import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:google_sign_in/google_sign_in.dart';
import 'package:t1/mytukang/screens/home_page.dart';

import '../api/apis.dart';
import '../helper/dialogs.dart';
import '../main.dart';

//login screen -- implements google sign in or sign up feature for app
class LoginScreen extends StatefulWidget {
  const LoginScreen({super.key});

  @override
  State<LoginScreen> createState() => _LoginScreenState();
}

class _LoginScreenState extends State<LoginScreen> {
  bool _isAnimate = false;
```

```

@override
void initState() {
  super.initState();

  //for auto triggering animation
  Future.delayed(const Duration(milliseconds: 500), () {
    setState(() => _isAnimate = true);
  });
}

// handles google login button click
_handleGoogleBtnClick() {
  //for showing progress bar
  Dialogs.showProgressBar(context);

  _signInWithGoogle().then((user) async {
    //for hiding progress bar
    Navigator.pop(context);

    if (user != null) {
      log('\nUser: ${user.user}');
      log('\nUserAdditionalInfo: ${user.additionalUserInfo}');

      if ((await APIs.userExists())) {
        Navigator.pushReplacement(
          context, MaterialPageRoute(builder: (_) => const HomePage1()));
      } else {
        await APIs.createUser().then((value) {
          Navigator.pushReplacement(
            context, MaterialPageRoute(builder: (_) => const HomePage1()));
        });
      }
    }
  });
}

Future<UserCredential?> _signInWithGoogle() async {
  try {
    await InternetAddress.lookup('google.com');
    // Trigger the authentication flow
    final GoogleSignInAccount? googleUser = await GoogleSignIn().signIn();

    // Obtain the auth details from the request
    final GoogleSignInAuthentication? googleAuth =
      await googleUser?.authentication;

    // Create a new credential
    final credential = GoogleAuthProvider.credential(
      accessToken: googleAuth?.accessToken,
      idToken: googleAuth?.idToken,

```



```

);

// Once signed in, return the UserCredential
return await APIs.auth.signInWithCredential(credential);
} catch (e) {
  log('\n_signInWithGoogle: $e');
  Dialogs.showSnackBar(context, 'Something Went Wrong (Check Internet!>');
  return null;
}
}

@override
Widget build(BuildContext context) {
  //initializing media query (for getting device screen size)
  // mq = MediaQuery.of(context).size;

  return Scaffold(
    //app bar
    appBar: AppBar(
      automaticallyImplyLeading: false,
      title: const Text('Welcome to MyTukang'),
    ),

    //body
    body: Stack(children: [
      //app logo
      AnimatedPositioned(
        top: mq.height * .15,
        right: _isAnimate ? mq.width * .25 : -mq.width * .5,
        width: mq.width * .5,
        duration: const Duration(seconds: 1),
        child: Image.asset('images/icon.png')),

      //google login button
      Positioned(
        bottom: mq.height * .15,
        left: mq.width * .05,
        width: mq.width * .9,
        height: mq.height * .06,
        child: ElevatedButton.icon(
          style: ElevatedButton.styleFrom(
            backgroundColor: const Color.fromARGB(255, 223, 255, 187),
            shape: const StadiumBorder(),
            elevation: 1),
          onPressed: () {
            _handleGoogleBtnClick();
          },

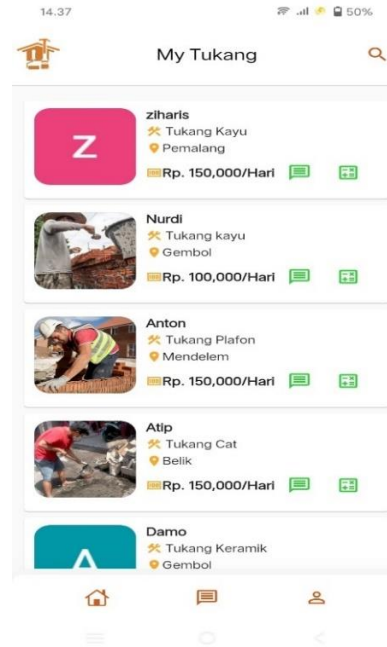
          //google icon
          icon: Image.asset('images/google.png', height: mq.height * .03),

```

```
//login with google label
label: RichText(
  text: const TextSpan(
    style: TextStyle(color: Colors.black, fontSize: 16),
    children: [
      TextSpan(text: 'Login with '),
      TextSpan(
        text: 'Google',
        style: TextStyle(fontWeight: FontWeight.w500)),
    ],
  )),
);
}
```

1.3 Halaman Home

1.3.1. Tampilan Halaman Home



Gambar 3 - Halaman Home

1.3.2. Source Code Halaman Home

a. Home_screen.dart

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:t1/mytukang/models/chat_user.dart';
import 'package:t1/mytukang/navigation_page.dart';
import 'package:t1/mytukang/screens/profile_screen.dart';
import 'package:t1/mytukang/widgets/tukang_card.dart';
import 'package:t1/mytukang/widgets/user_card.dart';

import '../main.dart';
import '../api/apis.dart';

class HomePage1 extends StatefulWidget {
  const HomePage1({Key? key}) : super(key: key);

  @override
  State<HomePage1> createState() => _HomePage1State();
}

class _HomePage1State extends State<HomePage1> {
  List<ChatUser> _list = [];
```

```

final List<ChatUser> _searchList = [];
// for storing search status
bool _isSearching = false;
@override
void initState() {
  super.initState();
  APIs.getSelfInfo();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      toolbarHeight: 80,
      leading: Image.asset(
        'images/icon.png',
        width: 24,
        height: 24,
      ),
      title: _isSearching
        ? TextField(
            decoration: const InputDecoration(
              border: InputBorder.none, hintText: 'Nama, Keahlian, ...'),
              autofocus: true,
              style: const TextStyle(fontSize: 17, letterSpacing: 0.5),
              //when search text changes then updated search list
              onChanged: (val) {
                //search logic
                _searchList.clear();

                for (var i in _list) {
                  if (i.name.toLowerCase().contains(val.toLowerCase()) ||
                      i.keahlian.toLowerCase().contains(val.toLowerCase())) {
                    _searchList.add(i);
                    setState() {
                      _searchList;
                    });
                  }
                }
              },
            ),
          : const Text('My Tukang'),
    actions: [
      //search user button
      IconButton(
        onPressed: () {
          setState() {
            _isSearching = !_isSearching;
          });
        },
      ],
    ),
  );
}

```

```

        icon: Icon(
          _isSearching
            ? CupertinoIcons.clear_circled_solid
            : Icons.search,
          color: Color.fromARGB(255, 209, 92, 29),
        )),
      ],
    ),
    body: Padding(
      padding: const EdgeInsets.all(8.0),
      child: StreamBuilder(
        // stream: APIs.getMyUsersId(),
        stream: APIs.firestore.collection('Tukang1').snapshots(),

        //get id of only known users
        builder: (context, snapshot) {
          switch (snapshot.connectionState) {
            //if data is loading
            case ConnectionState.waiting:
            case ConnectionState.none:
              return const Center(child: CircularProgressIndicator());

            //if some or all data is loaded then show it
            case ConnectionState.active:
            case ConnectionState.done:
              return Container(
                child: StreamBuilder(
                  // stream: APIs.getTukang(
                  //   snapshot.data?.docs.map((e) => e.id).toList() ?? []),

                  stream: APIs.firestore.collection('Tukang').snapshots(),
                  //get only those user, who's ids are provided
                  builder: (context, snapshot) {
                    switch (snapshot.connectionState) {
                      //if data is loading
                      case ConnectionState.waiting:
                      case ConnectionState.none:
                        // return const Center(
                        //   child: CircularProgressIndicator());

                      //if some or all data is loaded then show it
                      case ConnectionState.active:
                      case ConnectionState.done:
                        final data = snapshot.data?.docs;
                        _list = data
                          ?.map((e) => ChatUser.fromJson(e.data()))
                          .toList() ??
                        [];

                        if (_list.isNotEmpty) {

```



```

return

    ListView.builder(
      itemCount: _isSearching
        ? _searchList.length
        : _list.length,
      padding:
        EdgeInsets.only(top: mq.height * .01),
      physics: const BouncingScrollPhysics(),
      itemBuilder: (context, index) {
        return ChatTukangCard(
          user: _isSearching
            ? _searchList[index]
            : _list[index]);
      });
    } else {
      return const Center(
        child: Text('No Connections Found!',
          style: TextStyle(fontSize: 20)),
      );
    }
  },
),
);
}
},
),
),
bottomNavigationBar: myhi(),
);
}
}

```

b. navigation.dart

```

import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:t1/mytukang/api/apis.dart';
import 'package:t1/mytukang/screens/home_page.dart';
import 'package:t1/mytukang/screens/kontrak_page.dart';
import 'package:t1/mytukang/screens/message_screen.dart';
import 'package:t1/mytukang/screens/profile_screen.dart';

class myhi extends StatefulWidget {
  @override

```

```

_myhiState createState() => _myhiState();
}

class _myhiState extends State<myhi> {
  @override
  Widget build(BuildContext context) {
    return BottomAppBar(
      shape: CircularNotchedRectangle(),
      notchMargin: 6.0,
      color: Colors.transparent,
      elevation: 80,
      clipBehavior: Clip.none, // Gunakan nilai Clip.none
      child: Container(
        height: 60.0,
        decoration: BoxDecoration(
          borderRadius: BorderRadius.only(
            topLeft: Radius.circular(10.0),
            topRight: Radius.circular(10.0),
          ),
          color: Colors.white,
        ),
        child: Row(
          mainAxisAlignment: MainAxisAlignment.spaceEvenly,
          children: [
            IconButton(
              onPressed: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(builder: (_) => HomePage1()),
                );
              },
              icon: Icon(
                CupertinoIcons.home,
                color: Color.fromARGB(255, 209, 92, 29),
              ),
            ),
            IconButton(
              onPressed: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(builder: (_) => MessagePage()),
                );
              },
              icon: Icon(Icons.message_outlined,
                color: Color.fromARGB(255, 209, 92, 29)),
            ),
            IconButton(
              onPressed: () {
                Navigator.push(

```

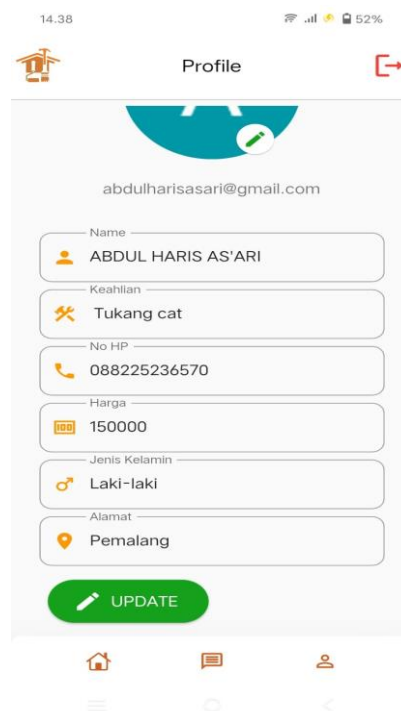
```

        context,
        MaterialPageRoute(
          builder: (_) => ProfilePage(user: APIs.me)),
      );
    },
    icon: Icon(Icons.person_outline,
      color: Color.fromARGB(255, 209, 92, 29))),
  ],
),
);
}
}

```

1.4 Halaman Profile

1.4.1 Tampilan Halaman Profile



Gambar 4 - Halaman Profile

1.4.2 Source Code Halaman Profile

a. Profile_screen.dart

```

import 'dart:developer';
import 'dart:io';
import 'package:cached_network_image/cached_network_image.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';

```

```

import 'package:google_sign_in/google_sign_in.dart';
import 'package:image_picker/image_picker.dart';
import 'package:t1/mytukang/navigation_page.dart';
import 'package:intl/intl.dart';
import '../api/apis.dart';
import '../helper/dialogs.dart';
import '.././main.dart';
import '../models/chat_user.dart';
import 'auth/login_screen.dart';

//profile screen -- to show signed in user info
class ProfilePage extends StatefulWidget {
  final ChatUser user;

  const ProfilePage({super.key, required this.user});

  @override
  State<ProfilePage> createState() => _ProfilePageState();
}

class _ProfilePageState extends State<ProfilePage> {
  final _formKey = GlobalKey<FormState>();
  String? _image;
  String? selectedGender;

  @override
  Widget build(BuildContext context) {
    return GestureDetector(
      // for hiding keyboard
      onTap: () => FocusScope.of(context).unfocus(),
      child: Scaffold(
        //app bar
        appBar: AppBar(
          toolbarHeight: 80,
          leading: Image.asset(
            'images/icon.png',
            width: 24,
            height: 24,
          ),
          title: Text('Profile'),
          actions: [
            IconButton(
              onPressed: () async {
                //for showing progress dialog
                Dialogs.showProgressBar(context);

                await APIs.updateActiveStatus(false);

                //sign out from app
                await APIs.auth.signOut().then((value) async {

```

```

await GoogleSignIn().signOut().then((value) {
  //for hiding progress dialog
  Navigator.pop(context);

  //for moving to home screen
  Navigator.pop(context);

  APIs.auth = FirebaseAuth.instance;

  //replacing home screen with login screen
  Navigator.pushReplacement(context,
    MaterialPageRoute(builder: (_) => const LoginScreen()));
  });
});
},
icon: const Icon(
  Icons.logout_rounded,
  color: Colors.red,
  size: 35,
),
),
],
),

//body
body: StreamBuilder<Object>(
  stream: null,
  builder: (context, snapshot) {
    return Padding(
      padding: const EdgeInsets.all(8.0),
      child: Container(
        child: Form(
          key: _formKey,
          child: Padding(
            padding: EdgeInsets.symmetric(horizontal: mq.width * .05),
            child: SingleChildScrollView(
              child: Column(
                children: [
                  // for adding some space
                  SizedBox(width: mq.width, height: mq.height * .03),

                  //user profile picture
                  Stack(
                    children: [
                      //profile picture
                      _image != null
                      ?

                      //local image
                      ClipRRect(

```



```

        borderRadius: BorderRadius.circular(
            mq.height * .1),
        child: Image.file(File(_image!),
            width: mq.height * .2,
            height: mq.height * .2,
            fit: BoxFit.cover))
    :

    //image from server
    ClipRRect(
        borderRadius: BorderRadius.circular(
            mq.height * .1),
        child: CachedNetworkImage(
            width: mq.height * .2,
            height: mq.height * .2,
            fit: BoxFit.cover,
            imageUrl: widget.user.image,
            errorWidget: (context, url, error) =>
                const CircleAvatar(
                    child: Icon(
                        CupertinoIcons.person)),
                ),
    ),

    //edit image button
    Positioned(
        bottom: 0,
        right: 0,
        child: MaterialButton(
            elevation: 1,
            onPressed: () {
                _showBottomSheet();
            },
            shape: const CircleBorder(),
            color: Colors.white,
            child: const Icon(Icons.edit,
                color:
                    Color.fromARGB(255, 22, 161, 29)),
        ),
    ),
],
),

// for adding some space
SizedBox(height: mq.height * .03),

// user email label
Text(widget.user.email,
    style: const TextStyle(
        color: Colors.black54, fontSize: 16)),

```

```

// for adding some space
SizedBox(height: mq.height * .05),

// name input field
TextFormField(
  initialValue: widget.user.name,
  onSave: (val) => APIs.me.name = val ?? "",
  validator: (val) => val != null && val.isNotEmpty
    ? null
    : 'Required Field',
  decoration: InputDecoration(
    prefixIcon: const Icon(Icons.person,
      color: Colors.orange),
    border: OutlineInputBorder(
      borderRadius: BorderRadius.circular(12)),
    hintText: 'Nama anda',
    label: const Text('Name')),
),

// for adding some space
SizedBox(height: mq.height * .01),

// keahlian input field
TextFormField(
  initialValue: widget.user.keahlian,
  onSave: (val) => APIs.me.keahlian = val ?? "",
  validator: (val) => val != null && val.isNotEmpty
    ? null
    : 'Required Field',
  decoration: InputDecoration(
    prefixIcon: const Icon(Icons.construction,
      color: Colors.orange),
    border: OutlineInputBorder(
      borderRadius: BorderRadius.circular(12)),
    hintText: 'Masukan keahlian anda',
    label: const Text('Keahlian')),
),

// for adding some space
SizedBox(height: mq.height * .01),

// alamat input field

TextFormField(
  initialValue: widget.user.no_hp,
  onSave: (val) => APIs.me.no_hp = val ?? "",
  validator: (val) => val != null && val.isNotEmpty
    ? null
    : 'Required Field',

```

```

decoration: InputDecoration(
  prefixIcon: const Icon(Icons.phone,
    color: Colors.orange),
  border: OutlineInputBorder(
    borderRadius: BorderRadius.circular(12)),
  hintText: 'Masukan no hp anda',
  label: const Text('No HP')),
),
SizedBox(height: mq.height * .01),

TextFormField(
  initialValue: widget.user.harga,
  //
  "${NumberFormat.decimalPattern().format(int.parse(widget.user.harga))}",
  // widget.user.harga,
  onSave: (val) => APIs.me.harga = val ?? "",
  validator: (val) => val != null && val.isNotEmpty
    ? null
    : 'Required Field',
  decoration: InputDecoration(
    prefixIcon: const Icon(Icons.money,
      color: Colors.orange),
    border: OutlineInputBorder(
      borderRadius: BorderRadius.circular(12)),
    hintText: 'masukan harga upah',
    label: const Text('Harga')),
  ),
SizedBox(height: mq.height * .01),
TextFormField(
  initialValue: widget.user.jk,
  //
  "${NumberFormat.decimalPattern().format(int.parse(widget.user.harga))}",
  // widget.user.harga,
  onSave: (val) => APIs.me.jk = val ?? "",
  validator: (val) => val != null && val.isNotEmpty
    ? null
    : 'Required Field',
  decoration: InputDecoration(
    prefixIcon: const Icon(Icons.male,
      color: Colors.orange),
    border: OutlineInputBorder(
      borderRadius: BorderRadius.circular(12)),
    hintText: 'masukan harga upah',
    label: const Text('Jenis Kelamin')),
  ),
SizedBox(height: mq.height * .01),
TextFormField(
  initialValue: widget.user.alamat,
  onSave: (val) => APIs.me.alamat = val ?? "",

```



```

        ),
    ),
    ),
    );
  }},
  bottomNavigationBar: myhi(),
),
);
}

// bottom sheet for picking a profile picture for user
void _showBottomSheet() {
  showModalBottomSheet(
    context: context,
    shape: const RoundedRectangleBorder(
      borderRadius: BorderRadius.only(
        topLeft: Radius.circular(20), topRight: Radius.circular(20))),
    builder: (_) {
      return ListView(
        shrinkWrap: true,
        padding:
          EdgeInsets.only(top: mq.height * .03, bottom: mq.height * .05),
        children: [
          //pick profile picture label
          const Text('Pick Profile Picture',
            textAlign: TextAlign.center,
            style: TextStyle(fontSize: 20, fontWeight: FontWeight.w500)),

          //for adding some space
          SizedBox(height: mq.height * .02),

          //buttons
          Row(
            mainAxisAlignment: MainAxisAlignment.spaceEvenly,
            children: [
              //pick from gallery button
              ElevatedButton(
                style: ElevatedButton.styleFrom(
                  backgroundColor: Colors.white,
                  shape: const CircleBorder(),
                  fixedSize: Size(mq.width * .3, mq.height * .15)),
                onPressed: () async {
                  final ImagePicker picker = ImagePicker();

                  // Pick an image
                  final XFile? image = await picker.pickImage(
                    source: ImageSource.gallery, imageQuality: 80);
                  if (image != null) {
                    log('Image Path: ${image.path}');
                    setState() {

```

```

        _image = image.path;
    });

    APIs.updateProfilePicture(File(_image!));
    // for hiding bottom sheet
    Navigator.pop(context);
  }
},
child: Image.asset('images/add_image.png')),

//take picture from camera button
ElevatedButton(
  style: ElevatedButton.styleFrom(
    backgroundColor: Colors.white,
    shape: const CircleBorder(),
    fixedSize: Size(mq.width * .3, mq.height * .15)),
  onPressed: () async {
    final ImagePicker picker = ImagePicker();

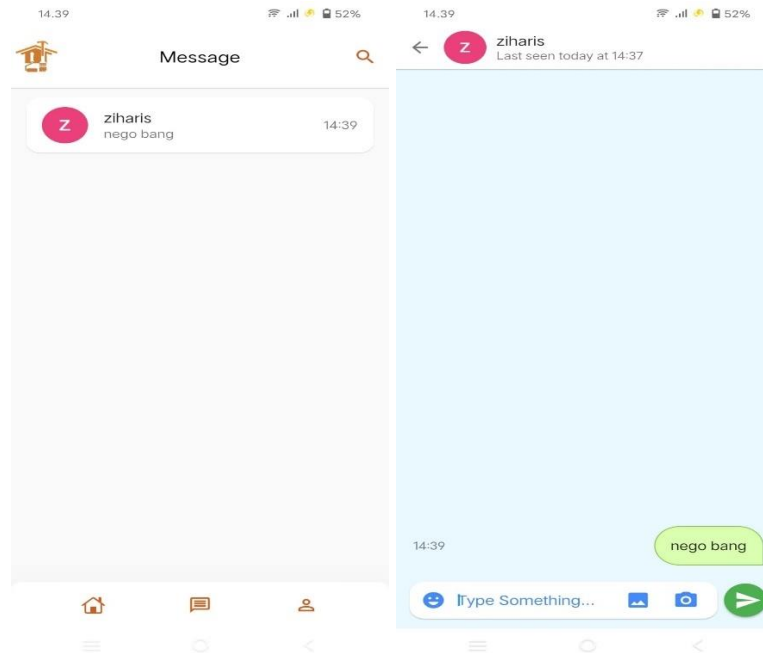
    // Pick an image
    final XFile? image = await picker.pickImage(
      source: ImageSource.camera, imageQuality: 80);
    if (image != null) {
      log('Image Path: ${image.path}');
      setState(() {
        _image = image.path;
      });

      APIs.updateProfilePicture(File(_image!));
      // for hiding bottom sheet
      Navigator.pop(context);
    }
  },
  child: Image.asset('images/camera.png')),
),
),
);
});
}
}

```

1.5 Halaman Messages

1.5.1 Tampilan Halaman Messages



Gambar 5 - Halaman Messages

1.5.2 Source Code Halaman Messages

a. messages_screen.dart

```
import 'dart:developer';

import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:t1/mytukang/navigation_page.dart';
import 'package:t1/mytukang/screens/chat_screen.dart';
import 'package:t1/mytukang/screens/home_page.dart';
import 'package:t1/mytukang/screens/message_screen.dart';
import 'package:t1/mytukang/widgets/user_card.dart';

import '../api/apis.dart';
import '../helper/dialogs.dart';
import '../main.dart';
import '../models/chat_user.dart';
import '../widgets/tukang_card.dart';
import 'profile_screen.dart';

//home screen -- where all available contacts are shown
class MessagePage extends StatefulWidget {
  const MessagePage({super.key});
```

```

@override
State<MessagePage> createState() => _MessagePageState();
}

class _MessagePageState extends State<MessagePage> {
  // for storing all users
  List<ChatUser> _list = [];

  // for storing searched items
  final List<ChatUser> _searchList = [];
  // for storing search status
  bool _isSearching = false;

  @override
  void initState() {
    super.initState();
    APIs.getSelfInfo();

    //for updating user active status according to lifecycle events
    //resume -- active or online
    //pause -- inactive or offline
    SystemChannels.lifecycle.setMessageHandler((message) {
      log('Message: $message');

      if (APIs.auth.currentUser != null) {
        if (message.toString().contains('resume')) {
          APIs.updateActiveStatus(true);
        }
        if (message.toString().contains('pause')) {
          APIs.updateActiveStatus(false);
        }
      }
    });

    return Future.value(message);
  }

  @override
  Widget build(BuildContext context) {
    return GestureDetector(
      //for hiding keyboard when a tap is detected on screen
      onTap: () => FocusScope.of(context).unfocus(),
      child: WillPopScope(
        //if search is on & back button is pressed then close search
        //or else simple close current screen on back button click
        onWillPop: () {
          if (_isSearching) {
            setState(() {
              _isSearching = !_isSearching;
            });
          }
        }
      )
    );
  }
}

```



```

        return Future.value(false);
    } else {
        return Future.value(true);
    }
},
child: Scaffold(
  //app bar
  appBar: AppBar(
    toolbarHeight: 80,
    leading: Image.asset(
      'images/icon.png',
      width: 24,
      height: 24,
    ),
    title: _isSearching
      ? TextField(
        decoration: const InputDecoration(
          border: InputBorder.none,
          hintText: 'Name, keahlian, ...'),
          autofocus: true,
          style: const TextStyle(fontSize: 17, letterSpacing: 0.5),
          //when search text changes then updated search list
          onChanged: (val) {
            //search logic
            _searchList.clear();

            for (var i in _list) {
              if (i.name.toLowerCase().contains(val.toLowerCase()) ||
                i.keahlian
                  .toLowerCase()
                    .contains(val.toLowerCase())) {
                _searchList.add(i);
                setState(() {
                  _searchList;
                });
              }
            }
          },
        )
      : const Text('Message'),
    actions: [
      //search user button
      IconButton(
        onPressed: () {
          setState(() {
            _isSearching = !_isSearching;
          });
        },
        icon: Icon(
          _isSearching

```

```

        ? CupertinoIcons.clear_circled_solid
        : Icons.search,
        color: Color.fromARGB(255, 209, 92, 29),
    )),
  ],
),

//floating button to add new user
// floatingActionButton: Padding(
//   padding: const EdgeInsets.only(bottom: 10),
//   child: FloatingActionButton(
//     onPressed: () {
//       _addChatUserDialog();
//     },
//     child: const Icon(Icons.add_comment_rounded)),
// ),

//body
body: StreamBuilder(
  stream: APIs.getMyUsersId(),
  // stream: APIs.firestore.collection('users').snapshots(),

  //get id of only known users
  builder: (context, snapshot) {
    switch (snapshot.connectionState) {
      //if data is loading
      case ConnectionState.waiting:
      case ConnectionState.none:
        return const Center(child: CircularProgressIndicator());

      //if some or all data is loaded then show it
      case ConnectionState.active:
      case ConnectionState.done:
        return StreamBuilder(
          stream: APIs.getAllUsers(
            snapshot.data?.docs.map((e) => e.id).toList() ?? []),

          //get only those user, who's ids are provided
          builder: (context, snapshot) {
            switch (snapshot.connectionState) {
              //if data is loading
              case ConnectionState.waiting:
              case ConnectionState.none:
                // return const Center(
                //   child: CircularProgressIndicator());

              //if some or all data is loaded then show it
              case ConnectionState.active:
              case ConnectionState.done:
                final data = snapshot.data?.docs;

```

```

        _list = data
            ?.map((e) => ChatUser.fromJson(e.data()))
            .toList() ??
            [];

        if (_list.isNotEmpty) {
            return ListView.builder(
                itemCount: _isSearching
                    ? _searchList.length
                    : _list.length,
                padding: EdgeInsets.only(top: mq.height * .01),
                physics: const BouncingScrollPhysics(),
                itemBuilder: (context, index) {
                    return UserCard(
                        user: _isSearching
                            ? _searchList[index]
                            : _list[index]);
                });
        } else {
            return const Center(
                child: Text('No Connections Found!',
                    style: TextStyle(fontSize: 20)),
            );
        }
    },
),
),
);
}
}
),
bottomNavigationBar: myhi(),
),
);
}

```

```

// for adding new chat user
void _addChatUserDialog() {
    String email = "";

    showDialog(
        context: context,
        builder: (_) => AlertDialog(
            contentPadding: const EdgeInsets.only(
                left: 24, right: 24, top: 20, bottom: 10),

            shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(20)),

            //title

```

```

title: Row(
  children: const [
    Icon(
      Icons.person_add,
      color: Colors.blue,
      size: 28,
    ),
    Text(' Add User')
  ],
),

//content
content: TextFormField(
  maxLines: null,
  onChanged: (value) => email = value,
  decoration: InputDecoration(
    hintText: 'Email Id',
    prefixIcon: const Icon(Icons.email, color: Colors.blue),
    border: OutlineInputBorder(
      borderRadius: BorderRadius.circular(15))),
),

//actions
actions: [
  //cancel button
  MaterialButton(
    onPressed: () {
      //hide alert dialog
      Navigator.pop(context);
    },
    child: const Text('Cancel',
      style: TextStyle(color: Colors.blue, fontSize: 16))),

  //add button
  MaterialButton(
    onPressed: () async {
      //hide alert dialog
      Navigator.pop(context);
      if (email.isNotEmpty) {
        await APIs.addChatUser(email).then((value) {
          if (!value) {
            Dialogs.showSnackBar(
              context, 'User does not Exists!');
          }
        });
      }
    },
    child: const Text(
      'Add',
      style: TextStyle(color: Colors.blue, fontSize: 16),

```

```

        ))
      ],
    ));
  }
}

```

b. in_messages.dart

```

import 'dart:developer';

import 'package:cached_network_image/cached_network_image.dart';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:gallery_saver/gallery_saver.dart';

import '../api/apis.dart';
import '../helper/dialogs.dart';
import '../helper/my_date_util.dart';
import '../main.dart';
import '../models/message.dart';

// for showing single message details
class MessageCard extends StatefulWidget {
  const MessageCard({super.key, required this.message});

  final Message message;

  @override
  State<MessageCard> createState() => _MessageCardState();
}

class _MessageCardState extends State<MessageCard> {
  @override
  Widget build(BuildContext context) {
    bool isMe = APIs.user.uid == widget.message.fromId;
    return InkWell(
      onLongPress: () {
        _showBottomSheet(isMe);
      },
      child: isMe ? _greenMessage() : _blueMessage();
    );
  }

  // sender or another user message
  Widget _blueMessage() {
    //update last read message if sender and receiver are different
    if (widget.message.read.isEmpty) {
      APIs.updateMessageReadStatus(widget.message);
    }
  }
}

```

```

return Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    //message content
    Flexible(
      child: Container(
        padding: EdgeInsets.all(widget.message.type == Type.image
          ? mq.width * .03
          : mq.width * .04),
        margin: EdgeInsets.symmetric(
          horizontal: mq.width * .04, vertical: mq.height * .01),
        decoration: BoxDecoration(
          color: const Color.fromARGB(255, 221, 245, 255),
          border: Border.all(color: Colors.lightBlue),
          //making borders curved
          borderRadius: const BorderRadius.only(
            topLeft: Radius.circular(30),
            topRight: Radius.circular(30),
            bottomRight: Radius.circular(30))),
        child: widget.message.type == Type.text
          ?
          //show text
          Text(
            widget.message.msg,
            style: const TextStyle(fontSize: 15, color: Colors.black87),
          )
          :
          //show image
          ClipRRect(
            borderRadius: BorderRadius.circular(15),
            child: CachedNetworkImage(
              imageUrl: widget.message.msg,
              placeholder: (context, url) => const Padding(
                padding: EdgeInsets.all(8.0),
                child: CircularProgressIndicator(strokeWidth: 2),
              ),
              errorWidget: (context, url, error) =>
                const Icon(Icons.image, size: 70),
            ),
          ),
      ),
    ),
    //message time
    Padding(
      padding: EdgeInsets.only(right: mq.width * .04),
      child: Text(
        MyDateUtil.getFormattedTime(
          context: context, time: widget.message.sent),

```

```

        style: const TextStyle(fontSize: 13, color: Colors.black54),
      ),
    ),
  ],
);
}

// our or user message
Widget _greenMessage() {
  return Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
      //message time
      Row(
        children: [
          //for adding some space
          SizedBox(width: mq.width * .04),

          //double tick blue icon for message read
          if (widget.message.read.isNotEmpty)
            const Icon(Icons.done_all_rounded, color: Colors.blue, size:
20),

          //for adding some space
          const SizedBox(width: 2),

          //sent time
          Text(
            MyDateUtil.getFormattedTime(
              context: context, time: widget.message.sent),
            style: const TextStyle(fontSize: 13, color: Colors.black54),
          ),
        ],
      ),

      //message content
      Flexible(
        child: Container(
          padding: EdgeInsets.all(widget.message.type == Type.image
            ? mq.width * .03
            : mq.width * .04),
          margin: EdgeInsets.symmetric(
            horizontal: mq.width * .04, vertical: mq.height * .01),
          decoration: BoxDecoration(
            color: const Color.fromARGB(255, 218, 255, 176),
            border: Border.all(color: Colors.lightGreen),
            //making borders curved
            borderRadius: const BorderRadius.only(
              topLeft: Radius.circular(30),
              topRight: Radius.circular(30),

```

```

        bottomLeft: Radius.circular(30))),
child: widget.message.type == Type.text
?
//show text
Text(
  widget.message.msg,
  style: const TextStyle(fontSize: 15, color: Colors.black87),
)
:
//show image
ClipRRect(
  borderRadius: BorderRadius.circular(15),
  child: CachedNetworkImage(
    imageUrl: widget.message.msg,
    placeholder: (context, url) => const Padding(
      padding: EdgeInsets.all(8.0),
      child: CircularProgressIndicator(strokeWidth: 2),
    ),
    errorWidget: (context, url, error) =>
      const Icon(Icons.image, size: 70),
  ),
),
),
),
],
);
}

// bottom sheet for modifying message details
void _showBottomSheet(bool isMe) {
  showModalBottomSheet(
    context: context,
    shape: const RoundedRectangleBorder(
      borderRadius: BorderRadius.only(
        topLeft: Radius.circular(20), topRight: Radius.circular(20))),
    builder: (_) {
      return ListView(
        shrinkWrap: true,
        children: [
          //black divider
          Container(
            height: 4,
            margin: EdgeInsets.symmetric(
              vertical: mq.height * .015, horizontal: mq.width * .4),
            decoration: BoxDecoration(
              color: Colors.grey, borderRadius: BorderRadius.circular(8)),
          ),

          widget.message.type == Type.text
          ?

```



```

//copy option
_OptionItem(
  icon: const Icon(Icons.copy_all_rounded,
    color: Colors.blue, size: 26),
  name: 'Copy Text',
  onTap: () async {
    await Clipboard.setData(
      ClipboardData(text: widget.message.msg))
      .then((value) {
        //for hiding bottom sheet
        Navigator.pop(context);

        Dialogs.showSnackBar(context, 'Text Copied!');
      });
  })
:
//save option
_OptionItem(
  icon: const Icon(Icons.download_rounded,
    color: Colors.blue, size: 26),
  name: 'Save Image',
  onTap: () async {
    try {
      log('Image Url: ${widget.message.msg}');
      await GallerySaver.saveImage(widget.message.msg,
        albumName: 'We Chat')
        .then((success) {
          //for hiding bottom sheet
          Navigator.pop(context);
          if (success != null && success) {
            Dialogs.showSnackBar(
              context, 'Image Successfully Saved!');
          }
        });
    } catch (e) {
      log('ErrorWhileSavingImg: $e');
    }
  }
),

//separator or divider
if (isMe)
  Divider(
    color: Colors.black54,
    endIndent: mq.width * .04,
    indent: mq.width * .04,
  ),

//edit option
if (widget.message.type == Type.text && isMe)
  _OptionItem(

```

```

        icon: const Icon(Icons.edit, color: Colors.blue, size: 26),
        name: 'Edit Message',
        onTap: () {
          //for hiding bottom sheet
          Navigator.pop(context);

          _showMessageUpdateDialog();
        }
      ),

      //delete option
      if (isMe)
        _OptionItem(
          icon: const Icon(Icons.delete_forever,
            color: Colors.red, size: 26),
          name: 'Delete Message',
          onTap: () async {
            await APIs.deleteMessage(widget.message).then((value) {
              //for hiding bottom sheet
              Navigator.pop(context);
            });
          }
        ),

      //separator or divider
      Divider(
        color: Colors.black54,
        endIndent: mq.width * .04,
        indent: mq.width * .04,
      ),

      //sent time
      _OptionItem(
        icon: const Icon(Icons.remove_red_eye, color: Colors.blue),
        name:
          'Sent At: ${MyDateUtil.getMessageTime(context: context,
            time: widget.message.sent)}',
        onTap: () {}),

      //read time
      _OptionItem(
        icon: const Icon(Icons.remove_red_eye, color: Colors.green),
        name: widget.message.read.isEmpty
          ? 'Read At: Not seen yet'
          : 'Read At: ${MyDateUtil.getMessageTime(context:
            context, time: widget.message.read)}',
        onTap: () {}),
    ],
  );
});
}

```

```

//dialog for updating message content
void _showMessageUpdateDialog() {
  String updatedMsg = widget.message.msg;

  showDialog(
    context: context,
    builder: (_) => AlertDialog(
      contentPadding: const EdgeInsets.only(
        left: 24, right: 24, top: 20, bottom: 10),

      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(20)),

      //title
      title: Row(
        children: const [
          Icon(
            Icons.message,
            color: Colors.blue,
            size: 28,
          ),
          Text(' Update Message')
        ],
      ),

      //content
      content: TextFormField(
        initialValue: updatedMsg,
        maxLines: null,
        onChanged: (value) => updatedMsg = value,
        decoration: InputDecoration(
          border: OutlineInputBorder(
            borderRadius: BorderRadius.circular(15))),
      ),

      //actions
      actions: [
        //cancel button
        MaterialButton(
          onPressed: () {
            //hide alert dialog
            Navigator.pop(context);
          },
          child: const Text(
            'Cancel',
            style: TextStyle(color: Colors.blue, fontSize: 16),
          )),

        //update button
        MaterialButton(

```

```

        onPressed: () {
          //hide alert dialog
          Navigator.pop(context);
          APIs.updateMessage(widget.message, updatedMsg);
        },
        child: const Text(
          'Update',
          style: TextStyle(color: Colors.blue, fontSize: 16),
        ))
      ],
    ));
  }
}

//custom options card (for copy, edit, delete, etc.)
class _OptionItem extends StatelessWidget {
  final Icon icon;
  final String name;
  final VoidCallback onTap;

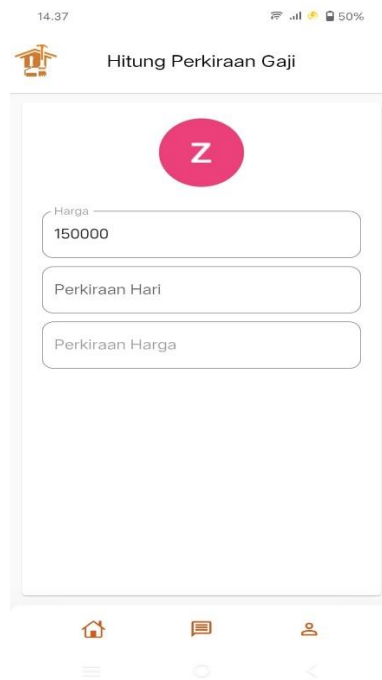
  const _OptionItem(
    {required this.icon, required this.name, required this.onTap});

  @override
  Widget build(BuildContext context) {
    return InkWell(
      onTap: () => onTap(),
      child: Padding(
        padding: EdgeInsets.only(
          left: mq.width * .05,
          top: mq.height * .015,
          bottom: mq.height * .015),
        child: Row(children: [
          icon,
          Flexible(
            child: Text(' $name',
              style: const TextStyle(
                fontSize: 15,
                color: Colors.black54,
                letterSpacing: 0.5)))
        ]),
      ));
  }
}

```

1.6 Halaman Hitung Gaji

1.6.1 Tampilan Hitung Gaji



Gambar 6 - Halaman Hitung Gaji

1.6.2 Source Halaman Hitung Gaji

a. Hitung_screen.dart

```
import 'dart:developer';

import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:t1/mytukang/navigation_page.dart';
import 'package:t1/mytukang/screens/chat_screen.dart';
import 'package:t1/mytukang/screens/home_page.dart';
import 'package:t1/mytukang/screens/message_screen.dart';
import 'package:t1/mytukang/widgets/user_card.dart';

import '../api/apis.dart';
import '../helper/dialogs.dart';
import '../main.dart';
import '../models/chat_user.dart';
import '../widgets/tukang_card.dart';
import 'profile_screen.dart';

//home screen -- where all available contacts are shown
class KontrakPage extends StatefulWidget {
  const KontrakPage({super.key});
```

```

    @override
    State<KontrakPage> createState() => _KontrakPageState();
  }

class _KontrakPageState extends State<KontrakPage> {
  // for storing all users
  List<ChatUser> _list = [];

  // for storing searched items
  final List<ChatUser> _searchList = [];
  // for storing search status
  bool _isSearching = false;

  @override
  void initState() {
    super.initState();
    APIs.getSelfInfo();

    //for updating user active status according to lifecycle events
    //resume -- active or online
    //pause -- inactive or offline
    SystemChannels.lifecycle.setMessageHandler((message) {
      log('Message: $message');

      if (APIs.auth.currentUser != null) {
        if (message.toString().contains('resume')) {
          APIs.updateActiveStatus(true);
        }
        if (message.toString().contains('pause')) {
          APIs.updateActiveStatus(false);
        }
      }
    });

    return Future.value(message);
  });
}

@override
Widget build(BuildContext context) {
  return GestureDetector(
    //for hiding keyboard when a tap is detected on screen
    onTap: () => FocusScope.of(context).unfocus(),
    child: WillPopScope(
      //if search is on & back button is pressed then close search
      //or else simple close current screen on back button click
      onWillPop: () {
        if (_isSearching) {
          setState(() {
            _isSearching = !_isSearching;
          });
        }
      },
    ),
  );
}

```

```

    return Future.value(false);
  } else {
    return Future.value(true);
  }
},
child: Scaffold(
  //app bar
  appBar: AppBar(
    toolbarHeight: 80,
    leading: Image.asset(
      'images/icon.png',
      width: 24,
      height: 24,
    ),
    title: _isSearching
      ? TextField(
          decoration: const InputDecoration(
            border: InputBorder.none,
            hintText: 'Name, Keahlian, ...'),
            autofocus: true,
            style: const TextStyle(fontSize: 17, letterSpacing: 0.5),
            //when search text changes then updated search list
            onChanged: (val) {
              //search logic
              _searchList.clear();

              for (var i in _list) {
                if (i.name.toLowerCase().contains(val.toLowerCase()) ||
                    i.email.toLowerCase().contains(val.toLowerCase())) {
                  _searchList.add(i);
                  setState(() {
                    _searchList;
                  });
                }
              }
            },
          )
      : const Text('Message'),
    actions: [
      //search user button
      IconButton(
        onPressed: () {
          setState(() {
            _isSearching = !_isSearching;
          });
        },
        icon: Icon(
          _isSearching
            ? CupertinoIcons.clear_circled_solid
            : Icons.search,

```

```

        color: Color.fromARGB(255, 209, 92, 29),
    )),
    ],
),

//body
body: StreamBuilder(
  stream:
    // APIs.getMyUsersId(),
    APIs.firestore
      .collection('Tukang')
      .doc()
      .collection('my_users')
      .snapshots(),

  //get id of only known users
  builder: (context, snapshot) {
    switch (snapshot.connectionState) {
      //if data is loading
      case ConnectionState.waiting:
      case ConnectionState.none:
        return const Center(child: CircularProgressIndicator());

      //if some or all data is loaded then show it
      case ConnectionState.active:
      case ConnectionState.done:
        return StreamBuilder(
          stream: APIs.getAllUsers(
            snapshot.data?.docs.map((e) => e.id).toList() ?? [],

          //get only those user, who's ids are provided
          builder: (context, snapshot) {
            switch (snapshot.connectionState) {
              //if data is loading
              case ConnectionState.waiting:
              case ConnectionState.none:
                // return const Center(
                //   child: CircularProgressIndicator());

              //if some or all data is loaded then show it
              case ConnectionState.active:
              case ConnectionState.done:
                final data = snapshot.data?.docs;
                _list = data
                  ?.map((e) => ChatUser.fromJson(e.data()))
                  .toList() ??
                [];

                if (_list.isNotEmpty) {
                  return ListView.builder(

```



```

        itemCount: _isSearching
            ? _searchList.length
            : _list.length,
        padding: EdgeInsets.only(top: mq.height * .01),
        physics: const BouncingScrollPhysics(),
        itemBuilder: (context, index) {
            return UserCard(
                user: _isSearching
                    ? _searchList[index]
                    : _list[index]);
        });
    } else {
        return const Center(
            child: Text('No Connections Found!',
                style: TextStyle(fontSize: 20)),
        );
    }
  },
),
bottomNavigationBar: myhi(),
),
);
}

```

```

// for adding new chat user
void _addChatUserDialog() {
  String email = "";

  showDialog(
    context: context,
    builder: (_) => AlertDialog(
      contentPadding: const EdgeInsets.only(
        left: 24, right: 24, top: 20, bottom: 10),

      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(20)),

      //title
      title: Row(
        children: const [
          Icon(
            Icons.person_add,
            color: Colors.blue,
            size: 28,
          ),

```

```

        Text(' Add User')
      ],
    ),

    //content
    content: TextFormField(
      maxLines: null,
      onChanged: (value) => email = value,
      decoration: InputDecoration(
        hintText: 'Email Id',
        prefixIcon: const Icon(Icons.email, color: Colors.blue),
        border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(15))),
    ),

    //actions
    actions: [
      //cancel button
      MaterialButton(
        onPressed: () {
          //hide alert dialog
          Navigator.pop(context);
        },
        child: const Text('Cancel',
          style: TextStyle(color: Colors.blue, fontSize: 16))),

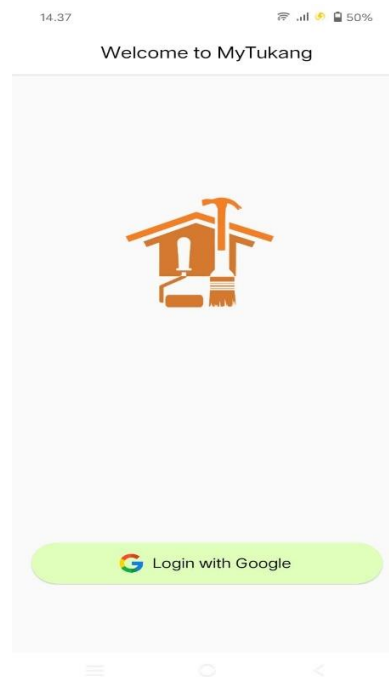
      //add button
      MaterialButton(
        onPressed: () async {
          //hide alert dialog
          Navigator.pop(context);
          if (email.isNotEmpty) {
            await APIs.addChatUser(email).then((value) {
              if (!value) {
                Dialogs.showSnackBar(
                  context, 'User does not Exists!');
              }
            });
          }
        },
        child: const Text(
          'Add',
          style: TextStyle(color: Colors.blue, fontSize: 16),
        )),
    ],
  ));
}
}

```


2. PENGGUNAAN PROGRAM

Aplikasi MyTukang adalah sebuah sistem yang dibangun dengan menggunakan framework flutter dengan Bahasa dart, untuk database menggunakan firebase. Sistem ini dibangun untuk memberikan kemudahan kepada para pencari jasa tukang dan dalam memperkiraan biaya yang harus dikeluarkan oleh pencari jasa tukang dan memudahkan tukang untuk mendapatkan customer.

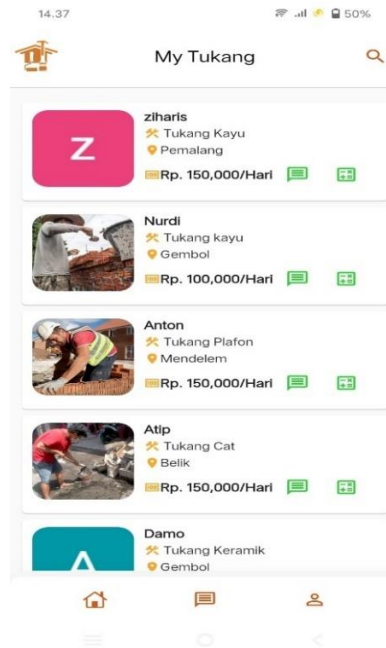
2.1 Melakukan Login



Gambar 7 - Alur Login

Tampilan *login* berupa *menu* untuk memulai masuk ke aplikasi dengan klik tombol *login with google* maka akan masuk kedalam aplikasi. Apabila belum punya akun bisa membuat akun *google* terlebih dahulu.

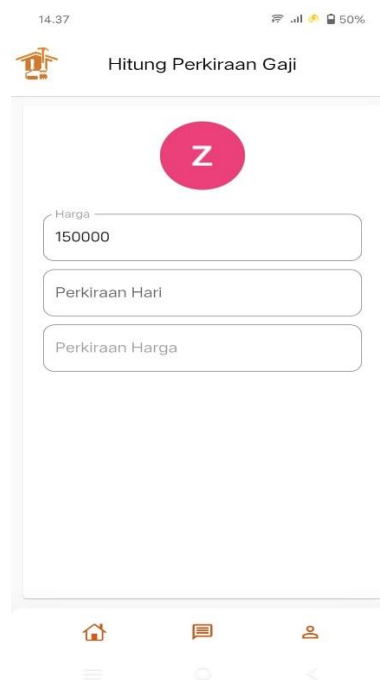
2.2 Melakukan Pencarian Tukang



Gambar 8 - Alur pencarian tukang

Pada menu home customer dapat melakukan pencarian tukang berdasarkan nama dan keahlian tukang. Dan pada masing-masing tukang tersedia icon button chat untuk pindah ke halaman chat dan calculator untuk pindah ke halaman hitung perkiraan gaji. Untuk dibawah tersedia navigasi untuk home, messages dan profile

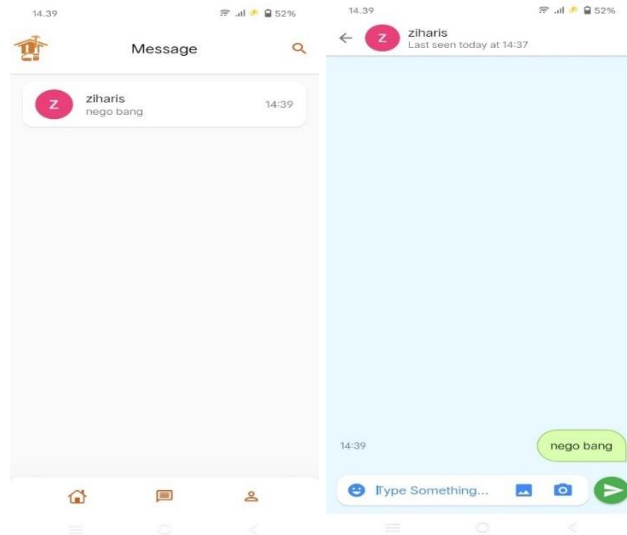
2.3 Melakukan Hitung Perkiraan Gaji



Gambar 9 - Alur Hitung Perkiraan Gaji

Setelah menemukan tukang yang diinginkan maka customer dapat mengklik icon calculator untuk pindah ke halaman hitung perkiraan gaji lalu customer memasukan perkiraan hari yang dibutuhkan oleh tukang. Maka perkiraan gaji akan muncul secara otomatis.

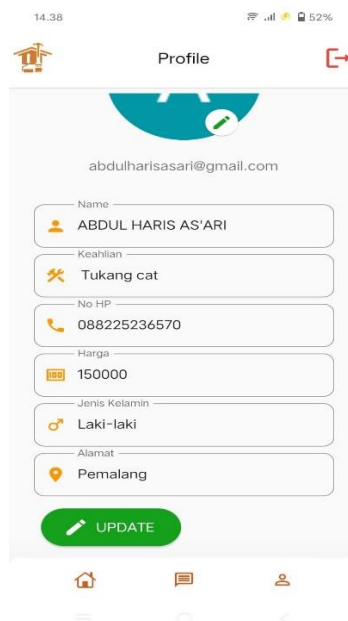
2.4 Melakukan Negosiasi Gaji



Gambar 10 - Alur Negosiasi

Apabila gaji yang telah dihitung tadi terasa agak kebesaran bisa melakukan negosiasi dengan melakukan chat kepada tukang dengan mengklik icon chat pada home di setiap tukang. Selain itu tukang dan customer juga dapat menentukan waktu pelaksanaannya.

2.5 Melakukan Edit Profile



Gambar 11 - Alur Edit profile

Pada menu profile pengguna dapat mengedit isi profile dan untuk tukang harus diisi semua seperti keahlian dan harga upah agar muncul dalam pencarian tukang. Sedangkan untuk customer dapat mengkosongkan kolom keahlian dan harga upah.

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini :

N a m a : Abdul Haris As'ari
Alamat : Rimpak RT.02, RW.06 , Kec.Belik, Kab.Pemalang Jawa Tengah

N a m a : Agus Suhendar,S.T.,M.Eng.
Alamat : Degan II, 082/041, Banjararum, Kalibawang, Kulon Progo, Yogyakarta 55672

Adalah **Pihak I** selaku pencipta, dengan ini menyerahkan karya ciptaan saya kepada :

N a m a : Universitas Teknologi Yogyakarta
Alamat : Jl. Siliwangi (Ringroad Utara), Jombor, Sleman, D.I.Yogyakarta, 55285

Adalah **Pihak II** selaku Pemegang Hak Cipta berupa Program Komputer yang berjudul Aplikasi Sistem Manajemen Jasa Tenaga Kerja Bangunan Berbasis Mobile untuk didaftarkan di Direktorat Hak Cipta dan Desain Industri, Direktorat Jenderal Kekayaan Intelektual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.

Demikianlah surat pengalihan hak ini kami buat, agar dapat dipergunakan .
sebagaimana mestinya.

Yogyakarta, 14 Juni 2023

Pemegang Hak Cipta
Rektor UTY

Dr. Bambang Moertono S, MM., Akt, CA.

Pencipta



Abdul Haris As'ari
Agus Suhendar,S.T.,M.Eng.

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta:

N a m a : 1. Abdul Haris As'ari
Kewarganegaraan : Indonesia
Alamat : Rimpak RT.02 RW.06, Kec Belik, Kab. Pemalang, Jawa Tengah
Kode Pos : 52356

N a m a : 2. Agus Suhendar,S.T.,M.Eng.
Kewarganegaraan : Indonesia
Alamat : Degan II, 082/041, Banjararum, Kalibawang, Kulon Progo, Yogyakarta, DIY
Kode Pos : 55672

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:
Berupa : Program Komputer
Berjudul : Sistem Manajemen Jasa Tenaga Kerja Bangunan Berbasis Mobile(Studi Kasus: SDM Pemalang)
 - Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyek kekayaan intelektual lainnya sebagaimana dimaksud dalam Pasal 68 ayat (2);
 - Bukan merupakan Ekspresi Budaya Tradisional sebagaimana dimaksud dalam Pasal 38;
 - Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
 - Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;
 - Bukan merupakan Ciptaan seni lukis yang berupa logo atau tanda pembeda yang digunakan sebagai merek dalam perdagangan barang/jasa atau digunakan sebagai lambang organisasi, badan usaha, atau badan hukum sebagaimana dimaksud dalam Pasal 65 dan;
 - Bukan merupakan Ciptaan yang melanggar norma agama, norma susila, ketertiban umum, pertahanan dan keamanan negara atau melanggar peraturan perundang-undangan sebagaimana dimaksud dalam Pasal 74 ayat (1) huruf d Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.
2. Sebagai pemohon mempunyai kewajiban untuk menyimpan asli contoh ciptaan yang dimohonkan dan harus memberikan apabila dibutuhkan untuk kepentingan penyelesaian sengketa perdata maupun pidana sesuai dengan ketentuan perundang-undangan.
3. Karya Cipta yang saya mohonkan pada Angka 1 tersebut di atas tidak pernah dan tidak sedang dalam sengketa pidana dan/atau perdata di Pengadilan.
4. Dalam hal ketentuan sebagaimana dimaksud dalam Angka 1 dan Angka 3 tersebut di atas saya / kami langgar, maka saya / kami bersedia secara sukarela bahwa:
 - a. permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau
 - b. Karya Cipta yang telah terdaftar dalam Daftar Umum Ciptaan Direktorat Hak Cipta, Direktorat Jenderal Hak Kekayaan Intelektual, Kementerian Hukum Dan Hak Asasi Manusia R.I dihapuskan sesuai dengan ketentuan perundang-undangan yang berlaku.
 - c. Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam berperkara, dan/atau sedang dalam gugatan di Pengadilan maka status kepemilikan surat pencatatan elektronik tersebut ditangguhkan menunggu putusan Pengadilan yang berkekuatan hukum tetap.

Demikian Surat pernyataan ini saya/kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Yogyakarta, 14 Juni 2023



1. Abdul Haris As'ari
2. Agus Suhendar, S.T., M.Eng

Pemegang Hak Cipta*

* Semua pemegang hak cipta agar menandatangani di atas materai.