

LAMPIRAN

Lampiran 1 Surat Kesiediaan Membimbing TA Pembimbing 1

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan dibawah ini:

Nama : Muhamad Bakhar, M.Kom
NIDN : 0622028602
NIPY : 04.014.179
Jabatan Struktural : Dosen Tetap
Jabatan Fungsional : Lektor

Dengan ini menyatakan bersedia untuk menjadi pembimbing I pada Tugas Akhir mahasiswa berikut:

Nama : Zidane Rizky Ramadhani
Nim : 22040045
Program Studi : DIII Teknik Komputer

Judul TA : IMPLEMENTASI SISTEM INFORMASI MOBILE DALAM
PENGAJUAN LAYANAN DISDUKCAPIL KOTA TEGAL

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

Mengetahui,
Ketua Program Studi DIII Teknik Komputer
POLITEKNIK HARAPAN BERSAMA


Ida Afriziana ST, M.Kom
NIPY: 07.011.083

Menyetujui,
Pembimbing I


Muhamad Bakhar, M.Kom
NIPY. 04.014.179

Lampiran 2 Surat Kesiediaan Membimbing TA Pembimbing 2

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan dibawah ini:

Nama : Safar Dwi Kurniawan, M.Kom
NIDN : 0624089101
NIPY : 03.021.487
Jabatan Struktural : Dosen
Jabatan Fungsional : Lektor

Dengan ini menyatakan bersedia untuk menjadi pembimbing I pada Tugas Akhir mahasiswa berikut:

Nama : Zidane Rizky Ramadhani
Nim : 22040045
Program Studi : DIII Teknik Komputer

Judul TA : IMPLEMENTASI SISTEM INFORMASI MOBILE DALAM
PENGAJUAN LAYANAN DISDUKCAPIL KOTA TEGAL

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

Mengetahui,
Ketua Program Studi DIII Teknik Komputer
POLITEKNIK HARAPAN BERSAMA


Ida Afriziana ST, M.Kom
NIPY: 07.011.083

Menyetujui,
Pembimbing I


Safar Dwi Kurniawan, M.Kom
NIPY. 03.021.487

Lampiran 3 Surat Observasi



POLITEKNIK HARAPAN BERSAMA
The True Educational Campus

D-3 Teknik Komputer

Nomor : 234.03/KOM-PHB/III/2025
Lampiran : -
Perihal : Permohonan Izin Observasi Tugas Akhir (TA)

Kepada Yth.

Kepala Dinas Kependudukan dan Pencatatan Sipil Kota Tegal

Di Jl. Lele No.14, Tegalsari, Kec. Tegal Bar., Kota Tegal, Jawa Tengah 52111

Dengan Hormat,

Sehubungan dengan tugas mata kuliah Tugas Akhir (TA) yang akan diselenggarakan di semester V Program Studi D III Teknik Komputer Politeknik Harapan Bersama Tegal, Maka dengan ini kami mengajukan izin observasi pengambilan data di Dinas Kependudukan dan Pencatatan Sipil Kota Tegal yang Bapak / Ibu Pimpin, untuk kepentingan dalam pembuatan produk Tugas Akhir, dengan Mahasiswa sebagai berikut:

No	NIM	NAMA	NO HP
1	22040045	ZIDANE RIZKY RAMADHANI	082325510421
2	22040118	FERNANDA ARYA PUTRA	088985017523

Demikian surat permohonan ini kami sampaikan atas izin dan kerjasamanya kami sampaikan terima kasih.

Tegal, 11 Maret 2025
Ka. Prodi DIII Teknik Komputer
Politeknik Harapan Bersama Tegal



Ida Afrilliana, ST, M.Kom
NIPY 12.013.168

Alamat: Jl. Mataram No. 9 Kota Tegal 52144, Jawa Tengah, Indonesia.
Telp: (0283) 352000

komputer@politektegal.ac.id
politektegal.ac.id

Lampiran 4 Source Code

```
import 'package:flutter/material.dart';
import 'package:file_picker/file_picker.dart';
import 'package:http/http.dart' as http;
import 'package:flutter/services.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'dart:typed_data';
import 'dart:convert';
import 'package:loading_animation_widget/loading_animation_widget.dart';

class FormPengajuanKTP extends StatefulWidget {
  @override
  _FormPengajuanKTPState createState() => _FormPengajuanKTPState();
}

class _FormPengajuanKTPState extends State<FormPengajuanKTP> {
  final TextEditingController namaController = TextEditingController();
  final TextEditingController nikController = TextEditingController();
  final TextEditingController noKKController = TextEditingController();
  // final TextEditingController catatanController = TextEditingController();
  final TextEditingController tempatLahirController = TextEditingController();
  final TextEditingController alamatController = TextEditingController();
  final TextEditingController pekerjaanController = TextEditingController();
  final TextEditingController golonganDarahController = TextEditingController();
  final TextEditingController teleponController = TextEditingController();
  DateTime? tanggalLahir;
  String? _selectedJenisKelamin = 'Jenis Kelamin';
  String? _selectedStatusPerkawinan = 'Pilih Status';
  String? _selectedAgama = 'Pilih Agama';
  String? _selectedTujuan = 'Pilih Tujuan';

  List<PlatformFile> dokumenFiles = [];
  List<PlatformFile> gambarFiles = [];

  bool isLoading = false;

  Future pickFiles({required bool isImage}) async {
    try {
      FilePickerResult? result = await FilePicker.platform.pickFiles(
        type: FileType.custom,
        allowedExtensions:
          isImage ? ['jpg', 'jpeg', 'png'] : ['pdf', 'doc', 'docx', 'txt'],
        allowMultiple: true,
        withData: true,
      );
      if (result != null) {
        setState(() {
          if (isImage) {
            gambarFiles.addAll(result.files);
          } else {
            dokumenFiles.addAll(result.files);
          }
        });
      }
    } catch (e) {
```

```

ScaffoldMessenger.of(context).showSnackBar(
  SnackBar(content: Text('Error picking file: $e')),
);
}
}

Future<void> submitForm() async {
  String nikInput = nikController.text.trim();
  String kkInput = noKKController.text.trim();

  if (namaController.text.isEmpty ||
    nikInput.isEmpty ||
    noKKController.text.isEmpty ||
    alamatController.text.isEmpty ||
    tempatLahirController.text.isEmpty ||
    pekerjaanController.text.isEmpty ||
    golonganDarahController.text.isEmpty ||
    teleponController.text.isEmpty ||
    tanggalLahir == null ||
    _selectedJenisKelamin == 'Jenis Kelamin' ||
    _selectedStatusPerkawinan == 'Pilih Status' ||
    _selectedAgama == 'Pilih Agama' ||
    _selectedTujuan == 'Pilih Tujuan') {
    _showErrorDialog('Silahkan isi semua data dengan benar!');
    return;
  }

  if (_selectedJenisKelamin == 'Jenis Kelamin') {
    _showErrorDialog('Silakan pilih jenis kelamin!');
    return;
  }

  if (_selectedStatusPerkawinan == 'Pilih Status') {
    _showErrorDialog('Silakan pilih status!');
    return;
  }

  if (_selectedAgama == 'Pilih Agama') {
    _showErrorDialog('Silakan pilih agama!');
    return;
  }

  if (_selectedTujuan == 'Pilih Tujuan') {
    _showErrorDialog('Silakan pilih Tujuan!');
    return;
  }

  if (nikInput.length != 16 || !RegExp(r'^\d+$').hasMatch(nikInput)) {
    _showErrorDialog('NIK harus terdiri dari 16 digit angka!');
    return;
  }

  if (kkInput.length != 16 || !RegExp(r'^\d+$').hasMatch(kkInput)) {
    _showErrorDialog('KK harus terdiri dari 16 digit angka!');
    return;
  }
}

```

```

    }

    if (teleponController.text.length > 12) {
        _showErrorDialog('Nomor telepon terdiri dari 12 digit angka!');
        return;
    }

    setState(() {
        isLoading = true;
    });

    try {
        final prefs = await SharedPreferences.getInstance();
        String? token = prefs.getString('auth_token');

        if (token == null) {
            ScaffoldMessenger.of(context).showSnackBar(
                SnackBar(
                    content: Text('Token tidak ditemukan, silakan login ulang!'),
                );
            return;
        }

        var uri = Uri.parse('http://127.0.0.1:8000/api/ktp');
        var request = http.MultipartRequest('POST', uri);

        request.headers['Authorization'] = 'Bearer $token';
        request.headers['Accept'] = 'application/json';

        request.fields['nama'] = namaController.text.trim();
        request.fields['nik'] = nikInput;
        request.fields['no_kk'] = kkInput;
        request.fields['jenis_kelamin'] = _selectedJenisKelamin!;
        request.fields['tempat_lahir'] = tempatLahirController.text.trim();
        request.fields['alamat'] = alamatController.text.trim();
        request.fields['tanggal_lahir'] =
            tanggalLahir!.toIso8601String().split('T').first;
        request.fields['status_perkawinan'] = _selectedStatusPerkawinan!;
        request.fields['tujuan'] = _selectedTujuan!;
        request.fields['pekerjaan'] = pekerjaanController.text.trim();
        request.fields['agama'] = _selectedAgama!;
        request.fields['gol_darah'] = golonganDarahController.text.trim();
        request.fields['telepon'] = teleponController.text.trim();

        for (var dokumen in dokumenFiles) {
            request.files.add(http.MultipartFile.fromBytes(
                'dokumen[]',
                dokumen.bytes!,
                filename: dokumen.name,
            ));
        }

        for (var gambar in gambarFiles) {
            request.files.add(http.MultipartFile.fromBytes(
                'gambar[]',

```

```

        gambar.bytes!,
        filename: gambar.name,
    ));
}

var response = await request.send();

if (response.statusCode == 201) {
    final responseBody = await response.stream.bytesToString();
    final jsonResponse = jsonDecode(responseBody);

    String kodePengajuan =
        jsonResponse['kode_pengajuan'] ?? 'Tidak ada kode';

    _showSuccessDialog(context, kodePengajuan);
} else {
    final responseBody = await response.stream.bytesToString();
    print('Error Response: $responseBody');
    _showErrorDialog(
        'Mohon maaf, terjadi kesalahan saat mengirim pengajuan.');
```

```

    }
} catch (e) {
    print('Exception: $e');
    ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('Terjadi kesalahan: $e')),
    );
} finally {
    setState(() {
        isLoading = false;
    });
}
}

@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            leading: IconButton(
                icon: Icon(Icons.arrow_back, color: Color(0xFF27488F)),
                onPressed: () {
                    Navigator.pop(context);
                },
            ),
            title: Text(
                'FORM PENGAJUAN KTP',
                style: TextStyle(
                    color: Color(0xFF27488F),
                    fontWeight: FontWeight.w500,
                    fontFamily: 'BakbakOne',
                ),
            ),
            centerTitle: true,
            backgroundColor: Color(0xFFE7DDF1),
        ),
        body: SingleChildScrollView(

```

```

scrollDirection: Axis.vertical,
child: Padding(
  padding: EdgeInsets.all(16.0),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      buildLabel('NAMA LENGKAP'),
      buildInputField(namaController, 'Masukkan nama lengkap anda',
        inputFormatters: [
          FilteringTextInputFormatter.allow(RegExp(r'^[a-zA-Z\s]*$')),
        ],
        SizedBox(height: 10),
      buildLabel('NOMOR INDUK KEPENDUDUK'),
      buildInputField(nikController, 'Masukkan NIK anda',
        isNumeric: true, maxLength: 16),
      SizedBox(height: 10),
      buildLabel('NOMOR KARTU KELUARGA'),
      buildInputField(noKKController, 'Masukkan KK anda',
        isNumeric: true, maxLength: 16),
      SizedBox(height: 15),
      _buildJenisKelamin(height: 50),
      SizedBox(height: 10),
      buildLabel('TANGGAL LAHIR'),
      buildDatePicker(),
      SizedBox(height: 10),
      buildLabel('TEMPAT LAHIR'),
      buildInputField(tempatLahirController, 'Masukkan tempat lahir',
        inputFormatters: [
          FilteringTextInputFormatter.allow(RegExp(r'^[a-zA-Z\s]*$')),
        ],
        SizedBox(height: 10),
      buildLabel('ALAMAT'),
      buildInputField(alamatController, 'Masukkan alamat', maxLines: 4),
      SizedBox(height: 15),
      _buildStatusPerkawinan(height: 50),
      SizedBox(height: 10),
      buildLabel('PEKERJAAN'),
      buildInputField(pekerjaanController, 'Masukkan pekerjaan',
        inputFormatters: [
          FilteringTextInputFormatter.allow(RegExp(r'^[a-zA-Z\s]*$')),
        ],
        SizedBox(height: 15),
      _buildAgama(height: 50),
      SizedBox(height: 10),
      buildLabel('GOLONGAN DARAH'),
      buildInputField(
        golonganDarahController, 'Masukkan golongan darah',
        inputFormatters: [
          FilteringTextInputFormatter.allow(RegExp(r'^[a-zA-Z\s]*$')),
        ],
        SizedBox(height: 15),
      _buildTujuan(height: 50),
      SizedBox(height: 10),
      buildLabel('NOMOR TELEPON'),
      buildInputField(teleponController, 'Masukkan no telepon anda',

```

```

        isNumeric: true, maxLength: 12),
    SizedBox(height: 30),
    buildFilePickerButton(
      'Upload Dokumen Anda',
      dokumenFiles,
      () => pickFiles(isImage: false),
    ),
    SizedBox(height: 5),
    Text(
      'Type File: pdf, doc, docx',
      style: TextStyle(
        fontFamily: 'Baloo2',
        color: Color(0xFFFF90707),
        fontSize: 17,
      ),
    ),
    SizedBox(height: 20),
    buildFilePickerButton(
      'Upload Gambar Anda',
      gambarFiles,
      () => pickFiles(isImage: true),
    ),
    SizedBox(height: 5),
    Text(
      'Type File: jpg, jpeg, png',
      style: TextStyle(
        fontFamily: 'Baloo2',
        color: Color(0xFFFF90707),
        fontSize: 17,
      ),
    ),
    SizedBox(height: 25),
    SizedBox(
      width: double.infinity,
      height: 49,
      child: ElevatedButton(
        onPressed: isLoading ? null : submitForm,
        style: ElevatedButton.styleFrom(
          padding: const EdgeInsets.symmetric(
            vertical: 15, horizontal: 20),
          backgroundColor: const Color(0xFF1DE01D),
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(10),
          ),
        ),
        child: isLoading
          ? LoadingAnimationWidget.discreteCircle(
              color: const Color(0xFF27488F), // Warna utama biru
              secondRingColor:
                Color(0xFFFFFDF00), // Warna cincin kedua
              thirdRingColor:
                Color(0xFFFF90707), // Warna cincin ketiga
              size: 24,
            )
          : const Text(

```

```

        'KIRIM PENGAJUAN',
        style: TextStyle(
          fontSize: 20,
          fontWeight: FontWeight.w500,
          fontFamily: 'BakbakOne',
          color: Colors.white,
        ),
      ),
    ),
  ],
),
),
);
}

```

```

Widget _buildAgama({double height = 50}) {
  return Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text(
        'AGAMA',
        style: TextStyle(
          color: Color(0xFF27488F),
          fontSize: 16,
          fontWeight: FontWeight.w500,
          fontFamily: 'BakbakOne',
        ),
      ),
      SizedBox(height: 5),
      Container(
        height: height,
        child: DropdownButtonFormField<String>(
          value: _selectedAgama,
          decoration: InputDecoration(
            filled: true,
            fillColor: const Color(0xFFF4E7E7),
            border: OutlineInputBorder(
              borderRadius: BorderRadius.circular(10),
            ),
            focusedBorder: OutlineInputBorder(
              borderRadius: BorderRadius.circular(10),
              borderSide: const BorderSide(
                color: Color(0xFF27488F),
                width: 2.0,
              ),
            ),
          ),
          contentPadding: EdgeInsets.symmetric(horizontal: 12, vertical: 8),
        ),
        items: [
          'Pilih Agama',
          'Islam',
          'Kristen Protestan',
          'Kristen Katolik',

```

```

        'Hindu',
        'Buddha',
        'Konghucu',
        'Kepercayaan terhadap Tuhan YME',
    ].map((String item) {
        return DropdownMenuItem<String>(
            value: item,
            child: Text(item,
                style: TextStyle(
                    fontSize: 16,
                    fontWeight: FontWeight.w900,
                    fontFamily: 'Baloo2',
                    color: Color(0xFF27488F))),
        );
    }).toList(),
    onChanged: (val) {
        setState(() {
            _selectedAgama = val;
        });
    },
),
),
],
);
}

```

```

Widget _buildJenisKelamin({double height = 50}) {
    return Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
            Text(
                'JENIS KELAMIN',
                style: TextStyle(
                    color: Color(0xFF27488F),
                    fontSize: 16,
                    fontWeight: FontWeight.w500,
                    fontFamily: 'BakbakOne',
                ),
            ),
            SizedBox(height: 5),
            Container(
                height: height,
                child: DropdownButtonFormField<String>(
                    value: _selectedJenisKelamin,
                    decoration: InputDecoration(
                        filled: true,
                        fillColor: const Color(0xFFFF4E7E7),
                        border: OutlineInputBorder(
                            borderRadius: BorderRadius.circular(10),
                        ),
                        focusedBorder: OutlineInputBorder(
                            borderRadius: BorderRadius.circular(10),
                            borderSide: const BorderSide(
                                color: Color(0xFF27488F),
                                width: 2.0,

```

```

    ),
  ),
  contentPadding: EdgeInsets.symmetric(horizontal: 12, vertical: 8),
),
items: [
  'Jenis Kelamin',
  'Laki-laki',
  'Perempuan',
].map((String item) {
  return DropdownMenuItem<String>({
    value: item,
    child: Text(item,
      style: TextStyle(
        fontSize: 16,
        fontWeight: FontWeight.w900,
        fontFamily: 'Baloo2',
        color: Color(0xFF27488F))),
  );
}).toList(),
onChanged: (val) {
  setState(() {
    _selectedJenisKelamin = val;
  });
},
),
),
],
);
}

```

```

Widget _buildStatusPerkawinan({double height = 50}) {
  return Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text(
        'STATUS PERKAWINAN',
        style: TextStyle(
          color: Color(0xFF27488F),
          fontSize: 16,
          fontWeight: FontWeight.w500,
          fontFamily: 'BakbakOne',
        ),
      ),
      SizedBox(height: 5),
      Container(
        height: height,
        child: DropdownButtonFormField<String>({
          value: _selectedStatusPerkawinan,
          decoration: InputDecoration(
            filled: true,
            fillColor: const Color(0xFFF4E7E7),
            border: OutlineInputBorder(
              borderRadius: BorderRadius.circular(10),
            ),
            focusedBorder: OutlineInputBorder(

```

```

        borderRadius: BorderRadius.circular(10),
        borderSide: const BorderSide(
          color: Color(0xFF27488F),
          width: 2.0,
        ),
      ),
      contentPadding: EdgeInsets.symmetric(horizontal: 12, vertical: 8),
    ),
    items: [
      'Pilih Status',
      'Belum Menikah',
      'Menikah',
      'Ceraid Hidup',
      'Ceraid Mati',
    ].map((String item) {
      return DropdownMenuItem<String>(
        value: item,
        child: Text(item,
          style: TextStyle(
            fontSize: 16,
            fontWeight: FontWeight.w900,
            fontFamily: 'Baloo2',
            color: Color(0xFF27488F))),
      );
    }).toList(),
    onChanged: (val) {
      setState(() {
        _selectedStatusPerkawinan = val;
      });
    },
  ),
),
],
);
}

```

```

Widget _buildTujuan({double height = 50}) {
  return Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text(
        'TUJUAN',
        style: TextStyle(
          color: Color(0xFF27488F),
          fontSize: 16,
          fontWeight: FontWeight.w500,
          fontFamily: 'BakbakOne',
        ),
      ),
      SizedBox(height: 5),
      Container(
        height: height,
        child: DropdownButtonFormField<String>(
          value: _selectedTujuan,
          decoration: InputDecoration(

```

```

        filled: true,
        fillColor: const Color(0xFFFF4E7E7),
        border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(10),
        ),
        focusedBorder: OutlineInputBorder(
          borderRadius: BorderRadius.circular(10),
          borderSide: const BorderSide(
            color: Color(0xFF27488F),
            width: 2.0,
          ),
        ),
        contentPadding: EdgeInsets.symmetric(horizontal: 12, vertical: 8),
      ),
      items: [
        'Pilih Tujuan',
        'Pembuatan KTP Baru',
        'Perekaman',
        'KTP Hilang',
        'Perubahan Data',
      ].map((String item) {
        return DropdownMenuItem<String>({
          value: item,
          child: Text(item,
            style: TextStyle(
              fontSize: 16,
              fontWeight: FontWeight.w900,
              fontFamily: 'Baloo2',
              color: Color(0xFF27488F))),
        });
      }).toList(),
      onChanged: (val) {
        setState(() {
          _selectedTujuan = val;
        });
      },
    ),
  ],
);
}

```

```

Widget buildLabel(String text) {
  return Padding(
    padding: EdgeInsets.only(top: 10, bottom: 5),
    child: Text(
      text,
      style: TextStyle(
        fontWeight: FontWeight.w500,
        fontFamily: 'BakbakOne',
        fontSize: 15,
        color: Color(0xFF27488F),
      ),
    ),
  );
}

```

```

}

Widget buildInputField(TextEditingController controller, String hint,
  {int maxLines = 1,
  bool isNumeric = false,
  int? maxLength,
  List<TextInputFormatter>? inputFormatters}) {
return TextFormField(
  controller: controller,
  maxLines: maxLines,
  maxLength: maxLength,
  keyboardType: isNumeric ? TextInputType.number : TextInputType.text,
  inputFormatters: inputFormatters ??
    (isNumeric
      ? [
        FilteringTextInputFormatter.digitsOnly,
        if (maxLength != null)
          LengthLimitingTextInputFormatter(maxLength),
      ]
      : []),
  style: const TextStyle(
    color: Color(0xFF27488F),
    fontWeight: FontWeight.w500,
    fontFamily: 'Baloo2',
  ),
  decoration: InputDecoration(
    hintText: hint,
    fillColor: const Color(0xFFF4E7E7),
    filled: true,
    contentPadding:
      const EdgeInsets.symmetric(vertical: 15, horizontal: 10),
    border: OutlineInputBorder(
      borderRadius: BorderRadius.circular(10),
    ),
    focusedBorder: OutlineInputBorder(
      borderRadius: BorderRadius.circular(10),
      borderSide: const BorderSide(
        color: Color(0xFF27488F),
        width: 2.0,
      ),
    ),
    counterText: "",
  ),
);
}

```

```

Widget buildDatePicker() {
return Padding(
  padding: const EdgeInsets.only(bottom: 10),
  child: InkWell(
    onTap: () async {
      DateTime? pickedDate = await showDatePicker(
        context: context,
        initialDate: DateTime.now(),
        firstDate: DateTime(0000),

```

```

        lastDate: DateTime(9999),
    );
    if (pickedDate != null) {
        setState(() {
            tanggalLahir = pickedDate;
        });
    }
},
child: Container(
    width: double.infinity,
    padding: const EdgeInsets.symmetric(vertical: 10, horizontal: 10),
    decoration: BoxDecoration(
        color: const Color(0xFFFF4E7E7),
        borderRadius: BorderRadius.circular(10),
        border: Border.all(color: Colors.black, width: 1.0),
    ),
    child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
            Text(
                tanggalLahir == null
                ? 'Pilih Tanggal Lahir'
                : '${tanggalLahir!.toLocal()}'.split(' ')[0],
                style: const TextStyle(
                    color: Color(0xFF27488F),
                    fontSize: 16,
                    fontWeight: FontWeight.w500,
                    fontFamily: 'Baloo2',
                ),
            ),
            const Icon(Icons.calendar_today, color: Color(0xFF27488F)),
        ],
    ),
),
);
}

```

```

Widget buildFilePickerButton(
    String title, List<PlatformFile> files, VoidCallback onPressed) {
    return Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
            SizedBox(
                width: double.infinity,
                height: 40,
                child: ElevatedButton(
                    onPressed: onPressed,
                    style: ElevatedButton.styleFrom(
                        backgroundColor: const Color(0xFFFF4E7E7),
                        shape: RoundedRectangleBorder(
                            borderRadius: BorderRadius.circular(10),
                            side: const BorderSide(
                                color: Colors.black,
                                width: 0.5,

```

```

    ),
  ),
),
child: Text(
  files.isEmpty ? title : files.map((file) => file.name).join(', '),
  style: const TextStyle(
    color: Color(0xFF27488F),
    fontSize: 16,
    fontWeight: FontWeight.w500,
    overflow: TextOverflow.ellipsis,
  ),
  maxLines: 1,
),
),
),
],
);
}

void _showSuccessDialog(BuildContext context, String kodePengajuan) {
  showDialog(
    context: context,
    builder: (BuildContext context) {
      return AlertDialog(
        title: const Text(
          'Pengajuan Berhasil!',
          style:
            TextStyle(color: Colors.yellowAccent, fontFamily: 'BakbakOne'),
        ),
        content: Column(
          mainAxisAlignment: MainAxisAlignment.min,
          children: [
            const Text(
              'Kode Pengajuan Anda:',
              style: TextStyle(color: Colors.white, fontFamily: 'BakbakOne'),
            ),
            const SizedBox(height: 8),
            SelectableText(
              kodePengajuan,
              style: const TextStyle(
                color: Colors.yellowAccent,
                fontWeight: FontWeight.bold,
                fontSize: 18,
              ),
              textAlign: TextAlign.center,
            ),
            const SizedBox(height: 12),
            ElevatedButton.icon(
              onPressed: () {
                Clipboard.setData(ClipboardData(text: kodePengajuan));
              },
              icon: const Icon(Icons.copy),
              label: const Text('Salin Kode'),
              style: ElevatedButton.styleFrom(
                backgroundColor: Colors.yellow[700],

```

```

        foregroundColor: Colors.black,
      ),
    ),
  ],
),
actions: [
  Align(
    alignment: Alignment.center,
    child: TextButton(
      style: TextButton.styleFrom(
        foregroundColor: Colors.white,
      ),
      onPressed: () {
        Navigator.of(context).pop();
        Navigator.pop(context);
      },
      child: const Text(
        'OK',
        style: TextStyle(
          color: Colors.yellowAccent, fontFamily: 'BakbakOne'),
      ),
    ),
  ),
],
backgroundColor: Colors.green[700],
);
},
);
}

void _showErrorDialog(String message) {
  showDialog(
    context: context,
    builder: (BuildContext context) {
      return AlertDialog(
        title: const Text(
          'Pengajuan Gagal!',
          style:
            TextStyle(color: Colors.yellowAccent, fontFamily: 'BakbakOne'),
        ),
        content: Text(
          message,
          style: TextStyle(color: Colors.white, fontFamily: 'BakbakOne'),
        ),
        actions: [
          Align(
            alignment: Alignment.center,
            child: TextButton(
              onPressed: () {
                Navigator.of(context).pop();
              },
              child: const Text(
                'OK',
                style: TextStyle(
                  color: Colors.yellowAccent, fontFamily: 'BakbakOne'),

```

```
        ),  
        ),  
        ),  
    ],  
    backgroundColor: Colors.red[700],  
);  
},  
);  
}  
}
```

Lampiran 5 Foto Dokumentasi

