

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 : Fernanda Arya Putra
Nim : 22040118
Program Studi : DIII Teknik Komputer

Judul TA : RANCANG BANGUN SISTEM INFORMASI PENGAJUAN
DOKUMEN ONLINE PADA DISDUKCAPIL KOTA TEGAL
BERBASIS WEB MENGGUNAKAN LARAVEL

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

Mengetahui,
Ketua Program Studi DIII Teknik Komputer
POLITEKNIK HARAPAN BERSAMA


Ida Afriliana 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 : Fernanda Arya Putra
Nim : 22040118
Program Studi : DIII Teknik Komputer

Judul TA : RANCANG BANGUN SISTEM INFORMASI PENGAJUAN
DOKUMEN ONLINE PADA DISDUKCAPIL KOTA TEGAL
BERBASIS WEB MENGGUNAKAN LARAVEL

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

Mengetahui,
Ketua Program Studi DIII Teknik Komputer
POLITEKNIK HARAPAN BERSAMA



Ida Afriliana 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 Realized 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 Afriliana, ST, M.Kom
NIPY 12.013.168

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

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

 Dipindai dengan CamScanner

Lampiran 4 Source Code

```
public function aktakelahiran_submit(Request $request)
{
    // Validasi input
    $request->validate([
        'nama' => 'required|string|max:255',
        'jenis_kelamin' => 'required|in:Laki-laki,Perempuan',
        'tempat_lahir' => 'required|string|max:255',
        'tanggal_lahir' => 'required|date',
        'agama' => 'required|in:Islam,Kristen Protestan,Kristen
Katolik,Hindu,Buddha,Konghucu,Kepercayaan terhadap Tuhan YME',

        // Data Ayah
        'nama_ayah' => 'required|string|max:255',
        'nik_ayah' => 'required|numeric|digits:16',
        'no_kk_ayah' => 'required|numeric|digits:16',
        'tempat_lahir_ayah' => 'required|string|max:255',
        'tanggal_lahir_ayah' => 'required|date',
        'pekerjaan_ayah' => 'required|string|max:255',
        'alamat_ayah' => 'required|string',

        // Data Ibu
        'nama_ibu' => 'required|string|max:255',
        'nik_ibu' => 'required|numeric|digits:16',
        'no_kk_ibu' => 'required|numeric|digits:16',
        'tempat_lahir_ibu' => 'required|string|max:255',
        'tanggal_lahir_ibu' => 'required|date',
        'pekerjaan_ibu' => 'required|string|max:255',
        'alamat_ibu' => 'required|string',

        // Data Saksi
        'nama_saksi1' => 'required|string|max:255',
        'nik_saksi1' => 'required|numeric|digits:16',
        'alamat_saksi1' => 'required|string',
        'nama_saksi2' => 'required|string|max:255',
        'nik_saksi2' => [
            'required', 'numeric', 'digits:16',
            function ($attribute, $value, $fail) use ($request) {
                if ($value === $request->nik_saksi_1) {
                    $fail('NIK saksi 2 tidak boleh sama dengan
saksi 1.');
```

```

        'nama.required' => 'Nama wajib diisi.',
        'jenis_kelamin.required' => 'Jenis kelamin wajib diisi.',
        'jenis_kelamin.in' => 'Jenis kelamin harus "Laki-laki"
atau "Perempuan".',
        'tempat_lahir.required' => 'Tempat lahir wajib diisi.',
        'tanggal_lahir.required' => 'Tanggal lahir wajib diisi.',
        'agama.required' => 'Agama wajib diisi.',
        'agama.in' => 'Agama harus salah satu dari: Islam, Kristen
Protestan, Kristen Katolik, Hindu, Buddha, Konghucu, Kepercayaan
terhadap Tuhan YME.',

        // Ayah
        'nama_ayah.required' => 'Nama ayah wajib diisi.',
        'nik_ayah.required' => 'NIK ayah wajib diisi.',
        'nik_ayah.digits' => 'NIK ayah harus berisi tepat 16
angka.',
        'no_kk_ayah.required' => 'Nomor KK ayah wajib diisi.',
        'pekerjaan_ayah.required' => 'Pekerjaan ayah wajib
diisi.',

        // Ibu
        'nama_ibu.required' => 'Nama ibu wajib diisi.',
        'nik_ibu.required' => 'NIK ibu wajib diisi.',
        'nik_ibu.digits' => 'NIK ibu harus berisi tepat 16
angka.',
        'no_kk_ibu.required' => 'Nomor KK ibu wajib diisi.',
        'pekerjaan_ibu.required' => 'Pekerjaan ibu wajib diisi.',

        // Saksi
        'nama_saksi1.required' => 'Nama saksi 1 wajib diisi.',
        'nik_saksi1.required' => 'NIK saksi 1 wajib diisi.',
        'nik_saksi1.digits' => 'NIK saksi 1 harus berisi tepat 16
angka.',
        'nama_saksi2.required' => 'Nama saksi 2 wajib diisi.',
        'nik_saksi2.required' => 'NIK saksi 2 wajib diisi.',
        'nik_saksi2.digits' => 'NIK saksi 2 harus berisi tepat 16
angka.',
        'nik_saksi2.different' => 'NIK saksi 2 tidak boleh sama
dengan saksi 1.',

        // Lainnya
        'status_perkawinan.required' => 'Status perkawinan wajib
diisi.',
        // 'nomor_kk.required' => 'Nomor KK wajib diisi.',
        // 'nomor_kk.digits' => 'Nomor KK harus berisi tepat 16
angka.',
        'nomor_akta_nikah.max' => 'Nomor akta nikah tidak boleh
lebih dari 255 karakter.',

        'dokumen.*.mimes' => 'Dokumen harus dalam format PDF, DOC,
atau TXT.',
        'dokumen.*.max' => 'Ukuran dokumen tidak boleh lebih dari
2MB.',
        'gambar.*.image' => 'File gambar harus berupa JPG, JPEG,
atau PNG.',
        'gambar.*.max' => 'Ukuran gambar tidak boleh lebih dari

```

```

1MB.',
]);

$jenispengajuan = "AKTA-KELAHIRAN";
$statuspengajuan = "Belum";
$catatanpengajuan = "-";

// Generate kode pengajuan
$kodePengajuan =
AktaKelahiran::generateKodePengajuan($jenispengajuan);

// Proses dokumen
$dokumenPaths = [];
if ($request->hasFile('dokumen')) {
    foreach ($request->file('dokumen') as $dokumen) {
        $path = $dokumen-
>store("Pengajuan/dokumen/akta_kelahiran", 'public');
        $dokumenPaths[] = $path;
    }
}

// Proses gambar
$gambarPaths = [];
if ($request->hasFile('gambar')) {
    foreach ($request->file('gambar') as $gambar) {
        $path = $gambar-
>store("Pengajuan/gambar/akta_kelahiran", 'public');
        $gambarPaths[] = $path;
    }
}

// Simpan ke database
$aktaKelahiran = new AktaKelahiran();
$aktaKelahiran->user_id = auth()->id();
$aktaKelahiran->kode_pengajuan = $kodePengajuan;
$aktaKelahiran->nama = $request->nama;
$aktaKelahiran->jenis_kelamin = $request->jenis_kelamin;
$aktaKelahiran->tempat_lahir = $request->tempat_lahir;
$aktaKelahiran->tanggal_lahir = $request->tanggal_lahir;
$aktaKelahiran->agama = $request->agama;
$aktaKelahiran->nama_ayah = $request->nama_ayah;
$aktaKelahiran->nik_ayah = $request->nik_ayah;
$aktaKelahiran->no_kk_ayah = $request->no_kk_ayah;
$aktaKelahiran->tempat_lahir_ayah = $request-
>tempat_lahir_ayah;
$aktaKelahiran->tanggal_lahir_ayah = $request-
>tanggal_lahir_ayah;
$aktaKelahiran->pekerjaan_ayah = $request->pekerjaan_ayah;
$aktaKelahiran->alamat_ayah = $request->alamat_ayah;
$aktaKelahiran->nama_ibu = $request->nama_ibu;
$aktaKelahiran->nik_ibu = $request->nik_ibu;
$aktaKelahiran->no_kk_ibu = $request->no_kk_ibu;
$aktaKelahiran->tempat_lahir_ibu = $request->tempat_lahir_ibu;
$aktaKelahiran->tanggal_lahir_ibu = $request-
>tanggal_lahir_ibu;
$aktaKelahiran->pekerjaan_ibu = $request->pekerjaan_ibu;

```

```

    $aktaKelahiran->alamat_ibu = $request->alamat_ibu;
    $aktaKelahiran->nama_saksi_1 = $request->nama_saksi1;
    $aktaKelahiran->nik_saksi_1 = $request->nik_saksi1;
    $aktaKelahiran->alamat_saksi_1 = $request->alamat_saksi1;
    $aktaKelahiran->nama_saksi_2 = $request->nama_saksi2;
    $aktaKelahiran->nik_saksi_2 = $request->nik_saksi2;
    $aktaKelahiran->alamat_saksi_2 = $request->alamat_saksi2;
    $aktaKelahiran->status_perkawinan = $request->
>status_perkawinan;
    $aktaKelahiran->nomor_akta_nikah = $request->nomor_akta_nikah;
    $aktaKelahiran->dokumen = json_encode($dokumenPaths);
    $aktaKelahiran->gambar = json_encode($gambarPaths);
    $aktaKelahiran->jenis = $jenispengajuan;
    $aktaKelahiran->status = $statuspengajuan;
    $aktaKelahiran->catatan = $catatanpengajuan;

    $aktaKelahiran->save();

    return redirect('/menu-layanan')->with('success_pengajuan',
"Pengajuan berhasil dikirim! Kode Pengajuan Anda:
$kodePengajuan");
}
public function aktakelahiran_update(Request $request,
$kode_pengajuan)
{
    // Cari data berdasarkan ID
    $aktaKelahiran = AktaKelahiran::where('kode_pengajuan', '=',
$kode_pengajuan)->first();

    // Update data
    $aktaKelahiran->nama = $request->nama;
    $aktaKelahiran->jenis_kelamin = $request->jenis_kelamin;
    $aktaKelahiran->tempat_lahir = $request->tempat_lahir;
    $aktaKelahiran->tanggal_lahir = $request->tanggal_lahir;
    $aktaKelahiran->agama = $request->agama;
    $aktaKelahiran->nama_ayah = $request->nama_ayah;
    $aktaKelahiran->nik_ayah = $request->nik_ayah;
    $aktaKelahiran->no_kk_ayah = $request->no_kk_ayah;
    $aktaKelahiran->tempat_lahir_ayah = $request->
>tempat_lahir_ayah;
    $aktaKelahiran->tanggal_lahir_ayah = $request->
>tanggal_lahir_ayah;
    $aktaKelahiran->pekerjaan_ayah = $request->pekerjaan_ayah;
    $aktaKelahiran->alamat_ayah = $request->alamat_ayah;
    $aktaKelahiran->nama_ibu = $request->nama_ibu;
    $aktaKelahiran->nik_ibu = $request->nik_ibu;
    $aktaKelahiran->no_kk_ibu = $request->no_kk_ibu;
    $aktaKelahiran->tempat_lahir_ibu = $request->tempat_lahir_ibu;
    $aktaKelahiran->tanggal_lahir_ibu = $request->
>tanggal_lahir_ibu;
    $aktaKelahiran->pekerjaan_ibu = $request->pekerjaan_ibu;
    $aktaKelahiran->alamat_ibu = $request->alamat_ibu;
    $aktaKelahiran->nama_saksi_1 = $request->nama_saksi_1;
    $aktaKelahiran->nik_saksi_1 = $request->nik_saksi_1;
    $aktaKelahiran->alamat_saksi_1 = $request->alamat_saksi_1;

```



```

        $aktaKelahiran->nama_saksi_2 = $request->nama_saksi_2;
        $aktaKelahiran->nik_saksi_2 = $request->nik_saksi_2;
        $aktaKelahiran->alamat_saksi_2 = $request->alamat_saksi_2;
        $aktaKelahiran->status_perkawinan = $request->status_perkawinan;
    }

    // Simpan dokumen baru berdasarkan jenis pengajuan
    if ($request->hasFile('dokumen')) {
        $dokumenPaths = []; // Array untuk menyimpan path dokumen
        foreach ($request->file('dokumen') as $file) {
            // Simpan file dokumen di folder yang sesuai
            $filename = $file->getClientOriginalName();
            >store("Pengajuan/dokumen/akta_kelahiran", 'public');
            $dokumenPaths[] = $filename;
        }

        // Jika dokumen sebelumnya ada, gabungkan dengan dokumen
        $existingDokumen = json_decode($aktaKelahiran->dokumen, true) ?? [];
        $aktaKelahiran->dokumen = json_encode(array_merge($existingDokumen, $dokumenPaths));
    }

    // Simpan gambar baru berdasarkan jenis pengajuan
    if ($request->hasFile('gambar')) {
        $gambarPaths = []; // Array untuk menyimpan path gambar
        foreach ($request->file('gambar') as $file) {
            // Simpan file gambar di folder yang sesuai
            $filename = $file->getClientOriginalName();
            >store("Pengajuan/gambar/akta_kelahiran", 'public');
            $gambarPaths[] = $filename;
        }

        // Jika gambar sebelumnya ada, gabungkan dengan gambar
        $existingGambar = json_decode($aktaKelahiran->gambar, true) ?? [];
        $aktaKelahiran->gambar = json_encode(array_merge($existingGambar, $gambarPaths));
    }

    // Simpan perubahan
    $aktaKelahiran->save();

    return redirect('/pengajuan-user')->with('success', "Pengajuan berhasil diperbarui!");
}

public function cekStatus(Request $request)
{

```

```

// Validasi input
$request->validate([
    'kode_pengajuan' => 'required|string|max:255',
], [
    'kode_pengajuan.required' => 'Kode pengajuan wajib
diisi.',
]);

// Cari kode_pengajuan yang benar-benar sama persis
$kode = $request->kode_pengajuan;

// Akta kematian
$pengajuanAktaKematian = AktaKematian::where('kode_pengajuan',
'=', $kode)->first();
if ($pengajuanAktaKematian) {
    $penilaian = Penilaian::where('kode_pengajuan', $kode)
->where('user_id', auth()->id())
->first();
    return view('detail-pengajuan.detail-kematian',
['akta_kematian' => $pengajuanAktaKematian, 'penilaian' =>
$penilaian]);
}
// Kedatangan
$pengajuanKedatangan = Kedatangan::where('kode_pengajuan',
'=', $kode)->first();
if ($pengajuanKedatangan) {
    $penilaian = Penilaian::where('kode_pengajuan', $kode)
->where('user_id', auth()->id())
->first();
    return view('detail-pengajuan.detail-kedatangan',
['kedatangan' => $pengajuanKedatangan, 'penilaian' =>
$penilaian]);
}
// AktaKelahiran
$pengajuanAktaKelahiran =
AktaKelahiran::where('kode_pengajuan', '=', $kode)->first();
if ($pengajuanAktaKelahiran) {
    $penilaian = Penilaian::where('kode_pengajuan', $kode)
->where('user_id', auth()->id())
->first();
    return view('detail-pengajuan.detail-kelahiran',
['akta_kelahiran' => $pengajuanAktaKelahiran, 'penilaian' =>
$penilaian]);
}
// Perpindahan
$pengajuanPerpindahan = Perpindahan::where('kode_pengajuan',
'=', $kode)->first();
if ($pengajuanPerpindahan) {
    $penilaian = Penilaian::where('kode_pengajuan', $kode)
->where('user_id', auth()->id())
->first();
    return view('detail-pengajuan.detail-perpindahan', [
        'perpindahan' => $pengajuanPerpindahan,
        'penilaian' => $penilaian
    ]);
}

```

```

        // Pernikahan
        $pengajuanPernikahan = Pernikahan::where('kode_pengajuan',
        '=', $kode)->first();
        if ($pengajuanPernikahan) {
            $penilaian = Penilaian::where('kode_pengajuan', $kode)
            ->where('user_id', auth()->id())
            ->first();
            return view('detail-pengajuan.detail-pernikahan', [
                'pernikahan' => $pengajuanPernikahan,
                'penilaian' => $penilaian
            ]);
        }
        // KartuIdentitasAnak
        $pengajuanKia = KartuIdentitasAnak::where('kode_pengajuan',
        '=', $kode)->first();
        if ($pengajuanKia) {
            $penilaian = Penilaian::where('kode_pengajuan', $kode)
            ->where('user_id', auth()->id())
            ->first();
            return view('detail-pengajuan.detail-kia', [
                'kia' => $pengajuanKia,
                'penilaian' => $penilaian
            ]);
        }

        // KTP
        $pengajuanKTP = KartuTandaPenduduk::where('kode_pengajuan',
        '=', $kode)->first();
        if ($pengajuanKTP) {
            $penilaian = Penilaian::where('kode_pengajuan', $kode)
            ->where('user_id', auth()->id())
            ->first();
            return view('detail-pengajuan.detail-ktp', [
                'pengajuan' => $pengajuanKTP,
                'penilaian' => $penilaian
            ]);
        }

        // KartuKeluarga
        $pengajuanKK = KartuKeluarga::where('kode_pengajuan', '=',
        $kode)->first();
        if ($pengajuanKK) {
            $penilaian = Penilaian::where('kode_pengajuan', $kode)
            ->where('user_id', auth()->id())
            ->first();
            return view('detail-pengajuan.detail-kk', [
                'pengajuan' => $pengajuanKK,
                'penilaian' => $penilaian
            ]);
        }

        // Jika tidak ditemukan di semua model
        return redirect()->back()->with('error', 'Kode Pengajuan tidak
        ditemukan.');
```

```

    }

    public function daftar_pengaduan()
    {
        // Mengambil pengaduan dengan pagination 5 per halaman
        $pengaduan = Pengaduan::latest()->paginate(5);

        return view('pengaduan', compact('pengaduan'));
    }

    public function tambah_pengaduan(Request $request){
        $request->validate([
            'judul' => 'required|string|max:255',
            'isi_pengaduan' => 'required',
        ]);

        Pengaduan::create([
            'user_id' => Auth::id(), // Ambil ID user yang login
            'judul' => $request->judul,
            'isi_pengaduan' => $request->isi_pengaduan,
        ]);

        session()->flash('success', 'Pengaduan Berhasil
Dikirim.');
```

```

        return redirect()->route('pengaduan-index');

        // return redirect()->route('pengaduan-index')-
>with('success', 'Pengaduan berhasil dikirim.');
```

```

    }

```

Lampiran 5 Foto Dokumentasi

