# LAPORAN PROJECT MEMBUAT APLIKASI PENDATAAN PEGAWAI

Disusun untuk memenuhi syarat mengikuti ujian kompetensi kejuruan pada

Kompetensi Keahlian Rekayasa Perangkat Lunak

**Disusun Oleh:**

**NAMA : RYAN FAHLEVI JAS**

**NISN : 0079741851**

**REKAYASA PERANGKAT LUNAK**

**SMK NEGERI 1 KARANG BARU**

**TAHUN 2024**

# LEMBAR PERSETUJUAN

**APLIKASI PENDATAAN PEGAWAI**

**Disusun Oleh :**

**NAMA : RYAN FAHLEVI JAS**

**NISN : 0079741851**

Telah disetujui oleh Guru Pembimbing pada tanggal 29 Agustus 2024

Untuk dipertanggungjawabkan di depan Penguji

Kompetensi Keahlian Rekayasa Perangkat Lunak

SMK Negeri 1 Karang baru

Mengetahui,

|  |  |
| --- | --- |
| Kepala Kompetensi Keahlian Rekayasa Perangkat Lunak | Guru Pembimbing |
| **Safrizal, S.ST**  **NIP. 19870530 201003 1 001** | **Ahmadi Muslim, MP**  **NIP.** **19860505 201403 1 003** |

# DAFTAR ISI

[LEMBAR PERSETUJUAN i](#_Toc188173773)

[DAFTAR ISI ii](#_Toc188173774)

[BAB I ANALISIS KEBUTUHAN APLIKASI PENDATAAN PEGAWAI 1](#_Toc188173775)

[**A.** Kuesioner Analisis Kebutuhan Aplikasi **1**](#_Toc188173776)

[Bagian I: Data Responden 1](#_Toc188173777)

[Bagian 2: Penggunaan Teknologi 1](#_Toc188173778)

[Bagian 3: Proses Pendataan Saat Ini 2](#_Toc188173779)

[Bagian 4: Kebutuhan Fitur Aplikasi 2](#_Toc188173780)

[Bagian 5: Saran dan Masukan 3](#_Toc188173781)

[B. Daftar Pertanyaan Wawancara 3](#_Toc188173782)

[Bagian 1: Pengalaman dan Kebutuhan Umum 3](#_Toc188173783)

[Bagian 2: Fitur dan Fungsi yang Diinginkan 3](#_Toc188173784)

[Bagian 3: Antarmuka Pengguna dan Pengalaman 3](#_Toc188173785)

[Bagian 4: Masukan dan Saran 3](#_Toc188173786)

[C. Dokumen Observasi 4](#_Toc188173787)

[D. Rekapitulasi Observasi 6](#_Toc188173788)

[E. Hasil Analisis Aplikasi 8](#_Toc188173789)

[F. Matriks Kebutuhan Aplikasi 9](#_Toc188173790)

[G. Jadwal Pembuatan Aplikasi 10](#_Toc188173791)

[BAB II Desain Aplikasi 11](#_Toc188173792)

[A. Low Fidelity Prototype (Wireframe) 11](#_Toc188173793)

[B. High Fidelity Prototype (Desain Figma jika ada) 15](#_Toc188173794)

[C. Diagram Aktifitas 18](#_Toc188173795)

[D. Entitas Relationship Diagram/ERT 19](#_Toc188173796)

[E. Hasil Perancangan Tabel Basis Data 21](#_Toc188173797)

[BAB III DESAIN APLIKASI 22](#_Toc188173798)

[A. Potongan Kode Program Route 22](#_Toc188173799)

[B. Potongan Kode Program Migrasi Database 23](#_Toc188173800)

[C. Potongan Kode Program Model 25](#_Toc188173801)

[D. Potongan Kode Program Controller 27](#_Toc188173802)

[E. Potongan Kode Program View 36](#_Toc188173803)

[F. Pengembangan (Github) 51](#_Toc188173804)

[BAB IV PENGUJIAN APLIKASI 52](#_Toc188173805)

[A. Instrumen Validasi Pengujian Aspek Fungsionalitas Aplikasi 52](#_Toc188173806)

[B. Hasil Pengujian Aspek Fungsionalitas Aplikasi Pembimbing 53](#_Toc188173807)

[BAB V PEMELIHARAAN APLIKASI 54](#_Toc188173808)

[A. Saran Pengembangan Aplikasi dari Penguji 54](#_Toc188173809)

[B. Kesimpulan 54](#_Toc188173810)

[LAMPIRAN 55](#_Toc188173811)

[FORMULIR BIMBINGAN LAPORAN PENGEMBANGAN APLIKASI 56](#_Toc188173812)

# BAB I ANALISIS KEBUTUHAN APLIKASI PENDATAAN PEGAWAI

## Kuesioner Analisis Kebutuhan Aplikasi

### Bagian I: Data Responden

Nama:

Jabatan:

Divisi/Bidang:

Lama bekerja di instansi:

* < 1 tahun
* 1-3 tahun
* 3-5 tahun
* > 5 tahun

### Bagian 2: Penggunaan Teknologi

* Seberapa sering Anda menggunakan sistem aplikasi untuk pengelolaan data pegawai saat ini?
* Mingguan
* Bulanan
* Tahunan
* Tidak pernah
* Seberapa nyaman Anda dalam menggunakan aplikasi atau perangkat lunak di tempat kerja?
* Sangat nyaman
* Nyaman
* Cukup nyaman
* Tidak nyaman

### Bagian 3: Proses Pendataan Saat Ini

* Bagaimana cara pendataan pegawai yang dilakukan saat ini?
* Manual (menggunakan kertas)
* Spreadsheet (Excel, Google Sheets)
* Sistem/aplikasi lain
* Seberapa sering Anda melakukan proses pendataan data pegawai?
* Mingguan
* Bulanan
* Tahunan
* Tidak pernah
* Siapa saja yang terlibat dalam proses pendataan pegawai di unit kerja Anda?
* Staf khusus
* Atasan langsung
* Pegawai terkait
* Apa saja kesulitan yang Anda hadapi dalam proses pendataan pegawai saat ini?
* Data tidak akurat
* Proses memakan waktu lama
* Proses pendataan masih manual
* Tidak ada

### Bagian 4: Kebutuhan Fitur Aplikasi

* Menurut Anda, fitur apa saja yang seharusnya ada dalam aplikasi pendataan pegawai? (bisa memilih lebih dari satu)
* Input data pegawai
* Delete data pegawai
* Update data pegawai
* Total pegawai
* Laporan data pegawai

### Bagian 5: Saran dan Masukan

* Apakah Anda memiliki saran atau masukan lainnya terkait pengembangan aplikasi pendataan pegawai ini?

## Daftar Pertanyaan Wawancara

### Bagian 1: Pengalaman dan Kebutuhan Umum

* Apakah Anda sudah pernah menggunakan aplikasi pendataan pegawai sebelumnya?
* Bagaimana proses pendataan pegawai saat ini di tempat Anda bekerja?
* Apa tantangan terbesar yang Anda hadapi dalam mengelola data pegawai saat ini?
* Menurut Anda, seberapa penting memiliki aplikasi khusus untuk pendataan pegawai?

### Bagian 2: Fitur dan Fungsi yang Diinginkan

* Apa saja fitur yang Anda anggap penting dalam aplikasi pendataan pegawai?
* Apakah ada fitur khusus yang menurut Anda akan mempermudah penggunaan aplikasi ini?

### Bagian 3: Antarmuka Pengguna dan Pengalaman

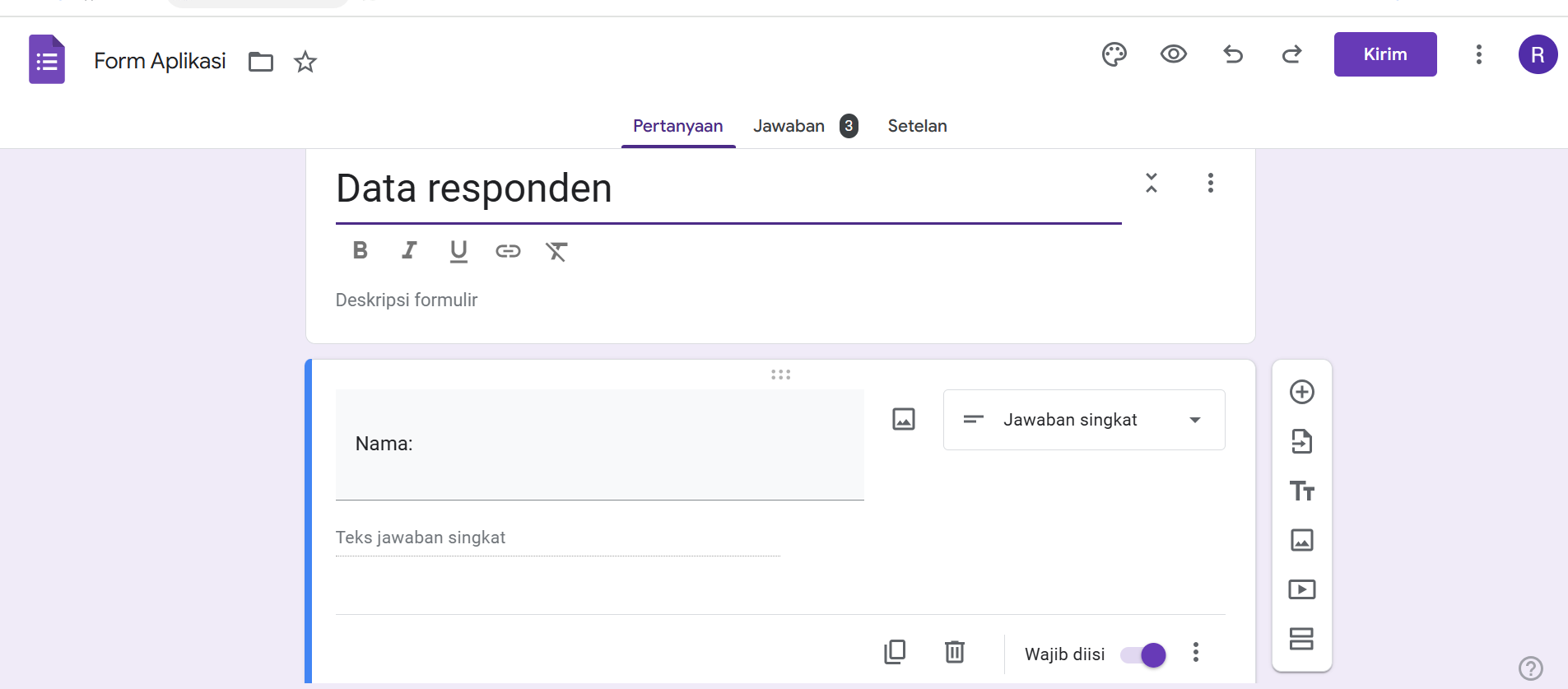
* Seberapa penting bagi Anda bahwa aplikasi ini mudah digunakan (userfriendly)?

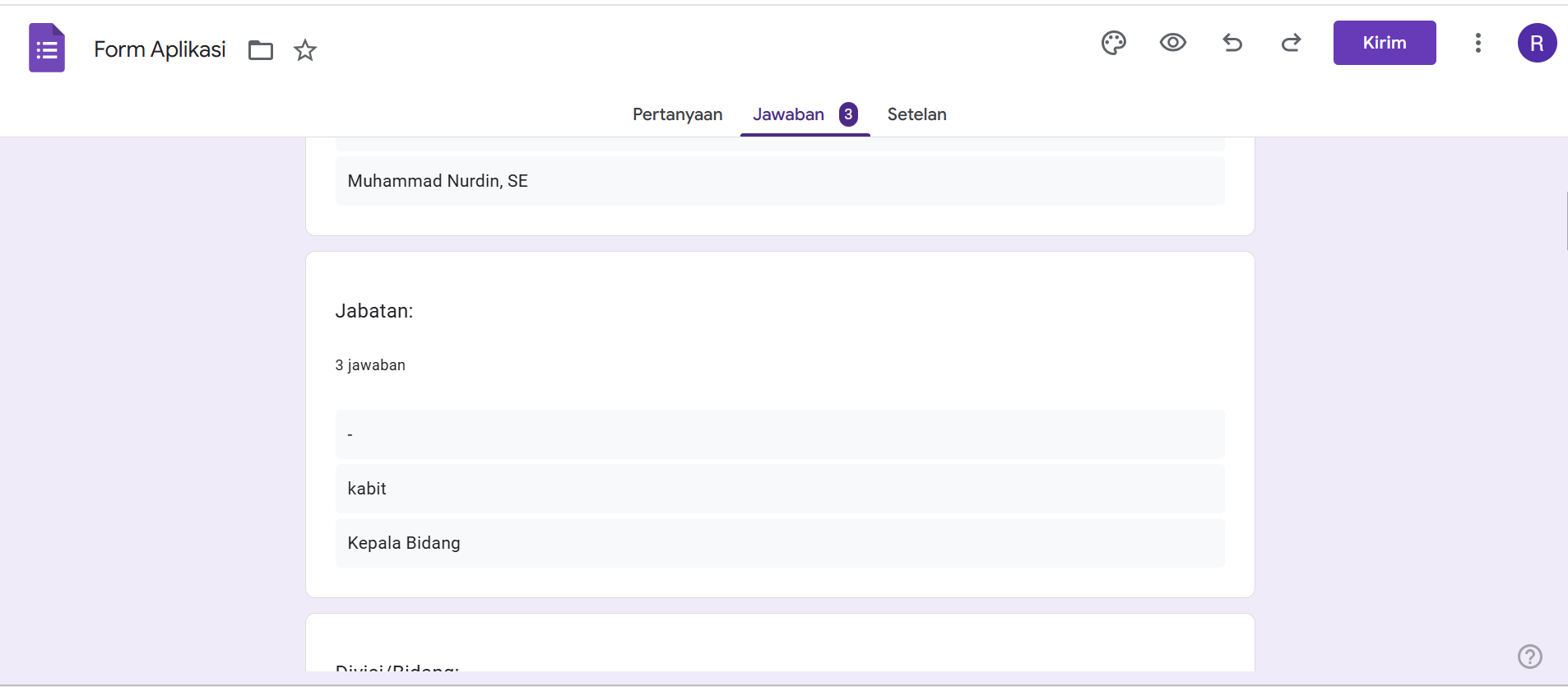
### Bagian 4: Masukan dan Saran

* Apa saran atau masukan tambahan yang Anda ingin berikan untuk pengembangan aplikasi ini?

## Dokumen Observasi

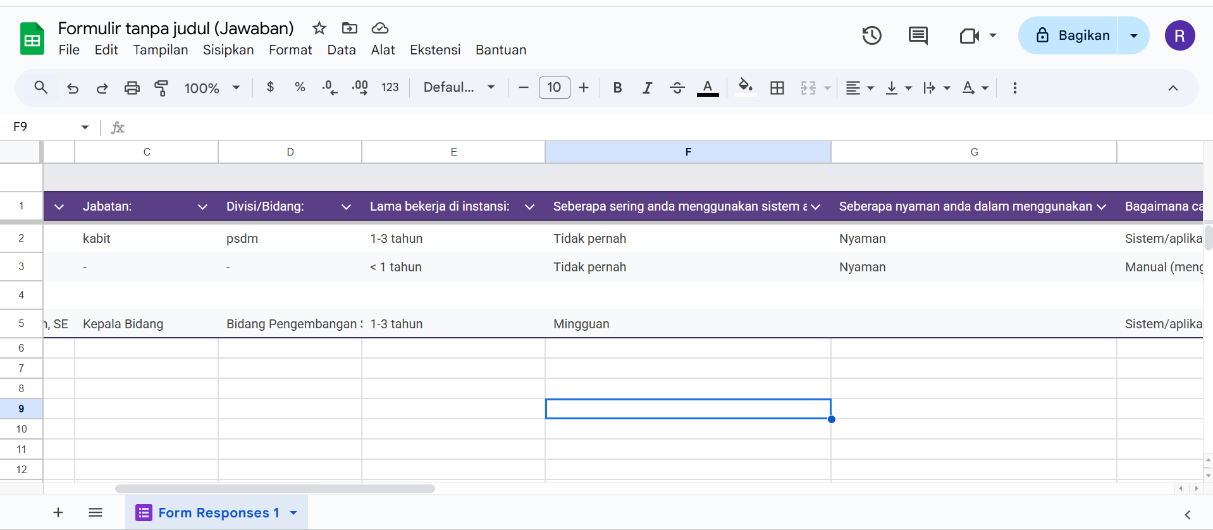
1. **Formulir dengan google form**

****

****

1. **Dokumentasi Pengisian Formulir**

****

1. **Hasil Pengisian Formulir**

## Rekapitulasi Observasi

|  |  |  |  |
| --- | --- | --- | --- |
| Daftar Pertanyaan | Jawaban | | |
| Responden A (Pemilik) | Responden B (Admin) | Responden C (Pelanggan) |
| **A. Kuesioner Analisis Kebutuhan Aplikasi** |  |  |  |
| **Bagian 2: Penggunaan Teknologi** |  |  |  |
| Seberapa sering anda menggunkan sistem aplikasi untuk pengolaan data pegawai saat ini? | Tidak pernah | Mingguan | Tidak pernah |
| Seberapa nyaman anda dalam menggunakan aplikasi atau perangkat lunak di tempat kerja? | Nyaman | Nyaman | Nyaman |
| **Bagian 3: Proses Pendataan Saat Ini** |  |  |  |
| Bagaimana cara pendataan pegawai yang di lakukan saat ini? | Excel | Aplikasi lain | Excel |
| Seberapa sering anda melakukan proses pendataan data pegawai? | Tidak pernah | Mingguan | Tidak pernah |
| Siapa saja yang terlibat dalam proses pendataan pegawai di unit kerja anda? | Pegawai terkait | Pegawai terkait | Pegawai terkait |
| **Bagian 4: Kebutuhan Fitur Aplikasi** |  |  |  |
| Menurut anda fitur apa saja yang seharusnya ada dalam aplikasi pendataan pegawai? (bisa memilih lebih dari satu) | Input data pegawai, Delete data pegawai, Update data pegawai, total pegawai, Laporan data pegawai | Input data pegawai, Delete data pegawai, Update data pegawai, total pegawai, Laporan data pegawai | Input data pegawai, Delete data pegawai, Update data pegawai, total pegawai, Laporan data pegawai |
| **Bagian 5: Saran dan Masukan** |  |  |  |
| Apakah anda memiliki saran atau masukan lainnya terkait pengembangan aplikasi pendataan pegawai ini? | Mudah dalam pencarian data pegawai | Mudah dalam proses pendataan | Tampilan yang menarik |
|  |  |  |  |
| **B. Daftar pertanyaan Wawancara** |  |  |  |
| **Bagian 1: Pengalaman dan Kebutuhan Umum** |  |  |  |
| Apakah anda sudah pernah menggunakan aplikasi pendataan pegawai sebelumnya? | tidak | iya | tidak |
| Bagaimana proses pendataan pegawai saat ini di tempat anda bekerja? | Excel | aplikasi lain | Excel |
| Bagaimana tantangan terbesar yang anda hadapi dalam mengelola data pegawai saat ini? | sulit mencari data pegawai | sulit dalam mencari data pegawai | sulit dalam mendata pegawai |
| Menurut anda, seberapa penting memiliki aplikasi khusus untuk pendataan pegawai? | sangat penting | sangat penting | sangat penting |
| **Bagian 2: Fitur dan Fungsi yang diinginkan** |  |  |  |
| Apa saja fitur yang anda anggap penting dalam aplikasi pendataan pegawai? | tambah, edit, hapus, laporan | tambah, edit, hapus,show, laporan | tambah, edit, hapus |
| Apakah ada fitur khusus yang menurut anda akan mempermudah penggunaan aplikasi ini? | show data | laporan data | tambah data |
| **Bagian 3: Antarmuka Pengguna dan pengalaman** |  |  |  |
| Seberapa penting bagi anda bahwa aplikasi ini untuk digunakan? | sangat penting | sangat penting | sangat penting |
| **Bagian 4: Masukkan dan Saran** |  |  |  |
| Apakah saran atau masukan tambahan yang anda ingin berikan untuk pengembangan aplikasi ini? | Mudah dalam pencarian data pegawai | Mudah dalam proses pendataan | Tampilan yang menarik |

## Hasil Analisis Aplikasi

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| fitur | pendukung | urgensi/tingkat kepentingan | | | | | |
| create | read(tampil) | update | delete | printout | pencarian |
| login | email |  | perlu |  |  |  |  |
| password |  | perlu |  |  |  |  |
| dashboard | admin |  | perlu |  |  |  |  |
| pegawai | perlu | perlu | perlu | perlu |  | perlu |
| admin | namaadmin | perlu |  | perlu | perlu |  | perlu |
| email | perlu |  | perlu | perlu |  | perlu |
| password | perlu |  | perlu | perlu |  | perlu |
| pegawai | namapegawai | perlu |  | perlu | perlu |  | perlu |
| nip | perlu | perlu | perlu | perlu |  | perlu |
| jabatan | perlu | perlu | perlu | perlu |  | perlu |
| jeniskelamin | perlu | perlu | perlu | perlu |  | perlu |
| ttl | perlu | perlu | perlu | perlu |  | perlu |
| usia | perlu | perlu | perlu | perlu |  | perlu |
| tmt | perlu | perlu | perlu | perlu |  | perlu |
| masakerja | perlu | perlu | perlu | perlu |  | perlu |
| statuskeluarga | perlu | perlu | perlu | perlu |  | perlu |
| golongandarah | perlu | perlu | perlu | perlu |  | perlu |
| agama | perlu | perlu | perlu | perlu |  | perlu |
| unitkerja | perlu | perlu | perlu | perlu |  | perlu |
| alamat | perlu | perlu | perlu | perlu |  | perlu |
| pegawai->jabatan | namajabatan | perlu |  | perlu | perlu |  | perlu |
| pegawai->unitkerja | namaunitkerja | perlu |  | perlu | perlu |  | perlu |
| Laporan dan cetak | berdasarkan tanggal |  | perlu |  |  | perlu |  |

## Matriks Kebutuhan Aplikasi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Fitur/Fungsionalitas** | **Deskripsi Singkat** | **Prioritas** | **Tingkat Kepentingan** | **Keterangan** |
| 1 | Pengelolaan Data Pegawai | Menginput, mengedit, dan menghapus data pegawai. | Tinggi | Sangat Penting | Fitur inti untuk pengelolaan data seluruh pegawai. Dengan hak akses pengguna tertentu. |
| 2 | Profil Pegawai | Menampilkan profil lengkap pegawai, termasuk nama, NIP, jabatan saat ini, unit kerja, dan informasi pribadi lainnya. | Tinggi | Sangat Penting | User/admin dapat mengakses data ini untuk melihat informasi terkini, juga dapat mengedit. |
| 3 | Laporan Data Pegawai | Membuat laporan berdasarkan seluruh data pegawai. | Sedang | Penting | Laporan dicetak dengan format Pdf. |
| 4 | Login | Melindungi data, membatasi hak akses. | Tinggi | Sangat Penting | Keamanan harus diterapkan untuk memastikan data pegawai tetap rahasia dan hanya diakses oleh pihak yang berwenang. |

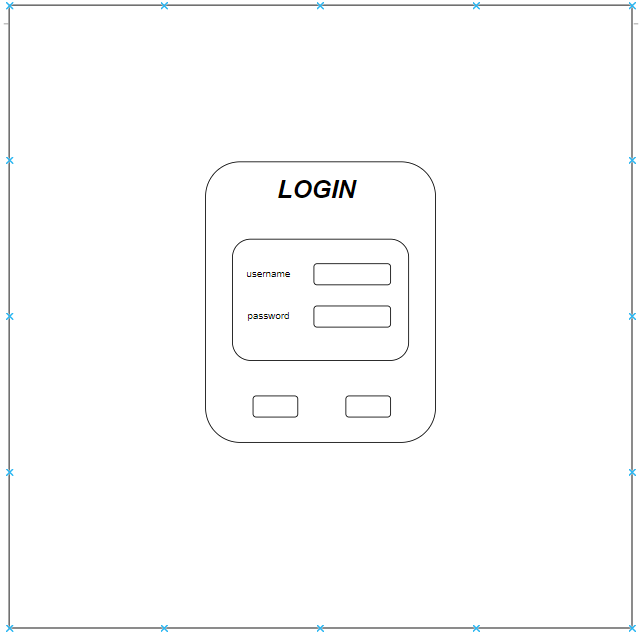
## Jadwal Pembuatan Aplikasi

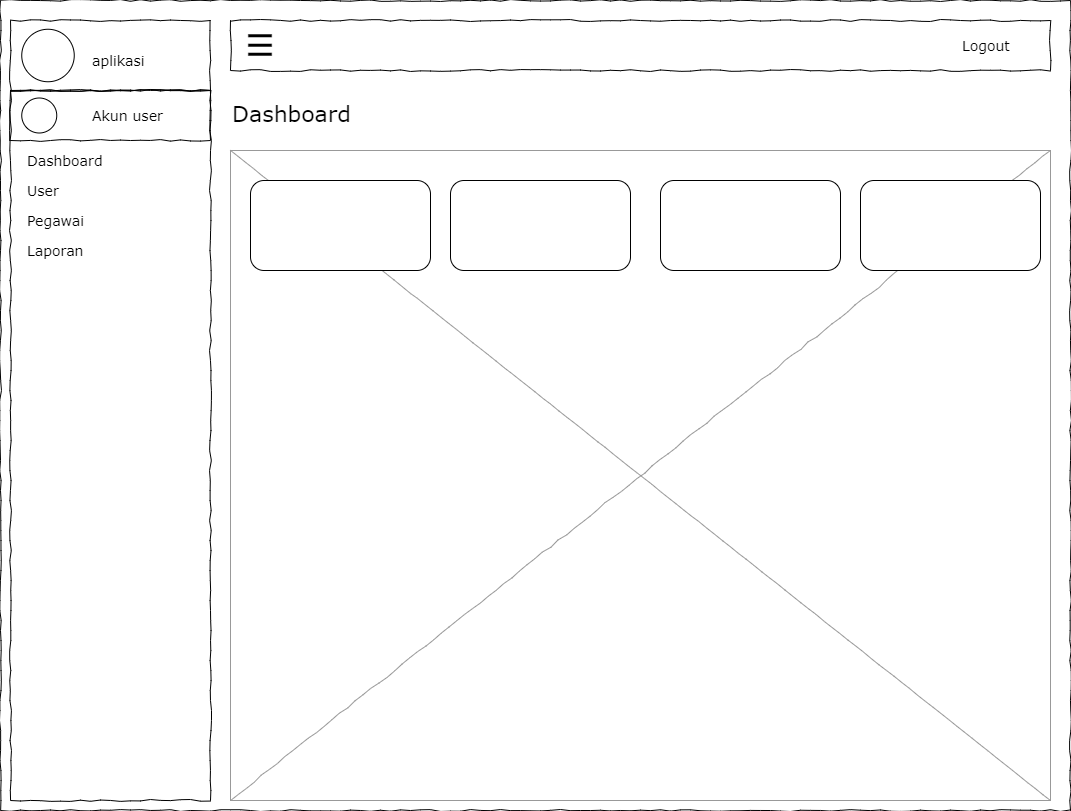
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Tahapan | Kegiatan |  | | | | | | | | | | | | | | | | |
| Agustus | | September | | | | Oktober | | | | | November | | | | Desember | |
| 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 1 | 2 |
| 1 | Perencanaan dan Analisis | Wawancara, Observasi dan Kuisioner |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Analisis Kebutuhan, Matriks Kebutuhan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Desain | Desain UI/UX |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Prototyping |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Desain Database |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Pengembangan (Coding) | User |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Jabatan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Unit Kerja |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pegawai |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Login |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Pengujian | Pengujian Fungsionalitas |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Peluncuran dan Pemeliharaan | Peluncuran, Monitoring, Pemeliharaan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# BAB II Desain Aplikasi

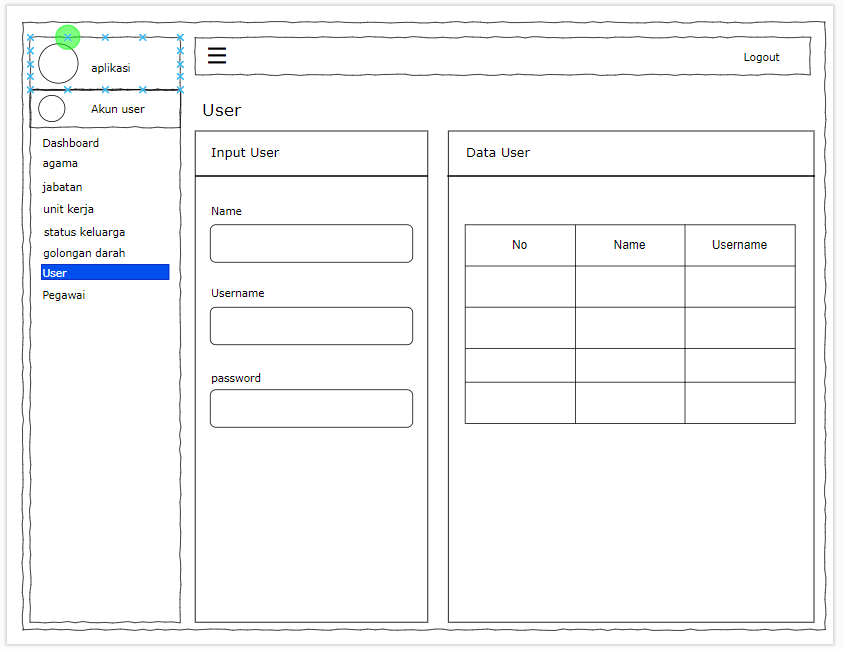
## Low Fidelity Prototype (Wireframe)

* 1. **Low Fidelity Prototype Login**

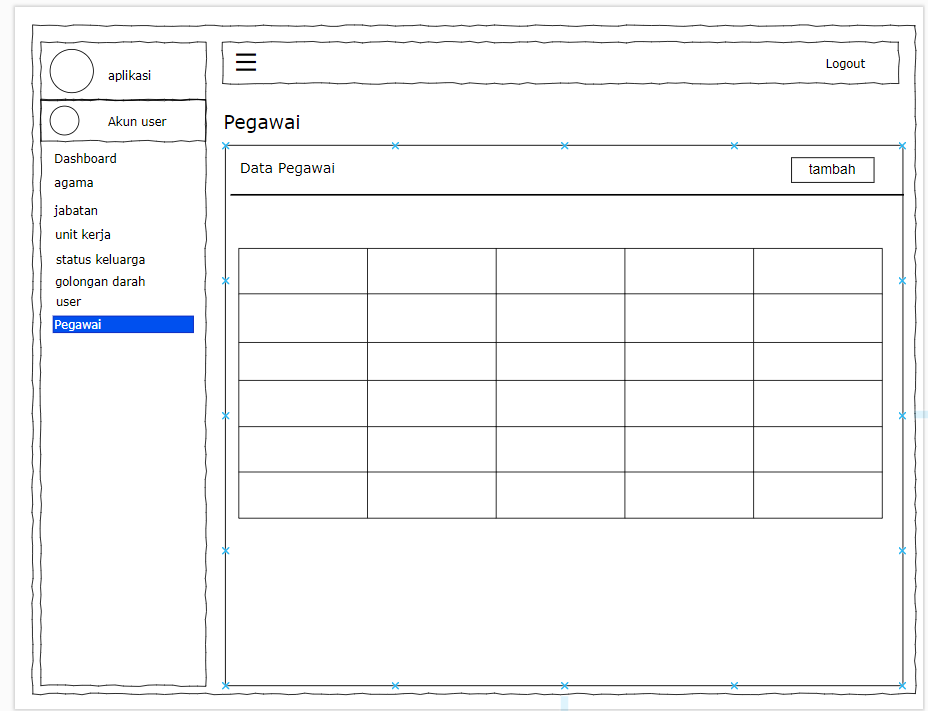


* 1. **Low Fidelity Prototype Dashboard**

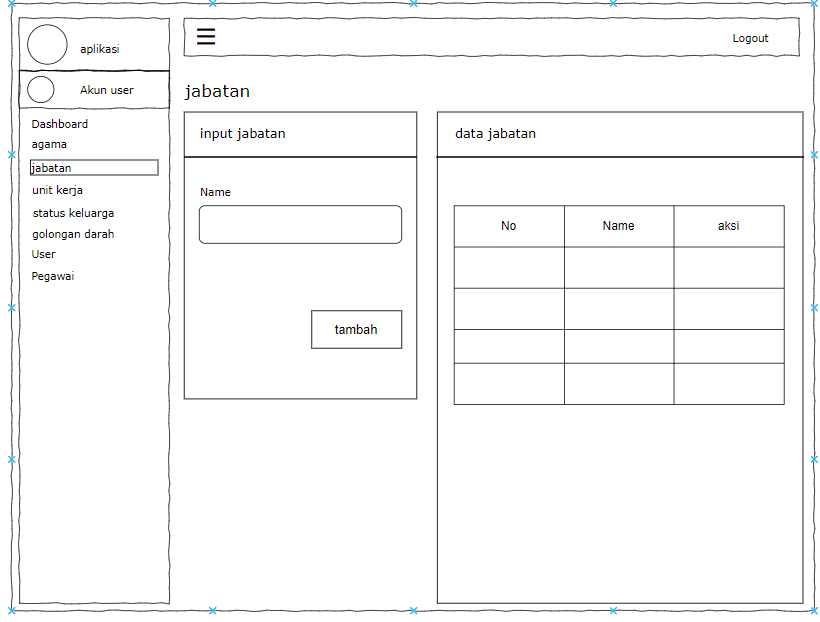
* 1. **Low Fidelity Prototype User**

****

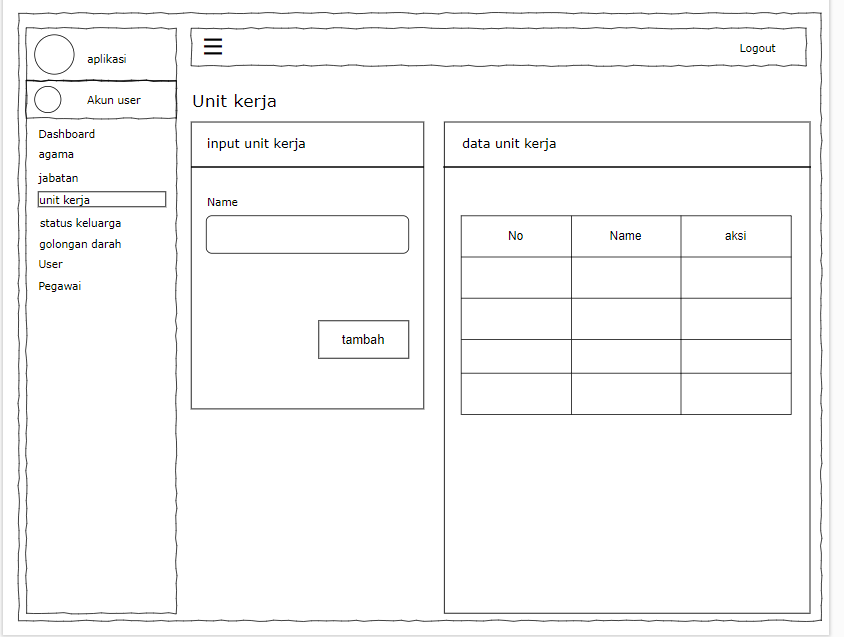
* 1. **Low Fidelity Prototype Pegawai**

****

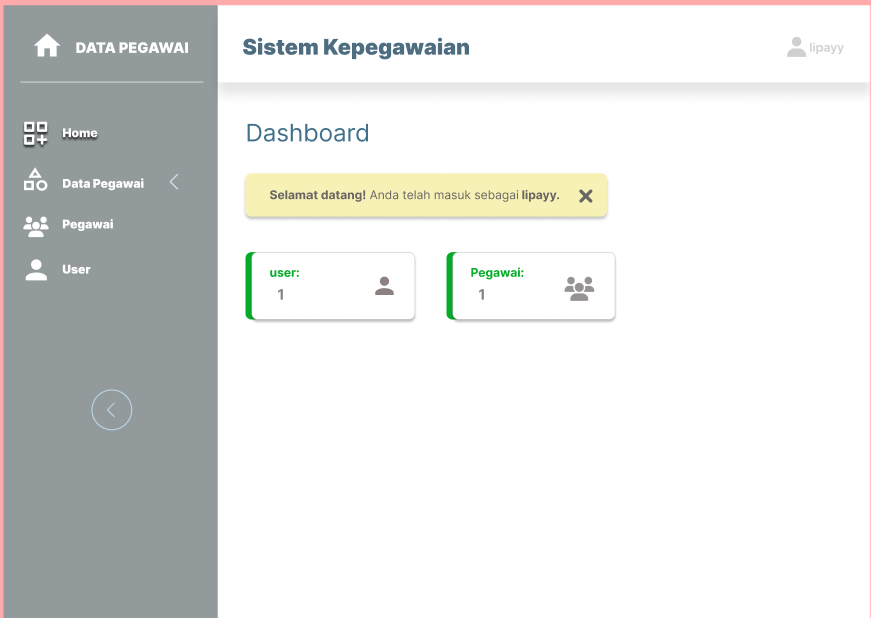
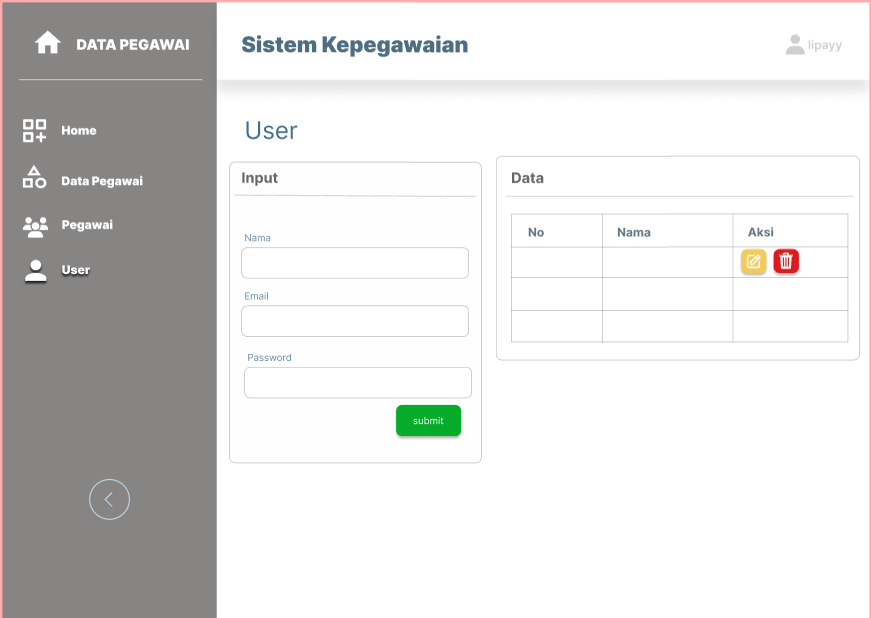
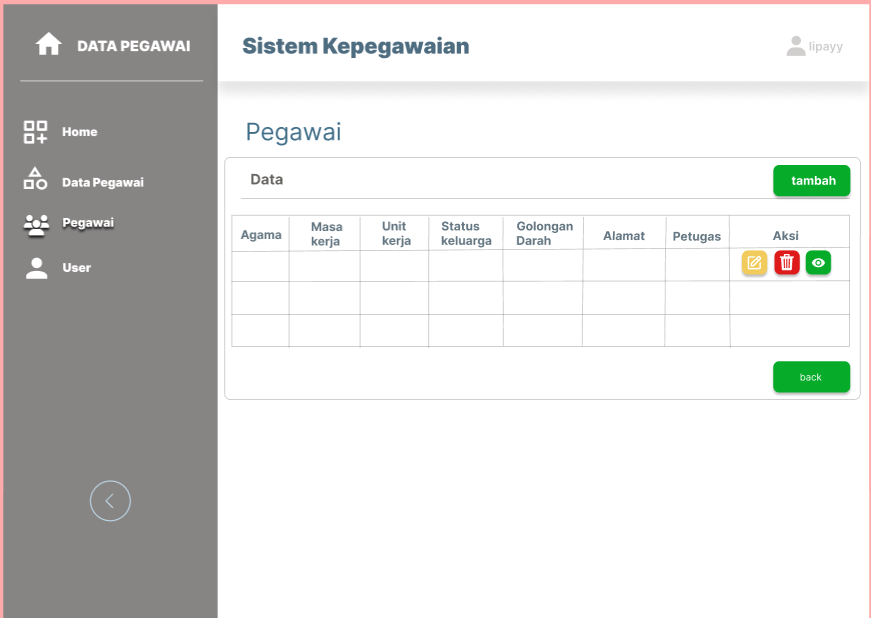
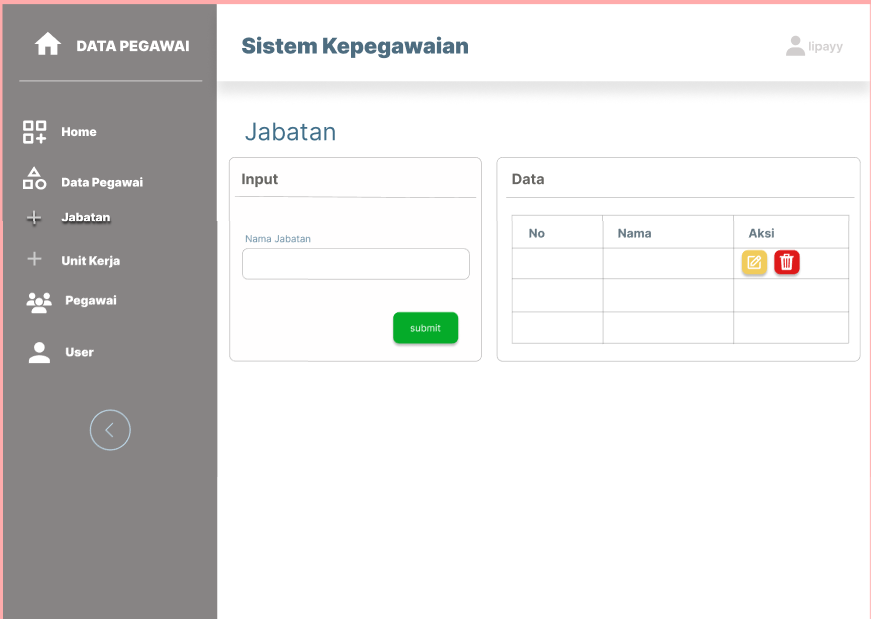
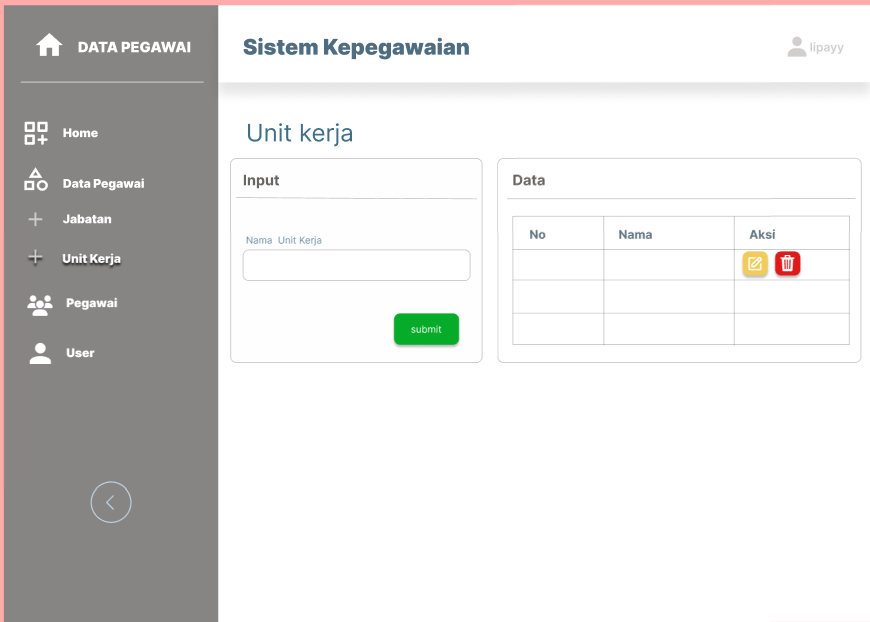
* 1. **Low Fidelity Prototype Jabatan**

****

* 1. **Low Fidelity Prototype Unit Kerja**

****

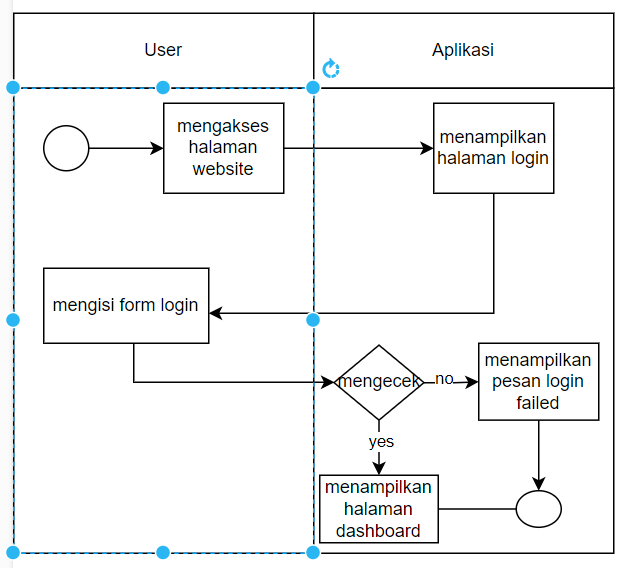
## High Fidelity Prototype (Desain Figma jika ada)

1. **High Fidelity Prototype Login**
2. **High Fidelity Prototype Dashboard**
3. **High Fidelity Prototype User**
4. **High Fidelity Prototype Pegawai**
5. **High Fidelity Prototype Jabatan**
6. **High Fidelity Prototype Unit Kerja**

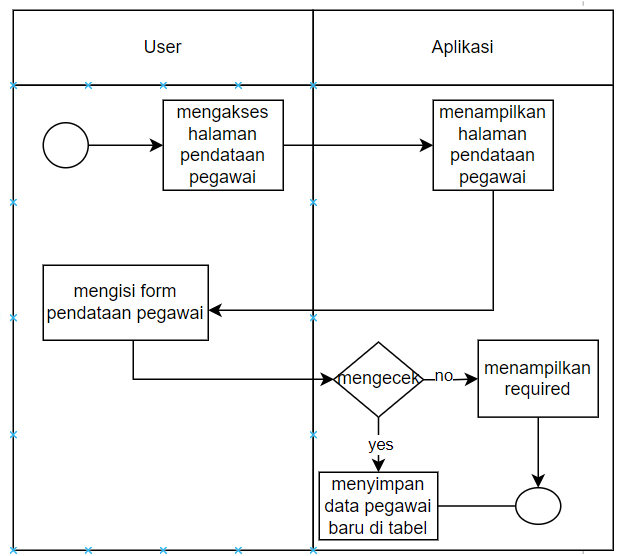
<https://bit.ly/PrototypeFigmaa> (Link Prototype Figma)

## Diagram Aktifitas

1. Aktifitas diagram login

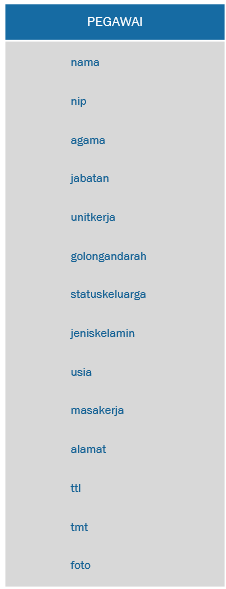
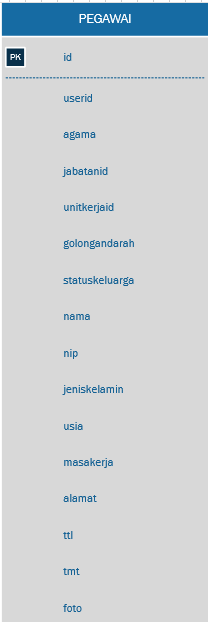


1. Aktifitas diagram Pendataan pegawai



## Entitas Relationship Diagram/ERT

**Unf 1nf**

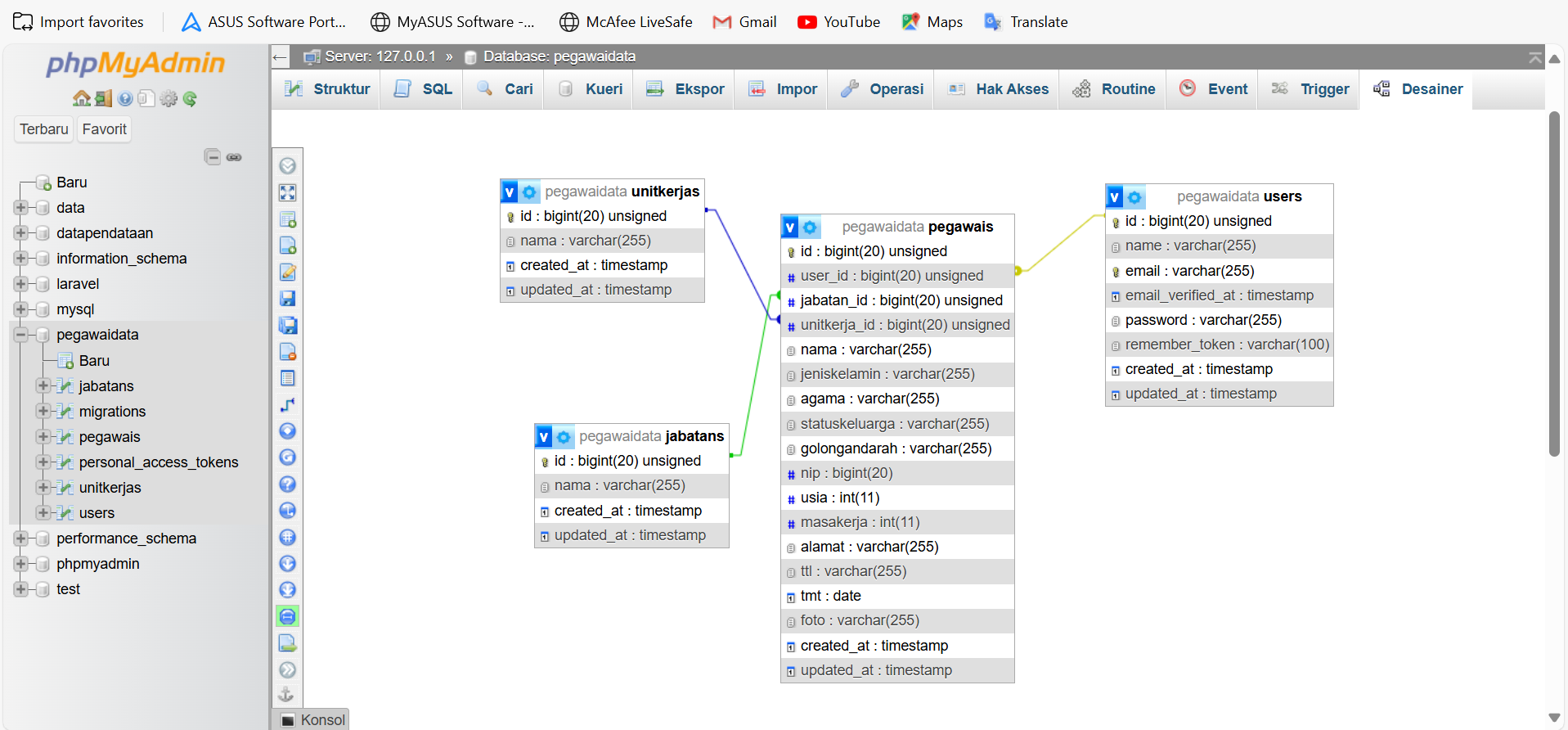


**2nf**



# 

## Hasil Perancangan Tabel Basis Data



# BAB III DESAIN APLIKASI

## Potongan Kode Program Route

|  |
| --- |
| **Web.php** |
| <?php  use Illuminate\Support\Facades\Route;  use App\Http\Controllers\UserController;  use App\Http\Controllers\LoginController;  use App\Http\Controllers\JabatanController;  use App\Http\Controllers\LaporanController;  use App\Http\Controllers\PegawaiController;  use App\Http\Controllers\DashboardController;  use App\Http\Controllers\UnitkerjaController;  );  Route::get('/laporan/pdf', [LaporanController::class, 'exportPdf'])->name('laporan.pdf');  Route::resource('/laporan', LaporanController::class);  Route::resource('pegawai', PegawaiController::class)->middleware('auth');  Route::get('/pegawai/pdf', [PegawaiController::class, 'exportPdf'])->name('pegawai.pdf');  Route::resource('jabatan', JabatanController::class)->middleware('auth');  Route::resource('unitkerja', UnitkerjaController::class)->middleware('auth');  Route::resource('user', UserController::class)->except('destroy', 'create', 'show', 'update', 'edit');  Route::get('login', [LoginController::class, 'loginView'])->name('login');  Route::post('login', [LoginController::class, 'authenticate']);  Route::post('logout', [LoginController::class, 'logout'])->name('auth.logout')->middleware('auth');  Route::get('/', [DashboardController::class, 'index'])->middleware('auth');  Route::get('/chart-data', [App\Http\Controllers\DashboardController::class, 'getChartData']); |

## Potongan Kode Program Migrasi Database

|  |  |  |
| --- | --- | --- |
| **Migrasi AdminMigrasi Admin** | | **Migrasi Pegawai** |
| <?php  use Illuminate\Database\Migrations\Migration;  use Illuminate\Database\Schema\Blueprint;  use Illuminate\Support\Facades\Schema;  return new class extends Migration  {  /\*\*  \* Run the migrations.  \*/  public function up(): void  {  Schema::create('users', function (Blueprint $table) {  $table->id();  $table->string('name');  $table->string('email')->unique();  $table->timestamp('email\_verified\_at')->nullable();  $table->string('password');  $table->rememberToken();  $table->timestamps();  });  }  /\*\*  \* Reverse the migrations.  \*/  public function down(): void  {  Schema::dropIfExists('users');  }  }; | | <?php  use Illuminate\Database\Migrations\Migration;  use Illuminate\Database\Schema\Blueprint;  use Illuminate\Support\Facades\Schema;  return new class extends Migration  {  public function up(): void  {  Schema::create('pegawais', function (Blueprint $table) {  $table->id();  $table->bigInteger('user\_id')->unsigned();  $table->bigInteger('jabatan\_id')->unsigned();  $table->bigInteger('unitkerja\_id')->unsigned();  $table->string('nama');  $table->string('jeniskelamin');  $table->string('agama');  $table->string('statuskeluarga');  $table->string('golongandarah');  $table->bigInteger('nip');  $table->integer('usia')->nullable();  $table->integer('masakerja');  $table->string('alamat');  $table->string('ttl');  $table->date('tmt');  $table->string('foto')->nullable(); // Kolom untuk menyimpan path foto  $table->timestamps();  });  }  public function down(): void  {  Schema::dropIfExists('pegawais');  }  }; |
| **Migrasi Jabatan** | **Migrasi Unitkerja** | |
| <?php  use Illuminate\Database\Migrations\Migration;  use Illuminate\Database\Schema\Blueprint;  use Illuminate\Support\Facades\Schema;  return new class extends Migration  {  /\*\*  \* Run the migrations.  \*/  public function up(): void  {  Schema::create('jabatans', function (Blueprint $table) {  $table->id();  $table->string('nama');  $table->timestamps();  });  }  /\*\*  \* Reverse the migrations.  \*/  public function down(): void  {  Schema::dropIfExists('jabatans');  }  }; | <?php  use Illuminate\Database\Migrations\Migration;  use Illuminate\Database\Schema\Blueprint;  use Illuminate\Support\Facades\Schema;  return new class extends Migration  {  /\*\*  \* Run the migrations.  \*/  public function up(): void  {  Schema::create('unitkerjas', function (Blueprint $table) {  $table->id();  $table->string('nama');  $table->timestamps();  });  }  /\*\*  \* Reverse the migrations.  \*/  public function down(): void  {  Schema::dropIfExists('unitkerjas');  }  }; | |

## Potongan Kode Program Model

|  |  |
| --- | --- |
| **Model Admin** | **Model Pegawai** |
| <?php  namespace App\Models;  // use Illuminate\Contracts\Auth\MustVerifyEmail;  use Illuminate\Database\Eloquent\Factories\HasFactory;  use Illuminate\Database\Eloquent\Relations\HasMany;  use Illuminate\Foundation\Auth\User as Authenticatable;  use Illuminate\Notifications\Notifiable;  use Laravel\Sanctum\HasApiTokens;  use Illuminate\Database\Relations\BelongsTo;  class User extends Authenticatable  {  use HasApiTokens, HasFactory, Notifiable;  protected $fillable = [  'name',  'email',  'password',  ];  protected $hidden = [  'password',  'remember\_token',  ];  protected $casts = [  'email\_verified\_at' => 'datetime',  'password' => 'hashed',  ];  public function user():HasMany  {  return $this->hasMany(User::class);  }  } | <?php  namespace App\Models;  use Illuminate\Database\Eloquent\Factories\HasFactory;  use Illuminate\Database\Eloquent\Model;  use Illuminate\Database\Eloquent\Relations\BelongsTo;  use Illuminate\Database\Relations\HasMany;  class Pegawai extends Model  {  use HasFactory;  protected $fillable=['user\_id','nama','nip','jeniskelamin','usia','masakerja','statuskeluarga',  'golongandarah','agama','unitkerja\_id','jabatan\_id','ttl','tmt','alamat','foto'];  public function user():BelongsTo  {  return $this->belongsTo(User::class);  }      public function unitkerja():BelongsTo  {  return $this->belongsTo(Unitkerja::class);  }  public function jabatan():BelongsTo  {  return $this->belongsTo(Jabatan::class);  }    } |

|  |  |
| --- | --- |
| **Model Jabatan** | **Model Unitkerja** |
| <?php  namespace App\Models;  use Illuminate\Database\Eloquent\Factories\HasFactory;  use Illuminate\Database\Eloquent\Model;  use Illuminate\Database\Eloquent\Relations\HasMany;  class Jabatan extends Model  {  use HasFactory;  protected $fillable=['nama'];  public function pegawai():HasMany  {  return $this->hasMany(Pegawai::class);  }  } | <?php  namespace App\Models;  use Illuminate\Database\Eloquent\Factories\HasFactory;  use Illuminate\Database\Eloquent\Model;  use Illuminate\Database\Eloquent\Relations\HasMany;  class Unitkerja extends Model  {  use HasFactory;  protected $fillable=['nama'];  public function pegawai():HasMany  {  return $this->hasMany(Pegawai::class);  } |

## Potongan Kode Program Controller

1. **AdminController**

|  |  |
| --- | --- |
| <?php  namespace App\Http\Controllers;  use App\Models\User;  use Illuminate\Http\Request;  use Illuminate\Http\RedirectResponse;  use Illuminate\Support\Facades\Hash;  use Illuminate\View\View;  class UserController extends Controller  {  // Fungsi untuk menampilkan daftar user  public function index(): View  {  // Mengambil semua data user dari database dan mengirimkannya ke view 'user.index'  return view('user.index', [  "title" => "Data user",  "data" => User::all()  ]);  }  // Fungsi untuk menyimpan user baru  public function store(Request $request): RedirectResponse  {  // Validasi input dari form tambah user  $request->validate([  "name" => "required",  "email" => "required",  "password" => "required"  ]);    // Melakukan hashing pada password sebelum disimpan ke database  $password = Hash::make($request->password); | // Menggabungkan hasil hash password ke dalam request yang akan disimpan  $request->merge([  "password" => $password  ]);  // Menyimpan data user yang baru ke database  User::create($request->all());  // Redirect ke halaman daftar user dengan pesan sukses  return redirect()->route('user.index')->with('success', 'Data User Berhasil Ditambahkan');  }  } |

1. **PegawaiController**

|  |  |  |
| --- | --- | --- |
| <?php  namespace App\Http\Controllers;  use App\Models\Pegawai;  use App\Models\User;  use App\Models\Jabatan;  use App\Models\Unitkerja;  use Illuminate\View\View;  use Illuminate\Http\RedirectResponse;  use Illuminate\Support\Facades\Storage;  use Illuminate\Http\Request;  use Barryvdh\DomPDF\Facade\Pdf;  use Illuminate\Support\Facades\Crypt;  class PegawaiController extends Controller  {  // public function laporan(Request $request)  // {  // // Validasi input  // $request->validate([  // 'start\_date' => 'nullable|date',  // 'end\_date' => 'nullable|date|after\_or\_equal:start\_date',  // 'search' => 'nullable|string',  // ]);    // // Inisialisasi query builder dari model Pegawai  // $query = Pegawai::query();    // // Filter berdasarkan pencarian NIP  // if ($request->filled('search')) {  // $query->where('nip', 'like', '%' . $request->search . '%');  // }    // // Filter berdasarkan tanggal  // if ($request->filled('start\_date') && $request->filled('end\_date')) {  // $query->whereBetween('created\_at', [$request->start\_date, $request->end\_date]);  // }    // // Paginasi hasil data  // $pegawai = $query->paginate(10);    // // Kirim data ke view  // return view('pegawai.laporan', [  // 'data' => $pegawai,  // 'pegawai' => $pegawai,  // 'title' => 'Pegawai',  // 'search' => $request->search,  // 'start\_date' => $request->start\_date,  // 'end\_date' => $request->end\_date,  // ]);  // }    // public function exportPdf(Request $request)  // {  // // Validasi input  // $request->validate([  // 'start\_date' => 'nullable|date',  // 'end\_date' => 'nullable|date|after\_or\_equal:start\_date',  // 'search' => 'nullable|string|max:255',  // ]);  // // Inisialisasi query builder  // $query = Pegawai::query();  // // Filter berdasarkan NIP  // if ($request->filled('search')) {  // $query->where('nip', 'like', '%' . $request->search . '%');  // }  // // Filter berdasarkan tanggal  // if ($request->filled('start\_date') && $request->filled('end\_date')) {  // $query->whereBetween('created\_at', [$request->start\_date, $request->end\_date]);  // }  // // Ambil data berdasarkan filter  // $pegawai = $query->get();  // // Generate PDF  // $pdf = Pdf::loadView('pegawai.pdf', compact('pegawai'));  // $pdf->setPaper('A4', 'landscape');  // // Download PDF  // return $pdf->download('pegawai.pdf');  // }  // Fungsi untuk menampilkan data pegawai  public function index(Request $request)  {  $pegawai = Pegawai::all();  \Carbon\Carbon::setLocale('id');  $search = $request->input('search'); | $pegawai = Pegawai::when($search, function ($query, $search) {  return $query->where('nip', 'like', '%' . $search . '%');  })->paginate(50);  return view('pegawai.index', [  "title" => "Pegawai",  "data" => $pegawai  ], compact('pegawai'));  }  // Fungsi untuk menampilkan form tambah data pegawai  public function create(): View  {  return view('pegawai.create')->with([  "title" => "Tambah Data Pegawai",  "user" => User::all(),  "unitkerja" => Unitkerja::all(),  "jabatan" => Jabatan::all()  ]);  }  // Fungsi untuk menyimpan data pegawai yang ditambahkan  public function store(Request $request): RedirectResponse  {  $validated = $request->validate([  "user\_id" => "required",  "nama" => "required",  "nip" => "required|digits\_between:6,16",  "jeniskelamin" => "required",  "usia" => "required",  "masakerja" => "required",  "golongandarah" => "required",  "statuskeluarga" => "required",  "agama" => "required",  "unitkerja\_id" => "required",  "jabatan\_id" => "required",  "tempat\_lahir" => "required",  "tanggal\_lahir" => "required|date",  "alamat" => "required",  "tmt" => "required",  "foto" => "nullable|image|mimes:jpeg,png,jpg|max:5048",  ]);  // $request->validate([  // 'nama' => 'required|string|max:255',  // 'tanggal\_lahir' => [  // 'required',  // 'date',  // 'before:' . now()->subYears(17)->format('Y-m-d'), // Validasi minimal umur 17 tahun  // ],  // ], [  // // Pesan error kustom (opsional)  // 'tanggal\_lahir.before' => 'Pegawai harus berusia minimal 17 tahun.',  // ]);  $validated['usia'] = \Carbon\Carbon::parse($validated['tanggal\_lahir'])->age;  $validated['masakerja'] = \Carbon\Carbon::parse($validated['tmt'])->age;  // Gabungkan tempat dan tanggal lahir  $ttl = $request->tempat\_lahir . ', ' . $request->tanggal\_lahir;  // Data yang akan disimpan  $data = $request->except(['tempat\_lahir', 'tanggal\_lahir']);  $data['ttl'] = $ttl;  if ($request->hasFile('foto')) {  $data['foto'] = $request->file('foto')->store('pegawai/foto', 'public');  }  Pegawai::create($data);  return redirect()->route('pegawai.index')->with('success', 'Data Pegawai Berhasil Ditambahkan');  }  // Fungsi untuk menampilkan form edit data pegawai  public function edit(Pegawai $pegawai): View  {  return view('pegawai.edit', compact('pegawai'))->with([  "title" => "Ubah Data Pegawai",  "user" => User::all(),  "unitkerja" => Unitkerja::all(),  "jabatan" => Jabatan::all()  ]);  } | // Fungsi untuk memperbarui data pegawai yang diedit  public function update(Pegawai $pegawai, Request $request): RedirectResponse  {  $validated = $request->validate([  "user\_id" => "required",  "nama" => "required",  "nip" => "required|digits\_between:6,16",  "jeniskelamin" => "required",  "usia" => "required",  "masakerja" => "required",  "golongandarah" => "required",  "statuskeluarga" => "required",  "agama" => "required",  "unitkerja\_id" => "required",  "jabatan\_id" => "required",  "tempat\_lahir" => "",  "tanggal\_lahir" => "",  "alamat" => "required",  "tmt" => "required",  "foto" => "nullable|image|mimes:jpeg,png,jpg|max:5048",  ]);  // Validasi data  // $request->validate([  // 'nama' => 'required|string|max:255',  // 'tanggal\_lahir' => [  // 'required',  // 'date',  // 'before:' . now()->subYears(17)->format('Y-m-d'),  // ],  // ], [  // 'tanggal\_lahir.before' => 'Pegawai harus berusia minimal 17 tahun.',  // ]);  $validated['usia'] = \Carbon\Carbon::parse($validated['tanggal\_lahir'])->age;  $validated['masakerja'] = \Carbon\Carbon::parse($validated['tmt'])->age;  // Data awal  $ttl = $request->tempat\_lahir . ', ' . $request->tanggal\_lahir;  // Pisahkan data berdasarkan koma  $data = explode(', ', $ttl);  // Pastikan array memiliki dua elemen  $tempat\_lahir = isset($data[0]) ? $data[0] : '';  $tanggal\_lahir = isset($data[1]) ? $data[1] : '';  // Output hasil  echo "Tempat Lahir: " . $tempat\_lahir . "<br>";  echo "Tanggal Lahir: " . $tanggal\_lahir . "<br>";  $data = $request->all();  if ($request->hasFile('foto')) {  // Hapus foto lama jika ada  if ($pegawai->foto) {  Storage::disk('public')->delete($pegawai->foto);  }  $data['foto'] = $request->file('foto')->store('pegawai/foto', 'public');  }  $pegawai->update($data);  return redirect()->route('pegawai.index')->with('success', 'Data Pegawai Berhasil Diubah');  }  // Fungsi untuk menampilkan detail data pegawai  public function show(): View  {  $pegawai = Pegawai::all();  return view('pegawai.show')->with([  "title" => "Tampil Data Pegawai",  "data" => $pegawai  ]);  }  // Fungsi untuk menghapus data pegawai  public function destroy($id): RedirectResponse  {  $pegawai = Pegawai::findOrFail($id);  // Hapus foto jika ada  if ($pegawai->foto) {  Storage::disk('public')->delete($pegawai->foto);  }  $pegawai->delete();  return redirect()->route('pegawai.index')->with('error', 'Data Pegawai Berhasil Dihapus');  }  } |

1. **JabatanController**

|  |  |
| --- | --- |
| <?php  namespace App\Http\Controllers;  use Illuminate\Http\Request;  use App\Models\Jabatan;  use Illuminate\View\View;  use Illuminate\Http\RedirectResponse;  class JabatanController extends Controller  {  // Fungsi untuk menampilkan data Jabatan  public function index()  {  return view('jabatan.index', [  "title" => "Jabatan",  "data" => Jabatan::all()  ]);  }  // Fungsi untuk menampilkan form tambah data Jabatan  public function create():View  {  return view('jabatan.index')->with(["title" => "Tambah Data Jabatan"]);  }  // Fungsi untuk menyimpan data Jabatan yang ditambahkan  public function store(Request $request): RedirectResponse  {  $request->validate([  "nama"=>"required",  ]);  if (empty($request['hp'])) {  $request['hp']='null';  if (empty($request['alamat']))  $request['alamat']='null'; | Jabatan::create($request->all());  return redirect()->route('jabatan.index')->with('success','Data Jabatan Berhasil Ditambahkan');  }  // Fungsi untuk menampilkan form edit data Jabatan  public function edit(Jabatan $jabatan):View  {  return view('jabatan.editjabatan',compact('jabatan'))->with([  "title" => "Ubah Data Jabatan",  ]);  }  // Fungsi untuk memperbarui data Jabatan yang diedit  public function update(Jabatan $jabatan, Request $request):RedirectResponse  {  $request->validate([  "nama"=>"required",  ]);  if (empty($request['hp'])) {  $request['hp']='null';  if (empty($request['alamat']))  $request['alamat']='null';  }  $jabatan->update($request->all());  return redirect()->route('jabatan.index')->with('updated','Data Jabatan Berhasil Diubah');  }  // Fungsi untuk menghapus data Jabatan  public function destroy($id):RedirectResponse  {  Jabatan::where('id',$id)->delete();  return redirect()->route('jabatan.index')->with('deleted','Data Pegawai Berhasil Dihapus');  }  } |

1. **UnitKerjaController**

|  |  |
| --- | --- |
| <?php  namespace App\Http\Controllers;  use Illuminate\Http\Request;  use App\Models\Unitkerja;  use Illuminate\View\View;  use Illuminate\Http\RedirectResponse;  class UnitkerjaController extends Controller  {  // Fungsi untuk menampilkan data Unitkerja  public function index()  {  // Mengambil semua data Unitkerja dari database  return view('unitkerja.index', [  "title" => "Unitkerja",  "data" => Unitkerja::all()  ]);  }  // Fungsi untuk menampilkan form tambah data Unitkerja  public function create():View  {  return view('unitkerja.index')->with(["title" => "Tambah Data Unitkerja"]);  }  // Fungsi untuk menyimpan data Unitkerja yang ditambahkan  public function store(Request $request): RedirectResponse  {  $request->validate([  "nama"=>"required",  ]);  if (empty($request['hp'])) {  $request['hp']='null';  if (empty($request['alamat']))  $request['alamat']='null'; | }  Unitkerja::create($request->all());  return redirect()->route('unitkerja.index')->with('success','Data Unitkerja Berhasil Ditambahkan');  }  // Fungsi untuk menampilkan form edit data Unitkerja  public function edit(Unitkerja $unitkerja):View  {  return view('unitkerja.editunitkerja',compact('unitkerja'))->with([  "title" => "Ubah Data Unitkerja",  ]);  }  // Fungsi untuk memperbarui data Unitkerja yang diedit  public function update(Unitkerja $unitkerja, Request $request):RedirectResponse  {  $request->validate([  "nama"=>"required",  ]);  if (empty($request['hp'])) {  $request['hp']='null';  if (empty($request['alamat']))  $request['alamat']='null';  }  $unitkerja->update($request->all());  return redirect()->route('unitkerja.index')->with('updated','Data Unitkerja Berhasil Diubah');  }  // Fungsi untuk menghapus data Unitkerja  public function destroy($id):RedirectResponse  {  Unitkerja::where('id',$id)->delete();  return redirect()->route('unitkerja.index')->with('deleted','Data Pegawai Berhasil Dihapus');  }  } |

1. **DashboardController**

|  |
| --- |
| <?php  namespace App\Http\Controllers;  use Illuminate\Http\Request;  use App\Models\Pegawai;  use App\Models\User;  class DashboardController extends Controller  {  public function index()  {  $data = Pegawai::all(); // Ambil data dari database  $title = 'Dashboard'; // Set variabel title  $user = User::all();  return view('welcome', compact('data', 'title', 'user'));  }    public function getChartData()  {  $totalPegawai = Pegawai::count(); // Menghitung jumlah pegawai  $totalUser = User::count(); // Menghitung jumlah user  return response()->json([  'pegawai' => $totalPegawai,  'user' => $totalUser  ]);    }  } |

1. **LoginController**

|  |  |
| --- | --- |
| <?php  namespace App\Http\Controllers;  use Illuminate\Http\Request;  use Illuminate\Support\Facades\Auth;  use Illuminate\Http\RedirectResponse;  class LoginController extends Controller  {  // Fungsi untuk menangani proses autentikasi login  public function authenticate(Request $request): RedirectResponse  {  // Validasi input dari form login (email dan password harus diisi)  $credentials = $request->validate([  'email' => ['required'], // Email wajib diisi  'password' => ['required'], // Password wajib diisi  ]);  // Memeriksa apakah kredensial yang dimasukkan cocok dengan data di database  if (Auth::attempt($credentials)) {  // Jika autentikasi berhasil, session di-re-generate untuk keamanan  $request->session()->regenerate();  // Redirect ke halaman yang dituju setelah login berhasil  return redirect()->intended('/');  }  // Jika autentikasi gagal, kembali ke halaman login dengan pesan error  return back()->with('loginError', 'Login Failed');  }  // Fungsi untuk menampilkan halaman login  public function loginView()  {  // Mengarahkan ke view 'login' untuk menampilkan form login  return view('login'); | }  // Fungsi untuk melakukan logout  public function logout(Request $request): RedirectResponse  {  // Proses logout user yang sedang login  Auth::logout();  // Invalidasi session agar tidak bisa dipakai lagi  $request->session()->invalidate();  // Regenerasi token untuk menghindari serangan CSRF  $request->session()->regenerateToken();  // Redirect ke halaman login setelah logout  return redirect('/login');  }  } |

1. **LaporanController**

|  |  |  |
| --- | --- | --- |
| <?php  namespace App\Http\Controllers;  use App\Models\Pegawai;  use Illuminate\Http\Request;  use Barryvdh\DomPDF\Facade\Pdf;  use Carbon\Carbon;  class LaporanController extends Controller  {  public function index(Request $request)  {  // Validasi input  $request->validate([  'start\_date' => 'nullable|date',  'end\_date' => 'nullable|date|after\_or\_equal:start\_date',  'search' => 'nullable|string',  ]);    // Inisialisasi query builder dari model Pegawai  $query = Pegawai::query();    // Filter berdasarkan pencarian NIP  if ($request->filled('search')) {  $query->where('nip', 'like', '%' . $request->search . '%');  }    // Filter berdasarkan tanggal  if ($request->filled('start\_date') && $request->filled('end\_date')) {  $startDate = Carbon::parse($request->start\_date)->startOfDay(); | $endDate = Carbon::parse($request->end\_date)->endOfDay();  $query->whereBetween('created\_at', [$startDate, $endDate]);  }    // Paginasi hasil data  $pegawai = $query->paginate(10);    // Kirim data ke view  return view('laporan.laporan', [  'data' => $pegawai,  'pegawai' => $pegawai,  'title' => 'Pegawai',  'search' => $request->search,  'start\_date' => $request->start\_date,  'end\_date' => $request->end\_date,  ]);  }  public function exportPdf(Request $request)  {  $request->validate([  'start\_date' => 'nullable|date|before\_or\_equal:today',  'end\_date' => 'nullable|date|after\_or\_equal:start\_date|before\_or\_equal:today',  'search' => 'nullable|string|max:255',  ]);    // Inisialisasi query builder  $query = Pegawai::query(); | // Filter berdasarkan NIP  if ($request->filled('search')) {  $query->where('nip', 'like', '%' . $request->search . '%');  }  // Filter berdasarkan tanggal  if ($request->filled('start\_date') && $request->filled('end\_date')) {  $startDate = Carbon::parse($request->start\_date)->startOfDay();  $endDate = Carbon::parse($request->end\_date)->endOfDay();  $query->whereBetween('created\_at', [$startDate, $endDate]);  }    // Ambil data berdasarkan filter  $pegawai = $query->get();  // Generate PDF  $pdf = Pdf::loadView('laporan.pdf', compact('pegawai'));  $pdf->setPaper('A4', 'landscape');  // Download PDF  return $pdf->download('laporan.pdf');  }  } |

## Potongan Kode Program View

1. **Dashboard**

|  |  |
| --- | --- |
| <div class="row">  <div class="col-xl-3 col-md-6 mb-4">  <div class="card border-left-success shadow h-100 py-2">  <div class="card-body">  <div class="row no-gutters align-items-center">  <div class="col mr-2">  <div class="text-xs font-weight-bold text-success text-uppercase mb-1">  Total Pegawai:  </div>  <div class="h5 mb-0 font-weight-bold text-gray-800">  {{ $data->count() }}  </div>  </div>  <div class="col-auto">  <i class="fas fa-arrow-down fa-2x text-gray-300"></i>  </div>  </div>  </div>  </div>  </div> | <div class="col-xl-3 col-md-6 mb-4">  <div class="card border-left-success shadow h-100 py-2">  <div class="card-body">  <div class="row no-gutters align-items-center">  <div class="col mr-2">  <div class="text-xs font-weight-bold text-success text-uppercase mb-1">  Total User:  </div>  <div class="h5 mb-0 font-weight-bold text-gray-800">  {{ $user->count() }}  </div>  </div>  <div class="col-auto">  <i class="fas fa-arrow-down fa-2x text-gray-300"></i>  </div>  </div>  </div>  </div>  </div>  </div> |

1. **Login**

|  |
| --- |
| <form action="/login" method="post">  @csrf  <div class="input-group mb-3">  <input type="email" class="form-control @error('email') is-invalid @enderror" placeholder="Email" id="email" name="email" value="{{ old('email') }}">  <div class="input-group-append">  <div class="input-group-text">  <span class="fas fa-envelope"></span>  </div>  </div>  @error('email')  <div class="invalid-feedback">  {{ $message }}  </div>  @enderror  </div>  <div class="input-group mb-3">  <input type="password" class="form-control" placeholder="Password" id="password" name="password">  <div class="input-group-append">  <div class="input-group-text">  <span class="fas fa-eye" id="togglePassword"></span>  </div>  </div>  </div>  <div class="card-footer">  <button type="reset" class="btn btn-warning">Cancel</button>  <button type="submit" class="btn btn-primary float-right">Login</button>  </div>  </form> |

1. **Admin**

|  |  |
| --- | --- |
| **Index.blade.php (Tambah)** | **Index.blade.php (Tampil)** |
| <form action="{{ route('user.store') }}" method="POST">  @csrf  <div class=" card-body">  <div class="form-group">  <label for="name">Nama</label>  <input type="text" class="form-control" id="name" name="name" placeholder=" Nama Lengkap" required value="{{old('name')}}">  </div>  <div class="form-group">  <label for="email">Email</label>  <input type="email" class="form-control @error('email') is-invalid @enderror" id="email" name="email" placeholder="ahmadi@example.com" required value="{{old('email')}}">  @error('email')  <div class="invalid-feedback">  {{ $message }}  </div>  @enderror  </div>  <div class="form-group">  <label for="password">Password</label>  <div class="input-group my-input-group"> <input type="password" class="form-control" id="password" name="password" required>  <div class="input-group-append">  <span class="input-group-text">  <i class="fa fa-eye" onclick="togglePassword()"></i>  </span>  </div>  </div>  </div>  </div>  <!-- /.card-body -->  <div class="card-footer">  <button type="submit" class="btn btn-success float-right">Simpan</button>  </div>  </form> | <table id="example1" class="table table-bordered table-striped ">  <thead>  <tr>  <th>No</th>  <th>Nama </th>  <th>Email</th>  </tr>  </thead>  <tbody>  @foreach($data as $dt)  <tr>  <td>{{ $loop->iteration }}</td>  <td>{{ $dt->name }}</td>  <td>{{ $dt->email }}</td>  </tr>  @endforeach  </tbody>  </table> |

1. **Layouts**

|  |
| --- |
| **Template.blade.php** |
| **<ul class="navbar-nav bg-gradient-dark sidebar sidebar-dark accordion" id="accordionSidebar">**  **<!-- Sidebar - Brand -->**  **<a class="sidebar-brand d-flex align-items-center justify-content-center" href="/">**  **<div class="sidebar-brand-icon rotate-n-15"></div>**  **<i class="fas fa-home"></i>**  **<div class="sidebar-brand-text mx-3">Data Pegawai</div>**  **</a>**  **<!-- Divider -->**  **<hr class="sidebar-divider my-0">**  **<!-- Nav Item - Dashboard -->**  **<li class="nav-item {{ ($title === 'Home') ? 'active' : '' }}">**  **<a class="nav-link" href="/">**  **<i class="fas fa-tachometer-alt"></i>**  **<span>Home</span>**  **</a>**  **</li>**  **<!-- Nav Item - Data Master -->**  **<li class="nav-item {{ ($title === 'Produk') ? 'active' : '' }}">**  **<a class="nav-link collapsed" href="#" data-toggle="collapse" data-target="#collapseDataMaster"**  **aria-expanded="true" aria-controls="collapseDataMaster">**  **<i class="fas fa-boxes"></i>**  **<span>Data Pegawai</span>**  **</a>**  **<div id="collapseDataMaster" class="collapse {{ ($title === 'Produk') ? 'show' : '' }}" aria-labelledby="headingDataMaster"**  **data-parent="#accordionSidebar">**  **<div class="bg-white py-2 collapse-inner rounded">**  **<h6 class="collapse-header">Master Data:</h6>**  **<a class="collapse-item" href="{{ route('unitkerja.index') }}">Unit Kerja</a>**  **<a class="collapse-item" href="{{ route('jabatan.index') }}">Jabatan</a>**  **</div>**  **</div>**  **</li>**  **<!-- Nav Item - Pegawai -->**  **<li class="nav-item {{ ($title === 'Pegawai') ? 'active' : '' }}">**  **<a class="nav-link" href="{{ route('pegawai.index') }}">**  **<i class="fas fa-users"></i>**  **<span>Pegawai</span>**  **</a>**  **</li>**  **<!-- Nav Item - Admin -->**  **<li class="nav-item {{ ($title === 'user') ? 'active' : '' }}">**  **<a class="nav-link" href="{{ route('user.index') }}">**  **<i class="fas fa-user"></i>**  **<span>Admin</span>**  **</a>**  **</li>**  **<li class="nav-item {{ ($title === 'Pegawai') ? 'active' : '' }}">**  **<a class="nav-link" href="{{ route('laporan.index') }}">**  **<i class="fas fa-regular fa-file-pdf"></i>**  **<span>Laporan</span>**  **</a>**  **</li>**  **<!-- Nav Item - Laporan -->**  **<div class="text-center d-none d-md-inline">**  **<br>**  **<button class="rounded-sn border-0" id="sidebarToggle"></button>**  **</div>**  **<hr class="sidebar-divider d-none d-md-block">**  **</ul>** |

1. **Pegawai**

|  |  |  |
| --- | --- | --- |
| **Index.blade.php** | | |
| <table id="example1" class="table table-bordered table-striped nowrap">  <thead>  <tr>  <th>No</th>  <th>Foto</th>  <th>Nama</th>  <th>NIP</th>  <th>Jabatan</th>  <th>Jenis Kelamin</th>  <th>Ttl</th>  <th>Usia</th>  <th>tmt</th>  <th>Masa Kerja</th>  <th>Status Keluarga</th>  <th>Golongan Darah</th>  <th>Agama</th>  <th>Unit Kerja</th>  <th>Alamat</th>  <th>Petugas</th>  <th>Aksi</th>  </tr>  </thead>  <tbody>  @foreach($data as $dt)  <tr>  <td>{{ $loop->iteration }}</td>  <td>  @if ($dt->foto)  <img src="{{ asset('storage/' . $dt->foto) }}" alt="Foto {{ $dt->nama }}" width="100">  @else  Tidak ada foto  @endif  </td> | <td>{{ $dt->nama }}</td>  <td>{{ $dt->nip }}</td>  <td>{{ $dt->jabatan->nama}}</td>  <td>{{ $dt->jeniskelamin }}</td>  <td>{{ $dt->ttl }}</td>  <td>{{ $dt->usia }}</td>  <td>{{ $dt->tmt }}</td>  <td>{{ $dt->masakerja }}</td>  <td>{{ $dt->statuskeluarga}}</td>  <td>{{ $dt->golongandarah }}</td>  <td>{{ $dt->agama}}</td>  <td>{{ $dt->unitkerja->nama }}</td>  <td>{{ $dt->alamat }}</td>  <td>{{ $dt->user->name}}</td>  <td>  <div class="btn-group" role="group">  <form action="{{ route('pegawai.destroy', $dt->id) }}" method="POST">  @csrf  @method('DELETE')  <button type="submit" class="btn btn-danger btn-sm">  <i class="fas fa-trash"></i>  </button>  </form>  <a href="{{ route('pegawai.edit', $dt->id) }}" class="btn btn-warning btn-sm">  <i class="fas fa-edit"></i>  </a>  <a href="{{ route('pegawai.show', $dt->id) }}" class="btn btn-success btn-sm">  <i class="fas fa-eye"></i>  </a>  </div>  </td>  </tr>  @endforeach  </tbody>  </table> | |
| **Create.blade.php** | | |
| <form action="{{ route('pegawai.store') }}" method="POST" enctype="multipart/form-data">  @csrf  <div class="card-body">  <div class="row">  <!-- Kolom Kiri -->  <div class="col-md-6">  <div class="form-group">  <label for="nama">Nama Pegawai</label>  <input type="text" class="form-control" id="nama" name="nama" placeholder="Masukkan Nama Pegawai" required>  </div>  <div class="form-group">  <label for="nip">NIP</label>  <input type="number" class="form-control" id="nip" name="nip" placeholder="Masukkan NIP" maxlength="16" required>  </div>  <div class="form-group">  <label for="jeniskelamin">Jenis Kelamin</label>  <select class="form-control" id="jeniskelamin" name="jeniskelamin" required>  <option value="Laki-laki" {{ old('jeniskelamin') == 'Laki-laki' ? 'selected' : '' }}>Laki-laki</option>  <option value="Perempuan" {{ old('jeniskelamin') == 'Perempuan' ? 'selected' : '' }}>Perempuan</option>  </select>  </div>  <div class="form-group">  <label for="tempat\_lahir">Tempat Lahir</label>  <input type="text" class="form-control" id="tempat\_lahir" name="tempat\_lahir" placeholder="Masukkan Tempat Lahir" value="{{ old('tempat\_lahir') }}">  </div>  <div class="form-group">  <label for="tanggal\_lahir">Tanggal Lahir</label>  <input type="date" class="form-control" id="tanggal\_lahir" name="tanggal\_lahir" value="{{ old('tanggal\_lahir') }}">  <small id="tanggalLahirError" class="text-danger" style="display: none;">  Pegawai harus berusia minimal 17 tahun!  </small>  </div>  <div class="form-group">  <label for="usia">Usia</label>  <input type="number" class="form-control" id="usia" name="usia" readonly>  </div>  <div class="form-group">  <label for="tmt">tmt</label>  <input type="date" class="form-control" id="tmt" name="tmt" value="{{ old('tmt') }}">    </div>  <div class="form-group">  <label for="masakerja">Masa Kerja</label>  <input type="number" class="form-control" id="masakerja" name="masakerja" placeholder="Masukkan Masa Kerja">  </div>  </div>  <!-- Kolom Kanan -->  <div class="col-md-6">  <div class="form-group">  <label>User</label>  <select class="form-control" name="user\_id">  @foreach($user as $dt)  <option value="{{ $dt->id }}">{{ $dt->name }}</option>  @endforeach  </select>  </div>  <div class="form-group">  <label for="statuskeluarga">statuskeluarga</label>  <select class="form-control" id="statuskeluarga" name="statuskeluarga">  <option value="" disabled selected>Pilih status keluarga</option>  <option value="menikah" {{ old('statuskeluarga') == 'menikah' ? 'selected' : '' }}>menikah</option>  </select> | | <option value="belum menikah" {{ old('statuskeluarga') == 'belum menikah' ? 'selected' : '' }}>belum menikah</option>  <option value="cerai" {{ old('statuskeluarga') == 'cerai' ? 'selected' : '' }}>cerai</option>  </select>  </div>  <div class="form-group">  <label for="agama">Agama</label>  <select class="form-control" id="agama" name="agama">  <option value="" disabled selected>Pilih Agama</option>  <option value="Islam" {{ old('agama') == 'Islam' ? 'selected' : '' }}>Islam</option>  <option value="Kristen Protestan" {{ old('agama') == 'Kristen Protestan' ? 'selected' : '' }}>Kristen Protestan</option>  <option value="Kristen Katolik" {{ old('agama') == 'Kristen Katolik' ? 'selected' : '' }}>Kristen Katolik</option>  <option value="Hindu" {{ old('agama') == 'Hindu' ? 'selected' : '' }}>Hindu</option>  <option value="Buddha" {{ old('agama') == 'Buddha' ? 'selected' : '' }}>Buddha</option>  <option value="Konghucu" {{ old('agama') == 'Konghucu' ? 'selected' : '' }}>Konghucu</option>  </select>  </div>  <div class="form-group">  <label>Unit Kerja</label>  <select class="form-control" name="unitkerja\_id">  @foreach($unitkerja as $dt)  <option value="{{ $dt->id }}">{{ $dt->nama }}</option>  @endforeach  </select>  </div>  <div class="form-group">  <label>Jabatan</label>  <select class="form-control" name="jabatan\_id">  @foreach($jabatan as $dt )  <option value="{{ $dt->id }}">{{ $dt->nama }}</option>  @endforeach  </select>  </div>  <div class="form-group">  <label for="alamat">Alamat</label>  <textarea class="form-control" id="alamat" name="alamat" rows="3" placeholder="Masukkan Alamat"></textarea>  </div>  <div class="form-group">  <label for="foto">Foto</label>  <input type="file" class="form-control-file" id="foto" name="foto">  </div>  </div>  </div>  </div>  <!-- Card Footer -->  <div class="card-footer text-right">  <button type="submit" class="btn btn-success"><i class="fas fa-save"></i> Simpan</button>  <a href="{{ route('pegawai.index') }}" class="btn btn-secondary"><i class="fas fa-arrow-left"></i> Batal</a>  </div>  </form> |
| **Edit.blade.php** | | |
| <form action="{{ route('pegawai.update', $pegawai->id) }}" method="POST" enctype="multipart/form-data">  @csrf  @method('PUT')  <div class="card-body">  <div class="row">  <!-- Kolom Kiri -->  <div class="col-md-6">  <div class="form-group">  <label for="nama">Nama Pegawai</label>  <input type="text" class="form-control" id="nama" name="nama" value="{{ old('nama', $pegawai->nama) }}" required>  </div>  <div class="form-group">  <label for="nip">NIP</label>  <input type="number" class="form-control" id="nip" name="nip" value="{{ old('nip', $pegawai->nip) }}" maxlength="16" required>  </div>  <div class="form-group">  <label for="jeniskelamin">Jenis Kelamin</label>  <select class="form-control" id="jeniskelamin" name="jeniskelamin" required>  <option value="Laki-laki" {{ old('jeniskelamin', $pegawai->jeniskelamin) == 'Laki-laki' ? 'selected' : '' }}>Laki-laki</option>  <option value="Perempuan" {{ old('jeniskelamin', $pegawai->jeniskelamin) == 'Perempuan' ? 'selected' : '' }}>Perempuan</option>  </select>  </div>  <div class="form-group">  <label for="tempat\_lahir">Tempat Lahir</label>  <input type="text" class="form-control" id="tempat\_lahir" name="tempat\_lahir" placeholder="Masukkan Tempat Lahir" value="{{ old('tempat\_lahir', $pegawai->tempat\_lahir) }}">  </div>  <div class="form-group">  <label for="tanggal\_lahir">Tanggal Lahir</label>  <input type="date" class="form-control" id="tanggal\_lahir" name="tanggal\_lahir" value="{{ old('tanggal\_lahir', $pegawai->tanggal\_lahir) }}">  </div>  <div class="form-group">  <label for="usia">Usia</label>  <input type="number" class="form-control" id="usia" name="usia" value="{{ old('usia', $pegawai->usia) }}">  </div>  <div class="form-group">  <label for="tmt">tmt</label>  <input type="date" class="form-control" id="tmt" name="tmt" value="{{ old('tmt', $pegawai->tmt) }}">  </div>  <div class="form-group">  <label for="masakerja">Masa Kerja</label>  <input type="number" class="form-control" id="masakerja" name="masakerja" value="{{ old('masakerja', $pegawai->masakerja) }}">  </div>  <div class="form-group">  <label>User</label>  <select class="form-control" name="user\_id">  @foreach($user as $dt)  <option value="{{ $dt->id }}" {{ $pegawai->user\_id == $dt->id ? 'selected' : '' }}>{{ $dt->name }}</option>  @endforeach  </select>  </div>  <div class="form-group">  <label for="statuskeluarga">statuskeluarga</label>  <select class="form-control" id="statuskeluarga" name="statuskeluarga">  <option value="" disabled selected>Pilih statuskeluarga</option>  <option value="menikah" {{ old('statuskeluarga', $pegawai->statuskeluarga) == 'menikah' ? 'selected' : '' }}>menikah</option>  <option value="belum menikah" {{ old('statuskeluarga', $pegawai->statuskeluarga) == 'belum menikah' ? 'selected' : '' }}>belum menikah</option>  <option value="cerai" {{ old('statuskeluarga', $pegawai->statuskeluarga) == 'cerai' ? 'selected' : '' }}>cerai</option> | | </select>  </div>  </div>  <!-- Kolom Kanan -->  <div class="col-md-6">  </div>  <div class="form-group">  <label for="agama">Agama</label>  <select class="form-control" id="agama" name="agama">  <option value="" disabled selected>Pilih Agama</option>  <option value="Islam" {{ old('agama', $pegawai->agama) == 'Islam' ? 'selected' : '' }}>Islam</option>  <option value="Kristen Protestan" {{ old('agama', $pegawai->agama) == 'Kristen Protestan' ? 'selected' : '' }}>Kristen Protestan</option>  <option value="Kristen Katolik" {{ old('agama', $pegawai->agama) == 'Kristen Katolik' ? 'selected' : '' }}>Kristen Katolik</option>  <option value="Hindu" {{ old('agama', $pegawai->agama) == 'Hindu' ? 'selected' : '' }}>Hindu</option>  <option value="Buddha" {{ old('agama', $pegawai->agama) == 'Buddha' ? 'selected' : '' }}>Buddha</option>  <option value="Konghucu" {{ old('agama', $pegawai->agama) == 'Konghucu' ? 'selected' : '' }}>Konghucu</option>  </select>  </div>  <div class="form-group">  <label>Unit Kerja</label>  <select class="form-control" name="unitkerja\_id">  @foreach($unitkerja as $dt)  <option value="{{ $dt->id }}" {{ $pegawai->unitkerja\_id == $dt->id ? 'selected' : '' }}>{{ $dt->nama }}</option>  @endforeach  </select>  </div>  <div class="form-group">  <label>Jabatan</label>  <select class="form-control" name="jabatan\_id">  @foreach($jabatan as $dt)  <option value="{{ $dt->id }}" {{ $pegawai->jabatan\_id == $dt->id ? 'selected' : '' }}>{{ $dt->nama }}</option>  @endforeach  </select>  </div>  <div class="form-group">  <label for="alamat">Alamat</label>  <textarea class="form-control" id="alamat" name="alamat" rows="3">{{ old('alamat', $pegawai->alamat) }}</textarea>  </div>  <div class="form-group">  <label for="foto" class="d-flex">Foto</label>  @if ($pegawai->foto)  <img src="{{ asset('storage/' . $pegawai->foto) }}" alt="Foto {{ $pegawai->nama }}" width="100" class="mb-3">  @else  Tidak ada foto  @endif    <input type="file" class="form-control-file" id="foto" name="foto">  </div>  </div>  </div>  </div>  <!-- Card Footer -->  <div class="card-footer text-right">  <button type="submit" class="btn btn-warning"><i class="fas fa-edit"></i> Ubah</button>  <a href="{{ route('pegawai.index') }}" class="btn btn-secondary"><i class="fas fa-arrow-left"></i> Batal</a>  </div>  </form> |
| **Show.blade.php** | | |
| <table class="table table-striped table-hover">  <tbody>  <tr>  <th>Nama Pegawai</th>  <td>{{ $data[0]->nama }}</td>  </tr>  <tr>  <th>NIP</th>  <td>{{ $data[0]->nip }}</td>  </tr>  <tr>  <th>jabatan</th>  <td>{{ $data[0]->jabatan->nama }}</td>  </tr>  <tr>  <th>Jenis Kelamin</th>  <td>{{ $data[0]->jeniskelamin }}</td>  </tr>  <tr>  <th>Ttl</th>  <td>{{ $data[0]->ttl }}</td>  </tr>    <tr>  <th>Usia</th>  <td>{{ $data[0]->usia }}</td>  </tr>  <tr>  <tr>  <th>tmt</th>  <td>{{ $data[0]->tmt }}</td>  </tr>  <tr>  <th>Masa Kerja</th>  <td>{{ $data[0]->masakerja }}</td>  </tr>  <tr> | | <th>User</th>  <td>{{ $data[0]->user->name }}</td>  </tr>  <tr>  <th>statuskeluarga</th>  <td>{{ $data[0]->statuskeluarga }}</td>  </tr>  <tr>  <th>Golongan darah</th>  <td>{{ $data[0]->golongandarah}}</td>  </tr>  <tr>  <th>Agama</th>  <td>{{ $data[0]->agama }}</td>  </tr>  <tr>  <th>Unit Kerja</th>  <td>{{ $data[0]->unitkerja->nama }}</td>  </tr>  <tr>  <th>Alamat</th>  <td>{{ $data[0]->alamat }}</td>  </tr>  </tbody>  </table> |

1. **Jabatan**

|  |  |  |
| --- | --- | --- |
| **Index.blade.php (Tampil)** | **Index.blade.php (Tambah)** | **Editjabatan.blade.php** |
| <table id="example1" class="table table-bordered table-striped ">  <thead>  <tr>  <th>No</th>  <th>Status</th>  <th>Aksi</th>  </tr>  </thead>  <tbody>  @foreach($data as $dt)  <tr>  <td>{{ $loop->iteration }}</td>  <td>{{ $dt->nama }}</td>  <td>  <div class="btn-group">  <a type="button" class="btn btn-warning" href="{{ route('jabatan.edit',$dt->id) }}">  <i class=" fas fa-edit"></i>  </a>  <form action="{{ route('jabatan.destroy',$dt->id)}}" method="POST">  @csrf  @method('DELETE')  <button type="submit" class="btn btn-danger">  <i class=" fas fa-trash"></i>  </button>  </form>  </div>  </td>  </tr>  @endforeach  </tbody>  </table> | <form action="{{ route('jabatan.store') }}" method="POST">  @csrf  <div class=" card-body">  <div class="form-group">  <label for="nama">Status Jabatan</label>  <input type="text" class="form-control" id="nama" name="nama" placeholder="">  </div>  </div>  <!-- /.card-body -->  <div class="card-footer">  <button type="submit" class="btn btn-success float-right">Simpan</button>  </div>  </form> | <form action="{{ route('jabatan.update',$jabatan->id) }}" method="POST">  @csrf  @method('PUT')    <div class=" card-body">  <div class="form-group">  <label for="nama">Nama</label>  <input type="text" class="form-control" id="nama" name="nama" placeholder="" value="{{$jabatan->nama}}">  </div>  </div>  <!-- /.card-body -->  <div class="card-footer">  <button type="submit" class="btn btn-warning float-right">Ubah</button>  </div>  </form> |

1. **Unit Jabatan**

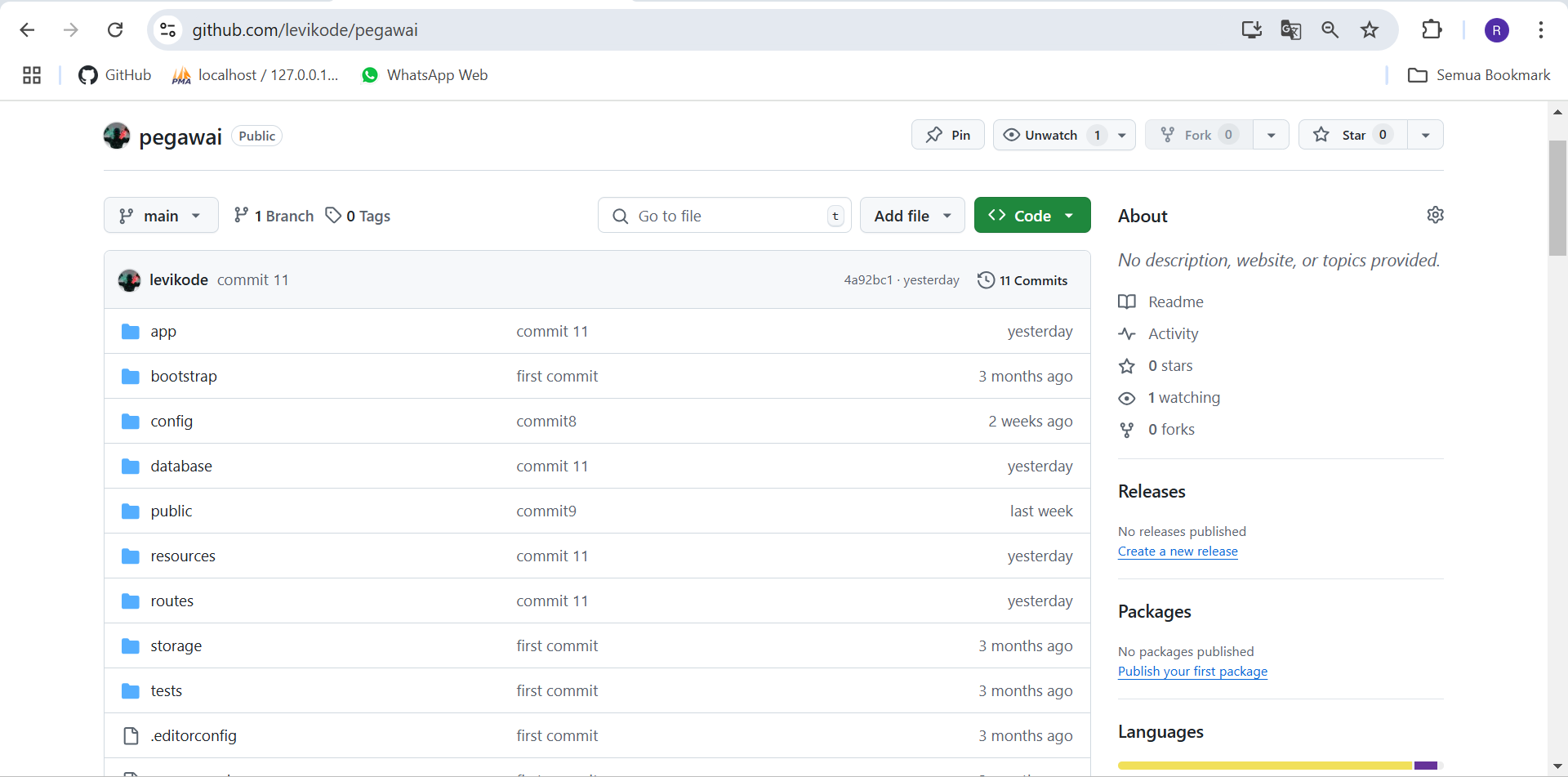
|  |  |  |
| --- | --- | --- |
| **Index.blade.php (Tampil)** | **Index.blade.php (Tambah)** | **Editunitkerja.blade.php** |
| <table id="example1" class="table table-bordered table-striped ">  <thead>  <tr>  <th>No</th>  <th>Nama</th>  <th>Aksi</th>  </tr>  </thead>  <tbody>  @foreach($data as $dt)  <tr>  <td>{{ $loop->iteration }}</td>  <td>{{ $dt->nama }}</td>  <td>  <div class="btn-group">  <a type="button" class="btn btn-warning" href="{{ route('unitkerja.edit',$dt->id) }}">  <i class=" fas fa-edit"></i>  </a>  <form action="{{ route('unitkerja.destroy',$dt->id)}}" method="POST">  @csrf  @method('DELETE')  <button type="submit" class="btn btn-danger">  <i class=" fas fa-trash"></i>  </button>  </form>  </div>  </td>  </tr>  @endforeach  </tbody>  </table> | <form action="{{ route('unitkerja.store') }}" method="POST">  @csrf  <div class=" card-body">  <div class="form-group">  <label for="nama">Nama Unit Kerja</label>  <input type="text" class="form-control" id="nama" name="nama" placeholder="">  </div>  </div>  <!-- /.card-body -->  <div class="card-footer">  <button type="submit" class="btn btn-success float-right">Simpan</button>  </div>  </form> | <form action="{{ route('unitkerja.update',$unitkerja->id) }}" method="POST">  @csrf  @method('PUT')    <div class=" card-body">  <div class="form-group">  <label for="nama">Nama</label>  <input type="text" class="form-control" id="nama" name="nama" placeholder="" value="{{$unitkerja->nama}}">  </div>  </div>  <!-- /.card-body -->  <div class="card-footer">  <button type="submit" class="btn btn-warning float-right">Ubah</button>  </div>  </form> |

1. **Laporan**

|  |  |
| --- | --- |
| **Laporan.blade.php** | |
| <table id="example1" class="table table-bordered table-striped nowrap">  <thead>  <tr>  <th>No</th>  <th>Foto</th>  <th>Nama</th>  <th>NIP</th>  <th>Jabatan</th>  <th>Jenis Kelamin</th>  <th>Ttl</th>  <th>Usia</th>  <th>tmt</th>  <th>Masa Kerja</th>  <th>Status Keluarga</th>  <th>Golongan Darah</th>  <th>Agama</th>  <th>Unit Kerja</th>  <th>Alamat</th>  <th>Petugas</th>  </tr>  </thead>  <tbody>  @foreach($data as $dt)  <tr>  <td>{{ $loop->iteration }}</td>  <td>  @if ($dt->foto)  <img src="{{ asset('storage/' . $dt->foto) }}" alt="Foto {{ $dt->nama }}" width="100">  @else  Tidak ada foto  @endif  </td> | <td>{{ $dt->nama }}</td>  <td>{{ $dt->nip }}</td>  <td>{{ $dt->jabatan->nama}}</td>  <td>{{ $dt->jeniskelamin }}</td>  <td>{{ $dt->ttl }}</td>  <td>{{ $dt->usia }}</td>  <td>{{ $dt->tmt }}</td>  <td>{{ $dt->masakerja }}</td>  <td>{{ $dt->statuskeluarga}}</td>  <td>{{ $dt->golongandarah }}</td>  <td>{{ $dt->agama}}</td>  <td>{{ $dt->unitkerja->nama }}</td>  <td>{{ $dt->alamat }}</td>  <td>{{ $dt->user->name}}</td>  </tr>  @endforeach  </tbody>  </table> |

|  |  |
| --- | --- |
| **Pdf.blade.php** | |
| <table>  <thead>  <tr>  <th>No</th>  <th>Nama</th>  <th>NIP</th>  <th>Jabatan</th>  <th>Jenis Kelamin</th>  <th>Ttl</th>  <th>Usia</th>  <th>tmt</th>  <th>Masa Kerja</th>  <th>Status Keluarga</th>  <th>Golongan Darah</th>  <th>Agama</th>  <th>Unit Kerja</th>  <th>Alamat</th>  <th>Petugas</th>  </tr>  </thead>  <tbody>  @foreach ($pegawai as $index => $pegawai)  <tr> | <td>{{ $loop->iteration }}</td>    <td>{{ $pegawai->nama }}</td>  <td>{{ $pegawai->nip }}</td>  <td>{{ $pegawai->jabatan->nama}}</td>  <td>{{ $pegawai->jeniskelamin }}</td>  <td>{{ $pegawai->ttl }}</td>  <td>{{ $pegawai->usia }}</td>  <td>{{ $pegawai->tmt }}</td>  <td>{{ $pegawai->masakerja }}</td>  <td>{{ $pegawai->statuskeluarga}}</td>  <td>{{ $pegawai->golongandarah}}</td>  <td>{{ $pegawai->agama}}</td>  <td>{{ $pegawai->unitkerja->nama }}</td>  <td>{{ $pegawai->alamat }}</td>  <td>{{ $pegawai->user->name}}</td>  </tr>    @endforeach  </tbody>  </table> |

## Pengembangan (Github)



**Source Code APK lengkapnya dapat dilihat pada link Github berikut:**

[**https://github.com/levikode/pegawai**](https://github.com/levikode/pegawai)

# BAB IV PENGUJIAN APLIKASI

## Instrumen Validasi Pengujian Aspek Fungsionalitas Aplikasi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| fitur | pendukung | urgensi/tingkat kepentingan | Hasil Pengujian | |
| CRUD | Lulus | Gagal |
| login | email | Read | lulus |  |
| password |
| dashboard | admin | Read | lulus |  |
| pegawai |
| admin | namaadmin | Create | lulus |  |
| email | Read | Lulus |  |
| password | Update | Lulus |  |
|  | Delete | Lulus |  |
| pegawai->jabatan | namajabatan | Create | Lulus |  |
| Read | Lulus |  |
| Update | Lulus |  |
| Delete | Lulus |  |
| pegawai->unitkerja | namaunitkerja | Create | Lulus |  |
| Read | Lulus |  |
| Update | Lulus |  |
| Delete | Lulus |  |
| pegawai | namapegawai | Create | Lulus |  |
| nip |
| jabatan |
| jeniskelamin | Read | Lulus |  |
| ttl |
| usia |
| tmt |
| masakerja | Delete | Lulus |  |
| statuskeluarga |
| golongandarah |
| agama | Update | Lulus |  |
| unitkerja |
| alamat |
| laporan & cetak | berdasarkan tanggal | Read | lulus |  |

## Hasil Pengujian Aspek Fungsionalitas Aplikasi Pembimbing

Berdasarkan pengujian aplikasi yang dilakukan oleh pembimbing maka didapatkan presentase sebagai berikut:

19

𝑃𝑟𝑒𝑠𝑒𝑛𝑡𝑎𝑠𝑒(%) = 𝑥 100

19

= 1 𝑥 100

= 100%

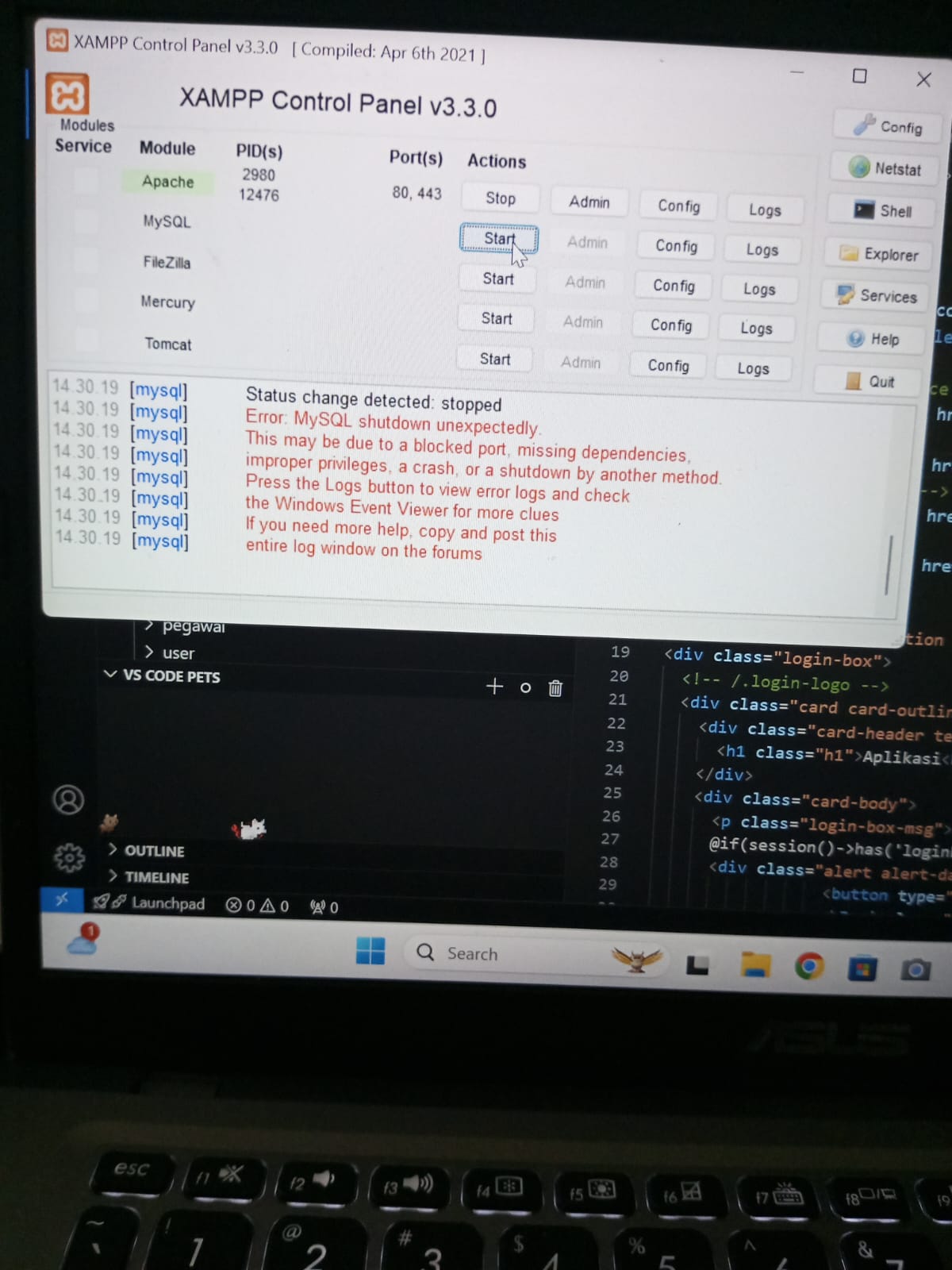
# BAB V PEMELIHARAAN APLIKASI

## Saran Pengembangan Aplikasi dari Penguji

## Kesimpulan

Aplikasi Pendataan Pegawai dirancang untuk memudahkan dalam proses pendataan. Sistem ini mencatat pendataan yang dilakukan oleh admin. Terdapat banyak fitur dalam aplikasi ini, salah satunya laporan untuk melihat rentang tanggal pendataan yang dilakukan oleh admin. Dengan banyaknya fitur ini sekiranya dapat membantu admin dalam proses pendataan pegawai.

# LAMPIRAN



Proses pembuatan aplikasi

# FORMULIR BIMBINGAN LAPORAN PENGEMBANGAN APLIKASI

# 

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Tanggal** | **Keterangan** | **Paraf** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 |  |  |  |
| 10 |  |  |  |
| 11 |  |  |  |