

BAB IV

RANCANGAN SISTEM DAN PROGRAM USULAN

4.1. Analisa Kebutuhan Software

Analisa kebutuhan sistem dilakukan dengan mengacu pada permasalahan yang ditemukan pada sistem berjalan, khususnya terkait keterlambatan rekapitulasi dan keterbatasan monitoring dokumen dari 73 OPD. Oleh karena itu, sistem usulan dirancang untuk mendukung proses unggah dokumen, verifikasi, dan monitoring secara terintegrasi. Berikut adalah rincian kebutuhan fungsional bagi masing-masing aktor:[18]:

- 1) Kebutuhan Fungsional untuk Operator OPD
 - a. **Otentikasi Akun:** Sistem menyediakan fitur login untuk menjamin keamanan akses operator ke dalam *dashboard* khusus masing-masing instansi.
 - b. **Modul Pengunggahan Dokumen:** Fasilitas bagi operator untuk mengunggah dokumen digital dengan menyertakan pemilihan kategori informasi (Berkala, Serta Merta, atau Setiap Saat) serta pengisian metadata seperti judul dan tahun laporan.
 - c. **Pemantauan Status:** Fitur untuk melacak status verifikasi dokumen yang telah dikirim, guna mengetahui apakah dokumen telah diterima atau memerlukan revisi.
 - d. **Pemberian Jawaban Permohonan:** Fitur khusus untuk merespon dan memberikan jawaban atas permohonan informasi publik yang diajukan oleh masyarakat kepada OPD terkait..
- 2) Kebutuhan Fungsional untuk Admin PPID

- a. **Manajemen Data Master:** Fitur untuk mengelola data fundamental sistem yang mencakup pengaturan daftar 73 OPD, klasifikasi kategori informasi, serta manajemen akun pengguna.
- b. **Verifikasi dan Validasi Dokumen:** Modul untuk meninjau berkas yang diunggah oleh OPD, memberikan catatan revisi, dan mengubah status dokumen menjadi "Terverifikasi" agar siap dipublikasikan.
- c. **Dashboard Monitoring Real-time:** Fasilitas visualisasi statistik yang menyajikan data penyerahan dokumen dari seluruh OPD secara langsung guna mempermudah pengawasan dan evaluasi.
- d. **Verifikasi Permohonan Informasi:** Fitur untuk memeriksa keabsahan identitas pemohon serta maksud permohonan informasi publik dari masyarakat sebelum diteruskan ke instansi terkait.

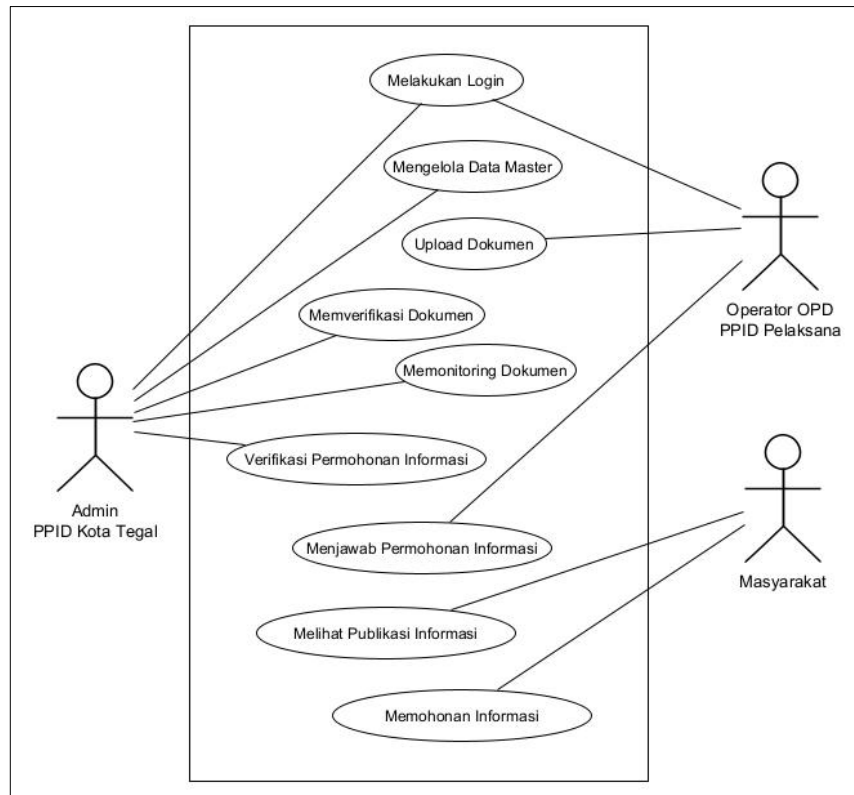
4.2. Desain

4.2.1. Desain Pemodelan Sistem

Pemodelan sistem dilakukan menggunakan Unified Modeling Language (UML) untuk menggambarkan interaksi antara pengguna (aktor) dengan sistem serta alur proses bisnis yang terjadi di dalamnya.

- a. Use Case Diagram

Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sistem dan bagaimana aktor berinteraksi dengan fungsi-fungsi tersebut. Terdapat tiga aktor utama dalam sistem ini: Admin PPID, Operator OPD.



Sumber : Hasil Penelitian (2025)

Gambar 4. 1 Use Case Diagram

Tabel 4. 1. Deskripsi Use Case Login

Use Case Name	:	Login
Use case Description	:	Proses bagi semua aktor (Admin & Operator OPD) untuk mendapatkan akses ke dalam sistem dengan memasukkan kredensial yang valid.
Actors	:	Admin PPID, Operator OPD
Pre-Condition	:	1. Sistem harus terhubung ke jaringan internet/lokal. 2. Pengguna berada di halaman login.
Post-Condition	:	Pengguna berhasil masuk ke dashboard sesuai dengan hak akses (level) masing-masing.
Fault Condition	:	1. Username atau password salah. 2. Akun dalam status non-aktif.
Main Scenarios	:	Serial No.
Extensions	:	2a. Data tidak cocok -> Sistem menampilkan pesan "Username/Password Salah".

Tabel 4. 2 Deskripsi Use Case Upload Dokumen

Use Case Name	:	Upload Dokumen
---------------	---	----------------

Use case Description	:	Operator dari 73 OPD mengunggah dokumen digital (seperti Lakin, Renstra, Laporan Keuangan) untuk dikelola oleh PPID Utama.
Actors	:	Operator OPD
Pre-Condition	:	1. Operator telah berhasil login. 2. Dokumen sudah tersedia dalam format digital (misal: PDF).
Post-Condition	:	Dokumen tersimpan di database dengan status "Menunggu Verifikasi".
Fault Condition	:	1. Format file tidak didukung (.exe, .zip, dll). 2. Ukuran file melebihi batas maksimal (misal: >10MB).
Main Scenarios	:	Serial No.
Extensions	:	3a. Format file salah -> Sistem menampilkan pesan error "Format file tidak valid". 3b. File terlalu besar -> Sistem menampilkan pesan error "Ukuran file terlalu besar".

Tabel 4. 3. Deskripsi Use Case: Verifikasi Dokumen

Use Case Name	:	Verifikasi Dokumen
Use case Description	:	Admin PPID meninjau dan memvalidasi kelayakan dokumen yang diunggah oleh OPD sebelum dipublikasikan.
Actors	:	Admin PPID
Pre-Condition	:	1. Admin telah login. 2. Terdapat dokumen yang masuk dengan status "Menunggu Verifikasi".
Post-Condition	:	Status dokumen diperbarui menjadi "Diterima", "Ditolak", atau "Revisi".
Fault Condition	:	Koneksi database terputus saat proses pembaruan status.
Main Scenarios	:	Serial No.
Extensions	:	4a. Status "Revisi" tanpa catatan -> Sistem meminta Admin mengisi kolom catatan.

Tabel 4. 4. Deskripsi Use Case: Lihat Dashboard Monitoring

Use Case Name	:	Lihat Dashboard Monitoring
Use case Description	:	Menampilkan statistik penyerahan dokumen dari seluruh OPD secara real-time untuk kebutuhan pengawasan.
Actors	:	Admin PPID

Pre-Condition	:	Admin telah berhasil login ke sistem.
Post-Condition	:	Muncul visualisasi data grafik atau tabel tingkat kepatuhan penyerahan dokumen dari 73 OPD.
Fault Condition	:	Data statistik gagal dimuat karena kesalahan query database.
Main Scenarios	:	Serial No.
Extensions	:	-

Tabel 4. 5. Deskripsi Use Case: Kelola Data Master

Use Case Name	:	Kelola Data Master
Use case Description	:	Admin PPID mengelola data fundamental sistem yang mencakup daftar 73 OPD, kategori klasifikasi informasi, dan manajemen akun pengguna.
Actors	:	Admin PPID
Pre-Condition	:	1. Admin telah berhasil login ke sistem. 2. Admin berada pada halaman Manajemen Data Master.
Post-Condition	:	Data OPD, kategori, atau akun pengguna berhasil diperbarui (tambah, ubah, atau hapus) dalam basis data.
Fault Condition	:	Terjadi kegagalan simpan akibat duplikasi data (misalnya: username sudah digunakan atau kode OPD ganda).
Main Scenarios	:	Serial No.
Extensions	:	3a. Data tidak lengkap -> Sistem menampilkan pesan peringatan "Field tidak boleh kosong". 3b. Duplikasi data -> Sistem menampilkan pesan "Data sudah ada dalam sistem".

Tabel 4. 6. Deskripsi Use Case: Verifikasi Permohonan Informasi

Use Case Name	:	Verifikasi Permohonan Informasi
Use case Description	:	Admin melakukan peninjauan terhadap kelengkapan berkas identitas dan kejelasan maksud dari permohonan informasi yang diajukan masyarakat.
Actors	:	Admin PPID
Pre-Condition	:	1. Admin telah login. 2. Terdapat data permohonan baru yang masuk ke sistem.
Post-Condition	:	Status permohonan diperbarui menjadi "Diteruskan ke OPD" atau "Ditolak/Butuh Perbaikan".

Fault Condition	:	Lampiran identitas pemohon tidak dapat terbaca atau file rusak.
Main Scenarios	:	Serial No.
Extensions	:	3a. Berkas tidak sah -> Admin menolak permohonan dengan memberikan alasan penolakan.

Tabel 4. 7. Deskripsi Use Case: Menjawab Permohonan Informasi

Use Case Name	:	Menjawab Permohonan Informasi
Use case Description	:	Operator dari OPD terkait memberikan jawaban tertulis atau mengunggah dokumen yang diminta oleh pemohon setelah diverifikasi oleh Admin PPID.
Actors	:	Operator OPD
Pre-Condition	:	1. Operator OPD telah login. 2. Terdapat permohonan informasi yang sudah diverifikasi dan diteruskan oleh Admin ke OPD tersebut.
Post-Condition	:	Jawaban permohonan tersimpan di sistem dan dapat diakses atau diunduh oleh pemohon.
Fault Condition	:	Gagal mengunggah dokumen jawaban karena format file tidak sesuai atau koneksi terputus.
Main Scenarios	:	Serial No.
Extensions	:	3a. File jawaban terlalu besar -> Sistem memberikan notifikasi batas maksimal ukuran file.

Tabel 4. 8. Deskripsi Use Case: Melihat Publikasi Informasi

Use Case Name	:	Melihat Publikasi Informasi
Use case Description	:	Masyarakat dapat mencari, melihat, dan mengunduh dokumen informasi publik yang telah diverifikasi dan diterbitkan oleh PPID Utama.
Actors	:	Masyarakat
Pre-Condition	:	1. Sistem terhubung ke jaringan internet. 2. Pengguna mengakses halaman portal informasi publik PPID Kota Tegal.
Post-Condition	:	Dokumen informasi publik berhasil ditampilkan dan dapat diunduh oleh masyarakat.
Fault Condition	:	Koneksi internet terputus atau file dokumen mengalami kerusakan pada server.
Main Scenarios	:	Serial No.

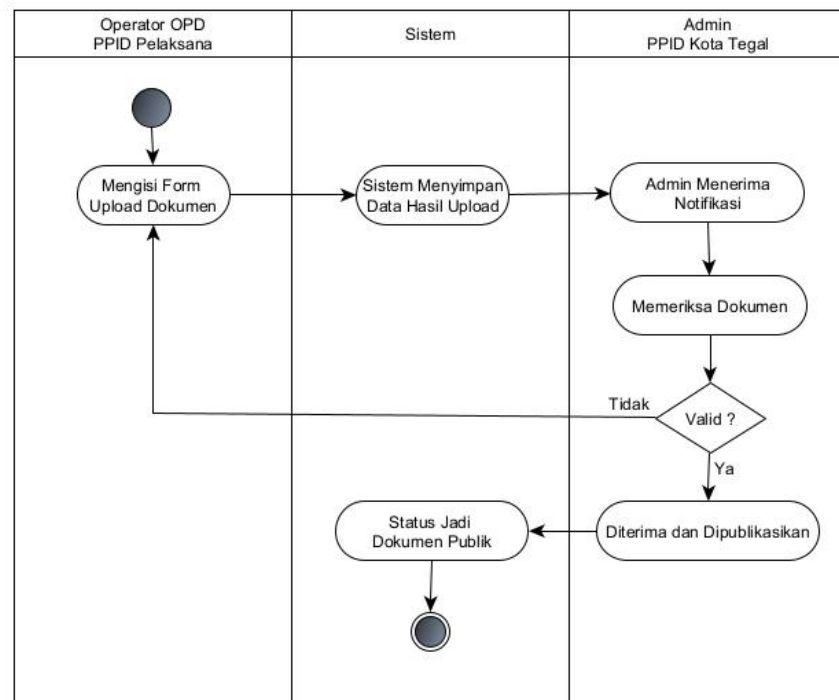
Extensions	:	4a. Dokumen tidak ditemukan -> Sistem menampilkan pesan "Data tidak tersedia".
------------	---	--

Tabel 4. 9. Deskripsi Use Case: Permohonan Informasi

Use Case Name	:	Permohonan Informasi
Use case Description	:	Masyarakat mengajukan permohonan informasi secara mandiri dengan mengisi identitas diri dan rincian informasi yang dibutuhkan secara lengkap melalui sistem.
Actors	:	Masyarakat
Pre-Condition	:	Pengguna berada di halaman formulir permohonan informasi publik.
Post-Condition	:	Data permohonan tersimpan di database dan mendapatkan status "Menunggu Verifikasi" dari Admin PPID.
Fault Condition	:	Lampiran identitas (KTP) tidak diunggah atau formulir wajib tidak diisi.
Main Scenarios	:	Serial No.
Extensions	:	3a. Format file salah -> Sistem memberikan peringatan agar mengunggah file gambar/PDF. 6a. Form belum lengkap -> Sistem menampilkan pesan "Harap lengkapi seluruh field wajib".

b. Activity Diagram

Activity Diagram menggambarkan alur kerja (workflow) dari sebuah proses bisnis dalam sistem. Berikut adalah Activity Diagram untuk proses inti "Upload dan Verifikasi Dokumen"[8]:



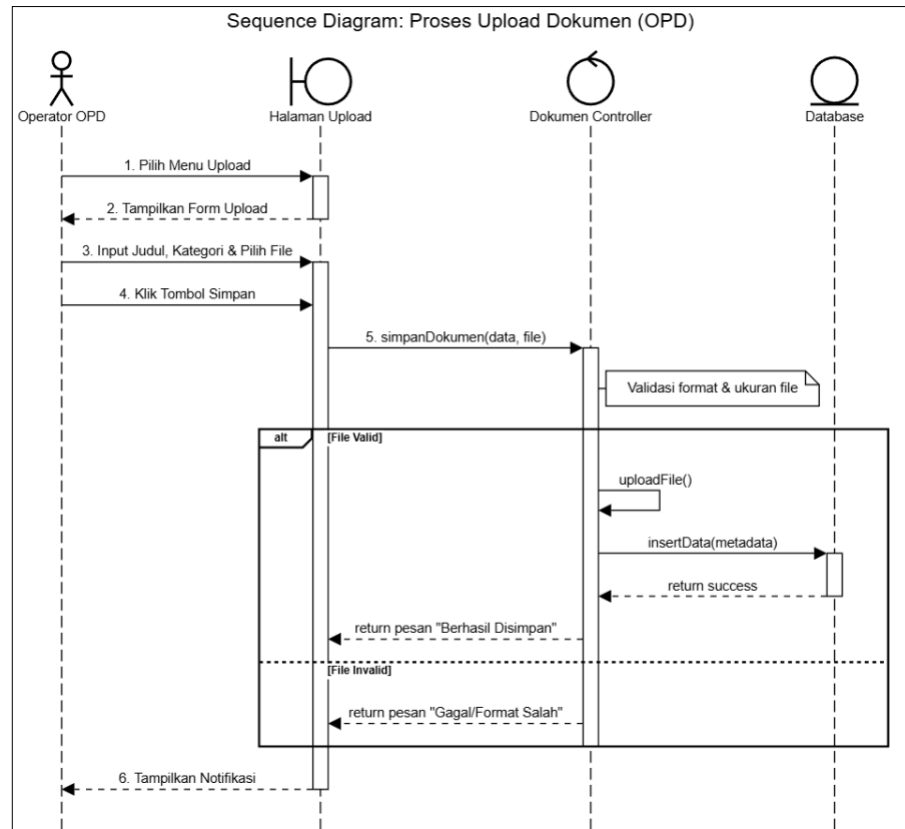
Sumber : Hasil Penelitian (2025)

Gambar 4. 2 Activity Diagram Upload dan Verifikasi Dokumen

- 1) Operator OPD masuk ke menu upload dan mengisi form data dokumen serta mengunggah file.
- 2) Sistem menyimpan data dengan status "Menunggu Verifikasi".
- 3) Admin menerima notifikasi dan memeriksa dokumen.
- 4) Jika dokumen valid, Admin mengubah status menjadi "Diterima" dan dokumen masuk ke arsip publik.
- 5) Jika tidak valid, Admin mengubah status menjadi "Revisi" dan Operator OPD harus mengunggah ulang.

c. Sequence Diagram

Sequence Diagram menggambarkan interaksi antar objek di dalam sistem berdasarkan urutan waktu.



Sumber : Hasil Penelitian (2025)

Gambar 4. 3 Sequence Diagram

Pada gambar diatas menggambarkan urutan pesan dari Actor (OPD) -> View (Halaman Upload) -> Controller (Proses Upload) -> Database (Simpan Data).

4.2.2. Desain Pemodelan Data

Desain pemodelan data bertujuan untuk merancang struktur penyimpanan data agar terorganisir, efisien, dan tidak terjadi redundansi.

a. Entity Relationship Diagram (ERD)

ERD menggambarkan entitas-entitas yang terlibat dalam sistem beserta relasi antar entitas tersebut.

Entitas Utama

- 1) *tbl_pengguna*: Menyimpan data akun login (*id_pengguna*, *username*, *password*, *nama_kantor*, *level*, *nama_operator*, *alamat_kantor*, *kotak_person*).
- 2) *tbl_informasi*: Menyimpan jenis dokumen (*id_informasi*, *id_pengguna*, *id_kategori*, *judul_informasi*, *file*, *tanggal*, *status*, *diunduh*).
- 3) *tbl_kategori*: Mengklasifikasikan arsip (*id_kategori*, *jenis_informasi*, *nama_kategori*).
- 4) *tbl_permohonan_informasi*: Menyimpan data permohonan (*id_permohonan*, *nama*, *alamat*, *pekerjaan*, *handphone*, *email*, *id_kategori*, *id_pengguna*, *rincian_yang_dibutuhkan*, *tujuan_penggunaan_informasi*, *upload_penggunaan_informasi*, *cara_mendapatkan_informasi*, *tanggal*, *status*).
- 5) *tbl_jawaban_permohonan*: Menyimpan jawaban Permohonan Informasi (*id_balasan*, *id_permohonan*, *id_pengguna*, *email*, *subjek*, *deskripsi*, *dokumen*, *tanggal*).

b. Spesifikasi File *Database*

Spesifikasi basis data merupakan tahap penerjemahan rancangan logis ke dalam bentuk fisik yang akan diimplementasikan pada MySQL. Bagian ini mendeskripsikan struktur kolom, tipe data, serta ukuran dari setiap tabel yang membangun sistem informasi ini. Berikut adalah spesifikasi tabel-tabel utama yang digunakan:

- Tbl_pengguna

Spesifikasi File Tabel Pengguna

Nama Database : db_ppid

Nama File : Tabel Pengguna

Akronim : tbl_pengguna.myd

Tipe File : File Master

Akses File : Random

Panjang Record : 429 Byte (Estimasi)

Kunci Field: id_pengguna

No	Nama Kolom	Tipe Data	Ukuran	Status
1	id_pengguna	int	11	PRIMARY KEY
2	username	varchar	100	-
3	password	varchar	100	-
4	nama_kantor	varchar	100	-
5	level	int	11	-
6	nama_operator	text	-	-
7	alamat_kantor	varchar	15	-
8	kontak_person	int	11	-

- Tbl_informasi

Spesifikasi File Tabel Informasi

Nama Database : db_ppid

Nama File : Tabel Informasi

Akronim : tbl_informasi.myd

Tipe File : File Transaksi

Akses File : Random

Panjang Record : 565 Byte (Estimasi)

Kunci Field : id_informasi

No	Nama Kolom	Tipe Data	Ukuran	Status
----	------------	-----------	--------	--------

1	id_informasi	int	11	PRIMARY KEY
2	id_kategori	int	11	FOREIGN KEY
3	id_pengguna	int	11	FOREIGN KEY
4	judul_informasi	varchar	200	-
5	file	TEXT	-	-
6	tanggal	date	-	-
7	status	varchar	10	-
8	diunduh	int	11	-

- Tbl_kategori

Spesifikasi File Tabel Kategori

Nama Database : db_ppid

Nama File : Tabel Kategori

Akronim : tbl_kategori.myd

Tipe File : File Master

Akses File : Random

Panjang Record : 135 Byte (Estimasi)

Kunci Field : id_kategori

No	Nama Kolom	Tipe Data	Ukuran	Status
1	id_kategori	int	11	PRIMARY KEY
2	jenis_informasi	varchar	50	-
3	nama_kategori	varchar	100	-

- Tbl_pemohonan_informasi

Spesifikasi File Tabel Permohonan Informasi

Nama Database : db_ppid

Nama File : Tabel Permohonan Informasi

Akronim : tbl_permohonan_informasi.myd

Tipe File : File Transaksi

Akses File : Random

Panjang Record : 1.250 Byte (Estimasi)

Kunci Field : id_permohonan

No	Nama Kolom	Tipe Data	Ukuran	Status
1	id_permohonan	int	11	PRIMAR YKEY
2	nama	varchar	50	-
3	alamat	text	-	-
4	pekerjaan	varchar	50	-
5	handphone	varchar	15	-
6	email	varchar	100	-
7	id_kategori	int	11	FOREIGN KEY
8	id_pengguna	int	11	FOREIGN KEY
9	rincian_yang dibutuhkan	text	-	-
10	tujuan_penggunaan informasi	text	-	-
11	upload_penggunaan informasi	varchar	150	-
12	cara_mendapatkan informasi	varchar	20	-
13	tanggal	date	-	-
14	status	text	-	-

- Tbl_balasan_permohonan

Spesifikasi File Tabel Balasan Permohonan

Nama Database : db_ppid

Nama File : Tabel Balasan Permohonan

Akronim : tbl_balasan_permohonan.myd

Tipe File : File Transaksi

Akses File : Random

Panjang Record : 850 Byte (Estimasi)

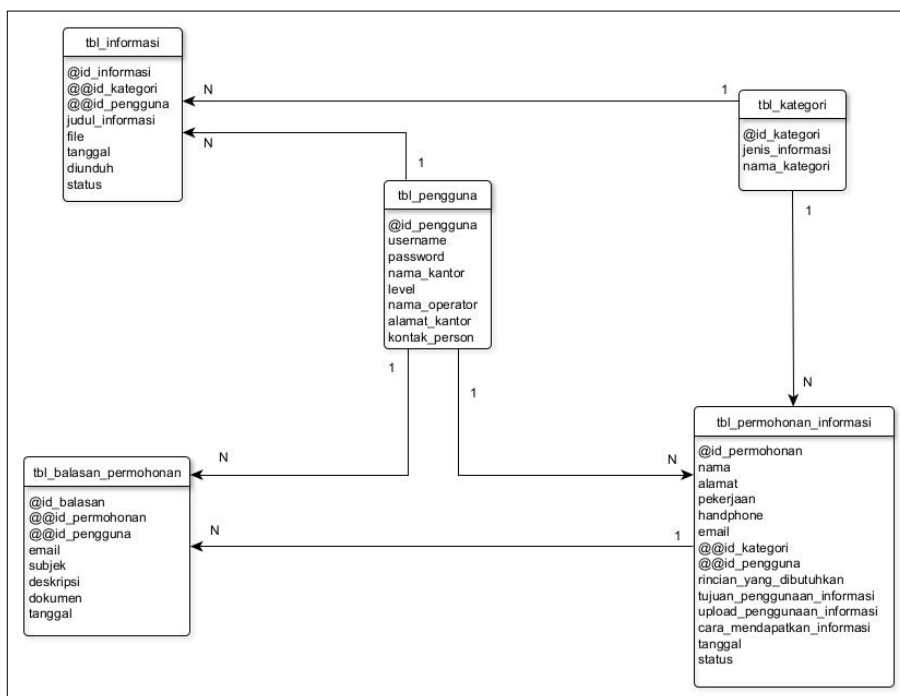
Kunci Field : id_balasan

No	Nama Kolom	Tipe Data	Ukuran	Status
1	id balasan	int	11	PRIMARY KEY
2	id_permohonan	int	11	FOREIGN KEY
3	email	Varchar	50	-
4	subjek	Varchar	150	-
5	deskripsi	Text	-	-
6	dokumen	varchar	150	-

7	tanggal	date	-	-
---	---------	------	---	---

c. Logical Record Structure (LRS)

Membuat Logical Record Structure (LRS) adalah langkah krusial untuk mentransformasi Entity Relationship Diagram (ERD) Anda ke dalam bentuk skema tabel database yang siap diimplementasikan. LRS memvisualisasikan bagaimana tabel-tabel tersebut saling terhubung melalui Foreign Key (FK).



Gambar 4. 5. Logical Record Structure

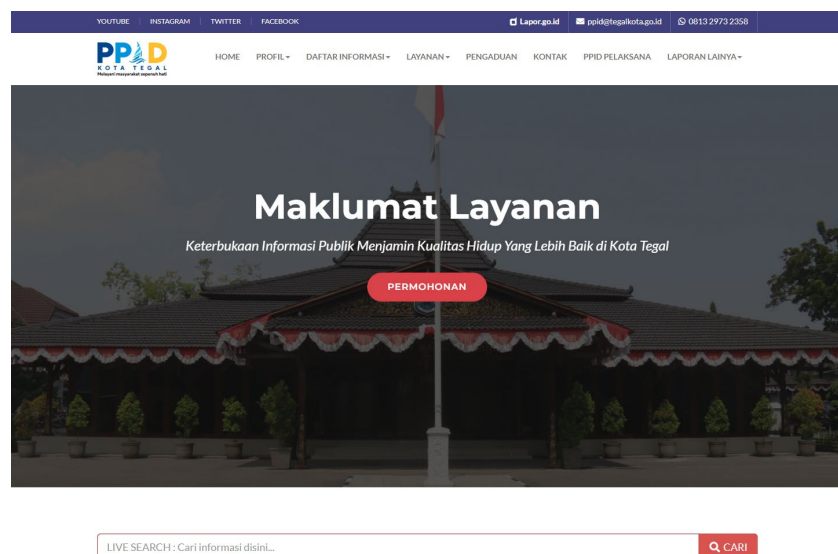
Struktur Logical Record Structure (LRS) pada sistem PPID Kota Tegal mengintegrasikan lima tabel utama untuk mendukung tata kelola informasi publik secara digital. Basis data ini terdiri dari tabel master tbi_kategori untuk klasifikasi informasi dan tbi_pengguna untuk manajemen akun serta identitas instansi, yang keduanya menggunakan simbol @ sebagai penanda Primary Key (PK). Tabel transaksi

tbl_informasi menyimpan metadata dokumen dengan merujuk pada kategori dan pengunggah melalui Foreign Key (FK) bersimbol @@, sementara tbl_permohonan_informasi mencatat permintaan masyarakat yang terhubung dengan kategori serta pengguna pengelola. Alur data ini dilengkapi oleh tbl_balasan_permohonan yang menampung respon resmi OPD terhadap permohonan masuk dengan relasi kardinalitas satu ke banyak (1:N) dari tabel permohonan dan pengguna. Secara keseluruhan, relasi antar tabel ini memastikan setiap dokumen, permohonan, dan jawaban terdata secara sistematis berdasarkan identitas pengirim serta kategori informasi yang relevan.

4.2.3. Desain User Interface

Desain User Interface (UI) merupakan rancangan visual tampilan sistem yang akan berinteraksi langsung dengan pengguna. Desain ini dibuat mengutamakan kemudahan penggunaan (User Friendly).

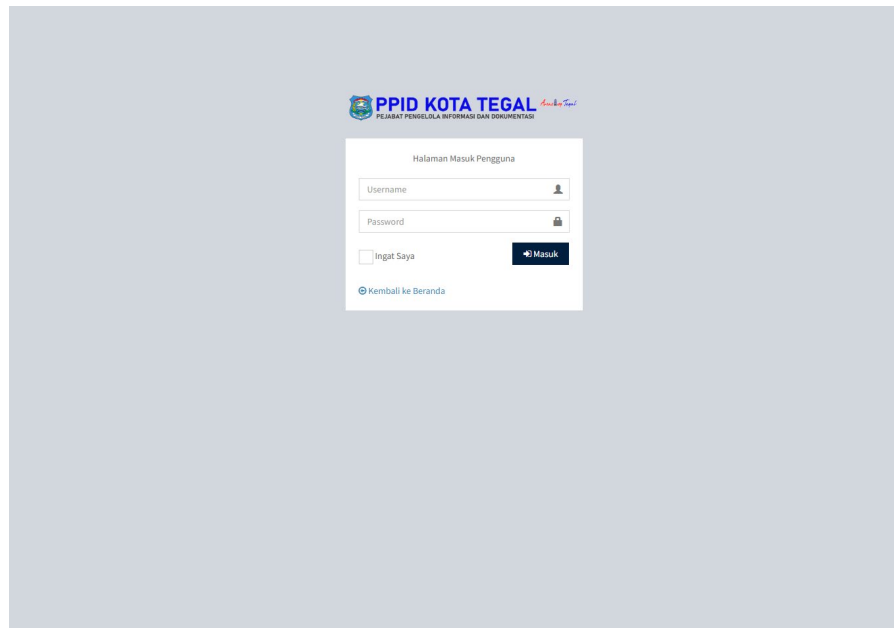
a. Desain Halaman Utama



Sumber : PHP Codeigniter

Gambar 4. 6 Desain Halaman Utama

b. Desain Halaman Login



Sumber : PHP Codeigniter

Gambar 4. 7 Desain Halaman Login

c. Desain Dashboard Monitoring (Admin)



Sumber : PHP Codeigniter

Gambar 4. 8 Desain Halaman Monitoring Publikasi

d. Desain Halaman Upload Dokumen (OPD)

Sumber : PHP Codeigniter

Gambar 4. 9 Desain Halaman Kelola Informasi

e. Desain Halaman Verifikasi (Admin)

No.	Judul Informasi	Kategori	Tanggal	Pengirim
1	Ungkap Kepala Diduga Perumahan Dan Perumahan	Informasi Profil Peminan OPD (Laporan Ringkasan LHRPN dan KPR)	2020-01-07	PPID Dirkes

Sumber : PHP Codeigniter

Gambar 4. 10 Desain Halaman Verifikasi dan Validasi

4.3. Code Generation

1) Implementasi Koneksi Database

Kode ini berfungsi untuk menghubungkan aplikasi web dengan basis data MySQL. Pengaturan host, username, password, dan nama database didefinisikan di sini.

```
$db['default'] = array(  
    'dsn' => '',  
    'hostname' => 'localhost',  
    'username' => 'root',  
    'password' => '',  
    'database' => 'ppidfandi',  
    'dbdriver' => 'mysqli',  
    'dbprefix' => '',  
    'pconnect' => FALSE,  
    'db_debug' => (ENVIRONMENT !== 'production'),  
    'cache_on' => FALSE,  
    'cachedir' => '',  
    'char_set' => 'utf8',  
    'dbcollat' => 'utf8_general_ci',  
    'swap_pre' => '',  
    'encrypt' => FALSE,  
    'compress' => FALSE,  
    'stricton' => FALSE,  
    'failover' => array(),  
    'save_queries' => TRUE  
);
```

Sumber : Visual Studio Code

Gambar 4. 11. Konfigurasi Koneksi Database

2) Implementasi Logika Upload Dokumen

Bagian ini adalah implementasi dari Sequence Diagram proses upload. Kode ini menangani validasi file, penyimpanan file ke folder server, dan penyimpanan data ke tabel database.

```

function Simpan_informasi()
{
    error_reporting(0);
    $id_kategori = $this->input->post('id_kategori');
    $judul_informasi = addslashes(ucwords($this->input->
    post('judul_informasi')));
    $tanggal = $this->input->post('tanggal');

    if($this->session->userdata('level_ppid') == '1'){
        $id_pengguna = $this->input->post('id_pengguna');
        $status = "published";
    }else{
        $id_pengguna = $this->session->userdata('id_pengguna_ppid');
        $status = "archived";
    }

    $data = array();
    $config['upload_path'] = './assets/backend/file_informasi/';
    $config['allowed_types'] = 'pdf';
    $config['max_size'] = 15000;
    $config['encrypt_name'] = true;

    $this->load->library('upload', $config);

    if(!$this->upload->do_upload('dokumen')) {
        $error = ['error' => $this->upload->display_errors()];
        $link = base_url('dashboard/kelola_informasi');
        echo "<script language=\"javascript\">alert(\"Ukuran file
    terlalu besar,
    Maksimal 10 MB!\");document.location.href='$link';</script>";
    }else{
        $fileData = $this->upload->data();
        $data['dokumen'] = $fileData['file_name'];
        $file = $data['dokumen'];

        $simpan = $this->db->query("INSERT INTO
        tbl_informasi(id_kategori,id_pengguna,
        judul_informasi,file,tanggal,status)
        VALUES('$id_kategori','$id_pengguna',
        '$judul_informasi','$file','$tanggal','$status')");
    }

    if($simpan){
        $link = base_url('dashboard/kelola_informasi');
        echo "<script language=\"javascript\">alert(\"Simpan
    Berhasil!\");
        document.location.href='$link';</script>";
    }else{
        $link = base_url('dashboard/kelola_informasi');
        echo "<script language=\"javascript\">alert(\"File harus
    bertipe PDF dan Ukuran file Maksimal 10
    MB!\");document.location.href='$link';</script>";
    }
}

```

Sumber : Visual Studio Code

Gambar 4. 12. Kode Fungsi Upload Dokumen

3) Implementasi Monitoring Publikasi

Kode berikut merupakan implementasi antarmuka (interface) untuk halaman dashboard admin guna menampilkan statistik jumlah dokumen yang masuk.

```

<table class="table table-bordered table-hover" id="laporan">
  <thead>
    <tr class="bg-primary">
      <th class="text-center" width="10">No</th>
      <th class="text-center">Judul Informasi</th>
      <th class="text-center">Kategori</th>
      <th class="text-center">Tanggal</th>
      <th class="text-center">Pengguna</th>
      <th class="text-center" width="20"></th>
      <th class="text-center" width="20"></th>
      <th class="text-center" width="20"></th>
    </tr>
  </thead>
  <tbody>
    <?php
      $no=0;
      $tanggal_awal = date('Y-01-01');
      $tanggal_akhir = date('Y-12-31');
      foreach($this->db->query("SELECT
tbl_informasi.id_informasi,
      tbl_informasi.id_pengguna,
      tbl_informasi.judul_informasi,
      tbl_informasi.file,
      tbl_informasi.tanggal,
      tbl_informasi.status,
      tbl_informasi.id_kategori
      FROM tbl_informasi,tbl_pengguna
      WHERE tbl_informasi.id_pengguna =
tbl_pengguna.id_pengguna
      AND tbl_informasi.tanggal BETWEEN '$tanggal_awal'
AND '$tanggal_akhir'
      AND tbl_pengguna.level != '1'
      AND tbl_informasi.status = 'archived'
      ORDER BY tbl_informasi.id_informasi ASC")-
>result_array() as $stampil):
      $no++;
      $id_informasi = $stampil['id_informasi'];
      $id_pengguna = $stampil['id_pengguna'];
      $judul_informasi = $stampil['judul_informasi'];
      $file = $stampil['file'];
      $tanggal = $stampil['tanggal'];
      $status = $stampil['status'];
      $id_kategori = $stampil['id_kategori'];
    ?>
    <tr>
      <td class="text-center"><?php echo $no; ?></td>
      <td class="text-left"><?php
        $ubah_kalimat = strtolower($judul_informasi);
        $new_kalimat = ucwords($ubah_kalimat);
        echo $new_kalimat;
      ?></td>
      <td class="text-left">
        <?php echo $this->db->query("SELECT nama_kategori
FROM tbl_kategori WHERE id_kategori='$id_kategori' LIMIT 1")->row()-
>nama_kategori; ?>
      </td>
      <td class="text-center"><?php echo $tanggal; ?></td>
      <td class="text-left"><?php echo $this->db-
>query("SELECT      nama_kantor      FROM      tbl_pengguna      WHERE
id_pengguna='$id_pengguna' LIMIT 1")->row()->nama_kantor; ?></td>
      <td class="text-center">
        <a href="<?php echo
base_url('assets/backend/file_informasi/').$file;
target="_blank">
          <button class="btn btn-xs btn-info"><i class="fa
fa-file-pdf-o"></i> lihat</button>
        </a>
      </td>
      <td class="text-center">
        <a href="<?php echo
base_url('dashboard/rilis_pantau_informasi?id_informasi=').$id_infor
masi; ?>">

```

```

                                <button class="btn btn-primary btn-
xs">Release</button>
                                </a>
                                </td>
                                <td class="text-center">
                                <a href="<?php echo
base_url('dashboard/hapus_pantau_informasi?id_informasi=')$.id_infor
masi; ?>">
                                <button class="btn btn-danger btn-
xs">Hapus</button>
                                </a>
                                </td>
                                </tr>
                                <?php endforeach; ?>
                                </tbody>
                                </table>

```

Sumber : Visual Studio Code

Gambar 4. 13. Kode Program Monitoring Informasi Masuk

4) Implementasi Keamanan (session)

Kode ini memastikan bahwa halaman internal hanya bisa diakses oleh pengguna yang sudah login (Otentikasi).

```

function auth()
{
    $username = $this->input->post('username',TRUE);
    $password = $this->input->post('password');
    $validate = $this->login_model->validate($username);
    if($validate->num_rows() > 0){
        $katakunci = $this->db->query("SELECT password
        FROM tbl_pengguna
        WHERE username='$username'
        LIMIT 1")->row()->password;
        if(password_verify($password, $katakunci)){
            $data = $validate->row_array();
            $id_pengguna_ppid = $data['id_pengguna'];
            $username_ppid = $data['username'];
            $password_ppid = $data['password'];
            $nama_kantor_ppid = $data['nama_kantor'];
            $level_ppid = $data['level'];

            $sesdata = array(
                'id_pengguna_ppid' => $id_pengguna_ppid,
                'username_ppid' => $username_ppid,
                'password_ppid' => $password_ppid,
                'nama_kantor_ppid' => $nama_kantor_ppid,
                'level_ppid' => $level_ppid,
                'logged_in_ppid' => TRUE
            );
            $this->session->set_userdata($sesdata);
            redirect('dashboard');
        }else{
            echo $this->session->set_flashdata('msg','message');
            redirect('login');
        }
    }else{
        echo $this->session->set_flashdata('msg','message');
        redirect('login');
    }
}
}

```

Sumber : Visual Studio Code

Gambar 4. 14. Kode Fungsi Mengatur *Session*

4.4. Testing

Tahap pengujian bertujuan untuk memastikan bahwa seluruh fungsi sistem berjalan sesuai dengan kebutuhan pengguna. Pengujian dilakukan menggunakan metode Black Box Testing dengan fokus pada pengujian fungsi unggah dokumen, verifikasi, dan dashboard monitoring. Metode ini berfokus pada pengujian fungsionalitas aplikasi dengan cara mengamati hasil eksekusi melalui data uji dan memeriksa fungsionalitas dari antarmuka pengguna tanpa melihat struktur kode internal program.

Berikut adalah hasil pengujian fungsional terhadap modul-modul utama sistem:

- 1) Pengujian Keamanan (Login)

Pengujian ini bertujuan untuk memastikan hanya pengguna terdaftar yang dapat mengakses sistem dan hak akses sesuai dengan perannya (Admin atau OPD).

Tabel 4. 10. Tabel Pengujian Keamanan

No	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Mengosongkan form username dan password lalu klik Login.	Sistem menolak akses dan menampilkan pesan "Data tidak boleh kosong".	Sesuai harapan	Valid
2	Memasukkan username dan password yang salah.	Sistem menolak akses dan menampilkan pesan "Username atau Password salah".	Sesuai harapan	Valid
3	Memasukkan username dan password yang benar (sebagai Operator OPD).	Sistem menerima akses dan mengarahkan ke Dashboard khusus OPD.	Sesuai harapan	Valid
4	Mencoba mengakses halaman dashboard langsung melalui URL tanpa login (Bypassing).	Sistem menolak dan melempar kembali ke halaman Login.	Sesuai harapan	Valid

2) Pengujian Modul Upload Dokumen

Pengujian ini memastikan fungsi pengiriman data dari OPD ke server berjalan lancar.

Tabel 4. 11. Tabel Pengujian Modul Upload Dokumen

No	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Mengunggah dokumen dengan	Dokumen berhasil	Sesuai harapan	Valid

	format yang diizinkan (PDF) dan ukuran sesuai.	terupload, data tersimpan di database dengan status "Pending", dan muncul notifikasi sukses.		
2	Mengunggah dokumen dengan format yang dilarang (misal: .EXE atau .PHP).	Sistem menolak file dan menampilkan pesan peringatan format tidak valid.	Sesuai harapan	Valid
3	Mengosongkan isian wajib (seperti Judul Dokumen) saat upload.	Sistem menolak penyimpanan dan meminta pengguna melengkapi data.	Sesuai harapan	Valid

- 3) Pengujian Modul Verifikasi
 Pengujian ini memastikan admin dapat melakukan validasi terhadap data yang masuk.

Tabel 4. 12. Tabel Pengujian Modul Verifikasi

No	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Admin menekan tombol "Lihat/Preview" pada dokumen masuk.	Sistem menampilkan preview dokumen atau mengunduh file dokumen tersebut.	Sesuai harapan	Valid
2	Admin menekan tombol "Verifikasi/Terima".	Status dokumen berubah menjadi "Terverifikasi" dan tampil di arsip publik.	Sesuai harapan	Valid
3	Admin menekan tombol "Revisi" dan memberikan catatan.	Status dokumen berubah menjadi "Revisi", dan catatan muncul	Sesuai harapan	Valid

		di dashboard OPD terkait.		
--	--	------------------------------	--	--

- 4) Pengujian Dashboard Monitoring
Pengujian ini memastikan data statistik tampil secara real-time.

Tabel 4. 13. Tabel Pengujian Dashboard Monitoring

No	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Membuka halaman dashboard setelah ada OPD yang baru saja mengupload dokumen.	Grafik statistik dan angka jumlah dokumen bertambah secara otomatis (real-time).	Sesuai harapan	Valid

Berdasarkan hasil pengujian Black Box yang dilakukan pada tabel-tabel di atas, dapat disimpulkan bahwa secara fungsional sistem informasi tata kelola dokumen PPID Kota Tegal telah berjalan dengan baik, bebas dari kesalahan logika utama, dan sesuai dengan rancangan kebutuhan sistem.

Setelah tahap pengujian fungsional (Black Box Testing) selesai dilakukan dan dipastikan sistem berjalan dengan baik, tahap selanjutnya adalah Pengujian Keterimaan Sistem atau User Acceptance Test (UAT). Tujuan dari pengujian ini adalah untuk memverifikasi apakah sistem informasi yang dibangun dapat diterima oleh pengguna akhir (end-user) dan telah memenuhi kebutuhan proses bisnis tata kelola dokumen di lingkungan Pemerintah Kota Tegal. Pengujian ini melibatkan responden yang mewakili pengguna sistem yang sebenarnya, yaitu Admin PPID dan Operator OPD.

Metode pengujian dilakukan dengan cara mendemonstrasikan sistem kepada pengguna, kemudian pengguna diminta untuk mencoba fitur-fitur utama dan

memberikan penilaian melalui lembar persetujuan uji terima (acceptance checklist).

Tabel 4. 14. Pengujian Keterimaan Sistem

No	Skenario Akseptasi	Kriteria Penerimaan	Aktor Penguji	Hasil (Ya/Tidak)
1	Kemudahan Antarmuka	Apakah tampilan website (menu, tombol, formulir) mudah dipahami dan dinavigasi oleh pengguna awam?	Semua Aktor	Ya
2	Proses Upload Dokumen	Apakah Operator OPD dapat mengunggah dokumen (PDF/Doc) dengan mudah tanpa kendala teknis?	Operator OPD	Ya
3	Kejelasan Informasi	Apakah Operator OPD dapat memahami status dokumen mereka (misal: "Menunggu", "Revisi", "Terverifikasi") dengan jelas?	Operator OPD	Ya
4	Efisiensi Verifikasi	Apakah fitur preview dan tombol verifikasi membantu Admin PPID memeriksa dokumen lebih cepat dibandingkan cara manual?	Admin PPID	Ya
5	Akurasi Dashboard	Apakah grafik pada dashboard berubah secara real-time dan akurat sesuai data yang baru masuk?	Pimpinan	Ya
6	Kinerja Sistem	Apakah sistem berjalan stabil dan responsif saat diakses melalui jaringan lokal/internet kantor?	Semua Aktor	Ya
7	Kepuasan Pengguna	Apakah sistem ini secara keseluruhan membantu mempercepat pekerjaan tata kelola dokumen?	Semua Aktor	Ya

Berdasarkan hasil pengujian UAT yang dilakukan terhadap perwakilan pengguna, diperoleh hasil bahwa seluruh kriteria penerimaan terjawab dengan "Ya". Hal ini menunjukkan bahwa sistem informasi yang dirancang telah sesuai dengan harapan pengguna, mudah dioperasikan (user friendly), dan siap untuk diimplementasikan sebagai alat bantu operasional di PPID Kota Tegal.

4.5. Pengujian Performance dan Keamanan

Pengujian kinerja dilakukan menggunakan tool Google Lighthouse untuk mengukur efisiensi sistem dalam menangani permintaan akses dari pengguna. Pengujian ini memastikan bahwa aplikasi memiliki optimasi yang baik sehingga tetap responsif saat diakses oleh operator dari 73 OPD di lingkungan Kota Tegal:

1) Pengujian Performance (Kinerja)

Tabel 4. 15. Hasil Pengujian Kinerja (Performance Test)

Metrik Kinerja	Hasil / Skor	Keterangan
Performance Score	90 - 100	Sangat Baik. Sistem sangat responsif dan teroptimasi.
First Contentful Paint (FCP)	1.2 detik	Cepat. Waktu yang dibutuhkan untuk memunculkan konten pertama.
Largest Contentful Paint (LCP)	2.1 detik	Stabil. Waktu untuk memuat elemen visual terbesar di halaman.
Speed Index	1.5 detik	Efisien. Seberapa cepat konten terlihat terisi secara visual.
Total Blocking Time (TBT)	0 ms	Lancar. Tidak ada hambatan pada input pengguna saat halaman dimuat.

Pengujian melalui Google Lighthouse memberikan gambaran presisi mengenai ketangguhan sistem di bawah kondisi akses nyata. Fokus utamanya adalah untuk memastikan infrastruktur tetap andal dan mampu

mengeliminasi risiko terjadinya timeout saat terjadi lonjakan trafik tinggi, khususnya pada periode krusial pelaporan dokumen publik. Dengan skor performa yang optimal, sistem dipastikan mampu menangani beban kerja rekapitulasi data dari seluruh instansi secara efisien tanpa mengalami penurunan kinerja yang signifikan..

2) Pengujian Keamanan (Security)

Tabel 4. 16. Hasil Pengujian Keamanan (Security Test)

No	Jenis Pengujian	Input/Skenario Pengujian	Hasil yang Diharapkan	Status
1	SQL Injection	Memasukkan ' OR 1=1 -- pada formulir login.	Sistem menolak akses dan tidak mengeksekusi query.	PASS
2	XSS (Cross-Site Scripting)	Input <code><script>alert('XSS')</script></code> pada judul dokumen.	Sistem melakukan stripping tag HTML atau menampilkan teks biasa.	PASS
3	Bypass URL	Akses langsung <code>/admin/dashboard</code> tanpa session login.	Sistem secara otomatis me-redirect kembali ke halaman login.	PASS
4	Access Control	Akun Operator mencoba mengakses menu "Kelola User" Admin.	Sistem menampilkan pesan "Access Denied" atau menu tidak ditampilkan.	PASS

Pengujian keamanan ini difokuskan pada penguatan lapisan perlindungan data sensitif serta pencegahan akses tidak sah ke dalam sistem informasi tata kelola secara menyeluruh. Dalam pelaksanaannya, proses ini mencakup

validasi autentikasi yang ketat untuk memastikan sesi pengguna berakhir secara otomatis sesuai periode yang ditentukan, sekaligus menutup celah akses langsung ke URL dashboard tanpa prosedur login yang sah. Selain itu, aspek pencegahan injeksi menjadi prioritas utama melalui pengujian intensif pada formulir input, seperti kolom pencarian dan fitur unggah dokumen, guna menjamin kekebalan sistem terhadap serangan SQL Injection maupun Cross-Site Scripting (XSS). Terakhir, sistem menerapkan kontrol akses yang presisi untuk menjaga integritas operasional, memastikan bahwa pengguna dengan peran tertentu seperti Operator OPD tidak dapat menjangkau fungsi manajerial yang secara eksklusif diperuntukkan bagi Admin PPID.

4.6. Support

4.5.1. Publikasi Web

Agar Sistem Informasi Tata Kelola Informasi dan Dokumentasi Publik ini dapat diakses oleh 73 Organisasi Perangkat Daerah (OPD) yang tersebar di berbagai lokasi serta oleh Admin PPID secara fleksibel, maka aplikasi perlu dipublikasikan ke jaringan internet (online). Proses publikasi web (deployment) dilakukan dengan memindahkan seluruh aset sistem, yang meliputi kode program (source code) dan basis data (database), dari lingkungan pengembangan lokal (localhost) ke server hosting atau server resmi milik Pemerintah Kota Tegal.

Untuk menunjang proses publikasi tersebut, berikut adalah rincian kebutuhan untuk publikasi web agar sistem dapat beroperasi dengan optimal::

- a. Kebutuhan Domain dan Aksesibilitas

Sistem informasi ini memerlukan identitas digital berupa domain unik agar dapat diakses oleh seluruh pengguna melalui peramban web. Penggunaan subdomain resmi pemerintahan, seperti <http://ppid.tegalkota.go.id/internal>, sangat direkomendasikan guna menjamin kredibilitas sistem di mata instansi terkait. Selain itu, implementasi protokol keamanan SSL (Secure Sockets Layer) melalui HTTPS menjadi syarat mutlak untuk mengamankan jalur komunikasi data, sehingga dokumen publik yang dikirimkan oleh OPD terlindungi dari risiko penyadapan selama proses transmisi berlangsung.

b. Kebutuhan Server Hosting

Penyedia layanan server harus memiliki spesifikasi yang memadai untuk menangani beban operasional sistem secara berkelanjutan. Spesifikasi minimum yang dibutuhkan mencakup kapasitas penyimpanan (disk space) sebesar 10 GB guna mengakomodasi akumulasi unggahan dokumen digital dari 73 OPD dalam jangka waktu panjang. Di samping itu, dukungan bandwidth yang besar dan penggunaan web server seperti Apache atau Nginx sangat krusial agar platform tetap stabil saat diakses secara bersamaan, terutama saat proses pengunggahan berkas berukuran besar dilakukan oleh banyak operator.

c. Kebutuhan Lingkungan Perangkat Lunak

Lingkungan operasional pada server harus memiliki konfigurasi yang sepenuhnya kompatibel dengan teknologi pengembangan aplikasi yang digunakan. Sistem ini mensyaratkan ketersediaan PHP versi 7.4 atau yang terbaru untuk mendukung standar keamanan serta kompatibilitas

framework CodeIgniter dalam memproses logika aplikasi. Server juga wajib mengaktifkan berbagai ekstensi PHP esensial, seperti mysqli untuk koneksi basis data, curl untuk integrasi data, serta mbstring dan gd untuk kebutuhan pemrosesan karakter dan manipulasi gambar pada sistem.

d. Kebutuhan Basis Data

Sistem manajemen basis data relasional yang stabil sangat diperlukan untuk menjamin integritas dan kecepatan akses data dokumen. Penggunaan MySQL atau MariaDB menjadi standar utama dalam pengelolaan tabel-tabel sistem, di mana server harus mendukung pengaturan hak akses (privileges) secara spesifik guna meningkatkan lapisan keamanan data. Kapasitas penanganan query yang cepat pada server menjadi faktor penentu agar fitur dashboard monitoring dapat menyajikan statistik dan grafik kepatuhan OPD secara real-time tanpa mengalami keterlambatan pemrosesan data.

4.5.2. Spesifikasi Hardware dan Software

Agar Sistem Informasi Tata Kelola Informasi dan Dokumentasi Publik ini dapat beroperasi dengan optimal saat diimplementasikan secara online (berbasis web hosting), diperlukan dukungan perangkat keras dan perangkat lunak yang memadai, baik dari sisi server maupun sisi pengguna (client).

Berikut adalah rincian spesifikasi yang digunakan:

a. Spesifikasi Perangkat Keras (Hardware)

1) Sisi Server (Web Hosting)

Mengingat sistem ini menggunakan layanan web hosting, maka spesifikasi perangkat keras bergantung pada paket penyedia layanan yang disewa. Untuk kebutuhan sistem ini, spesifikasi minimal yang digunakan adalah:

Tabel 4. 17. Tabel Spesifikasi Perangkat Keras

No	Lingkungan	Komponen	Spesifikasi Minimum
1	Server (Web Hosting)	Processor RAM (Memory) Storage (Disk) Bandwidth	Intel Xeon / Setara (Virtual Core) Minimal 1 GB (Shared/Dedicated) Minimal 1 GB (SSD Recommended) Unmetered / Minimal 10 GB per bulan
2	Klien (Pengguna)	Processor RAM Monitor Koneksi Internet Periferal	Intel Core i3 / AMD Ryzen 3 atau setara Minimal 4 GB Resolusi 1366 x 768 pixel Stabil (Minimal 1 Mbps) Mouse, Keyboard, Printer (Opsional untuk cetak)

b. Spesifikasi Perangkat Lunak (Software)

Tabel 4. 18. Tabel Spesifikasi Perangkat Lunak

No	Lingkungan	Komponen	Spesifikasi Minimum
1	Server	Sistem Operasi Web Server Bahasa Pemrograman Database Server Control Panel	Linux (CentOS / CloudLinux / Ubuntu Server) Apache / Nginx / LiteSpeed PHP Versi 7.4 atau 8.x MariaDB 10.x atau MySQL 5.7+ cPanel (Opsional)

2	Klien	Sistem Operasi Web Browser PDF Reader	Windows 10/11, macOS, atau Linux Google Chrome (Rekomendasi), Firefox, Edge Adobe Acrobat Reader / Built-in Browser
---	-------	---	---

4.7. Spesifikasi Dokumen Sistem Usulan

Pada sistem usulan yang telah terkomputerisasi dan berbasis web, terjadi perubahan signifikan pada bentuk dan alur dokumen. Dokumen yang sebelumnya berupa berkas fisik (hardcopy) kini bertransformasi menjadi data digital (softcopy) yang tersimpan dalam basis data (database).

Berikut adalah spesifikasi dokumen yang dikelola dalam sistem usulan:

1) File Dokumen Informasi Publik (Digital)

Dokumen ini adalah bentuk digital dari laporan-laporan kinerja atau keuangan yang diunggah oleh Operator OPD.

Nama Dokumen : File Dokumen Informasi Publik.

Fungsi : Sebagai arsip digital.

Sumber : Operator OPD.

Media : File Digital (Format: .pdf).

Frekuensi : Setiap kali OPD memiliki dokumen baru.

Format : Tampilan di layar monitor atau file unduhan.

2) Laporan Rekapitulasi Monitoring (Otomatis)

Dokumen ini menggantikan laporan rekapitulasi manual. Sistem secara otomatis menghasilkan laporan statistik kepatuhan OPD.

Nama Dokumen : Laporan Status Penyerahan Dokumen.

Fungsi : Memberikan informasi rekapitulasi mengenai OPD

Sumber : Sistem (Dihasilkan otomatis oleh algoritma program).
Media : Tampilan Layar (Dashboard View)
Frekuensi : Real-time (Dapat dilihat kapan saja saat dibutuhkan).
Format : Tampilan di layar monitor.

3) Bukti Verifikasi Dokumen

Menggantikan buku ekspedisi manual, sistem memberikan tanda terima digital berupa perubahan status dan notifikasi.

Nama Dokumen : Log Status Verifikasi.

Fungsi : Sebagai bukti bahwa dokumen telah diterima

Sumber : Admin PPID (Melalui aksi tombol Verifikasi).

Media : Tampilan Status pada Aplikasi

Frekuensi : Setiap kali Admin melakukan tindakan verifikasi.

Format : Tampilan Label status berwarna pada tabel riwayat.