

TUGAS AKHIR

**RANCANG BANGUN APLIKASI PENGELOLAAN ARSIP (E-ARSIP)
PADA BAGIAN PERENCANAAN DAN PENGELOLAAN ASET DI
PERUMDA AIR MINUM TIRTA KHATULISTIWA KOTA PONTIANAK**



OLEH:

JIDAN

3202116088

PROGRAM STUDI D3 TEKNIK INFORMATIKA

JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI PONTIANAK

2024

HALAMAN PENGESAHAN

RANCANG BANGUN APLIKASI PENGELOLAAN ARSIP (E-ARSIP) PADA BAGIAN PERENCANAAN DAN PENGELOLAAN ASET DI PERUMDA AIR MINUM TIRTA KHATULISTIWA KOTA PONTIANAK

Oleh:

JIDAN
3202116088

Tugas akhir ini Telah diterima dan disahkan sebagai salah satu syarat untuk menyelesaikan Program Pendidikan Diploma 3 pada Program Studi Teknik Informatika Jurusan Teknik Elektro Politeknik Negeri Pontianak.

Disahkan oleh:

Ketua Jurusan
Teknik Elektro



Hasan, S.T., M.T.
NIP 197108201999031003

Koordinator Program Studi
D3 Teknik Informatika



Mariana Syamsudin, S.T., M.T., Ph.D.
NIP 197503142006042001

Mengetahui,

Direktur Politeknik Negeri Pontianak



Dr. H. Widodo PS, S.T., M.T.
NIP 197504242000031001

HALAMAN PERNYATAAN

**RANCANG BANGUN APLIKASI PENGELOLAAN ARSIP (E-ARSIP)
PADA BAGIAN PERENCANAAN DAN PENGELOLAAN ASET DI
PERUMDA AIR MINUM TIRTA KHATULISTIWA KOTA PONTIANAK**

Oleh:

**JIDAN
3202116088**

Dosen Pembimbing:



**Ferry Faisal, S.S.T., M.T.
NIP. 197302061995011001**

**Telah dipertahankan di depan penguji pada tanggal 28 Agustus 2024 dan
dinyatakan memenuhi syarat sebagai Laporan Tugas Akhir.**

Dosen Penguji:

Penguji I



**Budianingsih, S.T., M.T.
NIP. 198011022012122003**

Penguji II



**Muhammad Diponegoro, S.Kom., M.Cs.
NIP. 198702082019031005**

HALAMAN PERNYATAAN ORISINALITAS

Saya yang bertanda tangan dibawah ini :

Nama : Jidan

NIM : 3202116088

Jurusan / Program Studi : Teknik Elektro / Teknik Informatika

Judul Proposal : Rancang Bangun Aplikasi Pengelolaan Arsip (E-arsip) Pada Bagian Perencanaan Dan Pengelolaan Aset Di Perumda Air Minum Tirta Khatulistiwa Kota Pontianak.

Menyatakan dengan sebenar-benarnya bahwa penulisan Tugas Akhir ini berdasarkan hasil penelitian, pemikiran dan pemaparan asli dari saya sendiri, baik untuk naskah tugas akhir maupun kegiatan yang tercantum sebagai bagian dari Tugas Akhir ini. Jika terdapat karya orang lain, saya akan mencantumkan sumber yang jelas.

Demikian pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini dan sanksi lain sesuai dengan peraturan yang berlaku di Politeknik Negeri Pontianak.

Demikian pernyataan ini saya buat dalam keadaan sadar tanpa paksaan dari pihak manapun.

Pontianak, 28 Agustus 2024
Yang membuat pernyataan,



RIWAYAT HIDUP



Biodata Mahasiswa:

Nama Mahasiswa	: Jidan
NIM	: 3202116088
Tempat / Tanggal Lahir	: Seladu, 04 Agustus 2003
Jenis Kelamin	: Laki-Laki
Agama	: Islam
Alamat	: Dusun Seladu, Desa Sepadu, RT 01, RW 01, Kecamatan Semparuk, Kabupaten Sambas, No.rumah 39
No.Handphone	: 085392145155
Email	: jidan482003@gmail.com

ABSTRAK

Perumda Air Minum Tirta Khatulistiwa, sebagai BUMD yang bertugas mendistribusikan air bersih di Kota Pontianak. Bagian perencanaan dan aset memiliki banyak dokumen yang perlu diarsipkan, baik dalam bentuk fisik maupun digital. Namun, penerapan metode pengarsipan yang dilakukan masih belum bisa memberikan kemudahan dalam hal pencarian data karena dokumentasi yang tidak tersusun secara rapi. Banyaknya dokumen fisik mengakibatkan dokumen menumpuk, dan kurangnya tempat penyimpanan membuat dokumen berpotensi hilang dan rusak. Penyimpanan yang dilakukan secara digital masih disimpan pada folder lokal dalam setiap komputer, sehingga jika *User* (staf) cuti atau tidak hadir, staf lain akan kesulitan mencari dokumen pada komputer staf yang cuti tersebut.

Dari permasalahan ini, untuk meningkatkan efisiensi dan efektivitas pengelolaan arsip, penelitian ini bertujuan untuk merancang dan membangun aplikasi pengelolaan arsip (*e-arsip*) berbasis web yang dapat mempermudah proses pencarian dan penyimpanan data dokumen.

Aplikasi dirancang menggunakan metode *Prototype* dan dibangun dengan memanfaatkan *framework CodeIgniter 4*, *bootstrap*, *HTML*, *php*, *CSS*, dan *JavaScript*, diharapkan aplikasi *e-arsip* dapat membuat proses pengarsipan di Perumda Air Minum Tirta Khatulistiwa Kota Pontianak agar menjadi lebih efektif dan efisien.

Aplikasi *e-arsip* yang telah dibangun memiliki beberapa fitur utama, seperti membuat, membaca, mengubah, menghapus (*CRUD*) dokumen, mencari, menyortir, mengunduh, dan mencetak dokumen. Selain itu, aplikasi *e-arsip* juga memiliki fitur manajemen pengguna, seperti membuat, mengedit, dan menghapus *User*, yang sudah berfungsi dengan baik. Dari pengujian yang telah dilakukan dengan metode *Black Box testing*, fitur aplikasi *e-arsip* sudah cukup berhasil. Nilai rata-rata *System Usability Scale* (SUS) yang didapatkan sebesar 66.67%, aplikasi *e-arsip* berada pada Rasio Efektivitas yaitu, Cukup Efektif sesuai dengan standar yang digunakan.

Kata Kunci: Perumda Air Minum Tirta Khatulistiwa, *e-arsip*, *Prototype*, *CRUD*, *Black box*, SUS.

ABSTRACT

Perumda Air Minum Tirta Khatulistiwa, as a Regional-Owned Enterprise (BUMD), is responsible for distributing clean water in Pontianak City. The planning and asset department has numerous documents that need to be archived, both in physical and digital formats. However, the current archiving methods have not provided ease in data retrieval due to poorly organized documentation. The large volume of physical documents leads to document accumulation, and the lack of storage space increases the risk of documents being lost or damaged. Digital storage is still maintained in local folders on individual computers, so if a staff member (User) is on leave or absent, other staff members will have difficulty finding documents on the absent staff member's computer.

To address these issues and improve the efficiency and effectiveness of archive management, this research aims to design and develop a web-based archive management application (e-arsip) that can simplify the process of searching for and storing document data. The application is designed using the Prototype method and built using the CodeIgniter 4 framework, Bootstrap, HTML, PHP, CSS, and JavaScript. It is expected that the e-arsip application will make the archiving process at Perumda Air Minum Tirta Khatulistiwa in Pontianak City more effective and efficient.

The e-arsip application developed has several key features, such as creating, reading, updating, deleting (CRUD) documents, searching, sorting, downloading, and printing documents. Additionally, the e-arsip application includes user management features, such as creating, editing, and deleting users, which have been successfully implemented. From the testing conducted using the Black Box testing method, the e-arsip application features have been quite successful. The average System Usability Scale (SUS) score obtained is 66.67%, indicating that the e-arsip application is rated as Moderately Effective according to the standard used.

Keywords: *Perumda Air Minum Tirta Khatulistiwa, e-arsip, Prototype, CRUD, Black box, SUS.*

PRAKATA

Puji dan syukur penulis panjatkan kehadiran Allah *Subhanahu Wa ta'ala* karena atas berkat rahmat dan karunia-Nya sehingga Tugas Akhir “Rancang Bangun Aplikasi Pengelolaan Arsip (*E-arsip*) Pada Bagian Perencanaan Dan Pengelolaan Aset Di Perumda Air Minum Tirta Khatulistiwa Kota Pontianak” sebagai salah satu syarat untuk menyelesaikan program Diploma 3 Program Studi Teknik Informatika Jurusan Teknik Elektro di Politeknik Negeri Pontianak ini dapat terselesaikan. Pada penyusunan Laporan Tugas Akhir dari awal hingga selesainya laporan ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, penulis sangat berterima kasih kepada:

- 1) Bapak Mistur dan Ibu Murti selaku orang tua penulis yang selalu memberikan nasihat, dukungan dan doa selama penulis mengikuti perkuliahan sampai dengan proses menyelesaikan Tugas Akhir ini dengan lancar dan sesuai dengan jadwal yang telah direncanakan;
- 2) Bapak Dr. H. Widodo PS, S.T., M.T., selaku Direktur Politeknik Negeri Pontianak;
- 3) Bapak Hasan, S.T., M.T., selaku Ketua Jurusan Teknik Elektro sekaligus sebagai dosen Penguji 2 yang telah memberikan saran dan masukan dalam penyelesaian Tugas Akhir ini;
- 4) Ibu Mariana Syamsudin, S.T., M.T., Ph.D., selaku Koordinator Program Studi D3 Teknik Informatika;
- 5) Ibu Isneini, S.ST., M.T., selaku Kepala Bagian Perencanaan dan Pengelolaan Aset yang telah membantu memberikan informasi dan data penelitian yang dibutuhkan penulis;
- 6) Bapak Safri Adam, S.Kom., M.Kom., selaku Koordinator Tugas Akhir Politeknik Negeri Pontianak yang telah memberikan arahan dan semangat penulis dalam penyelesaian Tugas Akhir;
- 7) Bapak Ferry Faisal, S.S.T., M.T., selaku Dosen Pembimbing Tugas Akhir yang telah memberikan bimbingan dan pengarahan dalam melaksanakan Tugas Akhir maupun dalam menulis Laporan Tugas Akhir;
- 8) Bapak Aditya Pratama Putra, A.Md.T., selaku pembimbing di kantor Perumda Air Minum Tirta Khatulistiwa Kota Pontianak Bagian perencanaan dan

Pengelolaan Aset yang telah membantu memberikan informasi dan data penelitian yang dibutuhkan penulis;

- 9) Ibu Budianingsih, S.T., M.T., selaku Dosen Penguji 1 yang telah memberikan saran dan masukan dalam penyelesaian Tugas Akhir ini;
- 10) Bapak Muhammad Diponegoro, S.Kom., M.Cs., selaku Dosen Penguji 2 yang telah memberikan saran dan masukan dalam penyelesaian Tugas Akhir ini;
- 11) Seluruh staf pengajar Program studi D3 Teknik Informatika Politeknik Negeri Pontianak;
- 12) Teman-teman mahasiswa Program Studi D3 Teknik Informatika Jurusan Teknik Elektro khususnya kelas 6A yang sangat mendukung sekaligus memberikan semangat selama kegiatan perkuliahan sampai pada penghujung masa studi yang sama-sama berjuang menyelesaikan Pendidikan di kampus.

Pada penyusunan Laporan Tugas Akhir ini tentu masih terdapat banyak kekurangan. Oleh karena itu penulis mengharapkan saran dan masukan yang membangun serta menyempurnakan laporan ini dimasa yang akan datang. Akhir kata penulis berharap semoga laporan ini bermanfaat bagi para pembaca dan penulis pada khususnya.

DAFTAR ISI

HALAMAN PENGESAHAN.....	i
HALAMAN PERNYATAAN	ii
HALAMAN PERNYATAAN ORISINALITAS.....	iii
RIWAYAT HIDUP.....	iv
ABSTRAK	v
<i>ABSTRACT</i>	vi
PRAKATA.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xv
DAFTAR LAMPIRAN.....	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Metodologi Penelitian	3
1.7 Sistematika Penulisan.....	6
BAB II DASAR TEORI	7
2.1 Tinjauan Pustaka	7
2.2 Dasar Teori	7
2.2.1 Figma	7
2.2.2 Website	8
2.2.3 HTML (<i>HyperText Markup Language</i>)	8
2.2.4 CSS (<i>Cascading Style Sheet</i>)	8
2.2.5 PHP (<i>Hypertext Preprocessor</i>)	9
2.2.6 XAMPP.....	9
2.2.7 ERD (<i>Entity-Relation Diagram</i>)	9
2.2.8 CI (<i>CodeIgniter</i>).....	10
2.2.9 Bootstrap	11
BAB III RANCANGAN SISTEM.....	12

3.1.	Gambaran Umum aplikasi.....	12
3.2.	Tahapan pengumpulan kebutuhan.....	12
3.2.1.	Teknik Pengumpulan Kebutuhan.....	12
3.3.	Use Case Diagram	15
3.4.	Membangun <i>Prototype</i>	17
3.4.1.	<i>Mockup</i>	17
3.5.	Evaluasi <i>Prototype</i>	24
3.5.1.	Halaman <i>dashboard</i>	24
3.5.2.	Halaman Lihat Dokumen	25
3.6.	Mengkodekan sistem.....	27
3.6.1.	<i>Entity Relationship Diagram</i>	27
3.6.2.	Struktur Databases	28
3.6.3.	Struktur proyek	28
3.6.4.	Login	32
3.6.5.	<i>dashboard</i>	34
3.6.6.	Pengolahan Dokumen	34
3.6.7.	Jenis Dokumen	36
3.6.8.	Kategori Dokumen	36
3.6.9.	<i>Profile</i>	36
3.6.10.	Logout/Sign Out	37
BAB IV HASIL DAN PEMBAHASAN		41
4.1.	Hasil Implementasi Antarmuka Pengguna	41
4.1.1	Implementasi Beranda.....	41
4.1.2	Implementasi Login	41
4.1.3	Implementasi <i>dashboard</i> (<i>Admin</i> dan Staf).....	42
4.1.4	Implementasi Menu Dokumen Teknik, Dokumen <i>Administrasi</i> , dan Dokumen Pelayanan (<i>Admin</i> dan Staf)	43
4.1.5	Implementasi Tambah Kategori (<i>Admin</i> dan Staf).....	45
4.1.6	Implementasi Edit Kategori (<i>Admin</i> dan Staf)	46
4.1.7	Implementasi Hapus Kategori (<i>Admin</i> dan Staf)	46
4.1.8	Implementasi Tambah Dokumen (<i>Admin</i> dan Staf).....	47
4.1.9	Implementasi Lihat Dokumen (<i>Admin</i>).....	48
4.1.10	Implementasi Lihat Dokumen (Staf).....	49
4.1.11	Implementasi Cetak Dokumen (<i>Admin</i>).....	49
4.1.12	Implementasi Edit Dokumen (<i>Admin</i> dan Staf)	50
4.1.13	Implementasi <i>Download</i> Dokumen (<i>Admin</i>)	50

4.1.14	Implementasi Hapus Dokumen (<i>Admin</i>).....	51
4.1.15	Implementasi Menu <i>Profile</i> dan <i>Sign out</i> (<i>Admin</i> dan Staf)	51
4.1.16	Implementasi <i>Profile</i> (<i>Admin</i>).....	52
4.1.17	Implementasi Tambah Pengguna (<i>Admin</i>).....	53
4.1.18	Implementasi Ubah Data Diri (<i>Admin</i>)	53
4.1.19	Implementasi Edit Pengguna (<i>Admin</i>)	54
4.1.20	Implementasi <i>Profile</i> (Staf).....	55
4.1.21	Implementasi Ubah Data Diri (<i>Staf</i>)	56
4.1.22	Implementasi Ubah Gambar <i>Profile</i> (<i>Admin</i> dan staf).....	56
4.2.	Pengujian Aplikasi	57
BAB V KESIMPULAN DAN SARAN.....		68
5.1.	Kesimpulan.....	68
5.2.	Saran	68
DAFTAR PUSTAKA		69
LAMPIRAN.....		70

DAFTAR GAMBAR

Gambar 1. 1 Metodologi Prototype[3]	4
Gambar 3. 1 Use Case Diagram	16
Gambar 3. 2 Mockup Halaman Login.....	17
Gambar 3. 3 Mockup Halaman dashboard.....	18
Gambar 3. 4 Mockup Halaman Tambah Kategori Dokumen	18
Gambar 3. 5 Mockup Halaman Edit Kategori Dokumen.....	19
Gambar 3. 6 Mockup Form Tambah Dokumen	19
Gambar 3. 7 Mockup Halaman Lihat Dokumen(admin)	20
Gambar 3. 8 Mockup Halaman Lihat Dokumen(staf)	20
Gambar 3. 9 Mockup Form edit Dokumen	21
Gambar 3. 10 Mockup Halaman Profile Admin	21
Gambar 3. 11 Mockup Form Tambah Akun.....	22
Gambar 3. 12 Mockup Form edit Akun.....	22
Gambar 3. 13 Mockup Halaman Profile User.....	23
Gambar 3. 14 Mockup Form Ubah Data Diri (Admin dan User)	23
Gambar 3. 15 Mockup Form edit Gambar Profile (Admin dan User)	24
Gambar 3. 16 Mockup Halaman Dashboard Sebelum Dievaluasi.....	25
Gambar 3. 17 Mockup Halaman Dashboard Setelah Dievaluasi.....	25
Gambar 3. 18 Mockup Halaman Lihat Dokumen Sebelum Dievaluasi.....	26
Gambar 3. 19 Mockup Halaman Lihat Dokumen Setelah Dievaluasi.....	26
Gambar 3. 20 Entity Relationship Diagram	27
Gambar 3. 21 Struktur Databases.....	28
Gambar 3. 22 struktur Proyek CodeIgniter	29
Gambar 3. 23 Direktori app	29
Gambar 3. 24 Direktori app/config	29
Gambar 3. 25 Direktori app/Controllers	30
Gambar 3. 26 Direktori app/Models	30
Gambar 3. 27 Direktori app/views/backend/login	31
Gambar 3. 28 app/views/backend/template	31
Gambar 3. 29 Direktori Public	31
Gambar 3. 30 Direktori public/Admin/asset dan public/Admin/fileArsip.....	32

Gambar 3. 31 Implementasi code Library Validation rules.....	32
Gambar 3. 32 Implementasi code session library adding session data	33
Gambar 3. 33 Implementasi code pemeriksaan akses.....	33
Gambar 3. 34 Implementasi code Library Times and Dates.....	34
Gambar 3. 35 Form upload file	35
Gambar 3. 36 implementasi code validasi data.....	35
Gambar 3. 37 implementasi code validasi file	35
Gambar 3. 38 implementasi code penentuan lokasi file	36
Gambar 3. 39 implementasi Library Session destory().....	37
Gambar 4. 1 Implementasi Beranda.....	41
Gambar 4. 3 Implementasi Login	42
Gambar 4. 2 Pesan Kesalahan Setelah Gagal Login	42
Gambar 4. 4 Implementasi dashboard (Admin dan Staf).....	43
Gambar 4. 5 Menu Jenis Dokumen (Admin dan Staf).....	43
Gambar 4. 6 Tampilan Tabel Dokumen Dengan Jenis Teknik (Admin dan Staf)	44
Gambar 4. 7 Tampilan Tabel Dokumen Dengan Jenis Administrasi (Admin dan Staf)	44
Gambar 4. 8 Tampilan Tabel Dokumen Dengan Jenis Pelayanan (Admin dan Staf)	45
Gambar 4. 9 Menu Tambah Kategori Dokumen (Admin dan Staf).....	45
Gambar 4. 10 Implementasi Tambah Kategoti (Admin dan Staf)	46
Gambar 4. 11 Implementasi Edit Kategori (Admin dan Staf)	46
Gambar 4. 12 Implementasi Hapus Kategori (Admin dan Staf)	47
Gambar 4. 13 Pesan Gagal Menghapus Kategori (Admin dan Staf)	47
Gambar 4. 14 Implementasi Tambah Dokumen (Admin dan Staf)	48
Gambar 4. 15 Implementasi Lihat Dokumen (Admin)	48
Gambar 4. 16 Implementasi Lihat Dokumen (Staf).....	49
Gambar 4. 17 Implementasi Cetak Dokumen (Admin)	49
Gambar 4. 18 Implementasi Edit Dokumen (Admin dan Staf).....	50
Gambar 4. 19 Implementasi Download Dokumen (Admin dan Staf).....	50
Gambar 4. 20 Implementasi Hapus Dokumen (Admin dan Staf)	51
Gambar 4. 21 Implementasi Menu Profile dan Sing Out (Admin dan Staf).....	51

Gambar 4. 22 Implementasi Profile Admin	52
Gambar 4. 23 Implementasi Tambah Pengguna	53
Gambar 4. 24 Implementasi Ubah Data Diri Admin	54
Gambar 4. 25 Implementasi Edit Pengguna.....	55
Gambar 4. 26 Implementasi Profile Staf.....	55
Gambar 4. 27 Implementasi Ubah Data Diri Staf	56
Gambar 4. 28 implementasi Ubah Gambar Profile (Admin dan Staf)	57

DAFTAR TABEL

Tabel 3. 1 Identifikasi pengguna.....	13
Tabel 3. 2 User Story	13
Tabel 3. 3 Kebutuhan Fungsional dan Hak Akses	14
Tabel 3. 4 Kebutuhan Non Fungsional	14
Tabel 4. 1 Pengujian Aplikasi Dengan Black Box.....	57
Tabel 4. 2 pengujian System Usability Scale (SUS).....	67
Tabel 4. 3 hasil rata-rata presentase.	68

DAFTAR LAMPIRAN

Lampiran 1 Hasil Wawancara	71
Lampiran 2 Daftar Dokumen Yang Akan Diarsipkan	72
Lampiran 3 Pengujian System Usability Scale	73
Lampiran 4 Surat Ijin penelitian	74

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pengarsipan adalah kegiatan pengelolaan yang meliputi penciptaan, penerimaan, pengumpulan, pengaturan, pengendalian, pemeliharaan, perawatan, dan penyimpanan suatu dokumen menurut sistem tertentu sehingga saat diperlukan arsip dapat ditemukan dengan cepat dan tepat. Kearsipan merupakan salah satu peran penting dalam kegiatan suatu perusahaan. Yakni sebagai sumber *informasi* dan referensi suatu perusahaan[1]. Arsip data sering disamakan dengan backup data, yaitu salinan data. Backup data berfungsi sebagai mekanisme pemulihan data, yang memungkinkan untuk memperoleh kembali data dalam kasus data rusak atau hancur. Arsip ini melindungi informasi dokumen lama yang tidak diperlukan untuk operasi sehari-hari namun mungkin harus diakses sesekali. Arsip data menjadi salah satu cara untuk mengurangi konsumsi penyimpanan primer dan biaya terkait[2].

Perumda Air Minum Tirta Khatulistiwa Kota Pontianak sebagai tempat penelitian dipilih dengan memepertimbangkan beberapa faktor. Pertama, perusahaan ini merupakan tempat PKL penulis sebelumnya. Kedua, merupakan perusahaan daerah yang bergerak dalam bidang pengelolaan air minum di wilayah Pontianak, sehingga memiliki jumlah dan jenis dokumen yang signifikan untuk diarsipkan. Ketiga, khususnya dalam bagian perencanaan dan pengelolaan aset, terdapat tanggung jawab utama dalam pengelolaan aset perusahaan, Mereka melakukan pendataan dan pembaruan database aset, merencanakan pengembangan, perbaikan, dan rehabilitasi aset, serta menyusun program pemeliharaan dan penghapusan aset. Bagian ini juga bertugas mengawasi seluruh kegiatan proyek di lapangan dan memastikan penerapan prosedur keselamatan kerja (K3). Secara keseluruhan, Bagian Perencanaan dan Pengelolaan Aset memainkan peran penting dalam memastikan bahwa aset perusahaan dikelola secara efektif dan efisien. Bagian ini, Memiliki volume arsip yang cukup besar, baik dalam bentuk fisik maupun digital, yang perlu dikelola secara sistematis dan terstruktur.

Arsip memiliki peran di dalam kegiatan perusahaan, terutama dalam memastikan ketersediaan sumber informasi dan referensi yang diperlukan.

Kegiatan-kegiatan yang dilakukan seperti kegiatan proyek atau yang lainnya pasti memiliki dokumen, Dokumen-dokumen yang disimpan dalam arsip meliputi berbagai jenis, seperti dokumen teknik (misalnya Rencana Anggaran Biaya, Laporan Kegiatan Proyek, Ass Build Drawing), dokumen *Administrasi* (misalnya laporan audit), dan dokumen pelayanan (misalnya permohonan swadaya dan permohonan bantuan). Penerapan metode pengarsipan yang dilakukan pada saat ini masih belum bisa memberikan kemudahan karena penyimpanan yang tidak tersusun secara rapi dan menyebabkan kesulitan dalam pencarian data, jika akan digunakan dikemudian hari.

Pada bagian perencanaan dan pengelolaan aset. Pengelolaan arsip masih dilakukan secara manual oleh staf/staf bagian perencanaan dan pengelolaan aset. Selama ini, dokumen yang selesai dikerjakan disimpan di rak arsip. Namun, penyimpanan dokumen secara fisik telah menimbulkan beberapa masalah, seperti penumpukan dokumen mengakibatkan kurangnya ruang penyimpanan, berpotensi hilang dan rusak serta kesulitan dalam pencarian dokumen. Selain itu, penyimpanan digital yang hanya dilakukan pada folder lokal di setiap komputer staf juga menimbulkan kendala dalam aksesibilitas dokumen antar staf.

Berdasarkan permasalahan yang telah diuraikan, penulis bermaksud untuk membangun sebuah aplikasi yang digunakan untuk pengelolaan arsip (*e-arsip*) berbasis web. Dengan dibangunnya *e-arsip* pada Perumda Air Minum Tirta Khatulistiwa Kota Pontianak diharapkan di kemudian hari untuk proses pencarian dan penyimpanan data dapat dilakukan dengan lebih efektif dan efisien pada proses pengarsip.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka dapat dirumuskan masalah yaitu, Bagaimana merancang dan membangun Aplikasi Pengelolaan Arsip (*E-arsip*) Pada Bagian Perencanaan Dan Pengelolaan Aset Di Perumda Tirta Khatulistiwa Kota Pontianak berbasis *web*.

1.3 Batasan Masalah

Adapun batasan masalah dalam perancangan dan pembangunan aplikasi ini adalah sebagai berikut:

- 1) Aplikasi *e-arsip* berbasis *web* hanya berada dalam lingkup hanya di Perumda Air Minum Tirta Khatulistiwa Kota Pontianak.
- 2) *Framework* yang digunakan dalam *website* ini adalah *CodeIgniter 4*, *Bootstrap*.
- 3) Ada riwayat waktu *update* dokumen
- 4) Dokumen yang akan ditangani aplikasi untuk sementara adalah dokumen Teknik, *Administrasi*, dan *Pelayanan*
- 5) *Software* yang digunakan untuk databases adalah *XAMPP v3.3.0*, *MariaDB*, dan *SQL*.
- 6) Fitur aplikasi yang akan dibuat dapat digunakan untuk *create*, *read*, *update*, *delete* (CRUD) adapun fitur *searching*, *sorting* dan *printing* data atau dokumen dalam aplikasi
- 7) Pengguna aplikasi sebagai *Admin* adalah kepala bagian perencanaan dan pengelolaan aset serta staf bagian perencanaan dan pengelolaan aset sebagai *User*.
- 8) Dokumen yang disimpan pada aplikasi tidak bersifat rahasia.

1.4 Tujuan Penelitian

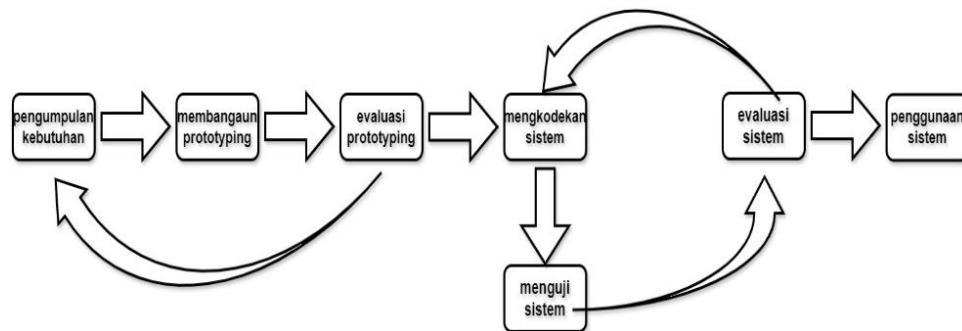
Tujuan dari penelitian ini adalah untuk menghasilkan aplikasi pengelolaan Arsip berbasis *Web* Pada Perumda Air Minum Kota Pontianak (Bagian Perencanaan Dan Pengelolaan Aset) yang diharapkan dapat membantu proses pencarian dan penyimpanan data lebih efektif dan efisien pada proses pengarsip.

1.5 Manfaat Penelitian

Manfaat bagi pengguna aplikasi khususnya pada bagian Perencanaan dan Pengelolaan Aset sebagai sarana untuk mempermudah dalam penyimpanan, pencarian dan melihat data dokumen di Perumda Air Minum Tirta Khatulistiwa Kota Pontianak. Dan membuat pekerjaan lebih efektif dan efisien.

1.6 Metodologi Penelitian

Karena penelitian tugas akhir ini bersifat eksperimental maka digunakanlah metode penelitian *Prototype*.



Gambar 1. 1 Metodologi Prototype[3]

Tahapan dalam metode *Prototype* adalah sebagai berikut :

1. Pengumpulan kebutuhan

Pengumpulan kebutuhan aplikasi dilakukan dengan mendengarkan keluhan dari pelanggan. Untuk membuat aplikasi yang sesuai kebutuhan, maka harus diketahui terlebih dahulu bagaimana sistem yang sedang berjalan untuk mengetahui masalah yang terjadi[3]. Pengumpulan kebutuhan dilakukan dengan beberapa metode yaitu:

- Metode Observasi, Metode ini menggunakan pengamatan langsung di kantor Perumda Air Minum Tirta Khatulistiwa Kota Pontianak, khususnya pada bagian perencanaan dan pengelolaan aset, untuk mendapatkan informasi tentang alur proses saat ini dilakukan dan informasi yang diperlukan untuk membangun aplikasi.
- Metode Wawancara, metode ini dilakukan dengan wawancara langsung dengan staf pengawas pada bagian perencanaan dan pengelolaan aset untuk mengumpulkan data yang dibutuhkan sesuai dengan studi kasus yang dipilih penulis. Melalui proses wawancara penulis akan menggali informasi mengenai kebutuhan sistem yang akan dibangun. Hasil wawancara ini akan menjadi landasan utama dalam pengembangan sistem yang memenuhi kebutuhan pengguna.
- Metode Studi literatur, metode ini adalah kegiatan yang dilakukan dengan metode pengumpulan data Pustaka, membaca dan mencatat serta mengolah

bahan penelitian. Metode ini mengumpulkan data penelitian dengan melakukan pencarian referensi dari sumber hasil penelitian sebelumnya yang relevan, seperti *e-book*, jurnal ilmiah, dan laporan penelitian tugas akhir yang digunakan sebagai bahan studi pustaka.

2. Merancang dan membangun *Prototype*

Pada tahapan ini, melakukan perancangan serta membangun *Prototype* sistem. yang dibuat disesuaikan dengan keluhan pelanggan atau pengguna yang telah diidentifikasi sebelumnya[3]. Perancangan dilakukan dengan menggunakan *tools* yang ada di figma. Seperti perancangan tampilan untuk *form login*, tampilan *dashboard*, *form input* dan tampilan lainnya sesuai dengan kebutuhan pengguna.

3. Uji coba dan evaluasi

Pada tahapan ini, *Prototype* sistem diuji coba oleh pelanggan atau pengguna. Setelah itu, kebutuhan pelanggan dievaluasi. Pengembang mempertimbangkan keluhan pelanggan dan memperbaiki *Prototype* sebelumnya setelah itu[3].

4. Mengkodekan sistem

Sebelum pengkodean, terlebih dahulu perlu memahami bahasa pemrograman yang akan digunakan. Pada tahap ini, penulis dapat merancang, membangun, dan mengaplikasikan aplikasi *web* atau aplikasi sesuai dengan persyaratan dalam bentuk kode program[3]. Pada saat melakukan pengkodean sistem *framework* untuk membangaun *front-and* dari aplikasi adalah bootsrap. Dan *framework* PHP yang digunakan adalah *CodeIgniter*.

5. Menguji sistem

Setelah pengkodean yang dilakukan selanjutnya adalah testing program. Banyak cara untuk menguji sistem, seperti menggunakan *white box* atau *black box*. Menggunakan *white box* berarti menguji kode programnya, sedangkan *Black Box* menguji fungsi dari tampilan apakah aplikasinya sudah benar atau tidak[3].

6. Evaluasi sistem

Mengevaluasi sistem, mempertimbangkan semua langkah yang pernah dilakukan. Apakah sudah sesuai atau belum. Jika belum atau masih ada perbaikan maka dapat mengulangi dan kembali di tahap sebelumnya[3].

7. Menggunakan sistem

Sistem sudah selesai diimplementasi, sebaiknya dilakukan upaya untuk *maintenance* sistem agar sistem terjaga dan berfungsi dengan baik serta dapat meningkatkan produktivitas dalam kinerja[3].

1.7 Sistematika Penulisan

Adapun sistematika penulisan Tugas Akhir ini secara garis besarnya terbagi menjadi 5 (lima) bab, sebagai berikut :

BAB I : PENDAHULUAN

Pada bab ini akan diuraikan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

BAB II : DASAR TEORI

Pada bab ini akan diuraikan beberapa hal yang berhubungan dan terkait dengan referensi-referensi yang berkaitan dengan pembuatan tugas akhir ini serta perangkat yang akan digunakan pada saat pembuatan tugas akhir ini.

BAB III : PERANCANGAN SISTEM

Pada bab ini akan diuraikan perancangan dari Aplikasi *E-arsip* pada Bagian Perencanaan dan Pengelolaan Aset Di Perumda Air Minum Tirta Khatulistiwa Kota Pontianak.

BAB IV : HASIL DAN PEMBAHASAN

Pada bab ini akan diuraikan mengenai hasil dan implementasi dari Aplikasi *E-arsip* pada Bagian Perencanaan dan Pengelolaan Aset Di Perumda Air Minum Tirta Khatulistiwa Kota Pontianak.

BAB V : PENUTUP

Pada bab ini akan diuraikan tentang kesimpulan dari pembahasan permasalahan serta saran-saran untuk membangun aplikasi atau sistem yang lebih baik untuk kedepannya.

BAB II

DASAR TEORI

2.1 Tinjauan Pustaka

Dari referensi yang didapatkan mengenai aplikasi pengarsipan sebelumnya, berdasarkan penelitian pertama pada tahun 2019 oleh Florentina Esti Nilawati¹, Mohammad Rizal, Eko Hari Rachmawanto, De Rosal Ignatius Moses Setiadi, Christy Atika Sari. Dengan judul penelitian “Implementasi *E-arsip* Untuk Penyimpanan Dokumen Digital Pada PT BPD Jateng (Bank Jateng)”, hasil dari penelitian tersebut berupa aplikasi sistem penyimpanan dokumen digital[2]. Penelitian kedua pada tahun 2020 dilakukan oleh Ridwanto, Dwi Ade Handayani Capah. Dengan judul penelitian “Aplikasi Pengelolaan Dokumen dan Arsip berbasis Web untuk mengatur Sistem kearsipan dengan menggunakan Metode Waterfall” di dalam jurnal tersebut merancang aplikasi kearsipan berbasis web[1]. Penelitian ketiga pada tahun 2023 ditulis oleh Ni Kadek Pebriyanti¹, A A Istri Ita Paramitha, Eka Grana Aristyana Dewi. Dengan judul penelitian “Model Sistem Informasi *E-arsip* Laporan Tugas Akhir Berbasis Website Pada Perguruan Tinggi. hasil dari penelitian ini berupa aplikasi informasi laporan Magang dan laporan Tugas Akhir mahasiswa[4].

Dari ketiga referensi yang telah diuraikan, penulis melakukan penelitian yang akan menghasilkan rancang bangun aplikasi pengelolaan arsip (*e-arsip*) pada Perumda Air Minum Tirta Khatulistiwa Kota Pontianak, yang dapat menyimpan, menyortir, mencetak, dan melakukan pencarian dokumen yang dapat meningkatkan efektivitas serta efisiensi dalam pekerjaan.

2.2 Dasar Teori

Dasar teori adalah sebuah konsep dengan pernyataan yang tertata rapi dan sistematis memiliki variabel dalam penelitian karena dasar teori menjadi landasan yang kuat dalam penelitian yang dilakukan.

2.2.1 Figma

Tools yang digunakan untuk mendesain dan membuat *Prototype* dari aplikasi *e-arsip* Figma. Figma merupakan *software* yang memungkinkan *User* untuk mendesain tampilan secara kolaboratif dan real time. Figma tersedia untuk desktop Windows atau MAC OS dan dapat digunakan di mana saja melalui internet,

kelebihan yang dimiliki oleh Figma antara lain dapat membuat desain *Prototype* dan mock up yang sangat cepat serta dapat melakukan kolaborasi dengan team saat mendesain *Prototype*. Terdapat berbagai macam palette yang sangat berguna untuk desain, dan tidak perlu lagi ngoding untuk membuat *Prototype* aplikasi berbasis *website* maupun *mobile*[5].

2.2.2 Website

Website adalah Sebuah halaman *web* atau sekumpulan halaman *web* yang saling terhubung dan dapat diakses dari mana saja yang terhubung ke *internet*. Setiap halaman *web* memiliki URL (*Uniform Resource Locator*). Teks, gambar, video, dan audio adalah beberapa jenis informasi yang dapat ditemukan di situs *web*. Selain itu, *web* mungkin juga memiliki fitur interaktif seperti *chat*, *form* kontak, atau komentar. Mempunyai kelebihan untuk menghubungkan (*link*) suatu dokumen dengan dokumen lainnya (*hypertext*), dapat diakses melalui sebuah *browser*[6].

2.2.3 HTML (*HyperText Markup Language*)

HTML adalah sebuah bahasa markah yang digunakan untuk membuat halaman *web* yang berisi berbagai informasi di penjelajah *web Internet*. HTML secara resmi hadir padatahun 1989 oleh Berners Lee dan dikembangkan oleh World Wide Web Consortium (W3C). HTML adalah kumpulan *script* yang dapat kita gunakan untuk membuat halaman *web*, yang memungkinkan kita menampilkan data dalam bentuk teks dan gambar di situs *web* yang kita buat[7].

2.2.4 CSS (*Cascading Style Sheet*)

CSS adalah aturan yang mengatur berbagai elemen dalam sebuah *web* untuk membuatnya lebih teratur dan konsisten. CSS dapat mengatur ukuran gambar, warna *body* teks, warna tabel, ukuran *border*, warna *border*, warna *hyperlink*, warna *mouse over*, spasi antar paragraf, margin kiri, kanan, dan atas, serta parameter lainnya. CSS adalah *style sheet language* yang digunakan untuk mengatur bagaimana tampilan *web* akan terlihat. Ini memungkinkan Anda menampilkan halaman yang sama dalam berbagai format. CSS tidak dapat digunakan secara mandiri, tetapi harus terintegrasi dalam paket pemrograman lain yang didasarkan pada HTML atau PHP. Beberapa fungsi dari CSS adalah untuk mempercantik

tampilan, membuat tampilan lebih rapi dan responsif, dan memudahkan pengelolaan kode[8].

2.2.5 PHP (*Hypertext Preprocessor*)

PHP adalah bahasa skrip yang dapat digunakan atau disisipkan ke dalam HTML dan banyak digunakan untuk membangun situs *web* dinamis. Namun, PHP berbeda dengan HTML karena PHP berada disisi *server* dan membutuhkan koneksi ke-*database*, yang memungkinkannya membuat situs *web* dinamis. PHP adalah bahasa pemrograman *script server-side* yang dimaksudkan untuk membuat *web*. Selain itu, PHP juga bisa digunakan untuk bahasa pemrograman umum. PHP ada sejak tahun 1995 oleh Rasmus Ledorf, dan sekarang dikelola oleh The PHP Group. Situs resmi PHP beralamat di <http://www.php.net>[9].

2.2.6 XAMPP

XAMPP adalah perangkat lunak yang digunakan untuk membuat dan menjalankan aplikasi *web* di komputer lokal. *XAMPP* terdiri dari beberapa program, yaitu X (Cross Platform), A (Apache), M (MySQL), P (PHP), dan P (Perl). Apache adalah *web server* yang digunakan untuk menampilkan halaman web, MySQL adalah *database server* yang digunakan untuk menyimpan data aplikasi *web*, dan PHP adalah bahasa pemrograman yang digunakan untuk membuat halaman *web* dinamis[10]. Perl adalah bahasa pemrograman yang digunakan untuk berbagai tujuan, seperti *scripting* dan pengembangan *web*. *XAMPP* memungkinkan Anda membuat berbagai aplikasi *web*, termasuk situs *web* statis, situs *web* dinamis, dan aplikasi *web* berbasis *database*.

XAMPP memberi pengembang *web* kemampuan untuk membuat dan menguji aplikasi *web* atau situs *web* secara lokal di komputer mereka sebelum mempublikasikannya di *server web* yang sebenarnya. *XAMPP* juga memiliki beberapa keuntungan tambahan yaitu, *XAMPP* gratis dan portabel, dan dapat diinstal dan digunakan dengan mudah bahkan oleh pemula dalam pengembangan *web*[10].

2.2.7 ERD (*Entity-Relation Diagram*)

Entity-Relationship Diagram (ERD), yang menggunakan beberapa notasi, digunakan untuk menjelaskan data atau objek yang dibuat berdasarkan entitas serta relasi antar entitas. Berbentuk notasi grafis yang terlibat dalam proses pembuatan

database yang menghasilkan database lebih transparan. ERD membantu proses pembuatan [11]. database dan menunjukkan cara kerja database yang dibuat ERD terdiri dari tiga komponen:

1. Entitas: entitas adalah entitas yang ada dalam database; entitas dapat berupa orang, barang, atau kondisi data. Entitas diwakili dengan bentuk persegi panjang.
2. Atribut: Atribut digunakan untuk menunjukkan informasi yang ada dalam entitas. Atribut biasanya ditempatkan dalam tabel entitas atau terpisah dari tabel entitas. Simbol atribut berbentuk elips.
3. Relasi: Menggunakan relasi untuk membuat hubungan antara entitas dua atau lebih.

ERD memiliki berbagai jenis hubungan, yaitu:

One to One, hubungan di mana setiap baris data dari tabel pertama hanya terhubung dengan satu baris data dari tabel kedua.

One to Many, setiap satu baris data dari tabel pertama memiliki hubungan dengan lebih dari satu baris data dari tabel kedua.

Many to Many, setiap lebih dari satu baris data dari tabel pertama memiliki hubungan dengan lebih dari satu baris data dari tabel kedua.

2.2.8 CI (CodeIgniter)

Untuk menghemat waktu dan menghindari penulisan *syntax* berulang-ulang serta untuk menghasilkan *source code* yang bersih dan terstruktur digunakanlah *framework*, *framework* adalah kumpulan instruksi dan instruksi yang dikumpulkan dalam *class* dan *function-function* dengan fungsi masing-masing, sehingga dapat digunakan tanpa harus menulis *syntax* program yang sama berulang kali.

CodeIgniter merupakan *framework* PHP *open source* yang gratis dan menggunakan pola MVC (*Model, View, Controller*). Dibat dengan tujuan mempermudah developer dalam membangun aplikasi *web* tanpa perlu memulai dari awal. MVC adalah konsep atau teknik yang membagi komponen utamanya menjadi tiga bagian: *model, view, dan controller*[12].

Instalasi CodeIgniter 4

Untuk memulai membangun aplikasi dengan *CodeIgniter 4* terdapat dua teknin instalasi yang bisa digunakan yaitu instalasi secara manual dan instalasi menggunakan *Composer*.

1. Instalasi Manual

Instalasi secara manual dapat dilakukan dengan mengunjungi <https://CodeIgniter.com/> dan unduh versi terbaru dari *CodeIgniter*, kemudian ekstrak didalam folder proyek aplikasi yang akan di bangun.

2. Instalasi Menggunakan *Composer*

Instalasi dengan *Composer*, *Composer* adalah program berbasis *command line (CLI)* untuk manajemen proyek *PHP*. Tugas dari *Composer* adalah melakukan instalasi paket, membuat proyek baru, menjalankan *script*, dan lain-lain. Instal *Composer* pada Git bass atau command prompt pada folder proyek untuk server lokal seperti folder htdocs atau www, dengan perintah “`apt install Composer`”. Dan instalasi file proyek *CodeIgniter* yang dilakukan dengan *Composer* dapat dilakukan dengan mengetikan perintah. “*Composer create-proyek CodeIgniter 4/appstarter* nama proyek”. Setelah prosesnya selesai, kita akan mendapatkan folder baru sesuai dengan nama proyek.

2.2.9 *Bootstrap*

Bootstrap adalah salah satu *framework* CSS paling *popular* dari sekian banyak *framework* CSS yang ada yang dikhususkan untuk pengembangan *front-end* website. *Bootstrap* memungkinkan desain *web* menjadi *responsive*, sehingga tampilan tetap menarik dari berbagai ukuran perangkat. *Bootstrap* juga membuat proses pengaturan desain menjadi lebih cepet Tidak perlu menulis banyak *css* lagi, kecuali jika memerlukan pengaturan desain yang berbeda dengan style *bootstrap*. Developer hanya perlu memanggil class untuk membuat tombol, panel, tabel, pesan peringatan, dan lainnya. Hampir semua *browser desktop* dan *mobile* mendukung *Bootstrap*[12].

Dalam proyek ini, *Bootstrap* diinstal dengan menggunakan template gratis dari situs <https://startbootstrap.com/>, dengan template yang dipilih adalah **Admin SB**. Kemudian mengunduh file ZIP yang berisi template tersebut. Setelah unduhan selesai, file ZIP di ekstrak ke dalam direktori proyek yang sudah menggunakan struktur *CodeIgniter* 4. Integrasi template ini memungkinkan pengembangan *front-end* yang lebih cepat dan terstruktur, serta memberikan tampilan yang profesional dan responsif untuk aplikasi yang sedang dibangun.

BAB III

RANCANGAN SISTEM

3.1. Gambaran Umum aplikasi

Aplikasi *E-arsip* merupakan suatu aplikasi untuk membantu PERUMDA Air Minum Tirta Khatulistiwa dalam pengelolaan dokumen (menyimpan, melihat, dan mencari dokumen). Pengguna dari aplikasi ini adalah staf pada bagian perencanaan dan pengelolaan aset.

3.2. Tahapan pengumpulan kebutuhan

Pengumpulan kebutuhan merupakan langkah yang harus dilakukan dalam sebuah penelitian. Tujuan dari pengumpulan kebutuhan adalah untuk memperoleh data yang sesuai dengan kebutuhan serta valid dan akurat. Oleh karena itu, pengumpulan kebutuhan dapat dilakukan dengan beberapa teknik seperti observasi, wawancara, dan studi literatur.

3.2.1. Teknik Pengumpulan Kebutuhan

1. Observasi

Tahapan pengumpulan informasi dan data-data awal, observasi dilakukan untuk mengumpulkan informasi yang diperlukan untuk membangun aplikasi pengarsipan dokumen pada Perumda Air Minum Tirta Khatulistiwa. Adapun hasilnya sebagai berikut: cara staf dalam melakukan pengelolaan dokumen masih secara manual dengan cara ditumpuk dan disimpan di dalam lemari arsip serta secara digital disimpan di folder lokal dalam setiap komputer staf. Jenis dokumen yang akan diarsipkan meliputi dokumen teknis, dokumen *Administrasi*, dan dokumen pelayanan.

- **Identifikasi Pengguna**

Pada aplikasi ini, terdapat dua jenis pengguna utama yang telah diidentifikasi, yaitu kepala bagian perencanaan dan pengelolaan aset sebagai *Admin*, serta staf pada bagian perencanaan dan pengelolaan aset sebagai *User*. Setiap jenis pengguna memiliki hak akses dan tanggung jawab yang berbeda dalam penggunaan aplikasi ini. Identifikasi dapat dilihat pada tabel 3.1 dibawah ini.

Tabel 3. 1 Identifikasi pengguna

No	Nama Pengguna	Peran
1	<i>Admin</i> (kepala bagian)	<i>Admin</i> memiliki akses penuh terhadap aplikasi web.
2	<i>User</i> (Staf)	staf memiliki akses penuh terhadap aplikasi web, kecuali dalam hal manajemen akun pengguna lainnya.

2. wawancara

Pada tahapan ini melibatkan proses wawancara dengan calon pengguna, yaitu *Admin* dan *User* (Staf) Perumda Air Minum Tirta Khatulistiwa bagian perencanaan dan pengelolaan aset, yang nantinya akan menggunakan aplikasi pengarsipan dokumen yang telah dirancang. Hasil dari wawancara ini akan disusun dan digunakan sebagai acuan dalam perancangan aplikasi pengarsipan dokumen(*e-arsip*).

- *User Story*

User story adalah penjelasan tentang kebutuhan sistem yang ditulis dengan bahasa yang natural dan sederhana yang dapat dipahami dengan mudah. *User story* biasanya ditulis dengan format yang serupa agar lebih seragam. Dari hasil wawancara dengan staf Perumda Air Minum Tirta Khatulistiwa bagian perencanaan dan pengelolaan aset. Berikut daftar *User story* yang dapat dilihat pada tabel 3.2

Tabel 3. 2 User Story

Kode	<i>User Story</i>
CASE-1	Sebagai staf, saya ingin dapat mengunggah dokumen secara mudah ke dalam sistem agar semua dokumen dapat diakses secara digital.
CASE-2	Sebagai staf, saya ingin dapat mengelola dokumen dengan cara yang efisien sehingga saya bisa mengkategorikan dan menemukan dokumen dengan cepat.
CASE-3	Sebagai staf, saya ingin dapat mengarsipkan dokumen lama secara digital agar mengurangi penggunaan ruang fisik di lemari arsip.

CASE-4	Sebagai staf, saya ingin melihat dokumen dengan mudah dan sederhana agar dapat mengakses informasi yang diperlukan dengan cepat.
CASE-5	Sebagai <i>Admin</i> , saya ingin memiliki hak yang sama dengan staf serta hak tambahan untuk mengelola pengguna aplikasi.

- **Kebutuhan Fungsional Dan Hak Pengguna**

Kebutuhan fungsional mendefinisikan apa yang layanan atau fasilitas yang ditawarkan oleh aplikasi untuk memenuhi kebutuhan pengguna. Berikut kebutuhan dan hak akses pengguna yang ditunjukkan pada tabel 3.3.

Tabel 3. 3 Kebutuhan Fungsional dan Hak Akses

No	Kebutuhan Fungsional	<i>Admin</i>	<i>Staf</i>
1	Menambah data dokumen	Ya	Ya
2	Mengedit data dokumen	Ya	Ya
3	Menghapus data dokumen	Ya	Ya
4	Melihat detail dan waktu update dokumen	Ya	Ya
5	Melihat dokumen	Ya	Ya
6	Mencari dokumen	Ya	Ya
7	Menyaring dokumen	Ya	Ya
8	Mencetak dan mengunduh dokumen	Ya	Tidak
9	Mengelola akun pengguna	Ya	Tidak

- **Kebutuhan Non Fungsional**

Kebutuhan non fungsional mendefinisikan layanan atau fasilitas pendukung serta memperkuat kebutuhan fungsional dari aplikasi. berikut kebutuhan dan hak akses pengguna yang ditunjukkan pada tabel 3.5.

Tabel 3. 4 Kebutuhan Non Fungsional

No	Kebutuhan Non Fungsional
1	Setiap pengguna harus melakukan login dengan <i>Username</i> dan <i>password</i> untuk mengakses sistem.
2	Sistem harus membedakan hak akses antara <i>Admin</i> dan <i>User(staf)</i>

3	Aplikasi dapat beroperasi terus menerus selama 24 jam, karena aplikasi bersipaf online
4	Aplikasi mudah digunakan oleh pengguna baru

3. Studi literatur

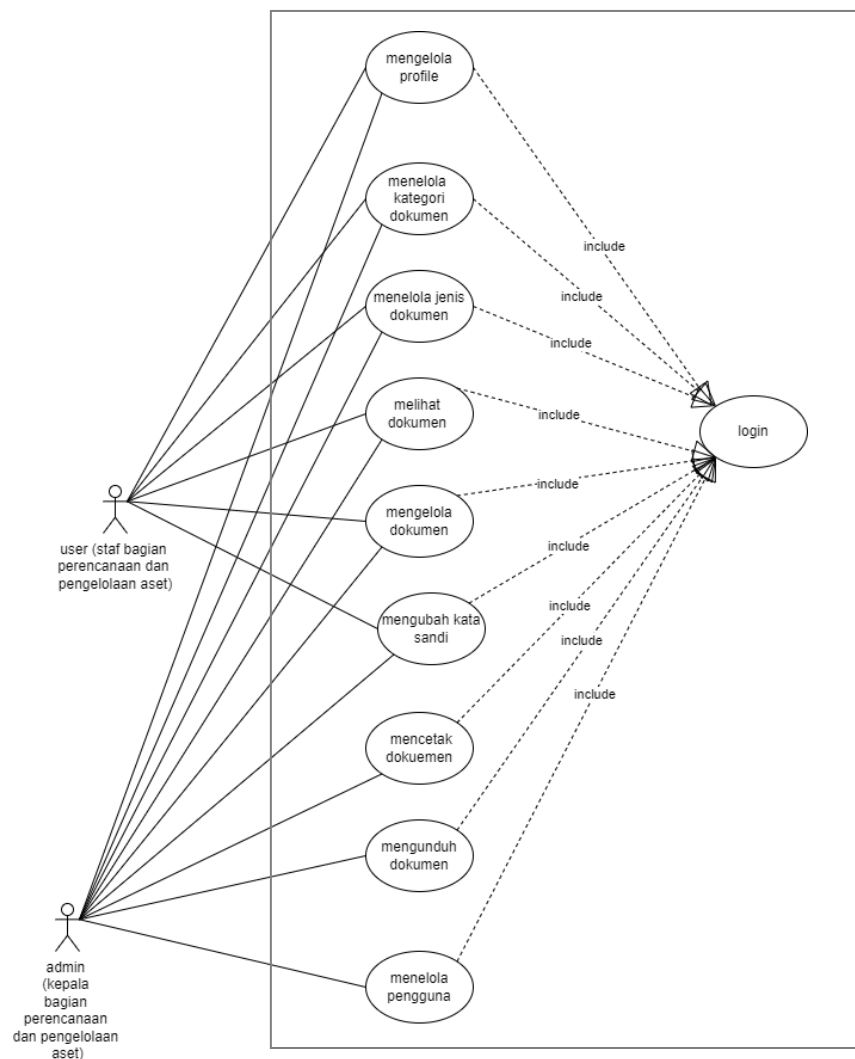
Data terkait sistem pengarsipan dokumen dikumpulkan melalui berbagai teknik untuk memperoleh informasi yang relevan. Referensi yang digunakan mencakup buku-buku, jurnal, artikel penelitian, dan sumber-sumber lainnya yang berhubungan dengan aplikasi yang sedang dibahas. referensi yang digunakan sebagai gambaran dari aplikasi yang dibangun adalah penelitian oleh Florentina Esti Nilawati¹, Mohammad Rizal, Eko Hari Rachmawanto, De Rosal Ignatius Moses Setiadi, Christy Atika Sari pada tahun 2019. Dengan judul penelitian “Implementasi *E-arsip* Untuk Penyimpanan Dokumen Digital Pada PT BPD Jateng (Bank Jateng)”, hasil dari penelitian tersebut berupa aplikasi sistem penyimpanan dokumen digital[2]. Penelitian kedua pada tahun 2020 dilakukan oleh Ridwanto, Dwi Ade Handayani Capah. Dengan judul penelitian “Aplikasi Pengelolaan Dokumen dan Arsip berbasis Web untuk mengatur Sistem kearsipan dengan menggunakan Metode Waterfall” di dalam jurnal tersebut merancangan aplikasi kearsipan berbasis web[1]. Penelitian ketiga pada tahun 2023 ditulis oleh Ni Kadek Pebriyanti¹, A A Istri Ita Paramitha, Eka Grana Aristyana Dewi. Dengan judul penelitian “Model Sistem Informasi *E-arsip* Laporan Tugas Akhir Berbasis Website Pada Perguruan Tinggi. hasil dari penelitian ini berupa aplikasi informasi laporan Magang dan laporan Tugas Akhir mahasiswa[4]. Dan Peraturan Walikota Pontianak Nomor 44 Tahun 2009 tentang Pedoman dan Struktur Organisasi, Tugas, Fungsi, dan Mekanisme Kerja Perusahaan Daerah Air Minum Tirta Khatulistiwa, bagian Seksi Logistik dan Manajemen Aset (Pasal 52 h) diatur mengenai penyelenggaraan pengelolaan dan pengamanan Administrasi atas aset utama, aset pendukung lainnya, dan dokumen penting lainnya.

3.3. Use Case Diagram

Use case diagram adalah gambaran penggunaan dari sebuah sistem. Use case menjelaskan secara garis besar dari aplikasi yang dibangun dan memberikan gambaran menyeluruh mengenai bagaimana pengguna dan sistem berinteraksi, serta kebutuhan fungsionalitas yang harus dipenuhi. Dalam konteks aplikasi E-arsip

yang dibangun, Use Case Diagram berfungsi untuk memvisualisasikan berbagai fungsi utama yang harus ada dalam sistem. Diagram ini menunjukkan interaksi antara pengguna dengan dua jenis otorisasi: *Admin* dan *User*(staf).

Pengguna dengan otorisasi sebagai *User* dapat mengelola kategori dokumen (tambah, edit dan hapus), mengelola jenis dokumen (memilih jenis), melihat dokumen, mengelola dokumen (tambah dan edit), *Profile* (melihat *Profile* milik sendiri) serta mengubah kata sandi, dan logout. Sedangkan pengguna dengan otorisasi sebagai *Admin* memiliki akses yang sama dengan staf tetapi memiliki hak khusus untuk mengelola dokumen mencetak dokumen, mengunduh dokumen, menghapus dokumen dan mengelola akun pengguna (tambah, edit dan hapus).



Gambar 3. 1 Use Case Diagram

3.4.Membangun *Prototype*

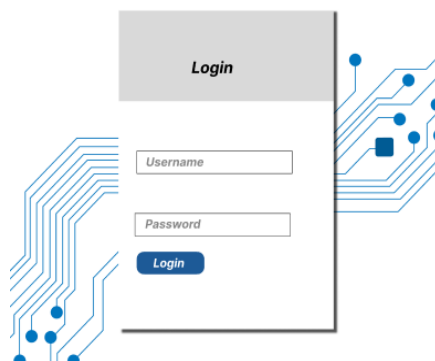
Prototype merupakan sebuah metode dalam mengembangkan aplikasi dengan cara memberikan contoh penawaran sebuah rancangan kepada calon pengguna dari aplikasi dan memberikan evaluasi *Prototype* sebelum dilakukan tahap melakukan pemrograman. Pada tahap ini, proses perancangan dan pembangunan *Prototype* sistem dilakukan dengan fokus pada kebutuhan dan keluhan pengguna yang telah diidentifikasi sebelumnya. *Prototype* dirancang sesuai dengan fungsi utama aplikasi yang telah diidentifikasi, seperti pengelolaan kategori dan jenis dokumen, serta kemampuan untuk melihat, mengelola, mencetak, dan mengunduh dokumen. Selain itu, fitur untuk mengelola akun pengguna juga diterapkan, memungkinkan *Admin* untuk menambah, mengedit, dan menghapus akun. Dengan membangun *Prototype* yang sesuai, diharapkan pengguna memiliki gambaran tentang aplikasi yang akan dibangun.

3.4.1. *Mockup*

Mockup adalah gambaran visual dari aplikasi yang menunjukkan tata letak dan elemen-elemen antarmuka pengguna. *Mockup* membantu dalam memahami bagaimana aplikasi akan terlihat dan berfungsi sebelum pengembangan dimulai.

1. *Mockup* Login

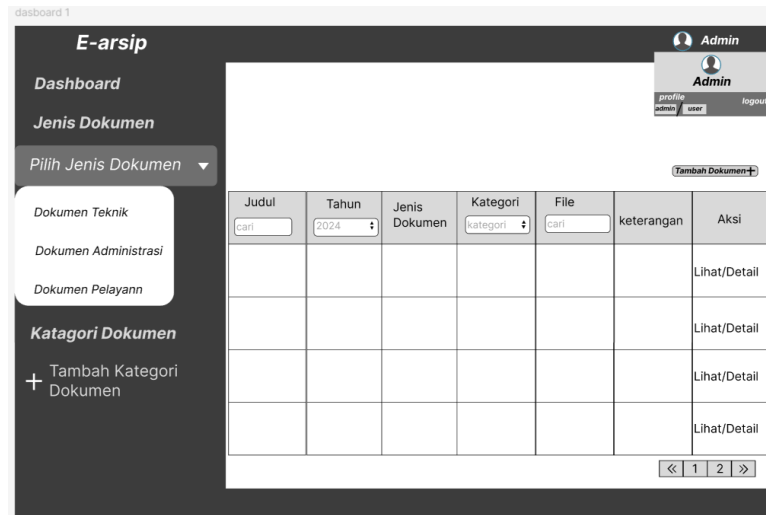
Halaman login merupakan halaman pertama yang diakses oleh pengguna. Halaman ini berfungsi sebagai autentikasi pengguna, di mana pengguna harus memasukkan *Username* dan *password* yang terdaftar. Gambar *Mockup* dari halaman login dapat dilihat pada Gambar 3.2.



Gambar 3. 2 *Mockup* Halaman Login

2. Mockup Dashboard (Admin dan Staf)

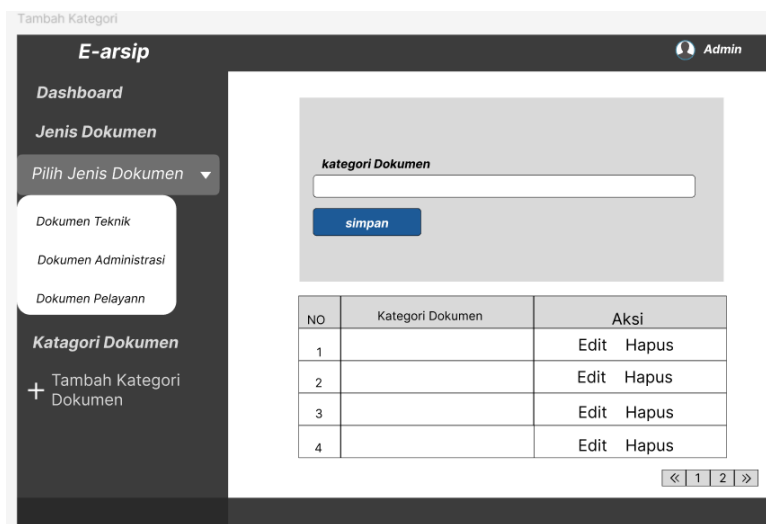
Dashboard adalah halaman utama yang ditampilkan setelah pengguna berhasil login. Halaman ini menampilkan informasi ringkasan dari berbagai fitur dalam aplikasi, seperti daftar dokumen, jenis dan kategori dokumen, dan *profile*. Gambar *Mockup* dari halaman *dashboard* dapat dilihat pada Gambar 3.3.



Gambar 3. 3 *Mockup* Halaman *dashboard*

3. Mockup Tambah Kategori Dokumen (Admin dan Staf)

Halaman tambah kategori dokumen digunakan oleh pengguna untuk menambahkan kategori baru bagi dokumen yang akan diarsipkan. ini mempermudah pengelolaan dokumen dengan mengkategorikan dokumen secara lebih terstruktur. Gambar *Mockup* dari halaman tambah kategori dokumen dapat dilihat pada Gambar 3.4.



Gambar 3. 4 *Mockup* Halaman Tambah Kategori Dokumen

4. *Mockup* Edit Kategori Dokumen (Admin dan Staf)

Halaman edit kategori dokumen memungkinkan pengguna untuk mengedit kategori dokumen yang telah ada, memastikan bahwa pengelolaan kategori tetap relevan dan up-to-date. Gambar *Mockup* dari halaman edit kategori dokumen dapat dilihat pada Gambar 3.5.

Gambar 3. 5 *Mockup* Halaman Edit Kategori Dokumen

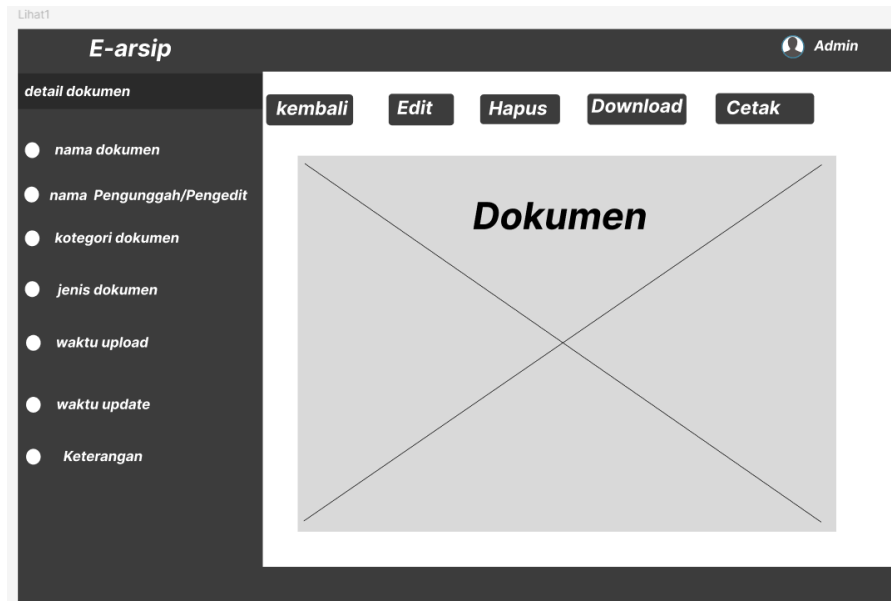
5. *Mockup* Form Tambah Dokumen (Admin dan Staf)

Form Tambah Dokumen memungkinkan pengguna untuk menambahkan dokumen baru ke dalam sistem, termasuk mengunggah file dan mengisi informasi terkait dokumen tersebut. Gambar *Mockup* dari *form* tambah dokumen dapat dilihat pada Gambar 3.6.

Gambar 3. 6 *Mockup* Form Tambah Dokumen

6. *Mockup* Lihat Dokumen (*Admin*)

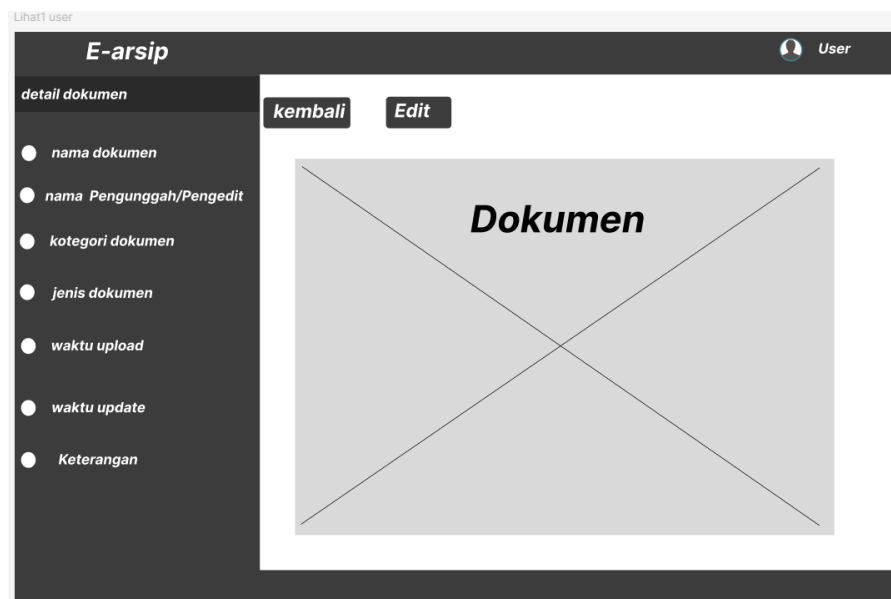
Halaman lihat dokumen memungkinkan *admin* untuk melihat detail dokumen yang telah diarsipkan, *preview* dari dokumen tersebut. Gambar *Mockup* dari halaman lihat dokumen dapat dilihat pada Gambar 3.7.



Gambar 3. 7 *Mockup* Halaman Lihat Dokumen(*admin*)

7. *Mockup* Lihat Dokumen (*Staf*)

Halaman lihat dokumen memungkinkan staf untuk melihat detail dokumen yang telah diarsipkan, *preview* dari dokumen tersebut. Gambar *Mockup* dari halaman lihat dokumen dapat dilihat pada Gambar 3.8.



Gambar 3. 8 *Mockup* Halaman Lihat Dokumen(*staf*)

8. Mockup Form edit Dokumen (Admin dan Staf)

Form edit dokumen memungkinkan pengguna untuk mengedit informasi dokumen yang telah diunggah sebelumnya. Hal ini mencakup perubahan informasi dokumen dan penggantian file dokumen jika diperlukan. Gambar Mockup dari form edit dokumen dapat dilihat pada Gambar 3.9.

Gambar 3. 9 Mockup Form edit Dokumen

9. Mockup profile admin

Halaman Profil Admin menampilkan tabel yang berisi informasi semua pengguna. Di halaman ini, Admin dapat melihat dan mengedit informasi pribadi. Admin juga memiliki akses untuk mengelola data pengguna lainnya. Gambar Mockup untuk halaman profil Admin dapat dilihat pada Gambar 3.10.

NO	username	Nama	NIP	Akses	Aksi
1					Ubah Hapus
2					Ubah Hapus
3					Ubah Hapus
4					Ubah Hapus

Gambar 3. 10 Mockup Halaman Profile Admin

10. *Mockup Form Tambah Akun*

Form tambah akun digunakan oleh *Admin* untuk menambahkan akun pengguna baru ke dalam sistem, termasuk menentukan peran atau akses pengguna (*Admin* atau staf). Gambar *Mockup* dari *form* tambah akun dapat dilihat pada Gambar 3.11.

Gambar 3. 11 *Mockup Form Tambah Akun*

11. *Mockup Form edit Akun*

Form edit akun memungkinkan *Admin* untuk mengedit informasi akun pengguna yang telah ada, merubah kata sandi dan peran atau akses. Gambar *Mockup* dari *form edit* akun dapat dilihat pada Gambar 3.12.

Gambar 3. 12 *Mockup Form edit Akun*

12. *Mockup Profile User*

Halaman *Profile User* menampilkan informasi profil pengguna yang memiliki peran sebagai *User*. Pengguna dapat melihat dan mengedit informasi pribadinya melalui halaman ini. Gambar *Mockup* dari halaman *Profile User* dapat dilihat pada Gambar 3.13.

The mockup shows the 'E-arsip' system interface. On the left is a dark sidebar with the following menu items: 'Dashboard', 'Jenis Dokumen' (with a dropdown 'Pilih Jenis Dokumen' containing 'Dokumen Teknik', 'Dokumen Administrasi', and 'Dokumen Pelayann'), and 'Kategori Dokumen' (with a '+ Tambah Kategori Dokumen' button). At the bottom of the sidebar is a 'kembali' button. The main content area has a top bar with 'E-arsip' and a 'User' profile icon. Below this is a large circular profile picture placeholder. To the right of the profile picture is a 'Ubah Data Diri' button. Below the profile picture is a table with the following data:

NO	username	Nama	NIP	Akses
1				

Below the table is a pagination control showing '<< 1 2 >>'. At the bottom left of the main content area is a 'kembali' button.

Gambar 3. 13 *Mockup* Halaman *Profile User*

13. *Mockup Form Ubah Data Diri (Admin dan User)*

Form ini memungkinkan pengguna untuk mengubah data diri, dan kata sandi. Gambar *Mockup* dari *form* ubah data diri dapat dilihat pada Gambar 3.14.

The mockup shows the 'E-arsip' system interface for the 'Ubah data diri' form. The top bar shows 'E-arsip' and an 'Admin' profile icon. The sidebar on the left has a 'Profile' section with a circular profile picture placeholder and labels for 'NAMA', 'NIP', and 'AKSES'. The main content area has a title 'Ubah data diri' and the following form fields:

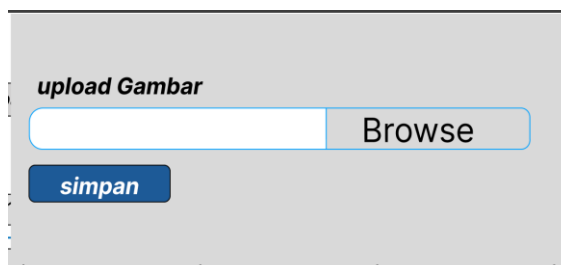
- Username :
- Nama :
- Password :
- Verifikasi Password :
- NIP :

At the bottom of the form is a blue 'Simpan' button with a right-pointing arrow.

Gambar 3. 14 *Mockup Form* Ubah Data Diri (Admin dan User)

14. *Mockup Form edit Gambar Profile* (Admin dan User)

Form edit gambar *Profile* digunakan untuk mengubah gambar profil pengguna dalam aplikasi. Pengguna dapat memilih file gambar baru dari perangkat mereka untuk dijadikan gambar profil. Setelah gambar dipilih, pengguna dapat menyimpan perubahan tersebut dan gambar profil akan diperbarui sesuai dengan gambar yang baru diunggah. Berikut adalah *Mockup* dari *form edit* gambar profil seperti pada gambar 3.15.



Gambar 3. 15 *Mockup Form edit Gambar Profile* (Admin dan User)

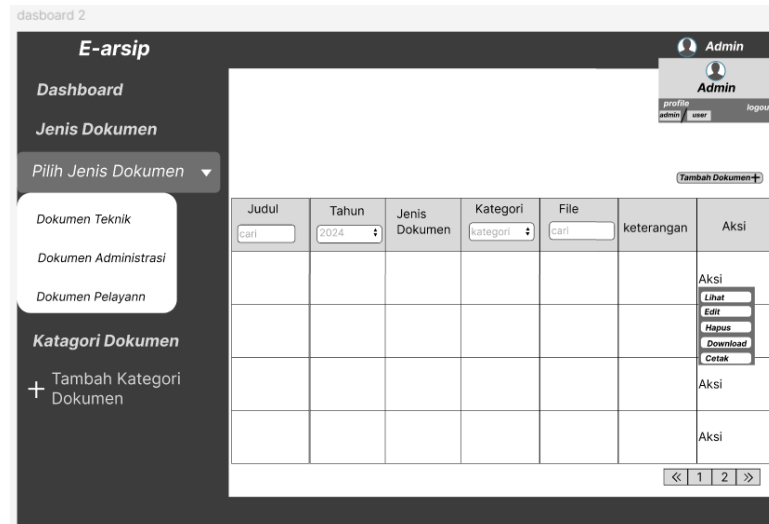
3.5.Evaluasi *Prototype*

Evaluasi *Prototype* dilakukan untuk menilai apakah *Prototype* yang dibuat telah sesuai dengan kebutuhan pengguna dan apakah ada perubahan yang perlu dilakukan.

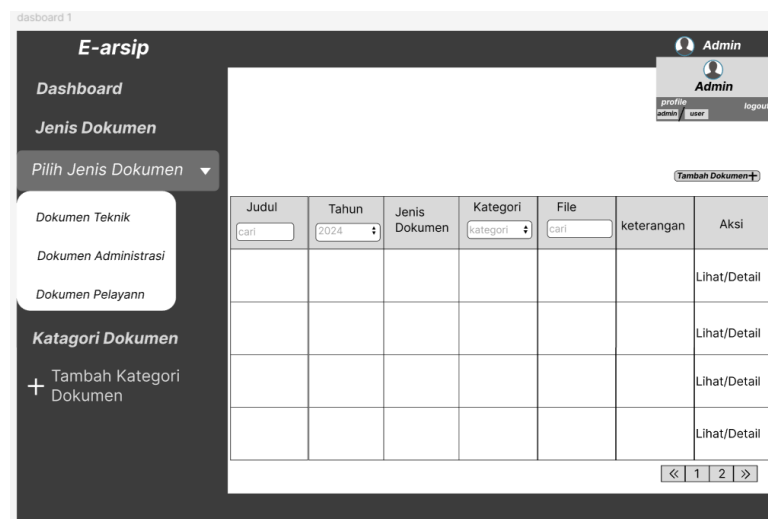
3.5.1. Halaman *dashboard*

Perubahan pada halaman *dashboard* dilakukan untuk meningkatkan efisiensi dan kemudahan penggunaan. Pada desain awal, kolom aksi pada tabel dokumen di halaman *dashboard* mengandung sebuah dropdown, seperti terlihat pada Gambar 3.16.

Namun, berdasarkan evaluasi dan tanggapan pengguna, pendekatan ini dianggap kurang sederhana dan dapat memperlambat interaksi pengguna dengan sistem. Oleh karena itu, dilakukan penyederhanaan pada kolom aksi dengan mengurangi jumlah tindakan langsung yang dapat diakses dari *dashboard*. Sebagai bagian dari perbaikan, tindakan-tindakan yang sebelumnya tersedia di dropdown menu dipindahkan ke halaman terpisah, yaitu halaman Lihat Dokumen. Akibatnya, kolom aksi pada halaman *dashboard* sekarang hanya menampilkan opsi Lihat/Detail, seperti pada Gambar 3.17. Perubahan ini bertujuan untuk menyederhanakan antarmuka pengguna dan memfokuskan pengguna pada tindakan yang paling sering dilakukan, yaitu melihat atau mengakses detail dokumen.



Gambar 3. 16 *Mockup* Halaman *Dashboard* Sebelum Dievaluasi

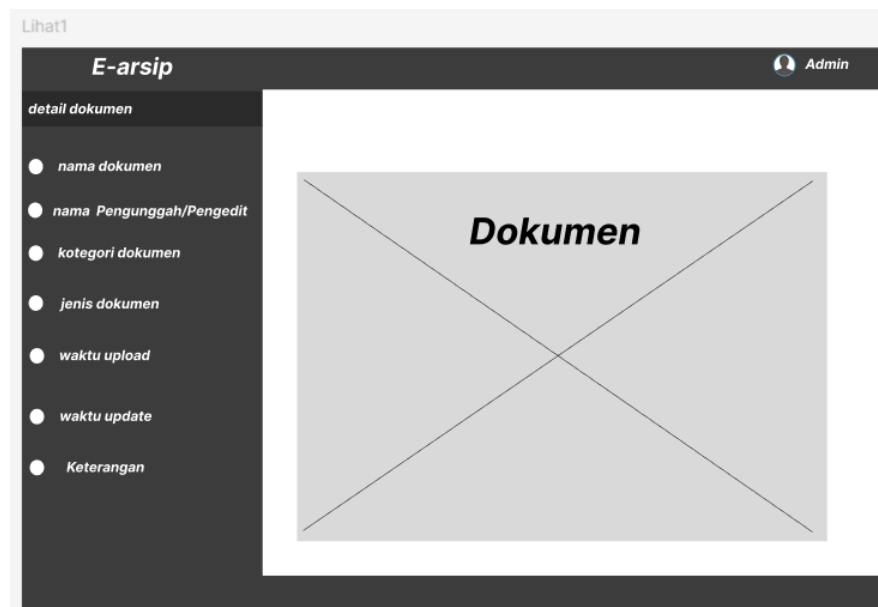


Gambar 3. 17 *Mockup* Halaman *Dashboard* Setelah Dievaluasi

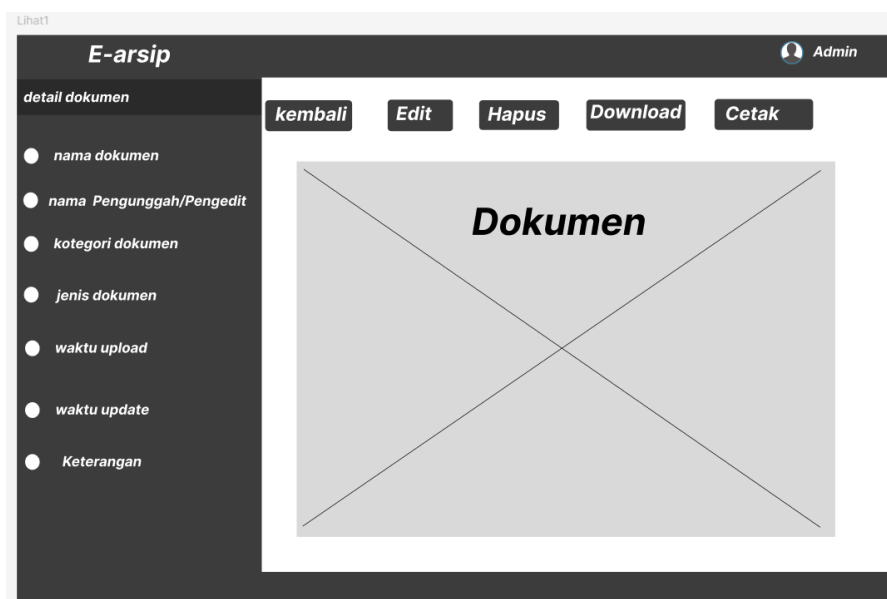
3.5.2. Halaman Lihat Dokumen

Sebagai tindak lanjut dari penyederhanaan yang dilakukan pada halaman *dashboard*, perubahan juga diterapkan pada Halaman Lihat Dokumen. Dalam desain awal, berbagai perintah aksi seperti mengedit, menghapus, atau mengunduh dokumen ditempatkan dalam dropdown menu pada kolom aksi di halaman *dashboard*. Gambar sebelum perubahan dapat dilihat pada gambar 3.18. Namun, perintah-perintah aksi ini telah dipindahkan ke halaman Lihat Dokumen. Dari perubahan ini ketika pengguna memilih untuk melihat detail suatu dokumen dari halaman *dashboard*, maka akan diarahkan ke Halaman Lihat Dokumen yang lebih lengkap. Di sini, selain melihat informasi lengkap terkait dokumen, pengguna juga

akan menemukan semua perintah aksi yang sebelumnya ada di kolom aksi *dashboard*. Perintah-perintah ini sekarang tersedia dalam bentuk tombol. Gambar setelah perubahan dapat dilihat pada gambar 3.19. Dari perubahan ini juga membantu dalam menjaga tampilan *dashboard* tetap bersih dan minimalis, memungkinkan pengguna untuk dengan cepat mendapatkan gambaran umum dari dokumen yang tersedia tanpa disibukkan oleh terlalu banyak pilihan aksi.



Gambar 3. 18 *Mockup* Halaman Lihat Dokumen Sebelum Dievaluasi



Gambar 3. 19 *Mockup* Halaman Lihat Dokumen Setelah Dievaluasi

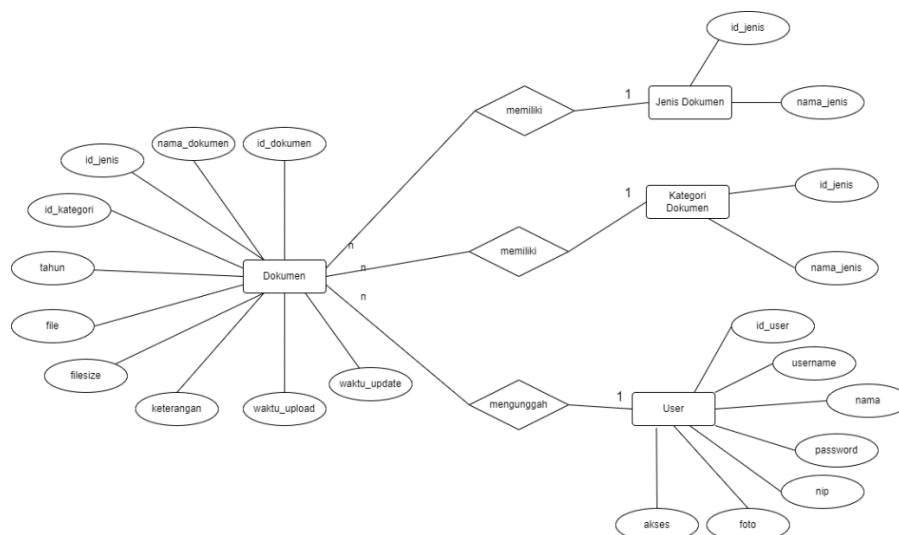
3.6. Mengkodekan sistem

Tahapan pengkodean sistem, proses pengkodean dilakukan berdasarkan desain dan evaluasi *Prototype* yang telah dibuat. Proses ini bertujuan untuk menerjemahkan desain menjadi aplikasi yang dapat dioperasikan. Dalam tahap ini, pengkodean dilakukan dengan memanfaatkan template Bootstrap untuk memastikan tampilan antarmuka pengguna yang responsif dan menarik, serta *framework CodeIgniter 4* menjadi kerangka kerja untuk pengembangan aplikasi.

Sebelum melakukan pengkodean sistem, maka akan dilakukan perancangan model yang akan digunakan sebagai dasar pengembangan aplikasi. Perancangan model ini mencakup Entity-Relationship Diagram (ERD) dan struktur tabel dalam database sebagai berikut.

3.6.1. Entity Relationship Diagram

ERD menggambarkan hubungan antara entitas-entitas dalam basis data yang digunakan oleh aplikasi. ERD memberikan gambaran struktural dari sistem yang mencakup tabel-tabel utama dan relasi antar tabel tersebut. Diagram ERD dari aplikasi e-arsip ini mencakup entitas seperti *User*, dokumen, kategori dokumen, dan jenis dokumen. ERD dapat dilihat pada gambar 3.20.



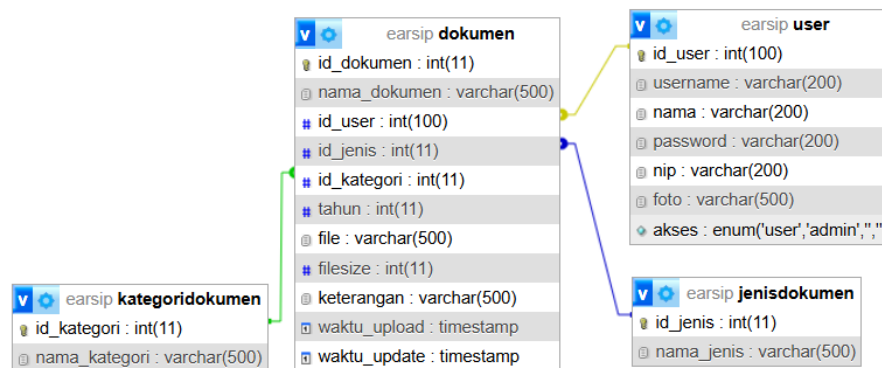
Gambar 3. 20 Entity Relationship Diagram

Dari gambar dapat dilihat bahwa tabel dokumen dan tabel kategori dokumen mempunyai relasi "memiliki" dengan kardinalitas n:1. Artinya, satu dokumen dapat dimiliki oleh satu kategori dokumen, sedangkan satu kategori dokumen dapat

memiliki banyak dokumen. Tabel dokumen dan tabel jenis dokumen mempunyai relasi "memiliki" dengan kardinalitas n:1. Artinya, satu dokumen dapat dimiliki oleh satu jenis dokumen, sedangkan satu jenis dokumen dapat memiliki banyak dokumen. Tabel user tidak memiliki relasi langsung dengan tabel lainnya, tabel user hanya berisi informasi terkait pengguna seperti username, nama, password, NIP, foto, dan akses, tanpa keterkaitan dengan data dokumen. Hal ini menunjukkan bahwa sistem ini tidak menghubungkan pengguna secara langsung dengan dokumen yang mereka unggah atau kelola.

3.6.2. Struktur Databases

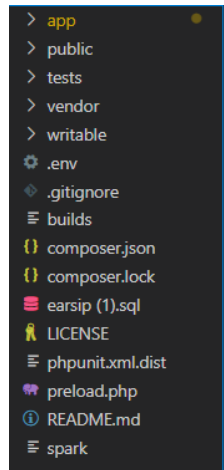
Basis data yang digunakan dalam aplikasi ini dirancang untuk mendukung semua fungsi yang diperlukan, termasuk penyimpanan informasi pengguna, dokumen, serta kategori dan jenis dokumen. Struktur tabel dalam basis data ini dapat dilihat pada Gambar 3.21.



Gambar 3. 21 Struktur Databases

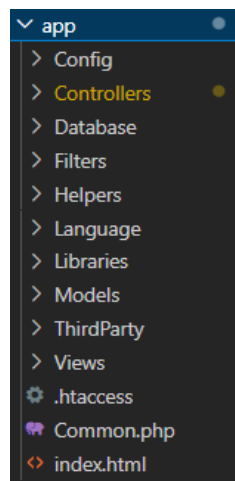
3.6.3. Struktur proyek

Struktur proyek mengatur file dan direktori dalam proyek perangkat lunak untuk mendukung pengembangan, pengujian, deployment, dan pemeliharaan aplikasi. Struktur ini biasanya dirancang agar mudah dipahami dan konsisten sehingga pengembang dapat dengan mudah menemukan dan meletakkan kode dan file yang relevan. Struktur proyek untuk *CodeIgniter* 4 seperti pada gambar 3.22.

Gambar 3. 22 struktur Proyek *CodeIgniter*

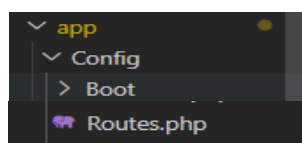
1. App

App adalah direktori utama yang berisi logika aplikasi seperti kontroler, model, pandang, filter, dan konfigurasi aplikasi. Pada pembangunan aplikasi pekerjaan untuk kode-kode utama aplikasi disimpan di app dalam beberapa direktori. Seperti ditunjukkan pada gambar 3.23.



Gambar 3. 23 Direktori app

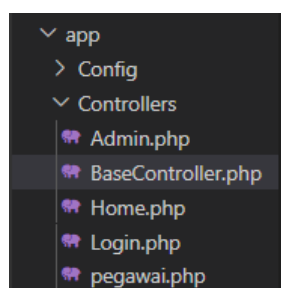
- app/config, pada direktori ini terdapat sebuah file yaitu routes.php. routes merupakan salah satu bagian penting dalam melakukan pengkodean dengan pada *CodeIgniter*. Dimana routes memiliki fungsi sebagai jalur URL yang



Gambar 3. 24 Direktori app/config

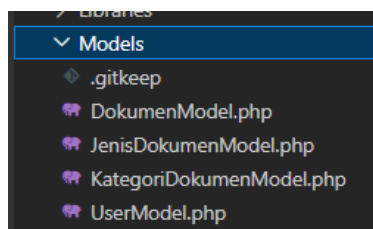
menghubungkan permintaan dari pengguna atau view ke controller. Seperti ditunjukkan pada gambar 3.24.

- `app/Controllers`, Berisi file-file *controller* yang mengatur logika aplikasi. Pada proyek ini memiliki tiga *controller* yaitu, *controller* login yang mengatur pembagian role pengguna sebagai *Admin* atau staf serta sesi dari aplikasi. *Controller Admin* yang mengatur semua logika untuk pengguna dengan akses sebagai *Admin*. Dan *controller* staf mengatur semua logika untuk pengguna dengan akses sebagai *User* biasa atau staf. Seperti ditunjukkan pada gambar 3.25.



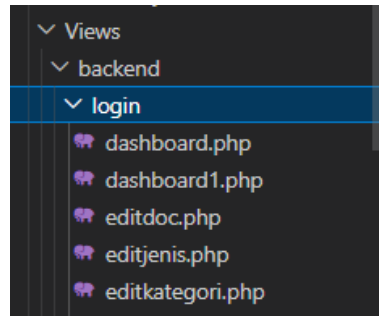
Gambar 3. 25 Direktori `app/Controllers`

- `app/Models`, Berisi file-file model yang berinteraksi dengan basis data. Pada proyek ini memiliki empat model yaitu. *UserModel* yang berinteraksi dengan tabel *user* pada database. *JeniasDokumenModel* yang berinteraksi dengan tabel *jenisdokumen* pada database. *KategoriDokumenModel* yang berinteraksi dengan tabel *kategoridokumen* pada database. *DokumenModel* yang berinteraksi dengan tabel *dokumen* pada database. Seperti ditunjukkan pada gambar 3.26.



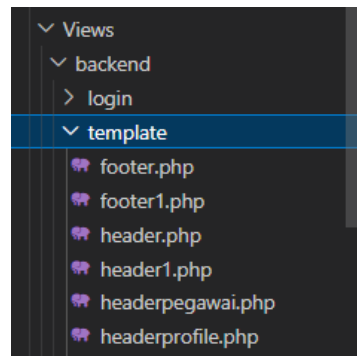
Gambar 3. 26 Direktori `app/Models`

- `app/views/backend/login`, Berisi file-file content dari *view* yang mengatur tampilan antarmuka pengguna. Seperti ditunjukkan pada gambar 3.27.



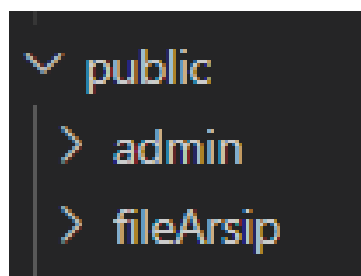
Gambar 3. 27 Direktori app/views/backend/login

- `app/views/backend/template`, Berisi file-file template seperti *navbar*, *sidebar*, dan footer dari *view* yang mengatur tampilan antarmuka pengguna. Seperti ditunjukkan pada gambar 3.28.

Gambar 3. 28 `app/views/backend/template`

2. Public

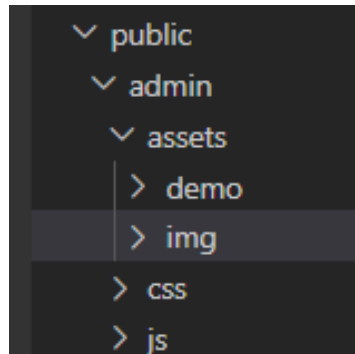
Public adalah tempat yang ideal untuk menyimpan file statis yang dapat diakses langsung oleh pengguna melalui URL. Ini mencakup file CSS, JavaScript, gambar, dan file media lainnya. Dengan menyimpan file ini di dalam folder public, browser dapat mengakses file tersebut tanpa harus menjalankan proses PHP. Tampilan direktori public ini ditunjukkan pada gambar 3.29.



Gambar 3. 29 Direktori Public

- `public/Admin/asset`: Berisi file-file publik seperti JavaScript, CSS, dan gambar. Seperti ditunjukkan pada gambar 3.30.

- `public/Admin/fileArsip`: Berisi file-file dokumen yang telah diupload. Seperti ditunjukkan pada gambar 3.30.



Gambar 3. 30 Direktori `public/Admin/asset` dan `public/Admin/fileArsip`

3.6.4. Login

Proses login melibatkan validasi dan pengolahan sesi. *view* menampilkan *form* login kepada pengguna. Dari segi tampilan halaman, aplikasi memanfaatkan template bootstrap yang sudah diinstal sebelumnya dan disesuaikan agar dapat terhubung atau sesuai dengan codeIgniter.

Setelah pengguna mengisi *form* tersebut, *routes* akan mengirimkan data tersebut dan *controller* menerima input dan mengelola alur autentikasi, controller bertanggung jawab melakukan validasi. Pada kasus ini *Library Validation rules* dimanfaatkan untuk memvalidasi *username* dan *password*. Library ini memastikan bahwa data yang diinputkan sesuai dimana dalam aplikasi, aturan rules yang digunakan adalah wajib mengisi kolom *username* dan *password*. Implementasi *Library Validation rules* dapat dilihat pada gambar 3.31.

```
$rules = [
    'username' => 'required',
    'password' => 'required'
];

if (!$this->validate($rules)) {
    $data['validation'] = $this->validator;
    return view('Backend/login/login', $data);
} else {
```

Gambar 3. 31 Implementasi code Library Validation rules

Setelah dilakukan validasi model akan melakukan pengecekan kesesuaiannya imputan dan data yang terdapat pada tabel user. *view* kemudian menampilkan hasil

kepada pengguna, baik jika login berhasil maupun jika terdapat kesalahan dalam proses autentikasi. Kemudian jika didapatkan kesesuaian data controller akan mengatur sesi bagi pengguna dengan memanfaatkan *session library adding session data*, dimana data akan tersedia secara global saat session dipanggil dihalaman berikutnya. Implementasi *session library adding session data* dapat dilihat pada gambar 3.32.

```
if ($verifikasi_pass) {
    // Set session data
    $session->set([
        'user_id' => $cekUsername['id_user'],
        'username' => $cekUsername['username'],
        'nama' => $cekUsername['nama'],
        'nip' => $cekUsername['nip'],
        'akses' => $cekUsername['akses'],
        'foto' => $cekUsername['foto'],
        'logged_in' => true,
        'last_activity' => time()
    ]);
}
```

Gambar 3. 32 Implementasi code session library adding session data

Controller akan memeriksa akses dari akun pengguna yang selanjutnya akan diarahkan dan disesuaikan tampilan aplikasi sesuai dengan akses yang dimiliki. Pada aplikasi ini saat akses adalah admin maka controller yang digunakan adalah admin dan jika akses adalah user maka controller yang digunakan adalah staf. Jalur penampilan url diatur didalam routes. Implementasi pemeriksaan akses dapat dilihat pada gambar 3.33.

```
switch ($cekUsername['akses']) {
    case "admin":
        return redirect()->to('admin/dashboard');
    case "user":
        return redirect()->to('pegawai/dashboard');
    default:
        $session->setFlashdata('pesan', 'Anda tidak punya akun');
        return redirect()->to('/login');
}
```

Gambar 3. 33 Implementasi code pemeriksaan akses

3.6.5. *dashboard*

dashboard adalah halaman utama setelah pengguna *login* tampilan dari halaman *dashboard* juga memanfaatkan template dari *bootstrap*. Pada awal nya template menampilkan semu tampilan dalam satu program. Setelah dilakukan penyesuaian tamplate dibagi menjadi empat bagian yaitu bagian header yang mencakup navbar, bagian *sidebar*, main content, dan footer. Ini dilakukan agar saat mengkodekan menjadi lebih efisien. Controller bertanggung jawab untuk mengatur halaman tersebut yang kemudian ditampilkan kepada pengguna.

View pada *dashboard* menampilkan data utama kepada pengguna, termasuk data dokumen dalam format tabel yang dapat di *filter* berdasarkan tahun dan kategori, daftar jenis dokumen pada *sidebar*, dan informasi pengguna di *navbar*. *Controller* mengelola logika pengambilan data yang sesuai dari berbagai tabel melalui model. Model bertanggungjawab mengambil informasi dari tabel dokumen, jenisdokumen, dan kategoridokumen dan *controller* menyusun data tersebut untuk ditampilkan dalam *view*. Salah satu *library* dari *codeIgniter* digunakan adalah *Library Times and Dates* digunakan untuk mengatur dan memanipulasi waktu dan tanggal. Dalam *dashboard*, data dokumen diurutkan berdasarkan waktu_update menggunakan *Time::parse()* untuk memastikan dokumen terbaru ditampilkan paling atas. Implementasi *Library Times and Dates* dapat dilihat pada gambar 3.34.

```
// Urutkan data berdasarkan waktu_update dari yang terbaru
usort($data['filteredData'], function ($a, $b) {
    return strtotime($b['waktu_update']) - strtotime($a['waktu_update']);
});
```

Gambar 3. 34 Implementasi code Library Times and Dates

3.6.6. Pengolahan Dokumen

Pengelolaan dokumen mencakup fungsi untuk menambah, mengedit, menghapus, mencetak, *download*, dan melihat dokumen. *view* menampilkan tombol tambah dokumen, daftar dokumen pada tabel, dan tombol detail/lihat dokumen pada *dashboard* kepada pengguna. untuk fungsi lain seperti mencetak, *mendownload*, menambah, mengedit, dan menghapus ditampilkan oleh *view* pada lihat dokumen.

Pada pengolahan dokumen khususnya pada bagian penambahan dan pengeditan dokumen memanfaatkan *Library Working with Uploaded Files* di

CodeIgniter 4 merupakan library utama yang digunakan untuk menangani proses pengunggahan file dari *form* ke server pada aplikasi e-arsip. *Library* ini memudahkan proses pengaturan, validasi, dan penyimpanan file yang diunggah. Fungsi yang disediakan oleh library untuk menangani proses pengunggahan file dengan lebih aman dan efisien. Form untuk upload file dibuat di bagian view, di mana validasi awal dilakukan yaitu tipe file. Seperti pada gambar 3.35.

```
<div class="form-group">
  <p>Silakan upload file dokumen dalam format PDF.</p>
  <label for="file">Upload File: <span class="wajib-diisi">* Wajib diisi</span></label>
  <input type="file" class="form-control-file" id="file" name="file" accept=".pdf" required>
</div>
```

Gambar 3. 35 Form upload file

Controller akan menerima file yang upload, mengatur validasi memastikan bahwa semua data yang diperlukan telah diisi dengan benar, implementasi validasi data dapat dilihat pada gambar 3.36.

```
// Validasi input
if (!$this->validate([
  'judul_dokumen' => 'required',
  'tahun' => [
    'required',
    'regex_match[/^\d{4}$/]', // Validasi untuk format tahun (4 digit angka)
  ],
  'jenis' => 'required',
  'nama_kategori' => 'required',
  'file' => [
    'uploaded[file]',
    'mime_in[file,application/pdf]',
    'max_size[file,100000]',
  ],
  // 'keterangan' => 'required'
])) {
  // Menampilkan pesan error validasi
  $errors = $this->validator->getErrors();
  return redirect()->to('/admin/tambahdoc')->withInput()->with('error', 'Gagal validasi: ' . implode(', ', $errors));
}
```

Gambar 3. 36 implementasi code validasi data

Setelah validasi, file yang diupload diambil dan diperiksa untuk memastikan file tersebut sesuai dan valid. Implementasi dapat dilihat pada gambar 3.37.

```
// Ambil file yang diupload
$file = $this->request->getFile('file');

// Debugging file
if (!$file->isValid()) {
  $error = $file->getErrorString() . ' (' . $file->getError() . ')';
  return redirect()->to('/admin/tambahdoc')->withInput()->with('error', 'File tidak valid: ' . $error);
}
```

Gambar 3. 37 implementasi code validasi file

dan Setelah file valid, file dipindahkan ke lokasi yang ditentukan. Implementasi dapat dilihat pada gambar 3.38.

```
// Pindahkan file ke direktori tujuan
$file->move(ROOTPATH . 'public/fileArsip', $fileName);
```

Gambar 3. 38 implementasi code penentuan lokasi file

Data dokumen yang valid, disimpan ke database menggunakan model DokumenModel untuk memudahkan pengaksesan dan pengelolaan di masa mendatang.

3.6.7. Jenis Dokumen

Jenis dokumen ditampilkan di *view* pada *sidebar* dan hasil *filter* dokumen pada main content kepada pengguna. *Controller* mengelola alur penyaringan dan penampilan daftar dokumen berdasarkan jenis yang dipilih. Model berinteraksi dengan tabel *jenisdokumen* untuk mengambil jenis dokumen yang sesuai untuk ditampilkan kembali pada *view*.

3.6.8. Kategori Dokumen

Pengelolaan kategori dokumen memungkinkan pengguna untuk mengatur kategori dokumen. *view* menampilkan *form* dan daftar kategori dokumen kepada pengguna dalam bentuk tabel. *Controller* mengelola logika untuk operasi *CRUD* (*Create, Read, Update, Delete*) pada kategori dokumen. Model berinteraksi dengan tabel *kategoridokumen* untuk mengelola data kategori sesuai dengan operasi yang dilakukan oleh *controller*.

3.6.9. Profile

- *Admin*

Halaman profil sebagai *Admin* memungkinkan *Admin* untuk melihat dan mengedit informasi pribadi serta mengelola pengguna yang menggunakan aplikasi seperti menambah, mengedit dan menghapus akun pengguna. *view* menampilkan informasi *Admin* pada *sidebar*, informasi pengguna lain pada main content dalam format tabel. *Controller* menangani alur untuk melihat dan memperbarui informasi pribadi serta mengelola pengguna lain sesuai dengan model yang berinteraksi dengan tabel *User* untuk mengelola data profil dan akun pengguna. Pada kasus ini *Library*

Validation rules dimanfaatkan untuk memvalidasi inputan saat mengelola pengguna dari aplikasi baik penambahan, edit ataupun dalam mengubah data diri.

- *User/Staf*

Halaman profil pengguna memungkinkan pengguna untuk melihat dan mengedit informasi pribadi. *view* menampilkan informasi pribadi pengguna *User* di main content. Pengguna dapat melihat dan mengedit informasi pribadi mereka. *Controller* mengelola alur untuk melihat dan memperbarui informasi pribadi dengan model yang berinteraksi dengan tabel *User* untuk mengelola data profil pengguna. Sama halnya dengan admin *Library Validation rules* dimanfaatkan untuk memvalidasi inputan saat mengubah data diri.

3.6.10. Logout/Sign Out

Proses logout/*Sign out* memastikan bahwa sesi pengguna diakhiri dengan benar. *Controller* menangani logika untuk mengakhiri sesi pengguna dan mengarahkan ke halaman beranda. *view* menampilkan halaman beranda sebagai umpan balik bahwa sesi telah diakhiri. *Library Session destroy()* dimanfaatkan untuk menghapus sesi saat ini. Implementasi dapat dilihat pada gambar 3.39.

```

public function logout()
{
    // Hapus semua data sesi
    session()->destroy();

    // Arahkan kembali ke halaman beranda
    return redirect()->to('/');
}

```

Gambar 3. 39 implementasi Library Session destroy()

BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil Implementasi Antarmuka Pengguna

Proses membangun sistem terdiri dari tahap implementasi, sesuai dengan desain dan rancangan sistem yang telah dirancang atau dibuat sebelumnya. Sehingga menghasilkan aplikasi yang siap untuk digunakan. Pengguna yang dapat mengakses aplikasi yaitu *Admin* dan staf.

4.1.1 Implementasi Beranda

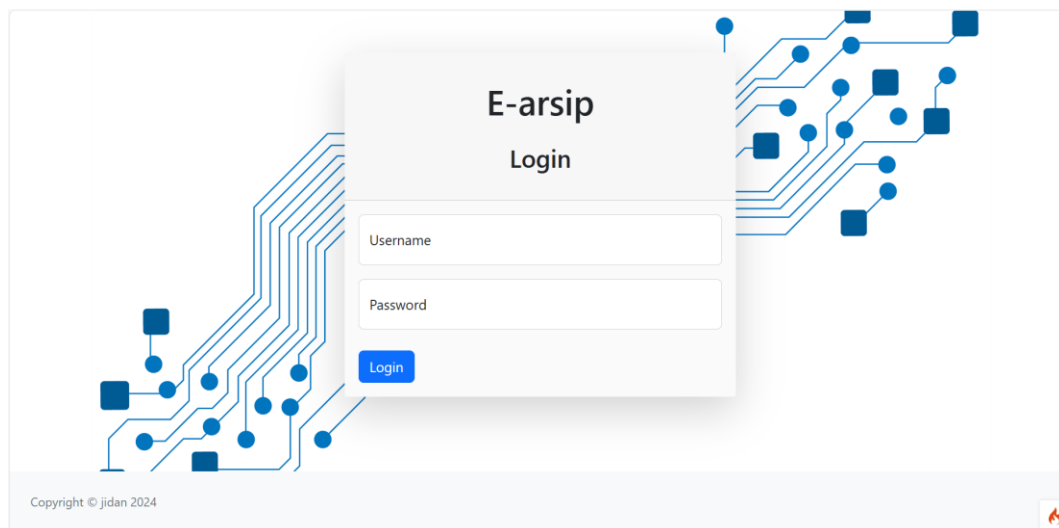
Beranda adalah halaman yang tampil saat aplikasi pertama kali diakses oleh admin atau staf. Ketika pengguna menekan *Login* pada halaman beranda maka akan diarahkan kehalaman login. Halaman beranda dapat dilihat pada gambar 4.1.



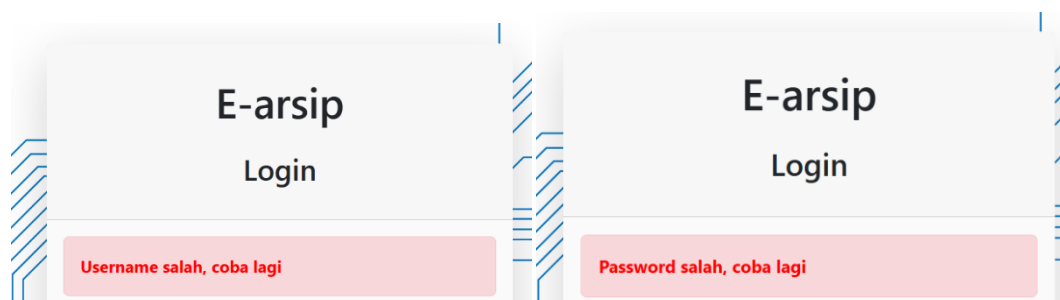
Gambar 4. 1 Implementasi Beranda

4.1.2 Implementasi Login

Halaman login adalah halaman yang digunakan untuk mengakses aplikasi dengan memasukkan *Username* dan *password*. Halaman login akan memberikan akses sesuai dengan akses dimiliki pengguna (*Admin* atau *User*). Halaman login dapat dilihat pada gambar 4.2. Ketika login yang dilakukan berhasil maka akan menuju ke halaman *dashboard* dan jika login yang dilakukan gagal maka akan muncul pesan bahwa login yang dilakukan tidak berhasil. Pesan kesalahan dapat dilihat pada gambar 4.3.



Gambar 4. 3 Implementasi Login



Gambar 4. 2 Pesan Kesalahan Setelah Gagal Login

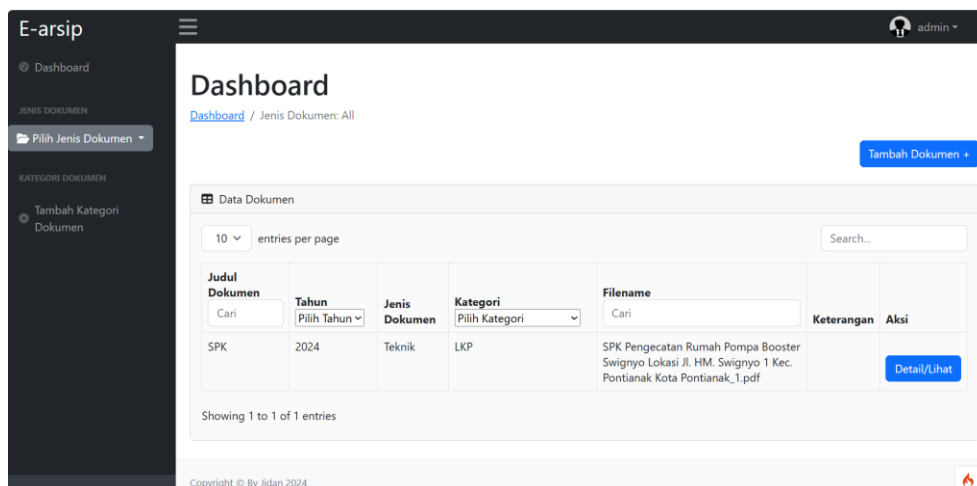
4.1.3 Implementasi *dashboard* (Admin dan Staf)

Halaman dashboard adalah halaman yang ditampilkan setelah pengguna berhasil melakukan login baik sebagai Admin ataupun Staf. Pada halaman dashboard menampilkan berbagai informasi. Pada bagian kiri menampilkan opsi di *sidebar* termasuk dashboard, yang mengarahkan pengguna ke halaman utama dashboard, serta kategori dokumen seperti Dokumen Teknik, Dokumen Administrasi, dan Dokumen Pelayanan, yang akan menampilkan dokumen sesuai dengan kategori yang dipilih. Selain itu, terdapat opsi Tambah Kategori Dokumen yang akan mengarahkan pengguna ke halaman untuk menambah kategori dokumen baru.

Pada bagian tengah atau *main content*, menampilkan tombol tambah dokumen, tabel dari semua dokumen, kolom pencarian untuk judul dokumen, kolom sortir berdasarkan tahun, kolom sortir berdasarkan kategori dokumen, kolom

pencarian berdasarkan nama file dan kolom yang berisi tombol detail/lihat yang akan mengarahkan pengguna ke halaman lihat dokumen.

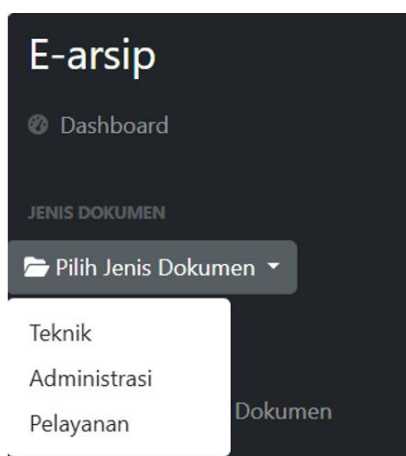
Pada bagian pojok kanan atas *navbar* menampilkan gambar profil user dengan dropdown menu yang memungkinkan pengguna untuk mengakses profil atau *logout/sign out*. Halaman dashboard dapat dilihat pada gambar 4.4.



Gambar 4. 4 Implementasi *dashboard* (Admin dan Staf)

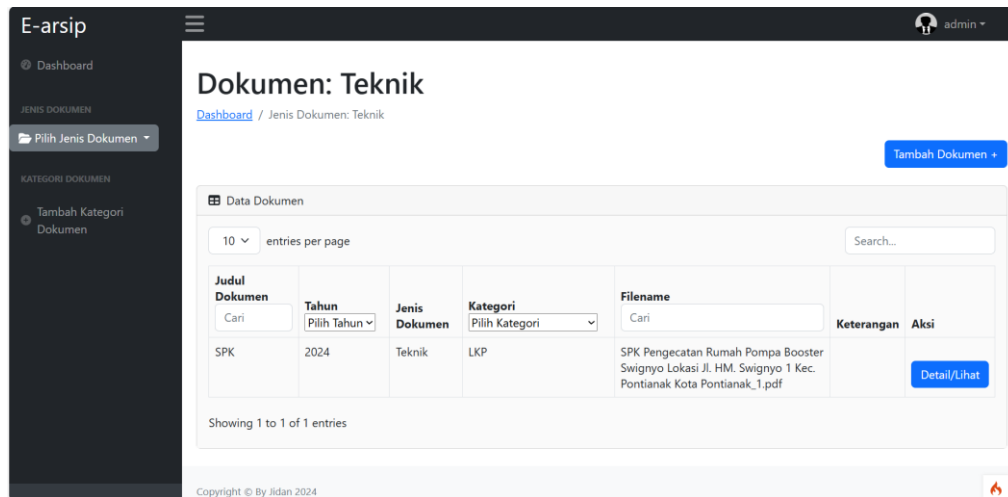
4.1.4 Implementasi Menu Dokumen Teknik, Dokumen Administrasi, dan Dokumen Pelayanan (Admin dan Staf)

Halaman dokumen teknik, dokumen Administrasi, dan dokumen pelayanan adalah halaman yang ditampilkan jika pengguna memilih jenis dokumen pada *sidebar*. Ketika admin atau staf menekan menu pilih jenis dokumen maka akan ditampilkan sub menu yang berisi dokumen teknik, dokumen Administrasi, dan dokumen pelayanan. Menu ditampilkan pada gambar 4.5.



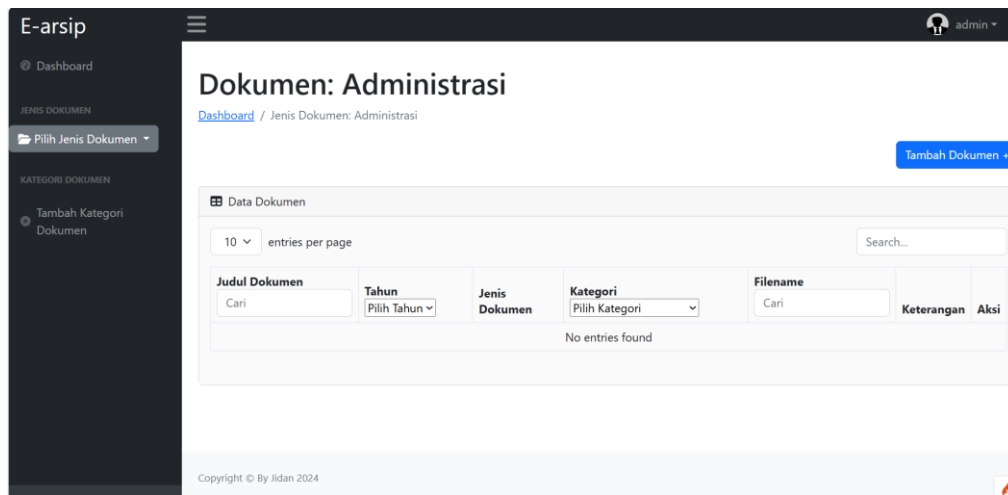
Gambar 4. 5 Menu Jenis Dokumen (Admin dan Staf)

Admin atau staf memilih dokumen dengan jenis dokumen teknik, maka pada main content akan menampilkan tabel yang khusus memuat dokumen teknik. Seperti pada gambar 4.6.



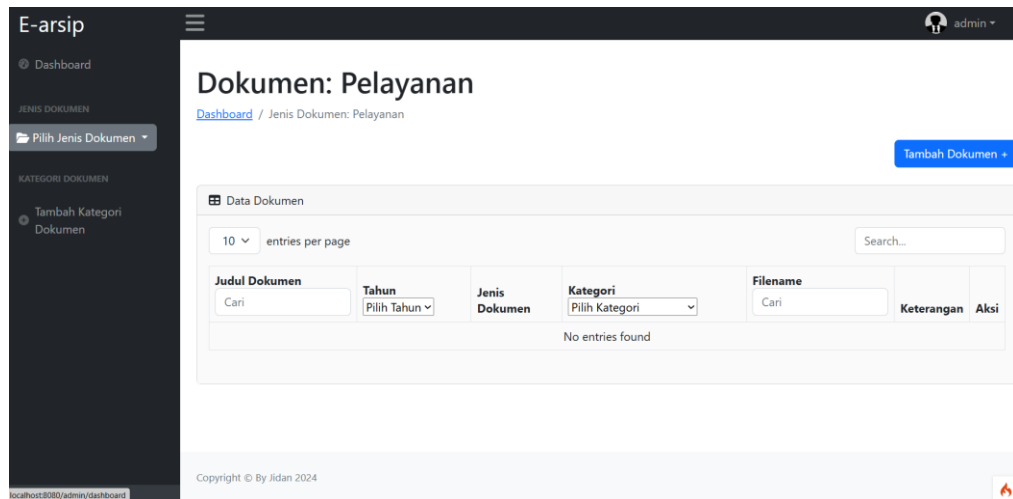
Gambar 4. 6 Tampilan Tabel Dokumen Dengan Jenis Teknik (*Admin dan Staf*)

Pengguna memilih dokumen dengan jenis dokumen *Administrasi*, maka pada main content akan menampilkan tabel yang khusus memuat dokumen *Administrasi*. Seperti pada gambar 4.7.



Gambar 4. 7 Tampilan Tabel Dokumen Dengan Jenis Administrasi (*Admin dan Staf*)

Pengguna memilih dokumen dengan jenis dokumen Pelayanan, maka pada main content akan menampilkan tabel yang khusus memuat dokumen pelayanan . Seperti pada gambar 4.8.



Gambar 4. 8 Tampilan Tabel Dokumen Dengan Jenis Pelayanan (*Admin* dan Staf)

4.1.5 Implementasi Tambah Kategori (*Admin* dan Staf)

Halaman tambah kategori merupakan halaman yang ditampilkan ketika admin atau staf memilih atau menekan tambah kategori dokumen yang terdapat pada bagian kiri atau *sidebar*. Menu dapat dilihat pada gambar 4.9.

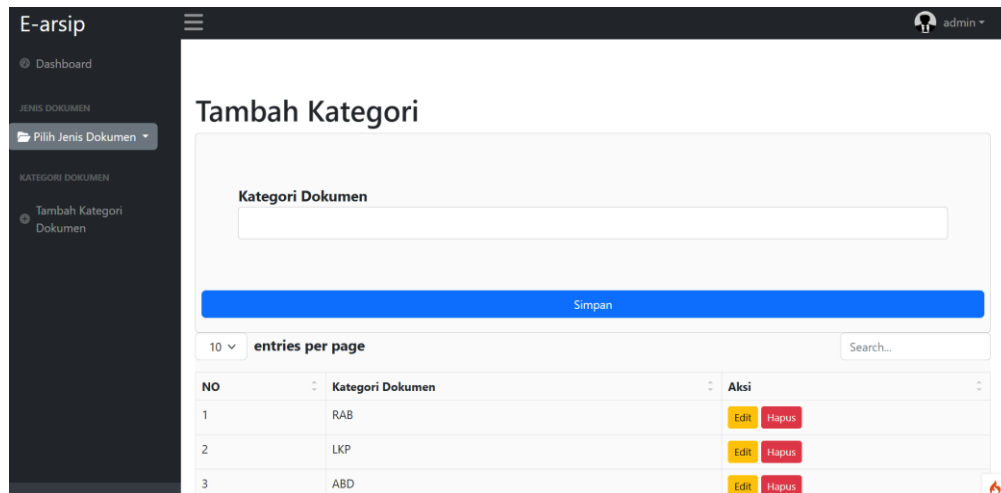


Gambar 4. 9 Menu Tambah Kategori Dokumen (*Admin* dan Staf)

Pada bagian tengah atau main content dari halaman tambah kategori menampilkan *form* penambahan kategori serta tabel yang menampilkan daftar kategori yang sudah ditambahkan dan kolom yang berisi tombol edit dan hapus.

pengguna menambahkan kategori dokumen baru dengan mengisi *form* yang disediakan, kemudian menekan tombol simpan untuk menyimpan kategori ke dalam database. Ketika berhasil disimpan pengguna akan diarahkan kembali kehalaman tambah kategori dokumen.

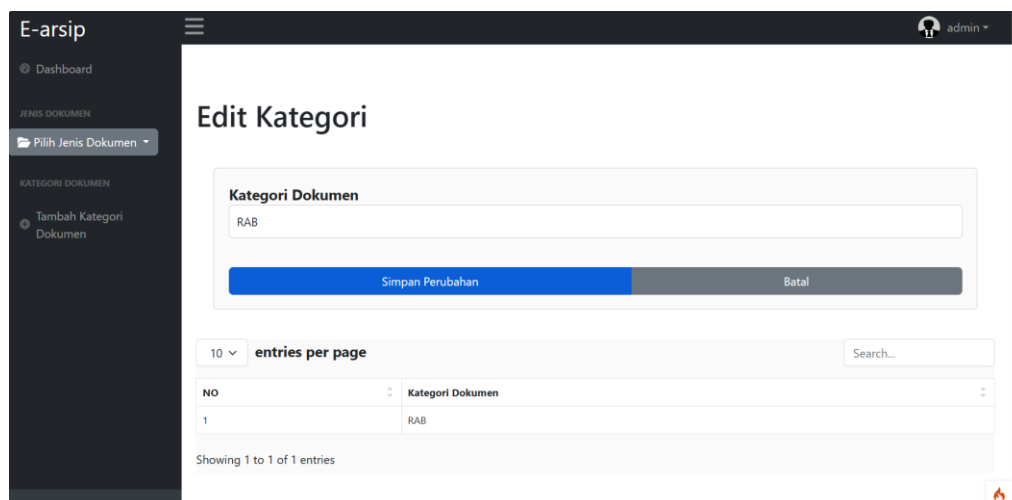
pengguna menekan tombol *edit* pengguna akan diarahkan kehalaman *form edit* kategori. pengguna menekan tombol hapus, menghapus kategori dokumen dari sistem setelah konfirmasi. Tambah kategori dapat dilihat pada gambar 4.10.



Gambar 4. 10 Implementasi Tambah Kategoti (*Admin* dan Staf)

4.1.6 Implementasi Edit Kategori (*Admin* dan Staf)

Halaman edit kategori adalah halaman yang ditampilkan saat pengguna menekan tombol edit yang terdapat pada kolom didalam tabel. Pada bagian tengah atau main content halaman edit kategori menampilkan *form* yang digunakan untuk merubah kategori dari dokumen. Tombol simpan digunakan untuk menyimpan perubahan, dan diarahkan kembali ke halaman untuk menambah kategori dokumen. serta tombol batal jika tidak melakukan perubahan. dan tabel yang berisi kategori yang akan dirubah. Halaman edit kategori dapat dilihat pada gambar 4.11.

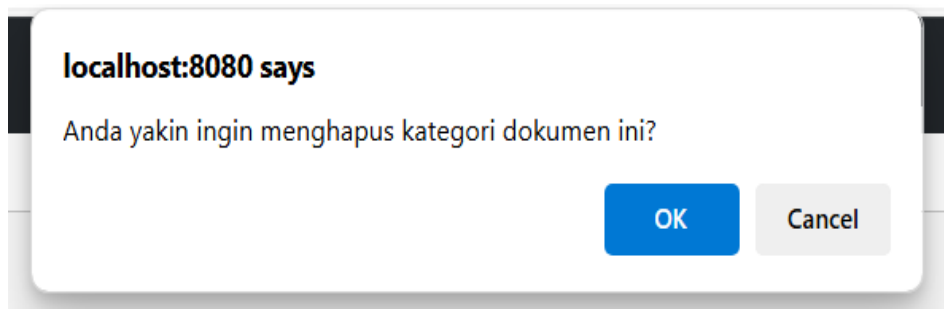


Gambar 4. 11 Implementasi Edit Kategori (*Admin* dan Staf)

4.1.7 Implementasi Hapus Kategori (*Admin* dan Staf)

Hapus kategori tampil saat pengguna menekan tombol hapus yang terdapat pada kolom didalam tabel. Halaman hapus kategori akan menampilkan pesan atau

validasi bahwa pengguna akan menghapus kategori tersebut. Halama hapus dapat dilihat pada gambar 4.12.



Gambar 4. 12 Implementasi Hapus Kategori (*Admin* dan Staf)

Jika pengguna menekan “OK” maka kategori akan terhapus dengan syarat bahwa kategori tersebut tidak terkait dengan sebuah dokumen. Dan jika kategori terkait dengan dokumen tertentu maka kategori tidak dapat dihapus dan diarahkan kehalaman tambah kategori, serta akan ditampilkan pesan seperti pada gambar 4.13.

Tidak dapat menghapus kategori dokumen ini karena masih terdapat dokumen yang menggunakan kategori ini.

Gambar 4. 13 Pesan Gagal Menghapus Kategori (*Admin* dan Staf)

4.1.8 Implementasi Tambah Dokumen (*Admin* dan Staf)

Halaman tambah dokumen ditampilkan saat pengguna menekan tombol tambah dokumen pada dashboard. Pada bagian tengah atau *main content* dari halaman tambah dokumen menampilkan *form* penambahan dokumen yang berisi judul dokumen, tahun dari dokumen, jenis dari dokumen, file dokumen dan keterangan jika diperlukan. Pengguna menekan tombol simpan untuk menyimpan dokumen dan diarahkan kembali kehalaman tambah dokumen. Jika pengguna menekan tombol batal mengarahkan pengguna kembali ke halaman sebelumnya tanpa menyimpan data. Gambar halaman tambah dokumen dapat dilihat pada gambar 4.14.

Gambar 4. 14 Implementasi Tambah Dokumen (*Admin* dan Staf)

4.1.9 Implementasi Lihat Dokumen (*Admin*)

Halaman lihat dokumen ditampilkan saat pengguna dengan akses *admin* menekan tombol Detail/Lihat pada tabel yang berisi dokumen pada dashboard. Pada bagian kiri atau *sidebar* berisi data detail dari dokumen yang diupload seperti judul dokumen, nama pengunggah/pengedit, nama dokumen, ukuran dokumen, tahun, waktu upload, waktu update, jenis dokumen, kategori dokumen dan keterangan dokumen.

Pada bagian tengah atau *main content* halaman lihat dokumen menampilkan preview dokumen, dan dapat menekan tombol kembali untuk kembali ke halaman dashboard, tombol edit untuk membuka *form* edit dokumen, hapus untuk menghapus dokumen dari sistem setelah konfirmasi, download untuk mengunduh file dokumen yang diupload, dan cetak untuk mencetak dokumen, pengguna diarahkan ke tab baru dan ditampilkan. Halaman lihat dokumen dapat dilihat pada gambar 4.15.

Gambar 4. 15 Implementasi Lihat Dokumen (*Admin*)

4.1.10 Implementasi Lihat Dokumen (Staf)

Halaman lihat dokumen ditampilkan saat staf menekan tombol Detail/Lihat pada tabel yang berisi dokumen pada dashboard. pada bagian kiri atau sidebar berisi data detail dari dokumen yang diupload seperti judul dokumen, nama pengunggah/pengedit, nama dokumen, ukuran dokumen, tahun, waktu upload, waktu update, jenis dokumen, kategori dokumen dan keterangan dokumen.

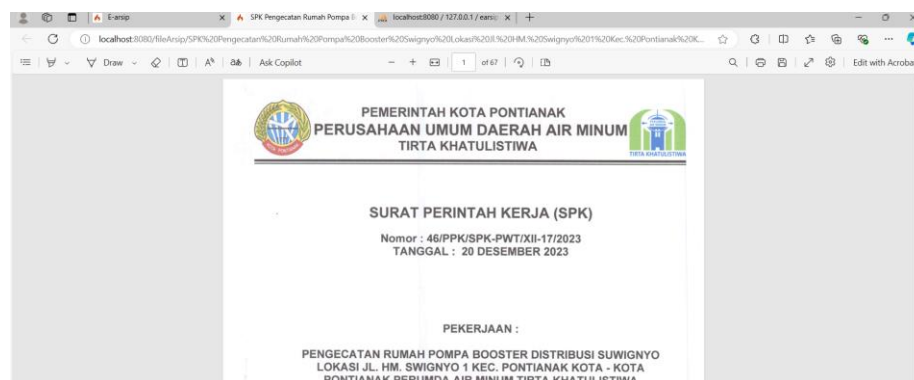
Pada bagian tengah atau main content halaman lihat dokumen menampilkan preview dokumen, dan dapat menekan tombol kembali untuk kembali ke halaman dashboard, dan tombol edit untuk membuka form edit dokumen. Halaman lihat dokumen dapat dilihat pada gambar 4.16.



Gambar 4. 16 Implementasi Lihat Dokumen (Staf)

4.1.11 Implementasi Cetak Dokumen (Admin)

Halaman cetak dokumen akan ditampilkan saat *admin* menekan tombol cetak pada halaman lihat dokumen. Dimana cetak dokumen akan halaman tau tab baru pada browser yang digunakan baru selanjutnya dicetak. Gambar halaman ceak dapat dilihat pada gambar 4.17.



Gambar 4. 17 Implementasi Cetak Dokumen (Admin)

4.1.12 Implementasi Edit Dokumen (*Admin dan Staf*)

Halaman edit dokumen ditampilkan setelah pengguna menekan tombol edit pada halaman lihat dokumen. Pada bagian kiri atau *sidebar* kembali menampilkan menu yang berisi *dashboard*, jenis dokumen dan tambah Kategori dokumen.

pada bagian tengah atau *main content* menampilkan *form* untuk merubah data atau file dokumen yang sudah ditambahkan sebelumnya. Setelah mengisi atau mengubah data yang diperlukan, pengguna dapat menekan tombol simpan untuk menyimpan perubahan tersebut kemudian pengguna akan diarahkan kembali ke halaman detail/lihat dokumen. Jika pengguna menekan tombol batal mengarahkan pengguna kembali ke halaman sebelumnya tanpa menyimpan data. Halaman edit dokumen dapat dilihat pada gambar 4.17.

Gambar 4. 18 Implementasi Edit Dokumen (*Admin dan Staf*)

4.1.13 Implementasi *Download* Dokumen (*Admin*)

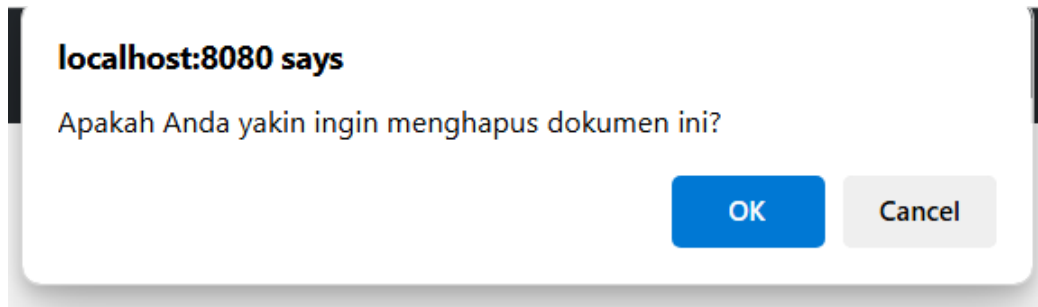
Download dokumen berfungsi saat *admin* menekan tombol *download* pada halaman lihat dokumen, maka secara otomatis akan *mendownload* dokumen yang telah dipilih seperti pada gambar 4.18.



Gambar 4. 19 Implementasi Download Dokumen (*Admin dan Staf*)

4.1.14 Implementasi Hapus Dokumen (*Admin*)

Hapus dokumen tampil saat *admin* menekan tombol hapus yang terdapat pada halaman lihat dokumen. Hapus dokumen akan menampilkan pesan atau validasi bahwa pengguna akan menghapus dokumen yang telah dipilih tersebut. Jika pengguna menekan ok maka pengguna akan diarahkan kembali kehalaman *dashboard*. Validasi hapus dapat dilihat pada gambar 4.19.



Gambar 4. 20 Implementasi Hapus Dokumen (*Admin* dan Staf)

4.1.15 Implementasi Menu *Profile* dan *Sign out* (*Admin* dan Staf)

Menu *Profile* dan *Sign out* akan tampil saat pengguna menekan gambar profile yang terdapat di bagian pojok kanan atas dari *navbar*. Pada menu ini menampilkan gambar profile *username*, nama, akses, dan nip serta tombol *profile* dan *sign out*. Ketika pengguna menekan *profile* maka pengguna akan diarahkan kehalaman *profile* sesuai dengan akses yang dimiliki baik sebagai *admin* atau *user*(staf). Dan saat pengguna menekan tombol *Sign out* maka pengguna telah mengakhiri sesi dan akan diarahkan ke halaman beranda. Menu *Profile* dan *Sign out* dapat dilihat pada gambar 4.20.



Gambar 4. 21 Implementasi Menu *Profile* dan *Sing Out* (*Admin* dan Staf)

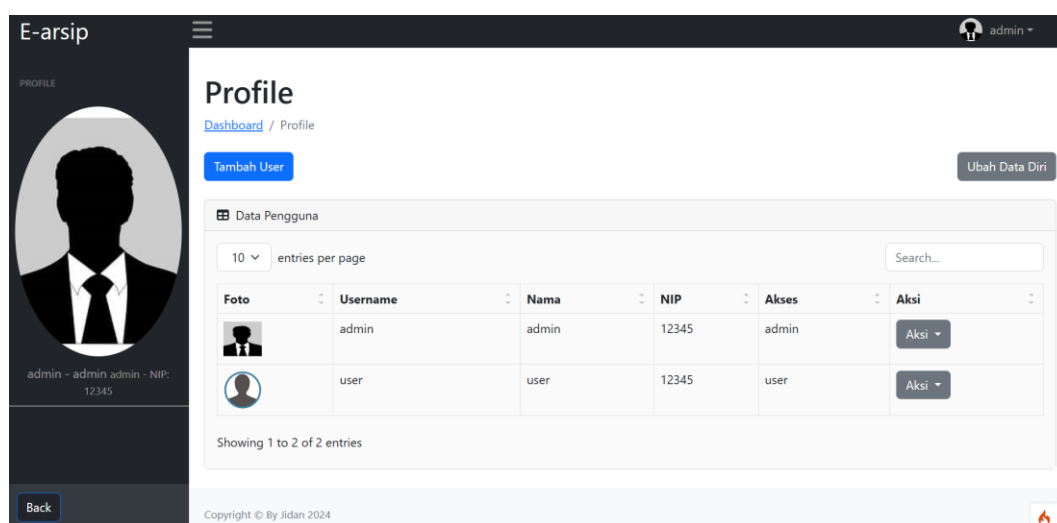
4.1.16 Implementasi *Profile (Admin)*

Halaman *profile admin* ditampilkan saat pengguna yang memiliki akses sebagai *admin* menekan tombol *Profile* yang berada pada pojok kanan atas pada *navbar*. Halaman *profile admin* menampilkan informasi diri sendiri dan informasi pengguna lain.

Pada bagian kiri atau *sidebar* terdapat gambar profil jika tekan *admin* maka akan ditampilkan *form* untuk mengubah gambar profil. *Sidebar* juga menampilkan informasi dari *admin* seperti *Username*, nama, akses. Pada bagian pojok kiri bawah *sidebar* terdapat tombol *back* yang mengarahkan *admin* kembali kehalaman *dashboard*.

Pada bagian tengah atau *main content* menampilkan tombol tambah *user* jika tekan akan mengarahkan *admin* kehalaman *form* untuk menambah pengguna atau akun. *main content* juga menampilkan tombol *ubah data diri* jika ditekan akan mengarahkan *admin* kehalaman *form* untuk mengubah data diri.

Tabel Pada *main content* memuat data dari akun atau pengguna dari aplikasi dan kolom aksi yang memiliki menu *edit* dan *hapus*. Aksi untuk *edit* yang mengarahkan pengguna ke halaman *edit pengguna* dan *hapus* akan menghapus akun dari pengguna setelah dikonfirmasi. Halaman *profile admin* dapat dilihat pada gambar 4.21.



Gambar 4. 22 Implementasi *Profile Admin*

4.1.17 Implementasi Tambah Pengguna (*Admin*)

Halaman tambah pengguna ditampilkan saat pengguna yang memiliki akses sebagai *admin* menekan tombol tambah *user* pada main content *profile admin*. Halaman tambah pengguna menampilkan sebuah *form* penambahan pengguna baru ditampilkan pada bagian tengah atau *main content*, *form* diisi dengan data *username*, nama, *password*, verifikasi *password* untuk memastikan *password* benar, NIP, dan akses dari pengguna apakah sebagai *admin* atau *use(staf)*. *Admin* menekan tombol simpan untuk menyimpan akun atau pengguna baru. Jika berhasil, berhasil atau gagal akan muncul notifikasi berhasil atau gagal dan *admin* diarahkan kehalaman Tambah user. Jika *admin* menekan tombol batal mengarahkan *admin* kembali ke halaman sebelumnya tanpa menyimpan data. Halaman tambah pengguna dapat dilihat pada gambar 4.22.

Gambar 4. 23 Implementasi Tambah Pengguna

4.1.18 Implementasi Ubah Data Diri (*Admin*)

Halaman ubah data diri ditampilkan saat pengguna yang memiliki akses sebagai *admin* menekan tombol ubah data diri pada main content *profile admin*. Halaman ubah data diri menampilkan sebuah *form* yang digunakan untuk merubah data diri yang ditampilkan pada bagian tengah atau *main content*, *form* diisi dengan informasi pribadi yang baru seperti *username*, nama, *password* (isi *password* baru jika ingin memperbaharui *password*), verifikasi *password* untuk memastikan *password* benar dan tidak terjadi kekeliruan, dan NIP. *Admin* menekan tombol simpan. Jika penyimpanan data berhasil atau gagal, akan muncul pesan berhasil

ataupun gagal, *admin* akan diarahkan ke halaman profil *admin*. Jika *admin* menekan tombol batal mengarahkan *admin* kembali ke halaman sebelumnya tanpa menyimpan data. Halaman ubah data diri *admin* dapat dilihat pada gambar 4.23.

Gambar 4. 24 Implementasi Ubah Data Diri Admin

4.1.19 Implementasi Edit Pengguna (*Admin*)

Halaman edit pengguna ditampilkan saat pengguna yang memiliki akses sebagai *admin* menekan tombol edit dikolom aksi pada tabel data pengguna pada *main content profile admin*. Halaman edit pengguna menampilkan sebuah *form* yang digunakan untuk merubah atau mengedit data pengguna yang ditampilkan pada bagian tengah atau main content, *admin* dapat merubah data informasi pengguna dengan merubah data, username, nama, *password* (isi *password* baru jika ingin memperbaharui *password*), verifikasi *password* untuk memastikan *password* benar dan tidak terjadi kekeliruan, NIP, dan akses. *admin* menekan tombol simpan. Jika berhasil atau gagal, akan muncul notifikasi berhasil atau gagal diubah dan diarahkan ke halaman *profile admin*. Jika *admin* menekan tombol batal mengarahkan *admin* kembali ke halaman sebelumnya tanpa menyimpan data. Halaman edit pengguna dapat dilihat pada gambar 4.24.

Gambar 4. 25 Implementasi Edit Pengguna

4.1.20 Implementasi *Profile* (Staf)

Halaman *profile* staf ditampilkan saat pengguna yang memiliki akses sebagai *user* menekan tombol *Profile* yang berada pada pojok kanan atas pada *navbar*. Halaman *Profile* staf menampilkan informasi diri sendiri. Pada bagian kiri menampilkan *sidebar* menampilkan opsi *dashboard* yang akan mengarahkan pengguna ke halaman *dashboard*. Jenis dokumen mengarahkan pengguna ke halaman yang menampilkan dokumen sesuai dengan jenis yang dipilih. Tambah kategori dokumen mengarahkan pengguna ke halaman untuk menambah kategori dokumen.

Pada bagian tengah atau *main content* menampilkan gambar profil dari staf, jika ditekan form untuk mengubah gambar profil akan ditampilkan. *main content* juga menampilkan tombol ubah data diri, jika ditekan akan mengarahkan *admin* ke halaman *form* untuk mengubah data diri. Tabel pada *main content* menampilkan data diri. Halaman *Profile* staf dapat dilihat pada gambar 4.25.

Foto	Username	Nama	NIP	Akses
	user	user	12345	user

Gambar 4. 26 Implementasi Profile Staf

4.1.21 Implementasi Ubah Data Diri (Staf)

Halaman ubah data diri ditampilkan saat pengguna yang memiliki akses sebagai staf menekan tombol ubah data diri pada main content *profile* staf. Halaman ubah data diri menampilkan sebuah *form* yang digunakan untuk merubah data diri. Pada bagian kiri atau *sidebar* terdapat gambar profil dan informasi seperti *username*, nama, akses. Pada bagian pojok kiri bawah *sidebar* terdapat tombol back yang mengarahkan staf kembali kehalaman dashboard.

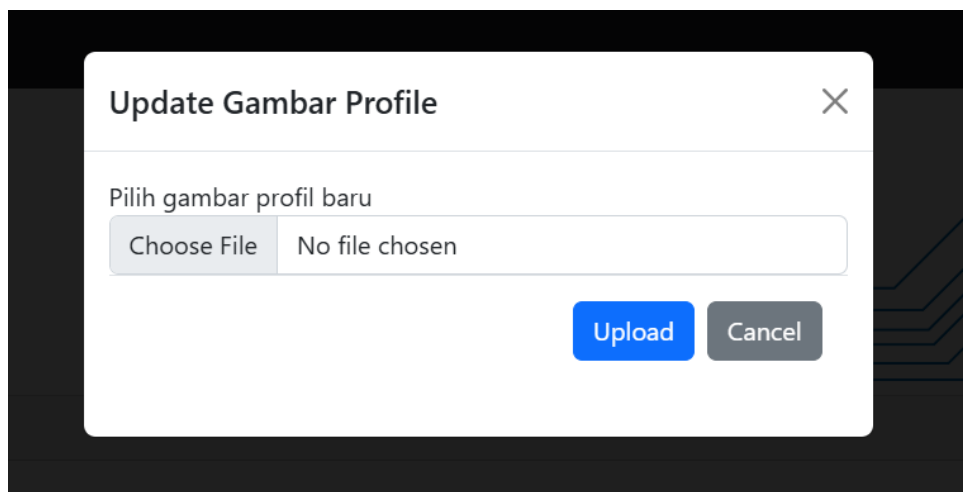
pada bagian tengah atau *main content*, *form* diisi dengan informasi pribadi yang baru seperti *username*, nama, *password* (isi *password* baru jika ingin memperbaharui *password*), verifikasi *password* untuk memastikan pasword benar dan tidak terjadi kekeliruan, dan NIP. Staf menekan tombol simpan. Jika penyimpanan data berhasil atau gagal, akan muncul pesan berhasil ataupun gagal, staf akan diarahkan ke halaman profil staf. Jika staf menekan tombol batal mengarahkan staf kembali ke halaman sebelumnya tanpa menyimpan data. Halaman ubah data diri staf dapat dilihat pada gambar 4.26.

Gambar 4. 27 Implementasi Ubah Data Diri Staf

4.1.22 Implementasi Ubah Gambar Profile (Admin dan staf)

Form ubah gambar *Profile* ditampilkan saat pengguna dengan akses sebagai *Admin* maupun *User*(staf) mengklik gambar yang ditampilkan di *sidebar* jika akses sebagai *Admin* dan di main content jika akses sebagai *User*(staf). Pengguna *admin* atau staf mengupload gambar profil baru, dan menekan simpan, jika proses penyimpanan berhasil atau gagal, pengguna akan menerima notifikasi bahwa gambar berhasil atau gagal diubah dan kemudian diarahkan kembali ke halaman

profil sesuai dengan akses yang dimiliki. Gambar halaman update gambar *Profile* dapat dilihat pada gambar 4.27.



Gambar 4. 28 implementasi Ubah Gambar Profile (*Admin* dan Staf)

4.2. Pengujian Aplikasi

Tahap penting dalam proses pembangunan aplikasi adalah pengujian aplikasi. E-arsip aplikasi berbasis web pada bagian perencanaan dan pengelolaan aset perumda air minum tirta khatulistiwa kota Pontianak. Tujuan dari pengujian ini adalah untuk memastikan bahwa fitur aplikasi berjalan sesuai dengan ekspektasi, beroperasi dengan baik, dan memenuhi kebutuhan pengguna. Hasil dari pengujian aplikasi yang dilakukan adalah sebagai berikut:

1. Pengujian *Black Box*

Pengujian *Black Box* digunakan untuk menguji aplikasi untuk menguji spesifikasi fungsional tanpa menguji desain atau kode program. Pengujian *Black Box* adalah teknik pengujian tanpa mengacu pada struktur internal dari komponen atau sistem. Tujuan pengujian ini adalah untuk mengetahui apakah fungsi, masukan, dan keluaran aplikasi memenuhi spesifikasi. Pendekatan yang dilakukan dimana pengujian dilakukan dari sudut pandang pengguna luar yang tidak tahu tentang internal aplikasi. Dalam hal ini, *XAMPP* digunakan sebagai lokal untuk pengujian stabilitas blackbox. Hasil pengujian dapat dilihat pada tabel 4.1

Tabel 4. 1 Pengujian Aplikasi Dengan Black Box

No	Fitur			Hasil
----	-------	--	--	-------

		Skenario Pengujian	Hasil yang diharapkan	ya	tidak
1	Login Dan Akses Pengguna	Input benar	Login berhasil dan diarahkan ke <i>dashboard</i>	√	
		Input <i>Username</i> salah	Sistem menampilkan pesan error " <i>Username</i> salah, coba lagi"	√	
		Input <i>password</i> salah	Sistem menampilkan pesan error " <i>Password</i> salah, coba lagi"	√	
		Input kosong (<i>Username/password</i>)	Sistem menampilkan pesan " please fill out this field dalam bahasa indonesia (silahkan isi kolom ini)"	√	
		Cek peran pengguna (<i>Admin</i>)	<i>Admin</i> dapat mengakses semua halaman	√	
		Cek peran pengguna (<i>User</i>)	<i>User</i> dapat mengakses semua halaman kecuali manajemen pengguna(menambah, mengedit,menghapus <i>User/pengguna</i>)	√	
2	Logout/Sign Out	Klik tombol sign out	Pengguna berhasil logout dan diarahkan ke halaman beranda.	√	
		Akses ulang setelah logout	Pengguna diarahkan ke halaman login	√	

			ketika mencoba mengakses halaman <i>dashboard</i> setelah logout.		
3	Control Access	<i>Admin</i> mengakses halaman <i>User</i>	<i>Admin</i> tidak dapat mengakses halaman dengan akses <i>User</i> , <i>Admin</i> diarahkan ke login, dan muncul pesan "Anda tidak memiliki akses"	√	
		<i>User</i> mengakses halaman <i>Admin</i>	<i>User</i> tidak dapat mengakses halaman dengan akses <i>Admin</i> , <i>User</i> diarahkan ke login, dan, muncul pesan "Anda tidak memiliki akses"	√	
4	Manajemen Dokumen	Tambah dokumen baru dengan data lengkap	Dokumen berhasil disimpan dan muncul di daftar dokumen.	√	
		Tambah dokumen tanpa, mengisi nama, tahun, jenis atau kategori	Sistem menampilkan pesan" please fill out this field dalam bahasa indonesia (silahkan isi kolom ini)"	√	
		Tambah dokumen dengan format tahun yang salah	Sistem menampilkan pesan"masukan tahun dengan format 4 digit, contoh: 2024"	√	

		Tambah dokumen dengan ukuran file valid	Sistem menampilkan pesan"Gagal validasi: file is too large of a file atau dalam bahasa indonesia(file terlalu besar dari sebuah file)”	√	
		Lihat dokumen	Sistem pindah kehalaman lihat dokumen dan menampilkan dokumen	√	
		Cetak dokumen	Sistem pindah ketab baru dengan menampilkan dokumen yang sesuai dan siap dicetak	√	
		<i>Download</i> dokumen	Dokumen yang <i>download</i> dapat dibuka dan sesuai dengan file yang diunggah	√	
		Edit dokumen dengan mengubah data tertentu seperti nama, tahun, jenis, kategori, atau file	Data dan file dokumen berubah di daftar dokumen setelah disimpan	√	
		Hapus dokumen	Dokumen yang dipilih berhasil dihapus dan tidak lagi muncul di daftar dokumen	√	

		Tampilan daftar dokumen setelah dihapus	Sistem menampilkan pesan konfirmasi sebelum dokumen dihapus dan memperbarui daftar dokumen setelah dihapus	√	
5	Manajemen Jenis Dan Kategori Dokumen	Tampilan jenis dokumen	Jenis dokumen baru muncul di dropdown saat diklik pengguna	√	
		Tambah kategori dokumen baru	Pengguna diarahkan ke halaman tambah kategori yang berisi <i>form</i> dan tabel daftar kategori	√	
		Edit kategori dokumen	Perubahan jenis atau kategori dokumen berhasil disimpan dan diperbarui di daftar pilihan	√	
		Hapus kategori dokumen	kategori dokumen berhasil dihapus jika tidak terkait dengan dokumen dan tidak lagi muncul di daftar pilihan	√	
			kategori dokumen tidak berhasil dihapus jika terkait dengan dokumen dan tidak lagi muncul di daftar pilihan dan muncul	√	

			pesan “Tidak dapat menghapus kategori dokumen ini karena masih terdapat dokumen yang menggunakan kategori ini”		
6	Pencarian Dan Penyortiran Dokumen	Pencarian dokumen berdasarkan nama	Dokumen yang sesuai dengan nama yang dicari muncul di daftar	√	
		Pencarian dokumen berdasarkan nama file	Dokumen yang sesuai dengan nama file yang dicari muncul di daftar	√	
		Penyortiran dokumen berdasarkan jenis	Dokumen disortir sesuai dengan jenis yang dipilih muncul di daftar	√	
		Penyortiran dokumen kategori	Dokumen disortir sesuai dengan kategori yang dipilih muncul di daftar	√	
		Penyortiran berdasarkan tahun	Dokumen disortir sesuai dengan tahun yang dipilih muncul di daftar	√	
		Penyortiran berdasarkan jenis, kategori, dan tahun	Dokumen disortir sesuai dengan jenis, kategori dan tahun yang dipilih muncul di daftar	√	

		Penyortiran berdasarkan kategori dan tahun	Dokumen disortir sesuai dengan kategori dan tahun yang dipilih muncul di daftar	√	
7	Navigasi Dan Antarmuka Pengguna	Klik menu navigasi (<i>dashboard</i>)	Pengguna diarahkan ke halaman <i>dashboard</i> dan data ditampilkan dengan benar	√	
		Klik menu navigasi (berdasarkan jenis dokumen)	Pengguna diarahkan ke halaman dengan data ditampilkan sesuai dengan jenis yang benar	√	
		Klik tombol "tambah dokumen"	Pengguna diarahkan ke halaman <i>form</i> untuk menambahkan dokumen baru	√	
		Klik tombol "tambah kategori"	Pengguna diarahkan ke halaman <i>form</i> untuk menambahkan kategori baru	√	
		Klik tombol "edit kategori"	Pengguna diarahkan ke halaman <i>form</i> untuk mengedit kategori	√	
		Klik gambar avatar pengguna	Pengguna klik <i>Profile</i> dan diarahkan ke halaman <i>profile</i>	√	
			Pengguna klik <i>Sign out</i> dan diarahkan ke halaman beranda	√	

		Klik tombol “detail/lihat”	Pengguna diarahkan ke halaman lihat dokumen dan data ditampilkan dengan benar	√	
		Klik tombol “kembali pada lihat dokumen”	Pengguna diarahkan ke halaman <i>dashboard</i> dan data ditampilkan dengan benar	√	
		Klik tombol “edit pada lihat dokumen”	Pengguna diarahkan ke halaman <i>form</i> untuk mengedit dokumen	√	
8	Feedback Pengguna	Semu operasi yang dilakukan berhasil	Sistem menampilkan pesan sukses	√	
		Semua operasi yang dilakukan gagal	Sistem menampilkan pesan error	√	
9	Profile	Buka halaman <i>profile admin</i>	Profil <i>Admin</i> ditampilkan dengan benar, termasuk data diri dan daftar pengguna	√	
		<i>Admin</i> tidak bisa menghapus akun sendiri	<i>Admin</i> dicegah untuk menghapus akun sendiri. Tombol hapus tidak ditampilkan	√	
		<i>Admin</i> klik tombol “tambah <i>User</i> ”	<i>Admin</i> diarahkan ke halaman <i>form</i> untuk menambah <i>User</i> /pengguna	√	

		<i>Admin</i> tambah pengguna valid	Data pengguna ditampilkan di daftar pengguna	√	
		<i>Admin</i> tambah pengguna nama tidak <i>format</i> huruf	Sistem menampilkan pesan error “The nama field may only contain alphabetical characters dalam bahasa indonesia (Bidang nama hanya boleh berisi karakter alfabet)”	√	
		<i>Admin</i> tambah pengguna <i>password</i> kurang dari 6 karakter	Sistem menampilkan pesan error “The <i>password</i> field must be at least 6 characters in length atau dalam bahasa indonesia (Bidang kata sandi harus terdiri dari minimal 6 karakter)”	√	
		<i>Admin</i> tambah pengguna NIP tidak dalam format angka	Sistem menampilkan pesan error “The nip field must contain only numbers atau dalam bahasa indonesia (Bidang nip hanya boleh berisi angka)”	√	
		<i>Admin</i> klik “edit <i>User</i> ”	<i>Admin</i> diarahkan ke halaman <i>form</i> untuk	√	

			mengedit <i>User/pengguna</i>		
		<i>Admin</i> hapus pengguna	Akun pengguna berhasil dihapus dan tidak lagi muncul di daftar pengguna	√	
		<i>Admin/User</i> Klik tombol “ubah data diri”	<i>Admin</i> diarahkan ke halaman mengubah data diri	√	
		<i>Admin/User</i> upload/ubah foto <i>profile</i>	Foto profil berhasil diubah dan gambar baru muncul di halaman profil	√	
		Buka halaman <i>Profile User</i>	Profil <i>User</i> ditampilkan dengan benar, termasuk data diri dan gambar profil	√	

2. Pengujian fungsional

Setelah tahap pengujian selesai, analisis hasil digunakan untuk mengevaluasi tingkat keberhasilan fungsional sistem; metode *System Usability Scale* (SUS) mengevaluasi tingkat kegunaan sistem berdasarkan umpan balik pengguna. Metode ini menggunakan kuesioner umum yang terdiri dari sepuluh pertanyaan. Pertanyaan yang memiliki nomor ganjil memiliki kalimat positif, sedangkan pertanyaan yang memiliki nomor genap memiliki kalimat negatif. Skala Likert digunakan untuk menilai hasil kuesioner[13]. Pertanyaan dapat dilihat pada tabel 4.2.

Tabel 4. 2 pengujian *System Usability Scale (SUS)*

NO	Pertanyaan
1	Saya merasa aplikasi e-arsip mudah digunakan
2	Saya merasa aplikasi e-arsip terlalu rumit
3	Saya merasa mudah menggunakan aplikasi e-arsip untuk mengelola dokumen
4	Saya memerlukan bantuan orang lain untuk bisa menggunakan aplikasi e-arsip dengan baik
5	Fitur-fitur aplikasi e-arsip terintegrasi dengan baik
6	Fitur-fitur aplikasi e-arsip tidak selaras
7	Saya merasa aplikasi e-arsip mudah dipelajari dan digunakan orang lain
8	Saya merasa aplikasi e-arsip membingungkan dan harus mempelajari banyak hal sebelum menggunakannya dengan efektif
9	Saya merasa mudah untuk mengakses dokumen dengan menggunakan aplikasi e-arsip
10	Saya merasa aplikasi e-arsip terlalu kaku dan monoton

Skala Likert : 1 = Sangat Tidak Setuju, 2 = Tidak Setuju, 3 = Ragu-Ragu, 4 = Setuju, 5 = Sangat Setuju.

Hasil pengujian digunakan untuk mengukur efektivitas aplikasi yang digunakan. Nilai presentase keberhasilan rata-rata dari setiap responden pengujian untuk semua jenis pengguna aplikasi, yang berjumlah 3 orang, digunakan untuk mengukur tingkat kesuksesan. Data tingkat kesuksesan responden untuk masing-masing jenis pengguna, yaitu staf dan administrator, sebelumnya dikumpulkan dengan teknik pengukuran kinerja menggunakan persamaan completion rate. Penghitungan skor dapat dihitung dengan rumus berikut.

Skor bernomor Ganjil = Likert – 1

Skor bernomor Genap = 5 – Likert

Total Skor * 2.5 (skala SUS adalah dari 0 hingga 100, sedangkan total skor maksimal adalah 40, jadi perlu dikalikan dengan 2.5 untuk mendapatkan hasil akhir dalam skala 100).

$$\text{Presentase Keberhasilan} = \frac{\text{jumlah task yang berhasil diselesaikan}}{\text{jumlah task keseluruhan}} \times 100\%$$

perhitungan skor SUS untuk tiga pengguna ditunjukkan sebagai berikut:

User 1: Skor SUS = 72.5

User 2: Skor SUS = 62.5

User 3: Skor SUS = 65.0

Presentase keberhasilan = $((72.5 + 62.5 + 65.0) / 3) * 100\% = 66.67\%$

hasil rata-rata presentase keberhasilan responden dalam menyelesaikan task scenario tersebut diinterpretasikan berdasarkan Standar Acuan Litbang Depdagri tahun 1991 (Ayu, 2017) seperti ditunjukan pada Tabel 4.3.

Tabel 4. 3 hasil rata-rata presentase.

No	Rasio Efektivitas	Tingkat Pencapaian
1	< 40%	Sangat Tidak Efektif
2	40% - 59,99%	Tidak Efektif
3	60% - 79,99%	Cukup Efektif
4	> 80%	Sangat Efektif

Nilai rata-rata SUS yang didapatkan sebesar 66.67%, aplikasi e-arsip berada pada Rasio Efektivitas yaitu, Cukup Efektif sesuai dengan standar yang digunakan.

BAB V

KESIMPULAN DAN SARAN

5.1.Kesimpulan

Setelah menyelesaikan tahapan pembangunan aplikasi pengelolaan arsip berbasis web (E-arsip) pada bagian perencanaan dan pengelolaan aset di perumda air minum tirta khatulistiwa kota Pontianak menggunakan metode *prototype* dengan template *bootstra 5* dan *framework CodeIgniter 4*, kesimpulan yang dihasilkan adalah sebagai berikut:

1. Dalam proses perancangan dan pengembangan aplikasi e-arsip, beberapa tahap telah dilakukan, seperti mengumpulkan kebutuhan, merancang dan membangun *prototype*, uji coba dan evaluasi, mengkodekan sistem, dan mengujinya. Setiap tahap telah diselesaikan dengan baik.
2. Aplikasi e-arsip yang telah dibangun memiliki beberapa fitur utama, seperti membuat, membaca, mengubah, menghapus (CRUD) dokumen, mencari, menyortir, mengunduh, dan mencetak dokumen. Selain itu, aplikasi e-arsip juga memiliki fitur manajemen pengguna, seperti membuat, mengedit, dan menghapus *User* sudah berfungsi dengan baik, memungkinkan pengguna untuk lebih mudah menyimpan dan mencari yang diperlukan dalam arsip.
3. Dari pengujian menunjukan fitur aplikasi sudah cukup berhasil dalam mengelola dokumen yang diarsipkan baik secara tampilan dan fungsionalitas.

5.2.Saran

Dari hasil implementasi metode *prototype* pada pembuatan aplikasi e-arsip pada bagian perencanaan dan pengelolaan aset di perumda air minum tirta khatulistiwa kota Pontianak berbasis web dengan *framework CodeIgniter 4* dan tamplate *bootstrap 5* serta bahasa pemrograman lain dalam laporan ini, saran yang dapat dilakukan dalam pengembangan aplikasi adalah sebagai berikut:

1. Menambahkan fitur backup dan restore, untuk memastikan keamanan data dokumen, dapat ditambahkan fitur backup dan restore data. Ini dapat mempermudah proses pemulihan data jika terjadi kehilangan data.
2. Fitur jenis dokumen sebaiknya dibuat dinamis, memungkinkan penambahan, pengeditan, dan penghapusan jenis dokumen langsung melalui aplikasi, agar lebih fleksibel dan sesuai dengan kebutuhan pengguna.

DAFTAR PUSTAKA

- [1] D. Ade Handayani Capah, “Aplikasi Pengelolaan Dokumen dan Arsip berbasis Web untuk mengatur Sistem kearsipan dengan menggunakan Metode Waterfall,” vol. 11.
- [2] F. Esti Nilawati, M. Rizal, E. Hari Rachmawanto, D. Rosal Ignatius Moses Setiadi, and C. Atika Sari, “Implementasi E-arsip Untuk Penyimpanan Dokumen Digital Pada PT BPD Jateng (Bank Jateng) E-arsip Implementation for Digital Document Storage in PT BPD of Central Java (Central Bank),” 2019.
- [3] J. Saptia Kurnia and F. Risya, “RANCANG BANGUN PENERAPAN MODEL PROTOTYPE DALAM PERANCANGAN SISTEM INFORMASI PENCATATAN PERSEDIAAN BARANG BERBASIS WEB.”
- [4] M. Sistem Informasi E-Arsip Laporan Tugas Akhir Berbasis Website Pada Perguruan Tinggi, N. Kadek Pebriyanti, A. A. Istri Ita Paramitha, and E. Grana Aristyana Dewi, “Model Sistem Informasi E-Arsip Laporan Tugas Akhir Berbasis Website Pada Perguruan Tinggi”.
- [5] A. Wardhanie and K. Lebdaningrum, “Pengenalan Aplikasi Desain Grafis Figma pada Siswa-Siswi Multimedia SMK PGRI 2 Sidoarjo,” *Yumary: Jurnal Pengabdian kepada Masyarakat*, vol. 3, no. 3, pp. 165–174, Mar. 2023, doi: 10.35912/yumary.v3i3.1536.
- [6] Rohi Abdulloh, *7 in 1 Pemrograman Web untuk Pemula*. Jakarta: PT Elex Media Komputindo., 2018.
- [7] Didik Setiawan, *Buku Sakti Pemrograman Web: HTML, CSS, PHP, MYSQL & Javascript*. Start Up, 2018.
- [8] S. K. M. K. Dendy Kurniawan, *BELAJAR PEMROGRAMAN WEB DASAR*. Semarang, 2023.
- [9] A. Oktarini, S. ; Ari, and A. ; Sunarti, *WEB PROGRAMMING*. 2019.
- [10] A. Annisya, D. A. Nastiti, M. Fikri, H. Bahri, and B. Pramono, “Rancang Bangun Sistem Pengarsipan Dokumen Berbasis Web Menggunakan Metode Prototype (Studi Kasus : ATR/BPN 1 Kantor Wilayah Provinsi Sulawesi Tenggara),” *ANIMATOR*, vol. 1, no. 3, pp. 1–5.
- [11] M. Fakultas, S. Dan, T. Prodi, and T. Informasi, “PENGEMBANGAN SISTEM INFORMASI ARSIP SURAT DENGAN METODE PROTOTYPE BERBASIS WEBSITE STUDI KASUS: PRODI TEKNOLOGI INFORMASI TUGAS AKHIR Diajukan Oleh: M. YOGA AGUSTIRANDA.”
- [12] F. A. M. Devie Rosa Anamisa, *Dasar Pemrograman Web*. Malang, 2020.
- [13] M. S. Tuloli, R. Patalangi, and R. Takdir, “Pengukuran Tingkat Usability Sistem Aplikasi e-Rapor Menggunakan Metode Usability Testing dan SUS,” *Jambura Journal of Informatics*, vol. 4, no. 1, pp. 13–26, Apr. 2022, doi: 10.37905/jji.v4i1.13411.

LAMPIRAN

Wawancara

27/02/2024

No
Date

Dokumen apa yg disimpan di aplikasi?

Jwb:

1. Dokumen Teknik
2. Dokumen Administrasi
3. Dokumen Pelayanan

Foto ngus

Siswa yang menggunakan aplikasi?

Jwb:

Staff bagian perencanaan dan pengelolaan aset

Staff / Kabark sebagai admin dapat menambah akun.


Lampiran 1 Hasil Wawancara

INFORMASI
SURAT
Masuk
Keluar
E - PLANING
Perencanaan swadaya
Perencanaan Rehab Jaringan
Perencanaan Pengembangan Jaringan
Perencanaan Sipil & ME
E - MONITORING
< 100 jt
100 jt - 300 jt
> 300 jt
E - PENCATATAN ASET
- Pemetaan Swadaya
- Pemetaan Rehab Jaringan
- Pemetaan Pengembangan Jaringan
- Sipil & ME
E - ARSIP
- Teknik
1. Rencana Anggaran Biaya
2. Laporan Kegiatan Proyek
3. Ass Build Drawing
4. Berita Acara Pemeriksaan Fisik
5. Berita Acara Serah Terima Pekerjaan
6. Dokumen Perencanaan Konsultan
7. DLL
- Administrasi dan Keuangan
1. Audit
2. DLL
- Pelayanan
1. Permohonan Swadaya
2. Permohonan Bantuan
3. DLL
<u>REPORT</u>
Dibuat dengan Mi Catatan

Lampiran 2 Daftar Dokumen Yang Akan Diarsipkan

NO	Pertanyaan	Skala Likert		
		User 1	User 2	User 3
1	Saya merasa aplikasi e-arsip mudah digunakan	4	5	4
2	Saya merasa aplikasi e-arsip terlalu rumit	3	3	3
3	Saya merasa mudah menggunakan aplikasi e-arsip untuk mengelola dokumen	4	4	4
4	Saya memerlukan bantuan orang lain untuk bisa menggunakan aplikasi e-arsip dengan baik	2	3	3
5	Fitur-fitur aplikasi e-arsip terintegrasi dengan baik	5	4	4
6	Fitur-fitur aplikasi e-arsip tidak selaras	3	3	3
7	Saya merasa aplikasi e-arsip mudah dipelajari dan digunakan orang lain	5	4	5
8	Saya merasa aplikasi e-arsip membingungkan dan harus mempelajari banyak hal sebelum menggunakannya dengan efektif	3	4	3
9	Saya merasa mudah untuk mengakses dokumen dengan menggunakan aplikasi e-arsip	5	4	5
10	Saya merasa aplikasi e-arsip terlalu kaku dan monoton	3	3	4

Lampiran 3 Pengujian System Usability Scale



**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN
RISET, DAN TEKNOLOGI**
POLITEKNIK NEGERI PONTIANAK
Jalan Jenderal Achmad Yani – Pontianak 78124, Kalimantan Barat
Telp. : (0561) 736180, Faksimile : (0561) 740143, Kotak Pos 1286
Laman: www.polnep.ac.id

Nomor : 2872 /PL.16/KM/2024
Hal : Pengambilan Data untuk Tugas Akhir (TA) Mahasiswa

Yth. Direktur Perumda Air Minum Tirta Khatulistiwa

Sehubungan dengan adanya kegiatan Tugas Akhir (TA) mahasiswa Program Studi Teknik Informatika, Jurusan Teknik Elektro, Politeknik Negeri Pontianak, maka dengan ini kami mohon agar Bapak/Ibu berkenan untuk di wawancara dan memberikan informasi mengenai data-data terkait dengan Judul Tugas Akhir (TA).

Adapun nama mahasiswa yang mengambil data sebagai berikut:

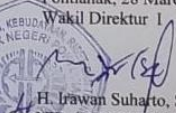
No.	Nama /NIM	Program Studi	Judul TA
1	Jidan NIM. 3202116088	Teknik Informatika	Rancang Bangun Aplikasi Pengelolaan Arsip (E-Arsip) Pada Bagian Perencanaan Dan Pengelolaan Aset Di Perumda Air Minum Tirta Khatulistiwa

Adapun data yang diperlukan adalah sebagai berikut :

1. Data Arsip Dokumen

Untuk konfirmasi kesediaan dapat menghubungi Bapak **Safri Adam, S.Kom., M.Kom.** selaku Koordinator Tugas Akhir (TA) dengan nomor **HP/WA. 081258811801**.

Demikianlah permohonan ini disampaikan. Atas perhatian dan bantuan Bapak/Ibu, kami ucapkan terima kasih.

Pontianak, 28 Maret 2024
Wakil Direktur I

H. Irawan Suharto, S.T., M.T.
NIP. 197103111998021001

Tembusan:

1. Koordinator TA
2. Tata Usaha Polnep
3. Jurusan Teknik Elektro

Lampiran 4 Surat Ijin penelitian