

**IMPLEMENTASI ALGORITMA SEQUENTIAL SEARCHING
PADA PEMBUATAN APLIKASI E-ARSIP BERBASIS WEB
DI UPK ARTHA RAHARJA KECAMATAN PACET**

SKRIPSI

Karya Tulis sebagai Syarat untuk Memperoleh
Gelar Sarjana Komputer dari Fakultas Teknologi Informasi
Universitas Bale Bandung

Disusun Oleh:

SABDA ALAM
NPM. C1A160015



PROGRAM STRATA 1
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG
BANDUNG
2020

LEMBAR PERSETUJUAN PEMBIMBING

**IMPLEMENTASI ALGORITMA SEQUENTIAL SEARCHING
PADA PEMBUATAN APLIKASI E-ARSIP BERBASIS WEB
DI UPK ARTHA RAHARJA KECAMATAN PACET**

Disusun Oleh:

SABDA ALAM
C1A160015

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

SARJANA KOMPUTER


Pada

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Baleendah, Juli 2020

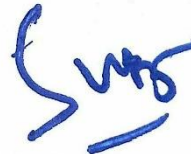
Disetujui Oleh:

Pembimbing 1



Zen Munawar, S.T., M.Kom.
NIDN. 0422037002

Pembimbing 2



Sutiyono, S.T., M.Kom.
NIDN. 0427038204

LEMBAR PENGESAHAN PENGUJI

**IMPLEMENTASI ALGORITMA SEQUENTIAL SEARCHING
PADA PEMBUATAN APLIKASI E-ARSIP BERBASIS WEB
DI UPK ARTHA RAHARJA KECAMATAN PACET**

Disusun Oleh:

SABDA ALAM
C1A160015

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

SARJANA KOMPUTER

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Baleendah, Juli 2020

Disetujui Oleh:

Penguji 1



Denny Rusdianto, S.T., M.Kom.
NIK. 04104808094

Penguji 2



Nurul Imamah, S.T., M.T.
NIK. 04104808121

LEMBAR PENGESAHAN PROGRAM STUDI

**IMPLEMENTASI ALGORITMA SEQUENTIAL SEARCHING
PADA PEMBUATAN APLIKASI E-ARSIP BERBASIS WEB
DI UPK ARTHA RAHARJA KECAMATAN PACET**

Disusun Oleh:

SABDA ALAM
C1A160015

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

SARJANA KOMPUTER

Pada

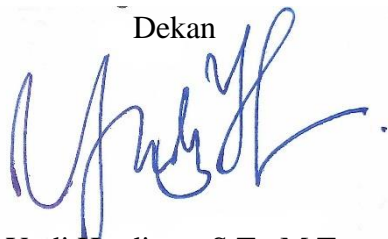
**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Baleendah, Juli 2020

Disetujui Oleh:

Mengetahui,

Dekan



Yudi Herdiana, S.T., M.T.

NIK. 04104808008

Mengesahkan,

Ketua Program Studi



Yaya Suharya, S.T., M.Kom.

NIK. 01043170007

LEMBAR PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : SABDA ALAM

NPM : C1A160015

Judul Skripsi : IMPLEMENTASI ALGORITMA SEQUENTIAL SEARCHING
PADA PEMBUATAN APLIKASI E-ARSIP BERBASIS WEB DI
UPK ARTHA RAHARJA KECAMATAN PACET

Menyatakan bahwa penulisan skripsi ini berdasarkan hasil penelitian, pemikiran dan pemaparan asli dari saya sendiri. Jika terdapat karya orang lain, saya mencantumkan sumber yang jelas. Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik yang berlaku di Fakultas Teknologi Informasi Universitas Bale Bandung. Demikian surat pernyataan ini saya buat dalam keadaan sadar tanpa paksaan dari pihak manapun.

Baleendah, Juli 2020

Yang membuat pernyataan



METERAI
TEMPEL
6000
ENAM RIBU RUPIAH

SABDA ALAM
NPM. C1A160015

ABSTRACT

The process of archiving is a routine operational activity carried out by each organization/agency or office including in the government environment or institutions such as UPK Artha Raharja, Pacet District. Processing and archiving, especially incoming and outgoing mail archives at UPK Artha Raharja, Pacet District so far has been going well, although it is still done manually by writing (handwritten) on the letter book.

This method is certainly not efficient because it requires a very complete book and even greater archive storage if it is installed in a long period of time and officers will have difficulty in finding certain documents that have been stored because the search process is done by checking one by one. in the file storage cabinet. The Electronic Archive Application is a system to facilitate the process of archiving special correspondence at UPK Artha Raharja, Pacet District. With this application, it can help the process of archival storage at UPK Artha Raharja and does not require paper or books that are full of large file cabinet storage because the records are stored in a database system. This application is created using PHP and MySQL with the addition of a sequential Search algorithm implementation in the search system to make it easier for officers to search for certain letter documents that have been stored in the system if the documents needed are needed or reused.

The conclusion of this research is to help develop an administrative system for archiving correspondence in UPK Artha Raharja, Pacet District, from a conventional system to a web-based computerized system using PHP and MySQL. With the Electronic Archive (E-Archive) application in Artha Raharja's Activity Management Unit in Pacet District, it will certainly facilitate the process of archiving documents, especially incoming and outgoing letters, so that they can be more easily controlled and more efficient.

Keywords: Sequential searching, electronic archive, search.

ABSTRAK

Proses pengarsipan merupakan kegiatan operasional rutin yang dilakukan oleh setiap organisasi/instansi atau perkantoran termasuk di lingkungan pemerintahan atau lembaga seperti halnya di UPK Artha Raharja Kecamatan Pacet. Pengolahan dan pengarsipan khususnya arsip surat masuk maupun surat keluar di UPK Artha Raharja Kecamatan Pacet sejauh ini berjalan dengan baik walaupun masih dilakukan secara manual dengan mencatat (tuliskan tangan) pada buku surat.

Dengan cara tersebut tentu tidak efisien karena memerlukan buku yang sangat tebal serta lemari penyimpanan arsip yang besar terlebih lagi jika untuk mencatat dalam jangka waktu yang panjang dan petugas akan kesulitan dalam mencari dokumen surat tertentu yang sudah tersimpan karena proses pencarian dilakukan dengan cara di cek satu-persatu pada lemari penyimpanan arsip. Aplikasi Arsip Elektronik merupakan sebuah sistem untuk memudahkan proses pengarsipan khususnya surat-menysurat di UPK Artha Raharja Kecamatan Pacet. Dengan adanya aplikasi ini tentunya dapat membantu proses pengelolaan arsip di UPK Artha Raharja dan tidak memerlukan kertas atau buku yang tebal maupun lemari penyimpanan yang besar karena arsip disimpan di dalam sistem *database*. Aplikasi ini dibuat menggunakan PHP dan MySQL dengan tambahan implementasi algoritma *Sequential searching* pada sistem pencariannya untuk memudahkan petugas mencari dokumen surat tertentu yang sudah di simpan di dalam sistem jika sewaktu-waktu dokumen yang dicari diperlukan atau digunakan kembali.

Kesimpulan dari penelitian kali ini adalah membantu mengembangkan sistem administrasi pengarsipan surat menyurat di UPK Artha Raharja Kecamatan Pacet dari sistem konvensional menjadi sistem komputerisasi berbasis *web* menggunakan PHP dan MySQL. Dengan adanya aplikasi Arsip Elektronik (E-Arsip) di Unit Pengelola Kegiatan Artha Raharja Kecamatan Pacet tentunya akan lebih memudahkan proses pengarsipan dokumen khususnya surat masuk dan surat keluar agar dapat lebih mudah dikendalikan dan lebih efisien.

Kata Kunci: *Sequential searching*, arsip elektronik, pencarian.

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas berkat dan limpahan rahmat serta karunia-Nya kepada kita semua sehingga penyusun dapat menyelesaikan laporan tugas akhir skripsi ini dengan judul “Implementasi Algoritma Sequential Searching Pada Pembuatan Aplikasi E-Arsip Berbasis Web Di UPK Artha Raharja Kecamatan Pacet” tepat pada waktunya. Laporan tugas akhir skripsi ini disusun sebagai salah satu syarat untuk mendapatkan gelar Sarjana Komputer di Fakultas Teknologi Informasi Universitas Bale Bandung.

Laporan tugas akhir skripsi ini dibuat dengan beberapa bantuan dari berbagai pihak untuk menyelesaikan tantangan dan hambatan selama pengerjaannya. Oleh karena itu, pada kesempatan kali ini penyusun ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Yudi Herdiana, S.T., M.T., selaku Dekan Fakultas Teknologi Informasi Universitas Bale Bandung.
2. Bapak Yaya Suharya, S.T., M.T., selaku Ketua Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Bale Bandung.
3. Bapak Zen Munawar, S.T., M.Kom., selaku Dosen pembimbing satu di Fakultas Teknologi Informasi Universitas Bale Bandung.
4. Bapak Sutiyono, S.T., M.Kom, selaku Dosen pembimbing dua di Fakultas Teknologi Informasi Universitas Bale Bandung.
5. Denny Rusdianto, S.T., M.Kom., selaku Dosen penguji satu di Fakultas Teknologi Informasi Universitas Bale Bandung.
6. Nurul Imamah, S.T., M.T., selaku Dosen penguji dua di Fakultas Teknologi Informasi Universitas Bale Bandung.
7. Mochamad Ridwan, S.T., selaku Dosen pembantu pembimbing di Fakultas Teknologi Informasi Universitas Bale Bandung.
8. Dosen beserta staf Fakultas Teknologi Informasi Universitas Bale Bandung.
9. Keluarga di rumah yang memberikan dukungan berupa moril, materil dan do’a selama proses penelitian dan pelaporan.

10. Rekan-rekan seperjuangan yang saling membantu dan memberikan semangat dalam proses penelitian maupun penulisan laporan.

Penyusun menyadari bahwa masih banyak kekurangan dalam laporan ini, segala kritik dan saran yang membangun akan penyusun terima dengan baik. Akhir kata semoga laporan skripsi ini bisa diterima dan bermanfaat bagi kita semua.

Bandung, Juli 2020

Sabda Alam
C1A160015

DAFTAR ISI

LEMBAR PERSETUJUAN PEMBIMBING	ii
LEMBAR PENGESAHAN PENGUJI	iii
LEMBAR PENGESAHAN PROGRAM STUDI.....	iv
LEMBAR PERNYATAAN	v
ABSTRACT.....	vi
ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR	xiv
DAFTAR LAMPIRAN	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan	3
1.5 Metodologi Penelitian	3
1.6 Metode Pengembangan Sistem	4
1.7 Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA.....	7
2.1 Landasan Teori.....	7
2.2 Dasar Teori.....	10
2.2.1 Algoritma	10
2.2.2 Algoritma Pencarian.....	10
2.2.3 <i>Sequential Searching</i>	11
2.2.4 Arsip.....	12
2.2.5 Waterfall.....	14
2.2.6 XAMPP.....	16
2.2.7 <i>HyperText Markup Language</i> (HTML)	17
2.2.8 <i>Cascading Style Sheet</i> (CSS)	17
2.2.9 PHP	18

2.2.10 <i>Bootstrap</i>	18
2.2.11 <i>Database</i>	19
2.2.12 <i>Database Management System (DBMS)</i>	19
2.2.13 <i>SQL</i>	20
2.2.14 <i>MySQL</i>	20
2.2.15 <i>Sublime Text</i>	20
2.2.16 <i>Web Browser</i>	22
2.2.17 <i>Web Hosting</i>	23
2.2.18 <i>Unified Modelling Language (UML)</i>	25
BAB III METODOLOGI PENELITIAN	30
3.1 <i>Kerangka Pikir</i>	30
3.2 <i>Deskripsi</i>	31
3.2.1 <i>Inisialisasi</i>	31
3.2.2 <i>Identifikasi Masalah</i>	31
3.2.3 <i>Pengumpulan Data</i>	32
3.2.4 <i>Analisis Kebutuhan</i>	33
3.2.5 <i>Desain</i>	34
3.2.6 <i>Implementasi Desain</i>	35
3.2.7 <i>Pengujian</i>	35
3.2.8 <i>Pembuatan Laporan</i>	36
BAB IV ANALISIS DAN PERANCANGAN	37
4.1 <i>Analisis</i>	37
4.1.1 <i>Instrumen Penelitian</i>	37
4.1.2 <i>Analisis Sistem</i>	39
4.1.3 <i>Analisis Kebutuhan</i>	42
4.1.4 <i>Hasil Analisis</i>	45
4.2 <i>Perancangan</i>	46
4.2.1 <i>Perancangan Sistem</i>	46
4.2.2 <i>Perancangan Basis Data</i>	62
4.2.3 <i>Perancangan Algoritma Sequential Searching</i>	63
4.2.4 <i>Perancangan User Interface</i>	65
BAB V IMPLEMENTASI DAN PENGUJIAN	74
5.1 <i>Implementasi</i>	74
5.1.1 <i>Implementasi User Interface</i>	74

5.1.2 Implementasi Algoritma <i>Sequential Searching</i>	85
5.2 Pengujian.....	86
5.2.1 Pengujian Sistem.....	86
5.2.2 Pengujian Algoritma <i>Sequential Searching</i>	93
BAB VI KESIMPULAN DAN SARAN	96
6.1 Kesimpulan	96
6.2 Saran.....	97
DAFTAR PUSTAKA	98
LAMPIRAN.....	100

DAFTAR TABEL

Tabel 2.1 Simbol <i>Use Case Diagram</i>	26
Tabel 2.2 Simbol <i>Activity Diagram</i>	27
Tabel 2.3 Simbol <i>Class Diagram</i>	29
Tabel 4.1 Spesifikasi Perangkat Keras	38
Tabel 4.2 Spesifikasi <i>Server Hosting</i>	39
Tabel 4.3 Analisis Kebutuhan Perangkat Keras.....	42
Tabel 4.4 Analisis Kebutuhan Sistem	44
Tabel 4.5 Deskripsi Aktor	47
Tabel 4.6 Deskripsi <i>Use Case</i>	47
Tabel 4.7 Skenario <i>Use Case Login</i>	49
Tabel 4.8 Skenario Menambah Data Surat Masuk.....	50
Tabel 4.9 Skenario Mengubah Data Surat Masuk	51
Tabel 4.10 Skenario Menghapus Data Surat Masuk	52
Tabel 4.11 Skenario Melihat Data Surat Masuk	53
Tabel 4.12 Skenario Mencari Data Surat Masuk	53
Tabel 4.13 Skenario Menambah <i>User</i>	54
Tabel 4.14 Skenario Mengubah <i>User</i>	55
Tabel 4.15 Skenario Menghapus <i>User</i>	56
Tabel 4.16 Tabel Surat Masuk	62
Tabel 4.17 Tabel Surat Keluar	62
Tabel 4.18 Tabel Akun.....	63
Tabel 5.1 Pengujian Proses <i>Login</i>	87
Tabel 5.2 Pengujian Manajemen Surat Masuk	88
Tabel 5.3 Pengujian Manajemen Surat Keluar	89
Tabel 5.4 Pengujian Manajemen Akun.....	91
Tabel 5.5 Pengujian Fitur Aplikasi Arsip	92
Tabel 5.6 Pengujian Algoritma <i>Sequential Searching</i>	93

DAFTAR GAMBAR

Gambar 1.1 Metode Pengembangan Sistem	4
Gambar 2.1 Contoh 1 <i>Sequential Searching</i>	11
Gambar 2.2 Contoh 2 <i>Sequential Searching</i>	11
Gambar 2.2 Contoh 3 <i>Sequential Searching</i>	12
Gambar 2.4 Model <i>waterfall</i>	15
Gambar 3.1 Kerangka Pikir.....	30
Gambar 4.1 <i>Flowmap</i> Sistem Pengarsipan Konvensional	40
Gambar 4.2 <i>Flowmap</i> Aplikasi E-Arsip.....	41
Gambar 4.3 <i>Use Case Diagram</i> Aplikasi Arsip Elektronik.....	48
Gambar 4.4 <i>Activity Diagram</i> Aplikasi Arsip Elektronik.....	57
Gambar 4.5 <i>Activity Diagram Login</i>	57
Gambar 4.6 <i>Activity Diagram</i> Tambah Surat Masuk.....	58
Gambar 4.7 <i>Activity Diagram</i> Edit Surat Masuk	58
Gambar 4.8 <i>Activity Diagram</i> Hapus Surat Masuk.....	59
Gambar 4.9 <i>Activity Diagram</i> Cari Surat Masuk	59
Gambar 4.10 <i>Activity Diagram</i> Cetak Laporan Surat Masuk	60
Gambar 4.11 <i>Activity Diagram</i> Tambah User	60
Gambar 4.12 <i>Activity Diagram</i> Edit User.....	61
Gambar 4.13 <i>Activity Diagram</i> Hapus User	61
Gambar 4.15 <i>Flowchart</i> Algoritma <i>Sequential Searching</i>	64
Gambar 4.16 Rancangan Halaman <i>Login</i>	65
Gambar 4.17 Rancangan Halaman Utama	66
Gambar 4.18 Rancangan Halaman Data Surat Masuk.....	66
Gambar 4.19 Rancangan Halaman Data Surat Keluar.....	67
Gambar 4.20 Rancangan Halaman Tambah Surat Masuk	67
Gambar 4.21 Rancangan Halaman Tambah Surat Keluar	68
Gambar 4.22 Rancangan Halaman Edit Surat Masuk.....	68
Gambar 4.23 Rancangan Halaman Edit Surat Keluar.....	69
Gambar 4.24 Rancangan Halaman Hapus Surat Masuk	69
Gambar 4.25 Rancangan Halaman Hapus Surat Keluar	70

Gambar 4.26 Rancangan Halaman Lihat Surat Masuk	70
Gambar 4.27 Rancangan Halaman Lihat Surat Keluar	71
Gambar 4.28 Rancangan Halaman Cetak Laporan	71
Gambar 4.29 Rancangan Halaman Manajemen Akun	72
Gambar 4.30 Rancangan Halaman Tambah Akun.....	72
Gambar 4.31 Rancangan Halaman Edit Akun	73
Gambar 5.1 Halaman <i>Login</i> Aplikasi Arsip.....	74
Gambar 5.2 Halaman Awal Aplikasi Arsip	75
Gambar 5.3 Halaman Surat Masuk Aplikasi Arsip.....	75
Gambar 5.4 Halaman Tambah Surat Masuk Aplikasi Arsip.....	76
Gambar 5.5 Halaman Edit Surat Masuk Aplikasi Arsip	76
Gambar 5.6 Halaman Lihat File Surat Masuk Aplikasi Arsip	77
Gambar 5.7 Halaman Hapus Surat Masuk Aplikasi Arsip	77
Gambar 5.8 Halaman Surat Keluar Aplikasi Arsip.....	78
Gambar 5.9 Halaman Tambah Surat Keluar Aplikasi Arsip.....	78
Gambar 5.10 Halaman Edit Surat Keluar Aplikasi Arsip	79
Gambar 5.11 Halaman Lihat File Surat Keluar Aplikasi Arsip	79
Gambar 5.12 Halaman Hapus Surat Keluar Aplikasi Arsip	80
Gambar 5.13 Halaman Manajemen Akun Aplikasi Arsip	80
Gambar 5.14 Halaman Tambah Akun Aplikasi Arsip	81
Gambar 5.15 Halaman Edit Akun Aplikasi Arsip	81
Gambar 5.16 Halaman Hapus Akun Aplikasi Arsip.....	82
Gambar 5.17 Halaman Cetak Laporan Aplikasi Arsip	82
Gambar 5.18 Halaman Hasil Cetak Laporan Aplikasi Arsip	83
Gambar 5.19 Halaman Kesalahan <i>Login Username</i>	83
Gambar 5.20 Halaman Kesalahan <i>Login Password</i>	84
Gambar 5.21 Halaman Kesalahan <i>Input Data Surat</i>	84
Gambar 5.22 Implementasi Algoritma <i>Sequential Searching</i>	85

DAFTAR LAMPIRAN

Lampiran 1 Kuesioner.....	98
Lampiran 2 <i>Source Code</i> Aplikasi	101
Lampiran 3 Riwayat Hidup Penulis	126

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi komputer dan telekomunikasi telah mengantar dan membawa manusia ke dalam dunia informasi yang mengalir tanpa batas. Manusia membutuhkan informasi guna menyelesaikan berbagai kebutuhan hidupnya, diantaranya dalam kehidupan di organisasi. (Sedarmayanti, 2018). Untuk mendapatkan informasi atau bertukar informasi di suatu organisasi, lembaga, perusahaan atau instansi lainnya contohnya di Unit Pengelola Kegiatan (UPK) Artha Raharja Kecamatan Pacet menggunakan surat-menyurat. Pengelompokan surat dibagi dalam dua jenis, yaitu surat masuk dan surat keluar. Surat masuk adalah surat yang masuk ke dalam instansi/perusahaan dari pihak eksternal maupun pihak internal, sedangkan surat keluar adalah surat yang dikirim oleh instansi/perusahaan itu sendiri ke instansi/perusahaan lain. Dalam kegiatan surat-menyurat tentunya membutuhkan administrasi kearsipan untuk mengelola data surat masuk maupun surat keluar agar mudah dikendalikan.

Administrasi kearsipan adalah segenap rangkaian perbuatan menyelenggarakan kearsipan sejak saatnya pengumpulan, pengelolaan warkat, sampai dengan penghapusannya. (Sedarmayanti, 2018). Pengelolaan arsip surat masuk dan surat keluar di UPK Artha Raharja Kecamatan Pacet melibatkan banyak jurnal atau kertas yang harus diisi untuk mencatat dan mengelompokkan surat masuk dan surat keluar. Contohnya ketika ada surat masuk, maka petugas akan mengisi buku daftar pengendali surat masuk sesuai dengan data surat masuk yang ada. Setelah itu surat akan diserahkan ke ketua untuk memberi instruksi selanjutnya. Kemudian surat tersebut akan kembali lagi ke petugas untuk melanjutkan proses sesuai instruksi yang diberikan. Langkah terakhir yang dilakukan petugas dengan mengisi buku kendali surat masuk dan menyimpan berkas surat di lemari penyimpanan arsip.

Administrasi kearsipan di UPK Artha Raharja Kecamatan Pacet khususnya di arsip surat masuk dan surat keluar masih menggunakan cara konvensional sehingga membutuhkan buku yang tebal terlebih lagi jika pencatatan dalam waktu yang panjang. Semakin banyak berkas maka tempat penyimpanan yang dibutuhkan semakin besar. Penyimpanan informasi surat yang dilakukan dengan cara menulis di buku surat serta penyimpanan berkas surat di lemari penyimpanan juga berakibat pada proses pencarian dokumen surat. Contohnya jika sewaktu-waktu dibutuhkan kembali surat tertentu baik surat masuk maupun surat keluar maka petugas harus mencari satu persatu pada buku surat untuk mencari data surat tersebut dan mencari berkas suratnya di lemari penyimpanan untuk mengetahui isi surat tersebut.

Berdasarkan masalah di atas, maka penyusun bermaksud untuk mengembangkan sistem untuk membantu mempermudah proses pengarsipan di UPK Artha Raharja Kecamatan Pacet dari sistem konvensional menjadi sistem komputerisasi berbasis *web* yaitu aplikasi Arsip Elektronik. Aplikasi tersebut dapat membantu petugas memasukkan data arsip surat masuk maupun surat keluar ke dalam sistem sehingga tidak perlu menggunakan buku yang tebal serta tidak perlu penyimpanan lemari yang lebih besar karena informasi surat dapat disimpan dalam *database*. Dengan adanya aplikasi tersebut Petugas juga dapat terbantu dalam proses pencarian dokumen surat dengan adanya fitur pencarian berdasarkan kriteria surat tertentu. Dan juga dalam segi keamanan akan terbantu, resiko terhadap berkas yang hilang maupun rusak dapat diperkecil karena data surat masuk maupun surat keluar dapat dilihat kembali di dalam sistem dan tentunya cara tersebut lebih efisien dalam proses pengarsipan.

Untuk mempermudah Pembuatan aplikasi Arsip Elektronik ini penyusun menggunakan *framework* Bootstrap dan bahasa pemrograman PHP, sedangkan untuk manajemen *database*-nya menggunakan MySQL. Dan untuk fitur pencarian dalam aplikasi tersebut penyusun membuat uji coba menggunakan penerapan algoritma *Sequential Searching* atau disebut juga pencarian beruntun yang prosesnya membandingkan setiap data atau elemen satu persatu secara beruntun, mulai dari elemen pertama sampai dengan elemen yang dicari ditemukan atau seluruh elemen sudah dibandingkan.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas maka didapatkan rumusan masalah diantaranya sebagai berikut:

1. Bagaimana mengatasi proses pengarsipan surat masuk dan surat keluar di UPK Artha Raharja agar lebih efisien?
2. Bagaimana cara mempermudah proses pencarian dokumen didalam arsip?
3. Bagaimana cara menerapkan algoritma *Sequential Searching* di dalam aplikasi Arsip Elektronik?

1.3 Batasan Masalah

Dalam penelitian dan pembuatan aplikasi ini terdapat beberapa batasan masalah diantaranya sebagai berikut:

1. Pembuatan aplikasi tersebut hanya melingkupi arsip surat masuk dan surat keluar di UPK Artha Raharja Kecamatan Pacet.
2. Aplikasi tersebut hanya dapat digunakan oleh pengelola arsip dan bagian Sekretariat.

1.4 Tujuan

Adapun tujuan dengan diadakannya penelitian ini adalah sebagai berikut:

1. Membuat aplikasi Arsip Elektronik untuk mempermudah proses pengarsipan di UPK Artha Raharja Kecamatan Pacet.
2. Membuat fitur pencarian pada aplikasi Arsip Elektronik untuk memudahkan proses pencarian dokumen.
3. Menerapkan algoritma pencarian *Sequential Searching* pada fitur pencarian.

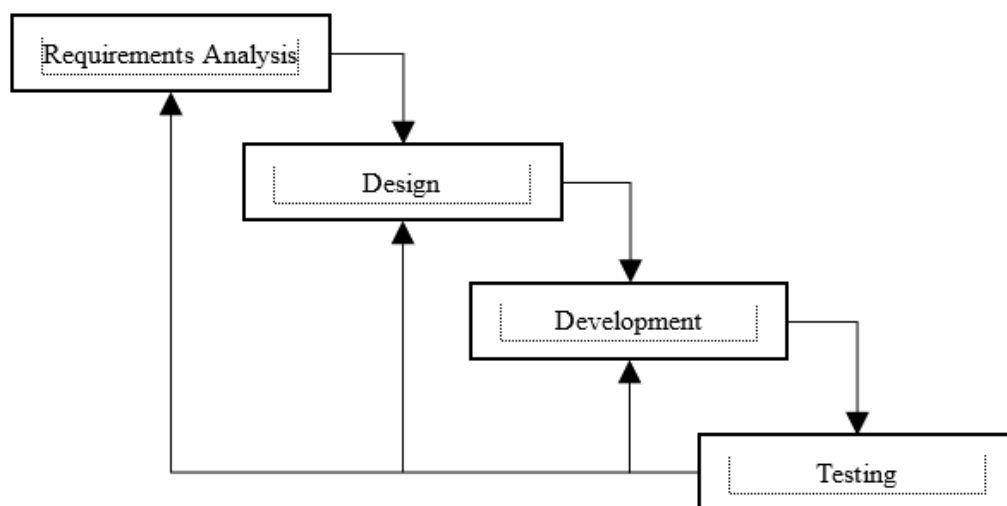
1.5 Metodologi Penelitian

Metode penelitian yang digunakan adalah metode wawancara untuk memperoleh keterangan yang dilakukan di Kantor UPK Artha Raharja Kecamatan Pacet. Tujuan dari wawancara ini yaitu untuk pengumpulan data, mulai dari data primer dan data sekunder. Data primer didapatkan langsung dari Kantor UPK Artha

Raharja Kecamatan Pacet dengan cara observasi dan wawancara dengan petugas bagian sekretariat yang berhubungan langsung dengan proses pengarsipan khususnya surat-menyurat. Sedangkan data sekunder didapatkan dengan cara studi pustaka yang bersumber dari penelitian sebelumnya dalam jurnal, buku dan artikel di internet yang berhubungan dengan topik penelitian.

1.6 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan pada pembuatan aplikasi arsip elektronik di UPK Artha Raharja Kecamatan Pacet adalah metode *waterfall*. Alasan menggunakan metode ini adalah karena metode *waterfall* melakukan pendekatan secara sistematis dan berurutan dalam membangun sebuah sistem. Berikut tahapan metode *waterfall*:



Gambar 1.1 Metode Pengembangan Sistem

Tahap pertama yaitu *requirements analysis*, pada proses ini dilakukan untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Selanjutnya adalah *design*, pada proses ini dilakukan *desain* pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi

desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Tahapan selanjutnya *development*, pada tahap ini dilakukan pembuatan kode program, desain ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain. Tahapan terakhir yaitu *testing*, pada tahap ini dilakukan pengujian fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

1.7 Sistematika Penulisan

Adapun sistematika penulisan karya tulis ilmiah ini terdapat beberapa bab diantaranya yaitu:

BAB I: PENDAHULUAN

Pada bab pertama ini membahas tentang latar belakang permasalahan, rumusan masalah yang ditemukan, batasan masalah untuk memfokuskan penelitian, tujuan penelitian, metode penelitian yang digunakan dan sistematika penulisan laporan penelitian.

BAB II: TINJAUAN PUSTAKA

Pada bab kedua ini berisi tentang tinjauan pustaka, bab ini menjelaskan tentang beberapa jurnal ilmiah yang dijadikan referensi yang digunakan pada penelitian sebagai landasan teori dan teori-teori penunjang penelitian yang berdasar dari buku maupun informasi lain sebagai dasar teori penelitian.

BAB III: METODOLOGI PENELITIAN

Pada bab ketiga ini berisi tentang kerangka pikir dan deskripsi dari kerangka pikir tersebut, bab ini menjelaskan alur kerangka pikir yang dibuat serta kebutuhan-kebutuhan selama proses penelitian.

BAB IV: ANALISIS DAN PERANCANGAN

Pada bab keempat berisi tentang analisis dan perancangan, bab ini menjelaskan tentang melakukan sebuah analisis terhadap masalah, membuat perancangan terkait pemecahan masalah dan melaporkan hasil yang didapatkan.

BAB V: IMPLEMENTASI DAN PENGUJIAN

Pada bab kelima berisi tentang implementasi dan pengujian, bab ini membahas tentang pengimplementasian dari hasil pemecahan masalah dan perancangan yang dibuat serta melakukan pengujian pada aplikasi yang dibuat.

BAB VI: KESIMPULAN DAN SARAN

Pada bab keenam ini berisi tentang kesimpulan dan saran, bab ini menjelaskan hasil yang didapatkan dari proses penelitian dan aplikasi yang dibuat serta saran untuk perbaikan atau pengembangan pada aplikasi agar lebih baik lagi.

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

Pada penelitian ini penyusun mempelajari beberapa penelitian terdahulu sebagai referensi yaitu dari jurnal yang berkaitan dengan judul penelitian, berikut beberapa jurnal yang digunakan:

2.1.1 Implementasi Algoritma *Sequential Searching* Untuk Pencarian Nomor Surat Pada Sistem Arsip Elektronik. (Anisya Sonita dan Mayang Sari, Jurnal Pseudocode Teknik Informatika Fakultas Teknik Universitas Muhammadiyah Bengkulu, 2018)

Pada jurnal ini membahas tentang pembuatan aplikasi pengolahan dan pengarsipan nomor surat di Tata Usaha Fakultas Teknik Universitas Muhammadiyah Bengkulu yang masih menggunakan cara manual yaitu dengan mencatat (tuliskan tangan) pada buku dan melihat ulang urutan surat sebelumnya untuk memberi nomor pada surat baru. Tentunya dengan cara tersebut tidak efisien karena memerlukan buku yang tebal terlebih lagi jika untuk mencatat dalam jangka waktu yang lama dan banyaknya jenis surat yang akan diarsipkan. Penelitian pada jurnal ini juga membahas tentang pengimplementasian algoritma *Sequential searching* untuk mempermudah pencarian nomor surat pada sistem arsip elektronik tersebut yang ditujukan untuk tata usaha fakultas teknik Universitas Muhammadiyah Bengkulu. Metode yang digunakan pada penelitian di jurnal ini yaitu menggunakan metode pengembangan sistem RAD atau *Rapid Application Development*. Karena *Rapid Application Development* (RAD) adalah salah satu metode pengembangan suatu sistem informasi dengan waktu yang relatif singkat. Penelitian pada jurnal ini lebih membahas ke pembuatan sistem dan penerapan algoritma *Sequential Searching* pada fitur pencariannya. Dari hasil analisa, perancangan, pengujian dan implementasi program yang telah dilakukan terhadap aplikasi Arsip Elektronik tersebut aplikasi dapat dijalankan

dengan baik. Algoritma *Sequential searching* dapat diimplementasikan pada pencarian arsip berdasarkan nomor surat pada surat masuk dan surat keluar.

2.1.2 Implementasi Algoritma *Sequential Searching* Pada Sistem Pelayanan Puskesmas Menggunakan *Bootstrap* (Studi kasus Puskesmas Kampung Bali Bengkulu). (Marissa Utami dan Yovi Apridiansyah, Jurnal Ilmiah dan Informatika Terapan Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Bengkulu, 2019)

Berdasar pada latar belakang dimana Puskesmas Kampung Bali yang merupakan suatu instansi yang bergerak di bidang pelayanan kesehatan masyarakat yang masih melakukan pelayanan terhadap pasien dengan cara manual seperti pencarian data pasien, pengelolaan data pasien, pencatatan pemeriksaan pasien masih menggunakan formulir, pengelolaan laporan data pemeriksaan dan data obat yang masih menggunakan pembukuan manual dan media penyimpanan data pasien yang masih menggunakan media kertas sehingga mengakibatkan pencarian data dilakukan dengan cara menelusuri arsip-arsip yang dapat menyita waktu dan masih dikerjakan secara manual dengan cara merekap data-data pasien melalui arsip-arsip yang dapat menyita waktu selain itu rentan dengan kesalahan. Penelitian pada jurnal ini bertujuan untuk mendukung metode pengembangan sistem pelayanan Puskesmas untuk mempermudah proses pelayanan terhadap pasien. Penelitian ini menggunakan metode waterfall pada pengembangan sistemnya, dan pengimplementasian Algoritma *Sequential Searching* pada sistem pencarian data pasien. Menurut isi dari jurnal ini algoritma *Sequential Searching* mempunyai kelebihan dalam proses pencariannya, contohnya jika data yang dicari terletak di depan maka data akan ditemukan dengan cepat. Pembuatan sistem informasi pada jurnal ini juga mengimplementasikan tampilan *Bootstrap*.

2.1.3 Implementasi Metode Sequential Dalam Pencarian Pendistribusian Barang Pada *Cargo Integration* Sistem. (Abdullah Muhazir, Muhammad Fakhriza, Eddy Sutejo, Jurnal dan Penelitian Teknik Informatika Sinkron, 2017)

Pada Jurnal ini dijelaskan tentang penawaran suatu aplikasi berbasis *web* yang memudahkan admin cabang dan pelanggan untuk dapat Meng-*cover* atau meminimalisir barang yang dikirim atau diterima dengan suatu metode *sequential searching* di CV. Duta Sarana Indocargo (DSI Cargo). Perkembangan ilmu pengetahuan dan teknologi saat ini sangat mempengaruhi kehidupan manusia. Dengan perkembangan teknologi, banyak aplikasi komputer yang dapat digunakan untuk mempermudah pekerjaan dan pembelajaran kehidupan sehari-hari. Sekarang ini telah banyak pengiriman barang dalam trayek darat, laut, dan udara. Pengiriman barang membutuhkan suatu pengiriman yang cepat dan terpercaya untuk pelanggan yang membutuhkan pengiriman barang secara cepat dan terpercaya. Saat pengiriman barang banyak pelanggan mengeluh pada admin cabang karena barang banyak penundaan saat pelanggan mengambil barang di kantor. Manfaat yang dihasilkan pada jurnal ini salah satunya yaitu untuk mempermudah mendapatkan informasi data yang ada di cabang-cabang kantor serta karyawan tidak perlu konfirmasi kepada karyawan cabang lain karena sudah langsung mendapatkan informasi lewat sistem tersebut. Dalam pembangunan aplikasi tersebut menggunakan MySQL sebagai *database* dan PHP sebagai bahasa pemrograman yang digunakan membuat program *website*. Pada sistem tersebut menggunakan metode *sequential searching* pada proses pencariannya, menurut jurnal tersebut proses *sequential searching* adalah teknik pencarian data secara urut dari depan sampai belakang atau dari awal sampai akhir berdasar *key* yang dicari. *Sequential searching* mempunyai kelebihan salah satunya jika data yang dicari terletak di depan maka data yang dicari akan ditemukan secara cepat. Tetapi jika data yang dicari terletak di belakang atau akhir maka proses pencarian membutuhkan waktu yang cukup lama. Kesimpulan dari jurnal tersebut berdasarkan hasil uji coba yang dilakukan yaitu dengan menggunakan *sequential searching* dapat mencari data lebih mudah untuk mendapatkan suatu informasi dan metode *sequential searching* dapat

diimplementasikan dalam sebuah aplikasi untuk mencari suatu data barang. Sistem yang dikembangkan memiliki fitur yang mendukung proses pelatihan pada pencarian dengan metode *sequential* serta fitur identifikasi yang mana menggunakan data-data maka hasil dari proses pelatihan untuk dapat menghasilkan suatu informasi yang tepat sesuai dengan data yang ditentukan.

2.2 Dasar Teori

Pada dasar teori ini berisikan tentang teori-teori penunjang yang berkaitan dengan judul penelitian. Berikut ini adalah beberapa teori yang digunakan sebagai penunjang pada penelitian Implementasi Algoritma *Sequential Searching* pada pembuatan Aplikasi E-Arsip di UPK Artha Raharja Kecamatan Pacet:

2.2.1 Algoritma

Algoritma adalah urutan langkah-langkah untuk memecahkan suatu persoalan. Terdapat beberapa definisi lain dari algoritma tetapi pada prinsipnya senada dengan definisi yang diungkapkan di atas yang dikutip dari beberapa literatur antara lain: Algoritma adalah deretan langkah-langkah komputasi yang mentransformasikan data masukan menjadi luaran [COR90]. Algoritma adalah deretan instruksi yang jelas untuk memecahkan persoalan, yaitu untuk memperoleh luaran yang diinginkan dari suatu masukan dalam jumlah waktu yang terbatas [LEV03]. Algoritma adalah prosedur komputasi yang terdefinisi dengan baik yang menggunakan beberapa nilai sebagai masukan dan menghasilkan beberapa nilai yang disebut keluaran jadi algoritma adalah deretan langkah komputasi yang mentransformasikan masukan menjadi keluaran [COR82]. (Munir, 2016).

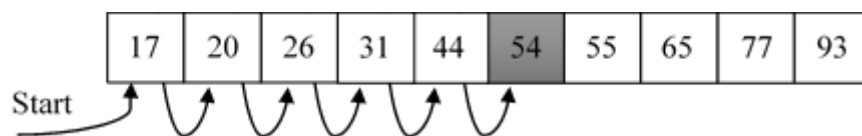
2.2.2 Algoritma Pencarian

Pencarian (*searching*) merupakan proses yang fundamental di dalam pengolahan data. Pencarian adalah proses menemukan nilai (data) tertentu di dalam sekumpulan data yang terstruktur. Sebagai contoh, untuk mengubah

(*update*) data tertentu di dalam sebuah larik, maka aktivitas awal yang harus dilakukan adalah mencari keberadaan data tersebut di dalam larik. Jika data yang dicari ditemukan, maka data tersebut dapat diubah nilainya dengan nilai yang baru. Aktivitas awal yang sama juga dilakukan pada proses penambahan (*insert*) data baru ke dalam larik (tidak boleh ada duplikasi). Proses penambahan data dimulai dengan mencari apakah data baru sudah terdapat di dalam larik. Jika sudah ada, maka data tersebut tidak perlu dimasukkan, tetapi jika belum ada, maka masukkan data baru tersebut ke dalam larik. (Munir, 2016).

2.2.3 Sequential Searching

Algoritma pencarian yang paling sederhana, yaitu Algoritma Pencarian Beruntun (*Sequential Searching*). Nama lain algoritma pencarian beruntun adalah pencarian lurus (*Linear Search*). Pada dasarnya, algoritma pencarian beruntun adalah proses membandingkan setiap elemen larik satu per satu secara beruntun, mulai dari elemen pertama, sampai dengan elemen yang dicari ditemukan atau seluruh elemen sudah habis diperiksa. (Munir, 2016).



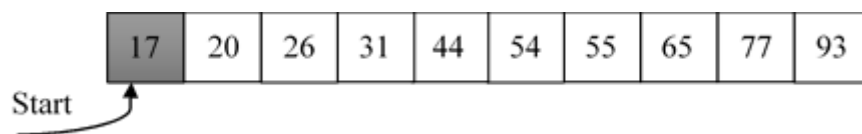
Gambar 2.1 Contoh 1 *Sequential Searching*

Contoh pada gambar 2.1 diatas dengan jumlah $n=10$ elemen:

Misalkan nilai yang dicari $x=54$

Elemen yang dibandingkan: 17, 20, 26, 31, 44, 54 (ditemukan)

Ditemukan pada $idx=6$

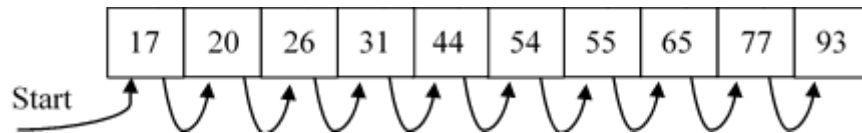


Gambar 2.2 Contoh 2 *Sequential Searching*

Misalkan nilai yang dicari adalah: $x=17$

Elemen yang dibandingkan: 17 (ditemukan)

Ditemukan pada $idx=1$



Gambar 2.3 Contoh 3 *Sequential Searching*

Misalkan nilai yang dicari adalah: $x=15$

Elemen yang dibandingkan: 17, 20, 26, 31, 44, 54, 55, 65, 77, 93 (tidak ditemukan)

Tidak ditemukan, maka $idx=-1$

Metode Sequential Searching atau pencarian beruntun dapat digunakan untuk melakukan pencarian data baik pada Array yang sudah terurut maupun belum terurut. Proses yang terjadi pada metode pencarian ini adalah sebagai berikut:

1. Membaca *Array* data
2. Menentukan data yang dicari
3. Mulai dari data pertama sampai dengan data terakhir data yang dicari dibandingkan dengan masing-masing data di dalam *Array*.

Jika data yang dicari tidak ditemukan maka semua data dibandingkan sampai selesai. Jika data yang dicari ditemukan maka perbandingan akan dihentikan. (Sonita, Sari, 2018).

2.2.4 Arsip

2.2.4.1 Pengertian Arsip

Kata istilah Arsip meliputi 3 pengertian yaitu:

1. Kumpulan naskah atau dokumen yang disimpan.

2. Gedung (ruang) penyimpanan kumpulan naskah atau dokumen.
3. Organisasi atau lembaga yang mengelola dan menyimpan kumpulan naskah atau dokumen.

Pada kenyataannya, pengertian arsip bukan hanya kertas saja, tetapi dapat berarti naskah, buku, foto, film, mikrofilm, rekaman suara, gambar peta, gambar bagan, dan dokumen-dokumen lain dalam segala macam bentuk dan sifatnya, asli atau salinan serta dengan segala macam penciptaannya, dan yang dihasilkan atau diterima oleh suatu organisasi/badan, sebagai bukti dari tujuan organisasi, fungsi, prosedur pekerjaan atau kegiatan pemerintah lainnya atau karena pentingnya informasi yang terkandung di dalamnya. (Sedarmayanti, 2018).

Pengertian arsip di negara indonesia, diatur dalam Undang-Undang Nomor 7 tahun 1971 tentang: “KETENTUAN POKOK KEARSIPAN” pada Bab 1 Pasal 1 berbunyi sebagai berikut:

Arsip adalah

1. Naskah yang dibuat dan diterima oleh Lembaga Negara dan Badan-badan Pemerintah dalam bentuk corak apapun, baik dalam keadaan tunggal maupun berkelompok dalam rangka pelaksanaan kegiatan Pemerintah.
2. Naskah-naskah yang dibuat dan diterima oleh Badan-badan Swasta dan/atau perorangan dalam bentuk corak apapun, baik dalam keadaan tunggal maupun berkelompok, dalam rangka pelaksanaan kehidupan kebangsaan.

2.2.4.2 Peranan Arsip

Sebagai sumber informasi, maka arsip dapat membantu mengingatkan dalam rangka pengambilan keputusan secara cepat dan tepat mengenai sesuatu masalah. Oleh karena itu, dapat disimpulkan bahwa peranan arsip adalah sebagai berikut:

1. Alat utama ingatan organisasi.
2. Bahan atau alat pembuktian.
3. Bahan dasar perencanaan dan pengambilan keputusan.
4. Barometer kegiatan suatu organisasi mengingat setiap kegiatan pada umumnya menghasilkan arsip.
5. Bahan informasi kegiatan ilmiah lainnya. (Sedarmayanti, 2018).

2.2.4.3 Tujuan kearsipan dan tugas pokok unit kearsipan

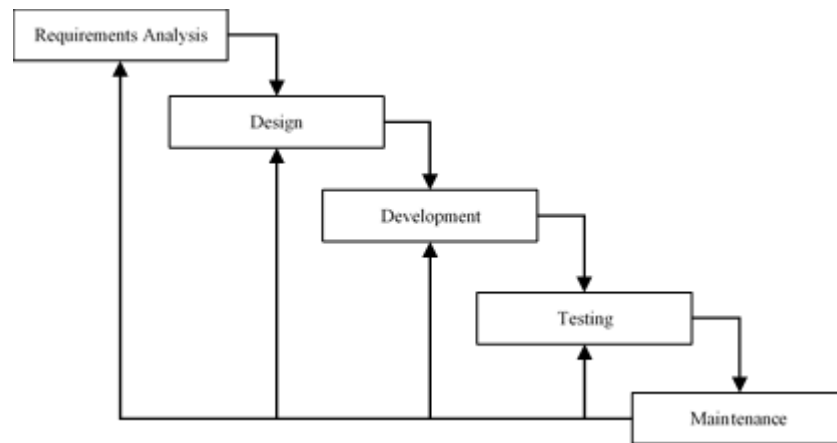
Tujuan kearsipan secara umum adalah untuk menjamin keselamatan bahan pertanggung jawaban nasional tentang rencana, pelaksanaan dan penyelenggaraan kehidupan kebangsaan, serta untuk menyediakan bahan pertanggung jawaban tersebut bagi kegiatan pemerintah.

Tugas pokok unit kearsipan pada dasarnya adalah sebagai berikut:

1. Menerima warkat.
 2. Mencatat warkat.
 3. Mendistribusikan warkat sesuai kebutuhan.
 4. Menyimpan, menata dan menemukan kembali arsip sesuai dengan sistem tertentu.
 5. Memberikan pelayanan kepada pihak-pihak yang memerlukan arsip.
 6. Mengadakan perawatan/pemeliharaan arsip.
 7. Mengadakan atau merencanakan penyusutan arsip, dan lain-lain.
- (Sedarmayanti, 2018).

2.2.5 Waterfall

Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (*support*). Berikut adalah gambar model air terjun:



Gambar 2.4 Model waterfall

2.2.5.1 Requirement Analysis

Requirement Analysis atau analisis kebutuhan perangkat lunak, proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2.2.5.2 Design

Desain perangkat lunak adalah proses multistep yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

2.2.5.3 Development

Development atau pembuatan kode program, desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

2.2.5.4 Testing

Testing atau pengujian fokus pada perangkat lunak dari segi *logic* dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

2.2.5.5 Maintenance

Maintenance atau pemeliharaan adalah proses pendukung (*support*). Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru. (Rosa dan Shalahuddin, 2018).

2.2.6 XAMPP

Sebetulnya, jika anda ingin supaya sebuah *web* dinamis anda (misal dalam PHP) bisa diakses secara lokal menggunakan *web server* lokal, maka anda harus instal PHP (untuk *script*), Apache (*web server*) dan MySQL (*database*) satu demi satu. Namun sekarang, ada beberapa komunitas programmer yang menyediakan solusi praktis dalam instal aplikasi-aplikasi tadi. Contohnya XAMPP.

XAMPP merupakan sebuah aplikasi *open source* yang di dalamnya berisi kumpulan kebutuhan ketika akan membuat suatu aplikasi *web* yang diantaranya seperti yang disebutkan diatas. XAMPP sendiri mempunyai arti di setiap huruf nya diantaranya:

- X = yang berarti *cross platform* karena XAMPP bisa di jalankan di Windows, Linux, Mac dan sebagainya.
- A = yang berarti Apache sebagai *web server*-nya.
- M = yang berarti MySQL sebagai *Database Management System (DBMS)*-nya.
- PP = yang berarti PHP dan Perl sebagai bahasa yang digunakannya.
(Hidayatulloh dan Kawistara, 2017).

2.2.7 HyperText Markup Language (HTML)

HyperText Markup Language (HTML) adalah bahasa *standard* yang digunakan untuk menampilkan halaman *web*. Yang bisa dilakukan dengan HTML yaitu:

- Mengatur tampilan dari halaman *web* dan isinya.
- Membuat tabel dalam halaman *web*.
- Mempublikasikan halaman *web* secara *online*.
- Membuat *form* yang bisa digunakan untuk menangani registrasi dan transaksi via *web*.
- Menambahkan objek-objek seperti citra, audio, video, animasi, java applet dalam halaman *web*.
- Menampilkan area gambar (*canvas*) di *browser*.
(Hidayatulloh dan Kawistara, 2017).

2.2.8 Cascading Style Sheet (CSS)

Sebuah *website* bisa terdiri dari berpuluh-puluh bahkan beratus-ratus halaman. Jika setiap kita mengubah halaman *website* tersebut kita harus merubah formatnya satu per satu maka akan sangat repot. Namun, jika kita menggunakan

CSS maka hal diatas bukan lagi sebuah masalah karena dengan CSS kita bisa menyimpan format dan menggunakannya kapanpun dan dimanapun kita inginkan.

Cascading Style Sheet (CSS) adalah salah satu bahasa desain *web (style sheet language)* yang mengontrol format tampilan sebuah halaman *web* yang ditulis dengan menggunakan penanda *markup language*. Biasanya CSS digunakan untuk mendesain sebuah halaman HTML atau XHTML. (Hidayatulloh dan Kawistara, 2017).

2.2.9 PHP

PHP adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi *web*. Ketika dipanggil dari *web browser*, program yang ditulis dengan PHP akan di-*parsing* di dalam *web server* oleh *interpreter* PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya akan ditampilkan kembali ke *web browser*. Karena pemrosesan program PHP dilakukan di lingkungan *web server*, PHP dikatakan sebagai bahasa sisi *server (server-side)*. Oleh karena itu, seperti yang telah dikemukakan sebelumnya, kode PHP tidak akan terlihat pada saat user memilih perintah “*View Source*” pada *web browser* yang mereka gunakan. Selain menggunakan PHP, aplikasi *web* juga dapat dibangun dengan Java (JSP – *JavaServer pages* dan *Servlet*), Perl, Python, Ruby maupun ASP (*Active Server Pages*). (Raharjo, 2016).

2.2.10 Bootstrap

Bootstrap merupakan sebuah *framework* CSS yang paling banyak diminati oleh para *developer website*. *Class-class* CSS dalam *bootstrap* sudah dibakukan sehingga pengerjaan sebuah *project* berbasis *web* menjadi semakin mudah dilakukan secara bersama-sama dalam sebuah tim kita dengan mudah dapat mendesain tampilan *website* yang responsif dengan menggunakan *bootstrap*. Responsif maksudnya adalah lebar halaman *website* akan disesuaikan secara otomatis berdasarkan perangkat yang digunakan untuk mengakses nya

baik itu ketika diakses menggunakan PC laptop tablet ataupun *smartphone* sehingga *website* akan menyesuaikan dengan lebar perangkat yang digunakan pengunjung. (Kaban, 2019).

2.2.11 Database

Secara umum, *database* dapat didefinisikan sebagai kumpulan tabel (walaupun lebih tepat dikatakan kumpulan objek karena yang dikandung di dalam database sebenarnya bukan hanya tabel, tapi ada indeks, *view*, *constraint*, *trigger*, dan sebagainya). Dalam sistem *database* relasional atau RDBMS (*Relational Database Management System*), tabel-tabel tersebut harus saling berelasi melalui kolom-kolom yang ada berdasarkan aturan-aturan tertentu. (Raharjo, 2016).

Sistem basis data (*database*) adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. (Rosa dan Shalahuddin, 2018).

2.2.12 Database Management System (DBMS)

DBMS (*Database Management System*) atau dalam bahasa Indonesia sering disebut sebagai Sistem Manajemen Basis Data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data. Suatu sistem aplikasi disebut DBMS jika memenuhi persyaratan minimal sebagai berikut:

- Menyediakan fasilitas untuk mengelola akses data.
- Mampu menangani integritas data.
- Mampu menangani akses data.
- Mampu menangani *backup* data.

(Rosa dan Shalahuddin, 2018).

2.2.13 SQL

SQL (biasa dibaca Sequel) adalah kependekan dari *Structured Query Language*. SQL merupakan bahasa yang dirancang untuk berkomunikasi dengan *database*. Tidak seperti bahasa pemrograman (seperti PHP, C, Python, dan sebagainya), SQL hanya memiliki beberapa kata kunci saja. Tujuan dirancangnya SQL adalah untuk menyederhanakan kita dalam membaca atau menulis data dari/ke dalam *database*. (Raharjo, 2016).

SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus. (Rosa dan Shalahuddin, 2018).

2.2.14 MySQL

MySQL adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh para pemrograman aplikasi *web*. Kelebihan dari MySQL adalah gratis, handal, selalu di-*update* dan banyak forum yang memfasilitasi para pengguna jika memiliki kendala. MySQL juga menjadi DBMS yang sering dibanding dengan *web server* sehingga proses instalasi nya jadi lebih mudah. (Hidayatulloh dan Kawistara, 2017).

2.2.15 Sublime Text

Sublime Text adalah editor kode sumber lintas *platform shareware* dengan antarmuka pemrograman aplikasi Python (API). Ini secara asli mendukung banyak bahasa pemrograman dan bahasa *markup*, dan fungsi dapat ditambahkan oleh pengguna dengan *plugin*, biasanya dibangun komunitas dan dikelola di bawah lisensi perangkat lunak bebas. (wikipedia.org)

Berikut beberapa fitur yang terdapat pada *sublime text*:

1. *Goto Anything*

Gunakan *Goto Anything* untuk membuka file hanya dengan beberapa penekanan tombol, dan langsung beralih ke simbol, garis atau kata-kata.

2. *Goto Definition*

Menggunakan informasi dari definisi sintaksis, *sublime text* secara otomatis menghasilkan indeks seluruh proyek dari setiap kelas, metode dan fungsi. Indeks ini memperkuat *Goto Definition*, yang diekspos dalam tiga cara berbeda:

- Munculan ditampilkan saat melayang di atas simbol.
- Menekan F12 ketika tanda sisipan berada pada simbol.
- Simbol *Goto Definition* dalam fungsionalitas proyek.

Pengindeksan simbol dapat dikustomisasi berdasarkan per-sintaks melalui file konfigurasi, yang memungkinkan pengguna untuk menyesuaikan fitur dengan kebutuhan mereka.

3. Banyak pilihan

Buat sepuluh perubahan pada saat yang sama, tidak satu perubahan sepuluh kali. Pilihan ganda memungkinkan untuk mengubah banyak baris secara bersamaan, mengubah nama variabel dengan mudah, dan memanipulasi file lebih cepat dari sebelumnya. Dengan menekan Ctrl+Shift+L untuk membagi pilihan menjadi beberapa baris dan Ctrl+D untuk memilih kemunculan kata berikutnya yang dipilih.

4. *Command Palette*

Command Palette memiliki fungsi yang jarang digunakan, seperti menyortir, mengubah sintaks, dan mengubah pengaturan indentasi. Dengan hanya beberapa penekanan tombol, Anda dapat mencari apa yang diinginkan, tanpa harus menavigasi menu atau mengingat ikatan kunci yang tidak jelas. Tampilkan *Command Palette* dengan Ctrl + Shift + P.

5. API dan ekosistem paket yang kuat

Sublime text memiliki API Python yang kuat dan memungkinkan *plugin* untuk menambah fungsionalitas bawaan. Kontrol paket dapat di instal melalui *Command Palette*, memberikan akses mudah ke ribuan paket yang dibangun oleh komunitas.

6. *Customize anything*

Binding kunci, menu, snippet, macro, perlengkapan dan lainnya hampir semua yang ada di *sublime text* dapat disesuaikan dengan file JSON sederhana. Sistem ini memberi fleksibilitas karena pengaturan dapat ditentukan berdasarkan jenis per file dan per proyek.

7. Pengeditan terpisah

Dapatkan hasil maksimal dari monitor layar lebar dengan dukungan pengeditan terpisah. Edit file berdampingan, atau edit dua lokasi dalam satu file. Kita dapat mengedit dengan baris dan kolom sebanyak yang Anda inginkan. Manfaatkan banyak monitor dengan mengedit dengan banyak jendela, dan menggunakan banyak pemisahan di setiap jendela.

8. Peralihan proyek instan

Proyek dalam *sublime text* menangkap seluruh isi ruang kerja, termasuk file yang dimodifikasi dan belum disimpan. Kita dapat beralih antar proyek dengan cara yang mirip dengan *Goto anything*, dan peralihannya instan, tanpa konfirmasi penyelamatan - semua modifikasi Anda akan dipulihkan saat proyek dibuka.

9. Lintas *platform*

Sublime text tersedia untuk Mac, Windows dan Linux. Satu lisensi adalah semua yang Anda perlukan untuk menggunakan *Sublime text* di setiap komputer yang kita miliki, apapun sistem operasi yang digunakannya. *Sublime text* menggunakan *toolkit* UI khusus, dioptimalkan untuk kecepatan dan keindahan, sekaligus memanfaatkan fungsionalitas asli pada setiap platform. (sublimetext.com)

2.2.16 *Web Browser*

Penjelajah web, peramban web atau penelusur web (*web browser*) adalah perangkat lunak yang berfungsi untuk menerima dan menyajikan sumber informasi dari Internet. Sebuah sumber informasi diidentifikasi dengan pengidentifikasian sumber seragam yang dapat berupa halaman *web*, gambar,

video, atau jenis konten lainnya. Meskipun penjelajah web terutama ditujukan untuk mengakses Internet, sebuah penjelajah juga dapat digunakan untuk mengakses informasi yang disediakan oleh *server web* dalam jaringan pribadi atau berkas pada sistem berkas. Beberapa penjelajah *web* yang populer adalah Google Chrome, Firefox, Internet Explorer, Opera, dan Safari.

Penjelajah *web* pertama bernama WorldWideWeb (tanpa spasi) diciptakan oleh Tim Berners-Lee. Nama penjelajah tersebut kemudian diubah menjadi Nexus. Pada tahun 1993, Marc Andreessen melakukan inovasi penjelajah *web* dengan merilis Mosaic (kemudian Netscape), "peramban web populer pertama di dunia", yang membuat sistem internet lebih mudah digunakan dan dapat diakses oleh lebih banyak orang. Penjelajah *web* Andreessen memicu ledakan popularitas di internet pada tahun 1990-an. Andreessen, pemimpin tim Mosaic di NCSA, segera mendirikan perusahaan sendiri, bernama Netscape dan merilis Mosaic-yang kemudian mempengaruhi Netscape Navigator pada tahun 1994, yang dengan cepat menjadi penjelajah yang paling populer di dunia, dengan menguasai 90% dari seluruh penggunaan penjelajah *web* di dunia. (Wikipedia.org).

2.2.17 Web Hosting

Layanan *web hosting* adalah jenis layanan *hosting* internet yang memungkinkan individu dan organisasi untuk membuat situs *web* mereka dapat diakses melalui World Wide Web. *Host web* adalah perusahaan yang menyediakan ruang pada *server* yang dimiliki atau disewa untuk digunakan oleh klien, serta menyediakan konektivitas internet, biasanya di pusat data. *Host web* juga dapat menyediakan ruang pusat data dan konektivitas ke internet untuk *server* lain yang terletak di pusat data mereka, yang disebut *colocation*. (Wikipedia.org).

Layanan *hosting* internet dapat menjalankan *server web*. Cakupan layanan *hosting web* sangat bervariasi, berikut adalah beberapa jenis *hosting* internet:

1. *Shared web hosting service*

Situs *web* seseorang ditempatkan di *server* yang sama dengan banyak situs lain, mulai dari beberapa situs sampai ratusan situs.

2. *Reseller web hosting*

Mengizinkan klien menjadi *host web* sendiri. *Reseller* dapat berfungsi untuk masing-masing domain di bawah kombinasi apapun jenis *hosting* yang terdaftar.

3. *Virtual dedicated server*

Membagi sumber daya *server* menjadi *server virtual*, dimana sumber daya dapat dialokasikan dengan cara yang tidak secara langsung.

4. *Dedicated hosting service*

Pengguna mendapatkan *server web* sendiri dan mendapatkan kontrol penuh.

5. *Managed hosting service*

Pengguna mendapatkan *server web* sendiri tetapi tidak diizinkan untuk kontrol penuh.

6. *Colocation web hosting service*

Mirip dengan layanan *hosting* khusus, tetapi pengguna memiliki *colocation server*.

7. *Cloud hosting*

Jenis *hosting* yang memungkinkan pelanggan *hosting* yang kuat, terukur, dan andal berdasarkan *server* yang seimbang.

8. *Clustered hosting*

Memiliki beberapa *server hosting* konten yang sama untuk pemanfaatan sumber daya yang lebih baik.

9. *Grid hosting*

Bentuk *hosting* terdistribusi ini adalah ketika *Cluster server* bertindak seperti kisi dan terdiri dari beberapa *node*.

10. *Home server*

Biasanya satu mesin ditempatkan di tempat tinggal pribadi dan dapat digunakan untuk meng-*host* satu atau lebih situs *web*.

2.2.18 *Unified Modelling Language (UML)*

Unified Modelling Language (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. (Rosa dan Shalahuddin, 2018).

Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak.

Seperti yang diketahui bahwa menyatukan banyak kepala untuk menceritakan sebuah ide dengan tujuan untuk memahami hal yang sama tidaklah mudah, oleh karena itu diperlukan sebuah bahasa pemodelan perangkat lunak yang dapat dimengerti oleh banyak orang.

Pada penelitian kali ini penyusun menggunakan beberapa jenis diagram bahasa pemodelan untuk membuat sebuah rancangan aplikasi arsip elektronik berbasis *web* di UPK Artha Raharja Kecamatan Pacet. Berikut adalah beberapa jenis diagram yang digunakan:

a. *Use Case Diagram*





Use case diagram merupakan pemodelan untuk melakukan (*behaviour*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat secara kasar yang digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. (Rosa dan Shalahuddin, 2018).



Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

- Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah orang, tapi aktor belum tentu merupakan orang.
- *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor

Simbol-simbol yang digunakan pada *use case* diagram bisa dilihat pada tabel 2.1.

Tabel 2.1 Simbol *Use Case Diagram*


NO	Simbol	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
3		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
4		<i>Extend</i>	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan

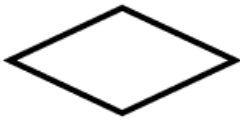


5		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
6		<i>Include</i>	Menspesifikasikan bahwa use case sumber secara eksplisit.

b. *Activity Diagram*

Activity Diagram menggambarkan *workflow* atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor jadi aktivitas yang dapat dilakukan oleh system. (Rosa dan Shalahuddin, 2018).

Tabel 2.2 Simbol *Activity Diagram*

NO	Simbol	Nama	Keterangan
1		Status Awal	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
2		Aktivitas	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor

3		Percabangan	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4		Penggabungan	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5		Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

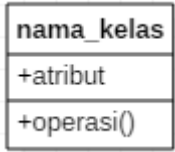






c. *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode operasi.

- Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
- operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Diagram kelas dibuat agar pembuat program membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Berikut adalah simbol-simbol yang dipakai dalam membuat kelas diagram:

Tabel 2.3 Simbol *Class Diagram*

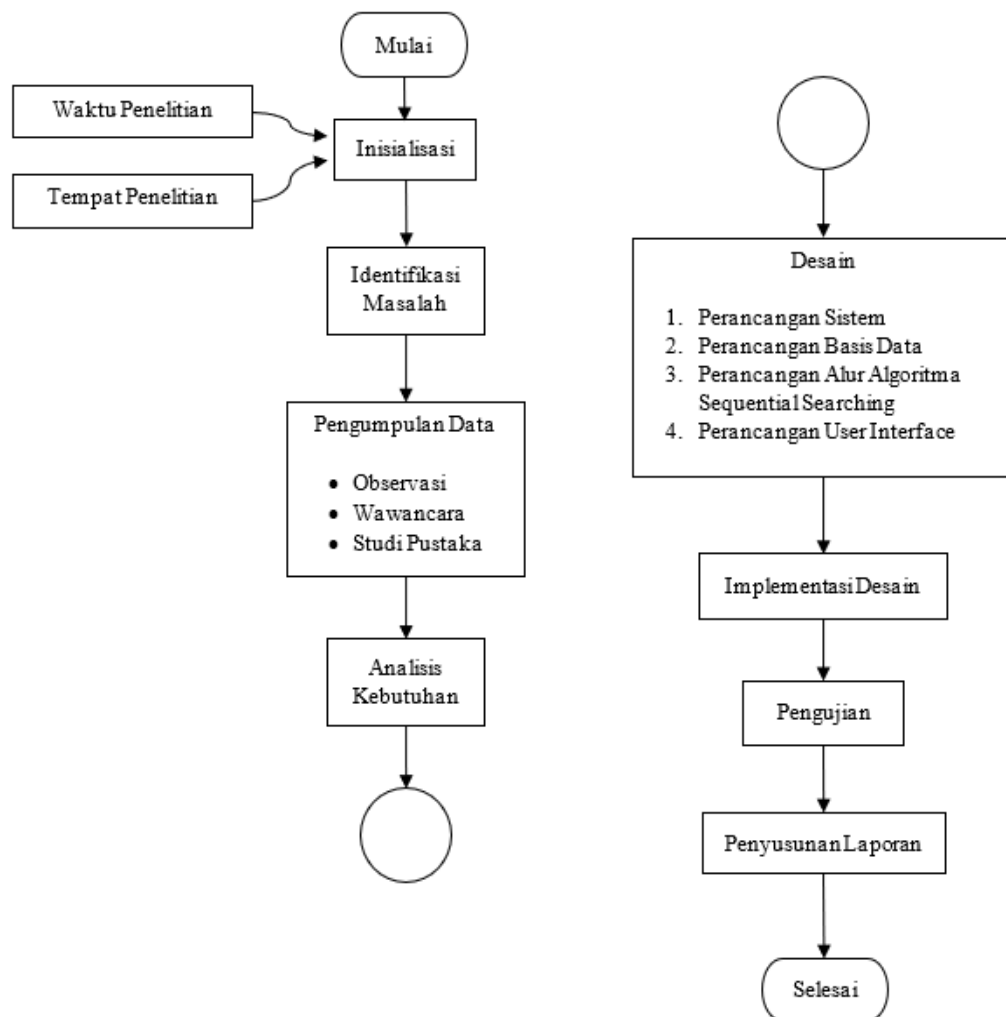
Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur sistem
<p>Antarmuka</p> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
<p>Asosiasi</p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p>Asosiasi berarah</p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
<p>Kebergantungan</p> 	Kebergantungan antar kelas
<p>Agregasi</p> 	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

BAB III

METODOLOGI PENELITIAN

3.1 Kerangka Pikir

Kerangka pikir ini merupakan langkah-langkah yang akan dilakukan dalam penyelesaian masalah pada penelitian ini. Adapun kerangka kerja penelitian yang digunakan seperti terlihat pada gambar berikut:



Gambar 3.1 Kerangka Pikir

3.2 Deskripsi

3.2.1 Inisialisasi

Pada tahap ini dilakukan penentuan tempat penelitian serta menyesuaikan waktu penelitian yang ditetapkan oleh Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Bale Bandung. Setelah waktu dan tempat ditentukan selanjutnya dilakukan proses pengumpulan data dan identifikasi masalah untuk dijadikan bahan serta judul penelitian.

3.2.1.1 Tempat Penelitian

Penelitian dilaksanakan di Unit Pengelola Kegiatan Artha Raharja Kecamatan Pacet yang beralamat di Kp. Butul RT/ RW 04/05 Desa Cipeujeuh Kecamatan Pacet Kabupaten Bandung.

3.2.1.2 Waktu Penelitian

Penelitian tentang Implementasi Algoritma *Sequential Searching* Pada Pembuatan Aplikasi Arsip Elektronik Berbasis Web ini dilakukan selama 5 bulan, dari mulai bulan maret sampai dengan bulan juli 2020.

3.2.2 Identifikasi Masalah

Pada tahap ini penyusun melakukan identifikasi masalah sebagai upaya mendefinisikan *problem* dan membuat definisi tersebut dapat diukur sebagai langkah awal penelitian. Pada tahap ini diambil beberapa masalah yang terjadi di UPK Artha Raharja Kecamatan Pacet diantaranya:

1. Sistem pengarsipan surat yang masih menggunakan pencatatan di buku agenda surat secara manual berakibat pada penggunaan buku jika pengarsipan dilakukan dalam jangka waktu yang panjang.
2. Semakin lama proses pengarsipan maka semakin banyak berkas surat masuk maupun surat keluar yang harus disimpan sehingga memerlukan

tempat penyimpanan yang lebih besar sampai waktu penghapusan berkas.

3. Keamanan berkas surat masuk dan surat keluar yang disimpan di lemari arsip tidak terjamin diantaranya berkas hilang atau sobek, tentunya hal tersebut menjadi masalah jika sewaktu-waktu berkas tertentu diperlukan kembali.
4. Proses pencarian berkas surat masuk maupun surat keluar memerlukan waktu yang cukup lama karena harus mencari satu persatu di lemari penyimpanan arsip.

Berdasar masalah yang ditemukan dari hasil, maka dilakukan pengembangan sistem dengan pembangunan aplikasi arsip elektronik untuk mempermudah proses pengarsipan di UPK Artha Raharja Kecamatan Pacet.

3.2.3 Pengumpulan Data

Dalam pelaksanaan penelitian ini diperlukan data dan informasi terkait yang nantinya akan digunakan sebagai bahan rujukan untuk proses penyelesaian masalah maupun dalam pengembangan sistem. Adapun metodologi pengumpulan data yang digunakan dalam penelitian ini adalah observasi, wawancara dan studi pustaka.

3.2.3.1 Observasi

Observasi dilakukan langsung di kantor UPK Artha Raharja Kecamatan Pacet. Tujuan dilakukannya observasi adalah untuk melihat dan mengamati secara langsung bagaimana sistem pengarsipan khususnya surat menyurat yang sedang berjalan dan mengetahui masalah-masalah apa saja yang timbul pada sistem yang sedang digunakan. Kemudian dari masalah-masalah yang ditemukan dapat dianalisa sistem seperti apa yang akan dikembangkan untuk membantu proses pengarsipan surat masuk dan surat keluar di UPK Artha Raharja Kecamatan Pacet agar lebih mudah dan efisien.

3.2.3.2 Wawancara

Wawancara dilakukan secara langsung dengan pihak terkait yaitu bagian sekretariat UPK Artha Raharja Kecamatan Pacet yang bertugas dalam proses pengarsipan surat menyurat. Tujuan dilakukannya wawancara untuk mengetahui secara detail bagaimana alur sistem pengarsipan yang sedang berjalan sebagai data untuk dilakukan pemecahan masalah dengan pengembangan dari sistem yang ada.

3.2.3.3 Studi Pustaka

Pada tahap ini penyusun melakukan pencarian data, membandingkan dan mempelajari penelitian sebelumnya yang berkaitan dengan topik penelitian ini seperti beberapa jurnal maupun buku yang digunakan sebagai referensi seperti yang tercantum di BAB II.

3.2.4 Analisis Kebutuhan

Pada tahap analisis kebutuhan penyusun mengumpulkan data-data dari hasil observasi dan wawancara untuk memenuhi kebutuhan aplikasi yang akan dibuat. Dalam menyelesaikan masalah berdasarkan hasil observasi dan wawancara, di ajukan rancangan pengembangan sistem dimana sistem pengarsipan konvensional di kembangkan menjadi sistem komputerisasi yaitu aplikasi arsip elektronik berbasis *web* berdasarkan sistem pengarsipan yang sedang berjalan.

Kesimpulan dari analisis kebutuhan berdasarkan data-data yang diperoleh dari hasil observasi dan wawancara diantaranya:

1. Aplikasi arsip elektronik harus dapat mencatat data agenda surat sehingga tidak memerlukan buku agenda surat karena data surat disimpan secara digital di dalam aplikasi.
2. Aplikasi arsip elektronik harus dapat menyimpan berkas surat masuk dan surat keluar dengan cara di *scan* dan hasil *scan* tersebut disimpan di

database aplikasi sehingga tidak memerlukan ruang penyimpanan dan terhindar dari kerusakan berkas atau hilang.

3. Aplikasi arsip elektronik harus dapat mempermudah proses pencarian berkas surat dalam *database* dengan adanya fitur pencarian.
4. Dalam fitur pencarian penyusun melakukan percobaan dengan melakukan penerapan algoritma *Sequential Searching*, dan melakukan pengujian apakah algoritma *Sequential Searching* dapat diterapkan di aplikasi arsip elektronik yang akan dibuat.

3.2.5 Desain

Setelah selesai dengan pengumpulan data dan analisis kebutuhan selanjutnya dilakukan tahap desain. Proses *Design* atau desain sistem meliputi proses gambaran alur kerja sistem, alur algoritma pencarian *sequential searching*, basis data (*database*), dan tampilan (*user interface*).

3.2.5.1 Perancangan Sistem

Perancangan sistem dilakukan bertujuan untuk mendapatkan gambaran jelas tentang alur aplikasi seperti apa yang akan dibuat dan dikerjakan yang akan membantu dalam proses pembuatan aplikasi. Perancangan sistem dilakukan dengan menggunakan pemodelan *Unified Modelling Language* (UML).

3.2.5.2 Perancangan Basis Data

Perancangan basis data dilakukan bertujuan untuk menentukan isi dan pengaturan data yang dibutuhkan untuk mendukung rancangan sistem yang akan dibuat. Perancangan basis data akan dibagi menjadi tiga tabel diantaranya tabel surat masuk untuk menampung data surat masuk, tabel surat keluar untuk menampung data surat keluar, dan tabel pengguna untuk menampung data pengguna.

3.2.5.3 Perancangan Algoritma *Sequential Searching*

Perancangan algoritma *sequential searching* dilakukan penyusun bertujuan untuk mempermudah dalam proses implementasi dimana algoritma *sequential searching* dibuat dalam bentuk *flowchart* untuk mempermudah menggambarkan alur prosesnya dan dibuat kedalam bentuk program.

3.2.5.4 Perancangan *User Interface*

Perancangan *user interface* dilakukan penyusun bertujuan untuk menggambarkan tampilan yang akan dibuat agar mempermudah pada tahap implementasi program. *User interface* dibuat sesederhana mungkin untuk memudahkan pengguna dalam mengakses aplikasi yang akan dibuat nantinya.

3.2.6 Implementasi Desain

Pada tahap implementasi desain, perancangan yang telah dibuat di terjemahkan kedalam bentuk pengkodean secara nyata ke dalam bahasa pemrograman. Pada aplikasi ini implementasi kode dibuat ke dalam bahasa pemrograman PHP dan MySQL sebagai *database*. Implementasi kode untuk membentuk fungsi-fungsi yang dibutuhkan oleh program dibuat menggunakan bantuan *framework* bootstrap agar sesuai dengan keinginan pengguna dan implementasi dapat berjalan cepat dan mudah. Tahap ini dibuat sesuai dengan desain yang telah dibuat sebelumnya.

3.2.7 Pengujian

Pengujian atau *testing* dilakukan setelah aplikasi selesai dibuat dan memenuhi kebutuhan-kebutuhan sesuai desain. Pengujian dilakukan untuk memastikan aplikasi berjalan dengan baik dan sesuai dengan sistem yang ada serta memastikan apakah algoritma *sequential searching* dapat diterapkan dan berjalan di aplikasi tersebut.

3.2.8 Pembuatan Laporan

Tahapan pembuatan laporan sebagai salah satu persyaratan kelulusan di Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Bale Bandung. Pembuatan laporan ini terdiri dari hasil identifikasi masalah dan implementasi dalam bentuk laporan tertulis yang terdiri dari 6 bab dengan sistematika penulisan yang tertera pada BAB I.

BAB IV

ANALISIS DAN PERANCANGAN

4.1 Analisis

Analisis dilakukan sebagai langkah awal pada pembangunan dan pengembangan penelitian ini. Dalam analisis ini penyusun mempunyai instrumen penelitian dan melakukan beberapa tahapan yaitu, analisis kebutuhan, analisis sistem dan hasil analisis.

4.1.1 Instrumen Penelitian

Instrumen penelitian pada penelitian kali ini terdiri dari perangkat lunak dan perangkat keras yang dibutuhkan penyusun untuk membangun sebuah aplikasi. Berikut penjelasan yang menjadi instrumen penelitian:

a. Wawancara

Pada tahap wawancara didapatkan data-data sebagai penunjang kebutuhan aplikasi yang akan dibuat untuk menyelesaikan masalah. Kesimpulan berdasarkan data-data yang diperoleh dari hasil wawancara diantaranya:

- 1) Aplikasi arsip elektronik yang akan dibuat harus dapat mencatat data agenda surat sehingga tidak memerlukan buku agenda surat karena data surat disimpan secara digital di dalam aplikasi.
- 2) Aplikasi arsip elektronik yang akan dibuat harus dapat menyimpan berkas surat masuk dan surat keluar dengan cara di *scan* dan hasil *scan* tersebut disimpan di *database* aplikasi sehingga tidak memerlukan ruang penyimpanan dan terhindar dari kerusakan berkas atau hilang.
- 3) Aplikasi arsip elektronik yang akan dibuat dapat mempermudah proses pencarian berkas surat dalam *database* dengan adanya fitur pencarian.
- 4) Algoritma *Sequential Searching* dapat diterapkan pada aplikasi dan dapat berjalan untuk melakukan sebuah pencarian.

b. Perangkat lunak

Pada penelitian ini penyusun menggunakan beberapa perangkat lunak yang akan menunjang dalam pembuatan aplikasi yang akan dibangun. Aplikasi yang digunakan penyusun pada penelitian ini diantaranya:

1. Sistem Operasi Microsoft Windows 10 Professional 64bit.
2. *Code editor* Sublime Text untuk proses pengkodean aplikasi.
3. Web Browser Google Chrome untuk menampilkan hasil *compile*.
4. XAMPP digunakan untuk *web server*.
5. Star UML untuk Pemodelan perancangan.

c. Perangkat Keras

Pada penelitian ini penyusun menggunakan laptop untuk menunjang pembuatan aplikasi serta pembuatan laporan. Berikut adalah beberapa perangkat keras yang menjadi penunjang dalam instrumen penelitian pada penelitian ini:

Perangkat Keras Laptop:

Tabel 4.1 Spesifikasi Perangkat Keras

Spesifikasi	Deskripsi
Tipe Laptop	Lenovo B41-35
RAM	5GB 1600MHz DDR3L
Hard Disk	500GB HDD 5400rpm
Optical Drive	9.0MM Super Multi ODD
Video Graphics	Radeon R5 GPU integrated Graphics
Battery	4Cell battery 35 Whr up to 5 hours

d. *Server*

Pada penelitian ini penyusun menggunakan *server hosting* sebagai *web server* agar dapat diakses dengan mudah saat pembangunan aplikasi. *server hosting* yang digunakan penyusun dengan spesifikasi sebagai berikut:

Server Hosting:

Tabel 4.2 Spesifikasi *Server Hosting*

Spesifikasi	Deskripsi
Core	1 Core
RAM	512 MB
SSD	2 GB

4.1.2 Analisis Sistem

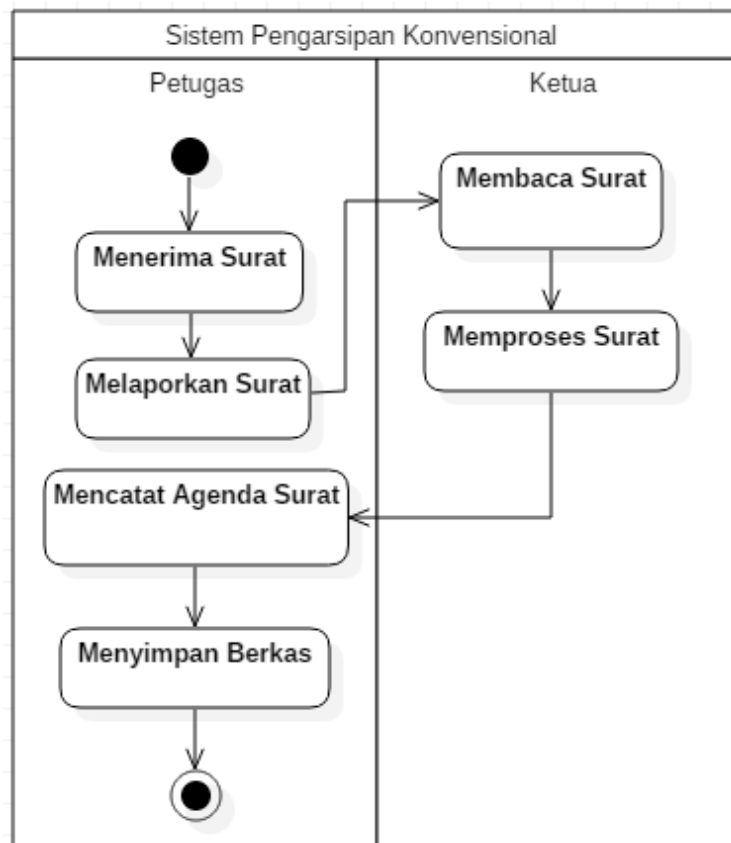
Sistem pengarsipan di UPK Artha Raharja yang sedang berjalan saat ini khususnya pada arsip surat masih menggunakan sistem konvensional yaitu dengan cara mencatat di dalam buku agenda surat dan menyimpan berkas fisiknya di dalam lemari penyimpanan surat. Cara konvensional dalam proses pengarsipan memang sudah berjalan dengan baik namun cara tersebut tentunya belum efisien jika proses pengarsipan dilakukan dalam jangka waktu yang panjang karena akan membutuhkan buku yang tebal untuk pencatatan dan lemari penyimpanan yang cukup untuk menyimpan berkas arsip surat. Terlebih lagi jika akan melakukan sebuah pencarian berkas yang sudah di arsipkan, tentunya petugas akan membutuhkan waktu untuk pencarian berkas tertentu yang akan di cari di dalam tumpukan arsip di dalam lemari surat.

Berdasarkan permasalahan yang dipaparkan di atas, maka penyusun berupaya untuk membantu dengan membangun sebuah aplikasi arsip elektronik berbasis *web* sebagai solusi untuk mempermudah proses pengarsipan di UPK Artha Raharja Kecamatan Pacet. Dimana jika menggunakan aplikasi tersebut proses pendataan tidak memerlukan buku agenda surat dan lemari penyimpanan arsip yang besar karena data dan file surat dapat disimpan di dalam *database* aplikasi. Untuk proses pencarian berkas di dalam arsip juga akan terbantu dengan adanya fitur pencarian di dalam aplikasi tersebut.

Pada fitur pencarian di dalam aplikasi arsip elektronik berbasis *web* di UPK Artha Raharja Kecamatan Pacet penyusun akan melakukan uji coba dengan

menerapkan algoritma *sequential searching*. Dari hasil uji coba nantinya akan didapatkan apakah algoritma *sequential searching* dapat diterapkan dan berjalan dengan baik di dalam aplikasi tersebut.

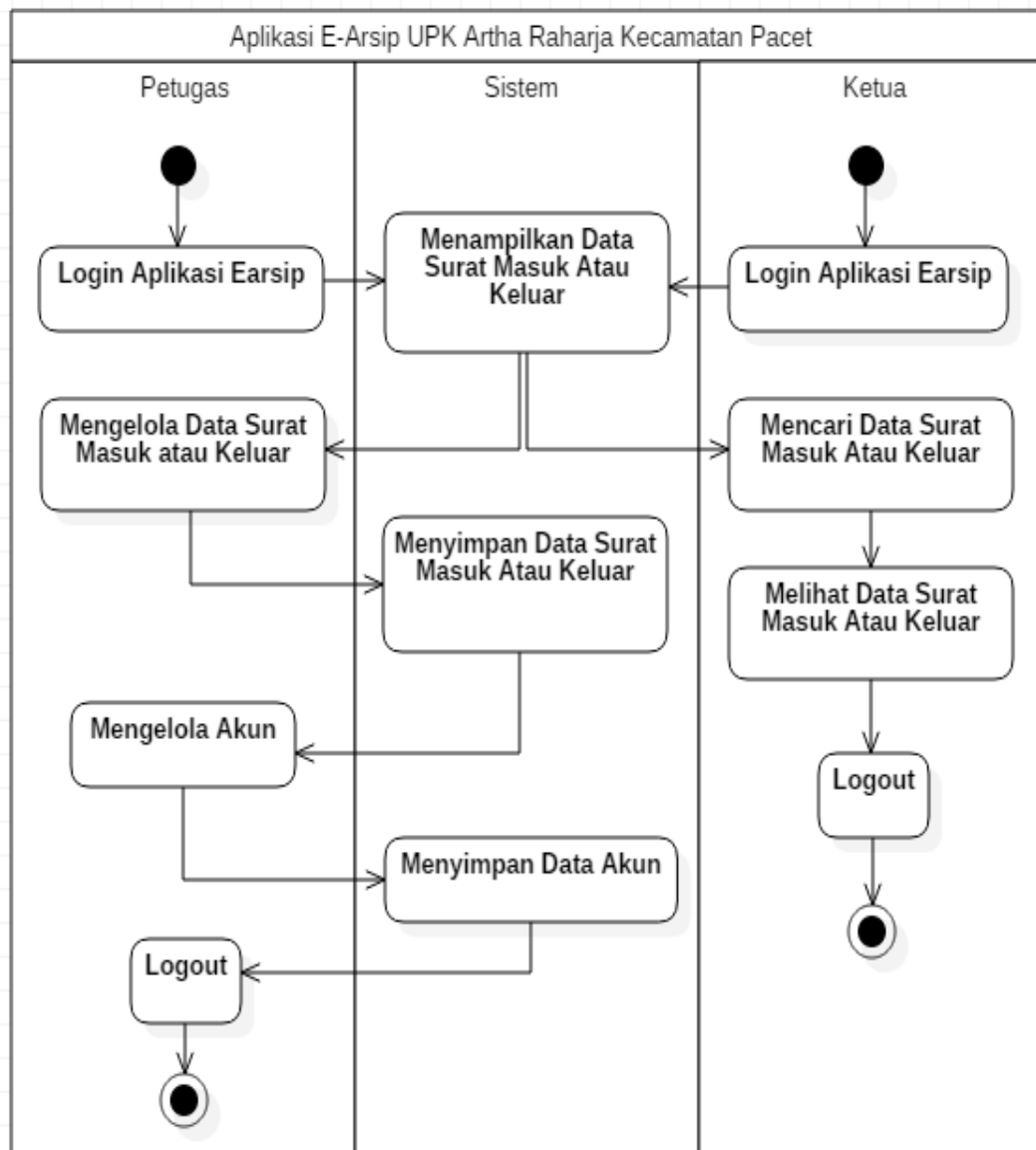
Berikut merupakan sistem pengarsipan konvensional yang sebelumnya dan sedang berjalan di UPK Artha Raharja Kecamatan Pacet:



Gambar 4.1 *flowmap* Sistem Pengarsipan Konvensional

Dari gambar 4.1 di jelaskan alur proses surat sampai menuju ke pengarsipan. Tahap awal petugas sekretariat menerima surat masuk dari instansi lain maupun individu, setelah itu surat dilaporkan kepada ketua untuk diproses lalu surat dikembalikan kepada petugas untuk di catat di buku agenda surat masuk setelah itu berkas surat disimpan untuk diarsipkan.

Berdasarkan sistem arsip konvensional yang sedang berjalan penyusun merekomendasikan rancangan aplikasi berbasis *web* yang akan dibuat untuk kebutuhan proses pengarsipan di UPK Artha Raharja Kecamatan Pacet:



Gambar 4.2 *Flowmap* Aplikasi E-Arsip

Gambar 4.2 merupakan rancangan aplikasi yang akan dibuat. Pada gambar tersebut di jelaskan alur proses pengarsipan surat berdasarkan sistem konvensional yang sedang berjalan dimana surat diterima oleh petugas, lalu petugas melaporkan ke ketua sekaligus menginput ke dalam sistem untuk diarsipkan. Dengan adanya aplikasi tersebut tentunya proses pengarsipan menjadi lebih efisien serta mempermudah untuk melihat kembali dokumen dan berkas surat yang sudah diarsipkan. Hak akses petugas dan ketua dibedakan, dimana petugas dapat menginput, mengubah dan menghapus data. Sedangkan hak akses ketua hanya bisa melihat data dan berkas surat saja.

4.1.3 Analisis Kebutuhan

Analisis kebutuhan menggambarkan keadaan sistem yang ada pada kantor UPK Artha Raharja Kecamatan Pacet di bagian sekretariat khususnya petugas surat, diantaranya kebutuhan perangkat keras, kebutuhan perangkat lunak dan kebutuhan pengguna sebagai bahan analisis kebutuhan yang harus dipenuhi dalam perancangan aplikasi yang akan digunakan.

4.1.3.1 Kebutuhan Perangkat Keras

Perangkat keras komputer atau *hardware* sangat mendukung dalam kinerja sistem. Semakin baik spesifikasi perangkat keras yang ada maka kinerja sistem pun akan semakin baik. Spesifikasi minimum untuk menjalankan aplikasi arsip yang akan digunakan di UPK Artha Raharja Kecamatan Pacet di bagian kearsipan adalah sebagai berikut:

Tabel 4.3 Analisis Kebutuhan Perangkat Keras

Spesifikasi	Deskripsi
Processor	CPU 1 GHz
RAM	2 GB
Hard Disk	80 GB
Video Graphics	DirectX 9 graphic device dengan WDDM driver

Dengan spesifikasi perangkat keras komputer yang digunakan saat ini sudah di UPK Artha Raharja Kecamatan Pacet bisa menjalankan aplikasi *web browser* untuk mengakses internet dan menjalankan aplikasi arsip elektronik berbasis *web* yang akan dibuat. Sehingga spesifikasi perangkat keras yang ada sudah mencukupi jika digunakan untuk proses pengarsipan pada aplikasi yang akan dibangun.

4.1.3.2 Kebutuhan Perangkat Lunak

Dalam melakukan kegiatan pengarsipan, petugas pengelola surat harus menggunakan *web browser* karena aplikasi arsip elektronik yang akan

dibangun berbasis *web*. Komputer yang digunakan sekarang menggunakan Sistem Operasi Windows 7 Professional dan sudah terdapat beberapa aplikasi *web browser* sehingga untuk kebutuhan perangkat lunak atau *software* sudah terpenuhi dan dapat menjalankan aplikasi yang akan dibuat

4.1.3.3 Kebutuhan Akses Internet

Untuk dapat menjalankan aplikasi arsip elektronik berbasis *web* yang akan di bangun harus menggunakan akses internet. Di kantor UPK Artha Raharja Kecamatan Pacet belum ada akses internet, sehingga untuk menjalankan aplikasi tersebut pihak dari UPK Artha Raharja Kecamatan Pacet harus menambahkan akses internet agar dapat menjalankan aplikasi tersebut.

4.1.3.4 Kebutuhan Pengguna

Pengguna komputer di UPK Artha Raharja Kecamatan Pacet yang akan mengakses aplikasi yang akan dibangun dibagi menjadi tiga pengguna diantaranya sebagai berikut:

a. Admin

Nama	: Asep Rudi S.Pdi
Bagian	: Sekretariat
Lama menggunakan komputer	: 8 Tahun
Latar belakang pendidikan	: Sarjana Pendidikan Islam
Sistem operasi yang digunakan	: Windows

b. Petugas

Nama	: Enjang Herdiana
Bagian	: Staf umum
Lama menggunakan komputer	: 5 Tahun
Latar belakang pendidikan	: SMA
Sistem operasi yang digunakan	: Windows

c. Ketua

Nama : Cepi Sopandi, S.Sy
 Bagian : Ketua
 Lama menggunakan komputer : 8 Tahun
 Latar belakang pendidikan : Sarjana Syariah
 Sistem operasi yang digunakan : Windows

4.1.3.5 Kebutuhan Sistem

Terdapat tiga proses dalam analisis kebutuhan sistem yang dibutuhkan untuk membangun aplikasi arsip elektronik yaitu analisis kebutuhan proses, analisis kebutuhan masukan dan analisis kebutuhan keluaran. Untuk penjelasan bisa dilihat pada tabel berikut:

Tabel 4.4 Analisis Kebutuhan Sistem

No.	Proses	Masukan	Keluaran
1	<i>Login</i>	<i>Username</i> dan <i>Password</i>	Informasi <i>Login</i>
2	Manajemen Surat Masuk (menambah, melihat, mengubah dan menghapus)	Nomor surat, tanggal surat, asal surat, gambar surat	Informasi surat masuk
3	Manajemen Surat Keluar (menambah, melihat, mengubah dan menghapus)	Nomor surat, tanggal surat, tujuan surat, file atau gambar surat	Informasi surat keluar
4	Manajemen Pencarian (algoritma <i>sequential search</i>)	Kata kunci	Hasil pencarian
5	Manajemen <i>User</i> (melihat, menambah,	Nama, <i>username</i> dan <i>password</i>	Informasi <i>user</i>

	mengubah dan menghapus)		
--	-------------------------	--	--

Terdapat lima proses yang dibutuhkan dalam aplikasi arsip elektronik yaitu proses *login*, manajemen surat masuk, manajemen surat keluar, manajemen pencarian dan manajemen akun. Proses *login* digunakan pengguna untuk masuk ke aplikasi sesuai dengan hak aksesnya. Manajemen surat masuk adalah proses dimana petugas pengolah surat mengolah data surat masuk beserta pengarsipannya. Manajemen surat keluar adalah proses dimana petugas pengolah surat mengolah data surat keluar beserta pengarsipannya. Manajemen laporan adalah proses rekap data arsip surat berdasarkan periode tertentu. Manajemen pencarian adalah proses pencarian data surat di dalam aplikasi dengan menggunakan penerapan algoritma *sequential searching*. Manajemen akun adalah proses petugas mengelola akun beserta hak akses untuk masuk ke aplikasi.

4.1.4 Hasil Analisis

Hasil analisis yang didapatkan pada penelitian ini yaitu dengan komputer yang ada saat ini cukup untuk menjalankan aplikasi yang akan dibangun, untuk kekurangannya di UPK Artha Raharja Kecamatan Pacet belum memenuhi kebutuhan akses internet serta kebutuhan *server hosting* untuk dapat menjalankan aplikasi nantinya. Aplikasi arsip elektronik untuk UPK Artha Raharja Kecamatan Pacet yang akan dibangun dapat melakukan pendataan berkas surat di dalam aplikasi sehingga tidak memerlukan buku agenda surat serta dapat menyimpan berkas surat berupa hasil *scan* dari berkas surat asli. Aplikasi tersebut juga dapat membantu dalam proses pencarian berkas surat yang telah diarsipkan dengan adanya fitur pencarian yang menggunakan algoritma *sequential searching* serta dapat melakukan rekap laporan agenda surat masuk maupun surat keluar secara otomatis berdasarkan periode tertentu. Untuk menunjang kegiatan pengarsipan menggunakan aplikasi arsip elektronik berbasis *web* ini tentunya membutuhkan akses ke internet dan *server hosting*. Dalam proses pembuatan aplikasi penyusun menggunakan *template bootstrap*.

4.2 Perancangan

Pada UPK Artha Raharja Kecamatan Pacet akan dibuat sebuah aplikasi arsip elektronik berbasis *web*. Perancangan aplikasi ini di bangun bertujuan untuk memudahkan proses pengarsipan data dan dokumen surat masuk dan surat keluar dengan cara mencatat dan menyimpan secara digital di dalam aplikasi.

Perancangan aplikasi arsip elektronik yang di bangun menggunakan *Unified Modelling Language* (UML) sebagai bahasa pemodelan. Pembangunan aplikasi arsip elektronik ini dilakukan dengan menggunakan *tools* utama sebagai berikut:

1. PHP sebagai *framework* aplikasi untuk membaca bahasa pemrograman PHP.
2. MySQL sebagai *Database Server*.
3. PHPMyAdmin sebagai *tools* untuk mengelola *Database* berbasis *web*.

4.2.1 Perancangan Sistem

Perancangan yang digunakan dalam proses pembuatan aplikasi ini menggunakan UML (*Unified Modelling Language*), yang merupakan metode pemodelan berorientasi objek. Diagram UML yang digunakan untuk perancangan ini adalah *use case diagram*, *sequence diagram* dan *activity diagram*.

4.2.1.1 Use Case Diagram

Use case diagram merupakan pemodelan untuk mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan aplikasi yang dibuat serta digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah aplikasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

1. Deskripsi Aktor

Deskripsi aktor merupakan penjelasan dari apa yang dilakukan oleh aktor-aktor yang terlibat dalam aplikasi yang akan dibangun.

Adapun deskripsi dari aktor-aktor yang terlibat dalam aplikasi arsip elektronik berbasis *web* UPK Artha Raharja Kecamatan Pacet adalah sebagai berikut:

Tabel 4.5 Deskripsi Aktor

No.	Aktor	Deskripsi
1	Admin	1. Melakukan <i>login</i> . 2. Mengelola (tambah, ubah, hapus) data surat masuk maupun keluar. 3. Mengelola akun
2	Ketua	1. Melakukan <i>login</i> 2. Melihat data surat masuk maupun keluar.
3	Petugas	1. Melakukan <i>login</i> 2. Mengelola (tambah, ubah, hapus) data surat masuk maupun keluar.

2. Deskripsi Use Case

Deskripsi *use case* merupakan penjelasan dari *use case* atau proses-proses yang berlangsung di dalam kegiatan aplikasi. Adapun deskripsi dari *use case* aplikasi arsip elektronik berbasis *web* UPK Artha Raharja Kecamatan Pacet adalah sebagai berikut:

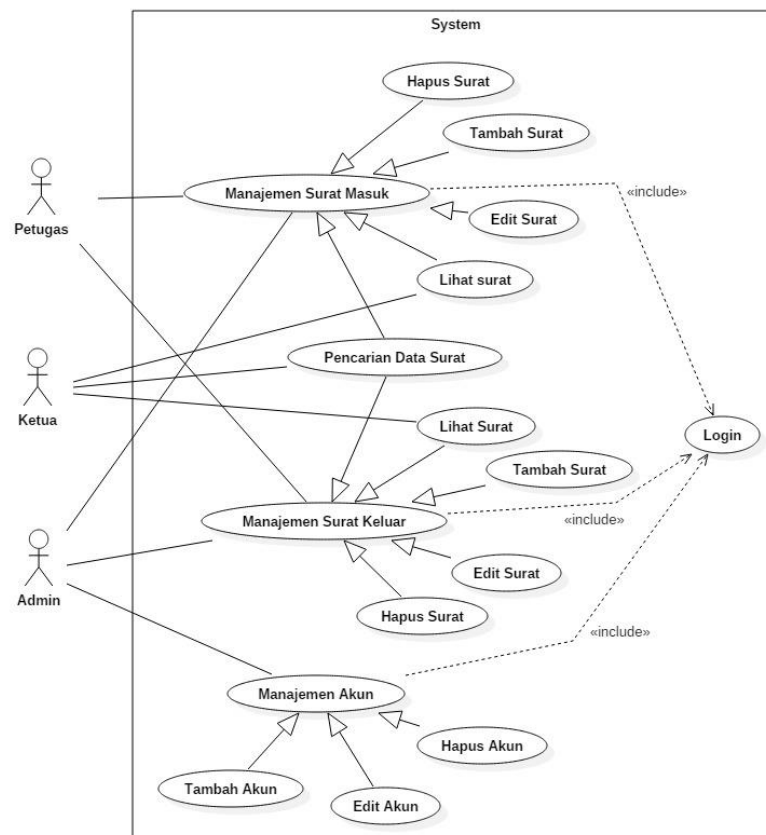
Tabel 4.6 Deskripsi Use Case

No.	Use Case	Deskripsi
1	<i>Login</i>	Aktor harus melakukan <i>login</i> menggunakan <i>Username</i> dan <i>Password</i> .
2	Manajemen <i>User</i>	Proses pengelolaan <i>user</i>

3	Manajemen Surat Masuk	Proses pengelolaan agenda surat masuk yaitu menambah, mengubah dan menghapus data surat masuk.
4	Manajemen Surat Keluar	Proses pengelolaan agenda surat keluar yaitu menambah, mengubah dan menghapus data surat keluar.
5	Pencarian Data	Proses pencarian data surat masuk dan surat keluar menggunakan algoritma <i>sequential searching</i> .

3. Use Case Aplikasi E-Arsip UPK Artha Raharja

Use case diagram aplikasi arsip elektronik UPK Artha Raharja Kecamatan Pacet adalah sebagai berikut:



Gambar 4.3 Use Case Diagram Aplikasi Arsip Elektronik

4. Skenario *Use Case Diagram*

Skenario yang menggambarkan untuk masing-masing *use case* dari aplikasi arsip elektronik UPK Artha Raharja Kecamatan Pacet adalah sebagai berikut:

a. Skenario *Use Case Login*

Berikut skenario *use case* yang menggambarkan proses *login* pada aplikasi:

Tabel 4.7 Skenario *Use Case Login*

Nama	<i>Login Admin</i>	
Level	2	
Aktor	Admin	
Tujuan	Masuk ke dalam aplikasi, mengelola semua manajemen didalam aplikasi	
Precondition	Memasukkan <i>Username</i> dan <i>Password</i>	
Postcondition	Berhasil masuk ke halaman awal aplikasi	
Langkah	Aktor	Sistem
	1. Memasukkan <i>Username</i> dan <i>Password</i>	
		2. Melakukan <i>Validasi Username</i> dan <i>Password</i>
		3. Menampilkan halaman awal aplikasi

b. Skenario *Use Case Menambah Data Surat Masuk*

Berikut skenario yang menggambarkan proses manajemen surat masuk untuk menambahkan data surat masuk:

Tabel 4.8 Skenario Menambah Data Surat Masuk

Nama	Menambah data surat masuk	
Level	1	
Aktor	Petugas	
Tujuan	Menambah data surat masuk pada aplikasi	
Precondition	Masuk pada halaman Surat Masuk	
Postcondition	Berhasil menambah data surat masuk	
Langkah	Aktor	Sistem
	1. Petugas masuk ke halaman surat masuk	
		2. Menampilkan menu halaman surat masuk
	3. Pilih tambah surat masuk	
		4. Menampilkan halaman tambah surat
	5. Memasukkan data nomor surat	
	6. Memasukkan data tanggal masuk surat	
	7. Memasukkan data pengirim	
	8. Menambahkan file surat berupa gambar hasil <i>scan</i>	
	9. Mengklik tombol tambah	
		10. Menerbitkan data surat masuk

c. Skenario *Use Case* Mengubah Data Surat Masuk

Berikut skenario yang menggambarkan proses manajemen surat masuk untuk mengubah data surat masuk yang sudah ada di dalam aplikasi:

Tabel 4.9 Skenario Mengubah Data Surat Masuk

Nama	Mengubah data surat masuk	
Level	1	
Aktor	Petugas	
Tujuan	Mengubah data surat masuk pada aplikasi	
Precondition	Masuk pada halaman Surat Masuk	
Postcondition	Berhasil mengubah data surat masuk	
Langkah	Aktor	Sistem
	1. Petugas masuk ke halaman surat masuk	
		2. Menampilkan menu halaman surat masuk
	3. Pilih edit surat masuk	
		4. Menampilkan halaman edit surat
	5. Memasukkan data identitas surat baru	
	6. Mengklik tombol edit	
		7. Menerbitkan data surat masuk

d. Skenario *Use Case* Menghapus Data Surat Masuk

Berikut skenario yang menggambarkan proses menghapus data surat masuk:

Tabel 4.10 Skenario Menghapus Data Surat Masuk

Nama	Menghapus data surat masuk	
Level	1	
Aktor	Petugas	
Tujuan	Menghapus data surat masuk pada aplikasi	
Precondition	Masuk pada halaman Surat Masuk	
Postcondition	Berhasil menghapus data surat masuk	
Langkah	Aktor	Sistem
	1. Petugas masuk ke halaman surat masuk	
		2. Menampilkan menu halaman surat masuk
	3. Pilih hapus surat masuk	
		4. Menampilkan konfirmasi hapus surat
	5. Mengklik oke untuk lanjut hapus	
		6. Menghapus data surat masuk terpilih

e. Skenario *Use Case* Melihat Data Surat Masuk

Berikut skenario yang menggambarkan proses manajemen data surat masuk untuk melihat file surat berupa hasil *scan* yang sudah ada pada aplikasi yang terletak di tabel data surat masuk untuk *user* ketua:

Tabel 4.11 Skenario Melihat Data Surat Masuk

Nama	Melihat data surat masuk	
Level	0	
Aktor	Ketua	
Tujuan	Melihat data surat masuk pada aplikasi	
Precondition	Masuk pada halaman Surat Masuk	
Postcondition	Berhasil masuk halaman data surat masuk	
Langkah	Aktor	Sistem
	1. Masuk ke halaman surat masuk	
		2. Menampilkan menu halaman surat masuk
	3. Pilih lihat surat masuk	
		4. Menampilkan Data surat masuk

f. Skenario *Use Case* Mencari Data Surat Masuk

Berikut skenario yang menggambarkan proses mencari data surat masuk untuk menemukan data dan dokumen surat masuk yang sudah di *input* kedalam aplikasi:

Tabel 4.12 Skenario Mencari Data Surat Masuk

Nama	Mencari data surat masuk	
Level	0	
Aktor	Ketua	
Tujuan	Mencari data surat masuk pada aplikasi	
Precondition	Masuk pada halaman Surat Masuk	
Postcondition	Berhasil menemukan data surat masuk yang dicari	
Langkah	Aktor	Sistem

	1. Masuk ke halaman surat masuk	
		2. Menampilkan menu halaman surat masuk
	3. Memasukkan kata kunci surat yang dicari	
		4. Menampilkan Data surat masuk yang dicari

g. Skenario *Use Case* Menambah *User*

Berikut skenario yang menggambarkan proses manajemen *user* untuk menambahkan *user* baru:

Tabel 4.13 Skenario Menambah *User*

Nama	Menambah <i>user</i>	
Level	2	
Aktor	Admin	
Tujuan	Menambah <i>user</i>	
Precondition	Masuk pada halaman Akun	
Postcondition	Berhasil menambahkan akun baru	
Langkah	Aktor	Sistem
	1. Admin masuk ke halaman akun	
		2. Menampilkan menu halaman akun
	3. Memasukkan data nama	
	4. Memasukkan data <i>username</i>	

	5. Memasukkan data jabatan	
	6. Memilih hak akses	
	7. Mengklik tambah	
		8. Menerbitkan Data akun baru

h. Skenario *Use Case* Mengubah *User*

Berikut skenario yang menggambarkan proses manajemen *user* untuk mengubah data *user* yang sudah terdaftar di dalam aplikasi:

Tabel 4.14 Skenario Mengubah *User*

Nama	Mengubah <i>user</i>	
Level	2	
Aktor	Admin	
Tujuan	Mengubah data <i>user</i>	
Precondition	Masuk pada halaman Akun	
Postcondition	Berhasil mengubah data akun	
Langkah	Aktor	Sistem
	1. Admin masuk ke halaman akun	
		2. Menampilkan menu halaman akun
	3. Memasukkan data identitas akun yang akan diubah	
	4. Mengklik edit	
		5. Menerbitkan Data identitas akun baru

i. Skenario *Use Case* Menghapus *User*

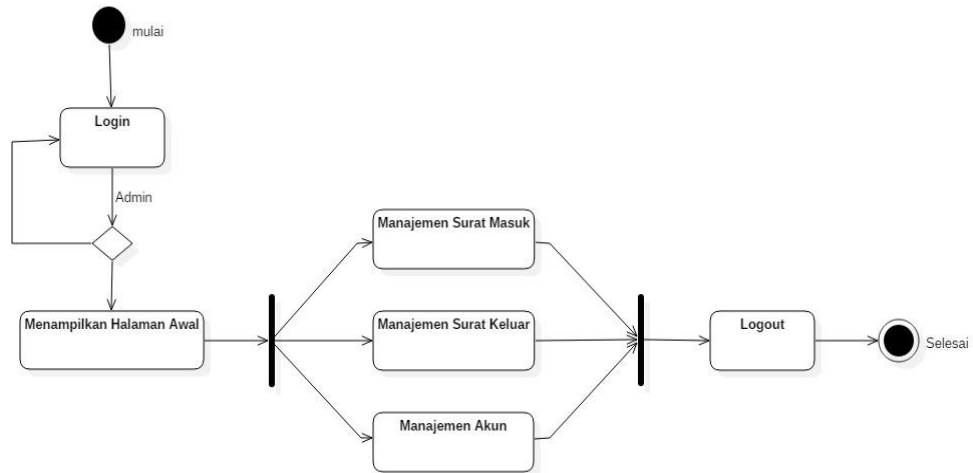
Berikut skenario yang menggambarkan proses menghapus data *user*:

Tabel 4.15 Skenario Menghapus *User*

Nama	Menghapus <i>user</i>	
Level	2	
Aktor	Admin	
Tujuan	Menghapus <i>user</i>	
Precondition	Masuk pada halaman Akun	
Postcondition	Berhasil menghapus data akun	
Langkah	Aktor	Sistem
	1. Admin masuk ke halaman akun	
		2. Menampilkan menu halaman akun
	3. Memilih hapus pada pilihan akun yang akan dihapus	
		4. Menampilkan konfirmasi hapus akun
	6. Mengklik oke untuk melanjutkan hapus	
		7. Menghapus akun yang dipilih

4.2.1.2 Activity Diagram

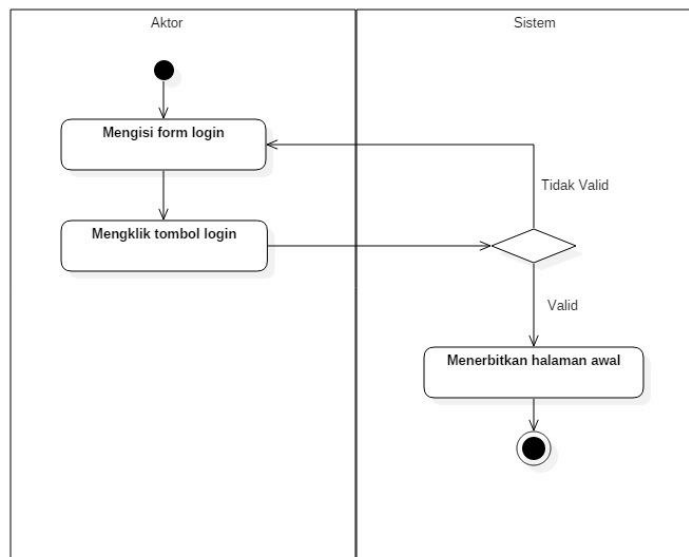
Activity diagram adalah salah satu cara memodelkan *event-event* yang terjadi dalam suatu *use case*. Berikut adalah *activity diagram* dari aplikasi arsip elektronik UPK Artha Raharja Kecamatan Pacet:



Gambar 4.4 Activity Diagram Aplikasi Arsip Elektronik

1. Activity Diagram Login

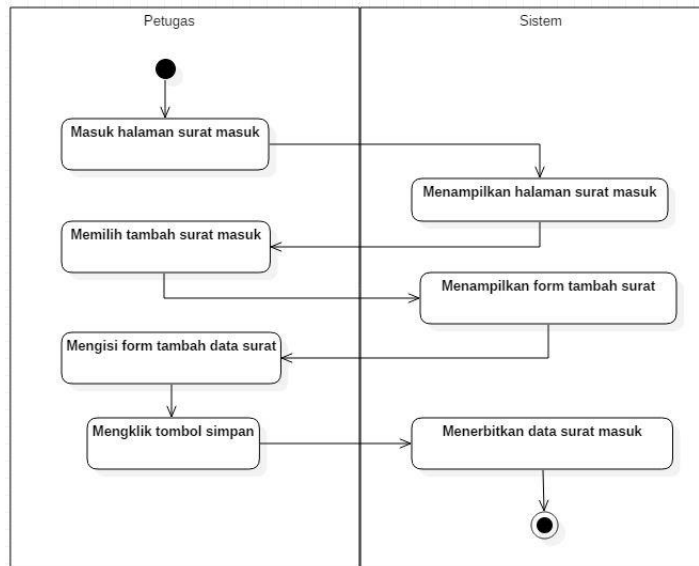
Activity Diagram berikut menggambarkan proses *login*:



Gambar 4.5 Activity Diagram Login

2. Activity Diagram menambah Surat Masuk

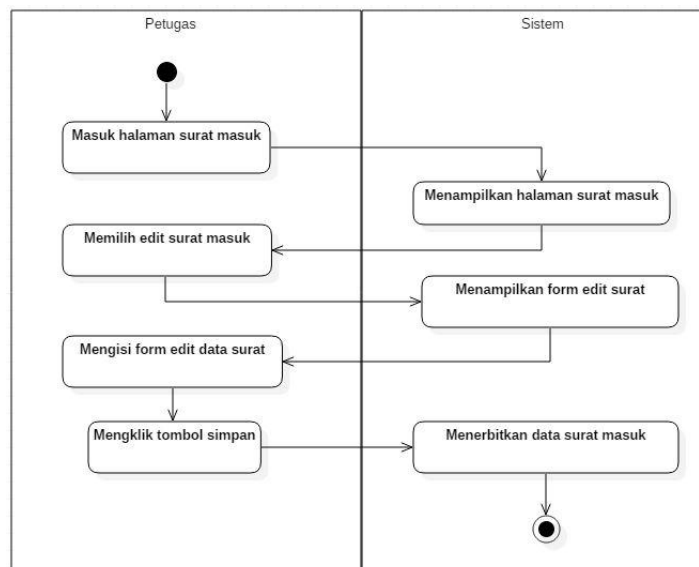
Activity Diagram berikut menggambarkan proses menambah data surat masuk:



Gambar 4.6 Activity Diagram Tambah Surat Masuk

3. Activity Diagram mengubah Surat Masuk

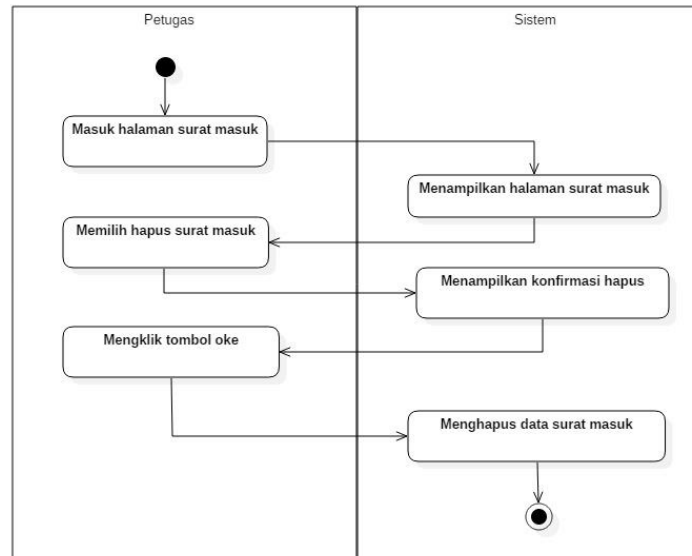
Activity Diagram berikut menggambarkan proses mengubah data surat masuk:



Gambar 4.7 Activity Diagram Edit Surat Masuk

4. *Activity Diagram* menghapus Surat Masuk

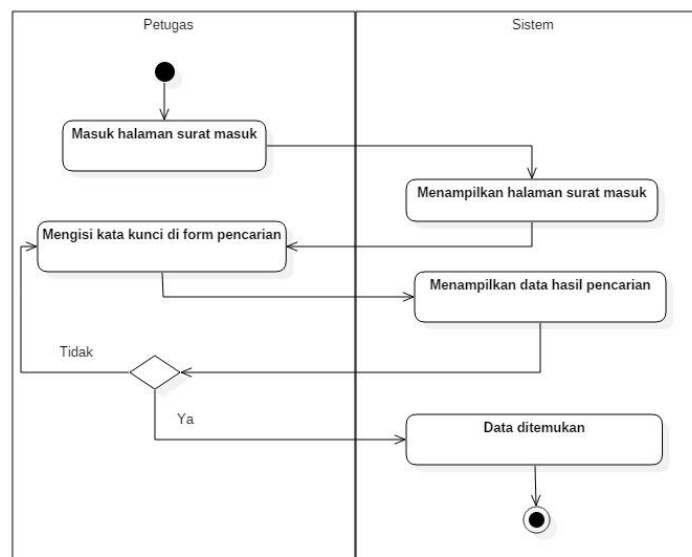
Activity Diagram berikut menggambarkan proses menghapus data surat masuk:



Gambar 4.8 *Activity Diagram* Hapus Surat Masuk

5. *Activity Diagram* mencari Data Surat Masuk

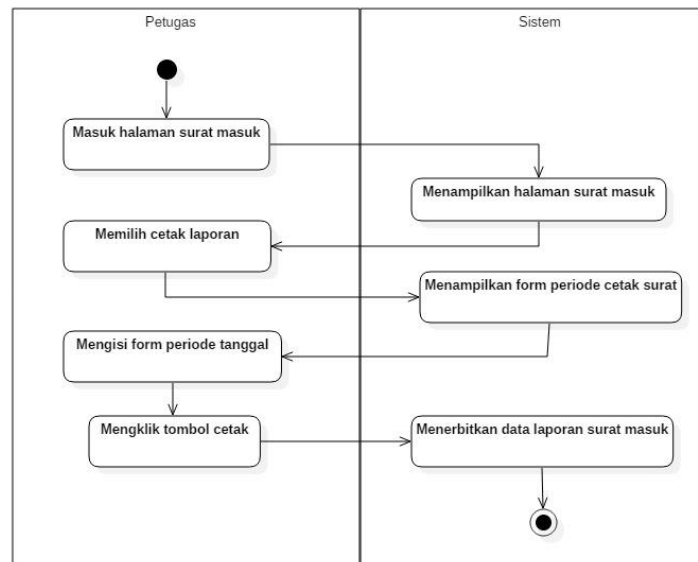
Activity Diagram berikut menggambarkan proses mencari data surat masuk:



Gambar 4.9 *Activity Diagram* Cari Surat Masuk

6. Activity Diagram Cetak Laporan Surat Masuk

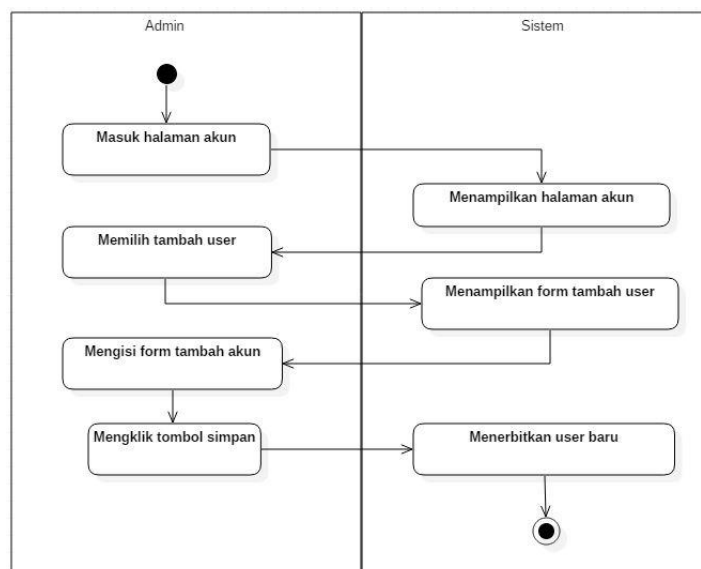
Activity Diagram berikut menggambarkan proses mencetak laporan data surat masuk:



Gambar 4.10 Activity Diagram Cetak Laporan Surat Masuk

7. Activity Diagram menambah User

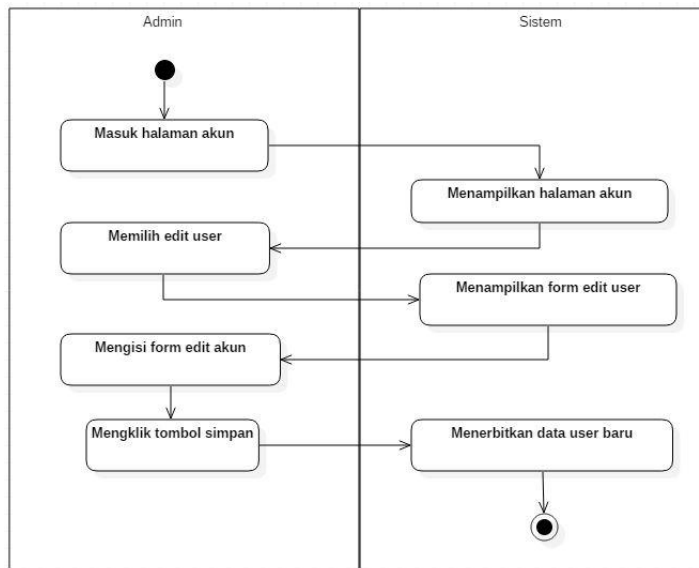
Activity Diagram berikut menggambarkan proses menambah user:



Gambar 4.11 Activity Diagram Tambah User

8. Activity Diagram mengubah User

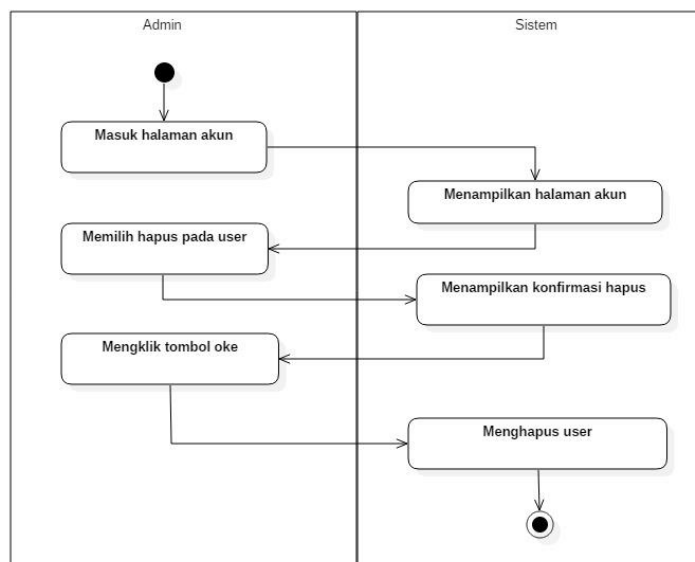
Activity Diagram berikut menggambarkan proses mengubah data user:



Gambar 4.12 Activity Diagram Edit User

9. Activity Diagram menghapus User

Activity Diagram berikut menggambarkan proses menghapus data user:



Gambar 4.13 Activity Diagram Hapus User

4.2.2 Perancangan Basis Data

4.2.2.1 Perancangan Tabel

Berikut dijelaskan tabel-tabel yang digunakan dalam perancangan tabel dengan deskripsi sebagai berikut:

1. Tabel Surat Masuk

Nama Tabel : surat_masuk

Keterangan : Menampung data surat masuk

Tabel 4.16 Tabel Surat Masuk

No.	Nama <i>Field</i>	Tipe Data	Panjang	Keterangan
1	Id	int	4	Nomor identitas untuk pengurutan secara otomatis
2	Nomor_surat	Varchar	20	Nomor surat
3	Tanggal_masuk	Date	-	Tanggal masuk surat
4	Pengirim	Varchar	30	Pengirim surat
5	Perihal	Varchar	20	Perihal Surat
6	File_surat	Blob	-	File surat

2. Tabel Surat Keluar

Nama Tabel : surat_keluar

Keterangan : Menampung data surat keluar

Tabel 4.17 Tabel Surat Keluar

No.	Nama <i>Field</i>	Tipe Data	Panjang	Keterangan
1	Id	int	4	Nomor identitas untuk pengurutan secara otomatis

2	Nomor_surat	Varchar	20	Nomor surat
3	Tanggal_keluar	Date	-	Tanggal terbit surat
4	Penerima	Varchar	30	Tujuan surat
5	Perihal	Varchar	20	Perihal Surat
6	File_surat	Blob	-	File surat

3. Tabel Akun

Nama Tabel : akun

Keterangan : Menampung data *user*

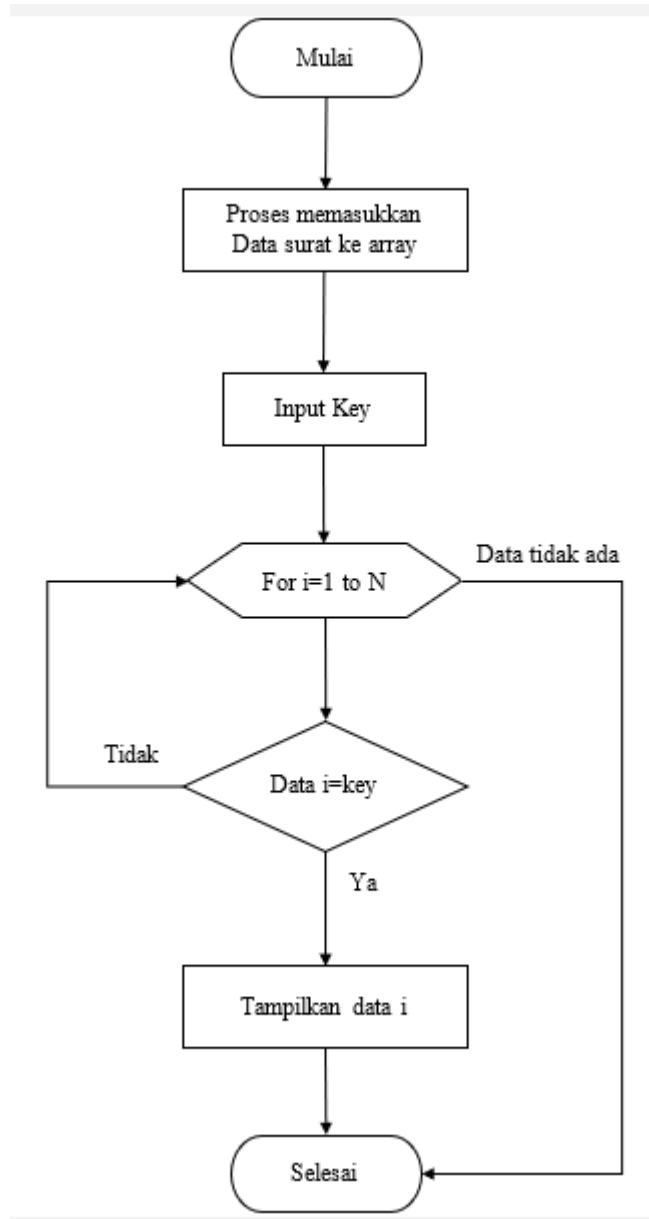
Tabel 4.18 Tabel Akun

No.	Nama <i>Field</i>	Tipe Data	Panjang	Keterangan
1	Id	int	4	Nomor identitas untuk pengurutan secara otomatis
2	Nama	Varchar	30	Nama pengguna
3	User_name	Varchar	10	<i>Username</i> akun
4	Password	Varchar	256	Password akun
5	Jabatan	Varchar	15	Keterangan Jabatan
6	akses	int	1	Hak akses akun

4.2.3 Perancangan Algoritma *Sequential Searching*

Pada pembuatan aplikasi arsip elektronik berbasis *web* di UPK Artha Raharja terdapat fitur pencarian data surat untuk memudahkan proses pencarian data surat tertentu. Pada fitur pencarian penyusun mengimplementasikan algoritma *sequential searching*, dimana cara kerja algoritma *sequential searching* yaitu mencari data berdasarkan *key* yang di *input* dengan cara membandingkan data satu-persatu secara beruntun dari data pertama sampai data yang dicari ditemukan atau seluruh data sudah dibandingkan. *Metode sequential searching* atau pencarian beruntun dapat digunakan untuk melakukan pencarian

data baik pada *array* yang sudah terurut maupun belum terurut. Untuk lebih jelasnya bisa dilihat pada gambar berikut:



Gambar 4.15 *Flowchart Algoritma Sequential Searching*

Penjelasan dari gambar 4.15 tentang alur kerja algoritma *sequential searching* adalah sebagai berikut:

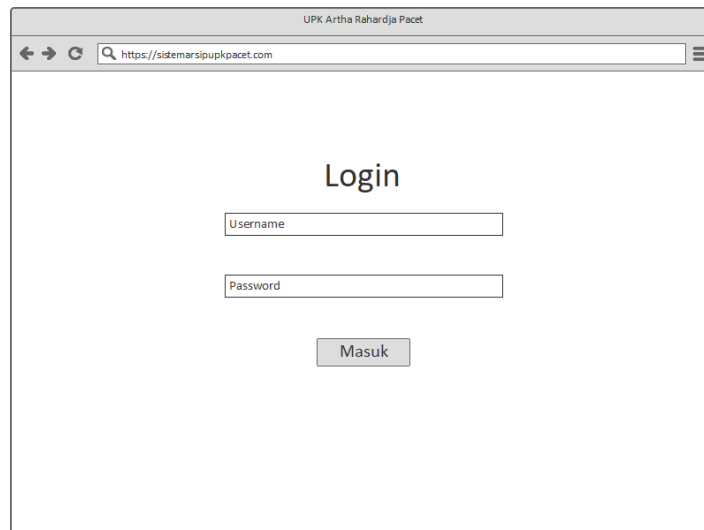
1. Mulai.
2. Pengguna memasukkan *key* data surat yang akan dicari.
3. Sistem akan memasukkan semua data yang terdapat di database surat masuk atau surat keluar kedalam array.

4. Sistem akan melakukan perulangan berdasarkan jumlah array (for $i=1$ to N).
5. Jika data yang dicari ditemukan (Data $i=key$), maka perulangan dan pencarian akan berhenti dan menampilkan data yang dicari.
6. Jika data yang dicari tidak ditemukan sampai dengan N kali, maka proses perulangan dan pencarian akan berhenti dan data tidak ditemukan atau data tidak ada.
7. Selesai

4.2.4 Perancangan *User Interface*

User interface merupakan tampilan yang dibuat agar memudahkan komunikasi antara pengguna dengan sistem aplikasi. Perancangan *user interface* sendiri dibuat sesederhana mungkin untuk memudahkan pengguna dalam mengakses aplikasi sehingga dapat mengurangi tingkat kebingungan pengguna dalam menggunakan aplikasi.

4.2.4.1 Tampilan Halaman *Login*

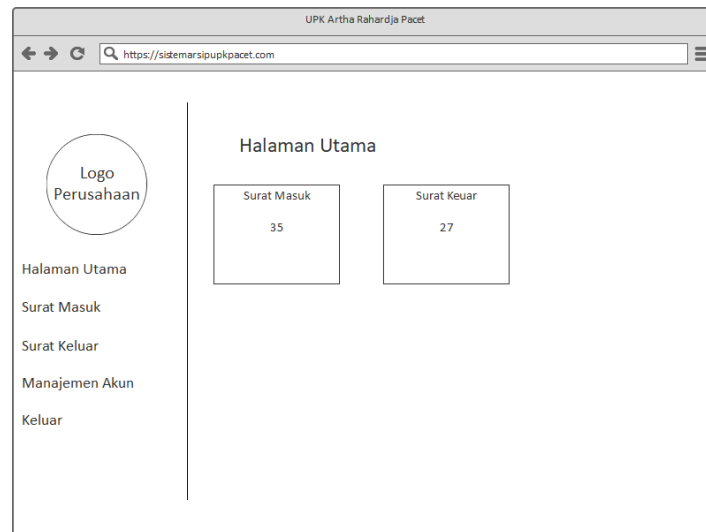


The image shows a web browser window with the address bar displaying 'https://sidemarsipupkpacet.com'. The page title is 'UPK Artha Rahardja Pacet'. The main content area displays a login form with the title 'Login'. Below the title, there are two input fields: 'Username' and 'Password'. At the bottom of the form is a button labeled 'Masuk'.

Gambar 4.16 Rancangan Halaman *Login*

Halaman *login* berisi *form* memasukkan akun pengguna berupa *username* dan *password* untuk mengakses aplikasi sesuai dengan hak akses masing-masing.

4.2.4.2 Tampilan Halaman Utama

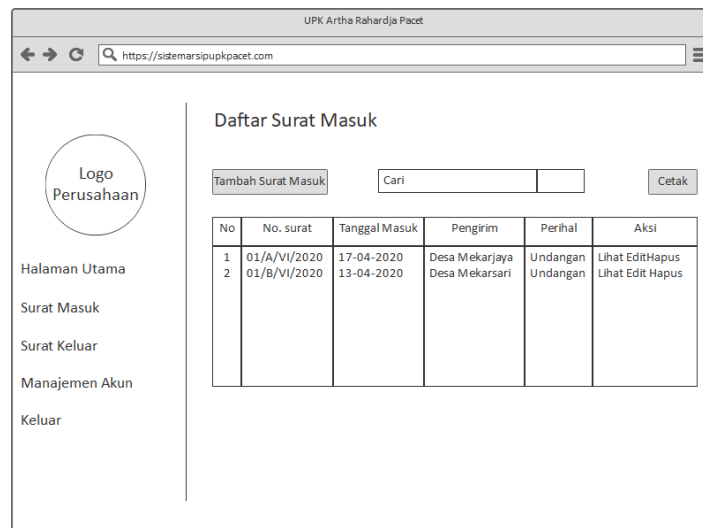


Gambar 4.17 Rancangan Halaman Utama

Halaman utama berisi menu-menu untuk mengakses seluruh fasilitas dari aplikasi, seperti menu surat masuk, surat keluar, manajemen akun untuk *user* admin serta tampilan jumlah dokumen surat masuk maupun surat keluar secara keseluruhan.

4.2.4.3 Halaman Data Surat

a. Data Surat Masuk



Gambar 4.18 Rancangan Halaman Data Surat Masuk

Halaman surat masuk berisi tabel daftar surat masuk yang berada di dalam sistem beserta aksi untuk data surat masuk seperti tambah, cari, edit dan hapus surat.

b. Data Surat Keluar

No	No. surat	Tanggal Keluar	Penerima	Perihal	Aksi
1	01/A/VI/2020	17-04-2020	Desa Mekarjaya	Undangan	Lihat EditHapus
2	01/B/VI/2020	13-04-2020	Desa Mekarsari	Undangan	Lihat Edit Hapus

Gambar 4.19 Rancangan Halaman Data Surat Keluar

Halaman surat keluar sama seperti halaman surat masuk berisi tabel daftar surat keluar yang berada di dalam sistem beserta aksi untuk data surat keluar seperti tambah, cari, edit dan hapus surat.

4.2.4.4 Halaman Tambah Surat

a. Tambah Surat Masuk

Gambar 4.20 Rancangan Halaman Tambah Surat Masuk

Halaman tambah surat masuk berisi tentang input data surat masuk diantaranya nomor surat, tanggal masuk surat, pengirim surat serta lampiran surat asli berupa file hasil *scan*.

b. Tambah Surat Keluar

Gambar 4.21 Rancangan Halaman Tambah Surat Keluar

Halaman tambah surat keluar berisi tentang input data surat keluar diantaranya nomor surat, tanggal terbit surat, penerima surat serta lampiran.

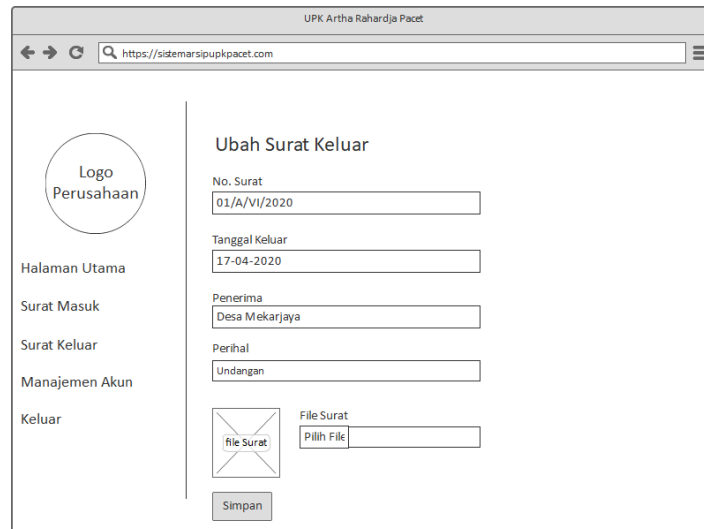
4.2.4.5 Halaman Edit Data Surat

a. Edit Surat Masuk

Gambar 4.22 Rancangan Halaman Edit Surat Masuk

Halaman edit surat masuk berisi *form* data surat masuk yang akan diubah diantaranya nomor surat, tanggal masuk surat, pengirim surat serta lampiran surat asli berupa file hasil *scan* jika ingin digantikan.

b. Edit Surat Keluar



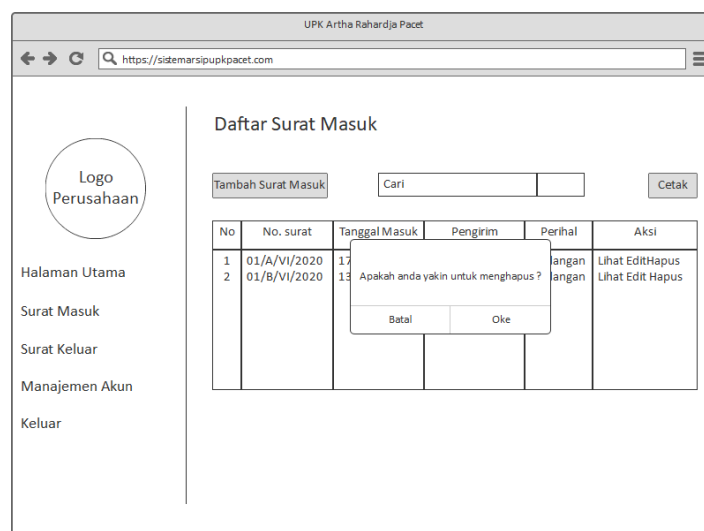
The image shows a web browser window with the URL <https://sidemarsipupkpacet.com>. The page title is "UPK Artha Rahardja Pacet". On the left, there is a sidebar with a "Logo Perusahaan" and a menu with items: "Halaman Utama", "Surat Masuk", "Surat Keluar", "Manajemen Akun", and "Keluar". The main content area is titled "Ubah Surat Keluar". It contains several input fields: "No. Surat" (01/A/VI/2020), "Tanggal Keluar" (17-04-2020), "Penerima" (Desa Mekarjaya), and "Perihal" (Undangan). There is a "File Surat" section with a "Pilih File" button. At the bottom, there is a "Simpan" button.

Gambar 4.23 Rancangan Halaman Edit Surat Keluar

Halaman edit surat keluar sama seperti halaman edit surat masuk yang berisi *form* data surat keluar yang akan diubah.

4.2.4.6 Halaman Hapus Surat

a. Hapus Surat Masuk

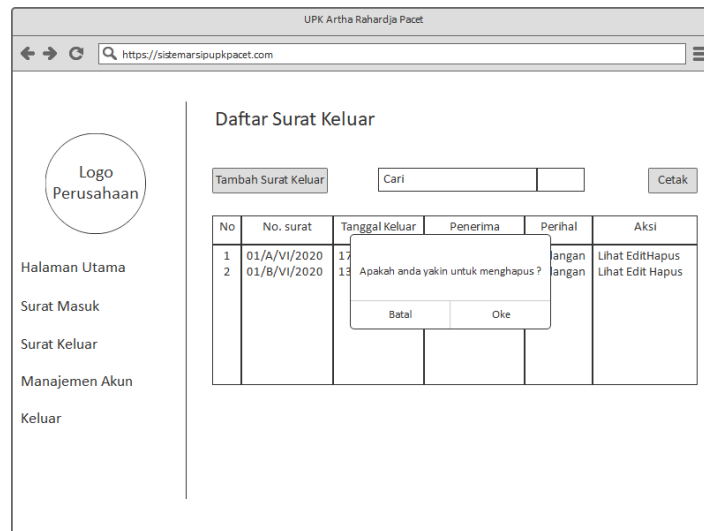


The image shows a web browser window with the URL <https://sidemarsipupkpacet.com>. The page title is "UPK Artha Rahardja Pacet". On the left, there is a sidebar with a "Logo Perusahaan" and a menu with items: "Halaman Utama", "Surat Masuk", "Surat Keluar", "Manajemen Akun", and "Keluar". The main content area is titled "Daftar Surat Masuk". It contains a "Tambah Surat Masuk" button, a "Cari" input field, and a "Cetak" button. Below these is a table with columns: "No", "No. surat", "Tanggal Masuk", "Pengirim", "Perihal", and "Aksi". The table has two rows of data. The first row has "1", "01/A/VI/2020", "17", "Desa Mekarjaya", "Undangan", and "Lihat EditHapus". The second row has "2", "01/B/VI/2020", "13", "Desa Mekarjaya", "Undangan", and "Lihat EditHapus". A modal dialog box is open over the second row, asking "Apakah anda yakin untuk menghapus?" with "Batal" and "Oke" buttons.

Gambar 4.24 Rancangan Halaman Hapus Surat Masuk

Halaman hapus surat masuk berupa pemberitahuan untuk melanjutkan menghapus dokumen surat masuk tertentu atau batal menghapus surat.

b. Hapus Surat Keluar

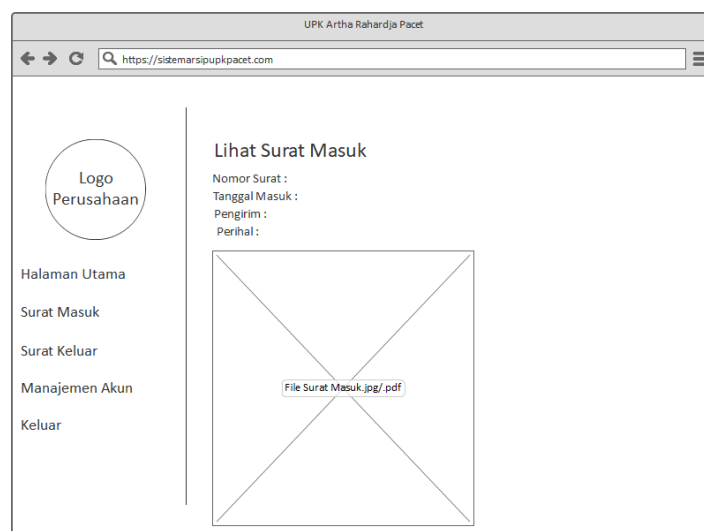


Gambar 4.25 Rancangan Halaman Hapus Surat Keluar

Halaman hapus surat keluar sama seperti halaman hapus surat masuk yang berupa pemberitahuan untuk melanjutkan menghapus dokumen surat keluar tertentu atau batal menghapus surat.

4.2.4.7 Halaman Lihat File Surat

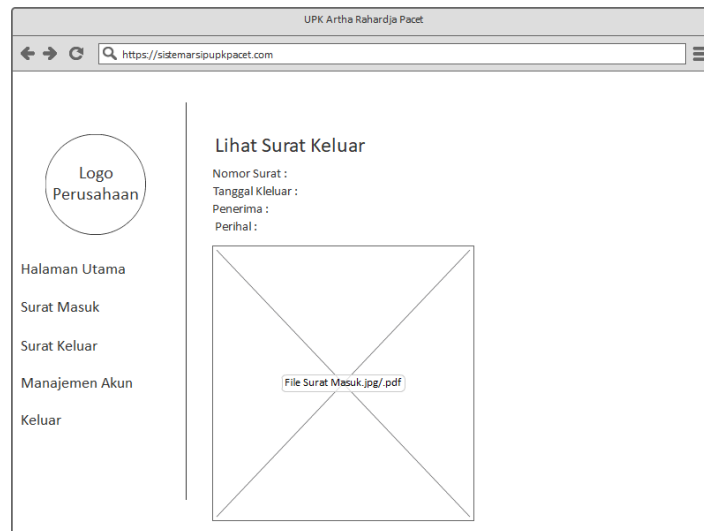
a. Lihat File Surat Masuk



Gambar 4.26 Rancangan Halaman Lihat Surat Masuk

Halaman lihat file surat masuk berisi gambar lampiran berkas surat beserta identitas surat.

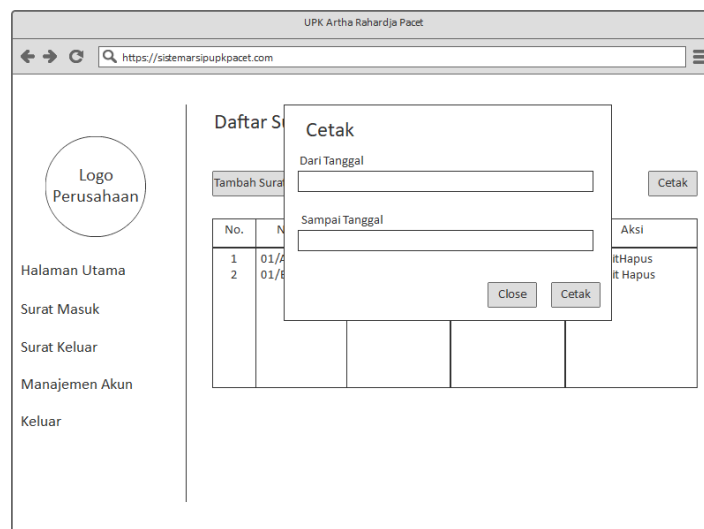
b. Lihat File Surat Keluar



Gambar 4.27 Rancangan Halaman Lihat Surat Keluar

Halaman lihat file surat keluar berisi gambar lampiran berkas surat beserta identitas surat.

4.2.4.8 Halaman Cetak Laporan



Gambar 4.28 Rancangan Halaman Cetak Laporan

Halaman cetak laporan berupa *form* cetak laporan surat berdasar periode tertentu dengan cara input dari tanggal dan sampai tanggal yang ditentukan.

4.2.4.9 Halaman Manajemen Akun

No	Nama	Username	Hak Akses	Aksi
1	Admin	Admin	2	Edit Hapus
2	Petugas	Petugas	1	Edit Hapus

Gambar 4.29 Rancangan Halaman Manajemen Akun

Halaman manajemen akun berisi data akun yang terdaftar diantaranya dan aksi dari data tersebut berupa edit dan hapus serta tambah akun.

4.2.4.10 Halaman Tambah Akun

Gambar 4.30 Rancangan Halaman Tambah Akun

Halaman tambah akun berisi *form* tambah data akun diantaranya nama pengguna, *username*, jabatan, *password*, serta pilihan hak akses. Untuk manajemen akun hanya bisa diakses oleh admin.

4.2.4.11 Halaman Edit Akun

UPK Artha Rahardja Pacet

← → ↻ 🔍 https://sidemarsipupkpacet.com

Logo Perusahaan

Halaman Utama

Surat Masuk

Surat Keluar

Manajemen Akun

Keluar

Edit User

Nama Pengguna

Username

Jabatan

Akses

Pilih Hak Akses

Simpan

Gambar 4.31 Rancangan Halaman Edit Akun

Halaman edit akun sama seperti halaman tambah akun yang berisi data akun yang akan diubah diantaranya nama pengguna, *username*, jabatan, *password*, serta pilihan hak akses.

BAB V

IMPLEMENTASI DAN PENGUJIAN

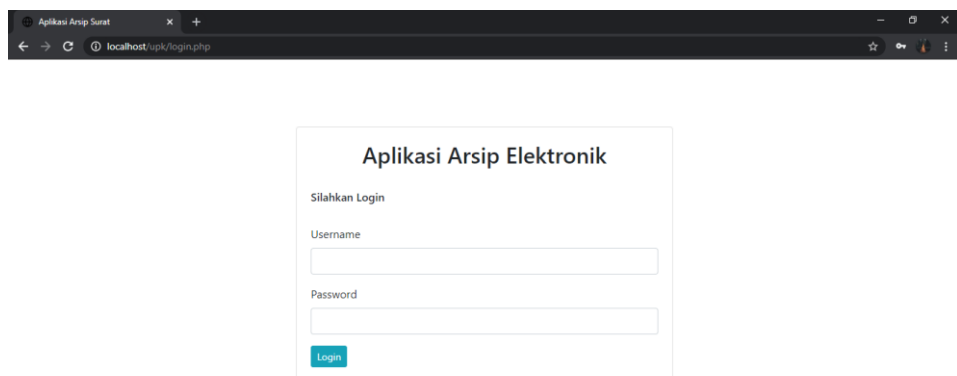
5.1 Implementasi

Implementasi merupakan tahap saat pembuatan aplikasi dimulai setelah dilakukan analisis dan desain. Rancangan sistem yang telah disiapkan kemudian diimplementasikan dalam bahasa pemrograman, sehingga semua fungsi dapat dijalankan dengan baik sesuai dengan yang diharapkan.

5.1.1 Implementasi *User Interface*

Implementasi *user interface* atau antarmuka dilakukan menggunakan bahasa pemrograman PHP dan bootstrap. *User interface* dibuat sesuai dengan rancangan yang sebelumnya telah dibuat.

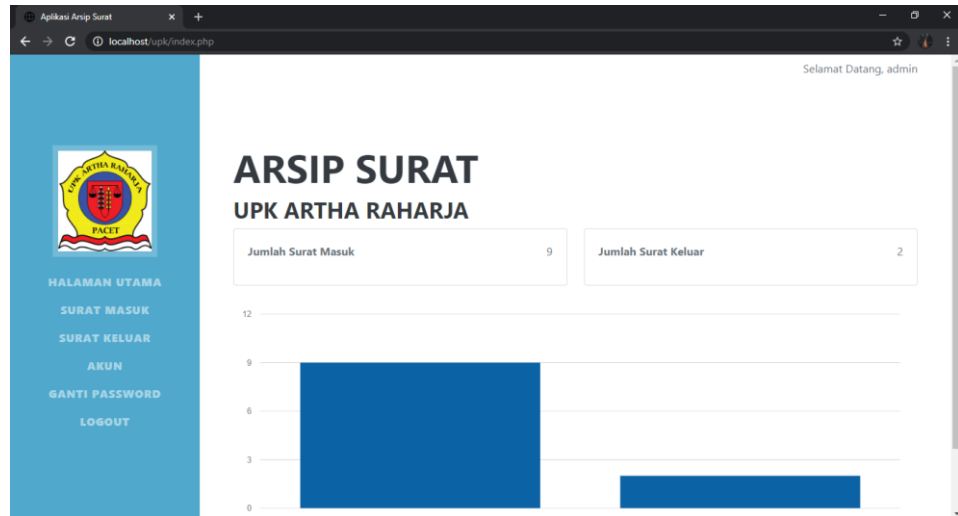
1. Halaman *Login*

The image shows a web browser window with the title 'Aplikasi Arsip Elektronik'. The address bar shows 'localhost:8080/login.php'. The page content is a login form with the heading 'Aplikasi Arsip Elektronik' and the instruction 'Silahkan Login'. Below this, there are two input fields: 'Username' and 'Password'. At the bottom of the form is a blue 'Login' button.

Gambar 5.1 Halaman *Login* Aplikasi Arsip

Pada halaman *login*, pengguna akan diminta memasukkan *username* dan *password* untuk melanjutkan masuk ke dalam aplikasi dan mengelola semua fitur didalam aplikasi sesuai hak akses yang telah dibuat sebelumnya.

2. Halaman Utama



Gambar 5.2 Halaman Awal Aplikasi Arsip

Halaman Utama atau Halaman awal menampilkan grafik jumlah keseluruhan surat masuk dan surat keluar.

3. Halaman Surat Masuk

No	No. Surat	Tanggal Masuk	Pengirim	Perihal	Aksi
1	3	2012-01-03	BKAD	Undangan	Lihat File Edit Hapus
2	4	2012-01-09	UPK Artha Raharja	Undangan	Lihat File Edit Hapus
3	5	2012-01-10	Tim Fasilitator PNPIM-MP	Undangan	Lihat File Edit Hapus

Gambar 5.3 Halaman Surat Masuk Aplikasi Arsip

Halaman surat masuk berisi tabel data surat surat masuk yang sudah tercatat diantaranya nomor surat, tanggal masuk, perihal.

4. Halaman Tambah Surat Masuk

Gambar 5.4 Halaman Tambah Surat Masuk Aplikasi Arsip

Halaman Tambah surat masuk berisi *form* untuk menambahkan data surat yang akan dicatat dan diarsipkan beserta lampiran file surat berupa gambar hasil *scan*.

5. Halaman Edit Surat Masuk

Gambar 5.5 Halaman Edit Surat Masuk Aplikasi Arsip

Halaman edit surat masuk berisi *form* untuk mengubah data surat masuk yang sudah tercatat di aplikasi.

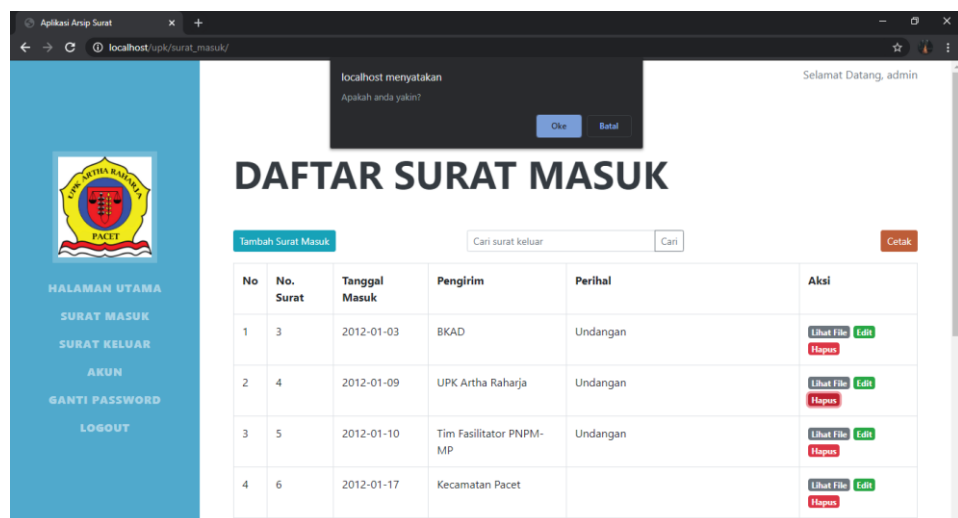
6. Halaman Lihat File Surat Masuk



Gambar 5.6 Halaman Lihat File Surat Masuk Aplikasi Arsip

Halaman lihat file surat masuk berisi rincian surat beserta gambar file surat hasil *scan*.

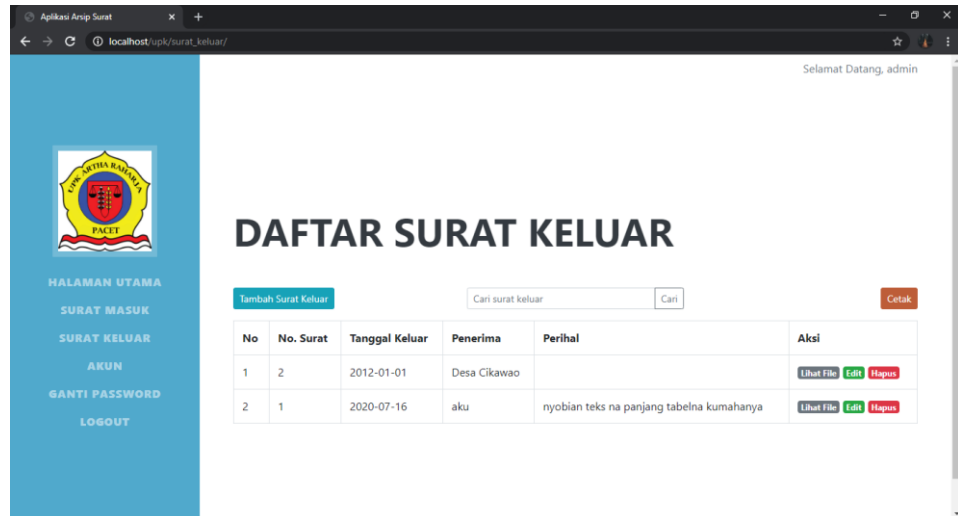
7. Halaman Hapus Surat Masuk



Gambar 5.7 Halaman Hapus Surat Masuk Aplikasi Arsip

Halaman hapus surat masuk berisi sebuah pemberitahuan untuk melanjutkan hapus data surat masuk atau membatalkannya.

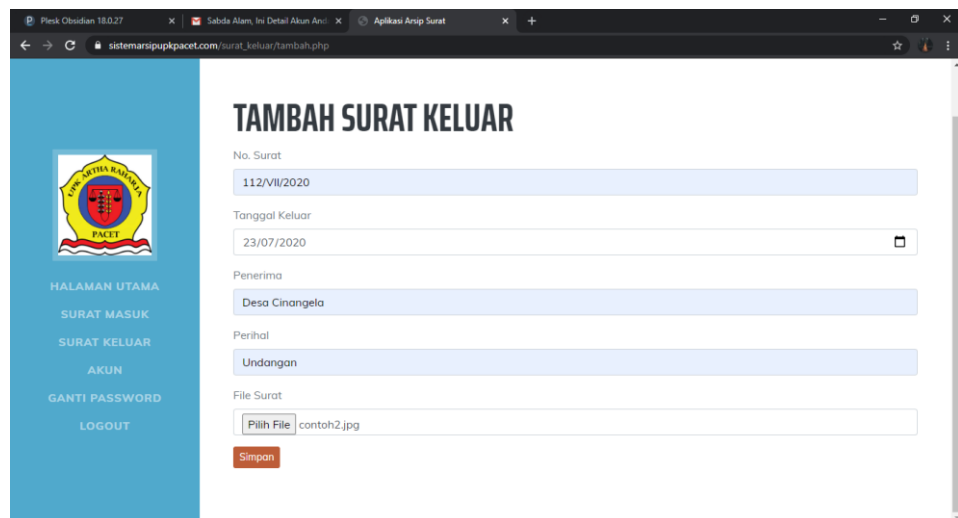
8. Halaman Surat Keluar



Gambar 5.8 Halaman Surat Keluar Aplikasi Arsip

Halaman surat keluar berisi tabel data surat surat keluar yang sudah tercatat diantaranya nomor surat, tanggal terbit, perihal.

9. Halaman Tambah Surat Keluar



Gambar 5.9 Halaman Tambah Surat Keluar Aplikasi Arsip

Halaman Tambah surat keluar berisi *form* untuk menambahkan data surat yang akan dicatat dan diarsipkan beserta lampiran file surat berupa gambar hasil *scan*.

10. Halaman Edit Surat Keluar

Gambar 5.10 Halaman Edit Surat Keluar Aplikasi Arsip

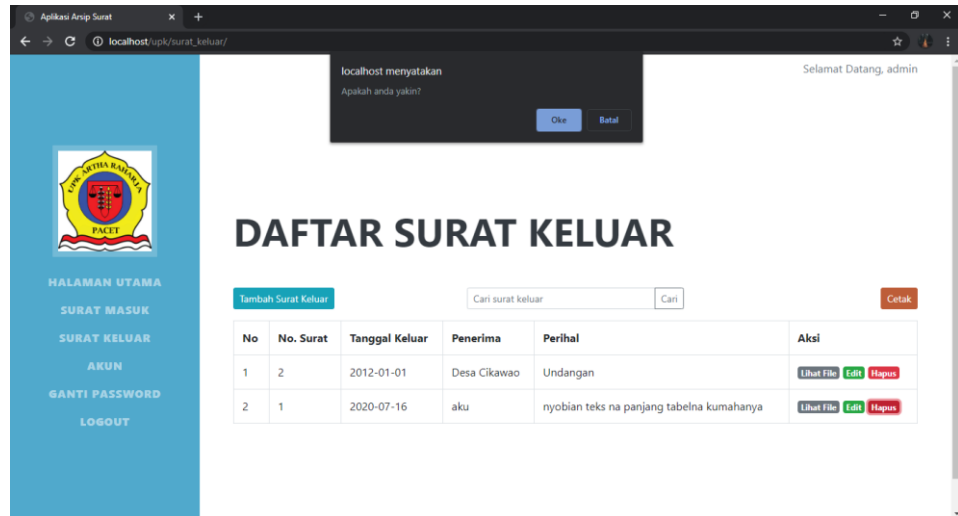
Halaman edit surat keluar berisi *form* untuk mengubah data surat keluar yang sudah tercatat di aplikasi.

11. Halaman Lihat File Surat Keluar

Gambar 5.11 Halaman Lihat File Surat Keluar Aplikasi Arsip

Halaman lihat file surat keluar berisi rincian surat beserta gambar file surat hasil *scan*.

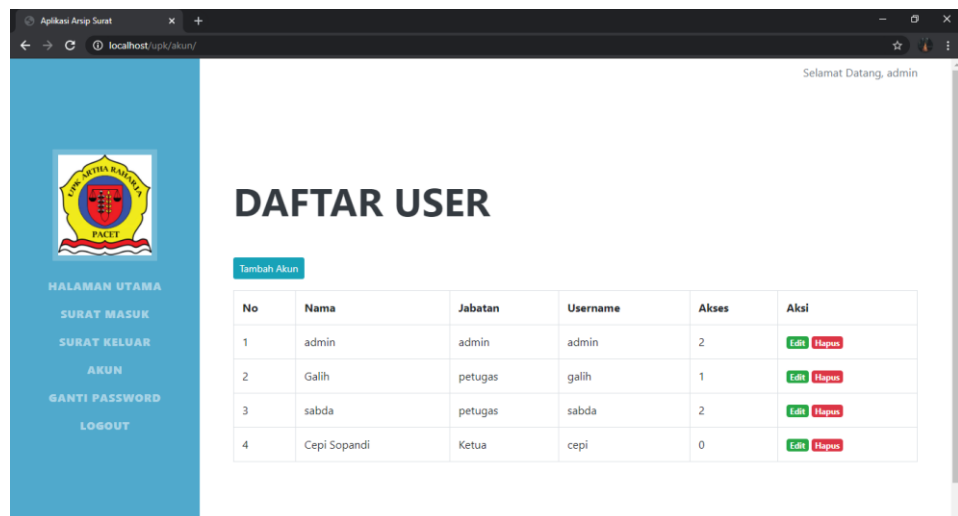
12. Halaman Hapus Surat Keluar



Gambar 5.12 Halaman Hapus Surat Keluar Aplikasi Arsip

Halaman hapus surat keluar berisi sebuah pemberitahuan untuk melanjutkan hapus data surat keluar atau membatalkannya.

13. Halaman Manajemen Akun



Gambar 5.13 Halaman Manajemen Akun Aplikasi Arsip

Halaman manajemen akun berisi data akun yang dibuat admin dan sudah terdaftar di aplikasi diantaranya nama, jabatan, *username*, dan akses.

14. Halaman Tambah Akun

Sistem Arsip Pupket.com/akun/tambah.php

Selamat Datang, admin

TAMBAH USER

Nama
Enjang Herdiana

Jabatan
Sraf Umum

Username
enjang

Akses
Petugas

Simpan

Gambar 5.14 Halaman Tambah Akun Aplikasi Arsip

Halaman tambah akun hanya bisa diakses oleh admin, dimana halaman ini berfungsi untuk menambah akun pengguna dengan mengisi data diantaranya nama, jabatan, *username*, dan memilih hak akses. Untuk password akun, akan terisi secara otomatis sama dengan *username*, jika pengguna ingin mengganti *password* pengguna bisa mengganti sendiri dengan mengakses menu Ganti Password.

15. Halaman Edit Data Akun

Aplikasi Arsip Sunat

localhost/upk/akun/edit.php?id=10

Selamat Datang, admin

EDIT USER

Nama
sabda

Jabatan
petugas

Username
sabda

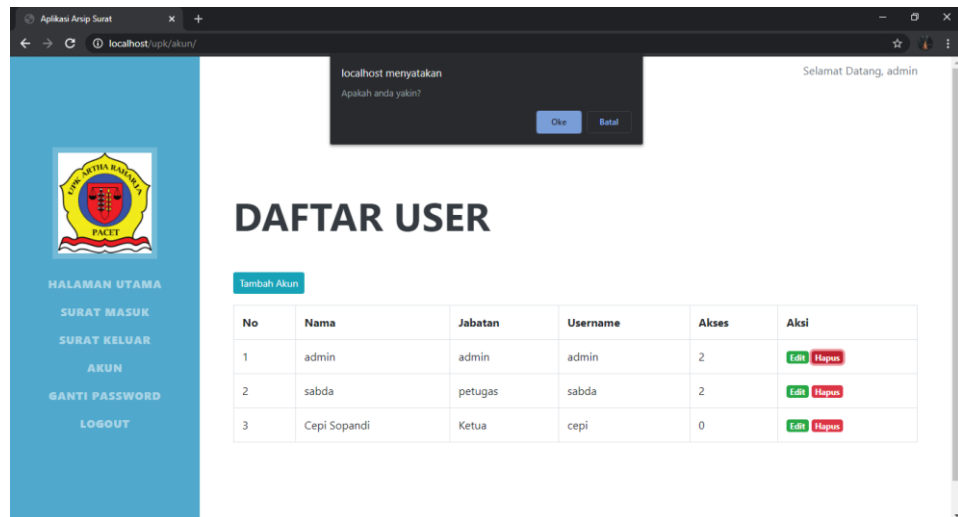
Akses
Admin

Edit

Gambar 5.15 Halaman Edit Akun Aplikasi Arsip

Halaman edit data akun atau *user* berfungsi untuk merubah data akun yang sudah ada.

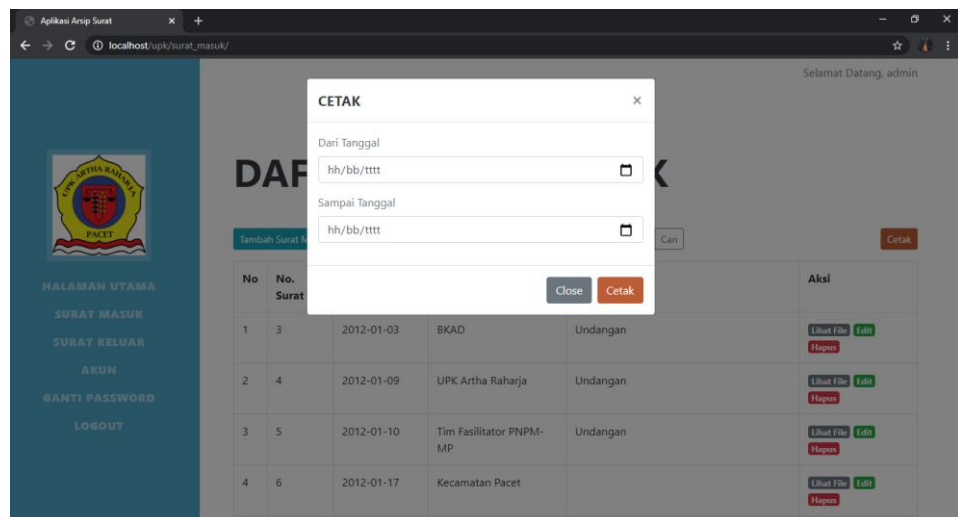
16. Halaman Hapus Akun



Gambar 5.16 Halaman Hapus Akun Aplikasi Arsip

Halaman hapus akun berisi sebuah pemberitahuan untuk melanjutkan hapus akun atau membatalkannya.

17. Halaman Cetak Laporan Surat

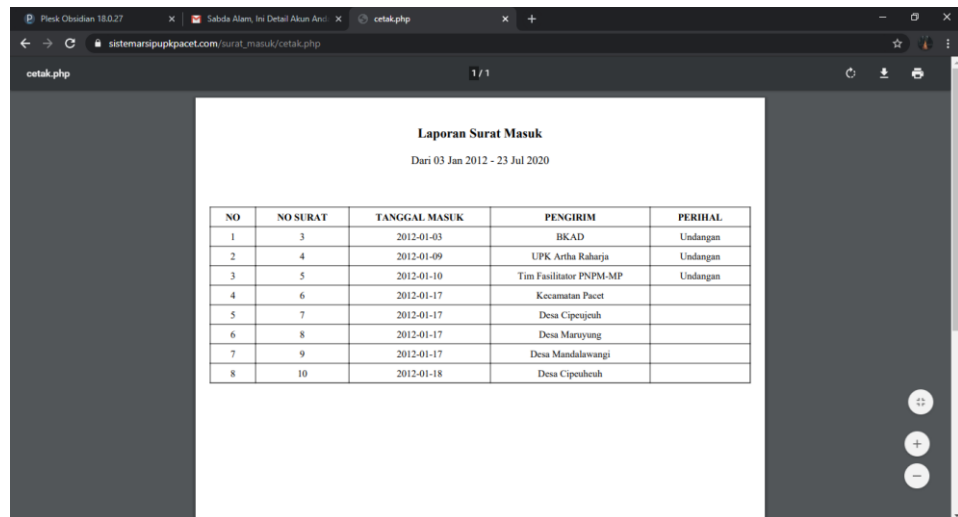


Gambar 5.17 Halaman Cetak Laporan Aplikasi Arsip

Halaman cetak laporan berfungsi untuk mencetak rekap laporan surat masuk maupun surat keluar berdasar periode tertentu dengan cara

menginputkan dari tanggal yang diinginkan sampai dengan tanggal yang diinginkan.

18. Halaman Hasil Cetak Laporan Surat

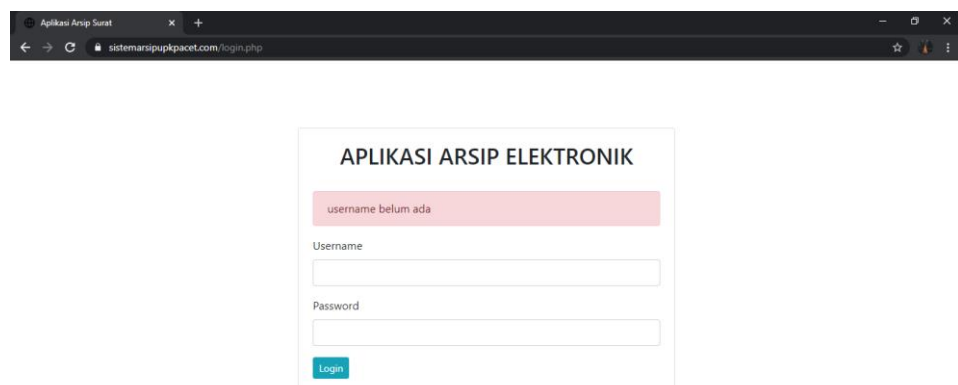


NO	NO SURAT	TANGGAL MASUK	PENGIRIM	PERHAL
1	3	2012-01-03	BKAD	Undangan
2	4	2012-01-09	UPK Artha Raharja	Undangan
3	5	2012-01-10	Tim Fasilitator PNPM-MP	Undangan
4	6	2012-01-17	Kecamatan Pacet	
5	7	2012-01-17	Desa Cipeujeuh	
6	8	2012-01-17	Desa Maruyung	
7	9	2012-01-17	Desa Mandalarwangi	
8	10	2012-01-18	Desa Cipeujeuh	

Gambar 5.18 Halaman Hasil Cetak Laporan Aplikasi Arsip

Halaman hasil cetak laporan surat berupa file .pdf yang akan di download secara otomatis ketika mengisi dari tanggal, sampai tanggal dan menekan tombol cetak.

19. Halaman Kesalahan *Login Username*



APLIKASI ARSIP ELEKTRONIK

username belum ada

Username

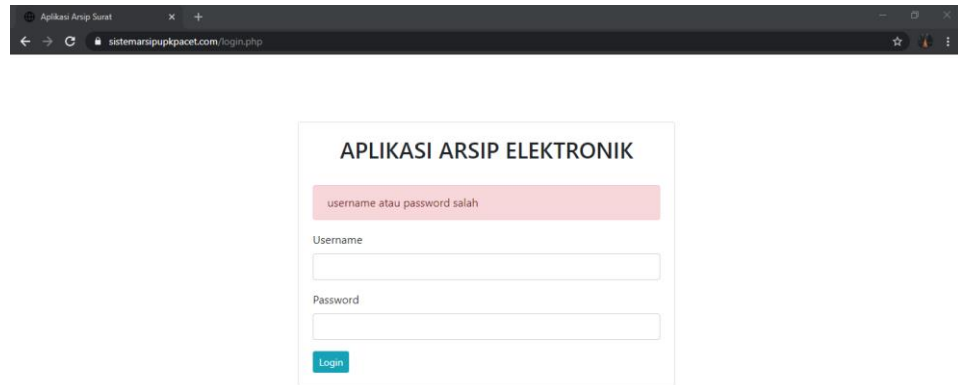
Password

[Login](#)

Gambar 5.19 Halaman Kesalahan *Login Username*

Halaman ini berisi pesan kesalahan saat *user* memasukkan *username* yang salah atau *username* yang belum ada pada sistem.

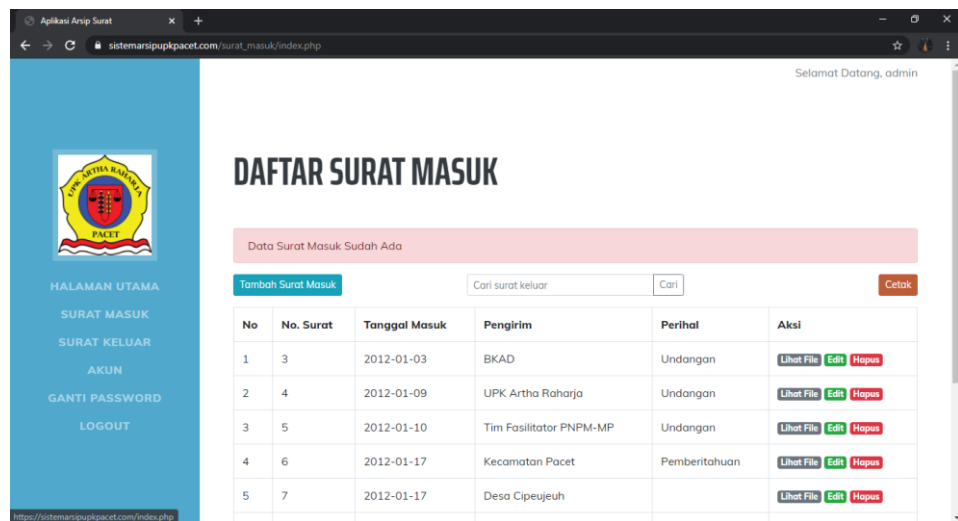
20. Halaman Kesalahan *Login Password*



Gambar 5.20 Halaman Kesalahan *Login Password*

Halaman ini berisi pesan kesalahan pada saat *user* memasukkan *password* yang salah pada sistem saat melakukan *login* kedalam aplikasi.

21. Kesalahan Input Data Surat



Gambar 5.21 Halaman Kesalahan *Input Data Surat*

Halam ini bersi pesan kesalahan pada saat *user* memasukkan data surat masuk atau keluar dengan nomor surat yang sudah ada didalam sistem atau data sudah ada dan tidak boleh memasukkan data yang sama atau ganda pada sistem.

5.1.2 Implementasi Algoritma *Sequential Searching*

Implementasi algoritma *sequential searching* dilakukan dengan cara pengkodean menggunakan bahasa pemrograman PHP pada aplikasi. Implementasi pengkodean algoritma *sequential searching* dilakukan dengan cara melakukan perulangan di PHP yaitu mencocokkan *key* yang dimasukkan dengan data tanggal surat atau alamat surat, ketika data belum ditemukan maka perulangan akan terus berlanjut sampai data ditemukan atau data pada *array* sudah habis dibandingkan. Berikut implementasi dari algoritma *sequential searching*:

```
//mengambil semua kolom pada tabel surat masuk
$sql_read="SELECT * FROM surat_masuk";
$query_read=mysqli_query($koneksi, $sql_read);

//memeriksa GET cari
if(isset($_GET['cari'])){
    while($row_surat = mysqli_fetch_assoc($query_read)){
        $surat[] = $row_surat;
    }
    //proses membandingkan kata yang dicari dengan data surat
    $cari = $_GET['cari'];
    $hasil = '';
    $ditemukan = false;
    $i=0;
    while($i<=count($surat) && !$ditemukan)
    {
        if( (isset($surat[$i])) && (($surat[$i]['nomor_surat'] == $cari
        ) || ($surat[$i]['pengirim'] == $cari) ) ){
            $hasil = $surat[$i];
            $ditemukan=true;
        }
        $i++;
    }
}
```

Gambar 5.22 Implementasi Algoritma *Sequential Searching*

Untuk penjelasan pengkodean algoritma *sequential searching* diatas adalah sebagai berikut:

1. Variabel *query_read* mengambil semua kolom pada data tabel surat masuk.
2. Ketika *user* melakukan sebuah pencarian dan memasukkan *key* pada pencarian, data surat yang ada pada *database* dimasukkan ke dalam *array* dengan nama variabel yaitu surat.

3. Inisialisasi variabel dengan nama cari yang akan diisi data dari *key* yang di *input* oleh *user*.
4. Dilanjutkan dengan melakukan proses perulangan untuk membandingkan data surat masuk.
5. Inisialisasi variabel dengan nama hasil yang diisi *string* kosong yang akan digunakan untuk menampung data hasil pencarian.
6. Inisialisasi variabel dengan nama ditemukan dengan nilai *false* yang akan digunakan untuk pengkondisian pada perulangan.
7. Inisialisasi variabel dengan nama *i* yang di isi 0 (nol).
8. Jika $i \leq \text{jumlah data surat}$, dan variabel ditemukan masih bernilai *false* lakukan perulangan.
9. Pada proses perulangan dilakukan perbandingan berdasarkan kolom nomor surat dan alamat surat, apakah data surat ke[*i*] sama dengan *key* yang di *input* oleh pengguna.
10. Jika data sama dan ditemukan maka variabel hasil akan diisi oleh data surat yang ditemukan dan variabel ditemukan bernilai *true*.
11. Perulangan berhenti dan hasil ditampilkan.
12. Jika data belum ditemukan perulangan akan terus berlanjut sampai data ditemukan atau data pada *array* tidak ada yang cocok.

5.2 Pengujian

Pengujian merupakan tahapan akhir dimana aplikasi diuji kemampuan dan keefektifannya sehingga didapatkan kekurangan dan kelemahan yang kemudian dilakukan pengkajian ulang dan perbaikan terhadap aplikasi menjadi lebih baik.

5.2.1 Pengujian Sistem

Pengujian sistem adalah proses menganalisis kinerja sistem dengan menyesuaikan proses sistem yang ada. Pengujian yang dilakukan menggunakan metode *black-box*, dimana pengujian difokuskan kepada fungsionalitas sistem dalam mengelola data surat masuk dan surat keluar, juga mengetahui

kemampuan sistem dalam menangani kesalahan-kesalahan yang dilakukan oleh pengguna.

1. Pengujian Proses *login*

Tabel 5.1 Pengujian Proses *Login*

No	Cara pengujian	Hasil yang diharapkan	Hasil pengujian
1	User mengetik <i>localhost/upk</i>	Menampilkan halaman <i>login</i>	Berhasil
2	Memasukkan <i>username</i> dan <i>password</i> dengan kondisi (benar) lalu menekan tombol <i>login</i>	Berhasil masuk ke halaman awal aplikasi	Berhasil
3	Memasukkan <i>username</i> dengan kondisi (salah) lalu menekan tombol <i>login</i>	Menampilkan pesan <i>username</i> atau <i>password</i> salah dan kembali ke halaman <i>login</i>	Berhasil
4	Memasukkan <i>password</i> dengan kondisi (salah) lalu menekan tombol <i>login</i>	Menampilkan pesan <i>username</i> atau <i>password</i> salah dan kembali ke halaman <i>login</i>	Berhasil
5	Memasukkan <i>username</i> dengan kondisi (salah) lalu menekan tombol <i>login</i>	Berhasil masuk ke halaman awal aplikasi	Gagal
6	Memasukkan <i>password</i> dengan kondisi (salah) lalu menekan tombol <i>login</i>	Berhasil masuk ke halaman awal aplikasi	Gagal

2. Pengujian Manajemen Surat Masuk

Tabel 5.2 Pengujian Manajemen Surat Masuk

No	Cara pengujian	Hasil yang diharapkan	Hasil pengujian
1	Memilih menu surat masuk	Menampilkan halaman surat masuk	Berhasil
2	Menekan tombol tambah surat masuk	Menampilkan halaman tambah surat masuk	Berhasil
3	Menambahkan data surat masuk dan menekan tombol simpan	Data surat masuk berhasil di simpan dan kembali ke halaman surat masuk	Berhasil
4	Menambahkan data surat masuk dengan kondisi data sama dengan data yang sudah ada dan menekan tombol simpan	Data tidak ditambahkan ke aplikasi	Gagal
5	Menekan tombol lihat file	Menampilkan halaman file surat	Berhasil
6	Menekan tombol edit	Menampilkan halaman edit surat masuk	Berhasil
7	Merubah data surat masuk dan menekan tombol simpan	Data surat masuk berhasil diubah dan kembali ke halaman surat masuk	Berhasil
8	Menekan tombol hapus	Menampilkan pesan peringatan untuk menghapus data surat masuk yang dipilih	Berhasil

9	Menekan tombol oke pada pesan peringatan hapus data surat masuk	Berhasil menghapus data surat masuk	Berhasil
10	Menekan tombol batal pada pesan peringatan hapus data surat masuk	Kembali ke halaman surat masuk	Berhasil
11	Menekan tombol cetak	Menampilkan form cetak data surat masuk	Berhasil
12	Memasukkan form cetak laporan dari tanggal dan sampai tanggal lalu menekan tombol cetak	Sistem memproses dan menampilkan konfirmasi download file	Berhasil

3. Pengujian Manajemen Surat Keluar

Tabel 5.3 Pengujian Manajemen Surat Keluar

No	Cara pengujian	Hasil yang diharapkan	Hasil pengujian
1	Memilih menu surat keluar	Menampilkan halaman surat keluar	Berhasil
2	Menekan tombol tambah surat keluar	Menampilkan halaman tambah surat keluar	Berhasil
3	Menambahkan data surat keluar dan menekan tombol simpan	Data surat keluar berhasil di simpan dan kembali ke halaman surat keluar	Berhasil
4	Menambahkan data surat keluar dengan kondisi data sama dengan data yang sudah ada dan menekan tombol simpan	Data tidak ditambahkan ke aplikasi	Gagal

5	Menekan tombol lihat file	Menampilkan halaman file surat	Berhasil
6	Menekan tombol edit	Menampilkan halaman edit surat keluar	Berhasil
7	Merubah data surat keluar dan menekan tombol simpan	Data surat keluar berhasil diubah dan kembali ke halaman surat keluar	Berhasil
8	Menekan tombol hapus	Menampilkan pesan peringatan untuk menghapus data surat keluar yang dipilih	Berhasil
9	Menekan tombol oke pada pesan peringatan hapus data surat keluar	Berhasil menghapus data surat keluar	Berhasil
10	Menekan tombol batal pada pesan peringatan hapus data surat keluar	Kembali ke halaman surat keluar	Berhasil
11	Menekan tombol cetak	Menampilkan form cetak data surat keluar	Berhasil
12	Memasukkan form cetak laporan dari tanggal dan sampai tanggal lalu menekan tombol cetak	Sistem memproses dan menampilkan konfirmasi download file	Berhasil

4. Pengujian Manajemen Akun

Tabel 5.4 Pengujian Manajemen Akun

No	Cara pengujian	Hasil yang diharapkan	Hasil pengujian
1	Menekan menu akun	Menampilkan halaman akun	Berhasil
2	Menekan tombol tambah akun	Menampilkan halaman tambah akun	Berhasil
3	Menambahkan data <i>user</i> dan menekan tombol simpan	Data <i>user</i> ditambahkan dan kembali ke halaman akun	Berhasil
4	Menambahkan data <i>user</i> dengan kondisi akun sudah ada dan menekan tombol simpan	Data tidak ditambahkan ke aplikasi	Gagal
5	Menekan tombol edit	Menampilkan halaman edit akun	Berhasil
6	Merubah data akun dan menekan tombol edit	Akun berhasil diedit dan kembali ke halaman akun	Berhasil
7	Menekan tombol hapus	Menampilkan pesan peringatan konfirmasi hapus	Berhasil
8	Menekan tombol oke pada pesan peringatan hapus	Akun berhasil dihapus dan kembali ke halaman akun	Berhasil
9	Menekan tombol batal pada pesan peringatan hapus	Kembali ke halaman akun	Berhasil

5. Pengujian Fitur

Tabel 5.5 Pengujian Fitur Aplikasi Arsip

No	Cara pengujian	Hasil yang diharapkan	Hasil pengujian
1	Memasukkan <i>key</i> pada kolom pencarian surat masuk	Menampilkan halaman hasil pencarian	Berhasil
2	Memasukkan <i>key</i> pada kolom pencarian surat keluar	Menampilkan halaman hasil pencarian	Berhasil
3	Menekan menu ganti <i>password</i>	Menampilkan halaman ganti <i>password</i>	Berhasil
4	Memasukkan <i>password</i> dengan kondisi (benar) dan menekan tombol ubah	Data <i>password</i> berhasil diubah	Berhasil
5	Memasukkan <i>password</i> lama dengan kondisi (salah) dan menekan tombol ubah	Menampilkan halaman peringatan <i>password</i> lama salah	Berhasil
6	Memasukkan konfirmasi <i>password</i> dengan kondisi (benar) dan menekan tombol ubah	Menampilkan halaman peringatan <i>password</i> baru tidak sama	Berhasil
7	Menekan tombol ubah tanpa mengisi data <i>password</i>	Menampilkan halaman peringatan data harus diisi	Berhasil

5.2.2 Pengujian Algoritma *Sequential Searching*

Penerapan algoritma *sequential searching* pada fitur pencarian pada aplikasi arsip elektronik di UPK Artha Raharja merupakan percobaan penyusun. Proses pencarian dibatasi berdasarkan nomor surat dan pengirim atau penerima surat. Untuk proses pengujian lebih jelasnya bisa dilihat pada tabel berikut:

Tabel 5.6 Pengujian Algoritma *Sequential Searching*

No	Cara pengujian	Hasil Pencarian	Hasil pengujian
1	Memasukkan <i>key</i> berdasarkan nomor surat pada kolom pencarian dengan kondisi (sesuai) dengan data yang dicari dan data ada dalam <i>database</i>	Data yang dicari ditemukan dan ditampilkan	Berhasil
2	Memasukkan <i>key</i> berdasarkan nomor surat pada kolom pencarian dengan kondisi (tidak sesuai) dengan data yang dicari dan data ada dalam <i>database</i> .	Data yang dicari tidak ditemukan dan tidak ditampilkan	Berhasil
3	Memasukkan <i>key</i> berdasarkan alamat surat pada kolom pencarian dengan kondisi (sesuai) dengan data yang dicari dan data ada dalam <i>database</i> .	Data yang dicari ditemukan dan ditampilkan	Berhasil

4	Memasukkan <i>key</i> berdasarkan alamat surat pada kolom pencarian dengan kondisi (tidak sesuai) dengan data yang dicari dan data ada dalam <i>database</i> .	Data yang dicari tidak ditemukan dan tidak ditampilkan	Berhasil
5	Memasukkan <i>key</i> berdasarkan kategori perihal surat pada kolom pencarian dengan kondisi (sesuai) dengan data yang dicari dan data ada dalam <i>database</i> .	Data yang dicari ditemukan dan ditampilkan	Berhasil
6	Memasukkan <i>key</i> berdasarkan kategori perihal surat pada kolom pencarian dengan kondisi (tidak sesuai) dengan data yang dicari dan data ada dalam <i>database</i> .	Data yang dicari tidak ditemukan dan tidak ditampilkan	Berhasil
7	Memasukkan <i>key</i> berdasarkan kategori tanggal surat pada kolom pencarian dengan kondisi (sesuai) dengan data yang dicari dan data ada dalam <i>database</i> .	Data yang dicari ditemukan dan ditampilkan	Berhasil
8	Memasukkan <i>key</i> berdasarkan kategori tanggal surat pada kolom pencarian dengan kondisi	Data yang dicari tidak ditemukan dan tidak ditampilkan	Berhasil

	(tidak sesuai) dengan data yang dicari dan data ada dalam <i>database</i>		
--	---	--	--

Berdasarkan hasil pengujian diatas, algoritma *sequential searching* dapat diterapkan dan berjalan pada fitur pencarian aplikasi arsip UPK Artha Raharja Kecamatan Pacet berbasis *web* dengan kelemahan *key* yang dimasukkan harus sama persis dengan data yang akan dicari pada *database*. Jumlah dan letak data juga mempengaruhi waktu pencarian, jika letak data yang di cari berada di posisi akhir *database* maka proses pencarian semakin lama.

BAB VI

KESIMPULAN DAN SARAN

Berdasarkan hasil dari penelitian ini yang telah penyusun lakukan, penyusun mencoba membuat suatu kesimpulan dan mengajukan beberapa saran-saran yang berhubungan dengan pembahasan yang telah dikemukakan pada bab-bab sebelumnya.

6.1 Kesimpulan

Dari beberapa yang telah diuraikan dalam laporan ini, maka disimpulkan beberapa hal sebagai berikut:

1. Aplikasi arsip elektronik dapat mencatat data agenda surat masuk maupun surat keluar sehingga tidak memerlukan buku agenda surat karena data surat disimpan secara digital di dalam aplikasi.
2. Aplikasi arsip elektronik dapat menyimpan berkas surat masuk dan surat keluar dengan cara di *scan* dan hasil *scan* tersebut disimpan di *database* aplikasi sehingga tidak memerlukan ruang penyimpanan dan terhindar dari kerusakan berkas atau hilang.
3. Aplikasi arsip elektronik dapat mempermudah proses pencarian berkas surat dalam *database* dengan adanya fitur pencarian.
4. Algoritma *Sequential Searching* dapat diterapkan pada aplikasi dan dapat berjalan untuk melakukan sebuah pencarian meskipun terdapat kelemahan dalam proses pencariannya.
5. Hasil pengujian menunjukkan bahwa fungsi-fungsi berjalan sesuai dengan rancangan yang dibuat di awal.

6.2 Saran

Adapun saran yang dapat penyusun berikan mengenai aplikasi arsip elektronik di UPK artha Raharja Kecamatan Pacet agar lebih baik adalah sebagai berikut:

1. Perlu adanya pengembangan aplikasi dalam segi fungsionalitas agar berkas yang diarsipkan bukan hanya sekedar surat masuk dan surat keluar.
2. Perlu adanya pengembangan tampilan agar lebih menarik dan rapi sesuai dengan kebutuhan.
3. Lakukan perbandingan pada fitur pencarian antara algoritma *sequential searching* dengan *SQL search* ataupun dengan sistem pencarian yang lain untuk mengetahui sistem pencarian mana yang akan diterapkan dan lebih cocok pada aplikasi yang akan dibuat nantinya.

DAFTAR PUSTAKA

- Abdullah M, Muhammad F, Eddy S. (2017). Implementasi metode sequential dalam pencarian pendistribusian barang pada cargo integration sistem. *Jurnal dan penelitian teknik informatika sinkron*, 2, 2.
- Anisya S, Mayang S. (2018). Implementasi Algoritma sequential searching untuk pencarian nomor surat pada sistem arsip elektronik. *Jurnal Pseudocode Teknik Informatika Fakultas Teknik Universitas Muhammadiyah Bengkulu*, V, 1.
- Hidayatullah, P. Kawistara, J, K. 2017. *Pemrograman web Edisi revisi*. Bandung: Informatika.
- Kaban, Roberto. 2019. *Bootstrap CSS Framework*. Yogyakarta: Andi.
- Marissa U, Yovi A. (2019). Implementasi algoritma sequential searching pada sistem pelayanan puskesmas menggunakan bootstrap (studi kasus puskesmas kampung bali bengkulu). *Jurnal Ilmiah dan Informatika Terapan Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Bengkulu*, 2, 1.
- Munir, R. Lidya L. 2016. *Algoritma dan pemrograman bahasa pascal dan C Edisi keenam*. Bandung: Informatika.
- Raharjo, Budi. 2016. *Modul pemrograman web (HTML, PHP & MySQL/MariaDB Edisi keempat*. Bandung: Modula.
- Rosa A, S. Shalahuddin, M. 2018. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.
- Sedarmayanti. 2018. *Tata kearsipan dengan manfaat teknologi modern Revisi ketiga*. Bandung: Mandar Maju.

Sublimetext.com. Sublime Text 3 (*Build 3211*) diakses pada 17 April 2020 dari
<https://www.sublimetext.com/>

Wikipedia.org. Sublime Text. Diakses pada 17 April 2020 dari
https://en.wikipedia.org/wiki/Sublime_Text

Wikipedia.org. Penjelajah Web. Diakses pada 17 april 2020 dari
https://id.wikipedia.org/wiki/Penjelajah_web

LAMPIRAN

Lampiran 1 Kuesioner

KUESIONER

Aplikasi Arsip Elektronik Berbasis Web di UPK Artha Rahardja Kecamatan Pacet

Nama : EMJANG HERDIANA
Jabatan : STAF UMUM

Petunjuk pengisian:

Berilah tanda Checklist (✓) pada salah satu jawaban yang sesuai pendapat saudara.

Kriteria Penilaian:

Setuju : S

Netral : Netral

Tidak Setuju : TS

No	Pernyataan	S	N	TS
1	Di UPK Artha Rahardja Kecamatan Pacet menggunakan surat-menyurat untuk berkomunikasi dengan instansi lain.	✓		
2	Berkas surat yang sudah dibuat atau diterima diarsipkan.	✓		
3	Di UPK Artha Rahardja Kecamatan Pacet mempunyai sistem administrasi pengarsipan.	✓		
4	Di UPK Artha Rahardja Kecamatan Pacet menggunakan sistem administrasi pengarsipan konvensional.	✓		
5	Dengan cara pengarsipan konvensional administrasi arsip berjalan dengan baik.	✓		
6	Pencatatan arsip dilakukan menggunakan buku.	✓	✓	

7	Penyimpanan arsip menggunakan penyimpanan khusus lemari arsip.	✓		
8	Penyimpanan arsip dilakukan dalam jangka waktu yang panjang.		✓	
9	Jangka waktu yang panjang membutuhkan buku yang tebal untuk proses pengarsipan.	✓		
10	Jangka waktu yang panjang akan memakan banyak tempat untuk berkas arsip.		✓	
11	Dengan cara menyimpan di lemari arsip, ada masalah di keamanan berkas.		✓	
12	Dengan cara menyimpan di lemari arsip berkas rentan hilang atau berkas rusak.	✓		
13	Direkomendasikan pembuatan Aplikasi arsip elektronik berbasis web untuk membantu proses pengarsipan.	✓		
14	Pemberkasan arsip di basis data komputer akan membantu proses pengarsipan lebih mudah.	✓		
15	Fitur penyimpanan di basis data akan membantu mengurangi tempat penyimpanan arsip.	✓		
16	Aplikasi arsip elektronik membantu pengelompokan data arsip.		✓	
17	Pendataan arsip di komputer membantu proses pengarsipan.	✓		
18	Aplikasi arsip elektronik membantu proses pengarsipan jadi lebih efisien.		✓	
19	Dengan aplikasi arsip elektronik arsip lebih mudah dikendalikan.		✓	
20	Di UPK Artha Rahardja Kecamatan Pacet ada petugas khusus untuk mengelola arsip.		✓	
21	Dua pengguna di aplikasi arsip elektronik cukup untuk mengoperasikan aplikasi administrasi arsip.	✓		
22	Dokumen yang sudah diarsipkan akan digunakan kembali		✓	

23	Dokumen yang dicari di cek manual di lemari arsip.	✓		
24	Pencarian dokumen dilakukan cek satu-persatu di lemari arsip		✓	
25	Pencarian dokumen satu-persatu membutuhkan waktu cukup lama.	✓		
26	Perlu ditambahkan fitur pencarian dokumen di aplikasi arsip elektronik.	✓		
27	Fitur pencarian di aplikasi arsip elektronik akan membantu pencarian dokumen.	✓		
28	Ada pelaporan administrasi arsip.	✓		
29	Pelaporan arsip dilakukan dengan cara diketik manual.		✓	
30	Perlu ditambahkan fitur laporan berdasar waktu yang ditentukan di aplikasi arsip elektronik.	✓		
31	Fitur laporan akan membantu proses pelaporan arsip menjadi lebih mudah.	✓		
32	Pengguna dengan hak akses admin bisa menghapus atau menambah pengguna lain.	✓		
33	Hak akses untuk petugas hanya bisa tambah, ubah dan melihat data.	✓		
34	Hak akses untuk admin bisa tambah, ubah, melihat dan hapus data.	✓		
35	Aplikasi arsip elektronik hanya bisa digunakan untuk petugas pengelola arsip dan sekretariat UPK Artha Rahardja Kecamatan Pacet.	✓		

Lampiran 2 *Source Code* Aplikasi

Halaman Utama

```

<?php
    $query_surat_masuk = mysqli_query($koneksi, "SELECT * FROM
surat_masuk");

    $jumlah_surat_masuk = mysqli_num_rows($query_surat_masuk);
    $query_surat_keluar = mysqli_query($koneksi, "SELECT * FROM
surat_keluar");

    $jumlah_surat_keluar = mysqli_num_rows($query_surat_keluar);
?>

<div class="container-fluid p-0">
    <section class="resume-section" id="about">
        <div class="resume-section-content">
            <h2>ARSIP SURAT</h2>
            <h3>UPK ARTHA RAHARJA</h3>
            <div class="row">
                <div class="col-md-6">
                    <div class="card">
                        <div class="card-body">
                            <strong>Jumlah Surat Masuk</strong>
                            <p class="float-right"><?php echo $jumlah_surat_masuk;
?></p>
                        </div>
                    </div>
                </div>
            </div>
            <div class="col-md-6">
                <div class="card">
                    <div class="card-body">

```



```

        <strong>Jumlah Surat Keluar</strong>
        <p class="float-right"><?php echo $jumlah_surat_keluar;
?></p>

    </div>
</div>
</div>
</div>
<div id="bar-chart" class="mt-3"></div>
</div>
</section>
<hr class="m-0" />
</div>

<?php require 'footer.php'; ?>

<script>
$(document).ready(function(){
    Morris.Bar({
        element: 'bar-chart',
        data: [
            { y: 'Surat Masuk', a: <?php echo $jumlah_surat_masuk; ?> },
            { y: 'Surat Keluar', a: <?php echo $jumlah_surat_keluar; ?>, },
        ],
        xkey: 'y',
        ykeys: ['a'],
        labels: ['Jumlah Surat'],
        hideHover: 'auto',
        parseTime: false
    });
});
</script>

```


Surat masuk

```

<?php
    //mengambil semua kolom pada tabel surat masuk
    $sql_read="SELECT * FROM surat_masuk";
    $query_read=mysqli_query($koneksi, $sql_read);
    //memeriksa GET cari
    if(isset($_GET['cari'])){
        while($row_surat = mysqli_fetch_assoc($query_read)){
            $surat[] = $row_surat;
        }
        //proses membandingkan kata yang dicari dengan data surat
        $cari = $_GET['cari'];
        $hasil = "";
        $ditemukan = false;
        $i=0;
        while($i<=count($surat) && !$ditemukan)
        {
            if( (isset($surat[$i])) && (($surat[$i]['nomor_surat'] ==
$cari) || ($surat[$i]['pengirim'] == $cari) ) ){
                $hasil = $surat[$i];
                $ditemukan=true;
            }
            $i++;
        }
    }
?>

<!-- Page Content-->
<div class="container-fluid p-0">
    <!-- About-->
    <section class="resume-section">
        <div class="resume-section-content"

```

```

<h2 class="mb-5">Daftar Surat Masuk</h2>
<?php if(isset($_SESSION['berhasil'])) : ?>
    <div class="alert alert-success">
        <?php
            echo $_SESSION['berhasil'];
            unset($_SESSION['berhasil']);
        ?>
    </div>
<?php endif; ?>
<div class="row">
    <div class="col-md-4">
        <?php if($_SESSION['user']['akses'] != 0) : ?>
            <a href="tambah.php" class="btn btn-info btn-sm
mb-3">Tambah Surat Masuk</a>
            <?php endif; ?>
        </div>
        <div class="col-md-4">
            <form action="" method="get">
                <div class="input-group input-group-sm
mb-3">
                    <input type="text"
class="form-control" name="cari" placeholder="Cari surat keluar">
                    <div class="input-group-
append">
                        <button class="btn btn-
outline-secondary" type="submit" id="button-addon2">Cari</button>
                    </div>
                </div>
            </form>
        </div>
    </div>
    <div class="col-md-4 align="right">

```

```

        <a href="#" class="btn btn-primary btn-sm" data-
toggle="modal" data-target="#cetakSuratMasuk">Cetak</a>

    </div>

</div>

<table class="table table-bordered">

    <tr>

        <th>No</th>
        <th>No. Surat</th>
        <th>Tanggal Masuk</th>
        <th>Pengirim</th>
        <th>Perihal</th>
        <?php
if($_SESSION['user']['akses'] != 0) : ?>
        <th>Aksi</th>
        <?php endif; ?>

    </tr>

    <?php if(mysqli_num_rows($query_read)
== 0) : ?>

        <td class="text-center"
colspan="6">Tidak ada data</td>

    <?php endif; ?>

    <?php if(!isset($_GET['cari'])) : ?>
        <?php $no=1;
while($row=mysqli_fetch_assoc($query_read)) : ?>
        <tr>

            <td><?php echo $no;
?></td>

            <td><?php echo
$row['nomor_surat']; ?></td>

```

```

<td><?php echo
$row['tanggal_masuk']; ?></td>

<td><?php echo
$row['pengirim']; ?></td>

<td><?php echo
$row['perihal']; ?></td>

<?php
if($_SESSION['user']['akses'] != 0) : ?>

<td>

<a
href="lihat.php?id=<?php echo $row['id']; ?>" class="badge badge-
secondary">Lihat File</a>

<a
href="edit.php?id=<?php echo $row['id']; ?>" class="badge badge-
success">Edit</a>

<a onclick="return
confirm('Apakah anda yakin?)" href="hapus.php?id=<?php echo $row['id'];
?>&file=<?php echo $row['file_surat']; ?>" class="badge badge-
danger">Hapus</a>

</td>

<?php endif; ?>
</tr>

<?php $no++; endwhile; ?>
<?php elseif($hasil != "") : ?>
<tr>

<td><?php echo 1; ?></td>

<td><?php echo
$hasil['nomor_surat']; ?></td>

<td><?php echo
$hasil['tanggal_masuk']; ?></td>

<td><?php echo
$hasil['pengirim']; ?></td>

```

```

<td><?php echo
$hasil['perihal']; ?></td>

<?php
if($_SESSION['user']['akses'] != 0) : ?>

<td>

<a
href="lihat.php?id=<?php echo $hasil['id']; ?>" class="badge badge-
secondary">Lihat File</a>

<a
href="edit.php?id=<?php echo $hasil['id']; ?>" class="badge badge-
success">Edit</a>

<a onclick="return
confirm('Apakah anda yakin?)" href="hapus.php?id=<?php echo $hasil['id'];
?>&file=<?php echo $hasil['file_surat']; ?>" class="badge badge-
danger">Hapus</a>

</td>

<?php endif; ?>
</tr>
<?php else : ?>
<tr>
<td class="text-
center" colspan="6">Tidak ada data</td>

</tr>

<?php endif; ?>
</table>

</div>
</section>
</div>
<!-- Modal -->
<div class="modal fade" id="cetakSuratMasuk" tabindex="-1" role="dialog"
aria-labelledby="exampleModalLabel" aria-hidden="true">
<div class="modal-dialog">

```

```

<div class="modal-content">
  <div class="modal-header">
    <h5 class="modal-title" id="exampleModalLabel">Cetak</h5>
    <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
      <span aria-hidden="true">&times;</span>
    </button>
  </div>
  <div class="modal-body">
    <form action="cetak.php" method="post">
      <div class="form-group">
        <label>Dari Tanggal</label>
        <input type="date" name="dari" class="form-control">
      </div>

      <div class="form-group">
        <label>Sampai Tanggal</label>
        <input type="date" name="sampai" class="form-control">
      </div>
    </div>
    <div class="modal-footer">
      <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
      <button type="submit" class="btn btn-primary">Cetak</button>
    </form>
  </div>
</div>
</div>

```

Surat keluar

```

<?php
    $sql_read="SELECT * FROM surat_keluar";
    $query_read=mysqli_query($koneksi, $sql_read);

    if(isset($_GET['cari'])){
        while($row_surat = mysqli_fetch_assoc($query_read)){
            $surat[] = $row_surat;
        }
        $cari = $_GET['cari'];
        $hasil = "";
        $ditemukan = false;
        $i=0;
        while($i<=count($surat) && !$ditemukan)
        {
            if( (isset($surat[$i])) && (($surat[$i]['nomor_surat'] ==
$cari) || ($surat[$i]['penerima'] == $cari) ) ){
                $hasil = $surat[$i];
                $ditemukan=true;
            }
            $i++;
        }
    }
?>

<!-- Page Content-->
<div class="container-fluid p-0">
    <!-- About-->
    <section class="resume-section">
        <div class="resume-section-content">

            <h2 class="mb-5">Daftar Surat Keluar</h2>

```

```

<?php if(isset($_SESSION['berhasil'])) : ?>
    <div class="alert alert-success">
        <?php
            echo $_SESSION['berhasil'];
            unset($_SESSION['berhasil']);
        ?>
    </div>
<?php endif; ?>
<div class="row">
    <div class="col-md-4">
        <?php if($_SESSION['user']['akses'] != 0) : ?>
            <a href="tambah.php" class="btn btn-info btn-sm
mb-3">Tambah Surat Keluar</a>
        <?php endif; ?>
    </div>

    <div class="col-md-4">
        <form action="" method="get">
            <div class="input-group input-group-sm
mb-3">
                <input type="text"
class="form-control" name="cari" placeholder="Cari surat keluar">
                <div class="input-group-
append">
                    <button class="btn btn-
outline-secondary" type="submit" id="button-addon2">Cari</button>
                </div>
            </div>
        </form>
    </div>

```



```

<div class="col-md-4" align="right">
    <a href="#" class="btn btn-primary btn-sm" data-
toggle="modal" data-target="#cetakSuratKeluar">Cetak</a>
</div>
</div>
<table class="table table-bordered">
    <tr>
        <th>No</th>
        <th>No. Surat</th>
        <th>Tanggal Keluar</th>
        <th>Penerima</th>
        <th>Perihal</th>
        <?php
if($_SESSION['user']['akses'] != 0) : ?>
            <th>Aksi</th>
        <?php endif; ?>
    </tr>
    <?php if(mysqli_num_rows($query_read)
== 0) : ?>
        <td class="text-center"
colspan="6">Tidak ada data</td>
    <?php endif; ?>

    <?php if(!isset($_GET['cari'])) : ?>
        <?php $no=1;
while($row=mysqli_fetch_assoc($query_read)) : ?>
            <tr>
                <td><?php echo $no;
?></td>
                <td><?php echo
$row['nomor_surat']; ?></td>

```

```

<td><?php echo
$row['tanggal_keluar']; ?></td>

<td><?php echo
$row['penerima']; ?></td>

<td><?php echo
$row['perihal']; ?></td>

<?php
if($_SESSION['user']['akses']) : ?>

<td>

<a
href="lihat.php?id=<?php echo $row['id']; ?>" class="badge badge-
secondary">Lihat File</a>

<a
href="edit.php?id=<?php echo $row['id']; ?>" class="badge badge-
success">Edit</a>

<a onclick="return
confirm('Apakah anda yakin?)" href="hapus.php?id=<?php echo $row['id'];
?>&file=<?php echo $row['file_surat']; ?>" class="badge badge-
danger">Hapus</a>

</td>

<?php endif; ?>
</tr>

<?php $no++; endwhile; ?>
<?php elseif($hasil != "") : ?>
<tr>

<td><?php echo 1; ?></td>

<td><?php echo
$hasil['nomor_surat']; ?></td>

<td><?php echo
$hasil['tanggal_keluar']; ?></td>

<td><?php echo
$hasil['penerima']; ?></td>

```

```

<td><?php echo
$hasil['perihal']; ?></td>

<?php
if($_SESSION['user']['akses'] != 0) : ?>

<td>

<a
href="lihat.php?id=<?php echo $row['id']; ?>" class="badge badge-
secondary">Lihat File</a>

<a
href="edit.php?id=<?php echo $row['id']; ?>" class="badge badge-
success">Edit</a>

<a onclick="return
confirm('Apakah anda yakin?)" href="hapus.php?id=<?php echo $row['id'];
?>&file=<?php echo $row['file_surat']; ?>" class="badge badge-
danger">Hapus</a>

</td>

<?php endif; ?>
</tr>
<?php else : ?>
<tr>

<td class="text-
center" colspan="6">Tidak ada data</td>

</tr>

<?php endif; ?>
</table>

</div>
</section>
</div>
<!-- Modal -->
<div class="modal fade" id="cetakSuratKeluar" tabindex="-1" role="dialog"
aria-labelledby="exampleModalLabel" aria-hidden="true">
<div class="modal-dialog">

```

```

<div class="modal-content">
  <div class="modal-header">
    <h5 class="modal-title" id="exampleModalLabel">Cetak</h5>
    <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
      <span aria-hidden="true">&times;</span>
    </button>
  </div>
  <div class="modal-body">
    <form action="cetak.php" method="post">
      <div class="form-group">
        <label>Dari Tanggal</label>
        <input type="date" name="dari" class="form-control">
      </div>

      <div class="form-group">
        <label>Sampai Tanggal</label>
        <input type="date" name="sampai" class="form-control">
      </div>
    </div>
    <div class="modal-footer">
      <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
      <button type="submit" class="btn btn-primary">Cetak</button>
    </form>
  </div>
</div>
</div>

```

Tambah surat

```

<?php
    if(isset($_POST['submit'])){
        $nomor_surat=$_POST['nomor_surat'];
        $tanggal_masuk=$_POST['tanggal_masuk'];
        $pengirim=$_POST['pengirim'];
        $perihal=$_POST['perihal'];
        $file_surat=time()."$_FILES['file_surat']['name'];
        $lokasi='./img/'.$file_surat;
        move_uploaded_file($_FILES['file_surat']['tmp_name'],
$lokasi);

        $sql="INSERT INTO surat_masuk(nomor_surat,
tanggal_masuk, pengirim, perihal, file_surat) VALUES('$nomor_surat',
'$tanggal_masuk', '$pengirim', '$perihal', '$file_surat')";
        $query=mysqli_query($koneksi, $sql);
        $_SESSION['berhasil'] = "Surat masuk berhasil ditambahkan";
        header('Location: index.php');
    }
?>

<!-- Page Content-->
<div class="container-fluid p-0">
    <!-- About-->
    <section class="resume-section">
        <div class="resume-section-content">
            <h2>Tambah Surat Masuk</h2>
            <form action="" method="post" enctype="multipart/form-data">
                <div class="form-group">
                    <label>No. Surat</label>
                    <input type="text"
name="nomor_surat" class="form-control">
                </div>
            </form>
        </div>
    </section>
</div>

```

```

<div class="form-group">
    <label>Tanggal Masuk</label>
    <input type="date"
name="tanggal_masuk" class="form-control">
</div>
<div class="form-group">
    <label>Pengirim</label>
    <input type="text"
name="pengirim" class="form-control">
</div>
<div class="form-group">
    <label>Perihal</label>
    <input type="text" name="perihal"
class="form-control">
</div>
<div class="form-group">
    <label>File Surat</label>
    <input type="file"
name="file_surat" class="form-control">
</div>
<button type="submit" name="submit"
class="btn btn-primary btn-sm">Simpan</button>
</form>
</div>
</section>
</div>

```

Lihat surat

```
<?php
    $id = $_GET['id'];
    $sql_read="SELECT * FROM surat_masuk WHERE id = $id";
    $query_read=mysqli_query($koneksi, $sql_read);
    $row = mysqli_fetch_assoc($query_read);
?>

<!-- Page Content-->
<div class="container-fluid p-0">
    <!-- About-->
    <section class="resume-section">
        <div class="resume-section-content">
            <h2>File Surat</h2>
            <p>Nomor Surat : <?php echo $row['nomor_surat'];?></p>
            <p>Tanggal Masuk : <?php echo $row['tanggal_masuk'];?></p>
            <p>Pengirim : <?php echo $row['pengirim'];?></p>
            <p>Perihal : <?php echo $row['perihal'];?></p>
            
        </div>
    </section>
</div>
```

Hapus surat

```
<?php
    session_start();
    $id=$_GET['id'];
    $file_lama = $_GET['file'];
    $query=mysqli_query($koneksi, "DELETE FROM surat_masuk
WHERE id='$id'");
    $row = mysqli_fetch_assoc($query);
```

```

$lokasi='./img/'.$file_lama;
unlink($lokasi);

$_SESSION['berhasil'] = "Surat masuk berhasil dihapus";
header('location: index.php');

?>

```

Edit surat

```

<?php
    $id=$_GET['id'];
    if(isset($_POST['submit'])){
        $nomor_surat=$_POST['nomor_surat'];
        $tanggal_masuk=$_POST['tanggal_masuk'];
        $pengirim=$_POST['pengirim'];
        $perihal=$_POST['perihal'];
        $file_surat=$_POST['file_lama'];
        $lokasi_lama = './img/'.$_POST['file_lama'];
        if ($_FILES['file_surat']['name']) {
            $file_surat=time()."$_FILES['file_surat']['name'];
            $lokasi='./img/'.$file_surat;
            unlink($lokasi_lama);
            move_uploaded_file($_FILES['file_surat']['tmp_name'],
$lokasi);
        }
        $sql="UPDATE surat_masuk SET nomor_surat='$nomor_surat',
tanggal_masuk='$tanggal_masuk', pengirim='$pengirim', perihal='$perihal',
file_surat='$file_surat' WHERE id='$id'";
        $query=mysqli_query($koneksi, $sql);
        $_SESSION['berhasil'] = "Surat masuk berhasil diubah";
        header('location: index.php');
    }
    $sql_read="SELECT * FROM surat_masuk WHERE id='$id'";

```



```

<input type="text" name="perihal"
class="form-control" value="<?php echo $row['perihal'] ?>">
</div>
<div class="row mb-3">
    <div class="col-md-4">
        <a href="#"></a>
    </div>
    <div class="col-md-8">
        <div class="form-group">
            <label>File
Surat</label>
            <input
type="hidden" name="file_lama" value="<?php echo $row['file_surat']; ?>">
            <input type="file"
name="file_surat" class="form-control">
        </div>
    </div>
</div>
<button type="submit" name="submit"
class="btn btn-primary btn-sm">Simpan</button>
</form>
</div>
</section>
</div>

<!-- Modal -->
<div class="modal fade" id="exampleModal" tabindex="-1"
role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog modal-xl">

```

```

<div class="modal-content">
  <div class="modal-header">
    <h5 class="modal-title" id="exampleModalLabel">Modal
title</h5>
    <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
      <span aria-hidden="true">&times;</span>
    </button>
  </div>
  <div class="modal-body">
    
  </div>
  <div class="modal-footer">
    <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
  </div>
</div>
</div>
</div>
</div>

```

Cetak surat

```

<?php
    require_once '../koneksi.php';
    require_once '../pdf/pdf.php';
    $dari = $_POST['dari'];
    $sampai = $_POST['sampai'];

    $query = mysqli_query($koneksi, "SELECT * FROM surat_masuk
WHERE tanggal_masuk BETWEEN '$dari' AND '$sampai'");
    $pdf =

```

```

'<style>
    @page {
        margin: 20px;
    }
    table {
        width: 100%;
        border-collapse: collapse;
        font-size: 14px;
    }
</style>
<h3 align="center">Laporan Surat Masuk</h3>
<p align="center">Dari '.date('d M Y', strtotime($dari)).' - '.date('d M
Y', strtotime($sampai)).'</p><br><br>
';
$pdf .=
'<table border="1" cellpadding="5" cellspacing="0">
    <tr align="center">
        <th>
            NO
        </th>
        <th>
            NO SURAT
        </th>
        <th>
            TANGGAL MASUK
        </th>
        <th>
            PENGIRIM
        </th>
        <th>
            PERIHAL
        </th>

```

```

';
$pdf .= '</tr>';
$no = 1;
while($data = mysqli_fetch_assoc($query)){
    $pdf .=
        '<tr align="center">
            <td>'.$no.'</td>
            <td>'.$data['nomor_surat'].'</td>
            <td>'.$data['tanggal_masuk'].'</td>
            <td>'.$data['pengirim'].'</td>
            <td>'.$data['perihal'].'</td>
        ';
    $pdf .= '</tr>';
    $no++;
}
$pdf .= '</table>';
$filename = 'Laporan-Surat-Masuk.pdf';
$print = new Pdf();
$print->loadHtml($pdf);
$print->render();
$print->stream($filename, array("Attachment" => false));
exit(0);

```

?>

Lampiran 3 Riwayat Hidup Penulis

RIWAYAT HIDUP PENULIS



Sabda Alam, lahir pada hari Selasa 27 Mei 1997 anak ke 2 dari 3 bersaudara. Memulai pendidikan di SDN Tenjonagara 2 pada tahun (2004-2010), setelah itu melanjutkan pendidikan ke SMPN 1 Pacet pada tahun (2010-2013), setelah lulus meneruskan sekolah ke SMK Itikurih Hibarna Ciparay pada tahun (2013-2016) dengan mengambil jurusan Teknik Komputer Jaringan. Setelah lulus, meneruskan pendidikan di Universitas Bale Bandung dengan mengambil jurusan Teknik Informatika untuk mendapatkan gelar S1. Selain aktif menjadi mahasiswa saya pun aktif di organisasi kepencahayaan di Universitas Bale Bandung yaitu Mapala Cantigi dan menjabat sebagai sekretaris selama 2 tahun, dari tahun 2017 sampai dengan 2018. Selain itu aktif di organisasi Fakultas Teknologi Informasi, salah satu jabatan yang pernah di terima adalah sebagai divisi Jaringan di Komunitas Epoch Digital Techonlogy. Dan saat ini alhamdulillah saya telah menyelesaikan skripsi untuk gelar Sarjana Komputer di Fakultas Teknologi Informasi Universitas Bale Bandung.