

**RANCANG BANGUN SISTEM INFORMASI E-ARSIP SKRIPSI DAN
LAPORAN KERJA PRAKTIK MAHASISWA TEKNIK INFORMATIKA
UNIVERSITAS LAMPUNG DENGAN METOD *SCRUM***

(Skripsi)

**Oleh
AHMAD ILHAM**



**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
2023**

DAFTAR ISI

DAFTAR ISI.....	i
DAFTAR GAMBAR	i
DAFTAR TABEL.....	ii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian.....	3
1.5 Batasan Masalah.....	4
1.6 Sistematika Penulisan Skripsi / tugas akhir.....	4
BAB II TINJAUAN PUSTAKA.....	6
2.1 Arsip	6
2.2 Sistem	6
2.3 Informasi	6
2.4 Sistem Informasi.....	7
2.5 Arsip Elektronik	7
2.6 Website	7

2.7	Framework.....	8
2.8	PHP (PHP Hypertext Preprocessor)	8
2.9	Laravel.....	8
2.10	<i>Database</i>	10
2.11	MySQL.....	10
2.12	<i>Scrum</i>	10
2.13	<i>Blackbox Testing</i>	14
2.14	<i>Unified Modelling Language (UML)</i>	14
2.15	Penelitian Terdahulu.....	17
2.15.1	Sistem Informasi E-Arsip Berbasis Web (Studi Kasus: Pt Haleyora Powerindo Cabang Sorong).....	17
2.15.2	Rancang Bangun Aplikasi Penjualan Online Berbasis Web Menggunakan Metode Scrum.....	18
2.15.3	Analisis Penerapan Metode Scrum Pada Pengembangan Sistem Informasi Akuntansi Koperasi	19
2.15.4	Analisis Perbandingan Metode Manajemen Proyek TI yang Paling Sering Digunakan Di Indonesia dan Luar Negeri: A Literature Review.....	19
2.15.5	A Modern Review On Laravel – PHP Framework	20
2.15.6	Blackbox and Whitebox Testing Techniques – A Literatur Review	
	21	
2.15.7	Pengujian pada Aplikasi Penggajian Pegawai dengan Menggunakan Metode Blackbox Testing.....	21

2.15.8 Pengembangan Aplikasi Tiga-Tingkat Menggunakan Metode Scrum pada Aplikasi Presensi Karyawan Glints Academy	22
2.16 State of the Art	23
BAB III METODOLOGI PENELITIAN	24
3.1 Waktu dan Tempat	24
3.2 Jadwal Penelitian	24
3.3 Alat Dalam Penelitian	24
3.3.1 Alat Penelitian	24
3.4 Tahapan Penelitian	25
3.4.1 Analisis Permasalahan	26
3.4.2 <i>Product Backlog</i>	27
3.4.3 <i>Sprint Planning</i>	27
3.4.4 <i>Sprint</i>	28
3.4.5 <i>Sprint Review</i>	29
3.4.6 <i>Sprint Retrospective</i>	29
3.4.7 Testing	29
3.4.8 Penulisan Laporan	29
BAB IV HASIL DAN PEMBAHASAN	30
4.1 Hasil	30
4.2 Implementasi	30
4.2.1 Analisis Permasalahan	30

4.2.2	<i>Product Backlog</i>	33
4.2.3	<i>Sprint Planning</i>	36
4.2.4	<i>Sprint</i>	36
4.2.5	<i>Sprint Review</i>	78
4.2.6	<i>Sprint Retrospective</i>	78
4.2.7	<i>Testing</i>	78
DAFTAR PUSTAKA		79

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada zaman sekarang ini, keberadaan dari teknologi informasi telah menjadi suatu hal yang sangat penting dalam berbagai hal. Salah satunya ialah sebagai alat yang membantu suatu organisasi, perusahaan, ataupun lembaga untuk dapat menjadi lebih berkembang dan maju. Beberapa sektor yang memerlukan penerapan dari teknologi informasi ini di antaranya perbankan, kesehatan, perusahaan, Pendidikan dan dunia bisnis. Banyak sekali manfaat yang bisa didapatkan dari adanya teknologi informasi. Dari sekian banyak manfaat yang bisa didapatkan, beberapa diantaranya yaitu lebih mudahnya dan cepatnya dalam melakukan proses komunikasi, proses perdagangan dapat dilakukan secara elektronik, mudahnya akses terhadap suatu informasi, pekerjaan bisa dilakukan secara daring, bahkan lingkungan pun menjadi lebih terjaga dikarenakan penggunaan kertas yang lebih sedikit [1]. Salah satu dari sekian banyak kemudahan yang bisa didapatkan dari kemudahan yang didampak oleh teknologi adalah teknologi dalam mengarsipkan suatu informasi. Dengan adanya arsip secara elektronik dapat mengurangi penggunaan kertas yang biasanya dilakukan secara konvensional dan kemudahan dalam mencari informasi tersebut dengan cepat, mudah dan akurat.

Menurut undang-undang Republik Indonesia Nomor 43 tahun 2009 tentang kearsipan, Arsip adalah rekaman kegiatan atau peristiwa dalam berbagai bentuk dan media sesuai dengan perkembangan teknologi informasi dan komunikasi yang dibuat dan diterima oleh Lembaga negara, pemerintahan daerah, Lembaga Pendidikan, perusahaan, organisasi politik, organisasi kemasyarakatan, dan perseorangan dalam pelaksanaan kehidupan bermasyarakat, berbangsa dan bernegara [2]. Pengelolaan arsip secara konvensional memiliki banyak kelemahan,

seperti memerlukan ruang penyimpanan yang cukup luas karena arsip selalu bertambah, mudah mengalami kerusakan dan kehilangan dokumen serta pencarian kembali dokumen memerlukan waktu yang lama. Untuk mewujudkan tata kelola arsip yang baik, pemanfaatan teknologi informasi sangat penting dalam hal tersebut. Sesuai dengan perkembangan teknologi informasi salah satu upaya untuk melakukan pengarsipan adalah dengan memanfaatkan teknologi informasi yang diberi nama elektronik arsip atau E-Arsip. Arsip elektronik atau E-Arsip merupakan catatan yang dibuat atau disimpan dalam bentuk elektronik, baik analog atau digital. Arsip elektronik menurut NARA(National Archives and Record Administration) adalah arsip-arsip yang disimpan dan diolah dalam suatu *format* dimana hanya mesin komputer yang dapat memprosesnya. Oleh karena itu arsip elektronik seringkali dikatakan sebagai machine readable records(arsip yang hanya bisa dibaca melalui mesin) [3].

Dalam pengembangan sistem informasi e-arsip, metode pengembangan yang tepat perlu dipilih untuk memastikan kesuksesan proyek dan memenuhi kebutuhan pengguna. Salah satu metode pengembangan yang populer dan efektif adalah metode Scrum. Scrum adalah kerangka kerja ringan yang membantu orang, tim, dan organisasi untuk menghasilkan nilai melalui solusi adaptif. Dalam konteks pengembangan sistem informasi e-arsip, penggunaan metode Scrum memiliki beberapa manfaat. Pertama, metode ini memungkinkan pengembang untuk secara iteratif mengembangkan dan menguji fitur-fitur sistem, sehingga meminimalkan risiko kesalahan dan meningkatkan kualitas perangkat lunak. Kedua, Scrum mendorong komunikasi yang intensif antara tim pengembang dan pemangku kepentingan, sehingga memastikan kebutuhan pengguna terpenuhi secara akurat. Ketiga, Scrum memungkinkan fleksibilitas dalam perubahann kebutuhan atau prioritas, sehingga sistem informasi dapat disesuaikan dengan perubahan bisnis yang terjadi.

Program studi Teknik Informatika Universitas Lampung memiliki arsip dari hasil penelitian yang dilakukan oleh mahasiswa baik itu laporan akhir skripsi ataupun laporan praktik kerja lapangan. Arsip tersebut masih disimpan secara konvensional dan suatu saat akan memerlukan ruang penyimpanan yang luas karena arsip akan

selalu bertambah. Terlepas dari hal tersebut tidak dapat dipungkiri juga bahwa penyimpanan arsip secara konvensional juga mudah mengalami kerusakan serta pencarian dokumen yang semakin lama akan semakin memakan waktu bagi mahasiswa yang ingin meminjam baik laporan akhir skripsi maupun laporan kerja praktik. Berdasarkan masalah yang diuraikan tersebut, perlu dibuatnya suatu sistem informasi elektronik arsip untuk penyimpanan data-data laporan akhir skripsi dan laporan kerja praktik mahasiswa Teknik Informatika Universitas Lampung. Sejalan dengan hal ini telah ada beberapa website e-arsip yang telah banyak digunakan seperti eprints. Tetapi pada sistem informasi e-arsip dapat dimodifikasi sedemikian rupa dan dapat menambahkan fungsi-fungsi lainnya yang nantinya akan ditambahkan dimasa mendatang. Dengan dikembangkannya sistem ini diharapkan arsip dapat terjaga dengan baik dan mahasiswa dapat menjadikan laporan akhir skripsi maupun laporan kerja praktik sebagai sumber bacaan untuk penelitian-penelitian mahasiswa berikutnya.

1.2 Rumusan Masalah

Berdasarkan pada uraian latar belakang, rumusan masalah dalam penelitian ini yaitu bagaimana merancang suatu sistem informasi elektronik arsip untuk skripsi dan laporan kerja praktik di program studi Teknik Informatika Universitas Lampung dengan menggunakan metode Scrum.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah merancang dan membangun sistem informasi e-arsip menggunakan Framework Laravel yang dapat membantu civitas akademik di program studi Teknik Informatika Universitas Lampung.

1.4 Manfaat Penelitian

Manfaat yang didapat dari dilakukannya penelitian ini adalah sebagai berikut:

1. Membantu admin program studi Teknik Informatika dalam menyimpan data-data laporan akhir skripsi dan laporan kerja praktik.
2. Membantu meningkatkan efisiensi dan efektivitas proses pengembangan dengan metode Scrum. Metode Scrum yang mengedepankan kolaborasi,

transparansi, dan adaptasi akan memungkinkan tim pengembang untuk lebih responsif terhadap perubahan kebutuhan pengguna dan memastikan pengembangan sistem berjalan dengan efisien.

1.5 Batasan Masalah

Batasan masalah dilakukan agar penelitian lebih terarah dan memudahkan dalam pembahasan, sehingga tujuan dari penelitian dapat tercapai. Batasan masalah dalam pengembangan sistem ini yaitu:

1. Penelitian ini akan memfokuskan pada tahapan rancang bangun sistem informasi e-arsip, termasuk analisis kebutuhan, perancangan arsitektur, pengelolaan backlog, perencanaan *sprint*, dan evaluasi kualitas perangkat lunak.
2. Penelitian ini akan mempertimbangkan penggunaan teknologi yang sesuai dengan pengembangan sistem informasi e-arsip, seperti bahasa pemrograman, framework, database, dan tools yang mendukung implementasi Scrum.

1.6 Sistematika Penulisan Skripsi / tugas akhir

Sistematika penulisan pada penelitian ini adalah sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini secara umum meliputi latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian serta sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini berisi tentang prinsip, dasar teori dan teori penunjang yang digunakan sebagai acuan dalam melakukan penelitian

BAB III METODOLOGI PENELITIAN

Pada bab ini membahas mengenai metodologi penelitian yang digunakan dalam merancang sistem informasi e-arsip menggunakan Framework Laravel.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini membahas mengenai hasil dan pembahasan dari perancangan sistem informasi e-arsip menggunakan Framework Laravel.

BAB V KESIMPULAN DAN SARAN

Pada bab ini membahas mengenai kesimpulan dan saran dari penelitian.

DAFTAR PUSTAKA

LAMPIRAN

BAB II

TINJAUAN PUSTAKA

2.1 Arsip

Arsip adalah rekaman kegiatan atau peristiwa dalam berbagai bentuk dan media sesuai dengan perkembangan teknologi informasi dan komunikasi yang dibuat dan diterima oleh Lembaga negara, pemerintahan daerah, Lembaga Pendidikan, perusahaan, organisasi politik, organisasi kemasyarakatan, dan perseorangan dalam pelaksanaan kehidupan bermasyarakat, berbangsa dan bernegara [2].

2.2 Sistem

Sistem berasal dari bahasa Yunani, yaitu *systema*, yang artinya himpunan bagian atau komponen yang saling berhubungan secara teratur dan merupakan suatu keseluruhan. Sistem merupakan kumpulan dari beberapa bagian yang memiliki keterkaitan dan saling bekerja sama serta membentuk suatu kesatuan untuk mencapai tujuan dari sistem tersebut. Maksud dari suatu sistem adalah untuk mencapai tujuan dan sasaran dalam ruang lingkup yang sempit [4].

2.3 Informasi

Informasi atau dalam bahasa Inggrisnya adalah *information*, berasal dari kata *informacion* bahasa Prancis. Kata tersebut diambil dari bahasa Latin, yaitu "*informationem*" yang artinya "konsep, ide, garis besar". Informasi adalah suatu data yang sudah diolah atau diproses sehingga menjadi suatu bentuk yang memiliki arti bagi penerima informasi yang memiliki nilai bermanfaat [4].

2.4 Sistem Informasi

Sistem Informasi adalah suatu kombinasi modul yang terorganisir yang berasal dari komponen-komponen yang terkait dengan hardware, software, people dan network berdasarkan seperangkat komputer yang saling berhubungan atau berinteraksi untuk melakukan pengolahan data menjadi informasi untuk mencapai tujuan [5]. Tujuan dari sistem informasi diantaranya adalah suatu sistem informasi yang baik haruslah relevan agar dapat memperoleh hasil yang bagus dalam menentukan suatu keputusan dalam sebuah kelompok. Seluruh komponen sistem informasi dan data-data, harus memberikan hasil yang bermanfaat sekurang-kurangnya sebanyak pengeluaran yang telah dilakukan [1].

2.5 Arsip Elektronik

Arsip elektronik merupakan catatan yang dibuat atau disimpan dalam bentuk elektronik, baik analog atau digital. Arsip elektronik menurut NARA(National Archives and Record Administration) adalah arsip-arsip yang disimpan dan diolah dalam suatu *format* dimana hanya mesin komputer yang dapat memprosesnya. Oleh karena itu arsip elektronik seringkali dikatakan sebagai machine readable records(arsip yang hanya bisa dibaca melalui mesin) [3].

2.6 Website

Website merupakan kumpulan halaman digital yang berisi informasi berupa teks, animasi, gambar, suara dan video atau gabungan dari semuanya yang terkoneksi oleh internet, sehingga dapat dilihat oleh seluruh siapapun yang terkoneksi jaringan internet. *Website* terdiri dari beberapa kategori seperti, Web Statis, Web Dinamis dan Web Interaktif [6].

2.7 Framework

Framework merupakan komponen pemrograman yang siap re-use(bisa digunakan ulang) kapan saja, sehingga programmer tidak harus membuat skrip yang sama untuk tugas yang sama. Singkatnya framework merupakan wadah atau kerangka kerja dari suatu website yang akan dibangun, dengan menggunakan kerangka tersebut waktu yang digunakan dalam membuat website lebih singkat dan memudahkan dalam melakukan perbaikan [7].

2.8 PHP (PHP Hypertext Preprocessor)

PHP atau kependekatan dari *PHP Hypertext Preprocessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan web dan dapat ditanamkan pada sebuah skrip *HTML*. *PHP* merupakan bahasa *scripting server-side*, dimana pemrosesan datanya dilakukan pada sisi *server*. Sederhananya, *server*lah yang akan menerjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan. Sistem kerja dari *PHP* diawali dengan permintaan yang berasal dari halaman *website* oleh *browser*. Berdasarkan *URL* atau alamat website dalam jaringan internet, *browser* akan menemukan sebuah alamat dari *webserver*, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *webserver*. Kemudian *webserver* akan mencari berkas yang diminta dan menampilkannya di *browser* [8].

2.9 Laravel

Laravel adalah kerangka kerja web berbasis *php* untuk membangun aplikasi web kelas atas menggunakan sintaks yang signifikan dan anggun, dikembangkan oleh Taylor Otwell pada Juli 2011 dan dirilis lebih dari lima tahun setelah rilis

Codeigniter. Muncul dengan kumpulan alat yang kuat dan menyediakan arsitektur aplikasi. dan diperuntukkan untuk pengembangan aplikasi web yang menggunakan struktur pola *MVC*. Struktur pola *MVC* membagi *Laravel* menjadi tiga bagian yaitu model yang mengurus interaksi antar aplikasi dan *database*, *view* yang mengurus interaksi antara tampilan dengan *controller*, dan *controller* yang mengurus logika pemrograman. Berikut penjelasan dari Model, View dan Controller [7]:

a. Model

Model Mengelola basis data (*RDBMS*) seperti *MySQL* ataupun *Oracle RDMS*. Model berhubungan dengan databases sehingga biasanya dalam model akan berisi *class* ataupun fungsi untuk membuat(*create*), melakukan pembaruan(*update*), menghapus data(*delete*), mencari data (*search*), dan mengambil data(*select*), pada *database*.

b. View

View adalah bagian *user interface* atau bagian yang nantinya merupakan tampilan untuk *end user*. *View* bisa berupa halaman *html*, *css*, *rss*, *javascript jquery*, *ajax*, dan lain lain. Karena metode yang dipakai merupakan *MVC* sehingga dalam *view* tidak boleh terdapat pemrosesan data ataupun pengaksesan yang berhubungan dengan *database*. Sehingga *view* hanya menampilkan data-data hasil dari model dan *controller*.

c. Controller

Controller adalah penghubung antara *view* dan model, maksudnya ialah karena model tidak dapat berhubungan langsung dengan *view* ataupun sebaliknya jadi, *controller* inilah yang digunakan sebagai jembatan dikeduanya. Sehingga tugas *controller* ialah sebagai pemrosesan data atau alur *logic* program, menyediakan variabel yang akan ditampilkan di *view*, pemanggil model sehingga model dapat mengakses *databases*, *error handling*, validasi atau *check* terhadap suatu inputan [9].

2.10 Database

Database adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. *Database* atau basis data dapat dibayangkan ibarat sebuah lemari basis data. Secara prinsip, dalam suatu *database* tercakup dua komponen penting, yaitu Data dan Informasi. Jadi tujuan akhirnya adalah bagaimana mengelola data sehingga mampu menjadi informasi yang diinginkan dan dapat dilakukan proses pengambilan, penghapusan, pengeditan terhadap data secara mudah dan cepat (efektif, efisien dan akurat) [10].

2.11 MySQL

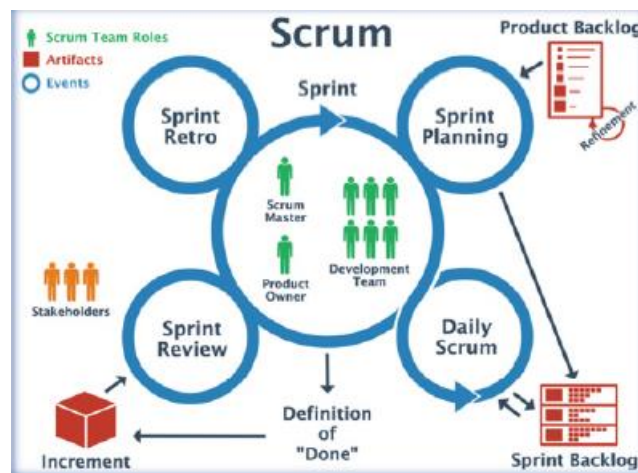
MySQL merupakan software database open source yang paling populer di dunia, dimana saat ini digunakan lebih dari 100 juta pengguna di seluruh dunia. Dengan kehandalan, kecepatan dan kemudahan penggunaan-nya, *MySQL* menjadi pilihan utama bagi banyak pengembang *software* dan aplikasi baik di platform web maupun *desktop*. Pengguna *MySQL* tidak hanya sebatas pengguna perseorangan maupun perusahaan kecil, namun perusahaan seperti *Yahoo!*, *Alcatel-Lucent*, *Google*, *Nokia*, *Youtube*, *Wordpress* dan *Facebook* juga merupakan pengguna *MySQL* [10]..

2.12 Scrum

Scrum adalah suatu metodologi atau kerangka kerja yang terstruktur untuk mendukung pengembangan produk yang kompleks. *Scrum* terdiri dari beberapa tim yang memiliki peran dan tugas masing-masing. Setiap komponen dalam kerangka melayani tujuan tertentu dan sangat penting untuk kesuksesan penggunaan *Scrum*. *SCRUM* adalah salah satu metode rekayasa perangkat lunak dengan menggunakan prinsip-prinsip pendekatan *AGILE*, yang bertumpu pada kekuatan kolaborasi tim, incremental product dan proses iterasi untuk mewujudkan hasil akhir. Proses pengembangan menggunakan metode *Scrum* terdapat lima tahapan pengembangan

yaitu: (1) *backlog refinement*, (2) *sprint planning*, (3) *daily meeting*, (4) *sprint reviews*, dan (5) *sprint retrospective*. Kelima proses pengembangan tersebut mengikuti tiga prinsip *Scrum* yaitu: *product owner (PO)*, *Scrum master (SM)*, dan *cross functional* [5].

Aktivitas-aktivitas wajib dalam *Scrum* dihadiri untuk menciptakan sebuah kesinambungan dan mengurangi adanya aktivitas-aktivitas lain yang tidak tercantum di dalam *Scrum*. Setiap acara di dalam *Scrum* memiliki batasan waktu, yang artinya selalu memiliki durasi maksimum. Acara-acara ini dirancang secara khusus untuk menciptakan transparansi dan peninjauan sampai ke tingkatan kritis. Tidak adanya pelaksanaan salah satu acara ini akan mengurangi transparansi dan menghilangkan kesempatan untuk meninjau dan membuat perubahan[5].



a. *Sprint*

Jantung dari *Scrum* adalah *sprint*, sebuah batasan waktu selama satu bulan atau kurang, di mana sebuah potongan produk yang “Selesai”, berfungsi, berpotensi untuk dirilis dikembangkan. *Sprint* biasanya memiliki durasi yang konsisten sepanjang proses pengembangan produk. *Sprint* yang baru, langsung dimulai setelah *sprint* yang sebelumnya berakhir. *Sprint* memuat dan terdiri dari *sprint planning*, *daily scrum*, pengembangan, *sprint review*, dan *sprint retrospective*.

b. *Sprint Planning*

Pekerjaan yang akan dilaksanakan di dalam *sprint* direncanakan pada saat *sprint planning*. Perencanaan ini dibuat secara kolaboratif oleh seluruh anggota tim. *Sprint Planning* dibatasi maksimum delapan jam untuk *sprint* yang berdurasi satu bulan. Untuk *sprint* yang lebih pendek, batasan waktunya biasanya lebih singkat. *Scrum Master* memastikan bahwa acara ini dilaksanakan dan setiap hadirin memahami tujuannya. *Scrum Master* mengedukasi tim untuk melaksanakannya dalam Batasan waktu yang telah ditentukan.

c. *Daily Scrum*

Merupakan kegiatan dengan batasan waktu maksimum selama 15 menit agar Tim Pengembang dapat mensinkronisasikan pekerjaan mereka dan membuat perencanaan untuk 24 jam kedepan. Hal ini dilakukan dengan meninjau pekerjaan semenjak acara *daily scrum* terakhir dan memperkirakan pekerjaan yang dapat dilakukan sebelum melakukan *daily scrum* berikutnya.

d. *Sprint Review*

Aktivitas yang diadakan di akhir *sprint* untuk meninjau potongan produk dan merubah product backlog bila diperlukan. Pada saat *sprint review*. Tim *Scrum* dan *stakeholder* berkolaborasi untuk membahas apa yang telah dikerjakan dalam *sprint* yang baru usai. Berdasarkan hasil tersebut dan semua perubahan *product backlog* pada saat *sprint*, para hadirin berkolaborasi menentukan apa yang dapat dikerjakan di *sprint* berikutnya untuk mengoptimalkan nilai produk. Pertemuan ini bersifat *informal*, bukan merupakan status *meeting*, dan presentasi dari potongan produk diharapkan dapat mengumpulkan masukan dan menumbuhkan semangat kolaborasi.

e. *Sprint Restrospective*

Sebuah kesempatan bagi tim untuk meninjau dirinya sendiri dan membuat perencanaan mengenai peningkatan yang akan dilakukan di *sprint* berikutnya. *Sprint Retrospective* dilaksanakan setelah *sprint review* selesai dan sebelum *sprint planning* berikutnya. Ini adalah acara dengan batasan waktu maksimum selama tiga jam untuk *sprint* yang berdurasi satu bulan. Untuk *sprint* yang lebih

pendek, batasan waktunya biasanya lebih singkat. *Scrum Master* memastikan bahwa acara ini dilaksanakan dan setiap hadirin memahami tujuannya. Scrum Master mengedukasi Tim Scrum untuk melaksanakannya dalam batasan waktu yang telah ditentukan. Scrum Master berpartisipasi sebagai rekan yang bertanggung-jawab terhadap proses scrum.

f. Product Backlog

Merupakan daftar terurut, dari setiap hal yang berkemungkinan dibutuhkan di dalam produk, dan juga merupakan sumber utama, dari daftar kebutuhan mengenai semua hal yang perlu dilakukan terhadap produk. Product Owner bertanggung-jawab terhadap product backlog, termasuk isinya, ketersediaannya, dan urutannya.

Discrum juga terdapat artefak scrum yang merepresentasikan pekerjaan atau nilai, bertujuan untuk menyediakan transparansi, dan kesempatan-kesempatan untuk peninjauan dan adaptasi. Artefak yang didefinisikan oleh scrum secara khusus dirancang untuk meningkatkan transparansi dari informasi kunci, dengan begitu semua pihak dapat memiliki pemahaman yang sama terhadap artefak. Berikut merupakan artefak scrum [5]:

a. Product Backlog.

Merupakan daftar terurut, dari setiap hal yang berkemungkinan dibutuhkan di dalam produk, dan juga merupakan sumber utama, dari daftar kebutuhan mengenai semua hal yang perlu dilakukan terhadap produk. Product Owner bertanggung-jawab terhadap product backlog, termasuk isinya, ketersediaannya, dan urutannya.

b. *Sprint* Backlog.

Merupakan sekumpulan item di dalam product backlog yang telah dipilih untuk dikerjakan di *sprint*, juga di dalamnya rencana untuk mengembangkan potongan tambahan produk dan merealisasikan *sprint* goal *Sprint* backlog adalah perkiraan mengenai fungsionalitas apa yang akan tersedia di Increment selanjutnya dan pekerjaan yang perlu dikerjakan

untuk menghantarkan fungsionalitas tersebut menjadi potongan tambahan produk yang “Selesai”.

c. Increment.

Increment (tambahan potongan produk) adalah gabungan dari semua item Product Backlog yang diselesaikan pada *Sprint* berjalan dan nilai-nilai dari increment *sprint-sprint* sebelumnya. Pada akhir *Sprint*, increment terbaru harus “Selesai” yang artinya berada dalam kondisi yang berfungsi penuh dan memenuhi definisi “Selesai” yang dibuat oleh tim. Terlepas apakah Product Owner akan merilis produknya, produk harus selalu berada dalam kondisi yang berfungsi penuh.

2.13 Blackbox Testing

Black-Box Testing merupakan Teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak. Blackbox Testing bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya difokuskan pada informasi domain. Blackbox Testing memungkinkan pengembang software untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Keuntungan penggunaan metode Blackbox Testing adalah (1) Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu; (2) Pengujian dilakukan dari sudut pandang pengguna, ini membantu untuk mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan; (3) Programmer dan tester keduanya saling bergantung satu sama lain [11].




2.14 Unified Modelling Language (UML)

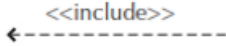
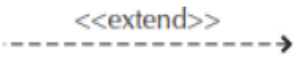
Unified Modelling Language (UML) adalah suatu “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan

sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem [12].

a. *Use case diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Berikut adalah elemen-elemen dari *use case diagram* [12]:

Nama	Simbol	Keterangan
Aktor		Actor/role adalah orang atau sistem lain yang berinteraksi dengan sistem saat ini
Use Case		Use case adalah bagian utama dari fungsionalitas sistem. Bisa extend (memperluas) use case lainnya. Ditempatkan di dalam system boundary (batasan sistem). Dilabeli dengan kata kerja – frase kata benda.
Association		Menghubungkan aktor dengan use case. Menunjukkan

		komunikasi dua arah (Menunjukkan komunikasi satu arah jika menggunakan tanda panah). Tanda * untuk keragaman dari asosiasi (multiplicity of the association). Namun umumnya hanya digambarkan garis saja.
<i>Include</i>		Memasukkan satu use case dalam use case lainnya. Perilaku (behavior) yang harus terpenuhi agar sebuah event dapat terjadi, di mana kondisi ini sebuah use case adalah bagian dari use case lainnya. Tanda panah mengarah dari base use case (pusat) menuju ke use case yang di-include.
<i>Extend</i>		Memperluas use case untuk memasukkan perilaku opsional. Tanda panah mengarah dari use

		case tambahan ke base use case (pusat).
--	--	---

b. *Activity Diagram*

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan state diagram khusus, di mana sebagian besar state adalah *action* dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum [12].

2.15 Penelitian Terdahulu

Penelitian Terdahulu merupakan penelitian yang telah dilakukan sebelumnya. Dalam penelitian ini tidak terlepas dari penelitian yang telah dilakukan sebelumnya dengan tujuan untuk mendukung penelitian ini. Berikut merupakan beberapa penelitian yang dijadikan referensi dalam penelitian ini.

2.15.1 Sistem Informasi E-Arsip Berbasis Web (Studi Kasus: Pt Haleyora Powerindo Cabang Sorong)

Penelitian oleh Tirsa Ninia Lina dan kawan-kawan yang berjudul Sistem Informasi E-Arsip Berbasis Web (Studi Kasus: Pt Haleyora Powerindo Cabang Sorong) pada tahun 2020, penelitian ini bertujuan untuk membuat suatu *website* tentang e-arsip dimana sebelumnya arsip masih dilakukan secara manual sehingga membuat pekerjaan menjadi tidak efektif dan efisien Ketika dihadapkan pada

kebutuhan informasi yang cepat. Metode penelitian yang digunakan pada penelitian ini yaitu wawancara, observasi dan studi pustaka kemudian Adapun metode pengembangan yang digunakan pada penelitian ini yaitu metode *R&D (Research and Development)* dimana metode ini memiliki beberapa tahap yaitu: *research and information collecting, planning, develop preliminary form of product, preliminary field testing, main product revision, dan main field testing*. Website dibangun dengan menggunakan bahasa pemrograman *PHP* dengan menggunakan *XAMPP* sehingga menghasilkan *website* yang sesuai dengan kebutuhan dalam pencarian data dan informasi arsip yang jadi lebih mudah dan akurat [13].

2.15.2 Rancang Bangun Aplikasi Penjualan Online Berbasis Web Menggunakan Metode Scrum

Penelitian oleh Aryanata Andi Pradana dan kawan-kawan yang berjudul Rancang Bangun Aplikasi Penjualan Online Berbasis Web Menggunakan Metode Scrum pada tahun 2021. Penelitian ini dilakukan untuk mengembangkan aplikasi penjualan online untuk UPTD SPNF SKB Salatiga dengan tujuan untuk menyediakan *platform* digital bagi UMKM untuk memasarkan produk mereka dan mengelola transaksi. Hal ini dilakukan karena pandemic COVID-19 telah berdampak pada usaha mikro kecil di Jawa Tengah, termasuk di Salatiga, sehingga dibutuhkan solusi digital untuk membantu UMKM memasarkan produk mereka. Penelitian ini menggunakan metode pengembangan sistem Scrum dengan tahapan yang ada hingga menghasilkan suatu aplikasi penjualan online yang utuh. Metode Scrum adalah salah satu metode rekayasa software dengan menggunakan prinsip-prinsip agile yang bertumpu pada kekuatan kolaborasi tim, incremental product, dan proses iterasi untuk mewujudkan hasil akhir. Dalam teknik Scrum terdapat tiga peran penting, yaitu Product Owner, Scrum Master, dan Tim Pengembangan. Metode ini dipilih karena fleksibilitasnya dalam beradaptasi dengan perubahan dan kemampuannya untuk menghasilkan produk berkualitas tinggi. Kesimpulan dari penelitian ini adalah metode pengembangan dengan menggunakan scrum membantu pemangku kepentingan dalam menyesuaikan perubahan kebutuhan

sistem Dengan sumber daya tim yang terbatas, scrum mampu menyelesaikan aplikasi dengan mengandalkan koordinasi antar anggota tim untuk mencapai tujuan dari setiap fitur-fitur yang dibutuhkan oleh pengguna dengan baik. Pemanfaatan sistem menggunakan basis web juga memudahkan *user* dalam menggunakan aplikasi [14].

2.15.3 Analisis Penerapan Metode Scrum Pada Pengembangan Sistem Informasi Akuntansi Koperasi

Penelitian oleh Alexander Rizaldi dan kawan-kawan yang berjudul Analisis Penerapan Metode Scrum Pada Pengembangan Sistem Informasi Akuntansi Koperasi pada tahun 2022, Penelitian ini tentang masalah pada operasional koperasi yang banyak ditemukan masalah seperti kurangnya informasi, tidak tercatatnya simpanan, jumlah pinjaman yang melebihi kapasitas, dan kekurangan telitian dalam penghitungan akuntansi. Metode pengembangan yang digunakan dalam penelitian ini adalah Scrum. Hasil yang didapatkan pada penelitian ini adalah berhasilnya dibangun sistem informasi akuntansi koperasi dengan menggunakan metode scrum. Metode scrum dapat mengatasi perubahan *requirement* pada fase pembuatan sistem. Scrum memiliki tahapan yang bersifat perulangan dimana jika fitur pada *sprint* pertama belum cukup memenuhi kebutuhan pengguna, maka pada *sprint* berikutnya dapat dikembangkan sistem yang sesuai dengan evaluasi pengguna sistem koperasi [15].

2.15.4 Analisis Perbandingan Metode Manajemen Proyek TI yang Paling Sering Digunakan Di Indonesia dan Luar Negeri: A Literature Review

Penelitian oleh Afifa Witania dan kawan-kawan yang berjudul Analisis Perbandingan Metode Manajemen Proyek TI yang Paling Sering Digunakan Di Indonesia dan Luar Negeri: A Literature Review pada tahun 2022, penelitian ini

bertujuan untuk mengetahui metode mana yang paling banyak digunakan oleh perusahaan dalam tata kelola manajemen proyek TI, penelitian ini juga bertujuan untuk metode mana yang paling efektif untuk digunakan. Metode yang digunakan pada penelitian ini adalah metode kualitatif dan masuk ke dalam jenis metode deskriptif dimana metode kualitatif adalah metode deskriptif penelitian yang lebih menekankan pada pengamatan sebuah fenomena dan meneliti substansi atau makna dari fenomena tersebut. Analisis pada penelitian ini dilakukan dengan *systematic literature review* (SLR). SLR adalah pendekatan *evidence-based* untuk mencari studi yang relevan dengan beberapa *research question* yang telah ditetapkan dengan memilih, menilai, dan membentuk temuan untuk menjawab *research question*. Adapun *research question* pada penelitian ini yaitu, 1. Metode apa yang paling sering digunakan oleh perusahaan, dalam tata kelola manajemen proyek TI baik perusahaan dalam negeri maupun luar negeri? 2. Bagaimana perbandingan dari metode yang digunakan oleh perusahaan, dalam tata kelola manajemen proyek TI baik perusahaan dalam negeri maupun luar negeri ?. Adapun metode yang dibandingkan terdiri dari 4 yaitu *Agile*, *Waterfall*, *Scrum* dan *Kanban*. Hasil dari penelitian ini yaitu metode manajemen proyek TI yang paling sering digunakan dan paling efektif di luar negeri adalah *agile* sedangkan didalam negeri adalah *scrum* hal itu berdasarkan dari hasil *Systematic Literature Review* dimana dari beberapa jurnal metode *waterfall* terbukti memiliki kekurangan dalam hal ketepatan dalam rencana yang membuatnya tidak fleksibel. sedangkan untuk *scrum* dikatakan efektif dan unggul pada beberapa kriteria seperti keefektifan waktu, fungsionalitas dan pengujian perangkat lunak, oleh karena itu metode *scrum* diimplementasikan pada pengembangan perangkat lunak suatu industri yang relative cepat, tidak memakan banyak waktu dan tidak banyak terintegrasi atau terlalu kompleks [16].

2.15.5 A Modern Review On Laravel – PHP Framework

Penelitian oleh Miss. Kausar I. Bagwan dan Prof. Swati D. Ghule yang berjudul A Modern Review On Laravel – PHP Framework pada tahun 2019, Penelitian ini membahas tentang keunggulan yang dimiliki oleh framework Laravel dalam

mengembangkan suatu website. Keunggulan tersebut yaitu: 1. Laravel menggunakan struktur MVC dimana struktur tersebut yang menyediakan pengembangan untuk membangun aplikasi berbasis web dengan aman dan tangguh. 2. Memiliki keamanan yang utuh dimana keamanan pada Laravel disediakan disisi client dan server sehingga pengembang tidak perlu mencari teknologi lain untuk membuat pengaman. Kesimpulan dari penelitian ini adalah kerangka kerja Laravel memberi solusi terbaik untuk mengembangkan aplikasi web kelas atas, tingkat kepuasan pengguna dan keamanan web dengan rentang waktu yang lebih singkat dan kode serta pengiriman produk yang lebih andal [17].

2.15.6 Blackbox and Whitebox Testing Techniques – A Literatur Review

Penelitian oleh Srinivas Nidra dan Jagruthi Dondeti yang berjudul Blackbox and Whitebox Testing Techniques – A Literatur Review, penelitian ini dilakukan studi tentang semua teknik pengujian yang terkait dengan blackbox dan whitebox testing. Blackbox testing disebut juga dengan pengujian fungsionalitas yaitu teknik pengujian fungsional dengan merancang test case berdasarkan informasi dari spesifikasi. Sedangkan whitebox testing disebut sebagai pengujian struktural yaitu teknik pengujian struktural yang merancang kasus uji berdasarkan informasi yang berasal dari source code. Hasil dari penelitian ini adalah pengujian blackbox dan whitebox sama-sama penting tergantung dari apa yang akan diuji, jika menguji kode maka dapat menggunakan whitebox testing seperti unit testing sedangkan jika menguji fungsionalitas dari suatu software maka dapat menggunakan blackbox testing [18].

2.15.7 Pengujian pada Aplikasi Penggajian Pegawai dengan Menggunakan Metode Blackbox Testing

Penelitian oleh Vadlan Febrian dan kawan-kawan yang berjudul Pengujian pada Aplikasi Penggajian Pegawai dengan Menggunakan Metode *Blackbox Testing* pada

tahun 2020. Penelitian ini membahas tentang pengujian pada aplikasi penggajian pegawai dengan menggunakan metode *blackbox testing*. Pengujian tersebut dilakukan karena jika terdapat *error* pada saat penggajian pegawai maka akan terjadi kerugian bagi pegawai dan perusahaan. Kerugian yang diterima oleh pegawai jika terjadi *error* pada program maka penurunan gaji akan mengalami keterlambatan. Dalam menyelesaikan masalah tersebut maka digunakan metode *blackbox testing*. Metode *blackbox testing* merupakan pengujian yang melihat hasil eksekusi melalui data uji dan memastikan fungsi dari *software*. Jenis teknik *blackbox testing* yang digunakan pada penelitian ini adalah teknik *equivalence partition* yaitu teknik pengujian dengan membagi menjadi dua kelas yaitu kelas valid dan kelas tidak valid. [11].

2.15.8 Pengembangan Aplikasi Tiga-Tingkat Menggunakan Metode Scrum pada Aplikasi Presensi Karyawan Glints Academy

Penelitian oleh Zidni Imam Solihati dan Imam Tahyudin tentang Pengembangan Aplikasi Tiga-Tingkat Menggunakan Metode Scrum pada Aplikasi Presensi Karyawan Glints Academy pada tahun 2022, Penelitian ini bertujuan untuk menerapkan metode scrum dalam lingkungan developer team yang heterogen dan latar belakang kemampuan divergen untuk membangun sebuah aplikasi dengan arsitektur tiga-tingkat. Aplikasi yang dibangun berupa aplikasi presensi karyawan dengan arsitektur tiga-tingkat yaitu sisi client, server, dan data. Sisi client aplikasi dalam bentuk front-end sedangkan sisi server dan sisi data berada dalam back-end server dengan framework Node.js dan Express.js. Autentikasi dengan JWT(JSON Web Token) menentukan hak akses terhadap fungsi dan sumber daya yang tersedia dalam back-end server. Metode yang digunakan pada penelitian ini adalah metode Scrum. Hasil yang diperoleh adalah sebuah aplikasi berbasis web yang telah memenuhi seluruh product backlog yang ditentukan oleh product owner dan metode yang digunakan mampu mengembangkan fitur ditengah proses pengembangan aplikasi tanpa mempengaruhi pengembangan fitur utama dan

metode ini efektif digunakan bagi lingkungan aplikasi yang pengembangannya berlangsung secara daring [19].

2.16 State of the Art

State of the Art langkah untuk mempresentasikan kebaruan dari hasil penelitian yang dilakukan. Berdasarkan masalah pada [13][20] sistem pengarsipan yang dilakukan secara elektronik memiliki banyak keunggulan seperti dapat meminimalkan kehilangan, dapat mencegah rusaknya dokumen-dokumen, efektif dan efisien dalam mencari dokumen yang diperlukan dalam waktu cepat. Berdasarkan pada [21] metode pengembangan *Scrum* lebih unggul dari metode pengembangan lainnya dikarenakan metode *scrum* sangat efektif diimplementasikan pada pengembangan perangkat lunak yang relatif cepat, tidak memakan waktu banyak dan tidak banyak terintegrasi atau tidak terlalu kompleks. Metode *scrum* pada dasarnya mengutamakan kecepatan dan fleksibilitas dalam proyek pengembangan perangkat lunak, selain itu metode *scrum* juga mengutamakan fungsionalitas yang didalamnya sudah terdapat hasil dari pengujian fungsionalitas yang mengakibatkan kualitas perangkat lunak dari segi fungsionalitas sudah terjamin. Pada penelitian [19][15] dijelaskan juga bahwasanya *scrum* adaptif dalam melakukan pekerjaan dimana *scrum* dapat mengerjakan atau menambahkan tugas-tugas baru ketika dalam proses pengembangan.

BAB III

METODOLOGI PENELITIAN

3.1 Waktu dan Tempat

Waktu dan tempat penelitian dilaksanakan pada:

- a. Waktu Penelitian : Februari sampai dengan Juni 2023
- b. Tempat Penelitian : Universitas Lampung

3.2 Jadwal Penelitian

Penelitian dilakukan lima bulan dengan jadwal penelitian sebagai berikut ini:

Tabel 3. 1 Jadwal Penelitian

Kegiatan	Bulan ke -				
	1	2	2	4	5
Analisis Masalah					
<i>Scrum</i>					
<i>Testing</i>					
Penulisan Laporan					

3.3 Alat Dalam Penelitian

Alat yang digunakan dalam rancang bangun sistem yang akan dibuat dalam penelitian adalah sebagai berikut:

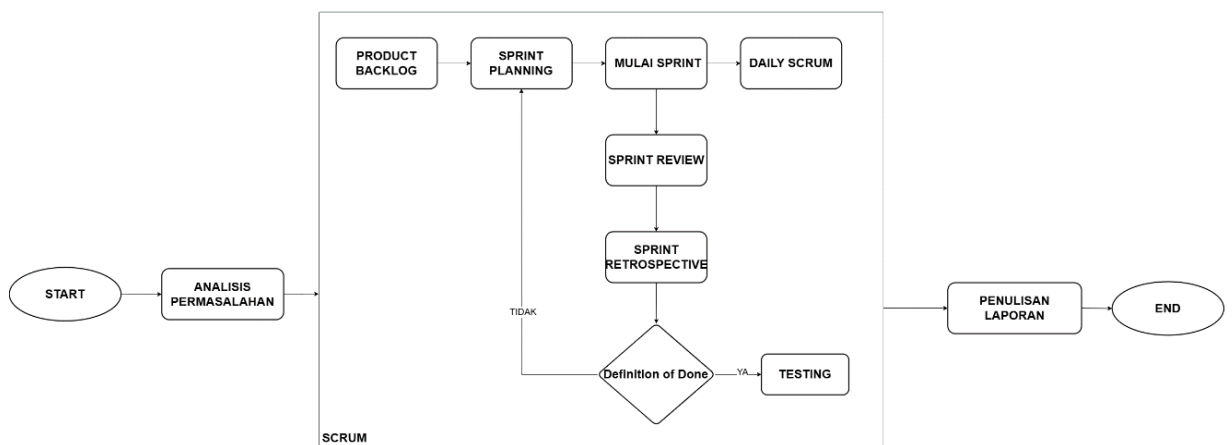
3.3.1 Alat Penelitian

Alat-alat yang digunakan dalam penelitian ini adalah sebagai berikut:

No	Nama Alat	Spesifikasi
1	<i>Laptop Acer Nitro 5 AN515-57</i>	<i>Intel@Coretm I5-11400H CPU@ 2.70 GHz, Ram 8 GB, Sistem Operasi Windows 11</i>
2	<i>Visual Studi Code</i>	Versi 1.75.1
3	<i>Laragon</i>	Versi 3.7.1
4	<i>Figma</i>	Online Figma
5	<i>PHP</i>	Versi 8.1.10
6	<i>Trello</i>	Online Trello
7	<i>Google Chrome</i>	Versi 110.0.1567.69
8	<i>Laravel</i>	Versi 10

3.4 Tahapan Penelitian

Terdapat beberapa tahapan yang dilakukan dalam penelitian ini, dapat dilihat pada Gambar dibawah ini.



Gambar 3. 1 Alur Penelitian

Gambar 3.1 menjelaskan tentang tahapan penelitian yang dimulai dengan melakukan Analisis permasalahan, pada tahap analisis masalah ini dilakukan

requirement gathering yaitu mengumpulkan informasi dan kebutuhan yang nantinya akan dilakukan di proses pengembangan sistem, kemudian dilakukan pengembangan sistem yang terdiri dari artefak *scrum* yaitu seperti *sprint backlog*, *sprint planning*, *sprint review*, dan *sprint retrospective*. Pada saat *Sprint Backlog* dilakukan penyusunan tugas-tugas yang nantinya akan dikerjakan selama pembuatan disistem. Setelah dikumpulkan semua tugas-tugas tersebut akan disusun suatu *Sprint Planning* yang dimana developer akan memecah tugas-tugas tersebut menjadi kesatuan yang lebih kecil yang akan di kembangkan di setiap *Sprint* nya. Kemudian setelah terciptanya *Sprint Planning* akan dilakukan *Sprint* sesuai dengan *Sprint Planning* yang telah ditetapkan dengan rentang waktu setiap *Sprint* nya ialah dua minggu, dalam melakukan *sprint* terdapat suatu proses yaitu *Daily Stand Up* dimana *developer* akan menyampaikan apa saja yang telah dikerjakan setiap harinya. Setelah selesai satu *Sprint* maka akan dilakukan *Sprint Review* di akhir *Sprint* untuk memeriksa hasil *Sprint* dan menentukan adaptasi selanjutnya. Kemudian terdapat *Sprint Retrospective* untuk meningkatkan kinerja *developer* dalam mengerjakan *project*. Kemudian tahap akhir yaitu testing dan penulisan laporan.

3.4.1 Analisis Permasalahan

Pada tahapan ini dilakukan analisis permasalahan pada sistem arsip di Program Studi Teknik Informatika Universitas Lampung. Saat ini arsip masih disimpan secara konvensional dan suatu saat akan memerlukan ruang penyimpanan yang luas karena arsip akan selalu bertambah. Terlepas dari hal tersebut tidak dapat dipungkiri juga bahwa penyimpanan arsip secara konvensional juga mudah mengalami kerusakan serta pencarian dokumen yang semakin lama akan semakin memakan waktu bagi mahasiswa yang ingin meminjam baik laporan akhir skripsi maupun laporan kerja praktik, dengan dibuatkannya sistem informasi e-arsip hal tersebut sangat efektif dalam mengurangi penggunaan kertas.

3.4.1.1 *User Story*

User Story digunakan sebagai acuan untuk membuat *Product Backlog*. *User Story* pada penelitian ini ditulis dalam *format* :

“Sebagai seorang [*Role*], Saya ingin [*Goal*], Sehingga [*Benefit*]

3.4.1.2 *Target Pengguna*

Adapun target pengguna dari website sistem informasi e-arsip ini adalah admin dan mahasiswa program studi Teknik Informatika Universitas Lampung.

3.4.1.3 *Kebutuhan Fungsional*

Kebutuhan fungsional pada sistem informasi e-arsip ini mencakup beberapa fitur seperti pengguna dapat melihat *list* mahasiswa, dosen dan arsip. Pengguna dengan *role* admin dapat menambahkan, menghapus, dan mengedit daftar mahasiswa, dosen, dan arsip.

3.4.1.4 *Kebutuhan Non-Fungsional*

Kebutuhan non-fungsional pada sistem informasi e-arsip ini mencakup *security* yang terdiri dari otorisasi yang digunakan untuk membagi otoritas atau hak dari setiap *role user* dan otentikasi yang digunakan untuk melakukan penyamaan identitas ketika melakukan *login* kedalam sistem.

3.4.2 *Product Backlog*

Setelah dilakukan analisis permasalahan pada sistem informasi e-arsip maka akan dibuatkan suatu *backlog* atau daftar fitur-fitur yang akan dikerjakan. *Backlog* tersebut akan dimasukkan kedalam *list* di *Trello* yang nantinya akan dikerjakan selama berlangsungnya suatu *sprint*.

3.4.3 *Sprint Planning*

Setelah didapatkan suatu *product backlog* dari sistem informasi e-arsip maka dibuatlah suatu *Sprint Planning* untuk merencanakan apa saja yang akan dikerjakan

selama satu *sprint* berlangsung. Kegiatan *Sprint Planning* sistem informasi e-arsip ini dilakukan setiap awal dari suatu *Sprint*. Pada pengembangan website sistem informasi e-arsip ini terdapat tiga *role* dari tim *scrum* yaitu :

Tabel 3. 2 *Role Scrum*

Role	Nama
<i>Scrum Master</i>	Dr. Eng. Mardiana, S.T.,M.T.
<i>Project Owner</i>	Rio Ariestia Pradipta, S.Kom, M.T.I.
<i>Developer</i>	Ahmad Ilham

Pada Tabel menjelaskan setiap *role* dari tim *scrum*, pertama terdapat Project Owner yang bertanggung jawab untuk memaksimalkan nilai produk yang dihasilkan dari kinerja tim *scrum*. Kemudian terdapat *Scrum Master* yang bertanggung jawab untuk menerapkan scrum sehingga dapat dilakukan *daily scrum*, *sprint review* dan *sprint retrospective* sehingga *scrum* dapat berjalan dengan lancar. Kemudian terakhir *developer* yang terdiri dari ahli profesi untuk menghantarkan *scrum* ke tahap selesai.

3.4.4 *Sprint*

Sprint pada sistem informasi e-arsip dilakukan selama empat *sprint* dengan rentang satu *sprint*nya adalah dua minggu setiap *sprint*. Pada tahap ini dilakukan pengerjaan berdasarkan *backlog* yang telah direncanakan pada tahapan *Sprint Planning*. Pada saat dilakukan *sprint* terdapat satu sesi yang dinamakan *Daily Scrum* pada pengembangan sistem informasi e-arsip dilakukan selama 10 menit. Hal ini dilakukan setiap hari selama *sprint* berlangsung untuk menyampaikan perkembangan dari proses pengembangan website sistem informasi e-arsip.

3.4.5 *Sprint Review*

Sprint Review pada pengembangan sistem informasi e-arsip ini dilakukan disetiap akhir *Sprint*. *Sprint review* ini dilakukan untuk meninjau tentang perkembangan pengembangan website sistem informasi e-arsip selama satu *sprint* berlangsung. Sehingga menghasilkan evaluasi untuk *Sprint planning* dan *Sprint* selanjutnya.

3.4.6 *Sprint Retrospective*

Sprint Retrospective pada pengembangan sistem informasi e-arsip dilakukan disetiap akhir dari *Sprint*. *Sprint Retrospective* dilakukan untuk merencanakan cara untuk meningkatkan kualitas dan efektivitas dari pengerjaan suatu *task* pada saat melakukan *Sprint*.

3.4.7 *Testing*

Tahapan testing dilakukan dengan menggunakan metode *black box testing* untuk menguji fungsionalitas dari fitur-fitur sistem informasi e-arsip. Teknik pengujian *black box testing* yang digunakan yaitu teknik *equivalance partitions*, dimana dengan teknik ini akan dibagi menjadi dua kelas ekuivalensi yaitu kelas valid dan tidak valid. Kelas valid dan tidak valid tersebut akan didapat dari *product backlog* yang telah dianalisis berdasarkan dari analisis permasalahan yang akan dilakukan.

3.4.8 *Penulisan Laporan*

Tahapan ini yaitu penulisan laporan. Tujuan dari penulisan laporan yaitu untuk memberikan dokumentasi terkait penelitian yang dilakukan, sehingga dapat dimanfaatkan oleh pembaca.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

Penelitian ini menghasilkan suatu website e-arsip dimana pengguna terdiri dari tiga *role* yaitu admin, mahasiswa dan dosen, setiap *role* dapat mengupload file skripsi ataupun laporan kerja praktek dan juga dapat melihat daftar dari skripsi, dan laporan kerja praktek.

4.2 Implementasi

4.2.1 Analisis Permasalahan

Pada tahap ini dilakukan analisis permasalahan tentang arsip di Teknik Informatika Universitas Lampung. Saat ini arsip masih dilakukan secara konvensional dan suatu saat akan memerlukan ruang penyimpanan yang luas karena arsip akan selalu bertambah. Terlepas dari hal tersebut tidak dapat dipungkiri juga bahwa penyimpanan arsip secara konvensional juga mudah mengalami kerusakan serta pencarian dokumen yang semakin lama akan semakin memakan waktu bagi mahasiswa yang ingin meminjam baik laporan akhir skripsi maupun laporan kerja praktik. Berdasarkan masalah yang diuraikan tersebut, perlu dibuatnya suatu sistem informasi elektronik arsip untuk penyimpanan data-data laporan akhir skripsi dan laporan kerja praktik mahasiswa Teknik Informatika Universitas Lampung. Dengan dikembangkannya sistem ini diharapkan arsip dapat terjaga dengan baik dan mahasiswa dapat menjadikan laporan akhir skripsi maupun laporan kerja praktik sebagai sumber bacaan untuk penelitian-penelitian mahasiswa berikutnya.

4.2.1.1 User Story

Adapun *User Story* yang didapat setelah dilakukan analisis permasalahan yaitu :

Tabel 4. 1 *User Story*

Saya Sebagai	Ingin...	Sehingga...
Admin	Adanya penyimpanan arsip secara elektronik	Pengarsipan menjadi lebih baik tanpa memenuhi ruangan
Admin	Arsip dapat terjaga dengan baik	Arsip tidak rusak seiring berjalannya waktu
Mahasiswa	Membaca hasil skripsi dan laporan kerja praktek	Dapat menjadi sumber bacaan untuk penelitian selanjutnya

4.2.1.2 Target Pengguna

Pada tahapan ini dilakukan analisis target pengguna yang akan menggunakan website e-arsip ini. Studi kasus pada penelitian ini dilakukan di Program Studi Teknik Informatika Universitas Lampung. Adapun target pengguna dari website ini yaitu sebagai berikut :

1. Admin Program Studi Teknik Informatik Universitas Lampung

Admin merupakan orang yang berkaitan dengan urusan administrasi dan operasional yang mendukung kelancaran dan efisiensi kegiatan pada suatu jurusan atau program studi

2. Mahasiswa Teknik Informatika Universitas Lampung

Mahasiswa merupakan seseorang yang tengah menimba ilmu atau belajar dan terdaftar pada salah satu bentuk perguruan tinggi, yang terdiri dari akademik, politeknik, sekolah tinggi, institusi hingga universitas.

4.2.1.3 Kebutuhan Fungsional

Adapun kebutuhan fungsional dari sistem e-arsip ini adalah sebagai berikut :

Tabel 4. 2 Kebutuhan Fungsional

No	ID	FITUR	DESKRIPSI
1.	SIE-01	List Mahasiswa	Website Menyediakan <i>list</i> dari mahasiswa Teknik Informatika Universitas Lampung
2.	SIE-02	List Dosen	Website Menyediakan <i>list</i> dari mahasiswa Teknik Informatika Universitas Lampung
3.	SIE-03	List Arsip	Website Menyediakan <i>list</i> dari arsip skripsi dan laporan kerja praktek mahasiswa Teknik Informatika Universitas Lampung
4.	SIE-04	Manajemen Pengguna	Admin dapat mengelola pengguna lainnya (Membuat, mengedit, dan menghapus pengguna)
5.	SIE-05	Penyimpanan Arsip	Pengguna dapat mengunggah dan menyimpan skripsi dan laporan kerja praktek
6.	SIE-06	Pencarian Arsip	Pengguna dapat mencari arsip berdasarkan nama, jenis, tanggal
7.	SIE-07	<i>Login</i>	Pengguna dapat melakukan <i>login</i> ke sistem
8.	SIE-08	<i>Logout</i>	Pengguna dapat melakukan <i>logout</i> dari sistem

9	SIE-09	Download Arsip	Pengguna dapat mendownload arsip yang telah diunggah ke sistem informasi e-arsip
10	SIE-10	Manajemen Arsip	Admin dapat mengelola arsip (Membuat, mengedit, dan menghapus pengguna)

4.2.1.4 Kebutuhan Non-Fungsional

Adapun kebutuhan non fungsional dari sistem e-arsip ini adalah sebagai berikut:

Tabel 4. 3 Kebutuhan Non-Fungsional

No	ID	FITUR	DESKIRPSI
1.	SIE-09	Otorisasi	Sistem memiliki otorisasi untuk memisahkan hak-hak setiap <i>role</i> dari sistem informasi e-arsip
2.	SIE-10	Otentikasi	Sistem memiliki otentikasi untuk verifikasi identitas ketika akan masuk ke sistem informasi e-arsip.

4.2.2 Product Backlog

Adapun product backlog yang didapat setelah dilakukan analisis permasalahan adalah sebagai berikut ini:

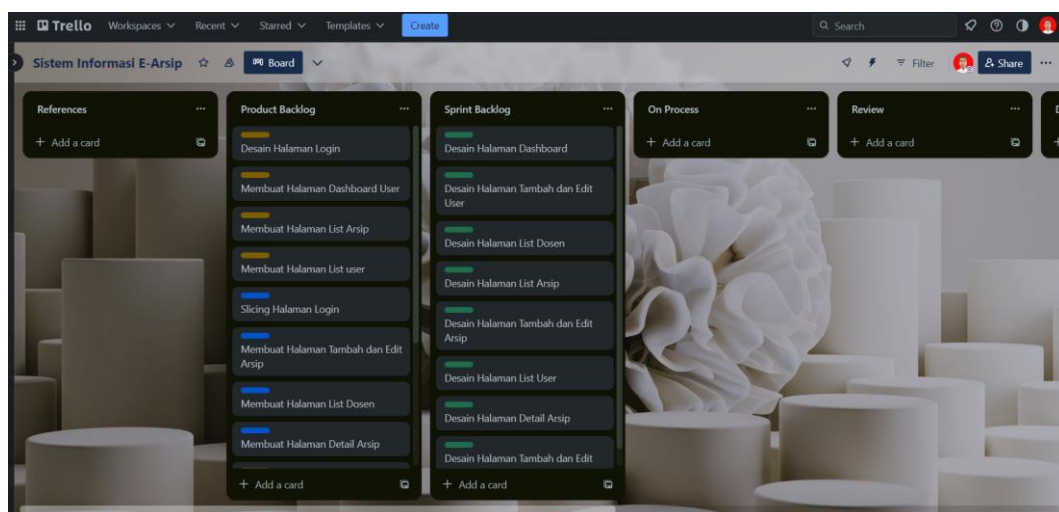
Tabel 4. 4 *List Product Backlog*

No	<i>Product Backlog</i>	Estimasi Waktu (Hari)	Prioritas
1.	Pembuatan <i>Use Case Diagram</i>	1	Tinggi

2.	Pembuatan <i>Activity</i> Diagram	3	Tinggi
3.	Pembuatan <i>Entity Relational</i> Diagram	1	Tinggi
4.	Desain Halaman <i>Dashboard</i>	1	Tinggi
5.	Desain Halaman <i>List</i> Arsip	1	Tinggi
6.	Desain Halaman <i>Login</i>	1	Sedang
7.	Desain Halaman <i>List User</i>	1	Tinggi
8.	Desain Halaman <i>List</i> Dosen	1	Sedang
9.	Desain Halaman tambah dan edit arsip	1	Tinggi
10.	Desain halaman tambah dan edit <i>user</i>	1	Tinggi
11.	Desain halaman tambah dan edit dosen	1	Sedang
12.	Membuat Halaman <i>Dashboard User</i>	2	Tinggi
13.	Membuat Halaman <i>List</i> Arsip	2	Tinggi
14.	<i>Slicing</i> Halaman <i>login</i>	1	Sedang
15.	Membuat Halaman <i>List user</i>	2	Tinggi
16.	Membuat Halaman <i>List</i> Dosen	1	Sedang
17.	Membuat Halaman Tambah dan Edit Arsip	3	Tinggi
18.	Membuat Halaman Tambah dan Edit <i>User</i>	1	Sedang
19.	Membuat Halaman Tambah dan Edit Dosen	1	Sedang

20.	Membuat Fitur <i>Login, Logout</i> dan Autentikasi	2	Sedang
21.	Desain Halaman Detail Arsip	2	Tinggi
22.	Membuat Halaman Detail Arsip	2	Tinggi

Tabel 4.4 merupakan daftar *product backlog* yang diambil berdasarkan dari analisis permasalahan, dimana didapatkan 22 *product backlog* dengan estimasi waktu total adalah 32 hari dengan tingkat prioritas dari sedang ke tinggi. *Product backlog* tersebut kemudian akan dimasukkan ke *software project manager* yaitu *trello* seperti pada gambar dibawah:



Gambar 4. 1 *Trello Product Backlog*

Gambar 4.17 merupakan *product backlog* yang telah disusun didalam *Trello*. *Product backlog* tersebut dibedakan berdasarkan warna di setiap *sprint*nya. Untuk warna hijau merupakan backlog yang dikerjakan di *sprint* 1. Untuk warna coklat digunakan untuk menandakan backlog yang akan dikerjakan di *sprint* 2. Untuk *sprint* 3 menggunakan warna biru dan *sprint* 4 menggunakan warna ungu.

4.2.3 *Sprint Planning*

Pada *Sprint Planning* telah dilakukan perencanaan untuk melakukan pengembangan website yang berdasarkan pada *product backlog*, estimasi waktu, dan prioritas dari tugas-tugas, rencana-rencana tersebut adalah sebagai berikut ini:

- a. Pada *Sprint Pertama* dilakukan perancangan sistem informasi yang berupa pemodelan sistem dan desain *interface* dari sistem informasi e-arsip.
- b. Pada *Sprint Kedua* dilakukan desain *interface* dan pembuatan halaman-halaman yang memiliki prioritas tinggi .
- c. Pada *Sprint ketiga* dilakukan pembuatan halaman-halaman yang belum dibuat di *sprint* sebelumnya.
- d. Pada *Sprint Keempat* dilakukan pembuatan desain *interface* dari *sprint* yang memiliki prioritas sedang dan pembuatan halaman-halaman serta *debuging*.

4.2.4 *Sprint*

Sprint seperti yang telah di jelaskan di bab sebelumnya dilakukan sebanyak empat *sprint* dengan jarak setiap *sprintnya* adalah dua minggu. *Sprint* dimulai pada tanggal 22 Mei 2023 sampai dengan 14 Juli 2023. *Sprint* dilaksanakan selama 8 jam dari rentang pukul 08.30 sampai dengan 16.30. Adapun pembagian *backlog* dari setiap *sprint* adalah sebagai berikut ini:

4.2.4.1 *Sprint 1*

Pada *Sprint* satu terdapat delapan *backlog* yang telah ditentukan berdasarkan estimasi waktu. Berikut *backlog* pada *sprint 1*:

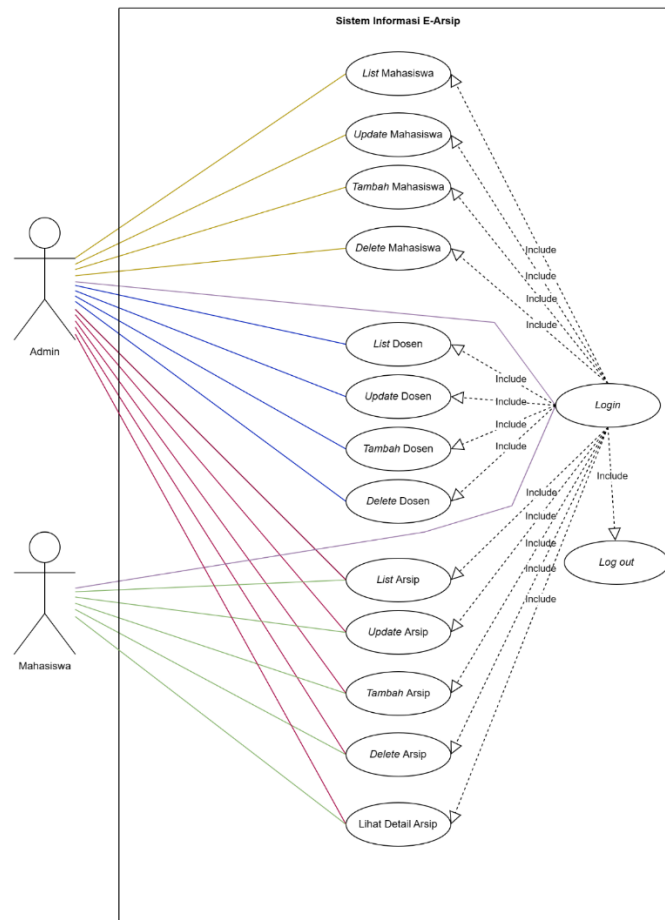
Tabel 4. 5 *Backlog Sprint 1*

No	<i>Backlog</i>	Waktu Pengerjaan
1	Pembuatan <i>Use Case</i> Diagram	22 Mei 2023 – 02 Juni 2023
2	Pembuatan <i>Activity</i> Diagram	
3	Pembuatan <i>Entity Relational</i> Diagram	
4	Desain Halaman <i>Dashboard</i>	
5	Desain Halaman <i>List</i> Arsip	
6	Desain Halaman tambah dan edit arsip	
7	Desain Halaman <i>List User</i>	

Pada tabel 4.5 merupakan *backlog* pada *sprint* 1. *Backlog-backlog* tersebut didasarkan pada *sprint planning* yang telah dibuat yaitu akan dilakukan pembuatan *design interface* dari sistem informasi e-arsip. *Design interface* dilakukan di *sprint* 1 karena agar memudahkan dalam proses pembuatan website yang dengan adanya visual maka akan diketahui perilaku dari website. Berikut merupakan hasil yang telah dicapai pada *sprint* 1:

1. Pembuatan *Use Case Diagram*

Berdasarkan kebutuhan fungsional dan *user story* didapatkan *use case* sebagai berikut:



Gambar 4. 2 *Use Case Diagram*

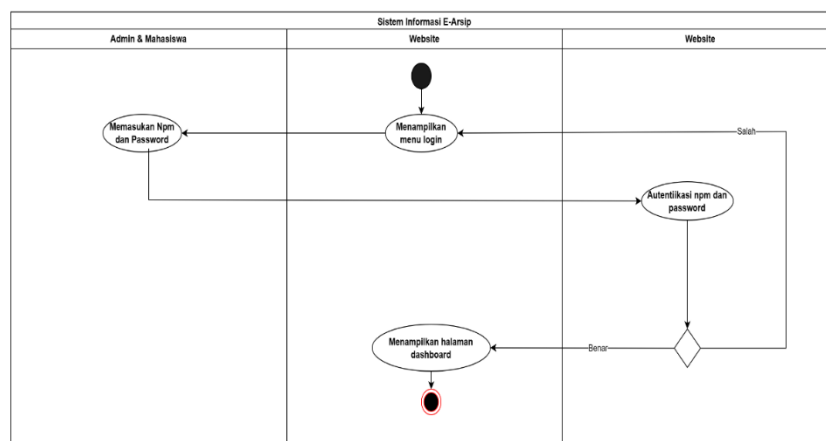
Gambar 4.2 merupakan *use case* diagram dari sistem informasi e-arsip yang terdiri dari dua *actor* yaitu Admin dan Mahasiswa. Admin dan Mahasiswa harus melakukan *login* terlebih dahulu untuk menggunakan fungsionalitas yang tersedia pada website sistem informasi e-arsip. Terdapat tiga fitur utama untuk admin pada sistem informasi e-arsip ini yaitu fitur mahasiswa yang terdiri dari *list* mahasiswa, tambah, edit, dan hapus mahasiswa. Kemudian terdapat fitur dosen untuk melihat, menambah, mengedit dan menghapus dosen serta terdapat fitur arsip yang terdiri dari menambah, mengedit, melihat dan menghapus arsip. Kemudian pada *actor* atau

role mahasiswa hanya memiliki fitur arsip yang terdiri dari *list*, *update*, *tambah* dan *delete* arsip. Tetapi pada arsip tersebut mahasiswa hanya dapat mengedit dan menghapus arsip yang ditambah oleh mahasiswa itu saja. Kemudian setiap *actor* memiliki *use case log out* jika telah melakukan *login* pada sistem informasi e-arsip. Untuk fitur dosen dan mahasiswa hanya dapat diakses oleh *actor* admin.

2. Pembuatan Activity Diagram

Berdasarkan dari *Use Case Diagram* maka diperoleh *Activity Diagram* sebagai berikut ini :

a. Activity Diagram Login Admin dan Mahasiswa



Gambar 4. 3 Activity Diagram Login Mahasiswa dan Admin

Gambar 4.3 merupakan *activity diagram login* dari mahasiswa dan admin, pada saat mengakses website admin atau mahasiswa akan langsung dialihkan ke halaman *login*, kemudian admin atau mahasiswa akan dimintai npm dan *password*, setelah itu data yang dimasukan oleh admin atau mahasiswa akan dilakukan autentikasi dengan data yang telah ada di database sehingga akan di cek kecocokan dari data tersebut, setelah data benar memiliki kecocokan maka akan langsung dialihkan ke halaman *dashboard*, sedangkan jika data yang dimasukan tidak cocok dengan data di database maka akan dikembalikan ke halaman *login* dengan pesan kesalahan, npm atau *password* tidak valid

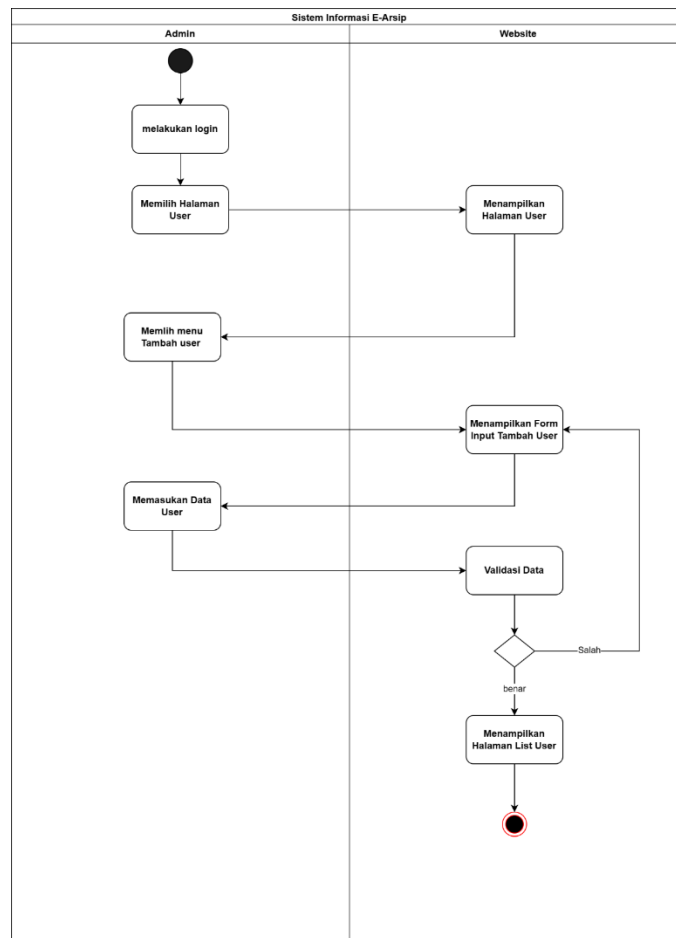
b. *Activity Diagram Logout Admin dan Mahasiswa*



Gambar 4. 4 *Activity Diagram Logout Admin dan Mahasiswa*

Gambar 4.4 merupakan *activity diagram logout* admin dan mahasiswa, pada saat admin atau mahasiswa berada di halaman *dashboard* dan memilih menu *logout* sistem akan memproses *logout* dengan menghentikan sesi autentikasi dari admin dan mahasiswa. Kemudian admin atau mahasiswa akan otomatis dialihkan ke halaman *login*.

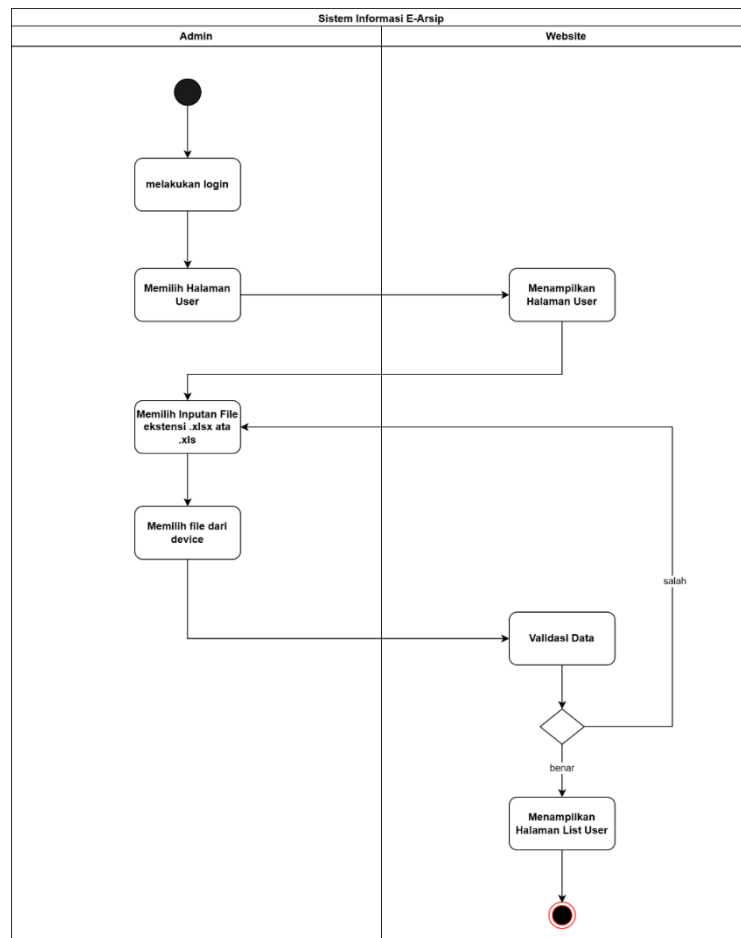
c. *Activity Diagram Menambah User*



Gambar 4. 5 *Activity Diagram Tambah User*

Gambar 4.5 merupakan *Activity Diagram* tambah *user*. *Activity diagram* tersebut dimulai ketika admin melakukan *login* ke sistem. Kemudian admin memilih Halaman *user* maka sistem akan menampilkan halaman *list user*, kemudian admin memilih menu tambah *user* dan sistem akan menampilkan *form* tambah *user* yang terdiri dari nama, npm, *roles*, dan *password* kemudian setelah data telah sesuai maka akan divalidasi oleh sistem apakah telah cocok dengan aturan atau kebijakan yang telah dibuat seperti panjang nama, panjang *password*. Jika data tidak valid maka akan menampilkan pesan kesalahan tergantung dari kesalahan inputan. Kemudian jika data valid maka akan dialihkan ke menu *list user* dengan pesan berhasil menambahkan data.

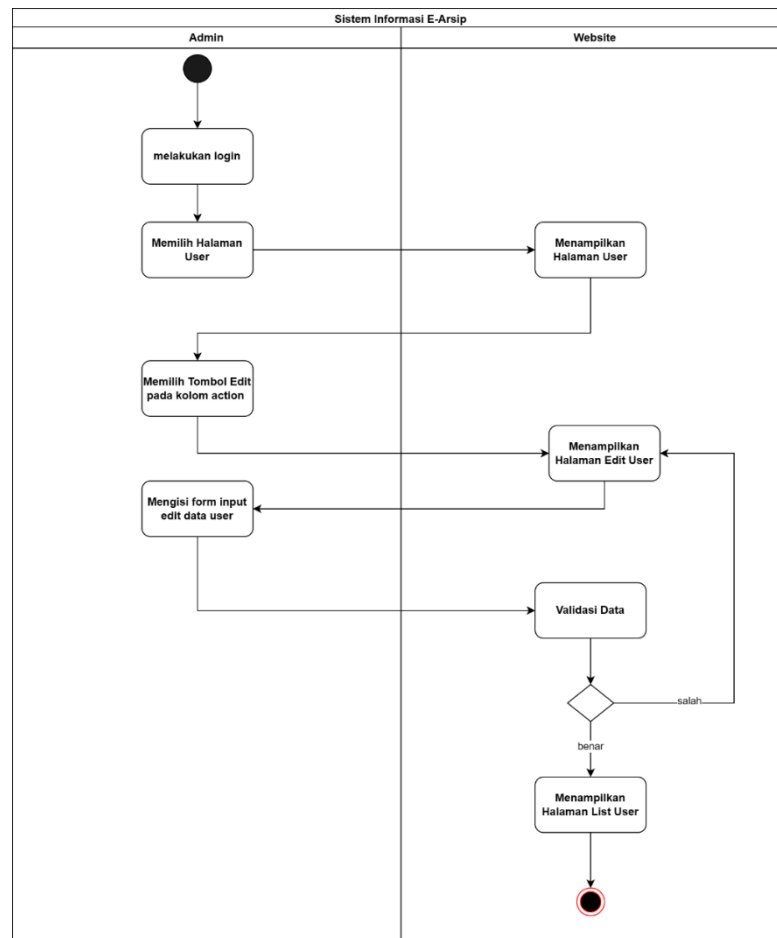
d. *Activity Diagram* tambah *user* dengan file ekstensi .xlsx atau .xls



Gambar 4. 6 *Activity Diagram* Menambah *User* dengan ekstensi .xlsx atau .xls

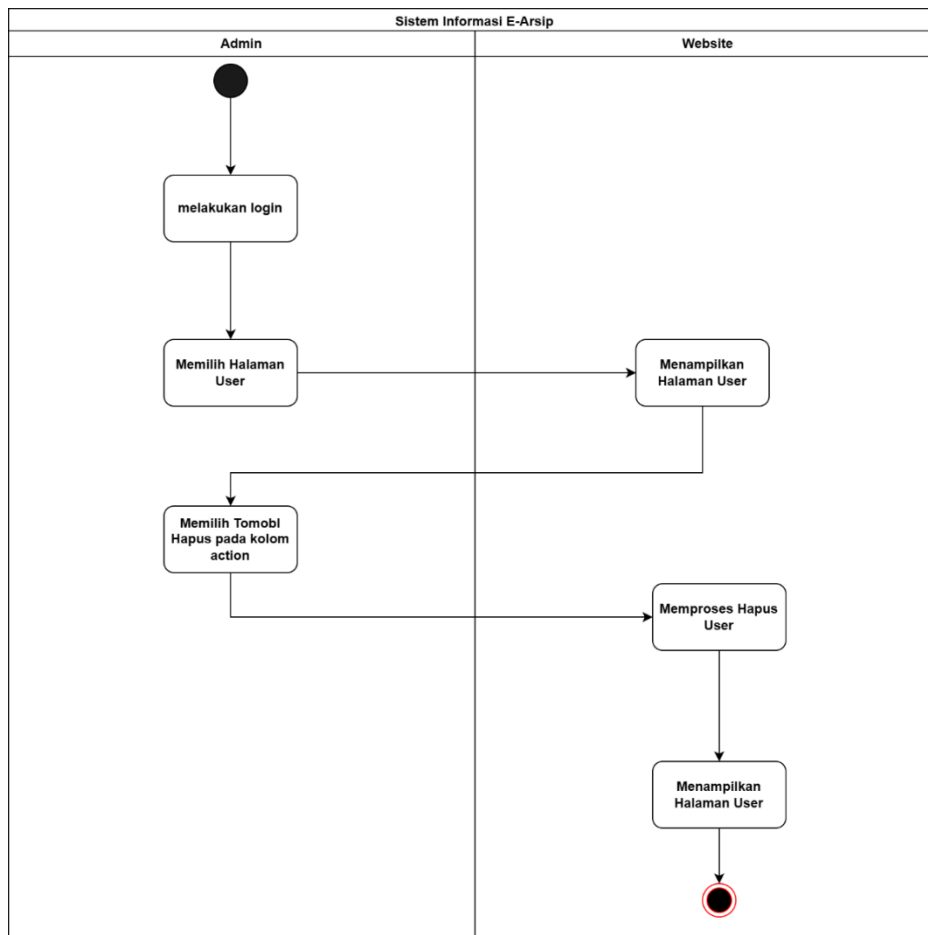
Gambar 4.6 merupakan *Activity Diagram* Menambah *User* dengan ekstensi .xlsx atau .xls. *Activity diagram* tersebut mulai ketika admin melakukan *login* ke sistem, kemudian admin memilih menu halaman *user*. Kemudian sistem akan menampilkan halaman *list user* dan terdapat input untuk menambahkan *user* dengan file berekstensi .xlsx atau .xls. Kemudian admin memilih input *user* dan memilih file dari device. Setelah itu dilakukan validasi data jika validasi belum benar maka akan dikembalikan ke halaman *list user* dan jika validasi benar maka akan menampilkan halaman *list user* dengan pesan sukses.

e. *Activity Diagram Mengedit User Oleh Admin*



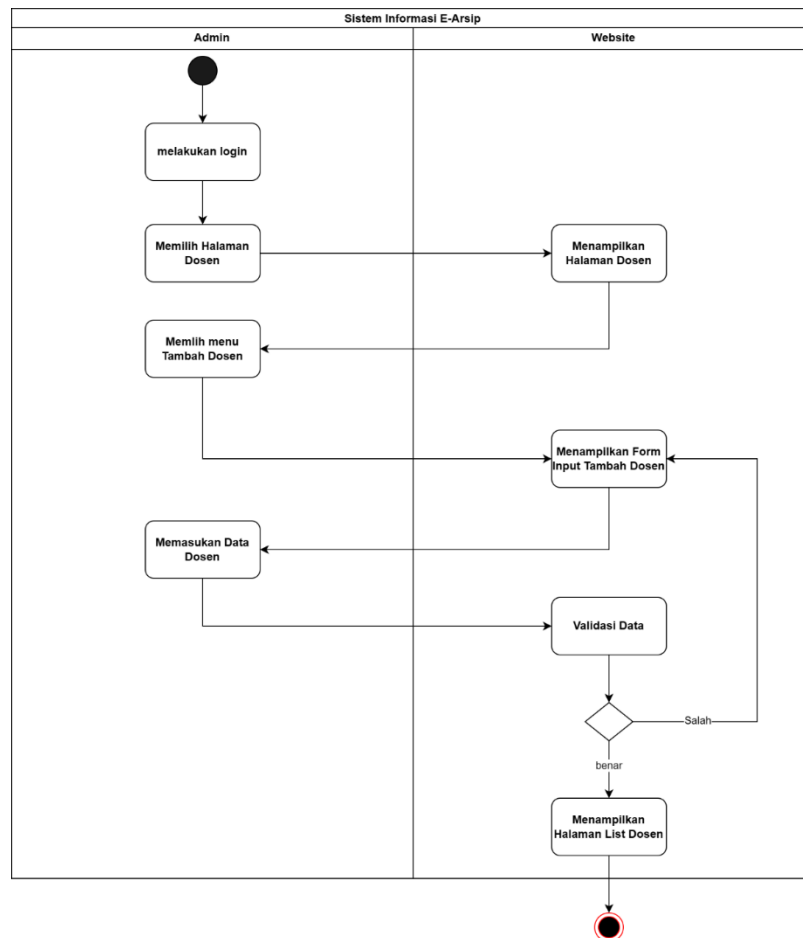
Gambar 4. 7 *Activity Diagram Edit User Oleh Admin*

Gambar 4.7 merupakan *Activity Diagram Edit User* oleh Admin. *Activity* diagram dimulai dari admin melakukan *login* ke sistem. Kemudian admin memilih halaman *user* kemudian akan ditampilkan *list user*. Setelah itu admin akan memilih tombol edit yang berada di kolom *action*. Kemudian halaman akan dialihkan ke halaman edit *user*. Admin akan mengedit data yang terdiri dari nama, *npm*, *roles* dan *password*. Setelah itu akan dilakukan validasi oleh sistem apakah telah cocok dengan kebijakan yang telah dibuat. Jika belum cocok maka akan dialihkan kembali ke menu *form* edit data. Jika telah cocok maka akan kembali ke halaman *list user* dengan pesan sukses.

f. *Activity Diagram Menghapus User*Gambar 4. 8 *Activity Diagram Menghapus User*

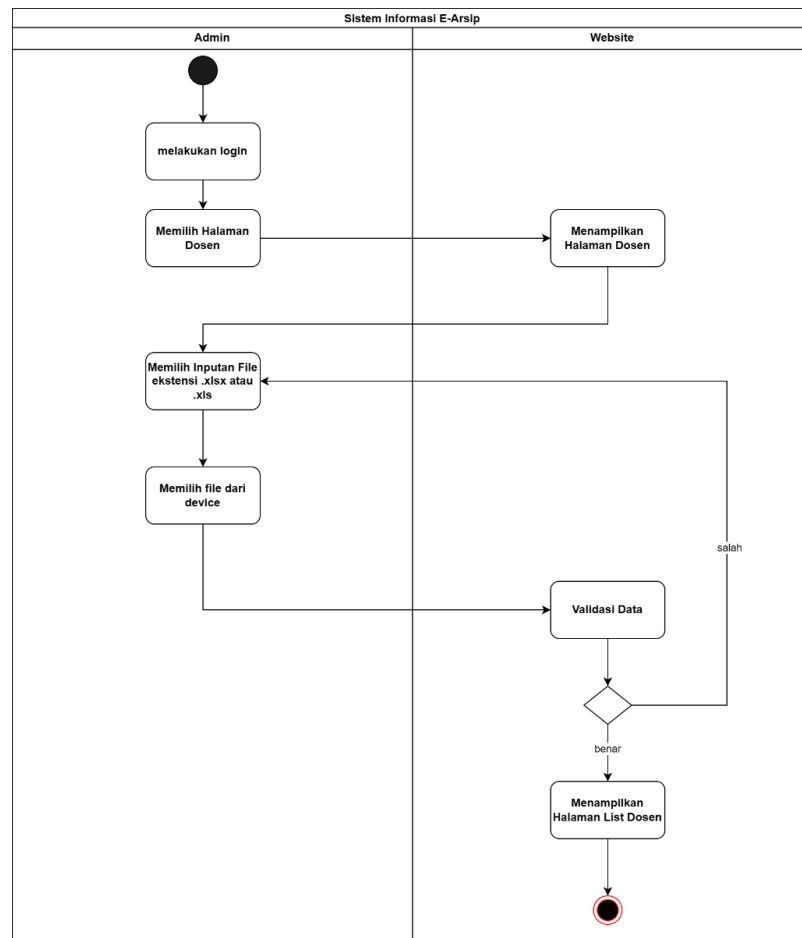
Gambar 4.8 merupakan *Activity Diagram* menghapus *user*. *Activity diagram* dimulai dari admin melakukan *login* kemudian memilih halaman *user* kemudian sistem akan menampilkan halaman *list user*. Setelah itu admin akan memilih tombol hapus pada kolom *action* kemudian sistem akan memproses hapus *user* dan akan tetap di halaman *list user* dengan pesan sukses.

g. Activity Diagram Menambah Dosen



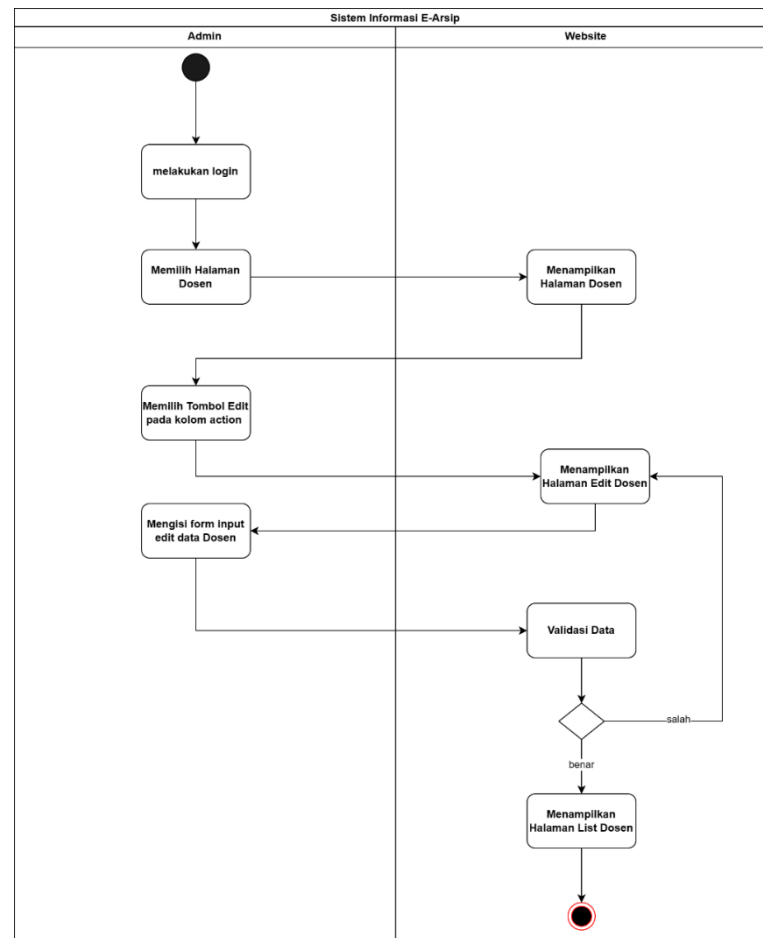
Gambar 4. 9 Activity Diagram menambah dosen

Gambar 4.9 merupakan Activity Diagram tambah dosen. Activity diagram tersebut dimulai ketika admin melakukan *login* ke sistem kemudian memilih Halaman dosen maka sistem akan menampilkan halaman *list* dosen, kemudian admin memilih menu tambah dosen dan sistem akan menampilkan *form* tambah dosen yang terdiri dari nama, nip. Kemudian setelah data telah sesuai maka akan divalidasi oleh sistem apakah telah cocok dengan aturan atau kebijakan yang telah dibuat seperti panjang nama, panjang *password*. Jika data tidak valid maka akan menampilkan pesan kesalahan tergantung dari kesalahan inputan. Kemudian jika data valid maka akan dialihkan ke menu *list* dosen dengan pesan berhasil menambahkan data.

h. *Activity Diagram Menambah dosen dengan file ekstensi .xlsx atau .xls*Gambar 4. 10 *Activity Diagram Menambah Dosen dengan ekstensi .xlsx atau .xls*

Gambar 4.10 merupakan *Activity Diagram Menambah Dosen dengan ekstensi .xlsx atau .xls*. *Activity diagram* tersebut mulai ketika admin melakukan *login* ke sistem kemudian memilih menu halaman dosen. Kemudian sistem akan menampilkan halaman *list* dosen dan terdapat input untuk menambahkan *user* dengan file berekstensi .xlsx atau .xls. Kemudian admin memilih input dosen dan memilih file dari device. Setelah itu dilakukan validasi data jika validasi belum benar maka akan dikembalikan ke halaman *list user* dan jika validasi benar maka akan menampilkan halaman *list* dosen dengan pesan sukses.

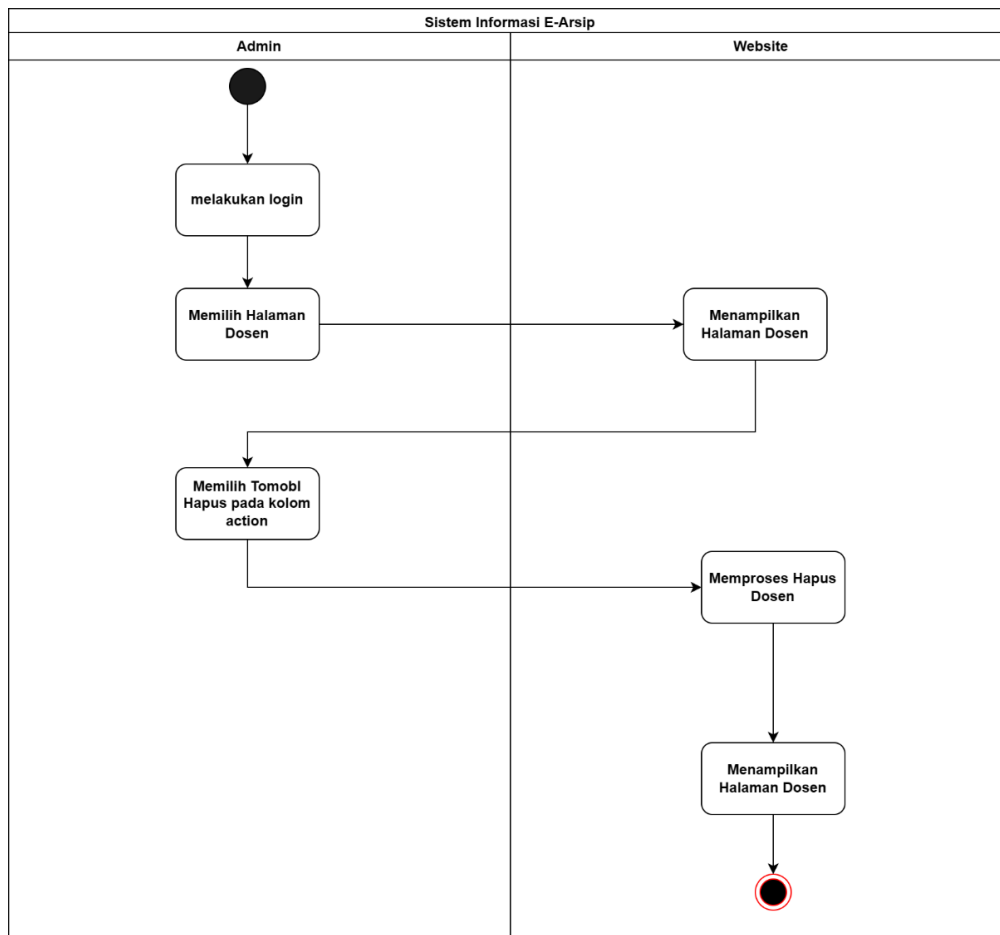
i. *Activity Diagram* mengedit Dosen



Gambar 4. 11 *Activity Diagram* mengedit Dosen

Gambar 4.11 merupakan *Activity Diagram* Edit Dosen. *Activity* diagram dimulai dari admin melakukan *login* ke sistem kemudian memilih halaman dosen kemudian akan ditampilkan *list* dosen. Setelah itu admin akan memilih tombol edit yang berada di kolom *action*. Kemudian halaman akan dialihkan ke halaman edit dosen. Admin akan mengedit data yang terdiri dari nama dan nip. Setelah itu akan dilakukan validasi oleh sistem apakah telah cocok dengan kebijakan yang telah dibuat. Jika belum cocok maka akan dialihkan kembali ke menu *form* edit data. Jika telah cocok maka akan kembali ke halaman *list* dosen dengan pesan sukses.

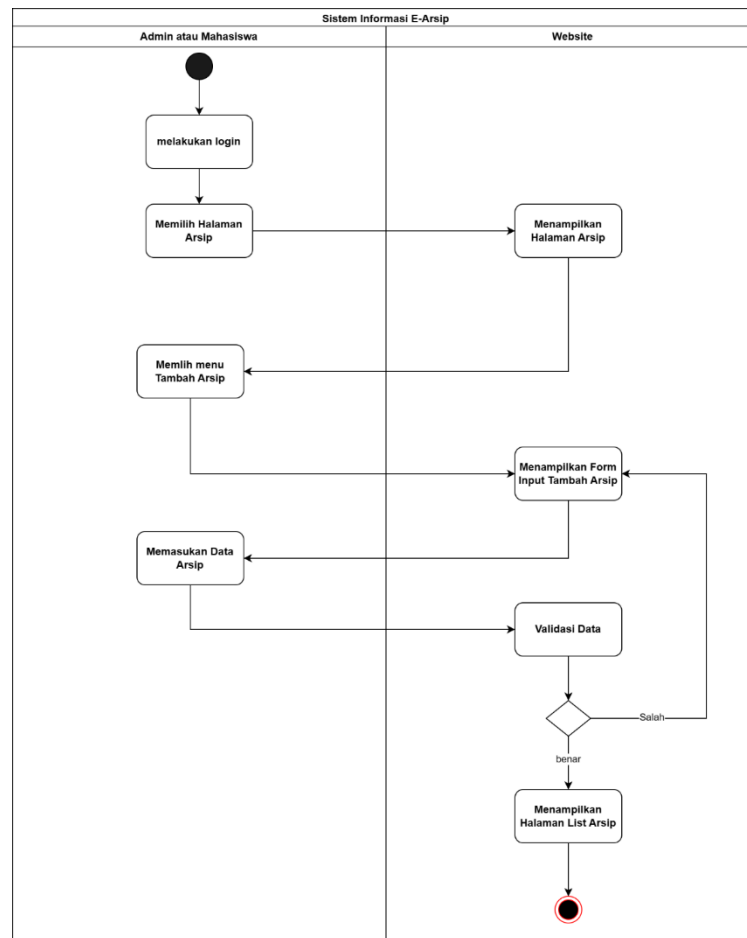
j. Activity Diagram Menghapus Dosen



Gambar 4. 12 Activity Diagram Menghapus Dosen

Gambar 4.12 merupakan Activity Diagram menghapus dosen. Activity diagram dimulai dari admin melakukan *login* kemudian memilih halaman dosen kemudian sistem akan menampilkan halaman *list* dosen. Setelah itu admin akan memilih tombol hapus pada kolom *action* kemudian sistem akan memproses hapus dosen dan akan tetap di halaman *list* dosen dengan pesan sukses.

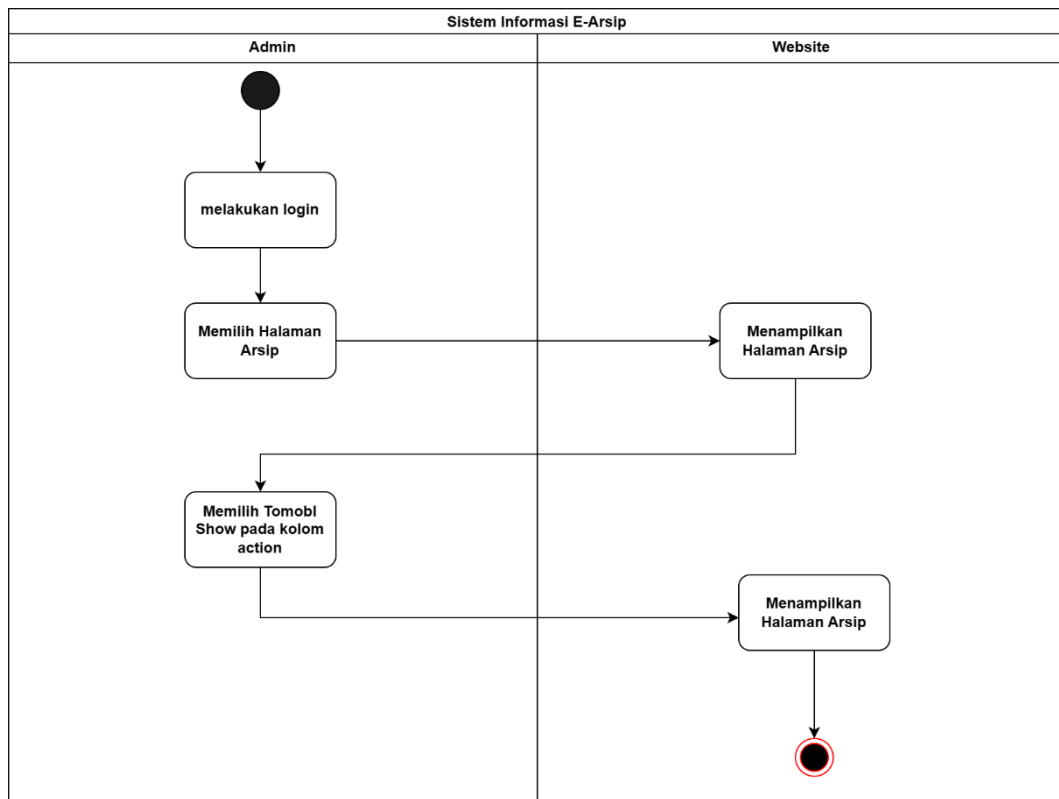
k. *Activity Diagram Menambah Arsip oleh admin dan mahasiswa*



Gambar 4. 13 *Activity Diagram Menambah Arsip oleh admin dan mahasiswa*

Gambar 4.13 merupakan *Activity Diagram* tambah arsip. *Activity diagram* tersebut dimulai ketika admin atau mahasiswa melakukan *login* ke sistem kemudian memilih Halaman arsip maka sistem akan menampilkan halaman *list* arsip, kemudian admin atau mahasiswa memilih menu tambah arsip dan sistem akan menampilkan *form* tambah arsip yang terdiri dari judul, ruang, jenis laporan, dosen pembimbing 1, dosen pembimbing 2, dosen pembahas, dokumen, dan tanggal seminar. Kemudian setelah data telah sesuai maka akan divalidasi oleh sistem apakah telah cocok dengan aturan atau kebijakan yang telah dibuat. Jika data tidak valid maka akan menampilkan pesan kesalahan. Kemudian jika data valid maka akan dialihkan ke menu *list* arsip dengan pesan berhasil menambahkan data.

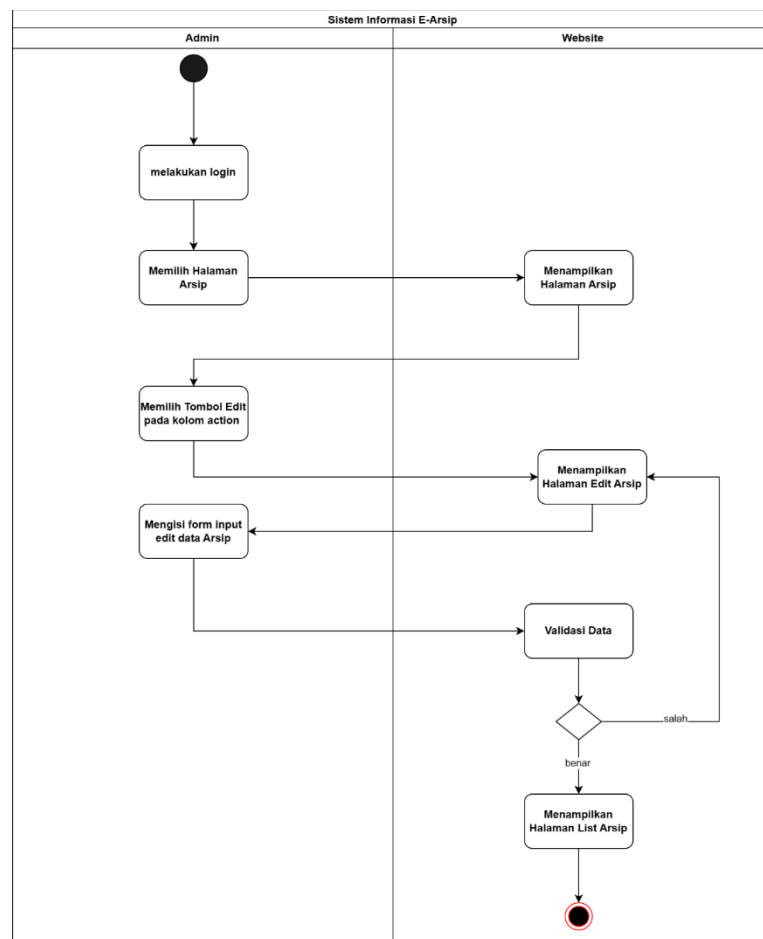
1. Activity Diagram Menampilkan arsip tertentu



Gambar 4. 14 Activity Diagram Menampilkan arsip tertentu

Gambar 4.14 merupakan Activity Diagram Menampilkan arsip tertentu arsip. Activity diagram tersebut dimulai ketika admin atau mahasiswa melakukan *login* ke sistem kemudian memilih Halaman arsip maka sistem akan menampilkan halaman *list* arsip, kemudian admin atau mahasiswa memilih menu *show* arsip dan sistem akan menampilkan data detail tentang arsip yang terdiri dari.

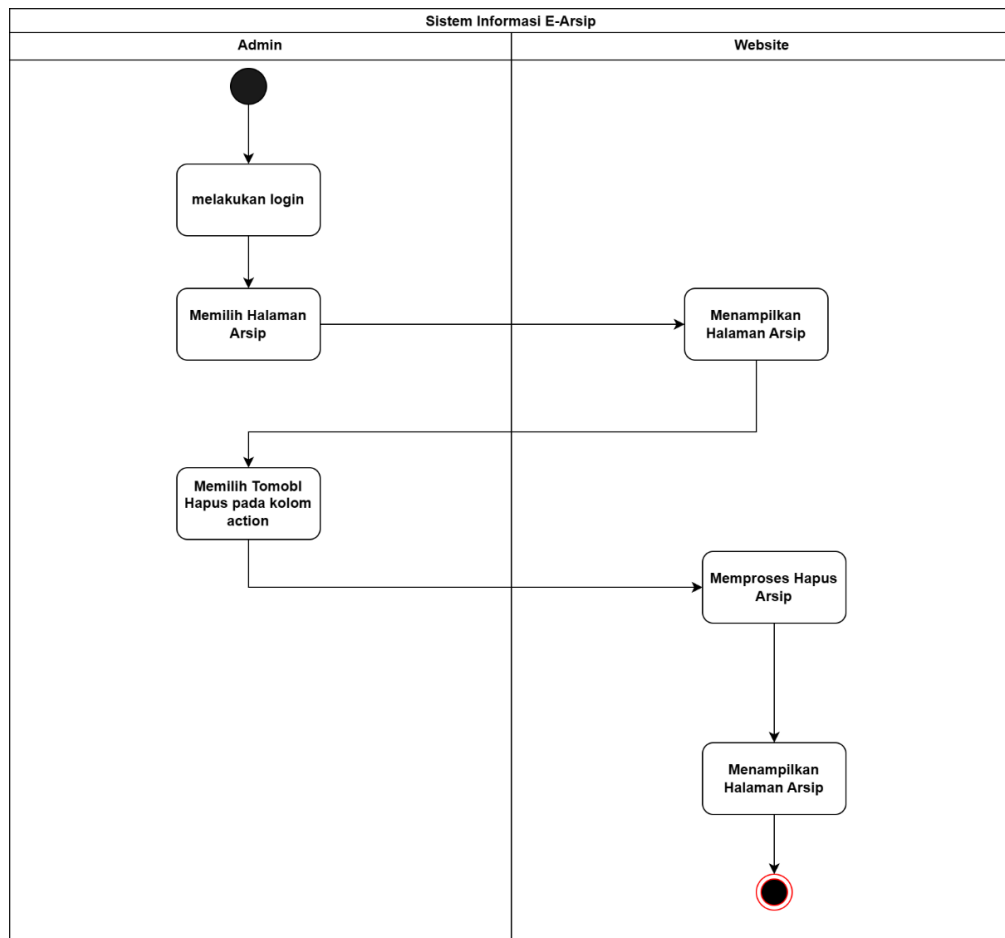
m. Activity Diagram Mengedit Arsip



Gambar 4. 15 Activity Diagram Mengedit Arsip

Gambar 4.15 merupakan Activity Diagram Edit Arsip. Activity diagram dimulai dari admin melakukan *login* ke sistem kemudian memilih halaman arsip kemudian akan ditampilkan *list* arsip. Setelah itu admin akan memilih tombol edit yang berada di kolom *action*. Kemudian halaman akan dialihkan ke halaman edit arsip. Admin akan mengedit data yang terdiri dari judul, ruang, jenis laporan, dosen pembimbing 1, dosen pembimbing 2, dosen pembahas, dokumen, tanggal seminar dan status arsip. Setelah itu akan dilakukan validasi oleh sistem apakah telah cocok dengan kebijakan yang telah dibuat. Jika belum cocok maka akan dialihkan kembali ke menu *form* edit data. Jika telah cocok maka akan kembali ke halaman *list* arsip dengan pesan sukses.

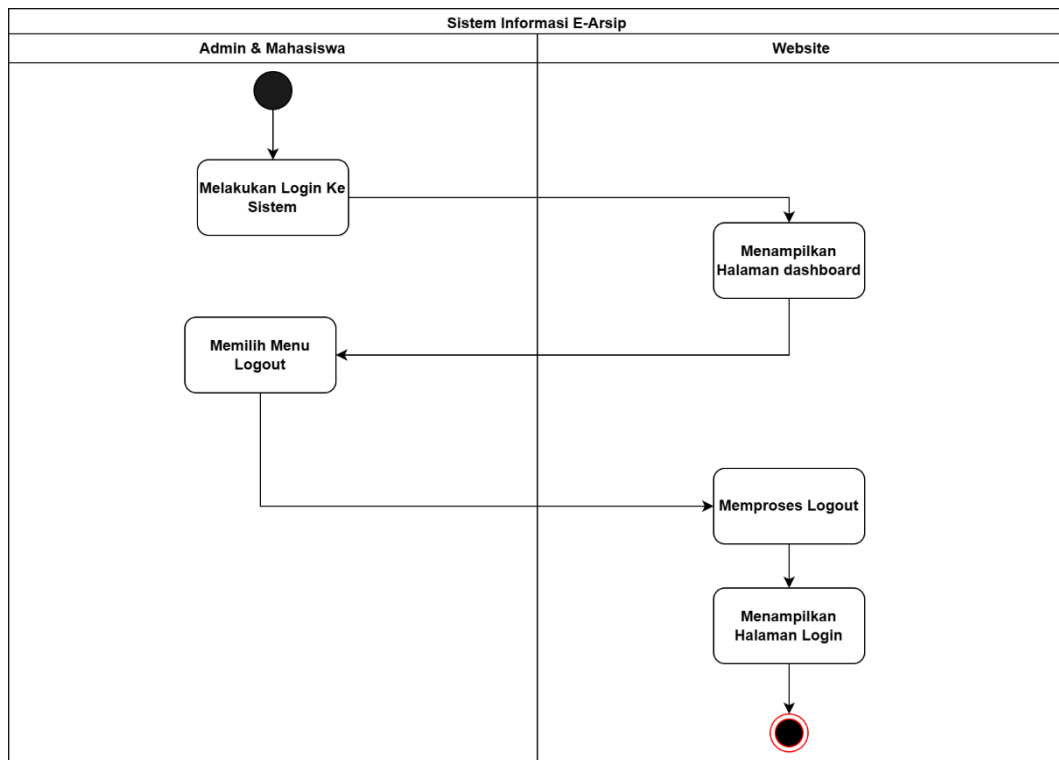
n. Activity Diagram Menghapus arsip



Gambar 4. 16 Activity Diagram Menghapus arsip

Gambar 4.16 merupakan *Activity Diagram* menghapus arsip. *Activity diagram* dimulai dari admin melakukan *login* kemudian memilih halaman arsip kemudian sistem akan menampilkan halaman *list* arsip. Setelah itu admin akan memilih tombol hapus pada kolom *action* kemudian sistem akan memproses hapus arsip dan akan tetap di halaman *list* arsip dengan pesan sukses.

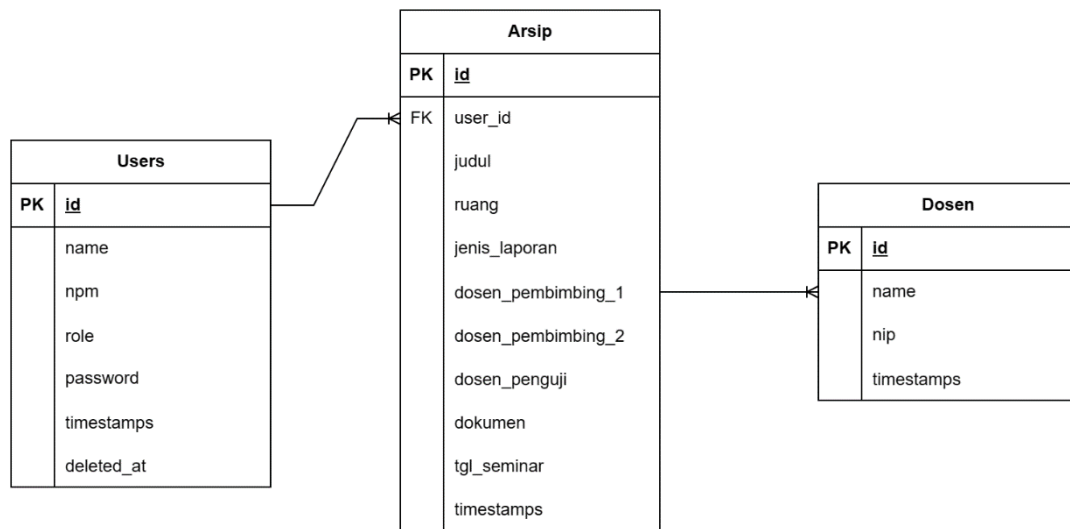
o. *Activity Diagram Logout*



Gambar 4. 17 *Activity Diagram Logout*

Gambar 4.17 merupakan *Activity Diagram Logout* untuk admin dan mahasiswa, *activity diagram* dimulai dari admin atau mahasiswa melakukan *login* ke sistem, kemudian akan masuk ke halaman *dashboard*. Setelah itu admin atau mahasiswa memilih menu *logout* dan sistem akan memproses *logout* setelah berhasil *logout* maka akan dialihkan ke halaman *login*.

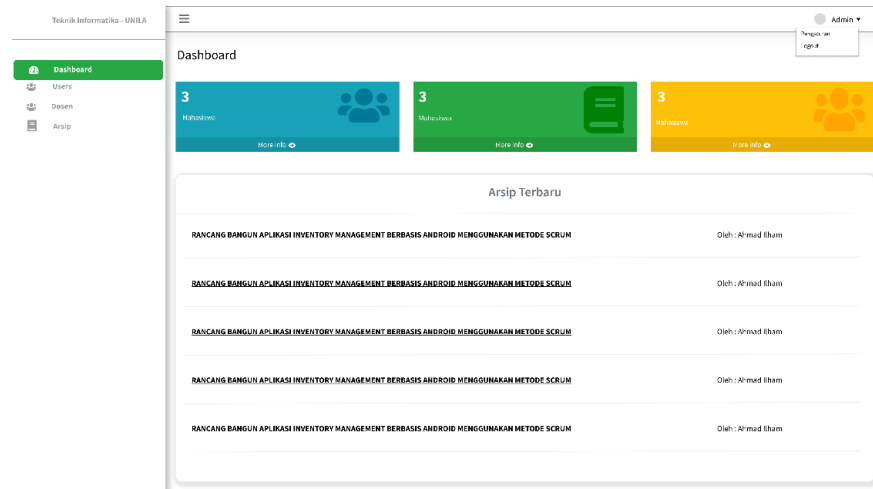
3. Membuat *Entity Relational Database*



Gambar 4. 18 *Entity Relational Diagram*

Gambar 4.18 merupakan entity relational diagram. Pada sistem informasi e-arsip memiliki tiga entitas yaitu *users*, *Arsip* dan *Dosen*. Pada Entitas *Users* terdapat attribute *id* yang merupakan primary key, *name* yang memiliki tipe data string, *npm*, *role* yang terdiri dari admin dan mahasiswa, *timestamps* yang terdiri dari *created_at* dan *updated_at* serta *deleted_at* yang digunakan sebagai *softDeletes* pada sistem. Kemudian terdapat entitas *Arsip* yang memiliki attribute *id* yang merupakan primary key, *user_id* yang merupakan foreign key dan memiliki hubungan *many to one* dengan entitas *users*, kemudian terdapat *judul*, *ruang*, *jenis_laporan* yang bertipe data string, kemudian terdapat *dosen_pembimbing_1*, *dosen_pembimbing_2* dan *dosen_penguji* yang nama-namanya diambil dari entitas *Dosen*, kemudian terdapat *dokumen* yang nantinya akan digunakan sebagai arsip yang akan disimpan, *tgl_seminar*, dan *timestamps*. Kemudian terdapat entitas *dosen* yang memiliki *id* sebagai primary key dan *name*, *nip* serta *timestamps* yang merupakan attribute dari entitas *Dosen* tersebut.

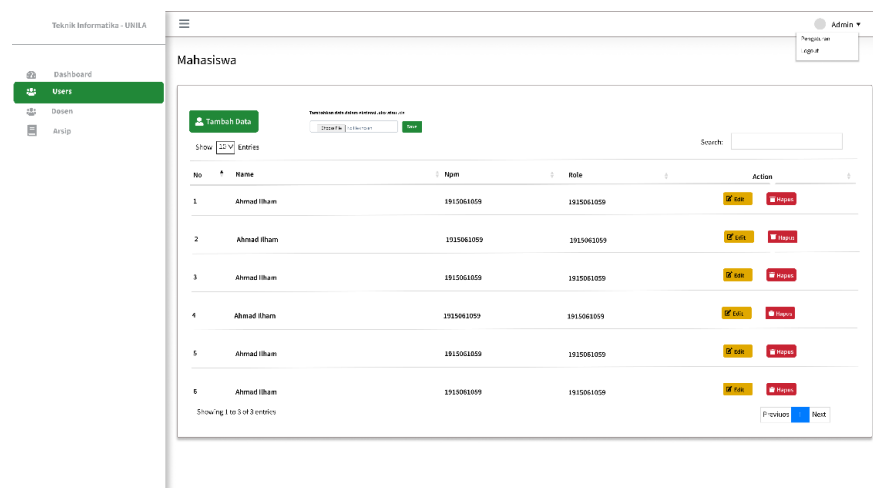
4. Desain *Dashboard User*



Gambar 4. 19 Desain *Dashboard User*

Gambar 4.19 merupakan Desain *Dashboard User*. Pada desain *dashboard user* terdapat navbar yang berisi dropdown untuk *logout*. Kemudian terdapat navbar yang terdiri dari nama program studi, *dashboard*, *user*, dosen dan arsip. Pada tengah halaman terdapat konten yang terdiri dari total *user*, dosen dan arsip serta daftar arsip-arsip yang baru saja di *upload*.

5. Desain List *User*



Gambar 4. 20 Desain List *User*

Gambar 4.20 merupakan Desain List *User*. Pada desain *list user* memiliki konten *list* dari *user* yang kolom-kolomnya terdiri dari no, nama, npm, *role* dan *action*. Kolom *action* berisikan tombol yang bisa mengarahkan ke halaman edit *user* dan tombol hapus untuk menghapus *user*. Selain itu juga terdapat tombol tambah data . Kemudian terdapat juga *form* input untuk menambahkan data *user* dengan menggunakan file berektensi .xlsx atau .xls

6. Desain List Arsip

No	Judul	Ruang	Jenis Seminar	Tanggal Seminar	Action
1	Rancang Bangun Sistem Informasi E-Arsip Laporan Akhir dan Kerja Praktek mahasiswa teknik Informatika Universitas Lampung	Ruang Seminar	Skipul	2023 - 10 - 22	<button>Add</button> <button>Edit</button> <button>Delete</button>
1	Rancang Bangun Sistem Informasi E-Arsip Laporan Akhir dan Kerja Praktek mahasiswa teknik Informatika Universitas Lampung	Ruang Seminar	Skipul	2023 - 10 - 22	<button>Add</button> <button>Edit</button> <button>Delete</button>
1	Rancang Bangun Sistem Informasi E-Arsip Laporan Akhir dan Kerja Praktek mahasiswa teknik Informatika Universitas Lampung	Ruang Seminar	Skipul	2023 - 10 - 22	<button>Add</button> <button>Edit</button> <button>Delete</button>
1	Rancang Bangun Sistem Informasi E-Arsip Laporan Akhir dan Kerja Praktek mahasiswa teknik Informatika Universitas Lampung	Ruang Seminar	Skipul	2023 - 10 - 22	<button>Add</button> <button>Edit</button> <button>Delete</button>
1	Rancang Bangun Sistem Informasi E-Arsip Laporan Akhir dan Kerja Praktek mahasiswa teknik Informatika Universitas Lampung	Ruang Seminar	Skipul	2023 - 10 - 22	<button>Add</button> <button>Edit</button> <button>Delete</button>

Gambar 4. 21 Desain List Arsip

Gambar 4.21 merupakan Desain List Arsip. Pada desain *list* arsip memiliki konten *list* dari arsip yang kolom-kolomnya terdiri dari no, judul, ruang, jenis seminar, tanggal seminar dan *action*. Kolom *action* berisikan tombol yang bisa mengarahkan ke halaman edit *user*, tombol hapus untuk menghapus *user*, dan tombol yang mengarahkan ke halaman detail arsip. selain itu juga terdapat tombol tambah data arsip yang akan diarahkan ke halaman tambah arsip.

7. Desain Tambah Arsip

Gambar 4. 22 Desain Halaman Tambah Arsip

Gambar 4.22 merupakan Desain Tambah Arsip. Pada desain tambah arsip terdapat *form* yang harus diisi untuk tambah arsip. *Form-form* tersebut terdiri dari judul, ruang, jenis laporan, dosen pembimbing 1 dan 2, dosen pembahas, dokumen, dan tanggal seminar.

8. Desain Edit Arsip

Gambar 4. 23 Desain Halaman Edit Arsip

Gambar 4.23 merupakan Desain Edit Arsip. Pada desain edit arsip terdapat *form* yang harus diisi untuk edit arsip. *Form-form* tersebut terdiri dari

judul, ruang, jenis laporan, dosen pembimbing 1 dan 2, dosen pembahas, dokumen, dan tanggal seminar.

4.2.4.2 *Sprint 2*

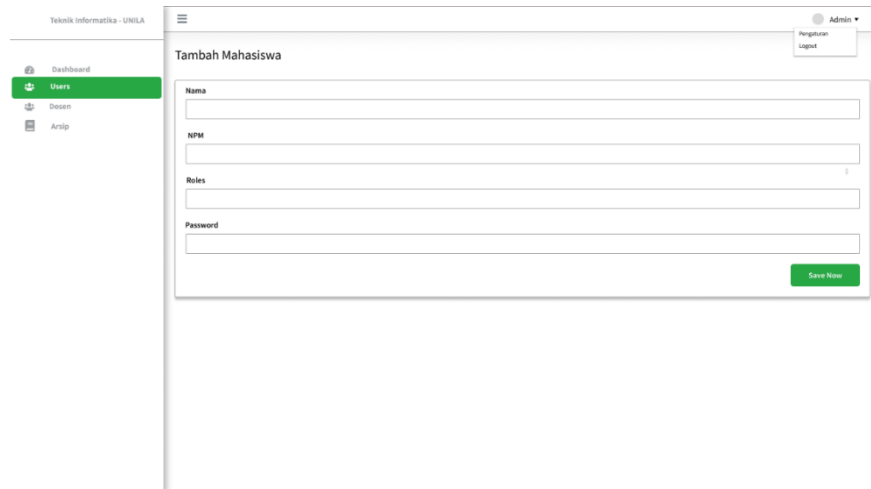
Pada *Sprint 2* terdapat empat backlog yang telah ditentukan berdasarkan estimasi waktu. Berikut backlog pada *sprint 2*:

Tabel 4. 6 Backlog *Sprint 2*

No	Backlog	Waktu Pengerjaan
1	Desain Halaman Tambah dan Edit <i>User</i>	05 Juni 2023 – 16 Juni 2023
2	Desain Halaman <i>Login</i>	
3	Desain Halaman Detail Arsip	
4	Membuat Halaman <i>Dashboard User</i>	
5	Membuat Halaman List Arsip	
6	Membuat Halaman List <i>user</i>	

Pada Tabel 4.6 merupakan backlog pada *sprint 2*. Backlog- backlog tersebut berdasarkan pada *sprint planning* yang telah dibuat yaitu dibuatnya desain halaman-halaman yang belum selesai kemudian disertai dengan pembuatan website dari halaman-halaman tersebut. Berikut merupakan hasil yang didapatkan dari *sprint 2*:

1. Desain Halaman Tambah *User*

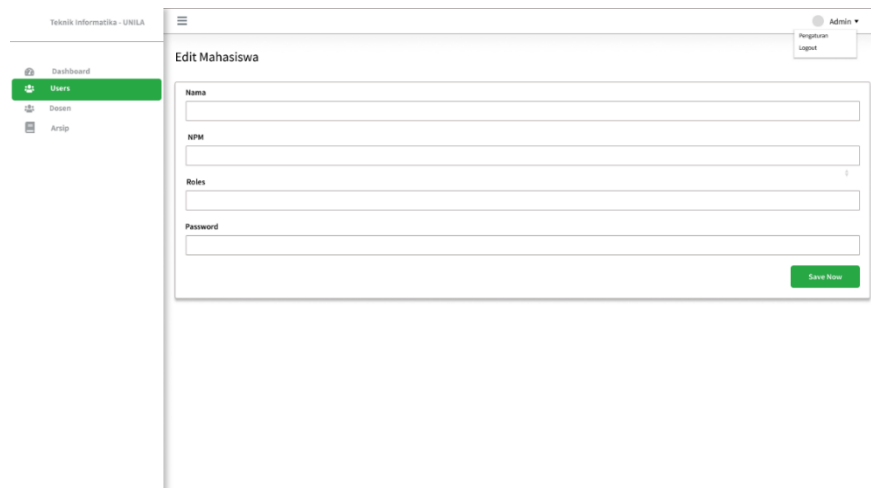


The screenshot shows a web application interface for 'Teknik Informatika - UNILA'. The left sidebar contains a menu with 'Dashboard', 'Users', 'Desain', and 'Artisip'. The 'Users' menu item is highlighted. The main content area is titled 'Tambah Mahasiswa'. It contains a form with four input fields: 'Nama', 'NPM', 'Roles', and 'Password'. A green 'Save Now' button is located at the bottom right of the form. In the top right corner, there is a user profile dropdown menu showing 'Admin' and options for 'Pengaturan' (Settings) and 'Logout'.

Gambar 4. 24 Desain Tambah *User*

Gambar 4.24 merupakan Desain Tambah *User*. Pada desain tambah *user* terdapat *form* yang harus diisi untuk tambah *user*. *Form-form* tersebut terdiri dari nama, npm, roles, dan password.

2. Desain Halaman Edit *User*



The screenshot shows the same web application interface as before, but the main content area is titled 'Edit Mahasiswa'. The form contains the same four input fields: 'Nama', 'NPM', 'Roles', and 'Password'. A green 'Save Now' button is located at the bottom right of the form. The sidebar and top navigation elements are identical to the previous screenshot.

Gambar 4. 25 Edit *User*

Gambar 4.25 merupakan Desain Edit *User*. Pada desain edit *user* terdapat *form* yang harus diisi untuk edit *user*. *Form-form* tersebut terdiri dari nama, npm, roles, dan password.

3. Desain Halaman Detail Arsip

The screenshot shows the 'Data Arsip' page. On the left is a sidebar with 'Arsip' highlighted. The top header includes 'Teknik Informatika - UNILA' and a user profile for 'Admin'. The main content area is titled 'Data Arsip' and contains a form with a 'Download File' button and a table of seminar details.

Jenis Laporan	Skiprol
Ruang Seminar	Ruang Seminar
Dosen Pembimbing 1	John Doe
Dosen Pembimbing 2	John Doe
Dosen Pembahas	John Doe
Tanggal Seminar	2023-07-31

Gambar 4. 26 Desain Halaman Detail Arsip

Gambar 4.26 merupakan Desain Halaman Detail Arsip. Pada desain detail arsip terdapat *form* yang harus diisi untuk detail arsip. *Form-form* tersebut terdiri dari judul, ruang, jenis laporan, dosen pembimbing 1 dan 2, dosen pembahas, dokumen, dan tanggal seminar.

4. Desain Halaman List Dosen

The screenshot shows the 'Dosen' page. On the left is a sidebar with 'Dosen' highlighted. The top header includes 'Teknik Informatika - UNILA' and a user profile for 'Admin'. The main content area is titled 'Dosen' and contains a table of lecturer data with 'Tambah Data' and 'Search' buttons.

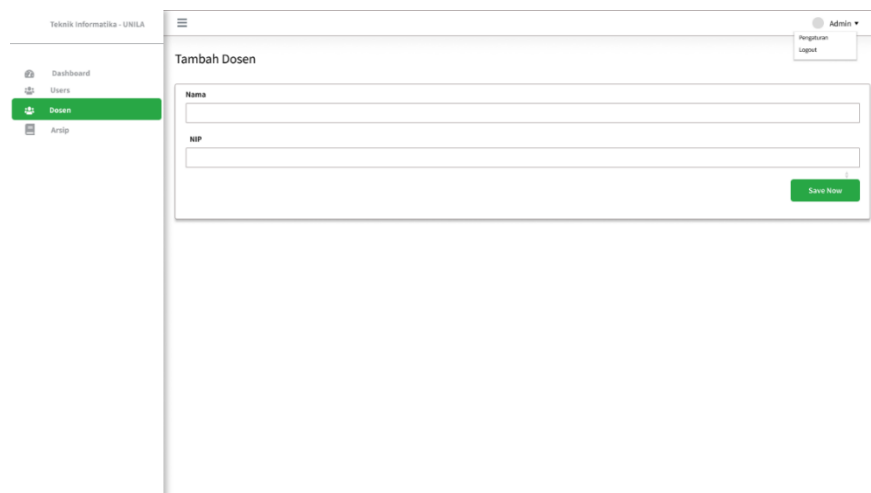
No	Name	Nip	Action
1	Ahmad Ihsan	1915061059	Edit Delete
2	Ahmad Ihsan	1915061059	Edit Delete
3	Ahmad Ihsan	1915061059	Edit Delete
4	Ahmad Ihsan	1915061059	Edit Delete
5	Ahmad Ihsan	1915061059	Edit Delete
6	Ahmad Ihsan	1915061059	Edit Delete

Gambar 4. 27 Desain Halaman List Dosen

Gambar 4.27 merupakan halaman *list* dosen. Halaman list dosen terdiri dari tombol tambah data untuk menambah data dosen seperti pada gambar 4.27.

terdapat kolom yang terdiri dari no, nama, nip dan *action*. *Action* pada *list* dosen memiliki dua tombol yaitu tombol edit yang mengarahkan ke halaman edit dosen dan hapus. Kemudian terdapat juga *form* input untuk menambahkan data *user* dengan menggunakan file berektensi .xlsx atau .xls.

5. Desain Halaman Tambah Dosen



The image shows a web application interface for adding a lecturer. The main content area is titled "Tambah Dosen". It contains a form with two input fields: "Nama" (Name) and "NIP" (National Identity Number). A green button labeled "Save New" is positioned at the bottom right of the form. The left sidebar shows a navigation menu with "Dashboard", "Users", "Dosen" (highlighted), and "Acadp". The top right corner displays the user's role as "Admin" and a "Logout" button.

Gambar 4. 28 Desain Halaman Tambah Dosen

Gambar 4.28 merupakan Desain Tambah Dosen. Pada desain tambah dosen terdapat *form* yang harus diisi untuk tambah dosen. *Form-form* tersebut terdiri dari nama dan nip.

6. Desain Halaman Edit Dosen

Gambar 4. 29 Desain Halaman Edit Dosen

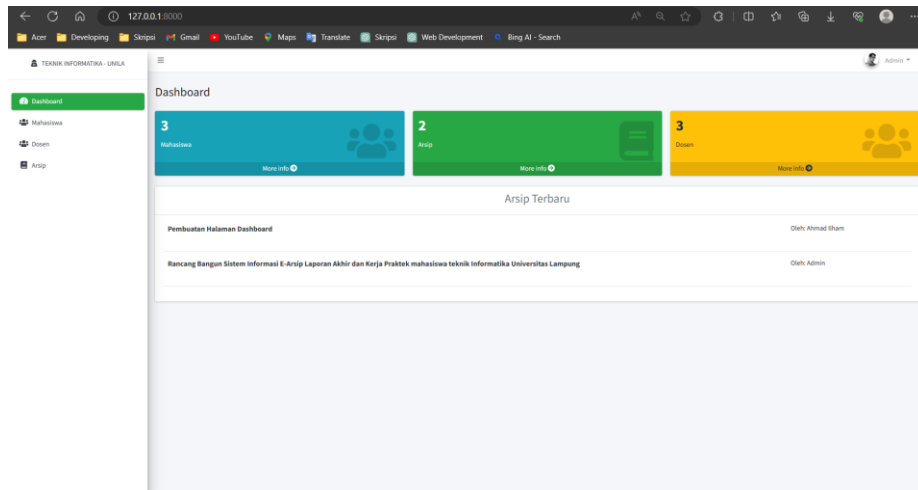
Gambar 4.29 merupakan Desain Edit Dosen. Pada desain edit dosen terdapat *form* yang harus diisi untuk edit dosen. *Form-form* tersebut terdiri dari nama dan nip.

7. Desain Halaman *Login*

Gambar 4. 30 Desain Halaman *Login*

Gambar 4.30 merupakan Desain Halaman *Login*. Pada Halaman *Login* terdapat nama dari program studi teknik informatika universitas lampung dan terdapat card yang terdiri dari npm dan *password* untuk masuk ke sistem.

8. Membuat Halaman *Dashboard User*



Gambar 4. 31 Halaman *Dashboard User*

Gambar 4.31 Merupakan Halaman *Dashboard User*. *Dashboard User* merupakan halaman awal sistem informasi e-arsip setelah melakukan *login*. Pada *Dashboard User* akan menampilkan Total *user*, dosen dan arsip. Serta terdapat arsip-arsip yang baru di *upload*. Seperti pada gambar 4.31 terdapat dua arsip baru yang di unggah oleh admin dan ahmad ilham. Berikut merupakan kode dari pembuatan halaman *Dashboard User*.

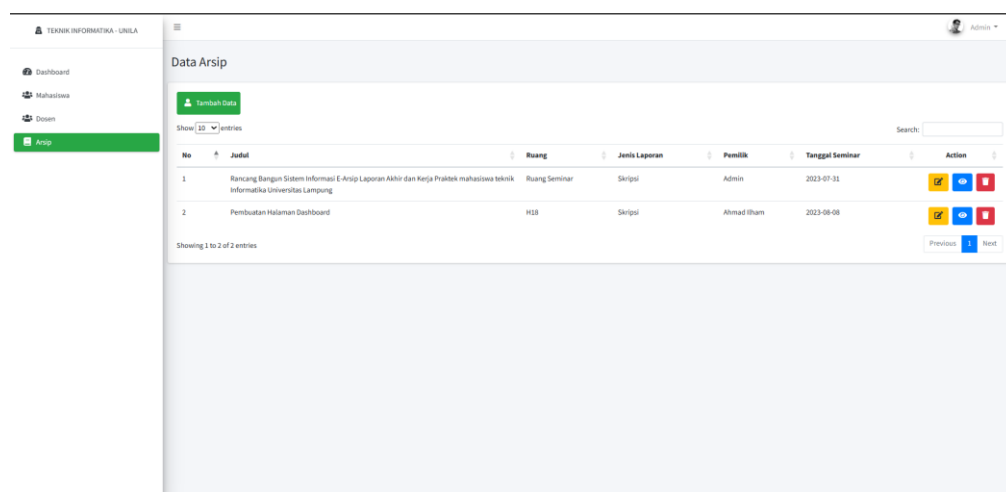
```
public function index()
{
    $jumlahArsip = Arsip::all()->count();
    $jumlahDosen = Dosen::all()->count();







    $user = User::all();
    $arsip = Arsip::with(['user' => function ($query) {
        $query->withTrashed();
    }])->latest('created_at')->take(7)->get();
    return view('pages.dashboard.index', [
        'JumlahArsip' => $jumlahArsip,
        'JumlahDosen' => $jumlahDosen,
        'arsip' => $arsip,
        'user' => $user
    ]);
}
```

Gambar 4. 32 Code *Dashboard User*

Gambar 4.32 merupakan *code dashboard user*. Pada *code* tersebut terdapat variabel \$jumlahArsip untuk menghitung jumlah arsip yang ada dan variabel \$jumlahDosen untuk menghitung jumlah dosen. Kemudian terdapat variabel \$user untuk mengambil semua data *user* dari *database* dan terdapat variabel \$arsip untuk mengambil arsip dari *database* dengan mengambil data arsip berdasarkan pada *created_at* atau tanggal dibuatnya arsip tersebut dan di tampilkan maksimal 7 arsip saja.

9. Membuat Halaman *List* Arsip



No	Judul	Ruang	Jenis Laporan	Pemilik	Tanggal Seminar	Action
1	Rancang Bangun Sistem Informasi E-Arsip Laporan Akhir dan Kerja Praktek mahasiswa teknik Informatika Universitas Lampung	Ruang Seminar	Skripsi	Admin	2023-07-31	  
2	Pembuatan Halaman Dashboard	H2B	Skripsi	Ahmad Iham	2023-08-08	  

Gambar 4. 33 Halaman *List* Arsip

Gambar 4.33 Merupakan Halaman *List* Arsip. Pada gambar tersebut menampilkan dua buah arsip yang telah di unggah dengan keterangan disetiap kolomnya terdiri dari judul, ruang, jenis laporan, pemilik dan tanggal seminar, serta kolom terakhir yang akan mengarahkan ke halaman edit arsip dan tombol untuk menghapus arsip. Berikut merupakan code dari halaman *list* Arsip:

```

public function index()
{
    if (request()->ajax()) {
        $query = Arsip::with(['user' => function ($query) {
            $query->withTrashed();
        }->get();

        return DataTables::of($query)
            ->addColumn('action', function ($arsip) {
                return view('components.arsipAction', compact('arsip'));
            })->make();
    }

    return view('pages.arsip.index');
}

```

Gambar 4. 34 *Code List Arsip*

Gambar 4.34 merupakan *code list* arsip. Pada *code list* arsip menggunakan *Datatables Laravel*. Data arsip di olah di sisi server dengan membuat if statement jika *request* merupakan suatu *ajax* maka akan diambil data arsip dengan *user* yang dimasukan ke dalam variabel *\$query*.

10. Membuat Halaman *List User*

No	Name	Npm	Role	Action
1	Admin	admin123	admin	Edit Hapus
2	Ahmad Iham	1915060093	mahasiswa	Edit Hapus
3	Lucky Albar	1915060096	mahasiswa	Edit Hapus

Gambar 4. 35 Halaman *List User*

Gambar 4.35 merupakan Halaman *List User*. Pada gambar tersebut menampilkan tiga orang *user* yang telah terdaftar pada sistem informasi e-arsip

dengan keterangan disetiap kolomnya terdiri dari no, nama, npm, dan *role*, serta kolom terakhir yang akan mengarahkan ke halaman edit *user* dan tombol untuk menghapus *user*. Halaman ini hanya dapat diakses oleh *user* yang memiliki *role* admin. Berikut merupakan *code* dari *list user*:

```
public function index()
{
    if (request()->ajax()) {
        $query = User::all();

        return DataTables::of($query)
            ->addColumn(
                'action',
                function ($user) {
                    return view('components.userAction', compact('user'));
                }
            )->make();
    }

    return view('pages.user.index');
}
```

Gambar 4. 36 *Code List User*

Gambar 4.36 merupakan *code list user*. Pada *code list user* menggunakan *Datatables Laravel*. Data *user* di olah di sisi server dengan membuat *if statement* jika *request* merupakan suatu *ajax* maka akan diambil *user* yang dimasukan ke dalam variabel *\$query*.

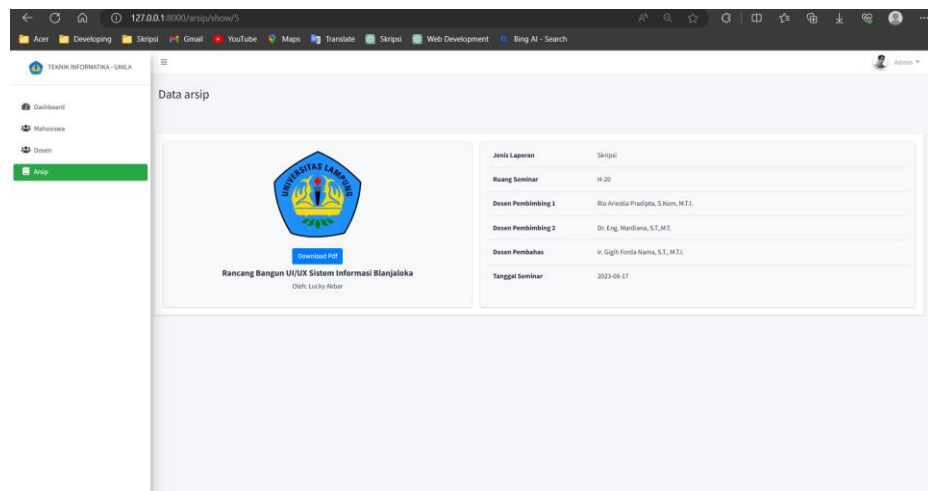
4.2.4.3 *Sprint 3*

Tabel 4. 7 Backlog *Sprint 3*

No	Backlog	Waktu Pengerjaan
1	Membuat Halaman Detail Arsip	19 Juni 2023 – 30 Juni 2023
2	Membuat Halaman Tambah dan Edit Arsip	
3	Membuat Halaman List Dosen	
4	Slicing Halaman <i>login</i>	

Pada Tabel 4.7 merupakan backlog pada *sprint* 3. Backlog-backlog tersebut berdasarkan pada *sprint* planning yang telah dibuat yaitu dilakukan pembuatan halaman-halaman yang belum dibuat di *sprint* sebelumnya.. Berikut merupakan hasil yang didapatkan dari *sprint* 2:

1. Membuat Halaman Detail Arsip



Gambar 4. 37 Halaman Detail Arsip

Gambar 4.37 merupakan halaman detail arsip yang telah diisi dengan data arsip. Pada gambar tersebut dapat dilihat memiliki dua card dimana card kiri terdiri dari logo Unviersitas Lampung serta terdapat tombol download, judul dan pemilik dari arsip tersebut. Kemudian pada card di sebelah kanan terdapat informasi tentang jenis laporan, ruang seminar, dosen pembimbing satu dan dua, dosen pembahas dan tanggal seminar dari laporan arsip tersebut. Berikut merupakan kode dari halaman detail arsip:



Gambar 4. 38 Code Detail Arsip

Gambar 4.38 merupakan *code* detail arsip. Pada *code* detail arsip dibuat variabel dengan nama `$data` untuk menampung data arsip disertai *user* untuk mengambil data arsip dan *user* secara bersamaan sesuai dengan id yang ditentukan. Kemudian data akan dikirim ke *view* dalam bentuk *array* dengan nama arsip.

2. Membuat Halaman Tambah Arsip

The screenshot shows a web browser window displaying a form titled "Tambah Data Arsip". The form includes several input fields and dropdown menus: "Judul" (text), "Ruang" (text), "Jenis Laporan" (dropdown with "Skip" as an option), "Penyakit" (dropdown with "Almond" as an option), "Dosen Pembimbing 1" (dropdown with "Dr. Eng. Mardiana, S.T., M.T." as an option), "Dosen Pembimbing 2" (dropdown with "Dr. Eng. Mardiana, S.T., M.T." as an option), and "Dosen Pengaji" (dropdown with "Dr. Eng. Mardiana, S.T., M.T." as an option). There is also a "Dokumen" section with a "Choose File" button and a "No file chosen" label, and a "Tanggal Seminar" field with a date picker set to "06/10/2022". A green "Simpan" button is located at the bottom right of the form.

Gambar 4. 39 Halaman Tambah Arsip

Gambar 4.39 merupakan halaman tambah arsip. Halaman tersebut berisi *form* untuk menambahkan data-data arsip. *Form* tersebut terdiri dari judul, ruang, jenis laporan,

pemilik, dosen pembimbing satu dan dua, dosen pembahas, dokumen arsip yang berekstensi *pdf*, dan tanggal seminar dari laporan tersebut. Berikut merupakan *code* tambah arsip:

```
public function store(ArsipRequest $request)
{
    $data = Arsip::create($request->all());

    $request->file('dokumen')->move('dokumen/dataArsip', $request->file('dokumen')->getClientOriginalName());
    $data->dokumen = $request->file('dokumen')->getClientOriginalName();

    $data->save();

    return redirect()->route('arsip.list')->with('success', 'Berhasil menambahkan data');
}
```

Gambar 4. 40 *Code* Tambah Arsip

Gambar 4.40 merupakan *code* tambah arsip. Pada *code* tersebut dilakukan pengambilan semua data dari *body*, data-data tersebut terdiri dari judul, ruang, jenis laporan, pemilik, dosen pembimbing satu dan dua, dosen pembahas, dokumen arsip yang berekstensi *pdf*, dan tanggal seminar dari laporan tersebut. Kemudian terdapat operasi untuk menyimpan file yang diambil dari *body*.

3. Membuat Halaman Edit Arsip

The screenshot shows a web browser window with the URL 127.0.0.1:8000/arsip/5. The page title is 'Edit Data Arsip'. The form fields are as follows:

- Judul:** Rancangan Bangun UI/UX Sistem Informasi Berjarkita
- Ruang:** H-20
- Jenis Laporan:** Skripsi
- Pemilik:** Admin
- Dosen Pembimbing 1:** Dr. Eng. Mardiana, S.T., M.T.
- Dosen Pembimbing 2:** Dr. Eng. Mardiana, S.T., M.T.
- Dosen Pembahas:** Dr. Eng. Mardiana, S.T., M.T.
- Dokumen:** A file upload field with a 'Choose File' button and the text 'No file chosen'.
- Tanggal Seminar:** 27/06/2023
- Save Data:** A green button at the bottom right.

Gambar 4. 41 Halaman Edit Arsip

Gambar 4.41 merupakan halaman edit arsip. Pada Halaman edit arsip memiliki *form* yang sama dengan tambah arsip yaitu terdiri dari judul, ruang, jenis laporan, pemilik, dosen pembimbing satu dan dua, dosen pembahas, dokumen dan tanggal seminar. Perbedaan dari halaman tersebut dengan halaman tambah arsip hanya terletak pada judul yaitu edit data arsip. Berikut merupakan *code* dari halaman edit arsip :

```
public function update(Request $request, string $id)
{
    $data = $request->all();
    $arsip = Arsip::findOrFail($id);
    $data = $request->except('user_id');
    $file = $request->file('dokumen');

    if ($file != null) {
        $request->file('dokumen')->move('dokumen/dataArsip', $request->file('dokumen')->getClientOriginalName());
        $data['dokumen'] = $request->file('dokumen')->getClientOriginalName();
    }

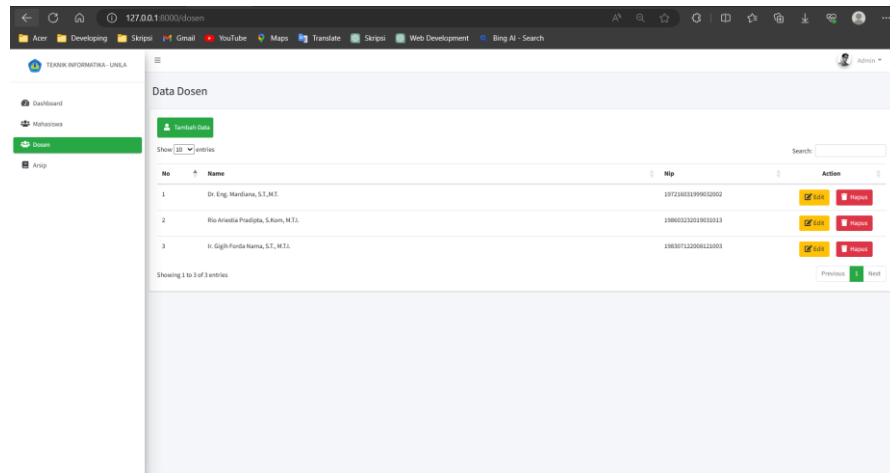
    $arsip->update($data);

    return redirect()->route('arsip.list')->with('success', 'Berhasil Mengedit Data');
}
```

Gambar 4. 42 *Code* Edit Arsip

Gambar 4.42 merupakan *code* edit arsip. Pada *code* tersebut akan diedit data dari masing-masing arsip berdasarkan id arsip kemudian data akan di *request* ke *body* yang terdiri dari judul, ruang, jenis laporan, pemilik, dosen pembimbing satu dan dua, dosen pembahas, dan tanggal seminar. Kemudian *users_id* tidak akan ikut diubah karena dikecualikan dan file dokumen pada database bisa tidak diubah juga dan tetap memakai file dokumen yang lama.

4. Membuat Halaman List Dosen



Gambar 4. 43 Halaman List Dosen

Gambar 4.43 merupakan halaman *list* dosen. Pada halaman tersebut terdapat judul Data Dosen. Kemudian terdapat tombol untuk menambahkan dosen yang akan mengarahkan ke halaman tambah dosen, kemudian ada input tipe file yang bisa digunakan untuk menambahkan dosen dengan menggunakan file yang berekstensi .xlsx atau .xls. Setelah itu terdapat tabel *list* dosen yang kolom-kolomnya terdiri dari no, nama, nip dan *action*. Pada *action* terdapat dua tombol yaitu tombol edit yang mengarahkan ke halaman edit untuk mengedit data dosen dan tombol hapus untuk menghapus data dosen. Halaman ini hanya dapat diakses oleh *user* yang memiliki *role* admin. Berikut merupakan *code* halaman *list* dosen:

```
public function index()
{
    if (request()->ajax()) {
        $query = Dosen::all();

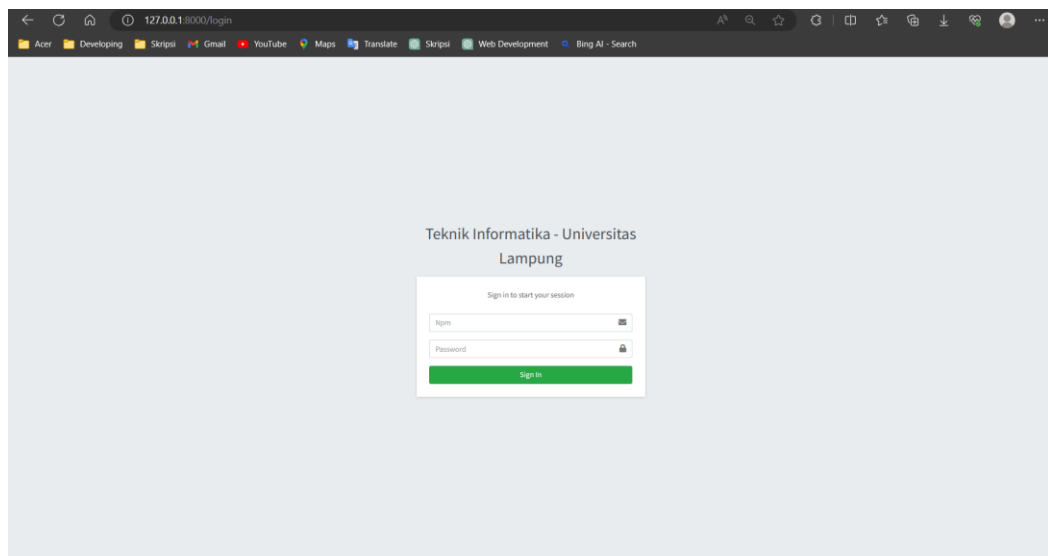
        return DataTables::of($query)
            ->addColumn('action', 'components.dosenAction')
            ->make();
    }

    return view('pages.dosen.index');
}
```

Gambar 4. 44 Code List Dosen

Gambar 4.44 merupakan *code list* dosen. Pada *code list* dosen menggunakan *Datatables Laravel*. Data dosen di olah di sisi server dengan membuat *if statement* jika *request* merupakan suatu *ajax* maka akan diambil data dosen yang dimasukan ke dalam variabel *\$query*.

5. Slicing Halaman *Login*



Gambar 4. 45 Slicing Halaman *Login*

Gambar 4.45 merupakan slicing halaman *login*. Pada halaman *login* terdapat title yang merupakan nama program studi Teknik Informatika Universitas Lampung. Pada card *login* terdapat *npm* dan *password* yang digunakan untuk mengakses halaman *dashboard user*. Kemudian terdapat tombol *login* dengan text sign in.

4.2.4.4 Sprint 4

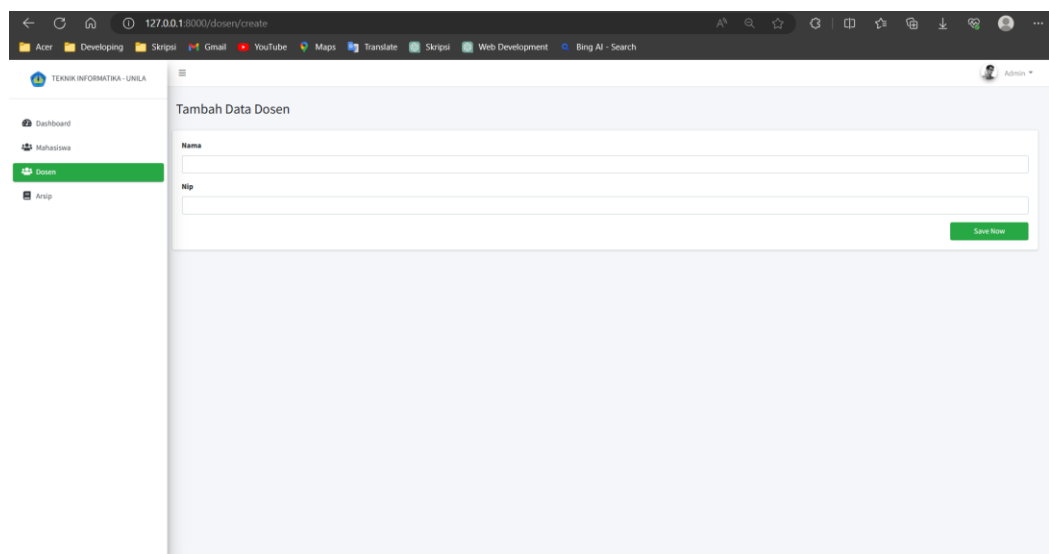
Tabel 4. 8 Backlog *Sprint 3*

No	Backlog	Waktu Pengerjaan
1	Membuat Halaman Tambah dan Edit Dosen	03 Juli 2023 – 14 Juli 2023

2	Membuat Halaman Tambah dan Edit <i>User</i>	
3	Membuat Fitur <i>Login, logout</i> dan Autentikasi	

Pada Tabel 4.8 merupakan backlog pada *sprint* 3. Backlog-backlog tersebut berdasarkan pada *sprint* planning yang telah dibuat yaitu dilakukan pembuatan desain interface dari *sprint* yang memiliki prioritas sedang dan pembuatan halaman-halaman serta debunging. Berikut merupakan hasil yang didapatkan dari *sprint* 2:

1. Membuat Halaman Tambah Dosen



Gambar 4. 46 Halaman Tambah Dosen

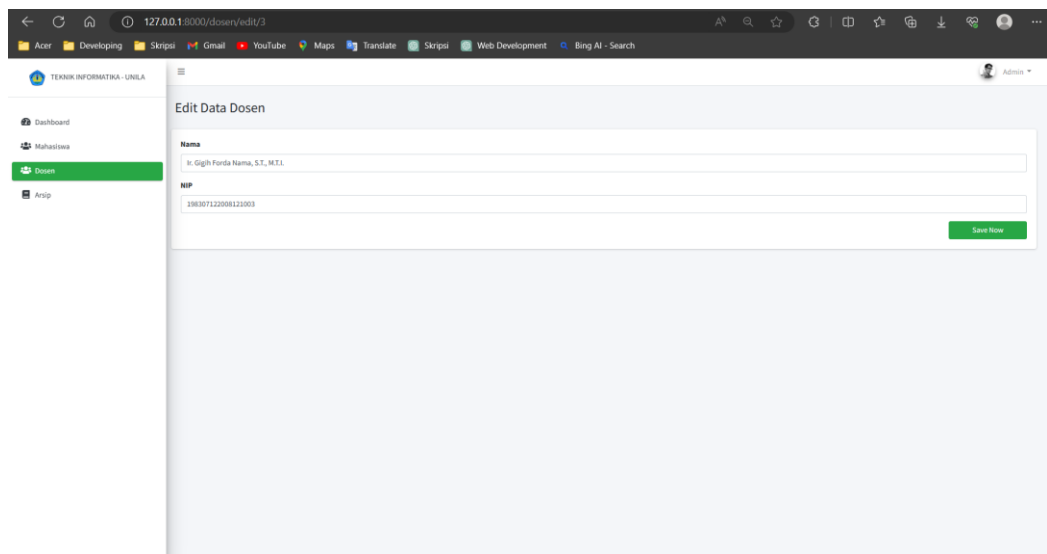
Gambar 4.46 merupakan halaman tambah dosen. Halaman tambah dosen memiliki title Tambah Data Dosen. Kemudian pada halaman tersebut terdapat card yang berisi *form* yang terdiri dari nama dan nip dari dosen serta terdapat tombol save data untuk menambahkan data dosen tersebut. Halaman ini hanya dapat diakses oleh *user* yang memiliki *role* admin. Berikut merupakan *code* halaman tambah dosen:



Gambar 4. 47 Code Tambah Dosen

Gambar 4.47 merupakan *code* tambah dosen. Pada *code* tersebut dilakukan *request* untuk mengambil semua data di *body*. Kemudian akan di buat data dosen berdasarkan variabel *\$data* yang diambil dari *body*.

2. Membuat Halaman Edit Dosen



Gambar 4. 48 Halaman Edit Dosen

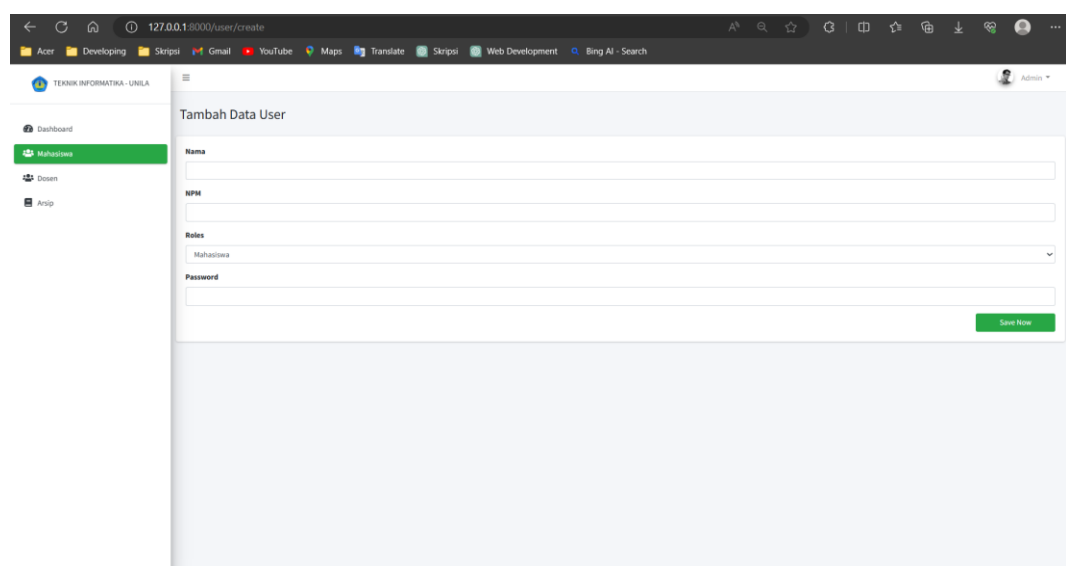
Gambar 4.48 merupakan halaman edit dosen. Halaman edit dosen memiliki title Edit Data Dosen. Kemudian pada halaman tersebut terdapat card yang berisi *form* yang terdiri dari nama dan nip dari dosen yang telah terisi oleh data, serta terdapat tombol save data untuk mengedit data dosen tersebut. Halaman ini hanya dapat diakses oleh *user* yang memiliki *role* admin. Berikut *code* halaman edit dosen:



Gambar 4. 49 *Code Edit Dosen*

Gambar 4.49 merupakan *code* edit dosen. Pada gambar tersebut data dosen akan di edit berdasarkan id yang dipilih. Kemudian akan diambil semua data yang di *request* di *body*.

3. Membuat Halaman Tambah *User*



Gambar 4. 50 Halaman Tambah *User*

Gambar 4.50 merupakan halaman tambah *user*. Pada halaman tersebut terdapat title Tambah Data *User*. Kemudian terdapat card yang berisi *form* input yang terdiri dari nama *user*, npm *user*, *role user* yang akan mengakses sistem informasi e-arsip yang terdiri dari admin dan mahasiswa, dan *password*. Kemudian terdapat tombol save

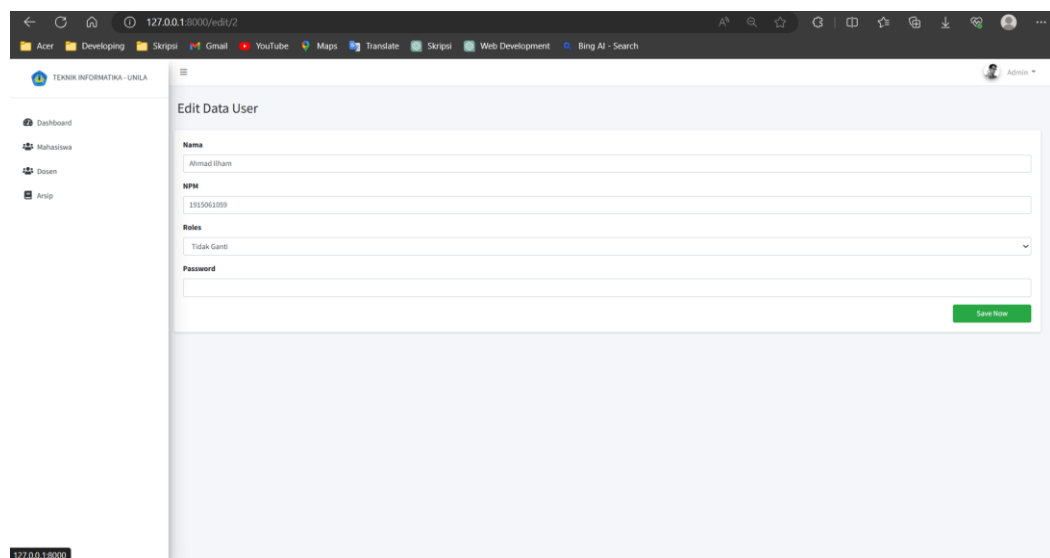
now untuk menyimpan data tersebut. Halaman ini hanya dapat diakses oleh *user* yang memiliki *role* admin. Berikut merupakan *code* tambah *user*:

```
public function store(UserRequest $request)
{
    $data = $request->all();
    $data['password'] = bcrypt($request->password);
    User::create($data);
    return redirect()->route('user.list')->with('success', 'Berhasil Menambahkan Data');
}
```

Gambar 4. 51 *Code* Tambah *User*

Gambar 4.51 merupakan *code* tambah *user*. Pada *code* tersebut dilakukan *request* untuk mengambil semua data di *body*. Kemudian akan di buat data *user* berdasarkan variabel *\$data* yang diambil dari *body* dengan *password* akan di *hash*.

4. Membuat Halaman Edit *User*



The screenshot shows a web browser window with the URL 127.0.0.1:8000/edit/2. The page title is 'Edit Data User'. On the left is a sidebar menu with 'Dashboard', 'Mahasiswa', 'Dosen', and 'Ansis'. The main content area contains a form with the following fields:

- Nama:** Ahmad Iham
- NPM:** 1215061059
- Roles:** Tidak Ganti (dropdown menu)
- Password:** (empty text field)

A green 'Save Now' button is located at the bottom right of the form.

Gambar 4. 52 Halaman Edit *User*

Gambar 4.52 merupakan halaman edit *user*. Pada halaman tersebut terdapat title Edit Data *User*. Kemudian terdapat card yang berisi *form* input yang terdiri dari

nama *user*, *npm user*, *role user* yang akan mengakses sistem informasi e-arsip yang terdiri dari admin dan mahasiswa, dan *password* yang telah diisi dengan data dari database. Kemudian terdapat tombol save now untuk menyimpan data tersebut. Halaman ini hanya dapat diakses oleh *user* yang memiliki *role* admin. Berikut merupakan *code* edit *user*:

```
public function update(UserRequest $request, string $id)
{
    $data = $request->all();
    $user = User::findOrFail($id);
    if ($request->password) {
        $data['password'] = bcrypt($request->password);
    } else {
        unset($data['password']);
    }
    $user->update($data);

    return redirect()->route('user.list')->with('success', 'Berhasil Mengedit Data');
}
```

Gambar 4. 53 *Code* Edit *User*

Gambar 4.53 merupakan *code* edit *user*. Pada gambar tersebut data dosen akan di edit berdasarkan id yang dipilih. Kemudian akan diambil semua data yang di *request* di *body* dan terdapat *if statement* jika terdapat permintaan untuk mengubah *password* maka *password* akan diubah tetapi jika pad *form input password* dikosongkan maka *password* tidak akan diubah.

5. Membuat Halaman *Login*, *logout* dan Autentikasi



```

public function login(Request $request)
{
    $request->validate([
        'npm' => 'required',
        'password' => 'required',
    ]);
    $data = $request->only('npm', 'password');

    if (Auth::attempt($data)) {
        return redirect()->route('dashboard')->with('success', 'Berhasil Login');
    } else {
        return redirect()->back()->withInput()->withErrors(['email' => 'Email atau password salah']);
    }
}

public function logout()
{
    Auth::logout();
    return redirect()->route('signin')->with('success', 'Berhasil logout');
}

```

Gambar 4. 54 *Login*, *Logout* dan Autentikasi

Gambar 4.54 merupakan proses pembuatan *login*, *logout* dan autentikasi. Pada method public *login* dilakukan proses validasi dimana harus menyertakan npm dan *password* untuk melakukan *login*, kemudian sistem akan meminta npm dan *password* dari halaman *login*. Setelah itu akan dilakukan proses autentikasi pengguna berdasarkan npm dan *password* yang dimasukan dengan data di database jika cocok maka akan dialihkan ke halaman *dashboard user* jika tidak akan tetap di halaman tersebut dengan peringatan npm atau *password* salah. Kemudian terdapat method public *logout* yang digunakan untuk keluar dari sistem informasi e-arsip.

4.2.5 *Sprint Review*

4.2.6 *Sprint Retrospective*

4.2.7 *Testing*

DAFTAR PUSTAKA

- [1] Juhriyansyah Dale, A Akrim, and Baharuddin, *Pengantar Teknologi Informasi*, 1st ed. Depok: PT RAJAGRAFINDO PERSADA, 2020.
- [2] Republik Indonesia, *Undang-Undang Republik Indonesia Nomor 43 Tahun 2009 Tentang Kearsipan*. Indonesia: www.bphn.go.id, 2009.
[Online]. Available: www.bphn.go.id
- [3] Rifauddin Machsun, “PENGELOLAAN ARSIP ELEKTRONIK BERBASIS TEKNOLOGI,” *Khazanah Al- Hikmah Jurnal Ilmu Perpustakaan, Informasi, dan Kearsipan*, vol. 4, no. 2, pp. 168–178, 2016.
- [4] H. A. Rusdiana, M. M. Moch, S. T. Irfan, M. Kom, and H. M. A. Ramdhadi, *Sistem Informasi Manajemen*, 1st ed. Bandung: PUSTAKA SETIA BANDUNG, 2014.
- [5] M. Prabowo, *METODOLOGI PENGEMBANGAN SISTEM INFORMASI*. Salatiga: Lembaga Penelitian dan Pengabdian kepada Masyarakat (LP2M) IAIN Salatiga, 2020.
- [6] A. Oktarini, S. ; Ari, and A. ; Sunarti, *WEB PROGRAMMING*, 1st ed. Yogyakarta: GRAHA ILMU, 2019.

- [7] D. Purnama Sari, R. Wijanarko, and J. X. Menoreh Tengah, "Implementasi Framework Laravel pada Sistem Informasi Penyewaan Kamera (Studi Kasus Di Rumah Kamera Semarang)," vol. 2, no. 1, pp. 32–36, 2019.
- [8] A. Firman, H. F. Wowor, and X. Najoan, "Sistem Informasi Perpustakaan Online Berbasis Web," *Teknik Elektro dan Komputer*, vol. 5, no. 2, 2016.
- [9] Endra Robby Yuli and Aprilita Synta Dwi, "E-Report Berbasis Web Menggunakan Metode Model View Controller Untuk Mengetahui Peningkatan Perkembangan Prestasi Anak Didik," *Jurnal Sistem Informasi dan Telematika*, vol. 9, no. 1, pp. 15–22, Jun. 2018.
- [10] A. Saputra, "Manajemen Basis Data MYSQL pada Situs PFT," *Berita Dirgantara*, vol. 13, no. 4, pp. 155–162, 2012.
- [11] V. Febrian, M. R. Ramadhan, M. Faisal, and A. Saifudin, "Pengujian pada Aplikasi Penggajian Pegawai dengan menggunakan Metode Blackbox," *Jurnal Informatika Universitas Pamulang*, vol. 5, no. 1, pp. 2622–4615, 2020, [Online]. Available: <http://openjournal.unpam.ac.id/index.php/JTSI/index61>
- [12] Darwiyanti Sri, *Pengantar Unified Modeling Language(UML)*. 2003.
- [13] T. N. Lina *et al.*, "SISTEM INFORMASI E-ARSIP BERBASIS WEB (STUDI KASUS: PT HALEYORA POWERINDO CABANG SORONG)," *JURNAL JENDELA ILMU*, vol. 1, no. 1, pp. 1–5, 2020.

- [14] A. Andipradana and Hartomo Dwi Kristoko, “Rancang Bangun Aplikasi Penjualan Online Berbasis Web Menggunakan Metode Scrum,” *Jurnal Algoritma*, vol. 19, no. 1, pp. 161–172, Aug. 2021.
- [15] A. Rizaldi, E. Maria, T. Wahyono, P. Purwanto, and K. D. Hartomo, “Analisis Penerapan Metode Scrum Pada Pengembangan Sistem Informasi Akuntansi Koperasi,” *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 6, no. 1, p. 57, Jan. 2022, doi: 10.30865/mib.v6i1.3349.
- [16] A. Witania, A. Dana Nugraha, L. Fajar Sari, N. Lia Megawati, and N. Nur Fadillah, “Analisis perbandingan metode manajemen proyek ti yang paling sering digunakan di indonesia dan luar negeri: a literature review,” *JOURNAL OF MANAGEMENT Small and Medium Entreprises (SME’s)*, vol. 15, no. 2, pp. 299–316, 2022.
- [17] M. I. Kausar Bagwan and P. D. Swati Ghule, “A Modern Review on Laravel-PHP Framework,” *IRE Journals*, vol. 2, no. 2, pp. 1–3, 2019.
- [18] S. Nidhra, “Black Box and White Box Testing Techniques - A Literature Review,” *International Journal of Embedded Systems and Applications*, vol. 2, no. 2, pp. 29–50, Jun. 2012, doi: 10.5121/ijesa.2012.2204.
- [19] I. Tahyudin and Zidni Iman Sholihati, “Pengembangan Aplikasi Tiga-Tingkat Menggunakan Metode Scrum pada Aplikasi Presensi Karyawan Glints Academy,” *Jurnal RESTI (Rekayasa Sistem dan*

- Teknologi Informasi*), vol. 6, no. 1, pp. 169–176, Feb. 2022, doi: 10.29207/resti.v6i1.3793.
- [20] M. Dedi Irawan and S. A. Simargolang, “Implementasi E-Arsip Pada Program Studi Teknik Informatika,” *Jurnal Teknologi Informasi*, vol. 2, no. 1, 2018.
- [21] H. R. Suharno, N. Gunantara, and M. Sudarma, “Analisis Penerapan Metode Scrum Pada Sistem Informasi Manajemen Proyek Dalam Industri & Organisasi Digital,” *Majalah Ilmiah Teknologi Elektro*, vol. 19, no. 2, p. 203, Dec. 2020, doi: 10.24843/mite.2020.v19i02.p12.