

Rancang Bangun Sistem Informasi Persuratan berbasis Web di Fakultas Vokasi Universitas Negeri Surabaya

Dyahayu Retno Wulan¹, Salamun Rohman Nudin²

Manajemen Informatika, Universitas Negeri Surabaya
Jl. Ketintang, Ketintang, Kec. Gayungan, Surabaya, Jawa Timur 60231

¹dyahayu.19041@mhs.unesa.ac.id

²salamun.nudin@unesa.ac.id

Abstrak— Pengelolaan surat dan arsip merupakan salah satu aktivitas yang sering ditangani oleh sebuah organisasi, institusi, dan perusahaan. Salah satu pelayanan yang dilakukan oleh staf Tata Usaha Fakultas Vokasi Universitas Negeri Surabaya adalah permohonan persuratan. Ada enam macam surat yang bisa diajukan oleh seluruh sivitas akademika dalam lingkup Fakultas Vokasi Universitas Negeri Surabaya, yaitu Surat Keterangan Mahasiswa Aktif, Surat Keterangan Tunjangan Anak pada Gaji Orang Tua, Surat izin Magang KKN Tematik, Surat Izin Magang/PKL, Surat Izin Observasi Mata Kuliah, dan Surat Izin Observasi Tugas Akhir. Saat ini, prosedur pengajuan surat tersebut masih dilakukan secara konvensional melalui template yang diunduh secara manual pada situs fakultas. Pengembangan Aplikasi pengajuan surat dimaksudkan agar sistem tata kelola permohonan menjadi lebih efisien. Pengembangan aplikasi pengajuan surat menggunakan Laravel sebagai Framework. Penggunaan teknologi meliputi bahasa pemrograman PHP dan My SQL untuk menyimpan basis data. Penelitian ini bertujuan untuk mengetahui pengembangan Aplikasi di Fakultas Vokasi Universitas Negeri Surabaya. Tahap pengembangan dilakukan dengan menerapkan siklus pengembangan sistem yang menyadur model prototyping.

Kata kunci— Aplikasi Pengajuan Surat, Laravel, PHP, Prototyping, Website

Abstract— Mail and archive management is one of the activities that is often handled by an organization, institution, and company. One of the services carried out by the Administration staff of the Faculty of Vocational Studies, Surabaya State University is the request for correspondence. There are six kinds of letters that can be submitted by all academicians within the scope of the Faculty of Vocational Studies, Surabaya State University, which are Surat Keterangan Mahasiswa Aktif, Surat Keterangan Tunjangan Anak pada Gaji Orang Tua, Surat izin Magang KKN Tematik, Surat Izin Magang/PKL, Surat Izin Observasi Mata Kuliah, dan Surat Izin Observasi Tugas Akhir. Currently, the letter submission procedure is still carried out conventionally through templates that are downloaded manually on the faculty website. The development of the letter submission application is intended to make the application governance system more efficient. The development of the letter submission application uses Laravel as the Framework. The use of technology includes the PHP programming language and My SQL to store the database. This study aims to determine the development of applications

at the Faculty of Vocational Studies, Surabaya State University. The development stage is carried out by applying a system development cycle that adapts the prototyping model.

Keywords— Laravel, Letter Submission Application, PHP, Prototyping, Website

I. PENDAHULUAN

Dalam keadaan perkembangan teknologi saat ini, perguruan tinggi harus memiliki berbagai sistem manajemen yang bertujuan untuk memberikan pelayanan prima. Universitas menyediakan layanan ini kepada pihak-pihak berkepentingan yang membutuhkannya. Pihak tersebut di antaranya adalah mahasiswa, pemerintah, dan sivitas akademika lainnya [1].

Universitas memiliki banyak sekali sistem manajemen, seperti sistem manajemen pembelajaran, sistem ujian, sistem pembelajaran online, sistem informasi pembelajaran, dan sistem informasi fakultas [2]. Salah satu contohnya adalah situs pelayanan persuratan melalui Tata Usaha Fakultas Vokasi Universitas Negeri Surabaya.

Saat ini, prosedur pengajuan surat tersebut masih dilakukan secara konvensional melalui template yang diunduh secara manual pada situs fakultas. Berdasarkan situs <https://vokasi.unesa.ac.id/>, jenis naskah tersebut meliputi, Surat Keterangan Mahasiswa Aktif, Surat Keterangan Tunjangan Anak pada Gaji Orang Tua, Surat izin Magang KKN Tematik, Surat Izin Magang/PKL, Surat Izin Observasi Mata Kuliah, Surat Izin Observasi Tugas Akhir, dan Surat Izin Penelitian Tugas Akhir.

Pengembangan aplikasi pengajuan surat menggunakan Laravel sebagai framework. Penggunaan teknologi meliputi bahasa pemrograman PHP dan MySQL untuk menyimpan basis data. Model-View-Controller diambil sebagai arsitektur pada pengembangan sistem ini.

Industri perangkat lunak menggunakan System Development Life Cycle (SDLC) untuk merancang, mengembangkan, dan menghasilkan produk perangkat lunak berkualitas tinggi, andal, hemat biaya, dan tepat waktu [3]. Tahap pengembangan dilakukan dengan menerapkan siklus pengembangan sistem yang menyadur model prototyping. Pengujian selanjutnya adalah pengujian sistem menggunakan Black Box Testing.

Penelitian ini dilakukan untuk mengetahui keberhasilan pengembangan aplikasi pengajuan surat Fakultas Vokasi Universitas Negeri Surabaya. Pengujian dilakukan dengan menggunakan *black box testing*.

Berdasarkan latar belakang permasalahan di atas, dilakukan penelitian dengan judul “Rancang Bangun Sistem Informasi Persuratan berbasis Web di Fakultas Vokasi Universitas Negeri Surabaya”.

II. KAJIAN PUSTAKA

A. Penelitian Terdahulu

Penelitian terdahulu yang akan dikembangkan pada penelitian ini dapat dilihat pada tabel 1.

TABEL I
PENELITIAN TERDAHULU

Judul Penelitian	Peneliti	Hasil	Persamaan dan Perbedaan
<i>Development of E-Office Information System in Administration Faculty of Engineering, State University of Padang</i>	Elsa Sabrina, Dedy Irfan, dan Asrul Huda (2022)	Hasil akhir dari penelitian ini adalah terciptanya sistem informasi <i>E-Office</i> yang menggabungkan berbagai fungsi kepegawaian dan kemahasiswaan.	Penelitian dilatarbelakangi oleh kebutuhan akan sistem informasi yang efektif dan efisien untuk mendukung manajemen data. Metode penelitian yang digunakan adalah <i>DevOps (Development and Operations)</i> dengan bahasa pemrograman <i>PHP Framework Yii2</i> .
<i>Bridal Management System</i>	Nurul Aishah Shairatul ikram dan Noraini Ibrahim (2022)	Dari hasil yang diperoleh, dapat dikatakan bahwa proyek tersebut berhasil sesuai dengan tujuan yang dicapai.	Model <i>Prototyping</i> untuk memenuhi keinginan pengguna dalam pengembangan sistem. Pada tahap implementasi, pengkodean program menggunakan bahasa pemrograman <i>PHP</i> .
<i>Design of E-Office System in Vocational School Diponegoro University using Code Igniter Framework</i>	A Subari, S Manan, dan E Ariyanto (2020)	Hasil yang diperoleh dari pengembangan ini menunjukkan bahwa proses manajemen dokumen dapat diselesaikan dari mana saja. Sehingga, tidak memerlukan waktu yang lama.	Permasalahan yang dikaji adalah pengembangan sistem <i>E-Office</i> . Metode pengembangannya menerapkan konsep MVC yang dipadukan pada <i>Framework CodeIgniter</i> .

Dari studi literatur yang diulas, penelitian yang dilakukan oleh [4] dan [5] menunjukkan pengembangan Aplikasi *E-Office* pada instansi perguruan tinggi. Keduanya menerapkan konsep *Model-View-Controller* pada *framework* yang berbeda. Penelitian [4] menggunakan *Framework Yii*, sedangkan [5]

menggunakan *CodeIgniter*. Berdasarkan penelitian [6], dikatakan bahwa model pengembangan prototipe berhasil membuat suatu proyek sesuai rencana dan jadwal.

B. E-Office

E-Office adalah layanan yang menciptakan data dan informasi melalui media komunikasi berdasarkan konsep *paperless* untuk meminimalkan penggunaan kertas dalam pekerjaan kantor [4]. Dengan menggunakan sistem komputer, hal ini akan memudahkan proses pengolahan data. Penyimpanan data yang sebelumnya berada di lemari khusus berupa buku janji/buku pendaftaran, akan menjadi lebih terstruktur dan rapi. Hal tersebut dikarenakan tempat penyimpanan digital yang disimpan dalam basis data [7].

C. Website

World Wide Web atau yang sering disebut *Web* menggunakan protokol *HTTP (Hypertext Transfer Protocol)* untuk menyediakan cara mengakses informasi di internet. *World Wide Web* adalah platform berbagi informasi berbasis internet yang memungkinkan komunikasi bahasa bebas antara perangkat yang terhubung ke internet. *HTTP* adalah salah satu dari beberapa protokol yang digunakan di internet [8].

D. System Development Life Cycle (SDLC)

Proses memahami bagaimana sistem informasi melayani tujuan bisnis, mendesainnya, membangunnya, dan mengirimkannya ke pengguna disebut *System Development Life Cycle (SDLC)*. Selama fase perencanaan *SDLC*, tim proyek menentukan nilai bisnis sistem, melakukan studi kelayakan, dan menyiapkan proyek. Tim mendefinisikan strategi analitis, mengumpulkan data, dan membangun satu set model analitis pada fase kedua, fase analisis. Pada langkah selanjutnya, fase desain, tim membuat strategi desain, desain fisik, desain arsitektur, desain antarmuka, persyaratan *database* dan file, serta desain program. Membangun, menginstal dan memelihara sistem dilakukan selama fase implementasi. Terdapat tiga metode pengembangan proyek sistem yang populer dalam praktik yang sebenarnya, diantaranya *Waterfall Development, Rapid and Developments (RAD)*, dan *Agile Development* [9].

E. PHP

Hypertext Preprocessor atau yang biasa disebut *PHP* adalah bahasa skrip sisi *server open source*. *PHP* mengeksekusi instruksi pemrograman pada saat *runtime*. Hasil yang ditampilkan tergantung pada data diproses. Sebagai bahasa pemrograman sisi *server*, skrip *PHP* akan diproses di *server*. Jenis *server* sering digunakan bersamaan dengan *PHP* termasuk *Apache, Nginx*, dan *LiteSpeed* [10].

F. Framework

Framework adalah produk kumulatif yang memiliki banyak *library* dan berguna yang dibuat oleh pengembang

lain. *Framework* mengurangi kegiatan tugas yang rumit dan berulang. Dari segi ini, pengembang dapat menulis lebih sedikit [11].

Dengan menggunakan *framework*, maka aplikasi menjadi terstruktur dan terorganisasi dengan jelas. Hal tersebut dikarenakan pembuat *framework* tentunya menggunakan model-model standar, misalnya MVC atau *Model View Controller* yang sudah umum dikenal di kalangan pengembang PHP [12].

G. Laravel

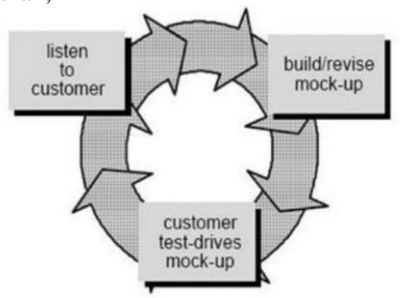
Laravel merupakan salah satu *Framework* PHP yang paling banyak digunakan. Penggunaan *Framework Laravel* untuk mengembangkan desain *web* sangat mudah dan sederhana. Namun, penting halnya mengetahui dasar-dasar PHP untuk memahami struktur dari *framework* tersebut [11].

Laravel membantu mewujudkan ide tanpa membuang kode menggunakan standar pengkodean modern yang dikelilingi oleh komunitas yang bersemangat dan ekosistem alat yang kuat [13].

III. METODOLOGI PENELITIAN

A. Jenis Penelitian

Penelitian ini menggunakan pendekatan kuantitatif. Penelitian kuantitatif dikembangkan untuk mengetahui nilai kebergunaan implementasi sistem melalui evaluasi kegunaan. Lokasi yang digunakan untuk penelitian ini adalah Fakultas Vokasi Universitas Negeri Surabaya. Populasi yang menjadi sasaran merupakan mahasiswa Fakultas Vokasi Universitas Negeri Surabaya. Gambar 2 di bawah ini merupakan bagan rancangan penelitian yang akan digunakan,



Gambar 1 Metodologi Penelitian

B. Tahap Awal

Langkah-langkah penelitian dimulai dengan tahap awal yang meliputi studi literatur pendahuluan dan penentuan metode evaluasi. Secara umum, fokus studi penelitian digali lebih luas pada tahap ini. Kemudian, peneliti melakukan ulasan pada literatur-literatur ilmiah yang linier dengan topik penelitian. Sumber literatur diambil dari referensi nonfisik berupa artikel jurnal internasional ataupun nasional, buku elektronik, buku panduan, dan situs internet.

C. Tahap Pengembangan

Prototyping melakukan proses analisis, desain, dan implementasi secara bersamaan untuk membuat versi singkat dari sistem yang diusulkan dengan cepat, sehingga tersedia bagi pengguna untuk ditinjau dan diberi umpan balik [9]. Pemodelan ini tersusun dari beberapa fase dalam pengaplikasiannya. Fase tersebut dimulai dari Perencanaan, Analisis, Desain, Implementasi, dan Pengujian.

1. Perencanaan

Tahap pertama adalah Perencanaan. Beberapa persiapan sebelum melakukan penelitian antara lain menentukan topik, mengusulkan judul proyek, menentukan ruang lingkup masalah, dan menetapkan tujuan penelitian.

2. Analisis

Setelah perencanaan tersusun, tahap selanjutnya adalah Analisis. Langkah identifikasi juga dilakukan sebagai bagian dari tahap analisis. Proses identifikasi tersebut meliputi identifikasi pemangku kepentingan, identifikasi masalah, dan pembelajaran kapabilitas sistem yang harus ada dalam sistem Aplikasi *E-Office*.

Pada penelitian ini, dua jenis analisis kebutuhan dilakukan saat membangun aplikasi, yaitu kebutuhan data fungsional dan analisis kebutuhan data non-fungsional. Adapun kebutuhan fungsional tersebut meliputi,

- Sistem dapat mendata *user* dan jenis surat
- Sistem dapat menampilkan hasil input
- Sistem dapat memperbaiki data yang telah diinput
- Sistem dapat mengunggah surat
- Sistem dapat mencetak surat

Sedangkan, kebutuhan nonfungsional dalam pengembangan sistem ini meliputi perangkat lunak dan keras seperti pada tabel 2.

TABEL 2
KEBUTUHAN FUNGSIONAL

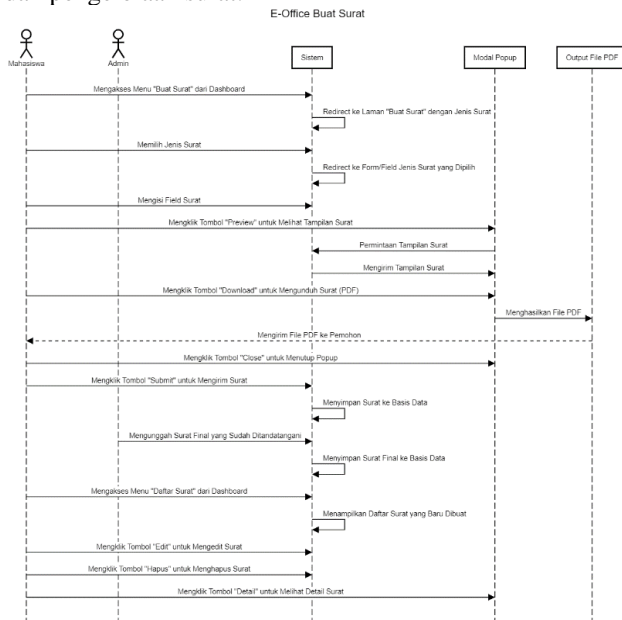
No.	Perangkat Lunak	Perangkat Keras
1.	Sistem operasi Windows	RAM 8 GB
2.	XAMPP	Storage 512 GB
3.	Browser	Jaringan internet
4.	Visual Studio Code	Processor Intel Core i5
5.	Laravel	

Gambar 3 merupakan proses bisnis jalannya sistem pada saat ini. Pengguna harus mengunduh template surat yang diinginkan melalui situs fakultas. Kemudian, pengguna melakukan perubahan informasi data dan mengajukan permohonan melalui *whatsapp*.



Gambar 6 PDM

Gambar 8 menunjukkan *Sequence Diagram* dari proses permohonan surat. Diagram urutan ini menggambarkan alur interaksi antara pemohon, admin, dan sistem *e-office* dalam proses pembuatan, manajemen, dan pengelolaan surat.

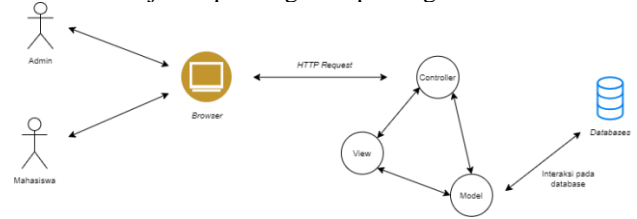


Gambar 7 Sequence Diagram

3. Desain

Berikutnya adalah tahap Desain. Tahap ini menitikberatkan pada pembuatan desain yang sesuai dengan hasil *user requirements* pada tahap sebelumnya.

Ini termasuk desain semua modul antarmuka pengguna, desain sistem, desain basis data dari sistem yang diusulkan, dan redefinisi antarmuka pengguna berdasarkan tinjauan pemangku kepentingan.



Gambar 8 Sistem Desain

4. Tahap Akhir

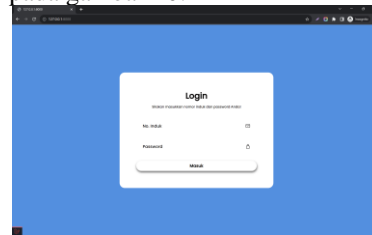
Langkah terakhir dalam proses penelitian ini adalah menarik kesimpulan dan pembuatan laporan. Dari pembahasan dan pengembangan sistem, akan diambil sebuah konklusi hasil berdasarkan data yang relevan. Hal tersebut juga menjadi dasar tersusunnya saran untuk penelitian selanjutnya.

IV. HASIL DAN PEMBAHASAN

A. Implementasi Sistem

Implementasi sistem aplikasi pengajuan surat dikembangkan menggunakan *framework* Laravel dengan bahasa pemrograman PHP. Aplikasi pengajuan surat dibangun untuk membantu pembuatan permohonan surat secara otomatis. Terdapat dua *role* pengguna, yaitu sebagai Admin dan Mahasiswa.

Fitur yang dapat dijalankan pada laman *login* yaitu *Account Login*. Di fitur ini, terdapat *field* "Email" dan "Password". *Field* "Email" dapat diisi dengan alamat email yang digunakan oleh user. *Field* "Password" dapat diisi dengan kata sandi yang sudah diatur sesuai dengan akun *email* masing-masing pengguna. Tampilan *login* ditunjukkan pada gambar 10.



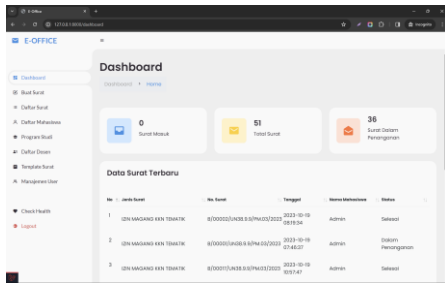
Gambar 9 Laman Login

1. Admin

Role pengguna sebagai Admin memiliki delapan fitur utama, yaitu *Dashboard*, *Buat Naskah*, *Daftar Surat*, *Daftar Mahasiswa*, *Program Studi*, *Daftar Dosen*, *Template Surat*, dan *Manajemen User*.

a. Dashboard

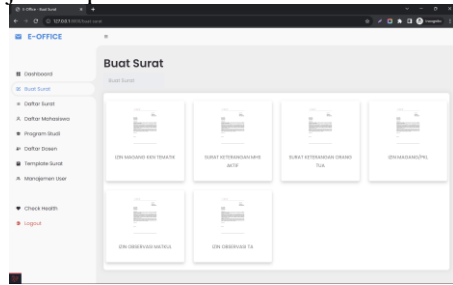
Halaman *Dashboard* untuk Admin menampilkan informasi mengenai statistik total surat. Selain itu, juga menampilkan berapa banyak surat yang masuk dan surat yang sedang dalam proses penanganan. Pengguna juga dapat melihat rincian surat-surat tersebut melalui tabel Data Surat Terbaru. Tampilan *Dashboard* untuk *role* Admin ditunjukkan pada Gambar 11.



Gambar 10 Dashboard Admin

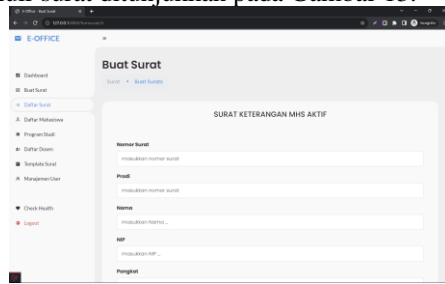
b. Buat Naskah

Halaman Buat Naskah untuk Admin menampilkan daftar jenis surat keterangan dan surat izin. Terdapat enam kartu surat, yaitu surat permohonan Surat Keterangan Mahasiswa Aktif, Surat Keterangan Tunjangan Anak pada Gaji Orang Tua, Surat Izin Magang KKN Tematik, Surat Izin Magang/PKL, Surat Izin Observasi Mata Kuliah, dan Surat Observasi Tugas Akhir. Tampilan Buat Naskah untuk kartu surat ditunjukkan pada Gambar 12.



Gambar 11 Buat Surat Admin

Admin dapat memilih jenis surat mana yang ingin diajukan. Setelah itu, Admin akan diarahkan ke halaman Buat Naskah yang menampilkan formulir surat. Tampilan Buat Naskah untuk formulir surat ditunjukkan pada Gambar 13.



Gambar 12 Formulir Buat Surat Admin

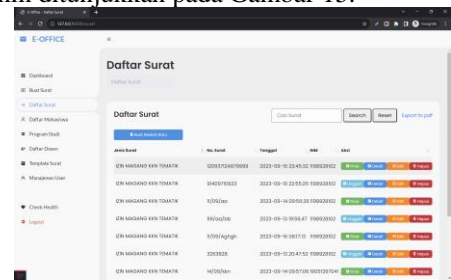
Setelah melakukan *Submit* pada formulir surat, pemohon dapat melihat tinjauan dari surat yang sedang diajukan. Sistem akan menampilkan popup untuk tinjauan surat. Terdapat pula tombol "Download" untuk mengunduh draft surat. Tampilan *popup* tinjauan surat ditunjukkan pada Gambar 14.



Gambar 13 PopUp Tinjauan Buat Surat Admin

c. Daftar Surat

Halaman Daftar Surat untuk Admin menampilkan informasi mengenai Riwayat surat-surat yang masuk. Terdapat tiga bentuk aksi yang dapat dilakukan, yaitu pada tombol "Detail", "Edit", dan "Hapus". Tampilan Daftar Surat untuk Admin ditunjukkan pada Gambar 15.



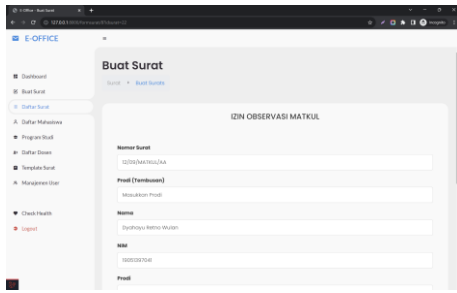
Gambar 14 Daftar Surat Admin

Admin dapat melihat kembali tinjauan dari surat yang diajukan melalui tombol "Detail". Sistem akan menampilkan *popup* untuk tinjauan surat. Terdapat pula tombol "Unduh" untuk mengunduh draft surat. Dari sisi Admin, terdapat fitur yang memungkinkan Admin dalam mengunggah surat yang sudah final, yaitu surat yang telah ditandatangani oleh tertanda di masing-masing surat. Tampilan tinjauan surat untuk Admin ditunjukkan pada Gambar 16.



Gambar 15 Tinjauan Detail Surat Admin

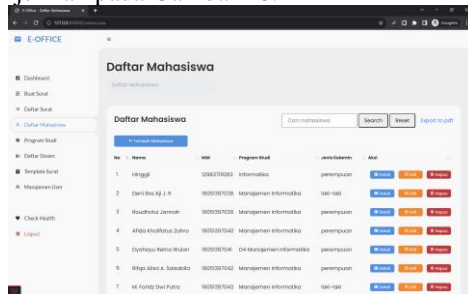
Admin dapat pula melakukan edit pada surat yang diajukan melalui tombol "Edit". Pengguna akan diarahkan ke halaman Buat Naskah yang menampilkan formulir surat. Pada form ini, beberapa *field* sudah terisi dan bisa diedit sesuai kebutuhan. Tampilan Edit Surat pada formulir surat ditunjukkan pada Gambar 17.



Gambar 16 Edit Surat Admin

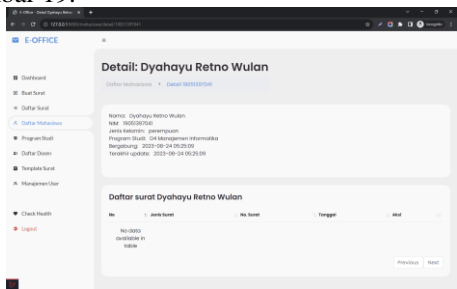
d. Daftar Mahasiswa

Halaman Daftar Mahasiswa sebagai master data menampilkan informasi data mahasiswa. Terdapat tiga bentuk aksi yang dapat dilakukan, yaitu pada tombol “Detail”, “Edit”, dan “Hapus”. Tampilan Daftar Mahasiswa untuk Admin ditunjukkan pada Gambar 18.



Gambar 17 Daftar Mahasiswa

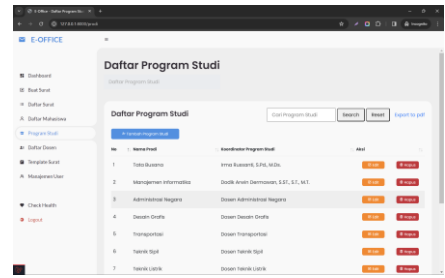
Pengguna dapat melihat detail mahasiswa melalui tombol “Detail”. Pengguna diarahkan ke laman Detail [Nama Mahasiswa]. Halaman ini menampilkan informasi dari mahasiswa, meliputi NIM, dan program studi. Tampilan Detail Mahasiswa untuk Admin ditunjukkan pada Gambar 19.



Gambar 18 Detail Mahasiswa

e. Program Studi

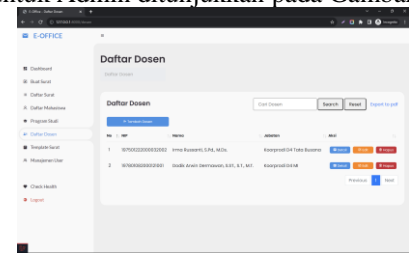
Program Studi sebagai *master data* menampilkan informasi data program studi. Terdapat dua bentuk aksi yang dapat dilakukan, yaitu pada tombol “Edit” dan “Hapus”. Tampilan Program Studi untuk Admin ditunjukkan pada Gambar 20.



Gambar 19 Daftar Program Studi

f. Daftar Dosen

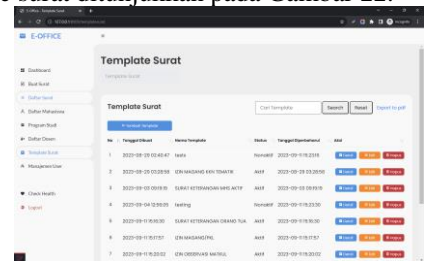
Daftar Dosen sebagai master data menampilkan informasi data dosen. Terdapat dua bentuk aksi yang dapat dilakukan, yaitu pada tombol “Edit” dan “Hapus”. Tampilan Daftar Dosen untuk Admin ditunjukkan pada Gambar 21.



Gambar 20 Daftar Dosen

g. Template Surat

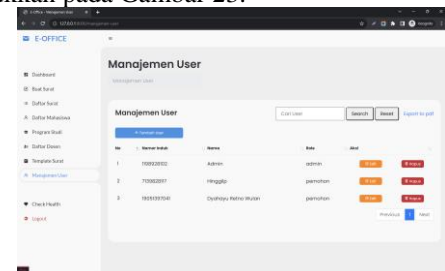
Template Surat sebagai master data menampilkan informasi data surat. Terdapat tiga bentuk aksi yang dapat dilakukan, yaitu pada tombol “Detail”, “Edit”, dan “Hapus”. Tampilan template surat ditunjukkan pada Gambar 22.



Gambar 21 Template Surat

h. Manajemen User

Manajemen User sebagai master data menampilkan informasi *user*. Terdapat dua bentuk aksi yang dapat dilakukan, yaitu pada tombol “Edit” dan “Hapus”. Tampilan Manajemen User ditunjukkan pada Gambar 23.

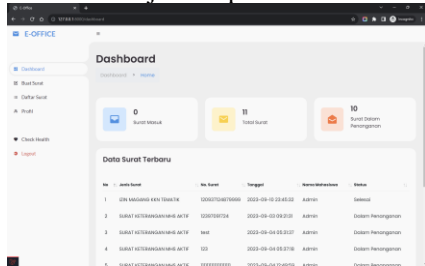


Gambar 22 Manajemen User

2. Mahasiswa
 Role pengguna sebagai Mahasiswa memiliki tiga fitur utama, yaitu *Dashboard*, *Buat Naskah*, dan *Daftar Surat*.

a. *Dashboard*

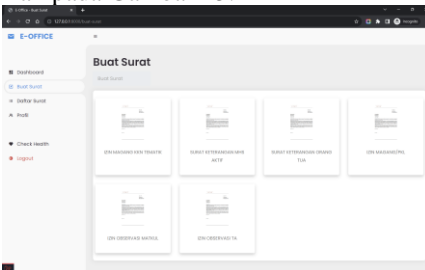
Halaman *Dashboard* untuk Mahasiswa menampilkan informasi mengenai statistik total surat. Selain itu, juga menampilkan berapa banyak surat yang masuk dan surat yang sedang dalam proses penanganan. Tampilan *Dashboard* untuk role Mahasiswa ditunjukkan pada Gambar 24.



Gambar 23 Dashboard Mahasiswa

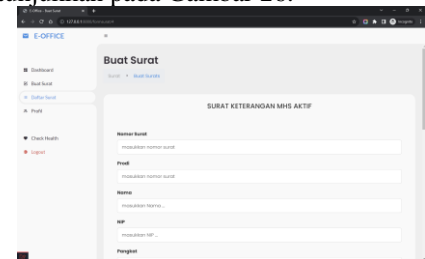
b. *Buat Naskah*

Halaman *Buat Naskah* untuk *User* menampilkan daftar jenis surat keterangan dan surat izin. Terdapat enam kartu surat, yaitu surat permohonan Surat Keterangan Mahasiswa Aktif, Surat Keterangan Tunjangan Anak pada Gaji Orang Tua, Surat Izin Magang KKN Tematik, Surat Izin Magang/PKL, Surat Izin Observasi Mata Kuliah, dan Surat Observasi Tugas Akhir. Tampilan *Buat Naskah* untuk kartu surat ditunjukkan pada Gambar 25.



Gambar 24 Buat Surat Mahasiswa

Mahasiswa dapat memilih jenis surat mana yang ingin diajukan. Setelah itu, *user* akan diarahkan ke halaman *Buat Naskah* yang menampilkan formulir surat. Pada form ini, terdapat beberapa *field* dari template surat yang harus diisi. Tampilan *Buat Naskah* untuk formulir surat ditunjukkan pada Gambar 26.



Gambar 25 Formulir Buat Surat Mahasiswa

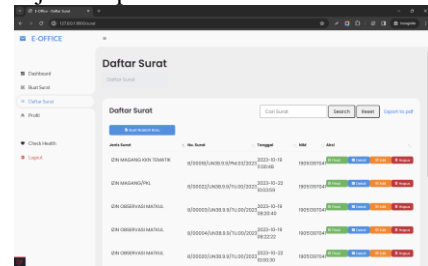
Setelah melakukan *Submit* pada formulir surat, pengguna atau pemohon dapat melihat tinjauan dari surat yang sedang diajukan. Sistem akan menampilkan *popup* untuk tinjauan surat. Terdapat pula tombol “Unduh” untuk mengunduh draft surat. Tampilan *popup* tinjauan surat ditunjukkan pada Gambar 27.



Gambar 26 Popup Buat Surat Mahasiswa

c. *Daftar Surat*

Halaman *Daftar Surat* untuk *User* menampilkan informasi mengenai Riwayat surat-surat yang masuk. Terdapat tiga bentuk aksi yang dapat dilakukan, yaitu pada tombol “Detail”, “Edit”, dan “Hapus”. Tampilan *Daftar Surat* untuk *User* ditunjukkan pada Gambar 28.



Gambar 27 Daftar Surat Mahasiswa

Pengujian fungsionalitas dilakukan dengan *Black Box Testing*. Pengujian ini menggunakan skenario uji yang didasarkan pada setiap fitur dari aplikasi. Selain itu, terdapat pula obyektif atau hasil yang diharapkan dari suatu kasus uji. Dari pengujian *Black Box Testing*, dapat diambil hasil apakah fungsi dari aplikasi sudah berhasil dijalankan dengan baik. Tabel 4 merupakan hasil pengujian dengan *Black Box Testing*.

TABEL 4
 PENGUJIAN BLACK BOX

No.	Kasus Uji	Obyektif yang Diharapkan	Hasil
1.	Masukkan <i>Email</i> dan <i>Password</i> yang sesuai.	Berhasil masuk ke laman <i>dashboard</i> .	Berhasil
2.	Mengosongkan <i>Field</i> Email dan <i>Password</i> .	Tidak berhasil masuk ke laman <i>Dashboard</i> .	Berhasil
3.	Mengosongkan salah satu <i>field</i> pada laman Login.	Tidak berhasil masuk ke laman <i>Dashboard</i> .	Berhasil

TABEL 5
 PENGUJIAN BLACK BOX DASHBOARD

No.	Kasus Uji	Obyektif yang Diharapkan	Hasil
1.	Klik menu <i>Dashboard</i> .	Berhasil menampilkan laman <i>Dashboard</i> .	Berhasil
2.	Klik “Sort”	Berhasil mengurutkan	Berhasil

		data.	
3.	Klik <i>pagination</i>	Berhasil menampilkan data sesuai dengan laman yang dipilih.	Berhasil

	pada <i>popup</i> peringatan.	data.	
16.	Klik tombol "Batal" pada <i>popup</i> peringatan.	Berhasil menutup <i>popup</i> .	Berhasil

TABEL 6
PENGUJIAN BLACK BOX BUAT NASKAH

No.	Kasus Uji	Obyektif yang Diharapkan	Hasil
1.	Klik menu Buat Naskah.	Berhasil menampilkan laman Buat Naskah.	Berhasil
2.	Klik jenis surat	Berhasil menampilkan laman form permohonan surat	Berhasil
3.	Masukkan data pada setiap <i>field</i> permohonan surat.	Berhasil menampilkan <i>field</i> yang sudah terisi dengan data yang sesuai	Berhasil
4.	Klik tombol "Submit".	Berhasil menampilkan <i>preview</i> surat.	Berhasil
5.	Klik tombol "Unduh" pada <i>preview</i> surat.	Berhasil mengunduh file surat.	Berhasil

TABEL 7
PENGUJIAN BLACK BOX DAFTAR SURAT

No.	Kasus Uji	Obyektif yang Diharapkan	Hasil
1.	Klik menu "Daftar Surat".	Berhasil menampilkan laman Daftar Surat.	Berhasil
2.	Klik tombol "Tambah Daftar Surat".	Berhasil menampilkan laman form Buat Naskah.	Berhasil
3.	Masukkan kata kunci surat yang dicari pada <i>field</i> pencarian.	Berhasil menampilkan data yang dicari.	Berhasil
4.	Klik tombol "Reset".	Berhasil menghapus <i>input</i> pada <i>field</i> pencarian.	Berhasil
5.	Klik tombol "Export to PDF".	Berhasil mengunduh data dalam bentuk file PDF.	Berhasil
6.	Klik "Sort"	Berhasil mengurutkan data.	Berhasil
7.	Klik <i>pagination</i>	Berhasil menampilkan data sesuai dengan laman yang dipilih.	Berhasil
8.	Klik tombol "Detail"	Berhasil menampilkan detail surat.	Berhasil
9.	Klik tombol "Unduh" pada <i>preview</i> surat.	Berhasil mengunduh file surat.	Berhasil
10.	Klik <i>field</i> untuk upload file.	Berhasil mengunggah file.	Berhasil
11.	Klik tombol "Edit".	Berhasil membuka form surat yang diajukan.	
12.	Masukkan perubahan pada salah satu <i>field</i> .	Berhasil mengubah isian <i>field</i> .	
13.	Klik tombol "Submit".	Berhasil menampilkan <i>preview</i> surat.	
14.	Klik tombol "Hapus".	Menampilkan <i>popup</i> peringatan.	Berhasil
15.	Klik tombol "Ok"	Berhasil menghapus	Berhasil

TABEL 8
PENGUJIAN BLACK BOX DAFTAR MAHASISWA

No.	Kasus Uji	Obyektif yang Diharapkan	Hasil
1.	Klik menu "Daftar Mahasiswa".	Berhasil menampilkan laman Daftar Mahasiswa.	Berhasil
2.	Klik tombol "Tambah Daftar Mahasiswa".	Berhasil menampilkan laman form Tambah Daftar Mahasiswa.	Berhasil
3.	Masukkan kata kunci surat yang dicari pada <i>field</i> pencarian.	Berhasil menampilkan data yang dicari.	Berhasil
4.	Klik tombol "Reset".	Berhasil menghapus <i>input</i> pada <i>field</i> pencarian.	Berhasil
5.	Klik tombol "Export to PDF".	Berhasil mengunduh data dalam bentuk file PDF.	Berhasil
6.	Klik "Sort"	Berhasil mengurutkan data.	Berhasil
7.	Klik <i>pagination</i>	Berhasil menampilkan data sesuai dengan laman yang dipilih.	Berhasil
8.	Klik tombol "Detail Mahasiswa"	Berhasil menampilkan detail mahasiswa.	Berhasil
9.	Klik tombol "Edit".	Berhasil membuka form data mahasiswa.	Berhasil
10.	Masukkan perubahan pada salah satu <i>field</i> .	Berhasil mengubah isian <i>field</i> .	Berhasil
11.	Klik tombol "Submit".	Berhasil menampilkan data mahasiswa yang baru.	Berhasil
12.	Klik tombol "Hapus".	Menampilkan <i>popup</i> peringatan.	Berhasil
13.	Klik tombol "Ok" pada <i>popup</i> peringatan.	Berhasil menghapus data.	Berhasil
14.	Klik tombol "Batal" pada <i>popup</i> peringatan.	Berhasil menutup <i>popup</i> .	Berhasil

TABEL 9
PENGUJIAN BLACK BOX DAFTAR PROGRAM STUDI

No.	Kasus Uji	Obyektif yang Diharapkan	Hasil
1.	Klik menu "Daftar Program Studi".	Berhasil menampilkan laman Daftar Program Studi.	Berhasil
2.	Klik tombol "Tambah Program Studi".	Berhasil menampilkan laman form Tambah Program Studi.	Berhasil
3.	Masukkan kata kunci surat yang dicari pada <i>field</i> pencarian.	Berhasil menampilkan data yang dicari.	Berhasil
4.	Klik tombol "Reset".	Berhasil menghapus <i>input</i> pada <i>field</i> pencarian.	Berhasil
5.	Klik tombol "Export to PDF".	Berhasil mengunduh data dalam bentuk file PDF.	Berhasil

6.	Klik "Sort"	Berhasil mengurutkan data.	Berhasil
7.	Klik <i>pagination</i>	Berhasil menampilkan data sesuai dengan laman yang dipilih.	Berhasil
8.	Klik tombol "Edit".	Berhasil membuka form data program studi.	Berhasil
9.	Masukkan perubahan pada salah satu <i>field</i> .	Berhasil mengubah isian <i>field</i> .	Berhasil
10.	Klik tombol "Submit".	Berhasil menampilkan data program studi yang baru.	Berhasil
11.	Klik tombol "Hapus".	Menampilkan <i>popup</i> peringatan.	Berhasil
12.	Klik tombol "Ok" pada <i>popup</i> peringatan.	Berhasil menghapus data.	Berhasil
13.	Klik tombol "Batal" pada <i>popup</i> peringatan.	Berhasil menutup <i>popup</i> .	Berhasil

TABEL 10
PENGUJIAN BLACK BOX DAFTAR DOSEN

No.	Kasus Uji	Obyektif yang Diharapkan	Hasil
1.	Klik menu "Daftar Dosen".	Berhasil menampilkan laman Daftar Dosen.	Berhasil
2.	Klik tombol "Tambah Daftar Dosen".	Berhasil menampilkan laman form Tambah Daftar Dosen.	Berhasil
3.	Masukkan kata kunci surat yang dicari pada <i>field</i> pencarian.	Berhasil menampilkan data yang dicari.	Berhasil
4.	Klik tombol "Reset".	Berhasil menghapus <i>input</i> pada <i>field</i> pencarian.	Berhasil
5.	Klik tombol "Export to PDF".	Berhasil mengunduh data dalam bentuk file PDF.	Berhasil
6.	Klik "Sort"	Berhasil mengurutkan data.	Berhasil
7.	Klik <i>pagination</i>	Berhasil menampilkan data sesuai dengan laman yang dipilih.	Berhasil
8.	Klik tombol "Edit".	Berhasil membuka form data dosen.	Berhasil
9.	Masukkan perubahan pada salah satu <i>field</i> .	Berhasil mengubah isian <i>field</i> .	Berhasil
10.	Klik tombol "Submit".	Berhasil menampilkan data dosen yang baru.	Berhasil
11.	Klik tombol "Hapus".	Menampilkan <i>popup</i> peringatan.	Berhasil
12.	Klik tombol "Ok" pada <i>popup</i> peringatan.	Berhasil menghapus data.	Berhasil
13.	Klik tombol "Batal" pada <i>popup</i> peringatan.	Berhasil menutup <i>popup</i> .	Berhasil

TABEL 11
PENGUJIAN BLACK BOX TEMPLATE SURAT

No.	Kasus Uji	Obyektif yang Diharapkan	Hasil
1.	Klik menu "Template Surat".	Berhasil menampilkan laman Template Surat.	Berhasil
2.	Klik tombol "Tambah	Berhasil menampilkan laman form Tambah	Berhasil

No.	Kasus Uji	Obyektif yang Diharapkan	Hasil
3.	Masukkan kata kunci surat yang dicari pada <i>field</i> pencarian.	Berhasil menampilkan data yang dicari.	Berhasil
4.	Klik tombol "Reset".	Berhasil menghapus <i>input</i> pada <i>field</i> pencarian.	Berhasil
5.	Klik tombol "Export to PDF".	Berhasil mengunduh data dalam bentuk file PDF.	Berhasil
6.	Klik "Sort"	Berhasil mengurutkan data.	Berhasil
7.	Klik <i>pagination</i>	Berhasil menampilkan data sesuai dengan laman yang dipilih.	Berhasil
8.	Klik tombol "Edit".	Berhasil membuka form template surat.	Berhasil
9.	Masukkan perubahan pada salah satu <i>field</i> .	Berhasil mengubah isian <i>field</i> .	Berhasil
10.	Klik tombol "Submit".	Berhasil menampilkan template surat yang baru.	Berhasil
11.	Klik tombol "Hapus".	Menampilkan <i>popup</i> peringatan.	Berhasil
12.	Klik tombol "Ok" pada <i>popup</i> peringatan.	Berhasil menghapus data.	Berhasil
13.	Klik tombol "Batal" pada <i>popup</i> peringatan.	Berhasil menutup <i>popup</i> .	Berhasil

TABEL 12
PENGUJIAN BLACK BOX MANAJEMEN SURAT

No.	Kasus Uji	Obyektif yang Diharapkan	Hasil
1.	Klik menu "Template Surat".	Berhasil menampilkan laman Template Surat.	Berhasil
2.	Klik tombol "Tambah Template Surat".	Berhasil menampilkan laman form Tambah Template Surat.	Berhasil
3.	Masukkan kata kunci surat yang dicari pada <i>field</i> pencarian.	Berhasil menampilkan data yang dicari.	Berhasil
4.	Klik tombol "Reset".	Berhasil menghapus <i>input</i> pada <i>field</i> pencarian.	Berhasil
5.	Klik tombol "Export to PDF".	Berhasil mengunduh data dalam bentuk file PDF.	Berhasil
6.	Klik "Sort"	Berhasil mengurutkan data.	Berhasil
7.	Klik <i>pagination</i>	Berhasil menampilkan data sesuai dengan laman yang dipilih.	Berhasil
8.	Klik tombol "Edit".	Berhasil membuka form template surat.	Berhasil
9.	Masukkan perubahan pada salah satu <i>field</i> .	Berhasil mengubah isian <i>field</i> .	Berhasil
10.	Klik tombol "Submit".	Berhasil menampilkan template surat yang baru.	Berhasil
11.	Klik tombol "Hapus".	Menampilkan <i>popup</i> peringatan.	Berhasil
12.	Klik tombol "Ok" pada <i>popup</i> peringatan.	Berhasil menghapus data.	Berhasil
13.	Klik tombol "Batal" pada <i>popup</i> peringatan.	Berhasil menutup <i>popup</i> .	Berhasil

	“Batal” pada <i>popup</i> peringatan.	<i>popup</i> .	
--	--	----------------	--

V. KESIMPULAN

A. Kesimpulan

Berdasarkan uraian dan pembahasan hasil penelitian, dapat ditarik kesimpulan bahwa:

1. Pengembangan sistem yang dapat membantu manajemen operasional dalam pengarsipan surat menyurat. Dengan metode prototyping, dapat membantu pengembangan aplikasi dengan lebih cepat dan baik. Beberapa hal yang bisa dilakukan adalah mendata user dan jenis surat; menampilkan hasil input dari form pembuatan surat; memperbaiki data yang telah diinput; mengunggah surat; serta mencetak surat.

2. Penelitian ini menguji kegunaan aplikasi pengajuan surat berbasis *website* untuk pengiriman surat dengan *black box testing*. Hasil yang didapatkan adalah semua fitur yang diujikan berhasil dijalankan dengan baik.

B. Saran

Berdasarkan analisis pembahasan yang dilakukan, penulis mencatat beberapa saran untuk perbaikan. Penelitian yang dilakukan masih jauh dari sempurna. Ada banyak hal yang dapat dikembangkan dan diperbaiki. Pada penelitian selanjutnya, penulis berharap beberapa menu pada aplikasi dapat diujicobakan secara luas dengan *usability testing*. Salah satu standar yang dapat digunakan adalah ISO 9241-11.

REFERENSI

- [1] K. Budiman, A. T. Putra, Alamsyah, E. Sugiharti, M. A. Muslim, and R. Arifudin, “Implementation of ERP system functionalities for data acquisition based on API at the study program of Universities,” in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Jun. 2021. doi: 10.1088/1742-6596/1918/4/042151.
- [2] A. Kumar, “Role of Information Systems in an University Setup-A Case Study,” 2016. [Online]. Available: <https://www.researchgate.net/publication/317649791>
- [3] N. Dwivedi, D. Katiyar, and G. Goel, “A Comparative Study of Various Software Development Life Cycle (SDLC) Models,” *International Journal of Research in Engineering, Science and Management*, vol. 5, no. 3, pp. 141–144, 2022, [Online]. Available: <https://www.ijresm.com>
- [4] E. Sabrina, D. Irfan, and A. Huda, “Development of E-Office Information System in Administration Faculty of Engineering, State University of Padang,” *Jurnal Paedagogy*, vol. 9, no. 3, p. 364, Jul. 2022, doi: 10.33394/jp.v9i3.5313.
- [5] A. Subari, S. Manan, and E. Ariyanto, “Design of E-office system in vocational school Diponegoro University using code igniter framework,” in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Jun. 2020. doi: 10.1088/1757-899X/801/1/012141.
- [6] N. A. Shairatulikram and N. Ibrahim, “Bridal Management System,” *Applied Information Technology And Computer Science*, vol. 3, no. 1, pp. 1253–1271, 2022, doi: 10.30880/aitcs.2022.03.01.082.
- [7] M. A. Lestari, M. Tabrani, and S. Ayumida, “SISTEM INFORMASI PENGOLAHAN DATA ADMINISTRASI KEPENDUDUKAN PADA KANTOR DESA PUCUNG KARAWANG,” 2018.
- [8] K. Wood, *Confident Web Design Master the Fundamentals of Website Creation and Supercharge Your Career*. 2018. Accessed: Jan. 23, 2023. [Online]. Available: https://www.google.co.id/books/edition/Confident_Web_Design/GMBDDwAAQBAJ?hl=id&gbpv=1
- [9] A. Dennis, B. H. Wixom, and R. M. Roth, *Systems Analysis & Design UML 5th ed*. 2021.
- [10] Muqorobin and N. A. R. Rais, “Comparison of PHP Programming Language with Codeigniter Framework in Project CRUD,” *International Journal of Computer and Information System (IJCIS)*, pp. 94–98, 2022, [Online]. Available: <https://ijcis.net/index.php/ijcis/index>
- [11] N. Yadav, D. S. Rajpoot, and S. K. Dhakad, “LARAVEL: A PHP Framework for E-Commerce Website,” *2019 Fifth International Conference on Image Information Processing (ICIIP)*, 2019.
- [12] Y. Yudhanto and H. A. Prasetyo, *Panduan Mudah Belajar Framework Laravel*. 2018.
- [13] M. Stauffer, *Laravel : up and running : a framework for building modern PHP apps*. 2016.