

Rancang Bangun Backend Sistem Pemantauan Dokumen Akreditasi Menggunakan Kerangka Kerja Laravel Berbasis RESTful API

Design and develop a Laravel-based RESTful API framework for monitoring accreditation documents in the backend system.

Galih Saputra*, Eko Didik Widiyanto, Adian Fatchur Rochim

Departemen Teknik Komputer, Fakultas Teknik, Universitas Diponegoro
Jl. Prof. Soedarto, SH, Kampus Undip Tembalang, Semarang, Indonesia 50275

How to cite: G. Saputra, E. D. Widiyanto, and A. F. Rochim, "Rancang Bangun Backend Sistem Pemantauan Dokumen Akreditasi Menggunakan Kerangka Kerja Laravel Berbasis RESTful API," *Jurnal Teknik Komputer*, vol. 2, no. 1, pp. 91-98, 2023. doi: 10.14710/jtk.v2i1.38320[Online].

Abstract – *The document monitoring system must be reliable and capable of handling user errors. The system is developed using the waterfall method, which includes requirements analysis, system design, implementation, testing, and evaluation. The system is built using the Laravel framework and is based on a RESTful API. The system is capable of performing add, view, edit, delete, and approval operations. Additionally, there are validations that can minimize user input errors into the system. These operations have been tested using blackbox testing and have achieved the expected results.*

Keywords – *Document; Laravel; RESTful API; Validation;*

Abstrak – *Sistem pemantauan dokumen harus mampu mengelola data yang ada dan menanggulangi galat oleh pengguna. Sistem ini disusun dengan metode waterfall yang meliputi analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan evaluasi. Sistem ini disusun dengan menggunakan kerangka kerja Laravel dan berbasis RESTful API. Sistem ini mampu melakukan operasi tambah, lihat, edit, hapus, dan persetujuan. Selain itu, terdapat validasi yang dapat meminimalkan galat masukan oleh pengguna ke sistem. Operasi tersebut sudah diuji dengan pengujian blackbox dan mendapatkan hasil yang sesuai dengan harapan.*

Kata kunci – *Dokumen; Laravel; RESTful API; Validasi;*

I. PENDAHULUAN

Perkembangan Teknologi membawa perubahan yang pesat. Sistem informasi diciptakan untuk memudahkan pengelolaan data yang ada. Perusahaan, organisasi, sekolah, dan kampus membutuhkan sistem informasi dalam mengelola data yang ada.

Dokumen resmi seperti dokumen akreditasi perlu dikelola agar tertata dan dapat dicari dengan mudah ketika sedang dibutuhkan. Selain itu, dokumen juga perlu didata agar mudah dalam mengelola dan mengenali

*) Corresponding author (Galih Saputra)

Email: gatra@students.undip.ac.id

setiap dokumennya. Karena seiring bertambah tahun dokumen akreditasi akan terus bertambah.

Proses akreditasi sering kali menyita waktu, tenaga dan pikiran. Akar permasalahannya adalah bukti pendukung akreditasi sering tercecer dimana-mana. Tercecer yang dimaksud baik secara fisik ataupun berkas digital. Asesor akan memberikan nilai yang baik jika Institusi Pendidikan atau Program Studi dapat memberikan bukti jika diminta. Aplikasi manajemen dokumen dapat menjadi solusi untuk menyimpan dan mencari dokumen yang dimaksud [Ardian].

Sistem pemantauan dokumen akreditasi merupakan penerapan dari pengelolaan data. Pengelolaan data tersebut dapat berupa tambah, ubah, lihat, hapus, dan cari. Selain itu, sistem juga memerlukan beberapa peran dalam mengelola dokumen. Seperti peran *Admin*, *Pengguna*, *Operator*, *Supervisor*, *Manajer*, dan masih banyak lagi, dengan setiap peran memiliki tugas masing-masing.

Sistem juga memerlukan keamanan akses data yang berlapis. Seperti Autentikasi dan Otorisasi, dengan setiap peran memiliki hak manajemen dalam sistem. Keamanan berlapis juga berupa validasi form masukkan dari pengguna. Hal ini akan membantu pengguna jika terjadi kesalahan dalam memasukkan data ke sistem. Adanya sistem ini, pengguna diharapkan dapat mengelola data dengan risiko kesalahan yang minim.

Operasi pengelolaan dan validasi dilakukan pada sisi backend yang mana berhubungan dengan basis data. Penelitian terkait pengelolaan dokumen sudah pernah dilakukan oleh Ardian dan Bambang pada tahun 2022. Namun dalam penelitian tersebut membahas pengelolaan dokumen menggunakan sistem aplikasi dan aplikasi yang dibuat belum membagi sisi *frontend* dan *backend* [1].

O. D. Arianto dan Y. A. Susetyo tahun 2022 membangun sistem informasi berbasis RESTful API untuk mengelola data sumber daya manusia. Dijelaskan pada penelitian tersebut bahwa dilakukan kolaborasi Web Service REST dengan *library* dari Laravel yakni Laravel Passport untuk mengamankan proses pertukaran data [2].

E. R. Amirussalam, W. H. N. Putra dan W. Purnomo pada tahun 2022 membangun e-commerce penjualan

tanaman anggrek berbasis RESTful API menggunakan laravel. Tujuan dari penelitian tersebut adalah membangun sistem yang dapat dikembangkan lebih lanjut dengan menggabungkan komponen-komponen dari *framework* lain agar lebih cepat dan efektif [3].

Dari tiga penelitian yang sudah disebutkan tidak ada yang fokus terhadap bagaimana mengurangi galat masukkan oleh pengguna ketika mengelola data. Sedangkan penelitian ini berfokus pada pengelolaan data serta meminimalkan galat masukkan oleh pengguna agar pengelolaan menjadi lebih efektif dan efisien.

III. METODE PENELITIAN

Penelitian ini disusun menggunakan metode waterfall. Metode Waterfall atau sering disebut Metode air terjun menggunakan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dari analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan evaluasi [4].

A. Analisis Kebutuhan

Kebutuhan sistem terdiri atas kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional menjelaskan terkait kinerja sistem pada saat masukkan tertentu. Sedangkan kebutuhan non-fungsional adalah kebutuhan yang diperlukan untuk membantu pembuatan sistem, meliputi kebutuhan perangkat keras dan perangkat lunak.

Adapun kebutuhan fungsional dari sistem ini dibagi atas beberapa peran, yaitu *admin*, *supervisor*, *manajer*, *operator*, dan *pengguna*.

Berikut merupakan kebutuhan fungsional aplikasi pada sisi *admin*:

1. *Admin* dapat masuk ke dalam sistem melalui proses *login*.
2. *Admin* dapat melihat, mengubah, menambah, menghapus, dan mencari data terkait pengguna.
3. *Admin* dapat keluar sistem melalui proses *logout*.

Berikut merupakan kebutuhan fungsional aplikasi pada sisi *supervisor*:

1. *Supervisor* dapat masuk ke dalam sistem melalui proses *login*.
2. *Supervisor* dapat menolak dan menyetujui penghapusan data kecuai data pengguna.
3. *Supervisor* dapat keluar sistem melalui proses *logout*.

Berikut merupakan kebutuhan fungsional aplikasi pada sisi *manajer*:

1. *Manajer* dapat masuk ke dalam sistem melalui proses *login*.
2. *Manajer* hanya dapat melihat dan mencari data kecuai data pengguna.
3. *Manajer* dapat keluar sistem melalui proses *logout*.

Berikut merupakan kebutuhan fungsional aplikasi pada sisi *operator*:

1. *Operator* dapat masuk ke dalam sistem melalui proses *login*.
2. *Operator* dapat melihat, mengubah, menambah, menghapus, dan mencari data kecuai data pengguna.

3. *Operator* dapat keluar sistem melalui proses *logout*. Berikut merupakan kebutuhan fungsional aplikasi pada sisi pengguna:

1. Pengguna dapat masuk ke dalam sistem melalui proses *login*.
2. Pengguna hanya dapat mencari letak dokumen yang terdaftar pada sistem.
3. Pengguna dapat keluar sistem melalui proses *logout*.

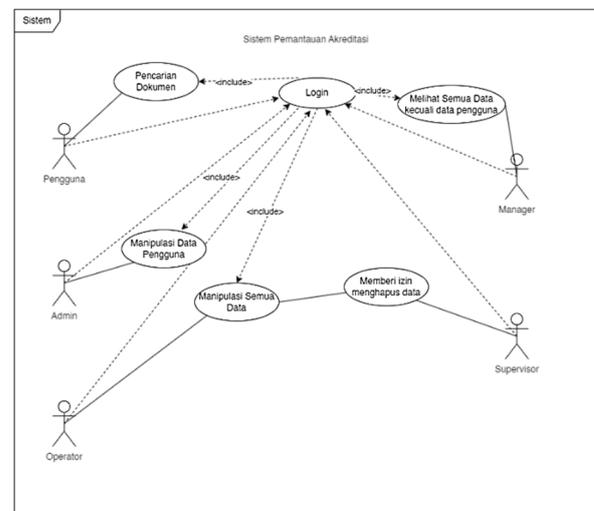
Selain kebutuhan fungsional, sistem juga memerlukan kebutuhan non-fungsional, meliputi kebutuhan perangkat keras yakni komputer atau laptop dengan spesifikasi *hardware* Processor Intel Core i3 Generasi 4 atau di atasnya, RAM 4 GB, SSD 250 GB SATA serta perangkat *peripheral* monitor, tetikus, dan papan ketik.

Adapun kebutuhan perangkat lunak dari sistem ini, meliputi Sistem Operasi Windows 11 64-bit sebagai sistem operasi untuk menjalankan sistem ini. Google Chrome sebagai peramban untuk menjalankan sistem, dan Visual Studio Code sebagai perangkat lunak pengembangan kode sistem.

B. Perancangan Sistem

Perancangan sistem ini direpresentasikan dengan dua macam diagram, yaitu diagram *use case* dan ERD (*Entity Relational Diagram*).

Diagram *use case* berfungsi untuk menjelaskan bagaimana seseorang dapat menggunakan sistem yang menjelaskan proses yang dilakukan pengguna terhadap sistem tersebut. Fungsi dari diagram *use case* adalah untuk menunjukkan proses kegiatan secara berurutan dalam sistem, mampu menggambarkan proses bisnis, mampu menampilkan kegiatan dalam suatu proses dan sebagai jembatan antara pengembang perangkat lunak dan klien untuk menggambarkan sebuah sistem. Berikut gambar dari diagram *use case* [5]. Berikut diagram *use case* dari sistem.



Gambar 1. Diagram *use case* sistem

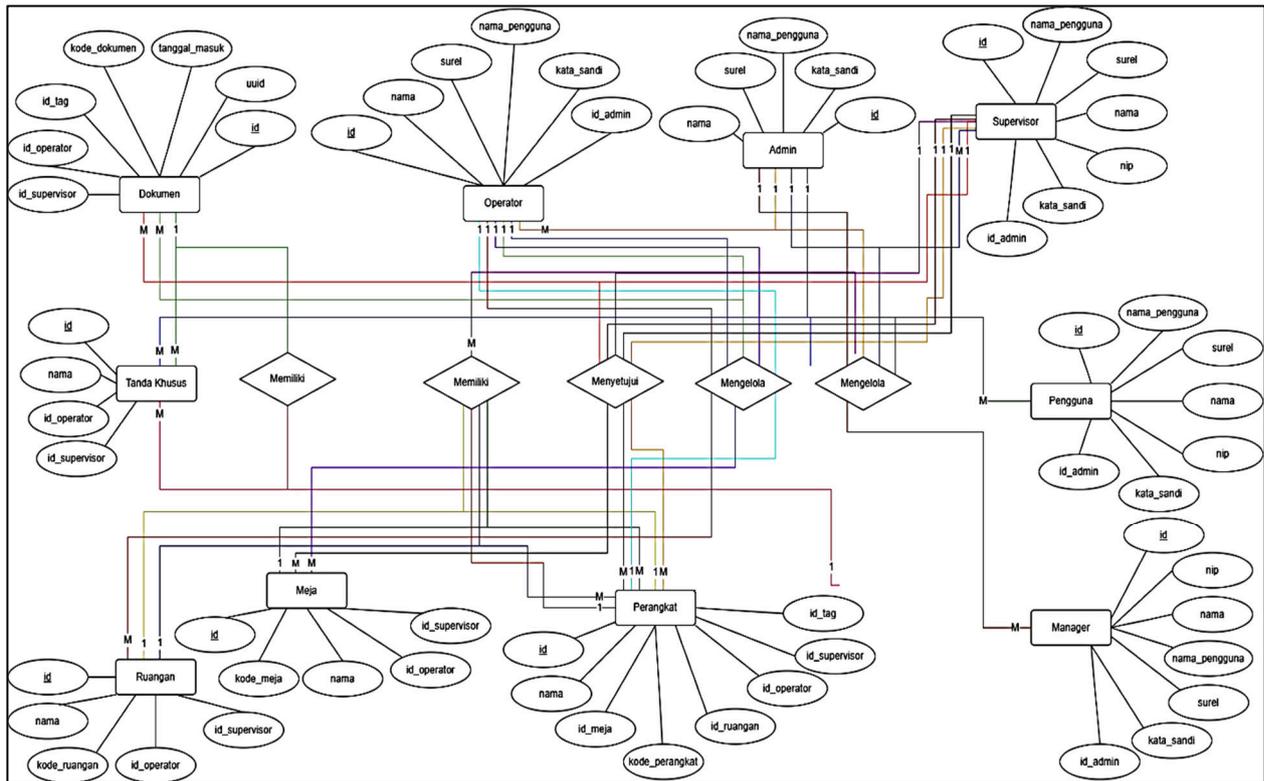
(Gambar 1) memperlihatkan diagram *use case* dari sistem yang meliputi 5 peran, yakni pengguna, *admin*,

operator, manajer, dan supervisor. Setiap peran melakukan *login* ke sistem dan bersamaan dengan itu sesi dimulai. Setelah *login*, setiap peran melakukan fungsi masing-masing yang dijabarkan sebagai berikut.

Pengguna memiliki peran untuk melakukan pencarian dokumen. *Admin* memiliki peran untuk mengelola data pengguna, meliputi melihat, mengubah, dan menghapus. Operator memiliki peran untuk mengelola data dokumen, ruangan, *tag*, dan meja. Peran tersebut meliputi menambahkan, melihat, mengubah, dan menghapus.

Supervisor memiliki peran untuk memberi izin penghapusan data yang dilakukan oleh operator. Manajer memiliki peran hanya dapat melihat semua data, yaitu dokumen, ruangan, *tag*, dan meja.

Perancangan berikutnya adalah skema basis data yang memuat rincian kebutuhan sistem. Skema ini disebut dengan ERD (*Entity Relational Diagram*) model. ERD model ini akan menjelaskan hubungan setiap entitas dalam suatu sistem yang akan ditunjukkan pada gambar berikut.



Gambar 2. Entity Relational Diagram sistem

Sistem ini memerlukan 10 entitas, yaitu Dokumen, Perangkat, Meja, Ruangan, Administrator, Operator, Supervisor, Manajer dan Tanda Khusus, dan Pengguna. Setiap entitas memiliki atribut dan memiliki satu kunci primer sebagai tanda unik, sehingga dapat mencegah adanya data yang sama. Setiap entitas memiliki hubungan satu sama lain dengan hubungan satu ke satu, satu ke banyak, dan banyak ke banyak. Penjelasan setiap atribut pada entitas ada pada tabel berikut.

Entitas dokumen berisi beberapa atribut, meliputi *id*, *id_operator*, *id_tag*, *tanggal_masuk*, *uuid*, *kode_dokumen*, dan *nama_dokumen*. Atribut *id* sebagai kunci primer dari entitas ini. Selain itu, entitas ini juga memiliki kunci asing yakni *id_operator* yang terhubung dengan entitas operator dengan hubungan *many to one* dan *id_tag* yang terhubung dengan entitas tanda khusus dengan hubungan *one to many*.

Entitas tanda khusus memiliki 4 atribut, yang meliputi *id* sebagai kunci primer, *id_operator* sebagai kunci asing yang terhubung dengan entitas operator dengan hubungan *many to one*, dan *nama* sebagai identitas tanda khusus.

Entitas ruangan memiliki 5 atribut. Atribut tersebut meliputi *id* sebagai kunci primer, *id_operator* dan *id_supervisor* sebagai kunci asing, *nama* sebagai identitas entitas, serta *kode_ruangan*. Atribut *id_operator* terhubung dengan entitas operator dengan hubungan *many to one*. Sedangkan *id_supervisor* terhubung dengan entitas supervisor dengan hubungan *many to one*.

Entitas meja memiliki 6 atribut. Atribut tersebut berisi *id* sebagai kunci primer, *id_operator*, *id_ruangan*, *id_supervisor* sebagai kunci asing, *nama* sebagai identitas, serta *kode_meja*. Atribut *id_operator* terhubung dengan entitas operator. Atribut *id_ruangan* terhubung dengan entitas ruangan. Atribut *id_supervisor* terhubung dengan entitas supervisor. Ketiga hubungan entitas ini dengan entitas lain adalah *many to one*.

Entitas perangkat memiliki 8 atribut. Atribut tersebut berisi *id* sebagai kunci primer, *id_operator*, *id_ruangan*, *id_meja*, *id_tag*, *id_supervisor* sebagai kunci asing yang, *nama* sebagai identitas, dan *kode_perangkat*. Atribut *id_operator* terhubung dengan entitas operator. Atribut *id_ruangan* terhubung dengan entitas ruangan. Atribut

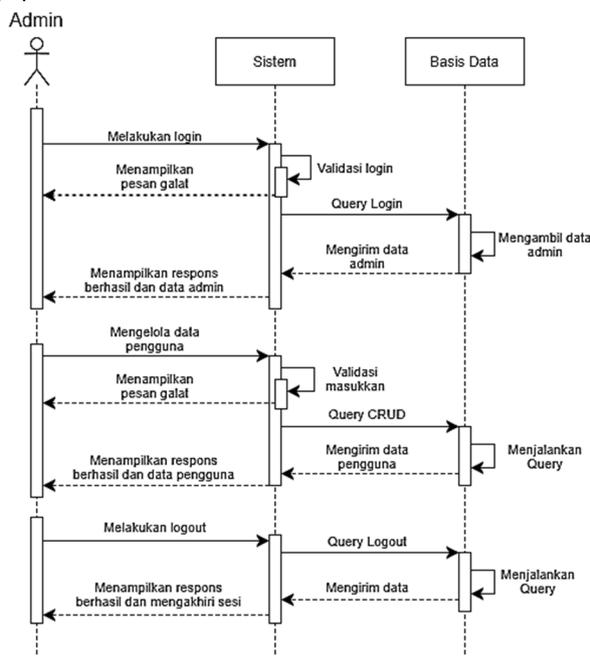


id_tag terhubung dengan entitas *tag*. Atribut *id_supervisor* terhubung dengan entitas supervisor. Atribut *id_supervisor* terhubung dengan entitas supervisor. Hubungan dengan entitas ruangan dan supervisor adalah *many to one*. Sedangkan hubungan dengan entitas tanda khusus adalah *many to many*.

Entitas pengguna, supervisor, dan manajer memiliki 6 atribut. Atribut tersebut berisi *id* sebagai kunci primer, nama sebagai identitas, surel dan *nama_pengguna* dengan keterangan unik, *kata_sandi*, serta *id_admin* sebagai kunci asing yang terhubung dengan entitas *admin* secara *many to one*.

Entitas *admin* memiliki 5 atribut. Entitas ini hampir sama dengan entitas pengguna, supervisor, dan manajer, hanya saja entitas ini tidak memiliki *id_admin*. Adapun atribut dari entitas *admin* meliputi *id* sebagai kunci primer, nama sebagai identitas, *nama_pengguna* dan surel dengan keterangan unik, serta *kata_sandi*.

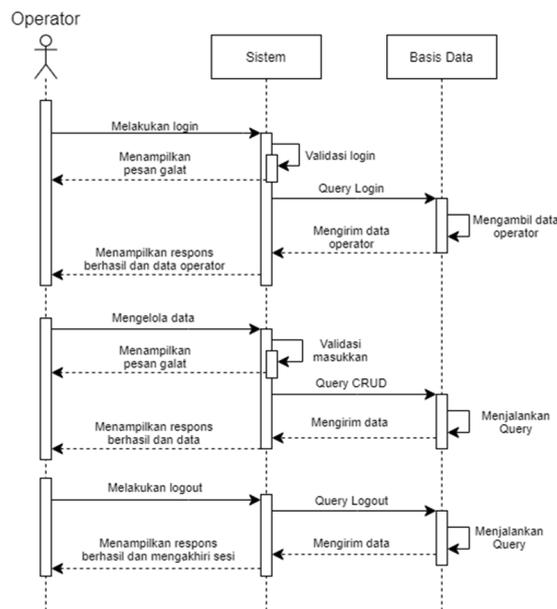
Sistem ini juga dirancang menggunakan sequence diagram. *Sequence* diagram merupakan diagram yang menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, tampilan, dan sebagainya) berupa pesan yang digambarkan terhadap waktu [6]. *Sequence* diagram adalah diagram yang mendeskripsikan interaksi antara objek-objek tersebut [7].



Gambar 3. Sequence diagram untuk peran admin

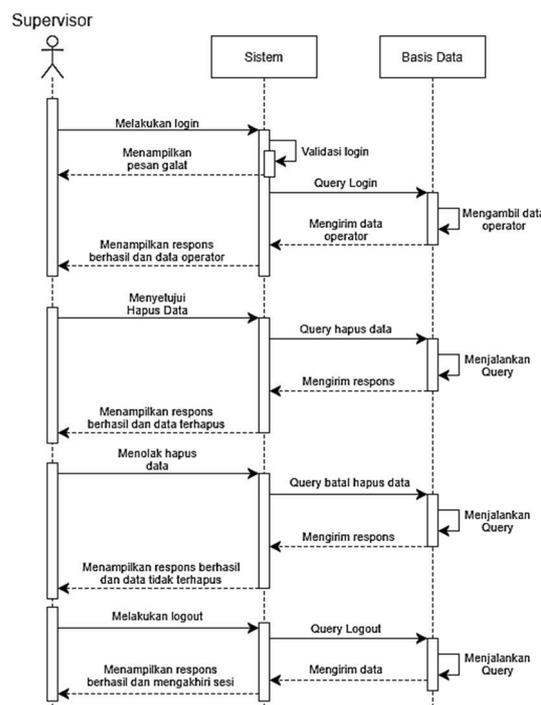
Berdasarkan pada Gambar 3 diperlihatkan diagram *sequence* dari peran *admin* yang dihubungkan dengan sistem dan basis data. Proses yang dilakukan *admin* diawali dengan melakukan *login* yang divalidasi oleh sistem. Sistem akan menampilkan pesan galat ketika *login* gagal. Jika masukkan lolos validasi maka sistem akan mengirimkan *query* ke basis data lalu mengirimkan respons dari basis data ke *admin*. Ketika *admin* sudah memulai sesi pada aplikasi, maka *admin* dapat mengelola data pengguna. Setelah itu jika sudah selesai, *admin*

dapat melakukan *logout* dari sistem untuk mengakhiri sesi.



Gambar 4. Sequence diagram untuk peran Operator

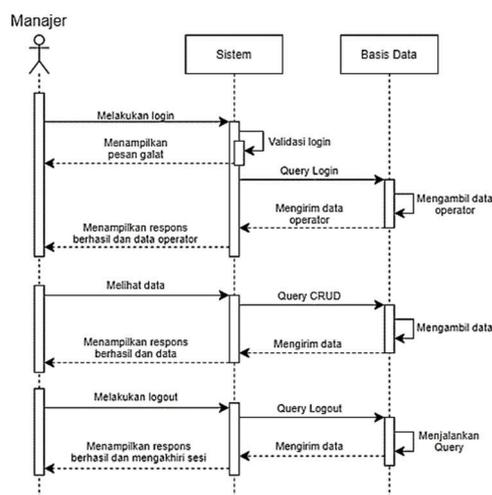
Berdasarkan pada Gambar 4 diperlihatkan diagram *sequence* dari peran operator yang dihubungkan dengan sistem dan basis data. Proses yang dilakukan operator diawali dengan melakukan *login* yang divalidasi oleh sistem. Sistem akan menampilkan pesan galat ketika *login* gagal. Jika masukkan lolos validasi maka sistem akan mengirimkan *query* ke basis data lalu mengirimkan respons dari basis data ke operator. Ketika operator sudah memulai sesi pada aplikasi, maka operator dapat mengelola data. Setelah itu jika sudah selesai, operator dapat melakukan *logout* dari sistem untuk mengakhiri sesi.



Gambar 5. Diagram Sequence untuk peran Supervisor

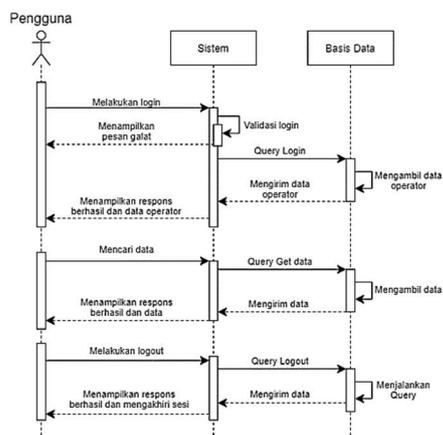


Berdasarkan pada Gambar 5 diagram *sequence* dari peran supervisor yang dihubungkan dengan sistem dan basis data. Proses yang dilakukan supervisor diawali dengan melakukan *login* yang divalidasi oleh sistem. Sistem akan menampilkan pesan galat ketika *login* gagal. Jika masukkan lolos validasi maka sistem akan mengirimkan *query* ke basis data lalu mengirimkan respons dari basis data ke supervisor. Ketika supervisor sudah memulai sesi pada aplikasi, maka supervisor dapat menyetujui atau menolak hapus data. Jika masukkan lolos validasi maka sistem akan mengirimkan *query* ke basis data lalu mengirimkan respons dari basis data ke supervisor bahwa data berhasil dihapus atau tidak terhapus. Setelah itu jika sudah selesai, operator dapat melakukan *logout* dari sistem untuk mengakhiri sesi.



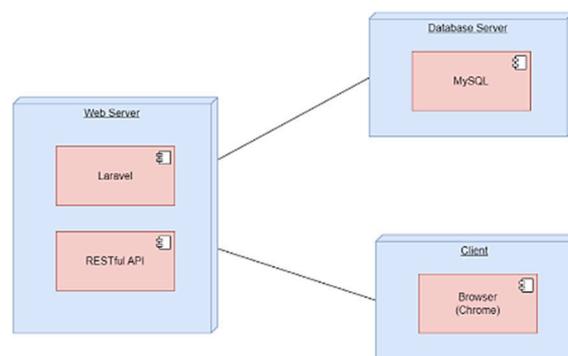
Gambar 6. Diagram *Sequence* untuk peran Manajer

Berdasarkan pada Gambar 6 diagram *sequence* dari peran manajer yang dihubungkan dengan sistem dan basis data. Proses yang dilakukan manajer diawali dengan melakukan *login* yang divalidasi oleh sistem. Sistem akan menampilkan pesan galat ketika *login* gagal. Jika masukkan lolos validasi maka sistem akan mengirimkan *query* ke basis data lalu mengirimkan respons dari basis data ke manajer. Ketika manajer sudah memulai sesi pada aplikasi, maka manajer dapat melihat data. Setelah itu jika sudah selesai, operator dapat melakukan *logout* dari sistem untuk mengakhiri sesi.



Gambar 7. Diagram *Sequence* untuk peran Pengguna

Berdasarkan pada Gambar 7 diagram *sequence* dari peran pengguna yang dihubungkan dengan sistem dan basis data. Proses yang dilakukan pengguna diawali dengan melakukan *login* yang divalidasi oleh sistem. Sistem akan menampilkan pesan galat ketika *login* gagal. Jika masukkan lolos validasi maka sistem akan mengirimkan *query* ke basis data lalu mengirimkan respons dari basis data ke pengguna. Ketika pengguna sudah memulai sesi pada aplikasi, maka pengguna dapat mencari data. Setelah itu jika sudah selesai, operator dapat melakukan *logout* dari sistem untuk mengakhiri sesi. Sistem ini juga dirancang menggunakan *deployment diagram*. *Deployment Diagram* adalah diagram yang menunjukkan konfigurasi komponen di dalam proses eksekusi aplikasi [8]. *Deployment Diagram* digunakan untuk menggambarkan detail bagaimana komponen disusun di infrastruktur [9]. Berikut deployment diagram dari sistem ini.



Gambar 8. *Deployment Diagram* sistem

Berdasarkan pada Gambar 8 infrastruktur dari sistem ini dibagi menjadi tiga, meliputi web server yang berisi Laravel dan RESTful API, database server yang berisi MySQL, dan client yang berisi Browser. Client dapat mengakses web server untuk menggunakan RESTful API. Data yang dibutuhkan oleh Client disimpan pada MySQL. MySQL diakses oleh Laravel untuk mengelola data yang kemudian ditampilkan pada Client.

C. Implementasi

Implementasi dari sistem ini menggunakan kerangka kerja Laravel untuk menyusun program. Laravel merupakan sebuah kerangka pengembangan web berbasis *Model View Controller* (MVC) yang didesain untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan dan perbaikan serta meningkatkan produktivitas pekerjaan dengan sintak yang bersih dan fungsional yang dapat mengurangi banyak waktu untuk implementasi [10].

Implementasi pada basis data dari sistem ini menggunakan MySQL. MySQL merupakan sebuah *Database Management System* (DBMS) populer yang memiliki fungsi sebagai *Relational Database Manajemen System* (RDBMS). Selain itu MySQL software merupakan suatu aplikasi yang sifatnya sumber terbuka serta kinerja server MySQL sangat cepat, *reliable*, dan mudah untuk digunakan serta bekerja



dengan arsitektur *client server* dan *embedded systems* [10].

D. Pengujian

Sistem ini diuji dengan menggunakan pengujian *black box*. Pengujian *Black Box* adalah pengujian yang memverifikasi hasil eksekusi aplikasi berdasarkan masukan yang diberikan (data uji) untuk memastikan fungsional dari aplikasi sudah sesuai dengan persyaratan [11].

Pengujian tersebut meliputi, membuat kasus uji untuk pengujian setiap fungsi, membuat kasus uji untuk pengujian kesesuaian alur dan hasil, mencari galat berdasarkan pesan yang ditampilkan sistem [12].

E. Evaluasi

Evaluasi dilakukan ketika pengujian mengalami kegagalan atau menampilkan galat. Selain itu, evaluasi juga dapat dilakukan untuk pengembangan sistem berikutnya, dengan begitu sistem menjadi semakin baik kinerjanya dan sesuai dengan kebutuhan pengguna.

III. HASIL DAN PEMBAHASAN

Berdasarkan implementasi dari perancangan yang ada, hasil yang didapat adalah semua fungsi sesuai dengan hasil yang diharapkan. Selain itu, pada fungsi *login*, tambah data, dan edit data berhasil dilakukan validasi. Hasil dari pengujian dibagi atas 5 peran, yakni *Admin*, *Pengguna*, *Operator*, *Supervisor*, dan *Manajer*. Masing-masing peran memiliki otoritas dalam mengelola data pada sistem ini.

A. Pengujian Peran Admin

Pengujian pada peran *admin* menguji semua fungsi yang masuk ke dalam otoritas *admin* meliputi *login*, *logout*, dan mengelola data pengguna. Berikut tabel rangkuman pengujian pada peran *admin*.

Tabel 1. Pengujian peran Admin

Nama Pengujian	Bentuk Pengujian	Hasil yang diharapkan	Hasil Pengujian
<i>Login</i>	Memasukkan <i>username</i> dan kata sandi	<i>Login</i> berhasil dan mendapatkan <i>token</i>	Berhasil
Validasi <i>username</i>	Memasukkan <i>username</i> yang salah	<i>Login</i> gagal dan diberikan pesan galat	Berhasil
Validasi kata sandi	Memasukkan kata sandi yang salah	<i>Login</i> gagal dan diberikan pesan galat	Berhasil
<i>Logout</i>	Melakukan <i>logout</i> dari sistem	<i>Pengguna</i> melakukan proses <i>logout</i> dari sistem dan mengakhiri sesi.	Berhasil
Tambah Data	Melakukan tambah data	Data berhasil ditambahkan	Berhasil
Validasi Tambah Data	Melakukan validasi tambah data	Muncul pesan galat jika data masukkan tidak sesuai atau kosong	Berhasil

Lihat Data	Melihat data keseluruhan	Data keseluruhan data dilihat	Berhasil
Lihat Detail Data	Melihat data secara detail satu persatu	Detail data berhasil ditampilkan satu persatu	Berhasil
Edit Data	Melakukan edit data	Data berhasil diubah	Berhasil
Validasi Edit Data	Melakukan validasi edit data	Muncul pesan galat jika data masukkan tidak sesuai atau kosong	Berhasil
Hapus Data	Menghapus data yang tidak digunakan	Data terhapus dengan status menunggu persetujuan	Berhasil

Berdasarkan pada Tabel 1 pengujian yang dilakukan berhasil pada setiap fungsi. Peran *admin* bertugas untuk mengelola data pengguna yang meliputi tambah, lihat, edit, dan hapus. Fungsi tambah dan edit divalidasi oleh sistem karena membutuhkan masukkan dari *admin*.

B. Pengujian Peran Operator

Pengujian pada peran operator menguji semua fungsi yang masuk ke dalam otoritas operator meliputi *login*, *logout*, mengelola data *tag*, mengelola data ruangan, mengelola data meja, mengelola data alat, dan mengelola data dokumen. Berikut tabel rangkuman pengujian pada peran operator.

Tabel 2. Pengujian peran Operator

Nama Pengujian	Bentuk Pengujian	Hasil yang diharapkan	Hasil Pengujian
<i>Login</i>	Memasukkan <i>username</i> dan kata sandi	<i>Login</i> berhasil dan mendapatkan <i>token</i>	Berhasil
Validasi <i>username</i>	Memasukkan <i>username</i> yang salah	<i>Login</i> gagal dan diberikan pesan galat	Berhasil
Validasi kata sandi	Memasukkan kata sandi yang salah	<i>Login</i> gagal dan diberikan pesan galat	Berhasil
<i>Logout</i>	Melakukan <i>logout</i> dari sistem	<i>Pengguna</i> melakukan proses <i>logout</i> dari sistem dan mengakhiri sesi.	Berhasil
Tambah Data	Melakukan tambah data	Data berhasil ditambahkan	Berhasil
Validasi Tambah Data	Melakukan validasi tambah data	Muncul pesan galat jika data masukkan tidak sesuai atau kosong	Berhasil
Lihat Data	Melihat data keseluruhan	Data keseluruhan data dilihat	Berhasil
Lihat Detail Data	Melihat data secara detail satu persatu	Detail data berhasil ditampilkan satu persatu	Berhasil
Edit Data	Melakukan edit data	Data berhasil diubah	Berhasil
Validasi Edit Data	Melakukan validasi edit data	Muncul pesan galat jika data masukkan tidak sesuai atau kosong	Berhasil
Hapus Data	Menghapus data yang tidak digunakan	Data terhapus dengan status menunggu persetujuan	Berhasil



Berdasarkan pada Tabel 2 hasil yang didapat juga sesuai dengan hasil yang diharapkan dengan tambahan validasi untuk mengurangi galat masukkan dari operator saat *login*, tambah data, dan edit data.

C. Pengujian Peran Supervisor

Pengujian pada peran supervisor menguji semua fungsi yang masuk ke dalam otoritas supervisor meliputi *login*, *logout*, melihat data persetujuan, menyetujui penghapusan data, dan menolak penghapusan data. Berikut tabel rangkuman pengujian pada peran operator.

Tabel 3. Pengujian peran Supervisor

Nama Pengujian	Bentuk Pengujian	Hasil yang diharapkan	Hasil Pengujian
<i>Login</i>	Memasukkan <i>username</i> dan kata sandi	<i>Login</i> berhasil dan mendapatkan <i>token</i>	Berhasil
Validasi <i>username</i>	Memasukkan <i>username</i> yang salah	<i>Login</i> gagal dan diberikan pesan galat	Berhasil
Validasi kata sandi	Memasukkan kata sandi yang salah	<i>Login</i> gagal dan diberikan pesan galat	Berhasil
<i>Logout</i>	Melakukan <i>logout</i> dari sistem	<i>Pengguna</i> melakukan proses <i>logout</i> dari sistem dan mengakhiri sesi.	Berhasil
Lihat Data	Melihat data keseluruhan	Data keseluruhan data dilihat	Berhasil
Setujui Hapus Data	Menyetujui penghapusan data oleh operator	Data berhasil terhapus dan status berubah menjadi disetujui pada daftar persetujuan	Berhasil

Berdasarkan pada Tabel 3 pengujian yang dilakukan berhasil pada semua fungsi. Validasi pada pengujian *login* juga berhasil berjalan. Supervisor akan diberi pesan galat jika salah dalam memasukkan *username* dan *password*.

D. Pengujian Peran Manajer

Pengujian pada peran manajer, akan menguji semua fungsi yang masuk ke dalam otoritas manajer meliputi *login*, *logout*, melihat data *tag*, melihat data ruangan, melihat data meja, melihat data alat, melihat data dokumen, dan melihat data persetujuan. Berikut tabel rangkuman pengujian pada peran manajer.

Tabel 4. Pengujian peran manajer

Nama Pengujian	Bentuk Pengujian	Hasil yang diharapkan	Hasil Pengujian
<i>Login</i>	Memasukkan <i>username</i> dan kata sandi	<i>Login</i> berhasil dan mendapatkan <i>token</i>	Berhasil
Validasi <i>username</i>	Memasukkan <i>username</i> yang salah	<i>Login</i> gagal dan diberikan pesan galat	Berhasil
Validasi kata sandi	Memasukkan kata sandi yang salah	<i>Login</i> gagal dan diberikan pesan galat	Berhasil
<i>Logout</i>	Melakukan <i>logout</i> dari sistem	<i>Pengguna</i> melakukan proses <i>logout</i> dari sistem dan mengakhiri sesi.	Berhasil

Lihat Data	Melihat data keseluruhan	Data keseluruhan data dilihat	Berhasil
Lihat Detail Data	Melihat data secara detail satu persatu	Detail data berhasil ditampilkan satu persatu	Berhasil

Berdasarkan pada Tabel 4 pengujian berhasil pada semua fungsi. Validasi fungsi hanya ada ketika manajer melakukan *login*. Validasi akan memunculkan pesan galat ketika manajer salah dalam memasukkan data atau memasukkan data kosong.

E. Pengujian Peran Pengguna

Pengujian pada peran pengguna, akan menguji semua fungsi yang masuk ke dalam otoritas pengguna meliputi *login*, *logout*, melakukan pencarian sekaligus melihat data dokumen. Berikut tabel rangkuman pengujian pada peran pengguna.

Tabel 5. Pengujian peran pengguna

Nama Pengujian	Bentuk Pengujian	Hasil yang diharapkan	Hasil Pengujian
<i>Login</i>	Memasukkan <i>username</i> dan kata sandi	<i>Login</i> berhasil dan mendapatkan <i>token</i>	Berhasil
Validasi <i>username</i>	Memasukkan <i>username</i> yang salah	<i>Login</i> gagal dan diberikan pesan galat	Berhasil
Validasi kata sandi	Memasukkan kata sandi yang salah	<i>Login</i> gagal dan diberikan pesan galat	Berhasil
<i>Logout</i>	Melakukan <i>logout</i> dari sistem	<i>Pengguna</i> melakukan proses <i>logout</i> dari sistem dan mengakhiri sesi.	Berhasil
Lihat Data	Melihat data keseluruhan	Data keseluruhan data dilihat	Berhasil
Lihat Detail Data	Melihat data secara detail satu persatu	Detail data berhasil ditampilkan satu persatu	Berhasil

Berdasarkan pada Tabel 5 pengujian yang dilakukan oleh pengguna berhasil. Validasi dari sistem juga berjalan pada fungsi *login*. Pengguna diberikan pesan galat jika terjadi kesalahan pada saat memasukkan data.

Berdasarkan hasil pengujian yang didapat, dikatakan bahwa masukkan dari setiap peran berbeda-beda. Hal tersebut disesuaikan berdasarkan otoritas dari masing-masing peran. Ini merupakan sebuah proteksi keamanan data dalam bentuk Otorisasi. Sebelum melakukan pengelolaan ataupun melihat data, setiap peran diwajibkan untuk *login*. Hal ini termasuk ke dalam proteksi keamanan berupa Autentikasi. Selain kedua proteksi tersebut, setiap masukkan yang dilakukan pengguna dengan peran masing-masing, terdapat validasi sistem yang memungkinkan galat masukkan dari pengguna dapat diminimalkan. Pengguna akan diberikan petunjuk berupa pesan galat dari sistem yang dapat dijadikan informasi untuk melakukan aksi pembenahan masukkan.

Adanya beberapa proteksi keamanan dari sistem, maka sistem akan menjadi lebih efektif dalam penggunaannya. Hal ini memungkinkan pengguna dapat



mengelola data dengan aman. Meskipun terjadi kesalahan, hal itu tidak akan merusak kinerja sistem. Karena data tidak akan disimpan jika terjadi kesalahan pada saat memasukkan data ke sistem. Hasil dari pengujian yang dilakukan menjelaskan bahwa setiap fungsi yang ada pada sistem sudah berjalan sesuai dengan perencanaan yang dilakukan sebelumnya. Hasil ini merupakan pengembangan dari penelitian sebelumnya yang hanya meneliti terkait dengan pengelolaan data, seperti tambah, lihat, edit, dan hapus. Penelitian ini memiliki dua fitur pengembangan, yakni persetujuan dan validasi. Kedua fitur itu digunakan untuk meminimalkan terjadinya kesalahan oleh pengguna.

IV. KESIMPULAN

Sistem yang dikembangkan dalam penelitian ini dapat mengelola data, seperti tambah, lihat, edit, dan hapus. Selain itu sistem ini dilengkapi dengan fitur validasi dan persetujuan untuk mengurangi kesalahan oleh pengguna serta menambah keamanan dalam mengelola data. Adapun hal yang masih perlu dikembangkan lagi adalah penambahan histori aktivitas yang dilakukan setiap peran agar pertukaran data dapat diketahui.

DAFTAR PUSTAKA

- [1] A. Ekawijaya dan B. Wisnuadhi, "Implementasi Aplikasi Kepengaturan Dokumen Akreditasi Program Studi dengan Metode Borg and Gall," *Jurnal Riset Komputer*, vol. IX, no. 2, pp. 289-295, 2022.
- [2] O. D. Arianto dan Y. A. Susetyo, "Penerapan RESTful Web Service Dengan Menggunakan Framework Laravel untuk Pembangunan Sistem Informasi Manajemen Sumber Daya Manusia," *Jurnal Ilmiah Penelitian dan Pembelajaran Informatika*, vol. VII, no. 2, pp. 522-532, 2022.
- [3] E. R. Amirussalam, W. H. N. Putra dan W. Purnomo, "Pengembangan Aplikasi e-Commerce Penjualan Tanaman Anggrek isitaman.com menggunakan Restful API," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. VI, no. 10, pp. 4731-4740, 2022.
- [4] R. H. Kusumodestoni, B. B. Wahono, G. Sudiryanto dan F. Shobah, "Penerapan Metode Waterfall Pada Aplikasi Pengenalan Huruf Hijaiyah Berbasis Android Pada Paud Nabata," *Jurnal Informatika, Manajemen, dan Teknologi*, vol. XXIV, no. 1, pp. 1-8, 2022.
- [5] M. L. Hamzah, A. A. Purwanti, Sutoyo, A. Marsal, Sarbani dan Nazaruddin, "Implementation of Internet of Things on Smart Posters using Near Field Communication Technology in The Tourism Sector," *Computer Science and Informaton Teknologi*, vol. III, no. 3, pp. 194-202, 2022.
- [6] O. Saputra dan W. Safitri, "Sistem Informasi Administrasi Pembayaran Sumbangan Pembinaan Pendidikan (SPP) Berbasis Whatsapp Gateway," *Jurnal Sistem Informasi dan Teknologi*, vol. IV, no. 1, pp. 1-6, 2022.
- [7] M. R. Effendi dan J. Saputra, "Design and Build an Employee Leave Application System," *Journal Of Information Systems and Management*, vol. I, no. 4, pp. 42-53, 2022.
- [8] R. Permana, D. N. Sulistyowati, A. O. Sari dan T. A. Mutiara, "Rancang Bangun Sistem Informasi Penjualan dan Pembelian Alat Tulis Kantor pada CV. Putra Mandiri," *Jurnal Teknik Komputer AMIK BSI*, vol. VI, no. 1, pp. 141-148, 2020.
- [9] A. Hendini, "Pemodelan Uml Sistem Informasi Monitoring Penjualan Dan Stok Barang (Studi Kasus: Distro Zhezha Pontianak)," *Jurnal Khatulistiwa Informatika*, vol. IV, no. 2, pp. 107-116, 2016.
- [10] F. Muhammad, R. Andreswari dan S. F. S. Gumilang, "Perancangan Backend Website Dengan Kerangka Kerja Vue JS dan Laravel pada Startup Manawa Dengan Metode Waterfall," *e-Proceeding of Engineering*, vol. VII, no. 2, pp. 7122-7130, 2020.
- [11] V. Febrian, M. R. Ramadhan, M. Faisal dan A. Saifudin, "Pengujian pada Aplikasi Pneggajian Pegawai dengan Menggunakan Metode Blackbox," *Jurnal Informatika Universitas Pamulang*, vol. V, no. 1, pp. 61-66, 2020.
- [12] B. A. Priyaangga, D. B. Aji, M. Syahroni, N. T. S. Ajil dan A. Saifudin, "Pengujian Black Box pada Aplikasi Perpustakaan Menggunakan Teknik Equivalence Partitions," *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. III, no. 3, pp. 150-157, 2020.



©2023. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution-ShareAlike 4.0 International License.