

## RANCANG BANGUN APLIKASI INTERNAL KANTOR UNTUK DOKUMENTASI RAPAT BERBASIS WEB

**Rafika Akhsani\***

Akademi Komunitas Negeri Putra Sang Fajar Blitar

**Ahmad Taufiq Hidayat**

Universitas PGRI Adi Buana Surabaya

---

### Abstract

The organization is a unit that can achieve various goals or targets carried out through joint activities. In achieving organizational goals, indeed will not be separated from various problems. One way to solve the problem is to discuss the problem in a meeting. Meetings are used to find solutions to problems that are expected to reach consensus, resolution, and decision. This study will develop an application that is used for meeting management. The meetings that have been held will be grouped according to their main activities. So it can be seen how many times a meeting has been held and what the results will be. Office internal applications for documentation of web-based meeting results are built using the PHP programming language and using a MySQL database. The framework used is Laravel. System development method with a waterfall model. From the results of testing with the black box method, the office's internal application for the documentation of the results of the web-based meeting is 100% as planned.

### Keywords:

*Internal Office Application Meetings, Frameworks, Laravel, Meeting Information Systems.*

---

### Abstrak

Organisasi merupakan suatu kesatuan yang merupakan sarana untuk mencapai berbagai tujuan atau sasaran yang dilakukan melalui kegiatan bersama. Dalam mencapai tujuan organisasi tentunya tidak akan lepas dari berbagai permasalahan. Salah satu cara untuk memecahkan permasalahan adalah dengan mendiskusikan permasalahan tersebut dalam suatu rapat. Rapat digunakan untuk menemukan solusi dalam suatu permasalahan yang diharapkan dapat mencapai suatu mufakat, penyelesaian, dan keputusan. Pada penelitian ini akan mengembangkan sebuah aplikasi yang digunakan untuk pengelolaan rapat. Rapat-rapat yang telah dilakukan nantinya akan dikelompokkan sesuai dengan kegiatan utamanya. Jadi dapat diketahui sudah berapa kali mengadakan rapat dan hasilnya seperti apa. Aplikasi internal kantor untuk dokumentasi hasil rapat berbasis *web* dibangun menggunakan bahasa pemrograman *PHP* dan menggunakan *database MySQL. Framework* yang digunakan adalah *Laravel*. Metode pengembangan sistem dengan model *waterfall*. Dari hasil pengujian dengan metode *blackbox* telah diketahui bahwa aplikasi internal kantor untuk dokumentasi hasil rapat berbasis *web* 100% sudah sesuai dengan yang telah direncanakan.

### Kata Kunci:

*Rapat Aplikasi Internal Kantor, Framework, Laravel, Sistem Informasi Rapat.*

---

DOI : [10.38038/vocatech.v2i2.29](https://doi.org/10.38038/vocatech.v2i2.29)

Received: 30 March 2020 ; Accepted: 06 April 2021 ; Published: 20 April 2021

### \*Corresponding author:

Rafika Akhsani, Program Studi Penyuntingan Audio dan Video, Akademi Komunitas Negeri Putra Sang Fajar Blitar, Jalan dr. Sutomo no. 29, Kota Blitar, Jawa Timur, Indonesia.

Email: [achsany@akb.ac.id](mailto:achsany@akb.ac.id)

**Citation in APA Style:** Akhsani, R., & Hidayat, A.T. (2021). Rancang Bangun Aplikasi Internal Kantor untuk Dokumentasi Rapat Berbasis *Web*. *VOCATECH: Vocational Education and Technology Journal*, Vol. 2, 2 (2021), 80-87.

## I. PENDAHULUAN

Organisasi merupakan suatu kesatuan yang merupakan sarana untuk mencapai berbagai tujuan atau sasaran yang dilakukan melalui kegiatan bersama. Di suatu organisasi pastinya memiliki banyak komponen yang melandasi diantaranya terdapat banyak orang, tata hubungan kerja, spesialis pekerjaan dan kesadaran rasional dari anggota sesuai dengan kemampuan dan spesialisasi mereka masing-masing.

Dalam mencapai tujuan organisasi tentunya tidak akan lepas dari berbagai permasalahan. Salah satu cara untuk memecahkan permasalahan adalah dengan mendiskusikan permasalahan tersebut dalam suatu rapat. Rapat digunakan untuk menemukan solusi dalam suatu permasalahan yang diharapkan dapat mencapai suatu mufakat, penyelesaian, dan keputusan.

Hasil rapat tentunya dicatat oleh notulen rapat yang nantinya diakhir rapat akan dibacakan. Semakin hari semakin bertambah jadwal rapat yang direncanakan oleh organisasi. Tentunya ini akan menjadi pekerjaan tersendiri kaitannya dengan pengarsipan data-data hasil rapat. Pengarsipan yang masih manual tentunya akan menyita banyak waktu jika kita akan melakukan pencarian data-data rapat yang dibutuhkan. Cara ini masih terdapat kelemahan dalam kemudahan pencatatan, keamanan data hasil pencatatan, proses rekapitulasi frekuensi rapat untuk setiap kegiatan. Pegawai juga tidak mengetahui dengan pasti sudah berapa kali rapat untuk kegiatan tertentu telah dilaksanakan serta tidak mengetahui ada rapat apa yang telah atau akan berlangsung di organisasi.

Perkembangan teknologi yang semakin pesat tentunya dapat dimanfaatkan untuk mengoptimalkan sistem dalam sebuah organisasi sehingga kinerja organisasi menjadi lebih baik. Seperti pada penelitian rancang bangun aplikasi *QR Code* berbasis *android* pada perpustakaan Akademi Komunitas Negeri Putra Sang Fajar Blitar, dengan memanfaatkan teknologi, kegiatan pencatatan transaksi diperpustakaan yang dilakukan dengan memanfaatkan teknologi ternyata lebih cepat dari pada dilakukan secara manual (Fu'ad et al., 2019) (Safriadi et al., 2020)

Sistem informasi agenda rapat berbasis web juga pernah dilakukan (Rismayana & Nur, 2016) pada penelitian dilakukan pembuatan aplikasi yang dapat membantu proses pembuatan agenda rapat, serta mempermudah staf bagian kepegawaian untuk proses pembuatan informasi agenda rapat. Akan tetapi pada penelitian ini belum ada dokumentasi

atas kegiatan rapat-rapat yang telah dilaksanakan (Rismayana & Nur, 2016).

Selanjutnya, penelitian pengembangan aplikasi penjadwalan rapat menggunakan metode *phased development*. Tujuan dari penelitian yang dilakukan tersebut adalah memberikan alternatif solusi permasalahan bentroknya jadwal rapat dengan membangun aplikasi penjadwalan rapat menggunakan Bahasa pemrograman *PHP* dan basis data *MySQL* (Subhiyakto & Astuti, 2019).

Selanjutnya, penelitian yang berjudul *traceability* kebutuhan dan perancangan konseptual manajemen rapat terintegrasi. Permasalahan yang diangkat dalam penelitian ini adalah informasi tentang pelaksanaan rapat masih kurang sehingga mempengaruhi tingkat partisipasi kehadiran peserta, diskusi yang terjadi dalam rapat yang belum terfokus sesuai agenda rapat, dan notulensi rapat belum lengkap. Hasil dari penelitian adalah usulan perencanaan perbaikan manajemen rapat dengan mendeskripsikan kebutuhan antarmuka, fungsional, dan non-fungsional (Saputra et al., 2020) (Sastypratiwi et al., 2018).

Berdasarkan latar belakang yang telah diuraikan maka penulis akan mengembangkan sebuah aplikasi internal kantor untuk dokumentasi hasil rapat berbasis *web*. Aplikasi ini digunakan untuk pengelolaan hasil rapat. Rapat-rapat yang telah dilakukan nantinya akan dikelompokkan sesuai dengan kegiatan utamanya. Jadi ke depan dapat diketahui kegiatan tertentu sudah berapa kali mengadakan rapat dan hasilnya seperti apa. Aplikasi akan dibangun menggunakan bahasa pemrograman *PHP* dan menggunakan *database MySQL*.

## II. STUDI PUSTAKA

### A. Rapat

Menurut Kamus Besar Bahasa Indonesia (KBBI), rapat adalah sebuah pertemuan (kumpulan) untuk membicarakan sesuatu, sidang, majelis. Rapat juga dapat diartikan sebagai Salah satu media komunikasi kelompok yang bersifat tatap muka, Musyawarah kelompok untuk mufakat, atau Suatu komunikasi kelompok secara resmi.

Manajemen rapat merupakan suatu kegiatan merencanakan, mengorganisasikan, memimpin, dan mengontrol rapat. Tujuan melaksanakan rapat adalah membahas suatu permasalahan, mencari jalan keluarnya, dan mengambil keputusan – keputusan supaya tujuan individu atau organisasi dapat tercapai dengan baik (Rebori, 2014).

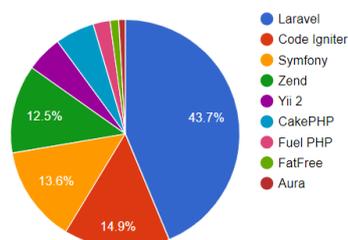
## B. Framework

*Framework* merupakan sebuah struktur konseptual dasar yang digunakan untuk memecahkan sebuah permasalahan. Sebuah *framework* berisi sekumpulan arsitektur atau konsep-konsep yang dapat mempermudah dalam pemecahan sebuah permasalahan. Tidak hanya permasalahan yang kecil, isu-isu permasalahan yang kompleks pun juga dapat diselesaikan. *Framework* menawarkan kemudahan bagi para penggunanya. Orang akan mengalami kenyamanan pada saat akan memulai dan mengembangkan proyeknya. Struktur yang biasa terdapat pada sebuah *framework* adalah (Ariesna, 2014)

- Standar *coding* adalah sebuah standar yang harus diikuti oleh *programmer* untuk menulis *code*.
- *Best practice* adalah kumpulan-kumpulan *action* yang telah teruji oleh para *expert*.
- *Design pattern* adalah teknik-teknik yang menjadi *best practice*.
- *Common function* adalah fungsi-fungsi atau *library* yang telah umum digunakan dalam pengembangan sebuah sistem.

## C. Laravel

*Laravel* merupakan sebuah kerangka kerja/*framework* berbasis *web* dengan *sintaks* yang ekspresif dan elegan. *Laravel* dikembangkan dengan menggunakan Bahasa pemrograman *PHP*. Dengan *Laravel* kita diharapkan dapat membuat aplikasi berbasis *web* dengan mudah. *Laravel* merupakan *framework* dengan pengguna paling banyak. Jumlah pengguna *framework Laravel* sampai dengan tahun 2019 adalah sebanyak 43% (Jasmin Ronald, n.d.). Kemudian baru disusul oleh *code igniter*, *symphony*, *Zend*, *Yii 2*, dan lain seterusnya. Peningkatan pengguna *Laravel* dapat dilihat pada Gambar 1.



Gambar 1. Top Pengguna *framework* di tahun 2019

*Laravel* merupakan *framework PHP* yang dirilis di bawah lisensi MIT. *Laravel* dibangun dengan konsep *Model View Controller (MVC)*. Dengan *Laravel*, kualitas perangkat lunak dapat

dioptimalkan karena dapat mengurangi biaya pengembangan awal dan biaya pemeliharaan. Fitur-fitur yang terdapat pada *Laravel* sebagai berikut (IdCloudHost, n.d.):

- *Bundles*, yaitu sebuah fitur *Laravel* yang berupa sistem pengemasan modular dan tersedia beragam di aplikasi.
- *Oloquent ORM*, yaitu fitur *Laravel* yang menyediakan metode internal dari pola “*active record*” yang mengatasi masalah pada hubungan objek basis data.
- *Application Logic*, yaitu bagian dari aplikasi, menggunakan *controller* atau bagian *Route*.
- *Reverse Routing*, yaitu fitur yang mendefinisikan relasi atau hubungan antara *Link* dan *Route*.
- *Restful controllers*, yaitu fitur yang menangani pemisahan logika dalam melayani *HTTP GET and POST*.
- *Class Auto Loading*, yaitu fitur yang menyediakan *loading* otomatis untuk *class PHP*.
- *View Composer*, yaitu kode unit logikal yang dapat dieksekusi ketika *view* sedang *loading*.
- *IoC Container*, yaitu fitur yang memungkinkan obyek baru dihasilkan dengan pembalikan *controller*.
- *Migration*, yaitu fitur menyediakan sistem kontrol untuk skema *database*.
- *Unit Testing*, yaitu fitur untuk mendeteksi dan mencegah regresi.
- *Automatic Pagination*, yaitu fitur untuk menyederhanakan tugas dari penerapan halaman.

## D. MySQL

*MySQL* adalah salah satu basis data yang cukup terkenal. *MySQL* karakteristik arsitektur yang *flexible*. Oleh karena itu, *MySQL* dapat berjalan dengan baik di desktop *application* maupun *web application*. *MySQL* dapat digunakan untuk *embedded application*, data *warehouses*, *content indexing and delivery software*, *highly redundant systems*, *online transaction processing (OLTP)*, dan masih banyak lagi. Dengan konfigurasi yang benar, *MySQL* dapat digunakan pada berbagai macam perangkat keras, dan juga *MySQL* mendukung banyak tipe data.

Salah satu fitur penting dari *MySQL* adalah *storage - engine architecture* yang *designnya* memisahkan *query processing* dan pekerjaan - pekerjaan *server* lainnya dari penyimpanan dan pengambilan data. Dengan adanya separasi ini

dapat membuat pengguna memilih bagaimana data disimpan, performa apa yang ingin dicapai, fitur - fitur apa aja yang diinginkan, dan karakteristik lainnya.

*MySQL* juga memiliki fitur *Partitioned Tables* sebagai *logical table* yang merupakan komposisi dari beberapa *sub - tables*. Tujuan dari *partitioning* ini adalah untuk membentuk *coarse form of indexing* dan *data clustering* dari tabel. Dengan begini sebagian besar dari tabel tidak perlu diakses dan untuk menyimpan *rows* yang berelasi dengan erat ([Schwartz, 2012](#)).

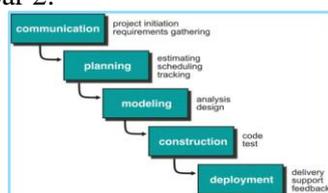
Beberapa keuntungan dari *MySQL* adalah sebagai berikut:

- *MySQL* dapat dijalankan di banyak *platform*
- Banyak *API* bahasa pemrograman yang terintegrasi dengan *MySQL*
- Performa dapat dioptimalkan
- *MySQL* merupakan *DBMS* yang *open-sources & freeware*
- *MySQL* irit *resources*.

### III. METODE

#### A. Metode

Penelitian ini menggunakan metode pengembangan sistem model *waterfall* dalam pelaksanaannya. Model *waterfall* melakukan pendekatan secara sistematis dan berurutan mulai dari tahap level kebutuhan sistem lalu menuju ke tahap analisis, desain, *coding*, *testing / verification*, dan *maintenance*. Model ini disebut juga dengan model air terjun karena setiap tahap demi tahap yang dilalui harus menunggu selesainya tahapan sebelumnya dan berjalan berurutan. Sebagai contoh, tahap desain harus menunggu selesainya tahap sebelumnya yaitu tahap *requirement* ([Roger & Pressman, 2012](#)). Model *waterfall* merupakan model pengembangan sistem yang sistematis dan sekuensial ([Sasmito, 2017](#)). Alur metode pengembangan *system model waterfall* dapat dilihat pada Gambar 2.



Gambar 2. Metode pengembangan *system model waterfall*

Penjelasan mengenai tahapan *model waterfall* sebagai berikut:

- *Communication*. Pada tahap ini dilakukan inisiasi proyek, seperti menganalisis masalah yang ada dan tujuan yang akan dicapai. Selain itu dilakukan juga *requirements gathering*, dimana akan dikumpulkan *requirement* dari *user*.
- *Planning*. Pada tahap ini dilakukan estimasi mengenai kebutuhan - kebutuhan yang diperlukan dalam membuat sistem.
- *Modelling*. Pada tahap ini dilakukan penerjemahan kebutuhan sistem ke dalam representasi untuk menilai kualitas sistem.
- *Construction*. Pada tahap ini dilakukan proses membuat kode. Kode dapat dibangun dengan menggunakan Bahasa pemrograman seperti *java*, *php*, dan lain sebagainya.
- *Deployment*. Tahapan final dalam pembuatan *software* atau sistem.

#### B. Kebutuhan fungsional

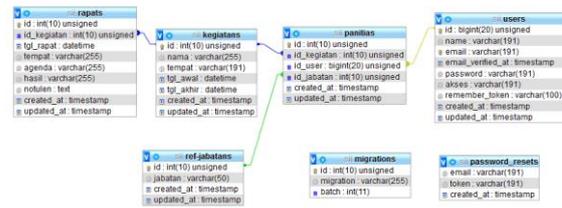
Kebutuhan fungsional merupakan fitur atau kebutuhan yang berkaitan dengan sistem secara langsung ([Saputra & Supriyono, 2020](#)). Kebutuhan fungsional dalam aplikasi internal kantor untuk dokumentasi hasil rapat berbasis *web* direncanakan berdasarkan tingkatan dari pengguna aplikasi. Pengguna aplikasi terdiri dari *admin* dan *user* biasa. Jadi, secara mendasar di dalam sistem ini terdapat 2 aktor yaitu *admin* dan *user* biasa. Pegawai dalam organisasi akan menjadi *user* biasa. Setiap pegawai memiliki hak akses sendiri berdasarkan jabatan dalam setiap kegiatan seperti ketua, wakil ketua, sekretaris, bendahara, dan anggota. Detail dari penjabaran kewenangan hak akses untuk masing-masing aktor dapat dilihat pada Tabel 1.

Tabel 1. Detail hak akses aktor

No	Jabatan	Hak akses
1	<i>Admin</i>	<ul style="list-style-type: none"> <li>• Dapat mengelola kegiatan</li> <li>• Dapat mengelola penjadwalan rapat</li> <li>• Dapat mengelola referensi jabatan</li> </ul>
1	Ketua ( <i>user</i> biasa)	<ul style="list-style-type: none"> <li>• Dapat menambahkan anggota panitia</li> <li>• Dapat melihat kepanitiaan</li> <li>• Dapat melihat daftar rapat dalam satu kegiatan dan hasil rapat</li> <li>• Dapat melihat daftar informasi kegiatan dan jadwal rapat</li> </ul>
2	Wakil ketua ( <i>user</i> biasa)	<ul style="list-style-type: none"> <li>• Dapat melihat kepanitiaan</li> <li>• Dapat melihat daftar rapat dalam satu kegiatan dan hasil rapat</li> </ul>

No	Jabatan	Hak akses
		<ul style="list-style-type: none"> <li>Dapat melihat daftar informasi kegiatan dan jadwal rapat</li> </ul>
3	Sekretaris (user biasa)	<ul style="list-style-type: none"> <li>Dapat melihat kepanitiaan</li> <li>Dapat menuliskan notulesi rapat</li> <li>Dapat melihat daftar rapat dalam satu kegiatan dan hasil rapat</li> <li>Dapat melihat daftar informasi kegiatan dan jadwal rapat</li> </ul>
4	Bendahara (user biasa)	<ul style="list-style-type: none"> <li>Dapat melihat kepanitiaan</li> <li>Dapat melihat daftar rapat dalam satu kegiatan dan hasil rapat</li> <li>Dapat melihat daftar informasi kegiatan dan jadwal rapat</li> </ul>
5	Anggota (user biasa)	<ul style="list-style-type: none"> <li>Dapat melihat kepanitiaan</li> <li>Dapat melihat daftar rapat dalam satu kegiatan dan hasil rapat</li> <li>Dapat melihat daftar informasi kegiatan dan jadwal rapat</li> </ul>

tabel ref-jabatans, tabel *password-resets*, dan tabel *migrations*. Reperesentasi dari relasi tabel – tabel yang telah disebutkan dapat dilihat pada Gambar 4.



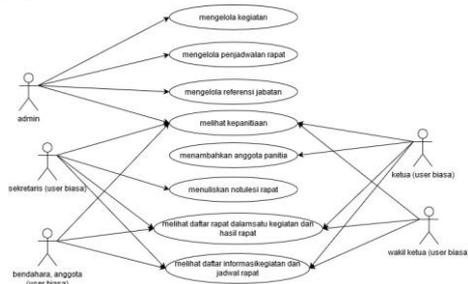
Gambar 4. Representasi relasi tabel

Semua tabel memiliki relasi antar tabel kecuali tabel *migrations* dan tabel *password-resets*. Hal itu dikarenakan tabel *migrations* dan tabel *password-resets* merupakan tabel bawaan dari *framework Laravel*. Jadi ketika melakukan instalasi *framework Laravel*, kedua tabel itu akan terbuat otomatis.

#### IV. HASIL DAN PEMBAHASAN

#### C. Desain system

Dalam mengembangkan sistem ini dilakukan dengan pendekatan *Use case diagram*. *Use case* menggambarkan fungsi dari suatu sistem. Fungsi dapat dilihat dari segi perspektif pengguna sehingga terminologinya dapat dimengerti (Whitten & Bentley, 2007). Adapun aktor yang terdapat pada *use case diagram* adalah ketua, wakil ketua, sekretaris, bendahara, dan anggota. *Use case* ini disesuaikan dengan kebutuhan fungsional dari sistem. *Use case diagram system* dapat dilihat pada Gambar 3.



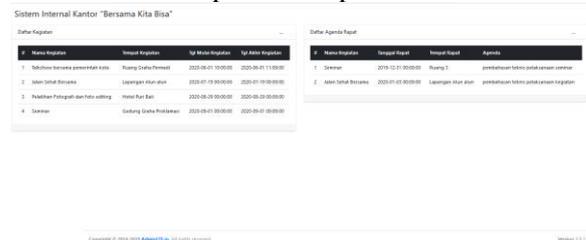
Gambar 3. Use case diagram

#### D. Basis Data

Basis data merupakan salah satu hal aspek yang sangat penting di dalam pengembangan aplikasi. Dengan memanfaatkan sistem manajemen *database*, pengguna dapat berinteraksi dalam pengelolaan data seperti mendefinisikan, membuat, memelihara, dan mengontrol akses suatu basis data (Connolly & Begg, 2010). Di dalam pengembangan aplikasi ini terdapat beberapa tabel untuk menyimpan data-data yang dibutuhkan seperti tabel *users*, tabel *panitias*, tabel *kegiatan*, tabel *rapats*,

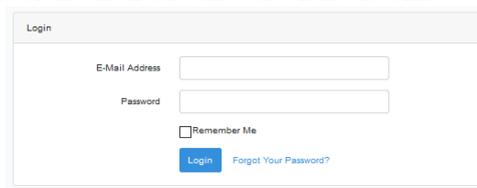
#### A. Hasil dan Pembahasan

Halaman utama dari sistem ini ketika diakses berupa tampilan list dari kumpulan kegiatan dan tampilan list daftar agenda rapat. Tampilan halaman utama dapat dilihat pada Gambar 5.



Gambar 5. Halaman Utama system

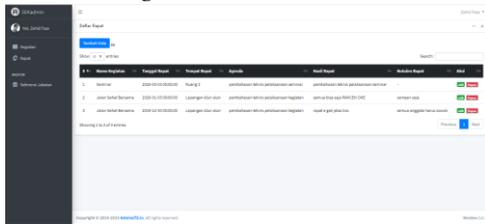
Halaman *login* pada Gambar 6. Halaman *login* digunakan oleh semua *user* atau pengguna baik *admin* atau *user* biasa. Untuk masuk ke dalam sistem, *user* atau pengguna harus memasukkan *username* (dalam bentuk *email*) dan *password* pada halaman *login* yang kemudian akan divalidasi oleh sistem.



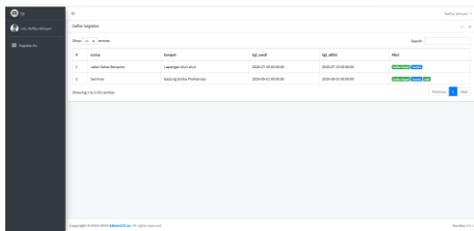
Gambar 6. Halaman login

Setelah *login* berhasil, maka *user* akan diarahkan ke halaman utama sesuai hak aksesnya.

Pada Gambar 7 dan 8 dapat dilihat halaman utama setelah sukses *login*.



Gambar 7. Halaman utama *admin*



Gambar 8. Halaman utama *user* biasa

Halaman kegiatan (Gambar 7) merupakan halaman untuk pengelolaan data kegiatan. Hanya *user admin* yang dapat melakukan akses ini. Di dalam halaman ini dapat dilakukan proses penambahan data kegiatan, pembaharuan data kegiatan, serta menghapus data kegiatan. *Form* untuk menambahkan data kegiatan dapat dilihat pada Gambar 9. *Form* tersebut berisikan data nama kegiatan, tempat kegiatan, serta tanggal pelaksanaan kegiatan (tanggal mulai kegiatan dan tanggal akhir kegiatan).

Gambar 9. *Form* tambah data kegiatan

Di level *admin* juga dapat mengelola terkait dengan data referensi jabatan seperti pada Gambar 10.

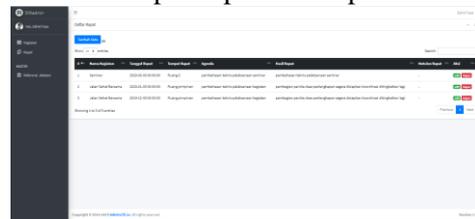
#	Jabatan	Aksi
1	Ketua	[edit] [hapus]
2	Wakil Ketua	[edit] [hapus]
3	Sekretaris	[edit] [hapus]
4	Bendahara	[edit] [hapus]
5	Anggota	[edit] [hapus]

Gambar 10. Referensi jabatan

Sedangkan *form* untuk melakukan pembaharuan data dapat dilihat pada Gambar 10.

Gambar 11. *Form* pembaharuan data kegiatan

Halaman rapat pada *user admin* digunakan untuk melakukan pengelolaan data rapat setiap kegiatan. Untuk melakukan pengelolaan data ini, *user admin* tinggal melakukan klik pada menu rapat. Halaman rapat dapat dilihat pada Gambar 11.



Gambar 12. Halaman rapat

Setelah melakukan rapat, masing-masing sekretaris kegiatan dapat melakukan pengisian data hasil rapat. *Form* untuk memasukkan data hasil rapat dapat dilihat pada Gambar 12.

Gambar 13. *Form* memasukkan data hasil rapat

## B. Pengujian

Setelah aplikasi yang telah dibangun dengan menggunakan *framework Laravel* dan basis data *MySQL* sudah selesai dikerjakan, maka dilakukan pengujian untuk mengetahui apakah aplikasi yang dibangun sudah sesuai yang direncanakan atau belum. Pengujian aplikasi dilakukan dengan metode *blackbox*. Pengujian ini dilakukan untuk mengetahui semua fitur yang telah dibuat apakah sudah sesuai dengan fungsinya dan berjalan dengan baik serta berjalan sesuai dengan rencana. Hasil pengujian dengan *blackbox* dapat dilihat pada tabel 2.

Tabel 2. Hasil pengujian dengan *blackbox*

Bagian pengujian	Fungsi yang diuji	Input	Output	Ket
Login admin	User admin melakukan login dan masuk ke halaman utama	Username dan password lalu klik login	Jika username dan password sesuai maka halaman utama akan muncul. Jika username dan password tidak sesuai maka muncul pemberitahuan proses login gagal dan kembali ke halaman login	sesuai
Data kegiatan	Melihat data kegiatan	Klik menu kegiatan	Menampilkan data kegiatan	sesuai
	Input data kegiatan	Input data nama kegiatan, tempat, tanggal awal kegiatan, dan tanggal akhir kegiatan	Menyimpan data ke dalam basis data dan menampilkannya	Sesuai
	Edit data kegiatan	nama kegiatan, tempat, tanggal awal kegiatan, dan tanggal akhir kegiatan	Menyimpan data ke dalam basis data dan menampilkannya	Sesuai
	Hapus data kegiatan	Klik menu hapus pada halaman kegiatan	Menghapus data di basis data dan menampilkan kembali data kegiatan	Sesuai
data rapat	Melihat data rapat	Klik menu rapat	Menampilkan data rapat	Sesuai
	Input data rapat	Input data tanggal rapat, tempat rapat, agenda rapat, hasil rapat, notulen rapat	Menyimpan data ke dalam basis data dan menampilkannya	Sesuai
	Edit data rapat	tanggal rapat, tempat	Menyimpan data ke dalam basis data dan	Sesuai

Bagian pengujian	Fungsi yang diuji	Input	Output	Ket
		rapat, agenda rapat, hasil rapat, notulen rapat	menampilkannya	
	Hapus data rapat	Klik menu hapus pada halaman rapat	Menghapus data di basis data dan menampilkan kembali data rapat	Sesuai
Referensi jabatan	melihat data referensi jabatan	Klik menu referensi jabatan	Menampilkan data referensi jabatan	Sesuai
	Input data referensi jabatan	Input data jabatan	Menyimpan data ke dalam basis data dan menampilkannya	Sesuai
	Edit data referensi jabatan	Data jabatan	Menyimpan data ke dalam basis data dan menampilkannya	Sesuai
	Hapus data referensi jabatan	Klik menu hapus pada halaman referensi jabatan	Menghapus data di basis data dan menampilkan kembali data kegiatan	Sesuai
Halaman utama	Menampilkan data kegiatan dan rapat	Akses halaman utama	Menampilkan list data kegiatan dan agenda rapat	Sesuai
Panitia	Menambahkan data panitia kegiatan	Input data ketua, wakil ketua, sekretaris, bendahara, dan anggota	Menyimpan data ke dalam basis data dan menampilkannya	Sesuai

## V. KESIMPULAN

### A. Kesimpulan

Berdasarkan hasil dan pembahasan yang telah diulas pada bab sebelumnya, maka dapat diambil kesimpulan sebagai berikut:

1. Aplikasi internal kantor untuk dokumentasi hasil rapat telah dapat dibangun dengan *framework Laravel* dan basis data *MySQL* dengan fitur yang telah direncanakan.
2. Telah dilaksanakan hasil pengujian dengan menggunakan metode pengujian *blackbox*. Dari hasil pengujian dapat diketahui bahwa semua fitur yang telah dibangun dapat berjalan

dengan baik dan sesuai dengan yang direncanakan pada tahap perencanaan pembuatan aplikasi. Hasil pengujian menyatakan bahwa 100% hasil pengujian fitur berjalan dengan baik dan sesuai dengan yang telah direncanakan

## B. Saran

Adapun Saran yang dapat saya suarakan adalah sebagai berikut:

1. Mengembangkan aplikasi ini dengan berbasis *mobile* karena untuk mengoperasikan aplikasi ini dengan basis *web* masih kurang maksimal.
2. Menambahkan fitur notifikasi jika terdapat data kegiatan atau agenda rapat yang baru.

## REFERENSI

- Ariesna, H. (2014). *Pembangunan Sistem Absensi dan Honor Guru SMA Negeri 2 Kotabumi Menggunakan Web Framework Codeigniter*. 15–16.
- Connolly, T., & Begg, C. (2010). *Database Systems A Practical Approach to Design, Implementation, and Management Fifth Edition (Fifth Edit)*. Pearson Education.
- Fu'ad, M. N., Kholil, M., & Wardhani, S. I. (2019). Rancang Bangun Aplikasi QR Code Berbasis Android pada Perpustakaan Akademi Komunitas Negeri Putra Sang Fajar Blitar. *VOCATECH: Vocational Education and Technology Journal*, 1. <https://doi.org/10.38038/vocatech.v1i0.2>
- IdCloudHost. (n.d.). *Pengertian dan Keunggulan Framework Laravel*. Retrieved April 9, 2020, from <https://idcloudhost.com/pengertian-dan-keunggulan-framework-laravel/>
- Jasmin Ronald. (n.d.). *Top 8 PHP Frameworks in 2019*. Retrieved April 9, 2020, from <https://medium.com/hackernoon/top-8-php-frameworks-in-2019-b6be163605c8>
- Rebori, M. K. (2014). *Fact Sheet 97-29 How to Organize and Run Effective Meetings*.
- Rismayana, A. H., & Nur, V. A. (2016). *Sistem Informasi Agenda Rapat Berbasis eb Menggunakan SMS Gateway ( Studi Kasus Politeknik TEDC Bandung )*. 10, 35–41.
- Roger, S. Pressman, P. . (2012). *Rekayasa Perangkat Lunak (Pendekatan Praktisi) Edisi 7 : Buku 1*. Andi.
- Safriadi, S., & Rahmadani, R. (2020). Klasifikasi Gender Berdasarkan Suara dengan Naive Bayes dan Mel Frequency Cepstral Coefficient. *VOCATECH: Vocational Education and Technology Journal*, 2(1), 19-26.
- Saputra, J., Rizaldi, R., Salahuddin, S., Mellyssa, W., & Usmardi, U. (2020). Sistem Pengaman Pintu Menggunakan Sidik Jari dan Android. *VOCATECH: Vocational Education and Technology Journal*, 2(1), 33-40.
- Saputra, N. E., & Supriyono, H. (2020). Rancang Bangun Sistem Penjadwalan pada SMA Muhammadiyah Al Kautsar Program Khusus Kartasura. *Jurnal Warta Lembaga Pengabdian pada Masyarakat*, 20(01), 1–6.
- Sasmito, G. W. (2017). Penerapan Metode Waterfall pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal. *Jurnal Informatika:Jurnal Pengembangan IT (JPIT)*, 2(1), 6–12.
- Sastypratiwi, H., Prihartini, N., Nyoto, R. D., & Anra, H. (2018). Traceability Kebutuhan dan Perancangan Konseptual Manajemen Rapat Terintegrasi. *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, 4(1), 43. <https://doi.org/10.26418/jp.v4i1.24499>
- Schwartz, B. (2012). *High Performance MySQL*. O'Reilly Media.
- Subhiyakto, E. R., & Astuti, Y. P. (2019). Pengembangan Aplikasi Penjadwalan Rapat Menggunakan Metode Phased Development. *Dinamika Rekayasa*, 15(1), 35. <https://doi.org/10.20884/1.dr.2019.15.1.248>
- Whitten, J. L., & Bentley, L. D. (2007). *System Analysis & Design Methods Seventh Edition*. McGraw-Hill.