

IMPLEMENTASI WEB SERVICE REST API UNTUK MERANCANG APLIKASI PUSAT INFORMASI MASJID AL MUHAJIRIN LARANGAN INDAH

Rizky Ramadhan^{1*}, Purwanto²

^{1,2}Fakultas Teknologi Informasi, Teknik Informatika, Universitas Budi Luhur, Jakarta, Indonesia

Email: ^{1*}rizkyramadhan978@gmail.com, ²purwanto@budiluhur.ac.id
(* : corresponding author)

Abstrak-Masjid Al-Muhajirin Larangan Indah masih belum dilengkapi dengan teknologi yang cukup. Informasi yang diberikan terkait jadwal kajian, laporan keuangan, serta kegiatan-kegiatan lainnya masih hanya sekedar diinformasikan melalui papan pengumuman atau spanduk, pengumuman disampaikan melalui pengeras suara, dan pencatatan laporan keuangan masih dicatat di buku tulis. Hal ini tentunya masih kurang efektif karena bisa jadi informasi yang diberikan kepada jamaah kurang merata dan bisa saja sewaktu-waktu buku catatan laporan keuangan hilang. Melalui *web service* dengan menggunakan metode REST API bisa menjadi salah satu cara untuk menangani permasalahan yang ada di Masjid Al-Muhajirin. Aplikasi dapat berjalan pada dua *platform*, yaitu *web* dan Android. Aplikasi *web* dan REST API dibangun menggunakan perangkat lunak Visual Studio Code, diantaranya aplikasi *web* dibuat dengan menggunakan *framework* Vue JS, sedangkan REST API dibuat dengan menggunakan *framework* CodeIgniter 4. Aplikasi Android dibangun menggunakan Android Studio dengan bahasa pemrograman Java. Aplikasi ini menyediakan informasi terkait jadwal kegiatan, laporan keuangan pengeluaran maupun pemasukan, informasi sewa aula, daftar anggota DKM, dan profil Masjid Al-Muhajirin. Metode pengujian dilakukan dengan menggunakan teknik pengujian *black box*. Dengan teknik pengujian tersebut, dapat diketahui apakah fitur yang disediakan sudah berjalan sesuai dengan apa yang diharapkan. Hasil pengujian yang sudah dilakukan pada aplikasi ini menunjukkan bahwa implementasi REST API dapat berjalan dengan baik di aplikasi *web* dan Android.

Kata Kunci: REST, API, *web service*

IMPLEMENTATION OF WEB SERVICE REST API TO DESIGN THE APPLICATION OF THE AL MUHAJIRIN MOSQUE INFORMATION CENTER LARANGAN INDAH

Abstract-Al-Muhajirin Larangan Indah Mosque is still not equipped with enough technology. The information provided regarding the schedule of studies, financial statements, and other activities is still only informed through notice boards or banners, announcements are made through loudspeakers, and financial statement recordings are still recorded in writing books. This is certainly still not effective because it could be that the information provided to pilgrims is not evenly distributed and it could be that at any time the financial statement notebook is lost. Through a web service using the REST API method, it can be one way to deal with problems in the Al-Muhajirin Mosque. The application can run on two platforms, namely the web and Android. Web applications and REST APIs are built using Visual Studio Code software, including web applications created using the Vue JS framework, while REST APIs are created using the CodeIgniter 4 framework. Android apps are built using Android Studio with the Java programming language. This application provides information related to the schedule of activities, financial reports on expenses and income, hall rental information, a list of DKM members, and a profile of the Al-Muhajirin Mosque. The test method is carried out using the black box testing technique. With this testing technique, it can be known whether the features provided have run according to what is expected. The results of tests that have been carried out on this application show that the REST API implementation can run well in web and Android applications.

Keywords: REST, API, *web service*

1. PENDAHULUAN

Masjid adalah tempat peribadatan bagi umat Islam. Masjid kerap kali difungsikan untuk proses belajar mengajar pendidikan Al-Qur'an atau mengaji dan juga biasanya digunakan sebagai sarana dakwah di dalam tatanan masyarakat. Dalam sejarah perkembangan umat Islam, masjid dinilai memiliki peranan begitu penting dalam menegakkan agama Islam.

Penelitian mengenai implementasi REST API telah dilakukan oleh [1]. Dari penelitian tersebut berhasil menyelesaikan permasalahan menggunakan metode RESTful API. Namun, sistem yang disediakan masih perlu ditingkatkan agar sistem tersebut lebih bisa dimaksimalkan dalam penggunaannya.

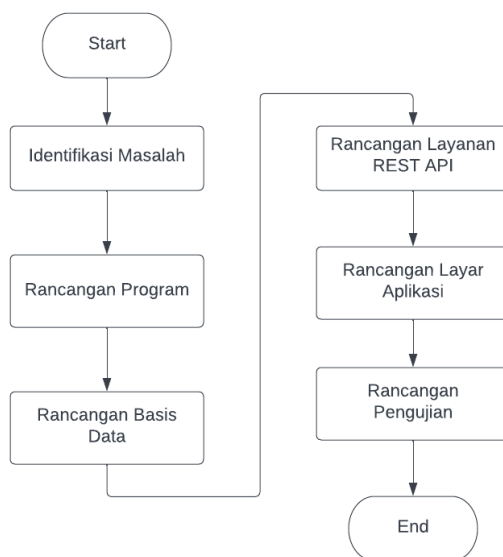
Saat ini banyak masjid yang masih belum menggunakan teknologi untuk keperluan ibadah, dakwah, dan sebagainya. Padahal banyak sekali manfaat yang bisa dirasakan ketika menggunakan bantuan teknologi, tidak terkecuali Masjid Al-Muhajirin Larangan Indah. Masjid Al-Muhajirin Larangan Indah berlokasi di Jl. Swadaya Raya, Larangan Indah, Tangerang. Masjid tersebut cukup besar dengan fasilitas yang bagus dan tingkat kepengurusan yang baik. Namun, Masjid Al-Muhajirin masih belum dilengkapi dengan teknologi yang cukup. Informasi yang diberikan terkait jadwal kajian, laporan keuangan mingguan atau bulanan, serta kegiatan-kegiatan lainnya masih hanya sekedar diinformasikan melalui papan pengumuman atau spanduk, pengumuman melalui pengeras suara, dan pencatatan laporan keuangan melalui buku tulis. Hal ini tentunya masih kurang efektif karena bisa jadi informasi yang diberikan kepada jamaah kurang merata dan bisa saja buku catatan laporan keuangan hilang. Untuk mengurangi kekurangan tersebut, maka diperlukannya sebuah media untuk menyebarkan informasi terkait kegiatan Masjid Al-Muhajirin dan juga laporan keuangan yang bertujuan agar informasi yang disampaikan dapat lebih merata dan laporan keuangan lebih terstruktur.

Melalui REST API bisa menjadi salah satu cara untuk menangani permasalahan yang ada di Masjid Al-Muhajirin. Dengan penggunaan metode tersebut, dapat dengan mudah untuk mengakses dan mengelola data melalui berbagai *platform*, seperti *web* dan *Android*. Dengan pemanfaatan REST API, penyebaran informasi kegiatan bisa lebih efektif, pencatatan laporan keuangan bisa lebih mudah dikelola, dan informasi seputar Masjid Al-Muhajirin lainnya dapat diakses dan disebar dengan lebih baik.

Tujuan dari penelitian ini adalah untuk memberikan sebuah sistem atau aplikasi yang dapat membantu pengurus masjid dalam menyebarkan informasi terkait Masjid Al-Muhajirin serta dapat membantu jamaah masjid dalam menerima atau melihat informasi seputar Masjid Al-Muhajirin dan juga dapat membantu pengurus masjid dalam mencatat/mengelola laporan keuangan Masjid Al-Muhajirin serta menyediakan akses kepada jamaah untuk melihat laporan keuangan Masjid Al-Muhajirin.

2. METODE PENELITIAN

Penelitian dilakukan di Masjid Al-Muhajirin Larangan Indah. Pengumpulan data dilakukan dengan melakukan observasi yaitu dengan mengikuti berbagai acara yang diselenggarakan oleh Masjid Al-Muhajirin dan ikut dalam organisasi keremajaan. Metode yang digunakan untuk menyelesaikan permasalahan pada penelitian ini adalah dengan menggunakan REST API. Metode penelitian yang digunakan pada penelitian ini dapat dilihat pada Gambar 1. dibawah ini



Gambar 1. Metode Penelitian

2.1 Identifikasi Masalah

Tahap pertama adalah melakukan identifikasi masalah. Permasalahan yang ditemukan pada Masjid Al-Muhajirin adalah masjid tersebut masih belum dilengkapi dengan teknologi yang cukup. Segala informasi yang diberikan terkait jadwal kajian atau Peringatan Hari Besar Islam (PHBI) masih diinformasikan melalui penyebaran pamflet, papan pengumuman, pengumuman melalui pengeras suara, dan spanduk. Informasi tersebut juga diinformasikan ketika akan dilaksanakan acara besar lainnya. Hal tersebut dinilai kurang efektif karena dibeberapa

kasus yang penulis temui, pada beberapa acara besar yang dijalankan di Masjid Al-Muhajirin jamaah yang datang jumlahnya sangat dikit. Permasalahan itu terjadi karena informasi yang disebar dengan cara seperti yang disebutkan diatas kurang efektif.

Permasalahan lainnya yaitu laporan keuangan mingguan atau bulanan yang pencatatannya masih dilakukan di buku tulis. Hal tersebut dinilai kurang efektif karena di zaman modern seperti sekarang ini penulisan laporan keuangan bisa lebih mudah di lakukan di komputer atau gadget. Penghitungan jumlah keuangan juga dihitung secara manual yang terdapat kemungkinan salah dalam perhitungannya.

2.2 Rancangan Program

Tahap yang kedua adalah merancang program. Pada tahap ini dijelaskan akan seperti apa aplikasi dibuat dengan menggunakan metode yang diusulkan dan bagaimana aplikasi dapat saling terhubung satu sama lain walaupun berbeda *platform*.

2.3 Rancangan Basis Data

Tahap selanjutnya adalah merancang basis data, pada tahapan ini berisi rancangan basis data yang disajikan menggunakan *class diagram*. *Class diagram* adalah diagram UML (*Unified Modelling Language*) yang menggambarkan kelas-kelas dalam sebuah sistem dan hubungannya antara satu dengan yang lain, serta dimasukkan pula atribut dan operasi.[2] Pada tahap ini dijelaskan bagaimana hubungan antara data satu dengan yang lain. Rancangan basis data berisi nama-nama tabel dan *field* dalam basis data.

2.4 Rancangan Layanan REST API

Setelah melakukan rancangan basis data, tahap selanjutnya adalah merancang layanan *web service* dengan menggunakan metode REST API. REST API digunakan sebagai penghubung atau jembatan antar dua aplikasi yang berbeda *platform*. Pada penelitian ini dua *platform* tersebut adalah *web* dan Android. Tahap rancangan layanan REST API disajikan dalam bentuk tabel yang berisi nama layanan, fungsi atau metode REST API yang digunakan, path atau endpoint, parameter, dan keluaran dari layanan yang disediakan.

Web service adalah fasilitas yang disediakan oleh situs *web* sebagai layanan dalam bentuk informasi untuk sistem lain untuk memungkinkan interaksi antara semua sistem terkait. *Web service* adalah sistem perangkat lunak yang terdiri dari kumpulan fungsi atau metode yang dapat disimpan di *server* dan dipanggil oleh klien.[3]

Web service dapat didefinisikan sebagai cara pertukaran data, terlepas dari mana *database* tertanam, bahasa di mana aplikasi yang memproses data dibangun, dan *platform* di mana data diproses. *Web service* dapat menjembatani antara berbagai sistem yang ada. Hal ini karena *web service* dapat mendukung interoperabilitas.[4]

Representational State Transfer (REST) adalah sebuah gaya arsitektur untuk pendistribusian sistem *hypermedia*. Arsitektur REST adalah arsitektur *client – server* dimana *client* mengirim *request* pada *server* dan *server* memproses *request* dan mengembalikan sebuah *response* (transaksi). Setiap transaksi bersifat independen dan tidak terkait dengan transaksi lainnya (*stateless*). Hal tersebut yang membuat aplikasi REST sederhana dan ringan.[5]

REST menggunakan konsep *Uniform Resource Identifier* (URI) yang dimana ada sebuah metode yang digunakan. Secara default yang digunakan adalah GET. Dalam arsitektur REST terdapat banyak metode HTTP yang bisa digunakan. Berikut adalah beberapa *method* yang umum digunakan:[6]

- GET adalah *method* yang digunakan untuk membaca suatu data pada *database*.
- POST adalah *method* yang digunakan untuk membuat data baru atau memperbaiki data yang sudah ada pada *database*.
- PUT adalah *method* yang digunakan untuk memperbarui data yang ada pada *database*. Mirip seperti POST, namun PUT hanya berfungsi untuk mengubah tidak bisa membuat.
- PATCH adalah *method* yang digunakan untuk memperbarui data yang ada pada *database*. Mirip seperti PUT, yang membedakan adalah PATCH hanya bisa digunakan untuk mengubah beberapa *field* dalam sebuah *record*, bukan melakukan *update* semua *field* dalam sebuah *record* seperti PUT.
- DELETE adalah *method* yang digunakan untuk menghapus *resource* yang ada.

API (*Application Programming Interface*) merupakan suatu *interface* yang dapat memberikan informasi dari satu aplikasi ke aplikasi lainnya. API memiliki peran sebagai perantara yang saling memberikan informasi antar berbagai aplikasi yang berbeda baik secara *offline* maupun *online* dan lintas *platform*. API adalah sebuah *tools* yang membuat beberapa sistem dapat saling terhubung. API mempunyai dua bagian, yaitu *server* yang berfungsi sebagai penyedia dari API dan *client* yang berbentuk sebuah program yang mengetahui data apa saja yang tersedia pada API dan data dapat dimanipulasi sesuai dengan *request* pengguna. Pada API berbasis *website* dibagi menjadi dua yaitu REST API dan SOAP API. REST API adalah API berbasis *website* yang menggunakan teknologi REST dan menggunakan format JSON (*JavaScript Object Notation*).[5]

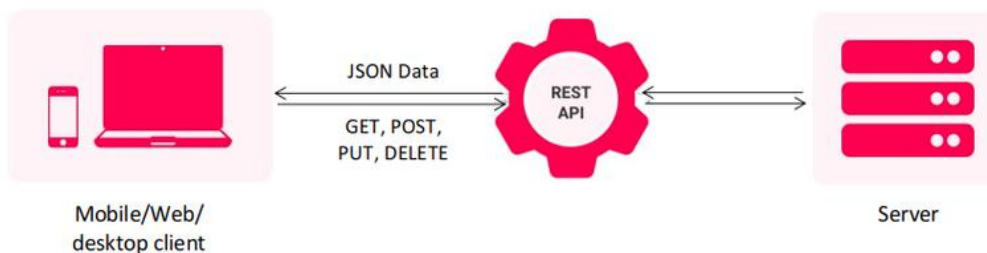
2.5 Rancangan Pengujian

Tahap terakhir adalah rancangan pengujian, tahap pengujian ini dilakukan dengan menggunakan pengujian *black box*. Pengujian *black box* adalah suatu pengujian untuk mengetahui apakah semua perangkat lunak telah berjalan dengan semestinya sesuai dengan kebutuhan fungsional yang didefinisikan. Metode pengujian *black box* memastikan bahwa perangkat lunak telah mendapatkan serangkaian kondisi inputan yang sepenuhnya telah menggunakan semua persyaratan fungsional pada suatu program [2].

3. HASIL DAN PEMBAHASAN

3.1 Rancangan Program

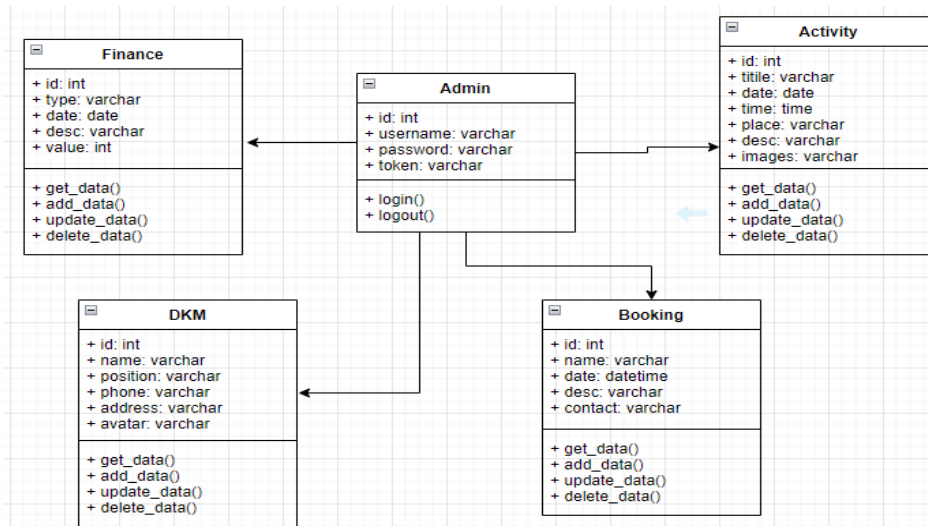
Cara kerja dari program ini adalah client dalam kasus dipenelitian ini yaitu *web* dan *Android* melakukan *request* ke *server* melalui perantara REST API. Setelah *client* meminta data ke *server*, *server* akan melakukan *response* terhadap data yang diminta *client*. *Request* atau permintaan *client* dilakukan dengan menggunakan metode HTTP GET, POST, PATCH, dan DELETE. *Response* yang dikembalikan oleh *server* dikirimkan berupa format JSON. Rancangan program dapat dilihat pada Gambar 2 dibawah ini.



Gambar 2. Rancangan Program

3.2 Rancangan Basis Data

Rancangan basis data ditampilkan dalam bentuk *class* diagram. *Class* diagram dapat menunjukkan keterkaitan antar setiap objek dalam aplikasi *web* dan *mobile* Masjid Al-Muhajirin, termasuk di dalamnya informasi kegiatan, laporan keuangan, daftar Dewan Kemakmuran Masjid, dan informasi sewa aula. *Class* diagram dapat dilihat pada Gambar 3 di bawah ini.



Gambar 3. Class Diagram

3.3 Rancangan Layanan REST API

Rancangan layanan REST API menjelaskan tentang berbagai layanan yang digunakan pada sistem REST API Masjid Al-Muhajirin. Rancangan layanan tersebut meliputi nama layanan, fungsi atau method yang digunakan, endpoint, parameter, dan keluaran dari layanan. Untuk lebih jelasnya rancangan ini bisa dilihat pada Tabel 1 di bawah ini.

Tabel 1. Rancangan Layanan REST API

Nama Layanan	Fungsi	Path	Parameter	Keluaran
Login	POST	/login	Id	Get token
Tampil data kegiatan	GET	/activity		Title, date, time, place, images
Detail kegiatan	GET	/activity/{id}	Id	Title, date, time, place, desc, other, images
Tambah kegiatan	POST	/activity	Request Body: Title, date, time, place, desc, other	Status, message
Edit kegiatan	PATCH	/activity/{id}	Parameter: id Title, date, time, place, desc, other, images	Status, message
Hapus kegiatan	DELETE	/activity/{id}	Id	Status, message
Tampil data keuangan	GET	/finance		Desc, date, type, value
Tambah laporan keuangan	POST	/finance	Request Body: Desc, type, value	Status, message
Edit laporan keuangan	PATCH	/finance/{id}	Parameter: id Desc, type, value	Status, message
Hapus laporan keuangan	DELETE	/finance/{id}	Id	Status, message
Tampil data sewa aula	GET	/booking		Name, date, desc, contact
Tambah sewa aula	POST	/booking	Request body: Name, date, desc, contact	Status, message
Edit sewa aula	PATCH	/booking/{id}	Parameter: id Name, date, desc, contact	Status, message
Hapus sewa aula	DELETE	/booking/{id}	Id	Status, message
Tampil data DKM	GET	/dkm		Name, position, avatar
Detail data DKM	GET	/dkm/{id}	Id	Name, position, phone, address, avatar
Tambah DKM	POST	/dkm	Request body: Name, position, phone, address, avatar	Status, message
Edit DKM	PATCH	/dkm/{id}	Parameter: id Name, position, phone, address, avatar	Status, message

Nama Layanan	Fungsi	Path	Parameter	Keluaran
Hapus DKM	DELETE	/dkm/{id}	Id	Status, message

3.4 Rancangan Layar

Rancangan layar merupakan desain untuk tampilan pada sebuah aplikasi *web* dan Android yang berfungsi untuk menggambarkan sistem yang akan dibangun.

3.4.1 Rancangan Layar Aplikasi Android

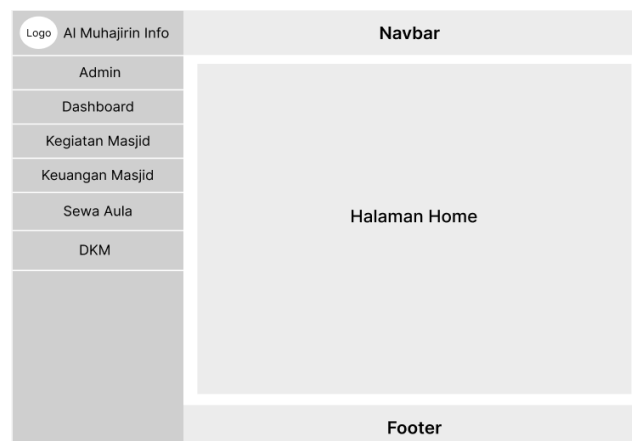
Rancangan layar aplikasi Android memberikan gambaran mengenai tampilan aplikasi yang akan dibangun. Gambar 4 merupakan tampilan dari rancangan layar aplikasi Android.



Gambar 4. Rancangan Layar Android

3.4.2 Rancangan Layar Aplikasi Web

Rancangan layar aplikasi *web* menggambarkan tampilan dari aplikasi *web* yang akan dibangun. Terdapat beberapa menu atau halaman dalam aplikasi *web*, diantaranya halaman *dashboard*, kegiatan masjid, keuangan masjid, sewa aula, dan DKM (Dewan Kemakmuran Masjid).



Gambar 5. Rancangan Layar Web

3.5 Pengujian Program

Pengujian program dilakukan menggunakan teknik pengujian black box. Pengujian ini dilakukan untuk menguji keberhasilan layanan REST API terhadap aplikasi *web* dan Android yang sudah dibangun dan memastikan kedua aplikasi dapat terhubung dan berjalan dengan baik tanpa ada kesalahan.

3.5.1 Pengujian Program Android

a. Pengujian *Method GET*

Tujuan dari pengujian ini adalah untuk mendapatkan data yang diminta oleh *client* dan ditampilkan pada aplikasi Android. Berikut hasil pengujian *method GET* pada Gambar 6.



Gambar 6. Pengujian *Method GET* Android

b. Pengujian *Method POST*

Tujuan dari pengujian ini adalah untuk mengubah data yang akan diubah pada *database* dan menampilkan hasilnya pada aplikasi. Berikut hasil pengujian *method POST* pada Gambar 7.



Gambar 7. Pengujian *Method POST* Android

c. Pengujian *method* PATCH

Tujuan dari pengujian ini adalah untuk mengubah data yang akan diubah pada *database* dan menampilkan hasilnya pada aplikasi. Berikut hasil pengujian *method* PATCH pada Gambar 8.

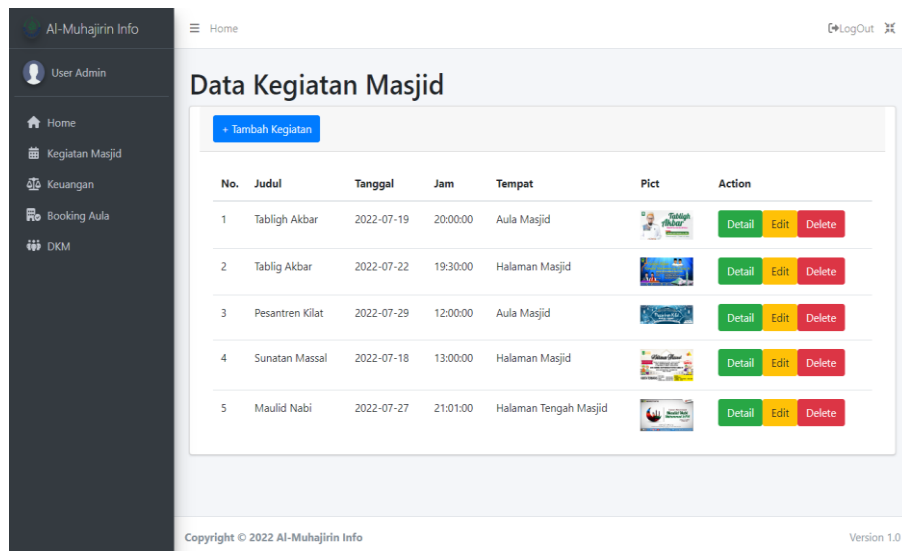


Gambar 8. Pengujian *Method* PATCH Android

3.5.2 Pengujian Program Web

a. Pengujian *method* GET

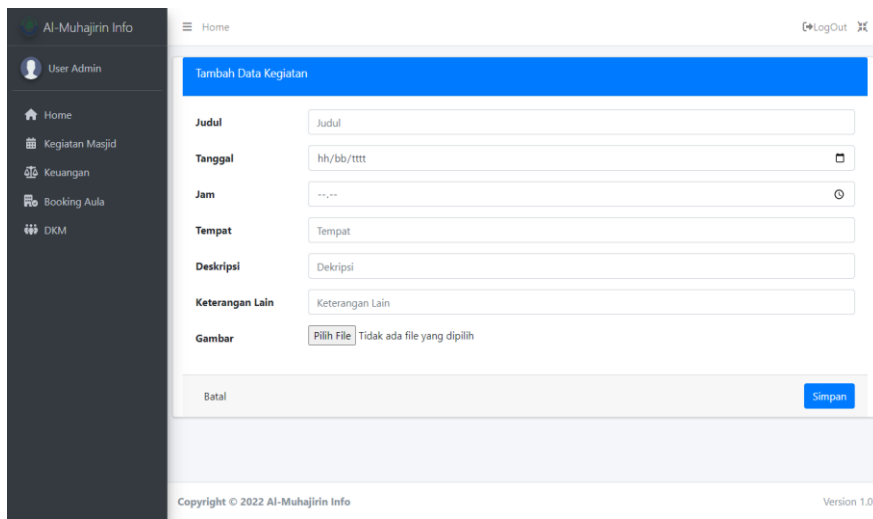
Tujuan dari pengujian ini adalah untuk mendapatkan data yang diminta oleh *client* dan ditampilkan pada aplikasi *web*. Berikut hasil pengujian *method* GET pada Gambar 9.



Gambar 9. Pengujian *Method* GET Web

b. Pengujian *method* POST

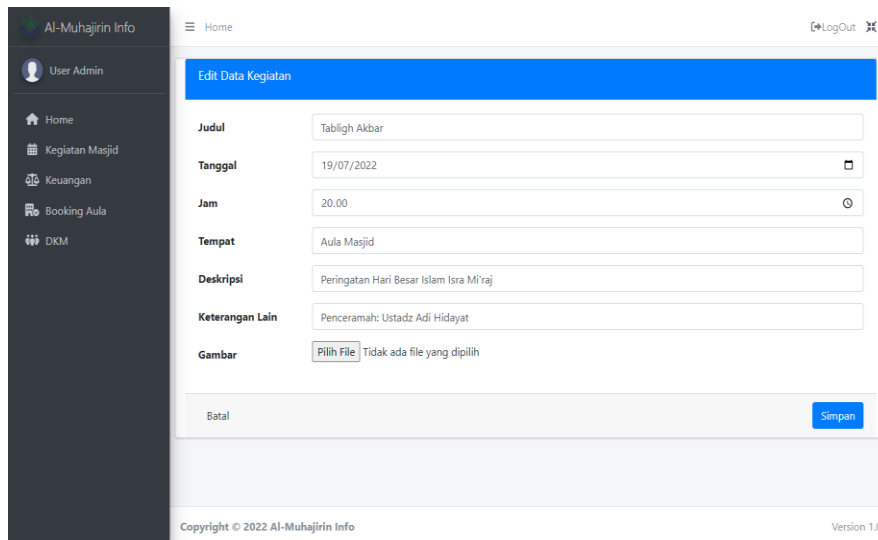
Tujuan dari pengujian ini adalah untuk menambah data yang diinput oleh *client* ke server dengan menggunakan *method* POST. Pengujian *method* POST pada web dapat dilihat pada Gambar 10.



Gambar 10. Pengujian *Method POST Web*

c. Pengujian *method PATCH*

Tujuan dari pengujian ini adalah untuk mengubah data yang akan diubah pada *database* dan menampilkan hasilnya pada aplikasi. Berikut hasil pengujian *method PATCH* pada Gambar 6.



Gambar 11. Pengujian *Method PATCH Web*

3.5.3 Hasil Pengujian *Black Box*

Dari pengujian yang sudah dilakukan dapat disimpulkan, aplikasi *web* dan Android dibangun tanpa ada kesalahan. Aplikasi pada dua platform tersebut dapat menjalankan layanan REST API dengan baik dan sesuai dengan harapan. Hasil pengujian *black box* dapat dilihat pada Tabel 2.

Tabel 2. Hasil Pengujian *Black Box*

No	Nama Layanan	Status
1.	Pengujian <i>method GET</i> kegiatan, keuangan, sewa aula, dan DKM	Berhasil
2.	Pengujian <i>method POST</i> kegiatan, keuangan, sewa aula, dan DKM	Berhasil
3.	Pengujian <i>method PATCH</i> kegiatan, keuangan, sewa aula, dan DKM	Berhasil
4.	Pengujian <i>method DELETE</i> kegiatan, keuangan, sewa aula, dan DKM	Berhasil

4. KESIMPULAN

Setelah melakukan proses perancangan, pembuatan, serangkaian pengujian, serta melakukan implementasi REST API pada aplikasi pusat informasi Masjid Al-Muhajirin Larangan Indah, maka dapat diambil beberapa kesimpulan. *Representational State Transfer* (REST) API dapat diterapkan pada *platform web* dan Android dengan baik. Hasil pengujian dengan menggunakan metode *black box* menunjukkan bahwa aplikasi *web*, Android, dan layanan REST API dapat berjalan sesuai harapan dan tanpa ada kesalahan. Sistem yang dibangun juga dapat digunakan untuk menyebarkan dan menerima segala informasi seputar Masjid Al-Muhajirin dengan lebih efektif. Sistem juga bisa melakukan pencatatan laporan keuangan dan menghitung total secara otomatis laporan keuangan yang diinput.

DAFTAR PUSTAKA

- [1] I. Hardiyatna, "Implementasi Web Service Menggunakan Metode RESTful untuk Aplikasi Publikasi Kegiatan dan Keuangan Pada Masjid Jami' Al Muhajirin Kabupaten Tangerang," Universitas Budi Luhur, 2018.
- [2] H. Wakkang and B. Irianto, "Implementasi Web Service Dengan Metode REST API Untuk Integrasi Data CoVid 19 Di Sulawesi Selatan," vol. 2, no. 1, pp. 12–22, doi: 10.31850/jsilog.v2i1.
- [3] F. Muzakki and A. W. Utami, "Rancang Bangun Sistem Informasi Penilaian Kinerja PNS Berbasis Web Service (Studi Kasus: DISKOMINFO Kabupaten Nganjuk)," *J. Manaj. Inform.*, vol. 11, pp. 164–172, 2020, [Online]. Available: <https://jurnalmahasiswa.unesa.ac.id/index.php/11/article/view/35981>
- [4] A. Pamuji, "Naskah publikasi rancang bangun web service menggunakan representational state transfer untuk pengolahan data barang," *Fak. Teknol. Inf. dan Elektro Univ. Teknol. Yogyakarta*, vol. 3, no. 1, pp. 100–104, 2020.
- [5] W. G. Wardhana, I. Arwani, and B. Rahayudi, "Implementasi Teknologi Restful Web Service Dalam Pengembangan Sistem Informasi Perekaman Prestasi Mahasiswa Berbasis Website (Studi Kasus: Fakultas Teknologi Pertanian Universitas Brawijaya)," *J. Pengemb. Teknol. Inf. dan Ilmu Komputer; Vol 4 No 2*, vol. 4, no. 2, pp. 680–689, 2020, [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/7024%0Ahttp://j-ptiik.ub.ac.id>
- [6] A. Firdaus, S. Widodo, A. Sutrisman, S. G. Fadhilah Nasution, and R. Mardiana, "Rancang Bangun Sistem Informasi Perpustakaan Menggunakan Web Sevice Pada Jurusan Teknik Komputer Polsri," *J. Inform.*, vol. 5, no. 2, pp. 81–87, 2019.
- [7] H. F. Herdiyatomoko, "Implementasi Web Service Pada Aplikasi Berbasis Web Untuk Mendukung Sistem Penilaian Siswa di SMP Indriasana Palembang Implementation of Web Services in Web-Based Applications to Support the Student Assessment System at SMP Indriasana Palembang," vol. 12, no. 1, pp. 13–24, 2022.
- [8] R. Somya, T. Michelle, and E. Nathanael, "Pengembangan Sistem Informasi Pelatihan Berbasis Web," vol. 16, no. 1, pp. 51–58, 2019.
- [9] R. Zarkasih, D. Febri Willianto, and U. Islam Nusantara Bandung, "Implementasi Restful Web Service Pada Aplikasi Pemesanan Suku Cadang Berbasis Web Di Ahass Munjul Motor," *J. Comput. Bisnis*, vol. 15, no. 1, pp. 38–45, 2021.
- [10] F. Al Isfahani, F. Nugraha, R. Mubarak, and A. Rahmatulloh, "Implementasi Web Service untuk Aplikasi Pemantau Coronavirus Disease 2019 (COVID-19)," *Innov. Res. Informatics*, vol. 2, no. 1, pp. 33–39, 2020, doi: 10.37058/innovatics.v2i1.1696.