BAB IV
HASIL PENGUJIAN DAN ANALISIS

Hasil dan pengujian sistem Layanan IoT dilakukan dengan pengujian aplikasi terhadap server, pengujian aplikasi dalam pembuatan API, pengujian aplikasi terhadap alat implementasi, respon server terhadap *request*, respon alat implementasi terhadap *request* dan berapa lama waktu yang di butuhkan saat menerima respon. Adapun alat implementasinya berupa tiga buah *internet lamp* yang dibangun menggunkan *microcontroller* ESP8266 yang telah dipasang API yang dihasilkan oleh sistem Layanan IoT. Pengujian pada *internet lamp* ini dengan cara menguji pengontrolan melalui jaringan internet, serberapa cepat respon yang diberikan saat request di kirim melalui aplikasi Layanan IoT dan pengujian terhadap tempat.

Hasil dan pengujian dilakukan menggunakan *smartphone* berbasis Android yang berfungsi sebagai antarmuka dari sistem ke pengguna. Sebelum pengujian, *smartphone* yang digunakan harus terlebih dahulu diinstal aplikasi yang telah dibangun.

4.1 Hasil

Antarmuka aplikasi merupakan tampilan yang akan digunakan user untuk menjalankan perangkat lunak Layanan IoT untuk menghasikan API dan mengontrol internet lamp yang sudah dibangun. Berikut adalah hasil dari perancangan yang telah dibuat:



(A)



(B)

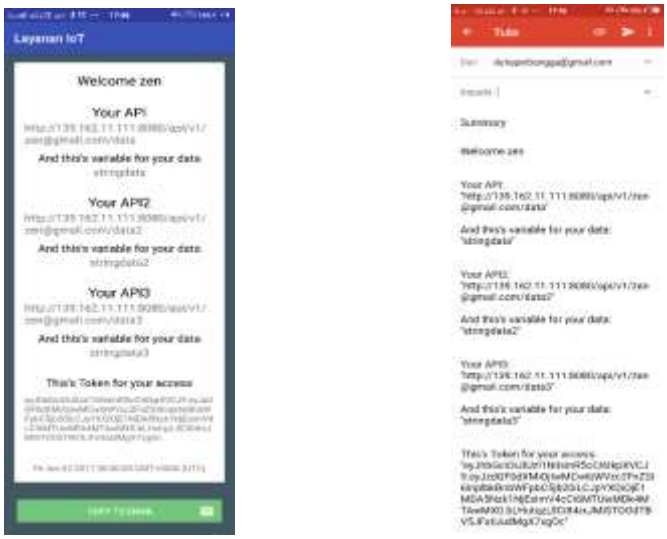
Gambar 4-1 (A) *Register* dan (B) *Login*

Gambar 4-1 merupakan antarmuka halaman untuk *user* melakukan *Registrasi* dan *Login*. Halaman *Register* yang ditunjukan oleh gambar (A) dan untuk halaman *Login* ditunjukan oleh gambar (B).



Gambar 4-2 Halaman muka Perangkat lunak Layanan IoT

Gambar 4-2 merupakan halaman utama perangkat lunaka Layanan IoT. Tombol DEVELOPER untuk menghasilkan API dari perangkat lunak tersebut, tombol ADD LAMP untuk menambah jumlah *internet lamp* yang akan dikontrol, rombol RESET untuk menghapus semua *internet lamp* yang sudah didaftarkan.



(A) (B)

Gambar 4-3 (A) Informasi API, (B) Informasi API disalin ke Email

Gambar 4-3 menunjukan halaman informasi API yang dihasilkan oleh perangkat lunak Layanan IoT yang disalin ke email. Halaman informasi API

ditunjukan oleh gambar (A). Pada gambar (A) terdapat tombol COPY TO EMAIL yang berfungsi untuk menyalin informasi API ke email. Halaman Informasi API disalin ke email ditunjukan oleh gambar (B).



(A)



(B)

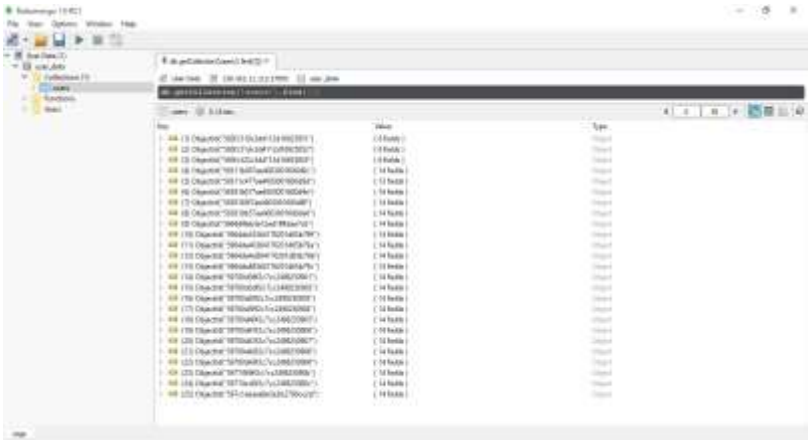
Gambar 4-4 (A)*Input Serial Number* dan (B)Kontrol *Internet lamp*

Gambar 4-4 murupakan halaman untuk *input serial number* dan kontrol *internet lamp.* Halaman input serial number di tunjukan oleh gambar (A). Pada gambar (A) terdpat tombol VALIDASI untuk memeriksa serial number yang di input benar atau salah. Apabila serial number yang di input benar maka akan langsung menuju halaman pengontrolan *internet lamp* yang ditunjukan oleh gambar (B). Pada gambar (B) terdapat tombol REFRESH, ON, OFF, dan BACK. Tombol REFRESH yang berfungsi untuk memeriksa apakah device *internet lamp* terhubung ke internet atau tidak, tombol ON untuk menghidupkan lampu, tombol OFF untuk mematikan lampu, dan tombol back untuk kembali ke halaman muka.



Gambar 4-5 *Device Internet lamp*

Gambar 4-5 yaitu merupakan hasil dari pembangunan *device internet lamp* yang berfungsi sebagai implementasi dari sistem Layanan IoT. *Device Internet lamp* dapat dikontrol dari jarak jauh melalui jaringan internet menggunakan perangkat lunak berbasis Android.



Gambar 4-6 Halaman Admin

Gambar 4-6 merupakan tampilan halaman admin yang berfungsi untuk mengatur user. Admin dapat menghapus, merubah, atau menambahkan data ke basis data server melalui halaman admin ini.

4.2 Pengujian Koneksi ke Server

Pengujian koneksi ke server bertujuan untuk mengetahui kesesuaian data dan kesetabilan koneksi untuk berkomunikasi dengan server dengan waktu *delay* yang ditentukan pada sistem. Tabel 4-1 menunjukan hasil pengujian koneksi ke server.

Tabel 4-1 Hasil Pengujuan Koneksi Keserver

No.	Waktu Delay /millisecond	Data	Respon Server	Rata-rata Waktu Respon /millisecond	Kesesuaian Data
1.	1000	Data1 – Data100	100%	3,716	100%
2.	500	Data1 – Data100	100%	3,570	100%
3.	400	Data1 – Data100	100%	4,348	100%
4.	300	Data1 – Data100	100%	3,869	94%

5.	200	Data1 – Data100	100%	4,234	62%

Hasil pengujian menunjukan bahwa minimal *delay* yang masih bisa digunakan sisitem agar memaksimalkan tingkat keberhasilan dalam komunikasi data yaitu 300 *millisecond* dengan tingkat keberhasilan diatas 90%.

4.3 Pengujian Komunikasi Data Aplikasi ke Server

Komunikasi data antara aplikasi dan server menggunakan HTTP *Request* dengan metode POST, PUT, GET. POST yang artinya aplikasi mengirim data ke server lalu server menyimpannya di basis data. PUT yang artinya aplikasi merubah data yang telah ada di basis data server. GET yang artinya aplikasi meminta data dari basis data server. Berikut tabel hasil pengujian:

Tabel 4-2 Hasil Pengujian POST Request dari Aplikasi ke Server

No	POST Request	Data	Respon Server	Waktu /milliesecond	Status Request
1.	Uji 1	Registrasi User	Ok	173,950	Berhasil
2.	Uji 2	Registrasi User	Ok	199,026	Berhasil
3.	Uji 3	Registrasi User	Ok	142,082	Berhasil
4.	Uji 4	Registrasi User	Ok	138,124	Berhasil
5.	Uji 5	Registrasi User	Ok	144,792	Berhasil

Hasil pengujian POST Request menunjukkan bahwa komunikasi data dari aplikasi ke server berhasil dengan kesesuaian data yang terdapat pada basis data server dan rata-rata waktu 159,594 *millisecond*.

Tabel 4-3 Hasil Pengujian PUT Request dari Aplikasi ke Server

No	PUT Request	Data	Respon Server	Waktu /millisecond	Status Request
1.	Uji 1	111	Ok	6,866	Berhasil
2.	Uji 2	222	Ok	3,647	Berhasil
3.	Uji 3	333	Ok	6,382	Berhasil
4.	Uji 4	444	Ok	2,600	Berhasil
5.	Uji 5	555	Ok	4,313	Berhasil

Hasil pengujian PUT Request menunjukkan bahwa komunikasi data dari aplikasi ke server berhasil dengan kesesuaian data yang terdapat pada basis data server dan rata-rata waktu 4,761 *millisecond.*

Tabel 4-4 Hasil Pengujian GET Request dari Aplikasi ke Server

No	GET Request	Data	Respon Server	Waktu / millisecond	Status Request
1.	Uji 1	Cek Device	Ok	2,672	Berhasil
2.	Uji 2	Cek Device	Ok	3,439	Berhasil
3.	Uji 3	Cek Device	Ok	2,639	Berhasil
4.	Uji 4	Cek Device	Ok	3,260	Berhasil
5.	Uji 5	Cek Device	Ok	3,647	Berhasil

Hasil pengujian GET Request menunjukkan bahwa komunikasi data dari aplikasi ke server berhasil dengan rata-rata waktu 3,131 *millisecond*.

4.4 Pembuatan API pada Aplikasi Layanan IoT

API berfungsi untuk komunikasi antara aplikasi, server dan *internet lamp*. API digunakan oleh developer sebagai data mentah yang akan diolah kembali sesuai keinginan developer untuk dipakai sesuai kepentinganya. Pada sistem ini API akan diimplementasikan untuk pengontrolan *internet lamp.* Gambar IV-3(A) menunjukan pembuatan API pada Aplikasi Layanan IoT dan gambar IV-3(B) menenjukan API berhasil disalin ke *email*.

4.5 Pengujian Komunikasi Data *Internet lamp* ke Server

Komunikasi data antara *internet lamp* dan server menggunakan HTTP *Request* dengan metode PUT, GET. PUT yang artinya *internet lamp* merubah data yang telah ada di basis data server. GET yang artinya *internet lamp* meminta data dari basis data server. Berikut tabel hasil pengujian.

Tabel 4-5 Hasil Pengujian PUT *Request* dari *internet lamp* 1 ke Server

No	Device	PUT Request	Data	Respon Server	Waktu /Second	Status Request
1.	*Internet lamp* 1	Uji 1	Status Device	Ok	13,282	Berhasil
		Uji 2	Status Device	Ok	3,411	Berhasil
		Uji 3	Status Device	Ok	2,199	Berhasil
		Uji 4	Status Device	Ok	2,400	Berhasil
		Uji 5	Status Device	Ok	2,044	Berhasil

Tabel 4-6 Hasil Pengujian GET *Request* dari *Internet lamp* 1 ke Server

No	Device	GET Request	Data	Respon Server	Waktu /Second	Status Request
1.	*Internet lamp* 1	Uji 1	Device Data	Ok	28,268	Berhasil
		Uji 2	Device Data	Ok	4,800	Berhasil
		Uji 3	Device Data	Ok	2,815	Berhasil
		Uji 4	Device Data	Ok	3,270	Berhasil
		Uji 5	Device Data	Ok	5,175	Berhasil

Tabel 4-7 Hasil Pengujian PUT *Request* dari *Internet lamp* 2 ke Server

No	Device	PUT Request	Data	Respon Server	Waktu /Second	Status Request
1.	*Internet lamp* 2	Uji 1	Status Device	Ok	12,669	Berhasil
		Uji 2	Status Device	Ok	4,109	Berhasil
		Uji 3	Status Device	Ok	2,217	Berhasil
		Uji 4	Status Device	Ok	2,122	Berhasil
		Uji 5	Status Device	Ok	14,617	Berhasil

Tabel 4-8 Hasil Pengujian GET *Request* dari *Internet lamp* 2 ke Server

No	Device	GET Request	Data	Respon Server	Waktu /Second	Status Request
1.	*Internet lamp* 2	Uji 1	Device Data	Ok	31,256	Berhasil
		Uji 2	Device Data	Ok	4,340	Berhasil
		Uji 3	Device Data	Ok	3,336	Berhasil
		Uji 4	Device Data	Ok	3,889	Berhasil
		Uji 5	Device Data	Ok	4,636	Berhasil

Tabel 4-9 Hasil Pengujian PUT *Request* dari *Internet lamp* 3 ke Server

No	Device	PUT Request	Data	Respon Server	Waktu /Second	Status Request
1.	*Internet lamp* 3	Uji 1	Status Device	Ok	3,427	Berhasil
		Uji 2	Status Device	Ok	2,959	Berhasil
		Uji 3	Status Device	Ok	2,642	Berhasil
		Uji 4	Status Device	Ok	3,691	Berhasil
		Uji 5	Status Device	Ok	3,606	Berhasil

Tabel 4-10 Hasil Pengujian GET *Request* dari *Internet lamp* 3 ke Server

No	Device	GET Request	Data	Respon Server	Waktu /Second	Status Request
1.	*Internet lamp* 3	Uji 1	Device Data	Ok	1,856	Berhasil
		Uji 2	Device Data	Ok	5,459	Berhasil
		Uji 3	Device Data	Ok	2,779	Berhasil
		Uji 4	Device Data	Ok	1,966	Berhasil
		Uji 5	Device Data	Ok	2,426	Berhasil

Hasil pengujian Berdasarkan tabel 4-5, tabel 4-6, tabel 4-7, tabel 4-8, tabel 4-9, tabel 4-10 *internet lamp* berhasil mengirim data ke server dengan kesesuaian data yang ada pada basis data pada server dan *internet lamp* berhasil menerima data yang diminta dari server

4.6 Pengujian Perangkat Lunak dan Pengontrolan *Internet lamp*

Pengontrolan *internet lamp* dilakukan menggunakan *smartphone* berbasis Android yang berfungsi sebagai antarmuka dari sistem ke pengguna. Untuk dapat mengontrol *internet lamp*, pengguna harus menghubungkan *internet lamp* dengan jaringan internet yang tersedia ditempat tersebut. Setelah berhasil terhubung, pengguna dapat menjalankan aplikasi yang telah pasang lalu *login*, apabila belum mempunyai akun bisa *register* terlebih. Setelah itu *Add Device*, lalu mengisi *serial number internet lamp* untuk mendaftarkan *internet lamp* pada sistem kemudia dapat melakukan pengontrolan *internet lamp*.

Pada tampilan halaman pengontrolan *internet lamp* terdapat 4 tombol yaitu Refresh, On, Off dan Back. Tombol Refresh berfungsi untuk cek status *internet lamp* terhubung ke jaringan internet atau tidak. Saat menekan tombol refresh sistem akan meminta data ke server untuk menampilkan status *internet lamp*. Tombol On berfungsi untuk menyalakan lampu dan akan menampilkan status lampu nyala diatasnya. Saat menekan tombol On sistem akan mengirim data ke server lalu server menyimpan data tersebut di basis data sebagai acuan agar lampu menyala saat *internet lamp* meminta data ke server. Tombol Off berfungsi untuk mematikan lampu dan akan menampilkan status lampu mati diatasnya. Saat menekan tombol Off sistem akan mengirim data ke server lalu server menyimpan data tersebut di basis data sebagai acuan agar lampu mati saat *internet lamp* meminta data ke server. Tombol Back berfungsi untuk kembali ke halaman muka. Tabel 4-11 menunjukan hasil pengujian kontrol *Internet lamp* dan tabel 4-12 menunjukan hasil pengujian kontrol *Internet lamp* terhadap tempat dan jarak.

Tabel 4-11 Hasil Pengujian Kontrol *Internet lamp*

No.	Device	Data di Aplikasi	Data di Server	Data di *Internet lamp*	Status Lampu
1.	*Internet lamp* 1	on	on	on	Nyala
2.	*Internet lamp* 2	on	on	on	Nyala
3.	*Internet lamp* 3	on	on	on	Nyala
4.	*Internet lamp* 1	off	off	off	Mati
5.	*Internet lamp* 2	off	off	off	Mati
6.	*Internet lamp* 3	off	off	off	Mati

Hasil Pengujian menunjukan bahwa aplikasi, server dan *internet lamp* mengirim dan menerima data dengan benar serta bekerja dengan baik sebagaimana fungsinya masing-masing.

Tabel 4-12 Hasil Pengujian Kontrol *Internet lamp* terhadap tempat dan jarak

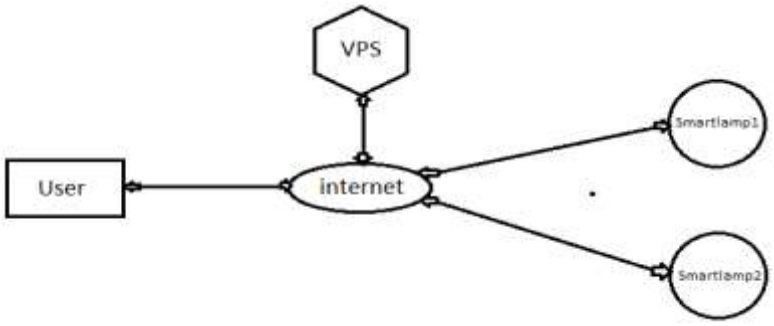
No.	Device	Tempat	Jarak wifi /meter	Terhalang Bangunan	Waktu Respon /milliesecond	Status Lampu
1.	*Internet lamp* 1	Dalam Ruangan	10	Ya	4000	Nyala
			20		6000	Nyala
			30		-	Mati
		Luar Ruangan	10	Tidak	3000	Nyala
			20		4000	Nyala
			30		7000	Nyala
2.	*Internet lamp* 2	Dalam Ruangan	10	Ya	4000	Nyala
			20		6000	Nyala
			30		-	Mati
		Luar Ruangan	10	Tidak	3000	Nyala
			20		3000	Nyala

			30		7000	Nyala
3.	*Internet lamp* 3	Dalam Ruangan	10	Ya	4000	Nyala
			20		7000	Nyala
			30		-	Mati
		Luar Ruangan	10	Tidak	3000	Nyala
			20		4000	Nyala
			30		-	Mati

Hasil pengujian kontrol *internet lamp* terhadap tempat dan jarak wifi pada tabel 4-12 menunjukan bahwa jarak wifi dan tempat berpengaruh terhadap kinerja *internet lamp* jika melihat pada Waktu Respon walaupun tidak signifikan. Dapat disimpulkan bahwa kinerja *internet lamp* akan sangat berpengaruh terhadapan kekuatan koneksi yang ada dan pada jarak wifi 30 meter berada didalam ruangan device *internet lamp* tidak mendeteksi wifi yang artinya device *internet lamp* tidak terhubung ke internet dan tidak dapat dikontrol.

4.7 Pengujian Satu User mengontrol Dua *Internet lamp*

Pengujian ini bertujuan untuk menunjukan bahwa satu user dapat mengontrol lebih dari satu *internet lamp* yang dapat dilihat di gambar 4-7. Hasil pengujian dapat dilihat pada tabel 4-13.



Gambar 4-7 Blok Diagram Satu User mengontrol Dua

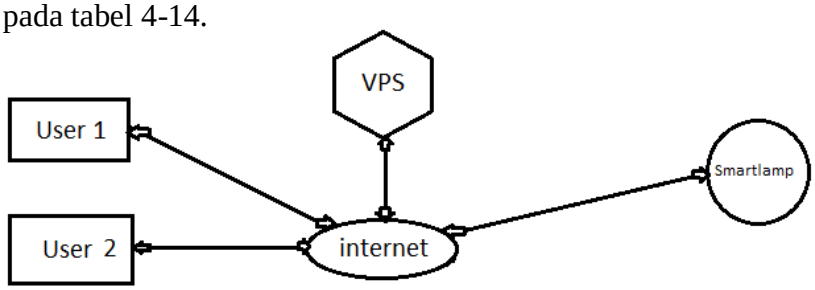
Tabel 4-13 Hasil Pengujian Satu User mengontrol Dua *Internet lamp*

No.	User	Data di Kirim	Data di Server	Data di *Internet lamp*	Status Device	Device
1.	User1	on	on	on	Nyala	*Internet lamp* 1
		on	on	on	Nyala	*Internet lamp* 2

Berdasarkan hasil pengujian menunjukan bahwa satu *user* dapat mengontrol lebih dari satu *internet lamp*.

4.8 Pengujian Dua *User* mengontrol Satu *Internet lamp*

Pengujian ini bertujuan untuk menunjukan bahwa satu *internet lamp* dapat dikontrol lebih dari satu *user* yang dapat dilihat di gambar 4-8. Hasil pengujian dapat dilihat pada tabel 4-14.



Gambar 4-8 Blok Diagram Dua User mengontrol Satu *Internet lamp*

Tabel 4-14 Hasil Pengujian Dua User mengontrol Satu *Internet lamp*

No.	User	Data di Kirim	Data di Server	Data di *Internet lamp*	Status Device	Device
1.	User1	on	on	on	Nyala	*Internet lamp*
2.	User2	off	off	off	Mati	

Berdasarkan hasil pengujian menunjukan bahwa satu *internet lamp* dapat dikontrol lebih dari satu *user*.