

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini menjelaskan tentang pengujian sistem yang sebelumnya telah diimplementasikan. Tahap ini dilakukan untuk mengetahui apakah sistem berjalan sesuai dengan kebutuhan fungsional serta non-fungsional yang telah dirancang sebelumnya pada bab rekayasa kebutuhan.

6.1 Pengujian Kebutuhan Fungsional

Pengujian kebutuhan fungsional bersifat kualitatif baik terhadap perangkat keras dan perangkat lunak dalam proses implementasi sistem. Pengujian pada kebutuhan fungsional disesuaikan dengan kebutuhan fungsional pada rekayasa kebutuhan.

6.1.1 Pengujian Mekanisme Persiapan Kerja Perangkat *Gateway*

6.1.1.1 Tujuan Pengujian

Pengujian mekanisme persiapan kerja perangkat *gateway* ini mengacu pada rekayasa kebutuhan dan implementasi yang telah dijelaskan pada bab sebelumnya. Tujuan dari pengujian ini adalah untuk melakukan verifikasi terhadap implementasi yang telah dilakukan.

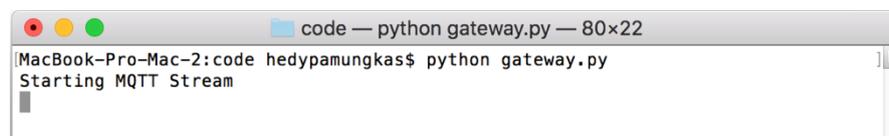
6.1.1.2 Prosedur Pengujian

Prosedur yang dilakukan pada pengujian ini yakni sebagai berikut :

1. Perangkat *gateway* melakukan koneksi terhadap *broker*.

6.1.1.3 Hasil dan Analisis Pengujian

Proses awal yang harus berjalan dari keseluruhan sistem ini ialah perangkat *gateway* harus sudah siap menerima permintaan baik dari perangkat sensor maupun aktuator. Pengujian mekanisme ini memastikan proses awal berjalan dengan baik dan sesuai dengan rekayasa kebutuhan.



```
code — python gateway.py — 80x22
MacBook-Pro-Mac-2:code hedyamungkas$ python gateway.py
Starting MQTT Stream
```

Gambar 6.1 *Gateway* telah siap menerima permintaan

Pada Gambar 6.1 menerangkan bahwa dengan keluaran seperti gambar diatas maka dapat dipastikan *gateway* sudah siap untuk menerima permintaan dari berbagai perangkat.

Tabel 6.1 Tabel Pengujian mekanisme persiapan kerja perangkat *gateway*

No.	Test Name	Test Case	Expected Result	Result	Status
1.	REQ-	Melakukan	Perangkat	Perangkat	Valid

No.	Test Name	Test Case	Expected Result	Result	Status
	PERVASIVE-KG-1000-Koneksi <i>gateway</i> ke <i>broker</i>	pengujian pada fungsi <i>gateway</i> <code>.connect ()</code> untuk menghubungkan perangkat <i>gateway</i> ke <i>broker</i>	<i>gateway</i> dapat terhubung ke <i>broker</i> . Hasilnya dapat dilihat pada terminal	<i>gateway</i> berhasil terhubung ke <i>broker</i> dan mencetak "Starting MQTT Stream", yakni perangkat <i>gateway</i> telah siap beroperasi.	

Pada Tabel 6.1 dapat diambil kesimpulan bahwa pengujian mekanisme persiapan kerja perangkat *gateway* sesuai antara hasil yang diharapkan dengan hasil yang terjadi pada pengujian ini serta dinyatakan 100% berhasil dilakukan pada sistem ini berdasarkan kolom status pada setiap prosedur pengujiannya.

6.1.2 Pengujian Mekanisme Persiapan Kerja dari Sensor dan Aktuator

6.1.2.1 Tujuan Pengujian

Pengujian mekanisme persiapan kerja dari perangkat sensor maupun aktuator ini mengacu pada rekayasa kebutuhan dan implementasi yang telah dijelaskan pada sebelumnya. Pengujian ini dilakukan untuk memverifikasi terhadap implementasi yang sudah dilakukan sebelumnya.

6.1.2.2 Prosedur Pengujian

Prosedur yang dilakukan pada pengujian ini yakni sebagai berikut :

1. Perangkat aktuator dan perangkat sensor melakukan koneksi ke jaringan *wifi*.
2. Perangkat aktuator dan perangkat sensor melakukan koneksi terhadap *broker*.
3. Perangkat aktuator dan perangkat sensor melakukan inisialisasi objek perangkat dari file *metadata*.

6.1.2.3 Hasil dan Analisis Pengujian

Pada mekanisme ini meliputi proses perangkat terkoneksi dengan *wifi* serta *broker* mqtt dan inisialisasi masing-masing perangkat berdasarkan *metadatanya*. Proses ini jg berlaku untuk keseluruhan perangkat baik perangkat aktuator maupun sensor.

```

cpp — BULB
PROBLEMS OUTPUT TERMINAL ... 4: Task - Monitor
> Executing task: platformio device monitor <
--- Miniterm on /dev/cu.wchusbserial1410 9600,8,N,1 ---
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---
...connecting... ok!
Model of Device Object
device Name      : BULB-001
device Category  : BULB
device Type      : actuator
device Ack Topic : kardel/ack/actuator/BULB-001
device Location  : kamar
Model of Service Object
service Name     : lamp
service Unit     :
service Data     : 0
service Name     : level
service Unit     : %
service Data     : 0
{"deviceName": "BULB-001", "category": "BULB", "deviceType": "actuator", "ackTopic": "karde
L/ack/actuator/BULB-001", "location": "kamar", "service": {"lamp": {"name": "lamp", "unit":
"0", "data": "0"}, "level": {"name": "level", "unit": "0", "data": "0"}}, "integration": {"m
ax": 2, "category": ["LDR", "PIR"]}}
kardel/broadcast
{"deviceName": "BULB-001", "category": "BULB", "deviceType": "actuator", "ackTopic": "karde
L/ack/actuator/BULB-001", "location": "kamar", "service": {"lamp": {"name": "lamp", "unit":
"0", "data": "0"}, "level": {"name": "level", "unit": "0", "data": "0"}}, "integration": {"m
ax": 2, "category": ["LDR", "PIR"]}}
kardel/broadcast
Registration Done

```

Gambar 6.2 Proses yang dilakukan perangkat aktuator

Pada Gambar 6.2 dijelaskan beberapa proses yang dilakukan perangkat gateway yakni meliputi pengujian koneksi perangkat ke jaringan *wifi* serta *broker* yang ditunjukkan pada output “connecting... ok”, sampai dengan tulisan “ok” maka perangkat sudah tersambung pada *wifi* dan *broker*. Kemudian perangkat aktuator akan mencetak model dari objek perangkat yang menandakan bahwa telah berhasil melakukan inisialisasi objek perangkat.

```

cpp — PIR
PROBLEMS OUTPUT TERMINAL ... 2: Task - Monitor
> Executing task: platformio device monitor <
--- Miniterm on /dev/cu.wchusbserial1410 9600,8,N,1 ---
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---
...connecting... ok!
Model of Device Object
device Name      : PIR-001
device Category  : PIR
device Type      : sensor
device Ack Topic : kardel/ack/sensor/PIR-001
device Location  : kamar
Model of Service Object
service Name     : motion
service Unit     :
service Data     : 0
msg sent
msg sent

incoming: kardel/ack/sensor/PIR-001 - {'replytopic_data': 'kardel/pervasive/kamar/senso
r/PIR-001/data', 'statusCode': 200}
Registration Done
connecting... ok!
{"deviceName": "PIR-001", "deviceType": "sensor", "service": {"motion": {"name": "motion"
, "unit": " ", "data": "1"}}}
{"deviceName": "PIR-001", "deviceType": "sensor", "service": {"motion": {"name": "motion"
, "unit": " ", "data": "1"}}}
{"deviceName": "PIR-001", "deviceType": "sensor", "service": {"motion": {"name": "motion"
, "unit": " ", "data": "1"}}}
{"deviceName": "PIR-001", "deviceType": "sensor", "service": {"motion": {"name": "motion"
, "unit": " ", "data": "1"}}}
{"deviceName": "PIR-001", "deviceType": "sensor", "service": {"motion": {"name": "motion"
, "unit": " ", "data": "1"}}}
{"deviceName": "PIR-001", "deviceType": "sensor", "service": {"motion": {"name": "motion"
, "unit": " ", "data": "1"}}}

```

Gambar 6.3 Proses yang dilakukan perangkat sensor PIR

Pada Gambar 6.3 dijabarkan beberapa proses yang dilakukan perangkat sensor menggunakan sensor PIR yakni pengujian koneksi perangkat untuk

terhubung ke jaringan *wifi* serta *broker* mqtt yang ditunjukkan pada keluaran “connecting... ok” pada *serial monitor*. Maka perangkat sudah terhubung ke jaringan *wifi* serta *broker*. Selanjutnya perangkat sensor PIR akan mencetak model dari objek perangkat yang menandakan bahwa telah berhasil melakukan inisialisasi objek perangkat.

```

> Executing task: platformio device monitor <

--- Miniterm on /dev/cu.wchusbserial1410 9600,8,N,1 ---
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---
.connecting... ok!
Model of Device Object
device Name      : LDR-001
device Category  : LDR
device Type      : sensor
device Ack Topic : kardel/ack/sensor/LDR-001
device Location  : kamar
Model of Service Object
service Name     : light
service Unit     :
service Data     : 0
msg sent
msg sent

incoming: kardel/ack/sensor/LDR-001 - {'replytopic_data': 'kardel/pervasive/kamar/sensor/LDR-001/data', 'statuscode': 200}
Registration Done
connecting... ok!
{"deviceName": "LDR-001","deviceType": "sensor","service": {"light": {"name": "light","unit": " ", "data": "1"}}}
{"deviceName": "LDR-001","deviceType": "sensor","service": {"light": {"name": "light","unit": " ", "data": "1"}}}
{"deviceName": "LDR-001","deviceType": "sensor","service": {"light": {"name": "light","unit": " ", "data": "1"}}}
{"deviceName": "LDR-001","deviceType": "sensor","service": {"light": {"name": "light","unit": " ", "data": "1"}}}
{"deviceName": "LDR-001","deviceType": "sensor","service": {"light": {"name": "light","unit": " ", "data": "1"}}}
{"deviceName": "LDR-001","deviceType": "sensor","service": {"light": {"name": "light","unit": " ", "data": "1"}}}

```

Gambar 6.4 Proses yang dilakukan perangkat sensor LDR

Pada Gambar 6.4 dijelaskan beberapa mekanisme yang dikerjakan oleh perangkat sensor LDR yakni meliputi pengujian koneksi perangkat untuk terhubung ke jaringan *wifi* serta *broker* mqtt. Keluaran yang ditunjukkan pada gambar tersebut menunjukkan bahwa perangkat tersebut telah berhasil terhubung ke jaringan *wifi* serta *broker*. Selain itu telah berhasil melakukan inisialisasi objek perangkat.

Tabel 6.2 Tabel Pengujian mekanisme persiapan kerja dari sensor dan aktuator

No	Test Name	Test Case	Expected Result	Result	Status
1.	REQ-PERVASIVE- PKSA- 1100- Koneksi aktuator ke jaringan <i>wifi</i>	Melakukan pengujian pada fungsi <i>Wifi.begin()</i> untuk menghubungkan perangkat aktuator ke	Perangkat aktuator dapat terhubung ke jaringan <i>wifi</i> . Hasilnya dapat dilihat pada <i>serial monitor</i>	Perangkat aktuator telah berhasil terhubung dengan memberikan keluaran berupa “connecting..	Valid

No	Test Name	Test Case	Expected Result	Result	Status
		jaringan <i>wifi</i>		. ok" pada <i>serial monitor</i>	
2.	REQ-PERVASIVE-PKSA-1101-Koneksi sensor PIR ke jaringan <i>wifi</i>	Melakukan pengujian pada fungsi <i>Wifi.begin()</i> untuk menghubungkan perangkat sensor PIR ke jaringan <i>wifi</i>	Perangkat sensor PIR dapat terhubung ke jaringan <i>wifi</i> . Hasilnya dapat dilihat pada <i>serial monitor</i>	Perangkat sensor PIR telah berhasil terhubung dengan memberikan keluaran berupa "connecting.. ok" pada <i>serial monitor</i>	Valid
3.	REQ-PERVASIVE-PKSA-1102-Koneksi sensor LDR ke jaringan <i>wifi</i>	Melakukan pengujian pada fungsi <i>Wifi.begin()</i> untuk menghubungkan perangkat sensor LDR ke jaringan <i>wifi</i>	Perangkat sensor LDR dapat terhubung ke jaringan <i>wifi</i> . Hasilnya dapat dilihat pada <i>serial monitor</i>	Perangkat sensor LDR telah berhasil terhubung dengan memberikan keluaran berupa "connecting.. ok" pada <i>serial monitor</i>	Valid
4.	REQ-PERVASIVE-PKSA-1103-Koneksi aktuator terhadap <i>broker</i>	Melakukan pengujian pada fungsi <i>client.begin()</i> untuk menghubungkan perangkat aktuator ke <i>broker</i>	Perangkat aktuator dapat terhubung ke <i>broker</i> . Hasilnya dapat dilihat" pada <i>serial monitor</i>	Perangkat aktuator telah berhasil terhubung dengan memberikan keluaran berupa "connecting.. ok" pada <i>serial</i>	Valid

No	Test Name	Test Case	Expected Result	Result	Status
				<i>monitor</i>	
5.	REQ-PERVASIVE-PKSA-1104-Koneksi sensor PIR terhadap <i>broker</i>	Melakukan pengujian pada fungsi <code>client.begin()</code> untuk menghubungkan perangkat sensor PIR ke <i>broker</i>	Perangkat aktuator dapat terhubung ke <i>broker</i> . Hasilnya dapat dilihat pada <i>serial monitor</i>	Perangkat sensor PIR telah berhasil terhubung dengan memberikan keluaran berupa "connecting.. . ok" pada <i>serial monitor</i>	Valid
6.	REQ-PERVASIVE-PKSA-1105-Koneksi sensor LDR terhadap <i>broker</i>	Melakukan pengujian pada fungsi <code>client.begin()</code> untuk menghubungkan perangkat sensor LDR ke <i>broker</i>	Perangkat aktuator dapat terhubung ke <i>broker</i> . Hasilnya dapat dilihat pada <i>serial monitor</i>	Perangkat sensor LDR telah berhasil terhubung dengan memberikan keluaran berupa "connecting.. . ok" pada <i>serial monitor</i>	Valid
7.	REQ-PERVASIVE-PKSA-1106-Inisialisasi objek aktuator	Melakukan pengujian pada fungsi <code>validateDataModel()</code> untuk melakukan verifikasi apakah inisialisasi objek aktuator yang dilakukan berhasil atau tidak. Inisialisasi ini berdasarkan informasi dari <i>metadata</i> perangkat	Perangkat aktuator telah memiliki objek aktuator. Hasilnya berupa tidak ada keterangan yang kosong setiap parameternya	Perangkat aktuator telah berhasil melakukan inisialisasi objek dengan memberikan keluaran <i>metadata</i> perangkat	Valid

No	Test Name	Test Case	Expected Result	Result	Status
		aktuator			
8.	REQ-PERVASIVE-PKSA-1107- Inialisasi objek sensor PIR	Melakukan pengujian pada fungsi <code>validate</code> <code>DataModel()</code> untuk melakukan verifikasi apakah inialisasi objek sensor PIR yang dilakukan berhasil atau tidak. Inialisasi ini berdasarkan informasi dari <i>metadata</i> perangkat sensor PIR	Perangkat sensor PIR telah memiliki objek aktuator. Hasilnya berupa tidak ada keterangan yang kosong setiap parameternya	Perangkat sensor PIR telah berhasil melakukan inialisasi objek dengan memberikan keluaran <i>metadata</i> perangkat	Valid
9.	REQ-PERVASIVE-PKSA-1108- Inialisasi objek sensor LDR	Melakukan pengujian pada fungsi <code>validate</code> <code>DataModel()</code> untuk melakukan verifikasi apakah inialisasi objek sensor LDR yang dilakukan berhasil atau tidak. Inialisasi ini berdasarkan informasi dari <i>metadata</i> perangkat sensor LDR	Perangkat sensor LDR telah memiliki objek aktuator. Hasilnya berupa tidak ada keterangan yang kosong setiap parameternya	Perangkat sensor LDR telah berhasil melakukan inialisasi objek dengan memberikan keluaran <i>metadata</i> perangkat	Valid

Pada Tabel 6.2 dapat diambil kesimpulan bahwa pengujian mekanisme persiapan kerja perangkat aktuator dan sensor sesuai antara hasil yang diharapkan dengan hasil yang terjadi pada pengujian ini serta dinyatakan 100% berhasil dilakukan pada sistem ini berdasarkan kolom status pada setiap prosedur pengujiannya.

6.1.3 Pengujian Mekanisme Pengenalan Perangkat dan Layanan Baru

6.1.3.1 Tujuan Pengujian

Pengujian mekanisme pengenalan perangkat dan layanan baru akan dilakukan terlebih dahulu untuk perangkat aktuator. Pengujian ini mengacu pada rekayasa kebutuhan dan implementasi yang telah dijelaskan pada bab sebelumnya. Pengujian ini dilakukan untuk memverifikasi terhadap proses implementasi yang sudah sebelumnya agar sesuai dengan perancangannya.

6.1.3.2 Prosedur Pengujian

Prosedur yang dilakukan pada pengujian ini yakni sebagai berikut :

1. Perangkat aktuator mengirimkan pesan pengenalan terhadap perangkat *gateway*.
2. Perangkat *gateway* menerima pesan pengenalan yang dikirim dari perangkat aktuator.
3. Perangkat *gateway* mendaftarkan perangkat aktuator kedalam objek aktuator.
4. Perangkat *gateway* mengirimkan pesan umpan balik terhadap hasil pesan pengenalan yang dikirimkan oleh perangkat aktuator.
5. Perangkat aktuator menerima pesan umpan balik yang dikirim dari perangkat *gateway*.

6.1.3.3 Hasil dan Analisis Pengujian

Pada Gambar 6.2 dijelaskan juga aktuator telah melakukan proses pengiriman *metadata* objek perangkat kepada *gateway* sebagai wujud permintaan untuk bergabung dengan *gateway*. Pesan permintaan tersebut juga dibungkus dalam format teks json. Selain itu, pada Gambar 6.2 juga diterangkan bahwa ketika *gateway* telah mengirimkan umpan balik sukses, maka perangkat aktuator mencetak hasil dari umpan balik tersebut.



```
code — python gateway.py — 80x22
'{"actuator": {"BULB-001": {"category": "BULB", "deviceName": "BULB-001", "service": {"lamp": {"data": "0", "name": "lamp", "unit": "0"}, "level": {"data": "0", "name": "level", "unit": "0"}}, "integration": {"category": ["LDR", "PIR"], "max": 2}, "ackTopic": "kardel/ack/actuator/BULB-001", "deviceType": "actuator", "location": "kamar"}, "sensor": {}, "locations": {"kamar": [{"deviceName": "BULB-001", "deviceType": "actuator", "topic_pervasive": "kardel/pervasive/kamar/actuator/BULB-001/data"}]}}
```

Gambar 6.5 Gateway telah menerima objek perangkat aktuator

Pada Gambar 6.5 dijelaskan bahwa *gateway* telah menerima objek dari perangkat aktuator sehingga perangkat tersebut dapat direlasikan ke berbagai perangkat sensor. Proses ini akan ditampilkan apabila mekanisme pengecekan ketersediaan aktuator pada suatu lokasi serta pengiriman umpan balik kepada perangkat aktuator sebagai pesan konfirmasi atas permintaan yang telah diajukan telah berhasil

Tabel 6.3 Tabel Pengujian mekanisme pengenalan perangkat dan layanan baru

No.	Test Name	Test Case	Expected Result	Result	Status
1.	REQ-PERVASIVE-PPLB-1200- Pengiriman pesan <i>broadcast</i> oleh aktuator	Melakukan pengujian pada fungsi <i>setup Broadcast()</i> untuk melakukan verifikasi pengiriman pesan yang dikirimkan. Sebelum pesan dikirimkan, pesan harus dibungkus didalam format teks json	Perangkat aktuator dapat mengirimkan <i>metadata</i> yang telah diinisialisasi sebelumnya dan dikirimkan ke <i>gateway</i> menggunakan mqtt serta format teks json	Perangkat aktuator telah berhasil mengirimkan pesan berupa informasi <i>metadata</i> menggunakan format json serta ditampilkan pada <i>serial monitor</i>	Valid
2.	REQ-PERVASIVE-PPLB-1201- <i>Gateway</i> menerima pesan <i>broadcast</i>	Melakukan inspeksi pada metode penerimaan pesan pada <i>gateway</i> . Hasil penerimaan akan tampil pada terminal	<i>Gateway</i> dapat menerima pesan yang dikirimkan oleh suatu perangkat.	<i>Gateway</i> telah berhasil menerima pesan <i>metadata</i> yang dikirimkan oleh perangkat aktuator yang selanjutnya akan berlanjut ke proses selanjutnya	Valid
3.	REQ-PERVASIVE-PPLB-1202- Pendaftaran aktuator oleh	Melakukan pengujian pada fungsi <i>device Connect()</i>	<i>Gateway</i> dapat menyimpan <i>metadata</i> perangkat	<i>Gateway</i> telah berhasil untuk menyimp	Valid

No.	Test Name	Test Case	Expected Result	Result	Status
	<i>gateway</i>	untuk melakukan verifikasi apakah <i>metadata</i> objek aktuator telah disimpan	aktuator pada objek aktuator yang ditampilkan di terminal	an <i>metadata</i> objek perangkat yang ditampilkan pada terminal dengan format teks json	
4.	REQ-PERVASIVE-PPLB-1203- Pengiriman umpan balik oleh <i>gateway</i>	Melakukan pengujian pada fungsi <i>client.publish()</i> untuk melakukan verifikasi pengiriman pesan yang dikirimkan. Sebelum pesan dikirimkan, pesan harus dibungkus didalam format teks json	<i>Gateway</i> dapat mengirimkan pesan mengenai status permintaan yang dilakukan oleh perangkat aktuator menggunakan mqtt serta format teks json	<i>Gateway</i> berhasil mengirimkan pesan mengenai status permintaan kepada perangkat aktuator	Valid
5.	REQ-PERVASIVE-PPLB-1204- Aktuator mendapatkan umpan balik	Melakukan inspeksi pada metode penerimaan pesan pada perangkat aktuator. Hasil penerimaan akan tampil pada <i>serial monitor</i>	Perangkat aktuator dapat menerima pesan yang dikirimkan oleh <i>gateway</i> . Hasilnya berupa pesan yang akan tampil di <i>serial monitor</i>	Perangkat aktuator telah sukses mengirimkan umpan balik yang ditampilkan pada <i>serial monitor</i>	Valid

Pada Tabel 6.3 dapat diambil kesimpulan bahwa pengujian mekanisme pengenalan perangkat dan layanan baru sesuai antara hasil yang diharapkan dengan hasil yang terjadi pada pengujian ini serta dinyatakan 100% berhasil dilakukan pada sistem ini berdasarkan kolom status pada setiap prosedur pengujiannya.

6.1.4 Pengujian Mekanisme Pembuatan Relasi antara Sensor dengan Aktuator

6.1.4.1 Tujuan Pengujian

Pengujian selanjutnya yakni mengenai mekanisme pembuatan relasi antara sensor dengan aktuator. Setelah aktuator terhubung dengan *gateway* pada suatu lokasi tertentu, maka selanjutnya perangkat sensor dapat terhubung juga dan akan membentuk relasi pada setiap perangkat yang memiliki lokasi yang sama. Pengujian ini dilakukan untuk memverifikasi terhadap implementasi yang telah dilakukan sebelumnya agar sesuai dengan perancangan.

6.1.4.2 Prosedur Pengujian

Prosedur yang dilakukan pada pengujian ini yakni sebagai berikut :

1. Perangkat sensor mengirimkan pesan pengenalan terhadap perangkat *gateway*.
2. Perangkat *gateway* menerima pesan pengenalan yang dikirim dari perangkat sensor.
3. Perangkat *gateway* melakukan pengecekan aktuator, maksimal integrasi dan pengecekan kategori sensor pada objek aktuator.
4. Perangkat *gateway* mendaftarkan perangkat sensor kedalam objek sensor.
5. Perangkat *gateway* mengirimkan pesan umpan balik terhadap hasil pesan pengenalan yang dikirimkan oleh perangkat sensor.
6. Perangkat sensor menerima pesan umpan balik yang dikirim dari perangkat *gateway*.
7. Perangkat *gateway* mengirimkan pesan update relasi kepada perangkat aktuator.
8. Perangkat aktuator menerima pesan update yang dikirim dari perangkat *gateway*.

6.1.4.3 Hasil dan Analisis Pengujian

Pada Gambar 6.3 maupun Gambar 6.4 juga ditunjukkan bahwa perangkat sensor baik yang menggunakan sensor PIR maupun LDR dapat memberikan pesan terhadap *gateway* untuk melakukan permintaan perijinan untuk terhubung ke suatu perangkat aktuator. Pada gambar tersebut juga dijelaskan bahwa pesan dibungkus kedalam format teks json, sehingga dapat diolah kembali oleh *gateway* untuk melewati beberapa kondisi pada mekanisme selanjutnya. Kemudian setelah semua kondisi sukses, maka perangkat sensor akan mendapatkan umpan balik yang akan ditampilkan pada *serial monitor*.

```

code — python gateway.py — 80x22
{"actuator": {"BULB-001": {"category": "BULB", "deviceName": "BULB-001", "service": {"lamp": {"data": "0", "name": "lamp", "unit": "0"}, "level": {"data": "0", "name": "level", "unit": "0"}}, "integration": {"category": ["LDR", "PIR"], "max": 2}, "ackTopic": "kardel/ack/actuator/BULB-001", "deviceType": "actuator", "location": "kamar"}}, "sensor": {"PIR-001": {"category": "PIR", "deviceName": "PIR-001", "service": {"motion": {"data": "0", "name": "motion", "unit": "0"}}, "ackTopic": "kardel/ack/sensor/PIR-001", "deviceType": "sensor", "location": "kamar"}}, "locations": {"kamar": [{"deviceName": "BULB-001", "deviceType": "actuator", "topic_pervasive": "kardel/pervasive/kamar/actuator/BULB-001/data"}, {"deviceName": "PIR-001", "deviceType": "sensor", "topic_pervasive": "kardel/pervasive/kamar/sensor/PIR-001/data"}]}'
masuk update
masuk update fix

```

Gambar 6.6 Gateway telah menerima objek perangkat sensor

Pada Gambar 6.6 ditampilkan bahwa *gateway* telah menerima objek perangkat sensor yang ingin terhubung sehingga perangkat-perangkat yang telah terdaftar pada objek relasi sudah bisa terhubung satu sama lain. Proses ini akan ditampilkan apabila mekanisme pengecekan ketersediaan aktuator pada lokasi yang sama dengan lokasi perangkat sensor yang melakukan permintaan, mekanisme pengecekan maksimal perangkat sensor yang diijinkan untuk terhubung serta mekanisme pengecekan kategori sensor yang dapat terhubung dengan perangkat aktuator telah berhasil dilakukan, namun jika tidak berhasil tidak akan tampil informasi objek berbagai perangkat pada gambar ini.

Tabel 6.4 Tabel Pengujian mekanisme pembuatan relasi antara sensor dengan aktuator

No	Test Name	Test Case	Expected Result	Result	Status
1.	REQ-PERVASIVE-PRSA-1300-Pengiriman pesan <i>broadcast</i> oleh sensor	Melakukan pengujian pada fungsi <i>setup Broadcast()</i> untuk melakukan verifikasi pengiriman pesan yang dikirimkan. Sebelum pesan dikirimkan, pesan harus dibungkus didalam format teks json	Perangkat sensor dapat mengirimkan <i>metadata</i> yang telah diinisialisasi sebelumnya dan dikirimkan ke <i>gateway</i> menggunakan mqtt serta format teks json	Perangkat sensor telah berhasil mengirimkan pesan berupa informasi <i>metadata</i> menggunakan format json serta menampilkan keluaran "msg sent" pada <i>serial monitor</i>	Valid
2.	REQ-PERVASIVE-PRSA-1301-Pengecekan	Melakukan pengujian pada fungsi <i>device Connect()</i>	Perangkat aktuator dapat melakukan pengecekan	<i>Gateway</i> telah berhasil menerima	Valid

No	Test Name	Test Case	Expected Result	Result	Status
	aktuator	untuk melakukan verifikasi apakah sudah tersedia aktuator pada lokasi tersebut	ketersediaan aktuator dengan mengidentifikasi <i>metadata</i> pada perangkat aktuator tersebut pada suatu lokasi	pesan <i>metadata</i> yang dikirimkan oleh perangkat sensor dan kemudian telah berhasil dilakukan mekanisme pengecekan ketersediaan aktuator dan yang selanjutnya akan berlanjut ke proses selanjutnya	
3.	REQ-PERVASIVE-PRSA-1302-Pengecekan maksimal integrasi	Melakukan pengujian pada fungsi <code>checkRelation()</code> untuk melakukan verifikasi jumlah perangkat sensor yang diijinkan untuk berintegrasi dengan suatu perangkat aktuator	Perangkat aktuator dapat melakukan pengecekan maksimal integrasi dengan mengidentifikasi <i>metadata</i> pada perangkat aktuator tersebut pada suatu lokasi	<i>Gateway</i> telah berhasil melakukan mekanisme pengecekan maksimal integrasi dan yang selanjutnya akan berlanjut pada proses selanjutnya	Valid
4.	REQ-PERVASIVE-PRSA-1303-Pengecekan kategori sensor	Melakukan pengujian pada fungsi <code>checkRelation()</code> untuk melakukan verifikasi kategori perangkat sensor	Perangkat aktuator dapat melakukan pengecekan kategori sensor dengan mengidentifikasi	<i>Gateway</i> telah berhasil melakukan mekanisme pengecekan kategori	

No	Test Name	Test Case	Expected Result	Result	Status
		yang diijinkan untuk berintegrasi dengan suatu perangkat aktuator	si <i>metadata</i> pada perangkat aktuator tersebut pada suatu lokasi	sensor yang selanjutnya akan berlanjut pada proses selanjutnya	
5.	REQ-PERVASIVE-PRSA-1304-Pendaftaran sensor oleh <i>gateway</i>	Melakukan pengujian pada fungsi <i>device Connect ()</i> untuk melakukan verifikasi apakah <i>metadata</i> objek sensor telah disimpan	<i>Gateway</i> dapat menyimpan <i>metadata</i> perangkat sensor pada objek sensor serta objek relasi yang ditampilkan di terminal	<i>Gateway</i> telah berhasil untuk menyimpan <i>metadata</i> objek perangkat yang ditampilkan pada terminal dengan format teks json	Valid
6.	REQ-PERVASIVE-PRSA-1305-Pengiriman umpan balik oleh <i>gateway</i>	Melakukan pengujian pada fungsi <i>client .publish ()</i> untuk melakukan verifikasi pengiriman pesan yang dikirimkan. Sebelum pesan dikirimkan, pesan harus dibungkus didalam format teks json	<i>Gateway</i> dapat mengirimkan pesan mengenai status permintaan yang dilakukan oleh perangkat sensor menggunakan mqtt serta format teks json	<i>Gateway</i> berhasil mengirimkan pesan mengenai status permintaan kepada perangkat sensor	Valid
7.	REQ-PERVASIVE-PRSA-1306-Sensor mendapatkan umpan balik	Melakukan inspeksi pada metode penerimaan pesan pada perangkat sensor. Hasil penerimaan	Perangkat sensor dapat menerima pesan yang dikirimkan oleh <i>gateway</i> . Hasilnya	Perangkat sensor telah sukses mengirimkan umpan balik yang memberikan	Valid

No	Test Name	Test Case	Expected Result	Result	Status
		akan tampil pada <i>serial monitor</i>	berupa pesan yang akan tampil di <i>serial monitor</i>	keluaran "registration done" pada <i>serial monitor</i>	
8	REQ-PERVASIVE-PRSA-1307-Pengiriman pesan <i>update</i> oleh <i>gateway</i>	Melakukan pengujian pada fungsi relational Compability() untuk melakukan verifikasi pengiriman pesan <i>update</i> kepada objek perangkat aktuator pada lokasi tersebut menggunakan mqtt serta format teks json	<i>Gateway</i> dapat mengirimkan pesan mengenai informasi <i>metadata</i> pesan sensor yang telah diijinkan untuk berintegrasi dengan suatu perangkat aktuator.	<i>Gateway</i> berhasil mengirimkan pesan <i>update</i> mengenai perangkat sensor yang akan terhubung kepada perangkat aktuator	Valid
9.	REQ-PERVASIVE-PRSA-1308-Aktuator mendapatkan pesan <i>update</i>	Melakukan inspeksi pada metode penerimaan pesan <i>update</i> pada perangkat aktuator. Hasil penerimaan akan tampil pada <i>serial monitor</i>	Perangkat aktuator dapat menerima pesan <i>update</i> yang dikirimkan oleh <i>gateway</i> . Hasilnya berupa pesan yang akan tampil di <i>serial monitor</i>	Perangkat aktuator telah sukses mendapatkan umpan balik berupa informasi perangkat sensor yang berhasil melakukan mekanisme pembentukan relasi.	Valid

Pada Tabel 6.4 dapat diambil kesimpulan bahwa pengujian mekanisme pembuatan relasi antara sensor dengan aktuator sesuai antara hasil yang diharapkan dengan hasil yang terjadi pada pengujian ini serta dinyatakan 100% berhasil dilakukan pada sistem ini berdasarkan kolom status pada setiap prosedur pengujiannya.

6.1.5 Pengujian Mekanisme Kerja dari Sensor

6.1.5.1 Tujuan Pengujian

Pengujian berikutnya yakni mekanisme kerja dari perangkat sensor baik yang menggunakan sensor PIR maupun LDR. Mekanisme berlaku untuk perangkat sensor yang telah berhasil membuat relasi dengan suatu perangkat aktuator, sehingga data perangkat sensor dapat terkirim ke suatu perangkat aktuator. Pengujian ini dilakukan untuk memverifikasi terhadap implementasi yang telah dilakukan sebelumnya agar sesuai dengan perancangan.

6.1.5.2 Prosedur Pengujian

Prosedur yang dilakukan pada pengujian ini yakni sebagai berikut :

1. Perangkat sensor melakukan pembacaan data sensor.
2. Perangkat sensor melakukan pengiriman data sensor.
3. Perangkat *gateway* menangkap data sensor yang dikirimkan.

6.1.5.3 Hasil dan Analisis Pengujian

Pada Gambar 6.3 dan Gambar 6.4 pada pengujian sebelumnya menunjukkan bahwa kedua perangkat telah berhasil dalam menjalankan mekanisme pembacaan serta pengiriman data sensor yang ditunjukkan pada *serial monitor* dan pesan tersebut juga dibungkus kedalam format teks json.



```
code — python gateway.py — 80x22
{"actuator": {"BULB-001": {"category": "BULB", "deviceName": "BULB-001", "service": {"lamp": {"data": "0", "name": "lamp", "unit": "0"}, "level": {"data": "0", "name": "level", "unit": "0"}}, "integration": {"category": ["LDR", "PIR"], "max": 2}, "ackTopic": "kardel/ack/actuator/BULB-001", "deviceType": "actuator", "location": "kamar"}}, "sensor": {"PTR-001": {"category": "PTR", "deviceName": "PTR-001", "service": {"motion": {"data": "1", "name": "motion", "unit": " "}}, "ackTopic": "kardel/ack/sensor/PIR-001", "deviceType": "sensor", "location": "kamar"}}, "locations": {"kamar": [{"deviceName": "BULB-001", "deviceType": "actuator", "topic_pervasive": "kardel/pervasive/kamar/actuator/BULB-001/data"}, {"deviceName": "PIR-001", "deviceType": "sensor", "topic_pervasive": "kardel/pervasive/kamar/sensor/PIR-001/data"}]}
masuk update
masuk update fix
```

Gambar 6.7 Gateway telah menerima data sensor

Pada gambar 6.7 menunjukkan bahwa *gateway* berhasil menangkap data sensor dan kemudian menyimpan data tersebut kedalam objek perangkat sensor tersebut, sehingga *gateway* pun dapat mengetahui data terakhir dari sensor tersebut.

Tabel 6.5 Tabel Pengujian mekanisme kerja dari sensor

No.	Test Name	Test Case	Expected Result	Result	Status
1.	REQ-PERVASIVE-KS-1400-Pembacaan data oleh	Melakukan inspeksi pada metode pembacaan data sensor	Perangkat sensor dapat membaca sensor PIR pada pin	Perangkat sensor telah berhasil membaca data sensor PIR	Valid

No.	Test Name	Test Case	Expected Result	Result	Status
	sensor PIR	PIR pada perangkat sensor. Hasil pembacaan sensor akan ditampilkan pada <i>serial monitor</i>	yang telah ditentukan pada <i>metadata</i> . Hasilnya berupa nilai 0 untuk terdapat cahaya serta nilai 1 untuk tidak terdapat cahaya		
2.	REQ-PERVASIVE-KS-1401- Pembacaan data oleh sensor LDR	Melakukan inspeksi pada metode pembacaan data sensor LDR pada perangkat sensor. Hasil pembacaan sensor akan ditampilkan pada <i>serial monitor</i>	Perangkat sensor dapat membaca sensor LDR pada pin yang telah ditentukan pada <i>metadata</i> . Hasilnya berupa nilai 0 untuk tidak terdapat gerakan serta nilai 1 untuk terdapat gerakan	Perangkat sensor telah berhasil membaca data sensor LDR	Valid
3.	REQ-PERVASIVE-KS-1402- Pengiriman data oleh sensor PIR	Melakukan pengujian pada fungsi <code>publish</code> <code>Data()</code> untuk melakukan verifikasi pengiriman data sensor PIR yang dikirimkan.	Perangkat sensor dapat mengirimkan data sensor PIR yang telah diakusisi sebelumnya dan dikirimkan ke perangkat aktuator	Perangkat sensor telah berhasil mengirimkan data sensor PIR dan menampilkan informasi pesannya pada <i>serial monitor</i>	Valid

No.	Test Name	Test Case	Expected Result	Result	Status
		Sebelum pesan dikirimkan, pesan harus dibungkus didalam format teks json	didalam relasinya		
4.	REQ-PERVASIVE-KS-1403-Pengiriman data oleh sensor LDR	Melakukan pengujian pada fungsi <code>publish Data()</code> untuk melakukan verifikasi pengiriman data sensor LDR yang dikirimkan. Sebelum pesan dikirimkan, pesan harus dibungkus didalam format teks json	Perangkat sensor dapat mengirimkan data sensor LDR yang telah diakusisi sebelumnya dan dikirimkan ke perangkat aktuator didalam relasinya	Perangkat sensor telah berhasil mengirimkan data sensor LDR dan menampilkan informasi pesannya pada <i>serial monitor</i>	Valid
5.	REQ-PERVASIVE-KS-1404-Gateway menangkap data setiap sensor	Melakukan inspeksi pada metode penerimaan data sensor / aktuator pada <i>gateway</i> . Hasil penerimaan akan tampil pada terminal	<i>Gateway</i> dapat menerima data sensor / aktuator yang dikirimkan oleh suatu perangkat. Hasilnya berupa informasi pengirim serta pesan yang akan	<i>Gateway</i> telah berhasil menerima data sensor dan menyimpannya kedalam objek perangkat tersebut untuk mengetahui data terakhir dari perangkat sensor tersebut	Valid

No.	Test Name	Test Case	Expected Result	Result	Status
			tampil di terminal		

Pada Tabel 6.5 dapat diambil kesimpulan bahwa pengujian mekanisme kerja dari sensor sesuai antara hasil yang diharapkan dengan hasil yang terjadi pada pengujian ini serta dinyatakan 100% berhasil dilakukan pada sistem ini berdasarkan kolom status pada setiap prosedur pengujiannya.

6.1.6 Pengujian Mekanisme Kerja dari Aktuator

6.1.6.1 Tujuan Pengujian

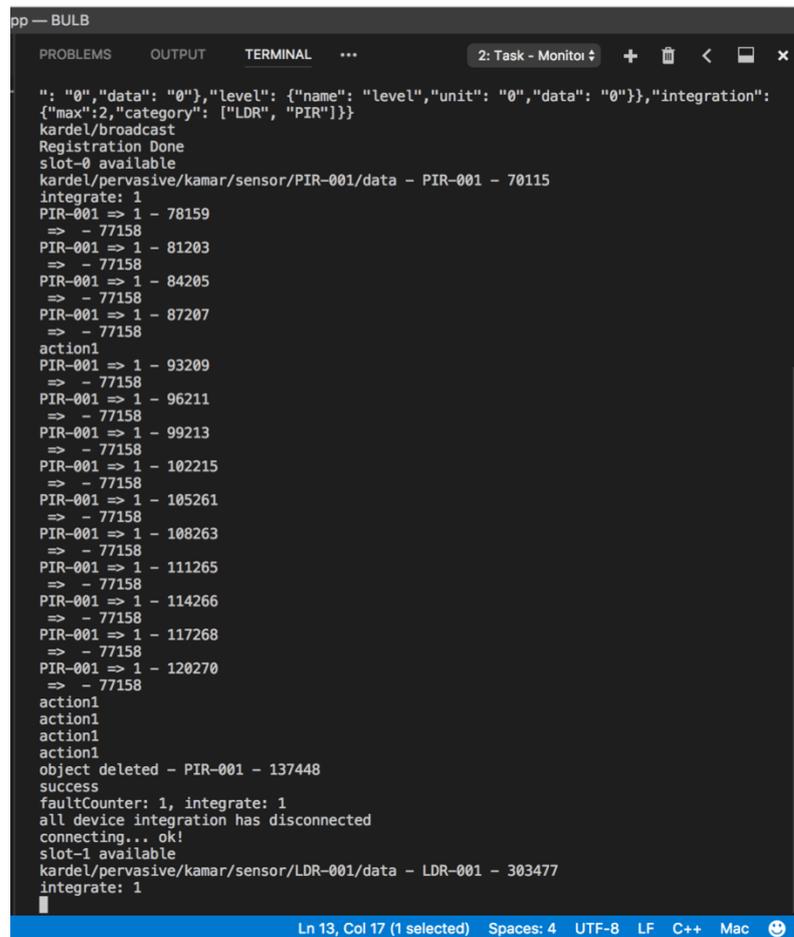
Pengujian mekanisme kerja dari aktuator merupakan pengujian selanjutnya setelah data sensor dari berbagai perangkat sensor telah berhasil dikirimkan kepada perangkat *gateway* yang berada pada relasi yang sama. Pengujian ini dilakukan untuk memverifikasi terhadap implementasi yang telah dilakukan sebelumnya agar sesuai dengan perancangan yang telah dibuat.

6.1.6.2 Prosedur Pengujian

Prosedur yang dilakukan pada pengujian ini yakni sebagai berikut :

1. Perangkat aktuator menerima data yang dikirimkan oleh perangkat sensor.
2. Perangkat aktuator melakukan aksi untuk mematikan atau menyalakan lampu.
3. Perangkat aktuator mengirimkan data status terakhir dari lampu kepada perangkat *gateway*.
4. Perangkat *gateway* menangkap data aktuator.

6.1.6.3 Hasil dan Analisis Pengujian



```
pp — BULB
PROBLEMS OUTPUT TERMINAL ... 2: Task - Monitor + - < □ ×

": {"0", "data": "0"}, "level": {"name": "level", "unit": "0", "data": "0"}, "integration":
{"max": 2, "category": ["LDR", "PIR"]}}
kardel/broadcast
Registration Done
slot-0 available
kardel/pervasive/kamar/sensor/PIR-001/data - PIR-001 - 70115
integrate: 1
PIR-001 => 1 - 78159
=> - 77158
PIR-001 => 1 - 81203
=> - 77158
PIR-001 => 1 - 84205
=> - 77158
PIR-001 => 1 - 87207
=> - 77158
action1
PIR-001 => 1 - 93209
=> - 77158
PIR-001 => 1 - 96211
=> - 77158
PIR-001 => 1 - 99213
=> - 77158
PIR-001 => 1 - 102215
=> - 77158
PIR-001 => 1 - 105261
=> - 77158
PIR-001 => 1 - 108263
=> - 77158
PIR-001 => 1 - 111265
=> - 77158
PIR-001 => 1 - 114266
=> - 77158
PIR-001 => 1 - 117268
=> - 77158
PIR-001 => 1 - 120270
=> - 77158
action1
action1
action1
action1
object deleted - PIR-001 - 137448
success
faultCounter: 1, integrate: 1
all device integration has disconnected
connecting... ok!
slot-1 available
kardel/pervasive/kamar/sensor/LDR-001/data - LDR-001 - 303477
integrate: 1
```

Gambar 6.8 Aktuator telah menerima data sensor PIR

Pada Gambar 6.8 menerangkan bahwa perangkat aktuator telah sukses menerima data dari sensor PIR serta menggunakan identitas *timer* untuk mengetahui data terakhir berada di waktu sekian. Selanjutnya perangkat aktuator akan memberikan keputusan terhadap objek lampu untuk menyalakan lampu berdasarkan data dari sensor yang diterima.

```
pp — BULB
PROBLEMS OUTPUT TERMINAL ... 2: Task - Monitor
> Executing task: platformio device monitor <

--- Miniterm on /dev/cu.wchusbserial1410 9600,8,N,1 ---
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---
Registration Done
slot-0 available
kardel/pervasive/kamar/sensor/LDR-001/data - LDR-001 - 51551
integrate: 1
LDR-001 => 0 - 56597
=> - 0
action0
LDR-001 => 1 - 62600
=> - 61598
LDR-001 => 1 - 65601
=> - 61598
LDR-001 => 1 - 68645
=> - 61598
LDR-001 => 0 - 71647
=> - 61598
action0
object deleted - LDR-001 - 122326
success
faultCounter: 1, integrate: 1
all device integration has disconnected
```

Gambar 6.9 Aktuator telah menerima data sensor LDR

Pada Gambar 6.9 menerangkan bahwa perangkat aktuator telah sukses menerima data dari sensor LDR serta menggunakan identitas *timer* untuk mengetahui data terakhir berada di waktu sekian. Selanjutnya perangkat aktuator akan memberikan keputusan terhadap objek lampu untuk mematikan lampu berdasarkan data dari sensor yang diterima.



Gambar 6.10 Aktuator telah mematikan lampu

Pada Gambar 6.10 dijelaskan aktuator telah mematikan lampu berdasarkan data yang telah di dapat pada Gambar 6.8. Data yang didapatkan ialah nilai 0 maka dengan itu aktuator akan memberikan keputusan untuk mematikan lampu.



Gambar 6.11 Aktuator telah menyalakan lampu

Pada Gambar 6.11 dijelaskan aktuator telah menyalakan lampu berdasarkan data yang telah didapat pada Gambar 6.9. Data yang didapatkan ialah nilai 1, sehingga aktuator akan memberikan keputusan untuk menyalakan lampu.

```
code — python gateway.py — 80x24
{"actuator": {"BULB-001": {"category": "BULB", "deviceName": "BULB-001", "service": {"lamp": {"data": "1", "name": "lamp", "unit": "0"}, "level": {"data": 0, "name": "level", "unit": "%"}, "integration": {"category": ["LDR", "PIR"], "max": 2}, "ackTopic": "kardel/ack/actuator/BULB-001", "deviceType": "actuator", "location": "kamar"}}, "sensor": {"PIR-001": {"category": "PIR", "deviceName": "PIR-001", "service": {"motion": {"data": "1", "name": "motion", "unit": ""}, "ackTopic": "kardel/ack/sensor/PIR-001", "deviceType": "sensor", "location": "kamar"}, "locations": {"kamar": [{"deviceName": "BULB-001", "deviceType": "actuator", "topic_pervasive": "kardel/pervasive/kamar/actuator/BULB-001/data"}, {"deviceName": "PIR-001", "deviceType": "sensor", "topic_pervasive": "kardel/pervasive/kamar/sensor/PIR-001/data"}]}}
```

Gambar 6.12 Gateway menangkap data aktuator

Pada Gambar 6.12 dijelaskan bahwa *gateway* berhasil menangkap data aktuator berupa kondisi terakhir pada lampu serta kemudian menyimpan data tersebut kedalam objek perangkat aktuator, sehingga *gateway* pun dapat mengetahui data terakhir dari sensor tersebut

Tabel 6.6 Tabel Pengujian mekanisme kerja dari aktuator

No.	Test Name	Test Case	Expected Result	Result	Status
1.	REQ-PERVASIVE-KA-1500-Aktuator Menerima data dari sensor PIR	Melakukan inspeksi pada metode penerimaan data sensor PIR pada perangkat aktuator. Hasil penerimaan akan tampil pada <i>serial</i>	Perangkat aktuator dapat menerima data sensor PIR yang dikirimkan oleh perangkat sensor PIR. Hasilnya	Perangkat aktuator telah berhasil mendapatkan sensor PIR dan menampilkan data tersebut ke <i>serial monitor</i> . Serta menambah	Valid

No.	Test Name	Test Case	Expected Result	Result	Status
		<i>monitor</i>	berupa pesan yang akan tampil di <i>serial monitor</i>	atribut waktu	
2.	REQ-PERVASIVE-KA-1501-Aktuator menerima data dari sensor LDR	Melakukan inspeksi pada metode penerimaan data sensor LDR pada perangkat aktuator. Hasil penerimaan akan tampil pada <i>serial monitor</i>	Perangkat aktuator dapat menerima data sensor LDR yang dikirimkan oleh perangkat sensor PIR. Hasilnya berupa pesan yang akan tampil di <i>serial monitor</i>	Perangkat aktuator telah berhasil mendapatkan sensor LDR dan menampilkan data tersebut ke <i>serial monitor</i> . Serta menambah atribut waktu	Valid
3.	REQ-PERVASIVE-KA-1502-Aktuator menyalakan lampu	Melakukan pengujian pada fungsi <code>action</code> <code>Actuator()</code> untuk melakukan verifikasi pemberian keputusan terhadap aktuator tersebut berdasarkan data sensor yang telah didapatkan	Perangkat aktuator dapat menyalakan lampu dengan kondisi jika data sensor hanya sensor PIR ataupun LDR maka harus bernilai 1 dan jika kedua data sensor tersedia maka kedua data harus memiliki nilai yang berbeda	Perangkat aktuator telah berhasil memberikan suatu keputusan berdasarkan data sensor untuk menyalakan lampu	Valid

No.	Test Name	Test Case	Expected Result	Result	Status
4.	REQ-PERVASIVE-KA-1503-Aktuator Mematikan lampur	Melakukan pengujian pada fungsi <code>action</code> <code>Actuator()</code> untuk melakukan verifikasi pemberian keputusan terhadap aktuator tersebut berdasarkan data sensor yang telah didapatkan	Perangkat aktuator dapat mematikan lampu dengan kondisi jika data sensor hanya sensor PIR ataupun LDR maka harus bernilai 0 dan jika kedua data sensor tersedia maka kedua data harus memiliki nilai yang sama	Perangkat aktuator telah berhasil memberikan suatu keputusan berdasarkan data sensor untuk mematikan lampu	Valid
5.	REQ-PERVASIVE-KA-1504-Pengiriman data oleh aktuator	Melakukan pengujian pada fungsi <code>publish</code> <code>Data()</code> untuk melakukan verifikasi pengiriman data aktuator berupa status terakhir dari lampu. Sebelum pesan dikirimkan, pesan harus dibungkus didalam format teks <code>json</code>	Perangkat aktuator dapat mengirimkan data kondisi lampu yang terakhir dan dikirimkan ke <i>gateway</i>	Perangkat aktuator telah berhasil mengirimkan data dari kondisi terakhir objek lampu dan menampilkan informasi pesannya pada <i>serial monitor</i>	Valid

No.	Test Name	Test Case	Expected Result	Result	Status
6.	REQ-PERVASIVE-KA-1505- <i>Gateway</i> menangkap data aktuator	Melakukan inspeksi pada metode penerimaan data sensor / aktuator pada <i>gateway</i> . Hasil penerimaan akan tampil pada terminal	<i>Gateway</i> dapat menerima data sensor / aktuator yang dikirimkan oleh suatu perangkat. Hasilnya berupa informasi pengirim serta pesan yang akan tampil di terminal	<i>Gateway</i> telah berhasil menerima data aktuator dan menyimpannya kedalam objek perangkat tersebut untuk mengetahui data terakhir dari perangkat aktuator	Valid

Pada Tabel 6.6 dapat diambil kesimpulan bahwa pengujian mekanisme kerja dari aktuator sesuai antara hasil yang diharapkan dengan hasil yang terjadi pada pengujian ini serta dinyatakan 100% berhasil dilakukan pada sistem ini berdasarkan kolom status pada setiap prosedur pengujiannya.

6.1.7 Pengujian Mekanisme Penghapusan Relasi Sensor

6.1.7.1 Tujuan Pengujian

Pengujian terakhir pada pengujian kebutuhan fungsional yakni mengenai mekanisme penghapusan relasi sensor. Mekanisme ini akan terjadi apabila perangkat sensor tidak mengirimkan data sensornya pada suatu rentan waktu tertentu maka apabila dapat dipastikan perangkat sensor tersebut mati ataupun memiliki kendala teknis maka perangkat tersebut akan dihapus didalam relasi tersebut. Pengujian ini dilakukan untuk memverifikasi terhadap implementasi yang telah dilakukan sebelumnya agar sesuai dengan perancangan.

6.1.7.2 Prosedur Pengujian

Prosedur yang dilakukan pada pengujian ini yakni sebagai berikut :

1. Perangkat aktuator mengecek ketersediaan perangkat sensor dengan melihat informasi waktu menggunakan milis pada setiap data sensornya.
2. Perangkat aktuator menghapus objek sensor.
3. Perangkat aktuator mengirimkan pesan update kepada perangkat gateway yang berisi id sensor yang ingin dihapus objeknya pada suatu relasi.

- Perangkat *gateway* menghapus objek sensor berdasarkan data yang dikirimkan dari perangkat aktuator.

6.1.7.3 Hasil dan Analisis Pengujian

Pada Gambar 6.8 maupun Gambar 6.9 telah berhasil menambahkan atribut waktu yang akan dijadikan acuan untuk mengambil tindakan penghapusan relasi apabila melanggar dari acuan tersebut. Selain itu pada gambar tersebut, perangkat aktuator telah berhasil untuk menghilangkan 2 perangkat sensor yang telah mati. Sehingga perangkat aktuator tidak perlu lagi berlangganan pada perangkat sensor tersebut.

```
code — python gateway.py — 80x21
{"actuator": {"BULB-001": {"category": "BULB", "deviceName": "BULB-001", "service": {"lamp": {"data": "0", "name": "lamp", "unit": "0"}, "level": {"data": "0", "name": "level", "unit": "0"}}, "integration": {"category": ["LDR", "PIR"], "max": 2}, "ackTopic": "kardel/ack/actuator/BULB-001", "deviceType": "actuator", "location": "kamar"}}, "sensor": {}}, "locations": {"kamar": [{"deviceName": "BULB-001", "deviceType": "actuator", "topic_pervasive": "kardel/pervasive/kamar/actuator/BULB-001/data"}]}}
```

Gambar 6.13 Gateway menghapus perangkat dari objek relasi

Pada Gambar 6.13 juga dijelaskan bahwa *gateway* telah berhasil mendapatkan informasi yang dikirimkan oleh aktuator untuk melakukan penghapusan suatu perangkat didalam relasi serta telah berhasil menghapus perangkat baik didalam objek relasi maupun pada objek perangkat

Tabel 6.7 Tabel Pengujian mekanisme penghapusan relasi sensor

No.	Test Name	Test Case	Expected Result	Result	Status
1.	REQ-PERVASIVE-PRS-1600-Aktuator mengecek data dari sensor	Melakukan pengujian pada fungsi <code>keepAlive</code> untuk melakukan verifikasi apakah data sensor dikirimkan datanya selama rentang waktu yang telah diatur	Perangkat aktuator dapat mendeteksi ketersediaan data sensor dalam suatu rentang waktu.	Perangkat aktuator berhasil memberikan atribut waktu pada data sensor serta mengecek waktu tersebut dengan waktu saat ini.	Valid
2.	REQ-	Melakukan	Perangkat	Perangka	Valid

No.	Test Name	Test Case	Expected Result	Result	Status
	PERVASIVE-PRS-1601-Aktuator menghapus objek sensor	pengujian pada fungsi <i>remove</i> <i>Integration()</i> untuk melakukan verifikasi pemberian tindak lanjutnya yakni dengan menghapus objek sensor	aktuator dapat menghapus objek sensor ketika perangkat aktuator dapat mendeteksi tidak adanya data sensor yang masuk	t aktuator berhasil untuk menghapus perangkat sensor tersebut didalam objek relasi	
3.	REQ-PERVASIVE-PRS-1602-Aktuator mengirimkan pesan ke <i>gateway</i>	Melakukan pengujian pada fungsi <i>update</i> <i>Integration Gateway()</i> untuk melakukan verifikasi pengiriman pesan ke <i>gateway</i> dengan memberikan informasi id perangkat yang telah dihapus relasinya	Perangkat aktuator dapat mengirimkan informasi perangkat sensor yang telah dihapus dan dikirimkan ke <i>gateway</i>	Perangkat aktuator berhasil mengirimkan informasi perangkat sensor yang tidak lagi mengirimkan data sensor secara berkala	Valid
4.	REQ-PERVASIVE-PRS-1603- <i>Gateway</i> menghapus objek sensor	Melakukan pengujian pada fungsi <i>remove</i> <i>Integration()</i> untuk melakukan verifikasi pemberian tindak lanjutnya yakni dengan menghapus objek sensor	<i>Gateway</i> dapat menghapus objek sensor berdasarkan data yang telah dikirimkan oleh perangkat aktuator	<i>Gateway</i> berhasil menghapus perangkat sensor tersebut didalam objek perangkat serta relasi	Valid

Pada Tabel 6.7 dapat diambil kesimpulan bahwa pengujian mekanisme penghapusan relasi sensor sesuai antara hasil yang diharapkan dengan hasil yang terjadi pada pengujian ini serta dinyatakan 100% berhasil dilakukan pada sistem ini berdasarkan kolom status pada setiap prosedur pengujianya.

6.2 Pengujian Kebutuhan Non-Fungsional

6.2.1.1 Tujuan Pengujian

Pengujian untuk kebutuhan non-fungsional dilakukan menyesuaikan kebutuhan non-fungsional pada rekayasa kebutuhan yang meliputi kebutuhan keamanan. Pengujian kebutuhan keamanan dilakukan sebagai tindakan preventif dari pihak-pihak yang tidak bertanggung jawab serta memberikan proteksi lebih sehingga data yang dikirimkan menjadi lebih aman.

6.2.1.2 Prosedur Pengujian

Prosedur yang dilakukan pada pengujian ini yakni sebagai berikut :

1. Perangkat aktuator dan sensor menggunakan *password* tertentu untuk melakukan koneksi ke jaringan *wifi*.
2. Perangkat *gateway*, aktuator dan sensor menggunakan *username* dan *password* tertentu untuk melakukan koneksi ke *broker*.

6.2.1.3 Hasil dan Analisis Pengujian

Pada Gambar 6.14 dari segi komunikasi jaringannya menggunakan media *access point* telah melakukan proteksi dengan menggunakan password jika ingin bergabung ke ssid tersebut.



Gambar 6.14 Access point telah terproteksi

Pada Gambar 6.1 dijelaskan *gateway* telah siap untuk menerima berbagai permintaan dari suatu perangkat, namun sebelumnya *gateway* perlu melakukan autentikasi kepada *broker* dengan menggunakan *username* serta *password* untuk bergabung kepada *broker* tersebut.

Tabel 6.8 Tabel Pengujian kebutuhan keamanan

No.	Test Name	Test Case	Expected Result	Result	Status
1.	REQ-PERVASIVE-NF-2000-Password <i>wifi</i>	Menunjukkan bahwa <i>access point</i> terproteksi menggunakan	<i>Access point</i> harus terproteksi dengan password.	<i>Access point</i> telah terproteksi menggunakan password,	Valid

No.	Test Name	Test Case	Expected Result	Result	Status
		password untuk memberikan proteksi lebih di segi keamanan.		sehingga komunikasi secara keseluruhan sistem dapat terjamin keamanan informasinya	
2.	REQ-PERVASIVE-NF-2001-Password broker	Menunjukkan bahwa <i>broker</i> pada perangkat <i>gateway</i> terproteksi menggunakan username dan password untuk memberikan proteksi lebih di segi keamanan.	<i>Broker</i> pada <i>gateway</i> harus terproteksi dengan username dan password	<i>Broker</i> telah terproteksi menggunakan username serta password, sehingga pertukaran data antar perangkat dapat terjamin keamanannya	Valid

Pada Tabel 6.8 dapat diambil kesimpulan bahwa pengujian kebutuhan non-fungsional sesuai antara hasil yang diharapkan dengan hasil yang terjadi pada pengujian ini serta dinyatakan 100% berhasil dilakukan pada sistem ini berdasarkan kolom status pada setiap prosedur pengujiannya.