

BAB VI PENGUJIAN DAN ANALISIS

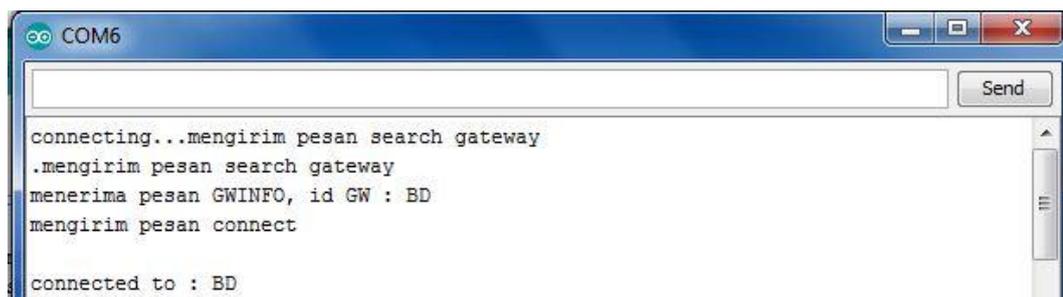
Pada bab ini membahas pengujian dari sistem yang telah diimplementasikan untuk mengetahui sistem berjalan baik dengan kebutuhan-kebutuhan yang telah dirancang pada bab sebelumnya.

6.1. Pengujian Kebutuhan Fungsional

Pengujian terkait dengan kebutuhan fungsional yang dilakukan pada perangkat keras dan perangkat lunak yang menyesuaikan dengan kebutuhan fungsional pada bab rekayasa kebutuhan.

6.1.1. Pengujian Mekanisme Kerja Awal Komunikasi *Client* dan *Gateway*

Pengujian mekanisme kerja awal komunikasi *client* dan *gateway* mengacu dengan bab rekayasa kebutuhan dan implementasi. Pengujian ini untuk mengetahui apakah implementasi yang telah dilakukan sesuai dengan perancangan. Proses awal yang berjalan terlebih dahulu adalah melakukan proses *connect* antara *client* dan *gateway*.



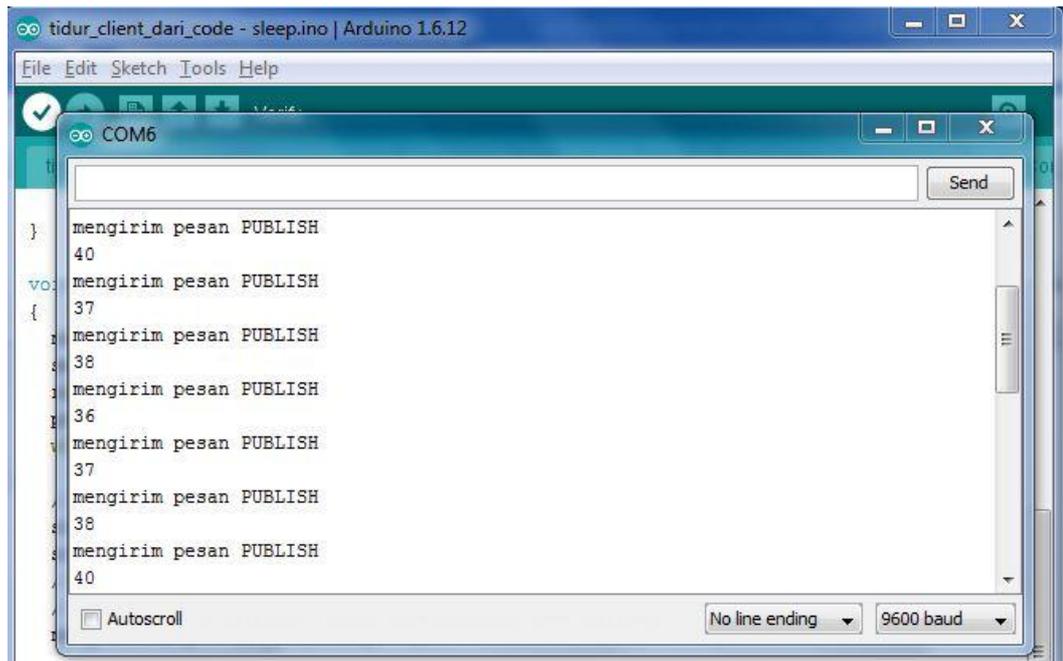
Gambar 6.1 Perangkat *client* mencari *gateway* dan menghubungkan

Pada Gambar 6.1 dijelaskan bahwa perangkat *client* mengirim dan menerima pesan yang digunakan untuk mengkoneksikan kepada *gateway*. Perangkat *client* berhasil menerima *id gateway* yaitu BD yang selanjutnya *client* dan berhasil terkoneksi kepada *gateway*



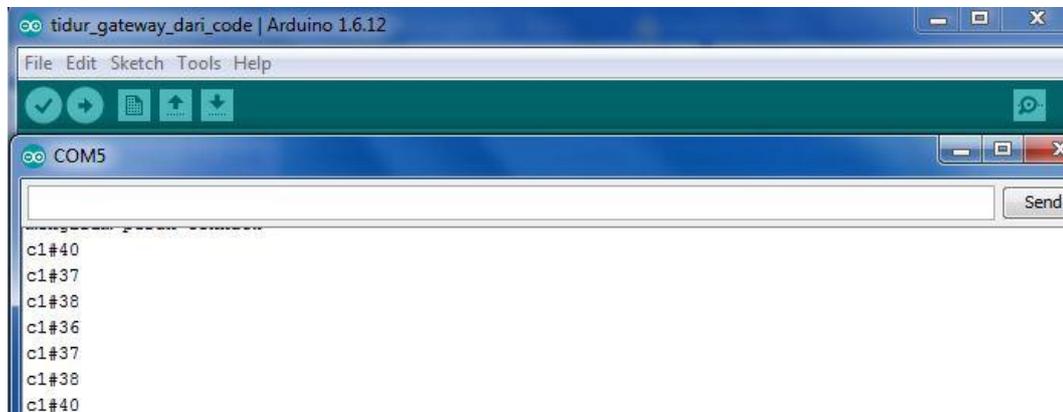
Gambar 6.2 Pesan yang dikirim *gateway* kepada *client*

Pada Gambar 6.2 merupakan hasil dari pengiriman pesan dari *gateway* menuju *client* dimana *gateway* mengirim *gateway info* yaitu *id gateway* dan mengirimkan *connect ACK*.



Gambar 6.3 Perangkat *client* mengirim data sensor kepada *gateway*

Pada Gambar 6.3 perangkat *client* telah mengirim data sensor kepada perangkat *gateway* yang menandakan bahwa konfigurasi awal telah berhasil terkoneksi dan dapat berkomunikasi.



Gambar 6.4 Perangkat *gateway* menerima data sensor yang dikirim *client*

Gambar 6.4 Perangkat *gateway* berhasil menerima data sensor yang berasal dari perangkat *client* dengan menampilkan *id client* yang telah mengirim data sensor tersebut.

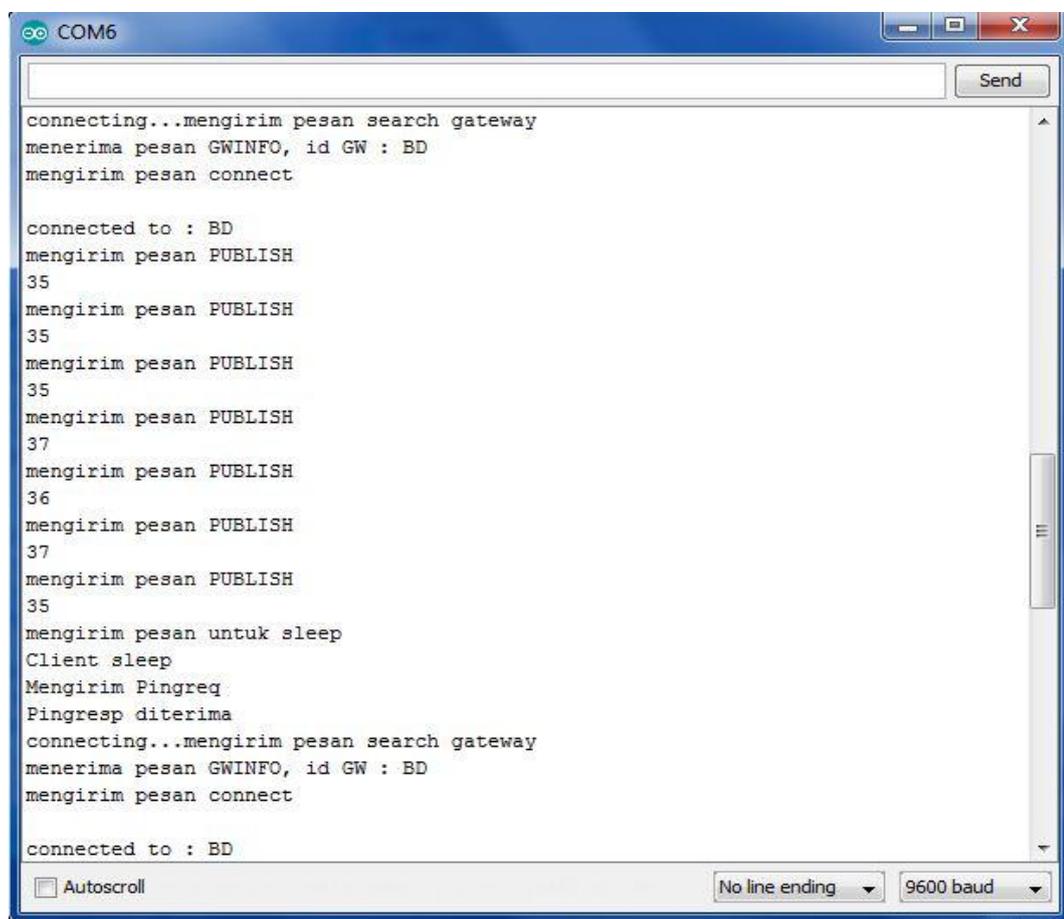
Tabel 6.1 Tabel Pengujian Mekanisme Kerja Awal Komunikasi *Client* dan *Gateway*

No.	Test Name	Test Case	Expected Result	Result	Status
1.	REQ-SLEEPING-AKCG-3000 - <i>Client</i> mencari <i>gateway</i>	Pengujian pada <i>client</i> fungsi <i>sendGWMsg()</i> untuk mencari <i>gateway</i>	Perangkat <i>gateway</i> mendapat pesan dari <i>client</i>	Perangkat <i>gateway</i> berhasil menerima pesan pencarian <i>gateway</i> dari <i>client</i>	Valid
2.	REQ-SLEEPING-AKCG-3001 - <i>Gateway</i> mengirim <i>id</i> kepada <i>client</i>	Pengujian pada <i>gateway</i> fungsi <i>sendGWinfo()</i> untuk mengirim <i>id</i> miliknya	Perangkat <i>client</i> mendapat pesan berisi <i>id gateway</i>	Perangkat <i>client</i> berhasil menerima pesan berisi <i>id gateway</i>	Valid
3.	REQ-SLEEPING-AKCG-3002 - <i>Client</i> mengirim pesan <i>connect</i>	Pengujian pada <i>client</i> fungsi <i>connect()</i> untuk mengirim pesan <i>connect</i> ke <i>gateway</i>	Perangkat <i>gateway</i> mendapat pesan <i>connect</i> dari <i>client</i>	Perangkat <i>gateway</i> berhasil menerima pesan <i>connect</i> dari <i>client</i>	Valid
4.	REQ-SLEEPING-AKCG-3003 - <i>Gateway</i> mengirim ACK terkait <i>connect</i>	Pengujian pada <i>gateway</i> pada fungsi <i>sendConnACK()</i> untuk menyetujui pesan <i>connect client</i>	Perangkat <i>client</i> mendapat pesan <i>ack gateway</i> dan langsung terkoneksi	Perangkat <i>client</i> berhasil mendapat pesan <i>ack gateway</i> dan melakukan koneksi	Valid
5.	REQ-SLEEPING-AKCG-3004 - <i>Client</i> melakukan pengiriman	Pengujian pada <i>client</i> pada fungsi <i>sendData()</i> untuk mengirim data sensor	Perangkat <i>gateway</i> menerima data sensor dari <i>client</i>	Perangkat <i>gateway</i> berhasil menerima data sensor	Valid

No.	Test Name	Test Case	Expected Result	Result	Status
	data ke gateway			beserta <i>id</i> dari <i>client</i>	

6.1.2. Pengujian Mekanisme Kerja Fitur *Sleep* dan *Awake*

Pengujian mekanisme kerja fitur *sleep* dan *awake* mengacu dengan bab rekayasa kebutuhan dan implementasi yang telah dijelaskan. Pengujian dilakukan untuk mengetahui apakah implementasi yang dilakukan sudah sesuai dengan perancangan. Pada mekanisme terdapat proses *client* menjalankan fitur *sleep* dan *awake*.



Gambar 6.5 Perangkat *client* melakukan *sleep* dan *awake*

Gambar 6.5 merupakan perangkat *client* melakukan pengiriman data sensor dan juga melakukan kondisi *sleep* dan *awake*. Perangkat *client* berhasil mengirim pesan *sleep* kepada *gateway* dan menerima ACK *sleep* tersebut serta melakukan fitur *sleep* dan *awake*. Selain itu perangkat *client* mengirim pesan pingreq ketika *awake* untuk memberitahu bahwa *client* telah aktif.

```

COM7
mengirim pesan gateway info
1 mengirim pesan connack
1 2 3 c1#36
1 c1#36
1 c1#37
1 c1#37
1 c1#37
1 c1#37
1 2 c1#35
1 Client sleep selama: 40
Client diijinkan sleep
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 Gateway menerima pingreq
mengirim pesan gateway info
1 mengirim pesan connack

```

Gambar 6.6 Perangkat *gateway* mengecek durasi *sleep*

Gambar 6.6 menjelaskan bahwa perangkat *gateway* berhasil menerima durasi *sleep client* yang diawasi durasi *sleep* tersebut untuk mengetahui apakah *client* akan *lost*. Perangkat juga berhasil menerima pesan pingreq dan mengirim pesan pingresp untuk mempersilahkan perangkat *client* terkoneksi kembali.

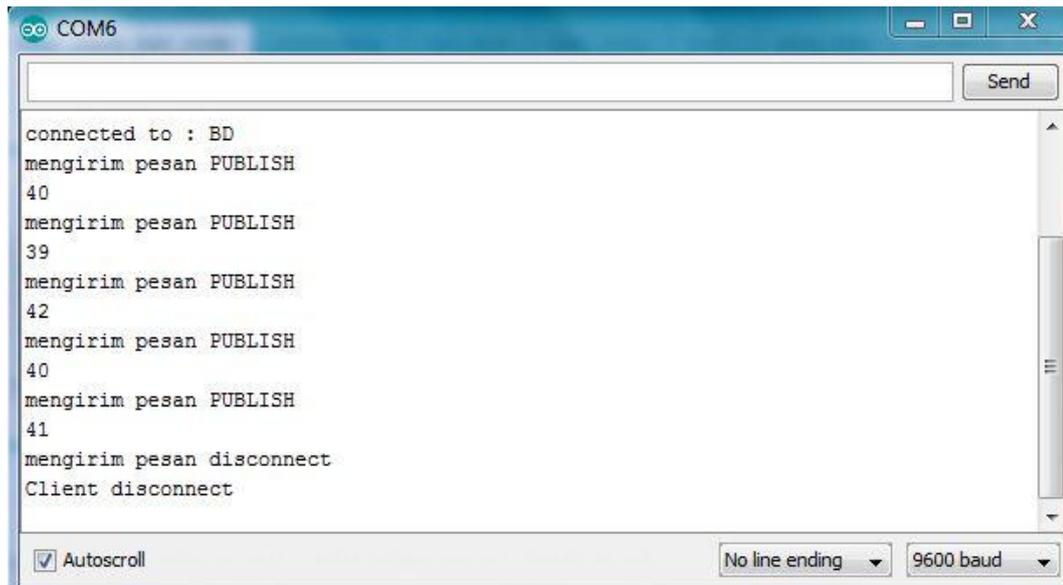
Tabel 6.2 Tabel Pengujian Mekanisme Kerja Fitur *Sleep* dan *Awake*

No.	Test Name	Test Case	Expected Result	Result	Status
1.	REQ-SLEEPING-KFSA-3100 –Client mengirim pesan sleep	Pengujian pada <i>client</i> fungsi <i>sleepMsg()</i> untuk mengirim pesan <i>sleep</i> kepada <i>gateway</i>	Perangkat <i>gateway</i> mendapat pesan dari <i>client</i> beserta durasi <i>sleep</i>	Perangkat <i>gateway</i> berhasil menerima pesan <i>sleep</i> dari <i>client</i> yang berisi durasi <i>sleep</i>	Valid
2.	REQ-SLEEPING-KFSA-3101 –Gateway mengirim <i>disconnect</i> (ack)	Pengujian pada <i>gateway</i> fungsi <i>sleepMsgACK()</i> untuk mengirim <i>id</i> miliknya	Perangkat <i>client</i> mendapat pesan persetujuan <i>sleep</i>	Perangkat <i>client</i> berhasil menerima pesan berisi ack tentang <i>sleep client</i>	Valid

No.	Test Name	Test Case	Expected Result	Result	Status
3.	REQ-SLEEPING-KFSA-3102 -Client melakukan sleep	Pengujian pada <i>client</i> fungsi <code>do_sleep()</code> untuk melakukan proses <i>sleep</i>	Perangkat <i>client sleep</i> dan <i>count down</i> durasi pada perangkat <i>gateway</i> tidak lebih	Perangkat <i>client</i> berhasil <i>sleep</i> dan durasi <i>sleep</i> tidak lebih	Valid
4.	REQ-SLEEPING-KFSA-3103 -Client <i>awake</i> dan mengirim pesan <code>pingreq</code>	Pengujian pada <i>client</i> fungsi <code>wakeup()</code> untuk mengaktifkan kembali perangkat <i>client</i> dan mengirim pesan <code>pingreq()</code>	Perangkat <i>client</i> kembali aktif dan perangkat <i>gateway</i> menerima pesan <code>pingreq</code>	Perangkat <i>client</i> berhasil aktif dan perangkat <i>gateway</i> menerima pesan <code>pingreq</code>	Valid
5.	REQ-SLEEPING-KFSA-3104 -Gateway mengirim <code>pingresp</code>	Pengujian pada <i>gateway</i> pada fungsi <code>pingresp()</code> menanggapi <code>pingreq</code> dari <i>client</i>	Perangkat <i>client</i> menerima pesan <code>pingresp</code>	Perangkat <i>client</i> berhasil menerima pesan <code>pingresp</code>	Valid
6.	REQ-SLEEPING-KFSA-3105 -Client terkoneksi kembali	Pengujian pada <i>client</i> pada fungsi <code>sendGWMsg()</code> untuk membangun koneksi kembali	Perangkat <i>gateway</i> mendapat pesan dari <i>client</i>	Perangkat <i>gateway</i> berhasil menerima pesan pencarian <i>gateway</i> dari <i>client</i>	Valid

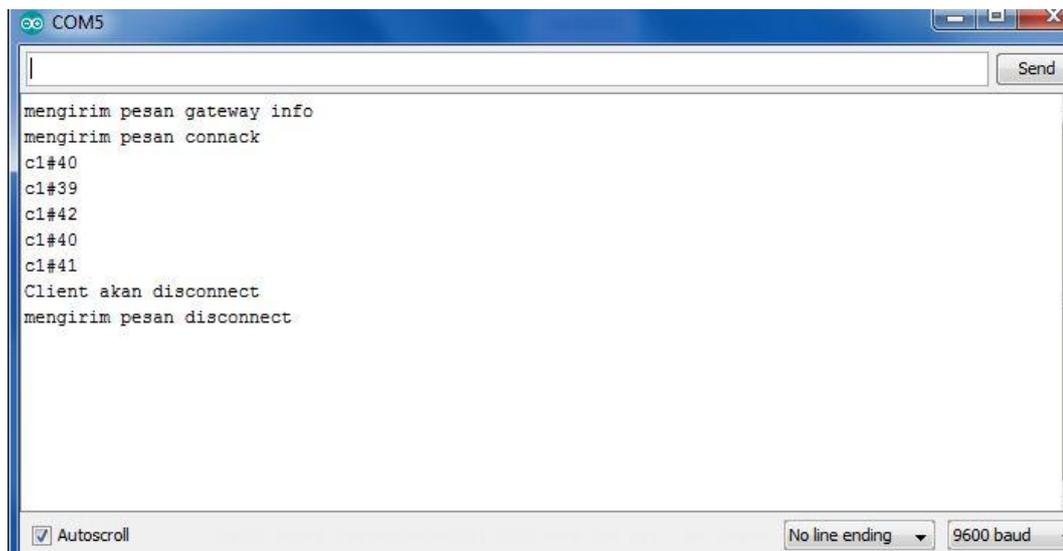
6.1.3. Pengujian Mekanisme Kondisi *Client Disconnect*

Pengujian mekanisme kondisi *client disconnect* mengacu dengan bab rekayasa kebutuhan dan implementasi yang telah dijelaskan. Pengujian dilakukan untuk mengetahui apakah implementasi yang dilakukan sudah sesuai dengan perancangan. Pada mekanisme ini *client* akan melakukan proses *disconnect* tanpa menggunakan durasi.



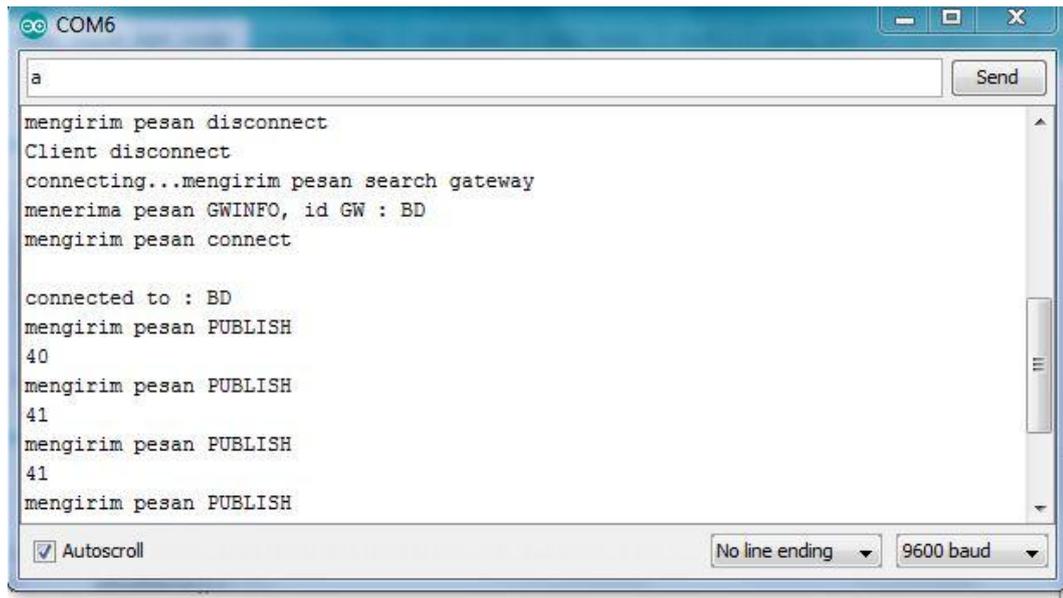
Gambar 6.7 Perangkat *client* mengirim pesan *disconnect* dan mendapat ACK

Gambar 6.7 menjelaskan bahwa perangkat *client* berhasil mengirim pesan *disconnect* kepada *gateway* dan menerima pesan *disconnect* ACK *gateway*. Selanjutnya perangkat *client* melakukan pemutusan koneksi



Gambar 6.8 Perangkat *gateway* menerima pesan *disconnect* dan mengirim ACK

Gambar 6.8 menjelaskan bahwa perangkat *gateway* berhasil menerima pesan *disconnect* dari *client* dan mengirim pesan *disconnect* ACK kepada *client*. Perangkat *gateway* akan menunggu pesan yang akan dilakukan oleh perangkat *client*.



Gambar 6.9 Perangkat *client* terkoneksi dengan *interrupt*

Gambar 6.9 menjelaskan bahwa *client* terknoneksi kembali dengan *interrupt* dari *serial monitor* dengan memasukkan sebuah karakter yaitu ‘a’. dengan demikian perangkat *client* dapat berkomunikasi kembali dengan perangkat *gateway*.

Tabel 6.3 Tabel Pengujian Mekanisme Kondisi *Client Disconnect*

No.	Test Name	Test Case	Expected Result	Result	Status
1.	REQ-SLEEPING-KCD-3200 – <i>Client</i> mengirim pesan disconnect	Pengujian pada <i>client</i> fungsi <code>disMsg()</code> untuk mengirimkan pesan <i>disconnect</i>	Perangkat <i>gateway</i> mendapat pesan dari <i>client</i> berisi <i>disconnect</i> tanpa durasi	Perangkat <i>gateway</i> berhasil menerima pesan <i>disconnect</i> tanpa durasi dari <i>client</i>	Valid
2.	REQ-SLEEPING-KCD-3201– <i>Gateway</i> mengirim pesan <code>disconnect(ACK)</code>	Pengujian pada <i>gateway</i> fungsi <code>disMsgACK()</code> untuk menyetujui proses <i>disconnect</i>	Perangkat <i>client</i> mendapat pesan <i>disconnect</i>	Perangkat <i>client</i> berhasil menerima pesan berisi <i>disconnect</i>	Valid
3.	REQ-SLEEPING-KCD-3202–	Pengujian pada <i>client</i> fungsi	Perangkat <i>client</i>	Perangkat <i>gateway</i>	Valid

No.	Test Name	Test Case	Expected Result	Result	Status
	<i>Client disconnect</i>	<i>disconnect</i> untuk membuat koneksi <i>client</i> terputus	memutuskan koneksi dari <i>gateway</i>	berhasil memutuskan koneksi dair <i>gateway</i>	
4.	REQ-SLEEPING-KCD-3203 – <i>Client</i> terkoneksi kembali	Pengujian pada <i>client</i> dimana mendapat <i>interrupt</i> dari serial monitor berupa karakter 'a'	Perangkat <i>client</i> mendapat masukan karakter dan memulai koneksi ke <i>gateway</i>	Perangkat <i>client</i> berhasil mendapat masukan karakter dan memulia koneksi ke <i>gateway</i>	Valid

6.1.4. Pengujian Mekanisme Kejadian *Client* pada *State Lost*

Pengujian mekanisme kejadian *client lost* mengacu dengan bab rekayasa kebutuhan dan implementasi yang telah dijelaskan. Pengujian dilakukan untuk mengetahui apakah implementasi yang dilakukan sudah sesuai dengan perancangan. Pada mekanisme ini *gateway* mendeteksi keadaan *client* dalam keadaan *lost*.

```

COM7
Client diijinkan sleep
Gateway menerima pingreq
mengirim pesan gateway info
mengirim pesan connack
c1#36
c1#35
c1#36
c1#36
c1#36
41
Client lost
42
Client lost
Autoscroll No line ending 9600 baud

```

Gambar 6.10 Perangkat *gateway* mendeteksi perangkat *client* dalam *state lost*

Gambar 6.10 menjelaskan perangkat *gateway* berhasil mendeteksi kondisi perangkat *client* dalam keadaan *lost*. Kondisi *lost* pada gambar terjadi karena

gateway tidak menerima pesan dari perangkat *client* yang melebihi durasi *sleep* dari perangkat *client*.

```

COM6
a| Send
mengirim pesan PUBLISH
34
mengirim pesan PUBLISH
35
mengirim pesan untuk sleep
Client sleep
Mengirim Pingreq
Pingresp diterima
connecting...mengirim pesan search gateway
menerima pesan GWINFO, id GW : BD
mengirim pesan connect

connected to : BD
mengirim pesan PUBLISH
36
mengirim pesan PUBLISH
35
mengirim pesan PUBLISH
36
mengirim pesan PUBLISH
36
mengirim pesan PUBLISH
36
mengirim pesan PUBLISH
36
connecting...mengirim pesan search gateway
menerima pesan GWINFO, id GW : BD
mengirim pesan connect

connected to : BD

```

Gambar 6.11 Perangkat *client* melakukan koneksi kembali

Gambar 6.11 menjelaskan bahwa setelah perangkat *gateway* mendeteksi terjadinya *lost* maka perangkat *client* melakukan koneksi kembali dengan menerima *interrupt* dari *serial monitor* dengan karakter ‘a’.

Tabel 6.4 Tabel Pengujian Mekanisme Kejadian *Client* pada *State Lost*

No.	Test Name	Test Case	Expected Result	Result	Status
1.	REQ-SLEEPING-KCL-3300– <i>Gateway</i> mendeteksi tidak ada pesan	Pengujian pada <i>gateway</i> dengan fungsi <i>countdown</i> bahwa <i>client</i> tidak	Tidak ada pesan dari perangkat <i>client</i> hingga melewati <i>durasi</i>	Perangkat <i>client</i> berhasil tidak mengirim pesan hingga melewati durasi	Valid

		memberikan pesan			
2.	REQ-SLEEPING-KCL-3301- <i>Gateway</i> mendeteksi <i>sleep client</i> melebihi durasi	Pengujian pada <i>gateway</i> dengan fungsi <i>countdown</i> bahwa <i>client sleep</i> terlalu lama	<i>Sleep</i> Perangkat <i>client</i> melewati durasi yang berada di perangkat <i>gateway</i>	<i>Sleep</i> Perangkat <i>client</i> berhasil melewati durasi <i>countdown</i> yang berada di <i>gateway</i>	Valid
3.	REQ-SLEEPING-KCL-3302- <i>Client</i> melakukan koneksi kembali	Pengujian pada <i>client</i> yang melakukan koneksi kembali	Perangkat <i>client</i> mengirim pesan <code>sendGWMsg()</code>	Perangkat <i>client</i> berhasil mengirim pesan <code>sendGWMsg()</code>	Valid

6.1.5. Pengujian Mekanisme Kerja 2 Client dan Gateway

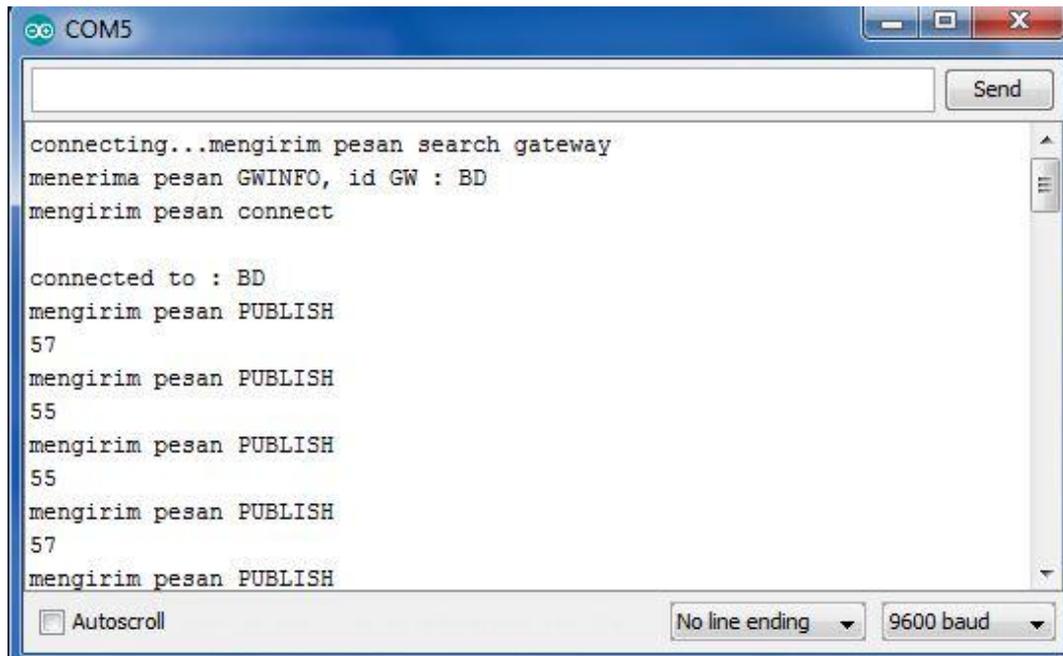
Pengujian mekanisme kerja 2 *client* dan 1 *gateway* mengacu dengan bab rekayasa kebutuhan dan implementasi yang telah dijelaskan. Pengujian dilakukan untuk mengetahui apakah implementasi yang dilakukan sudah sesuai dengan perancangan. Pada mekanisme ini *client* terdapat 2 buah yang mengirim data sensor kepada 1 *gateway*.

```

COM6
connecting...mengirim pesan search gateway
.mengirim pesan search gateway
menerima pesan GWINFO, id GW : BD
mengirim pesan connect

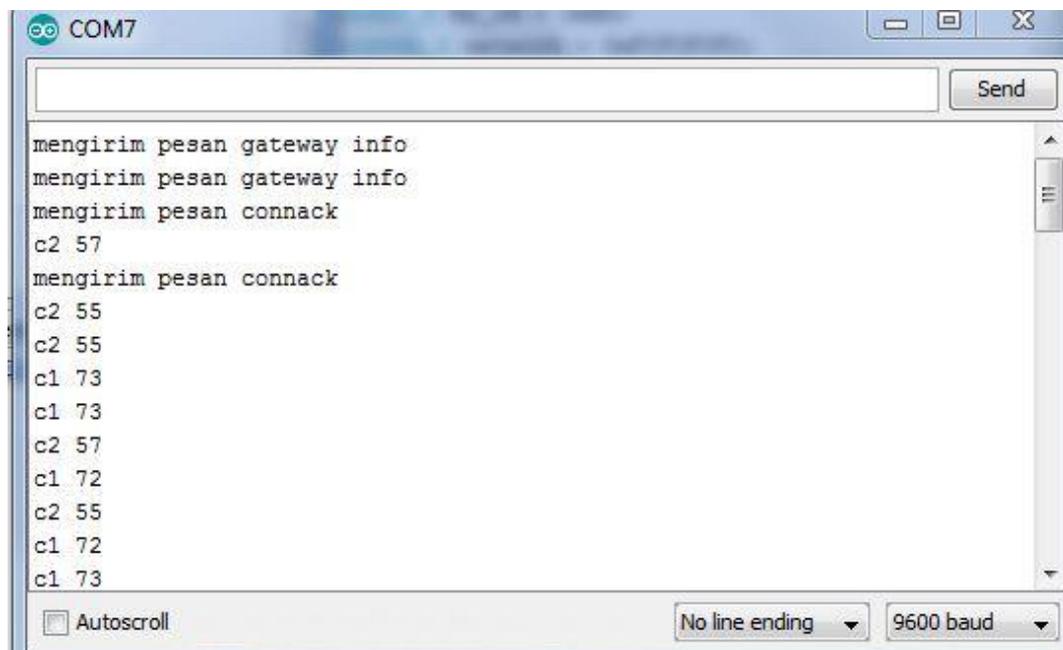
connected to : BD
mengirim pesan PUBLISH
73
mengirim pesan PUBLISH
73
mengirim pesan PUBLISH
72
mengirim pesan PUBLISH
72
Autoscroll
No line ending
9600 baud
  
```

Gambar 6.12 Perangkat client1 berhasil melakukan koneksi ke *gateway*



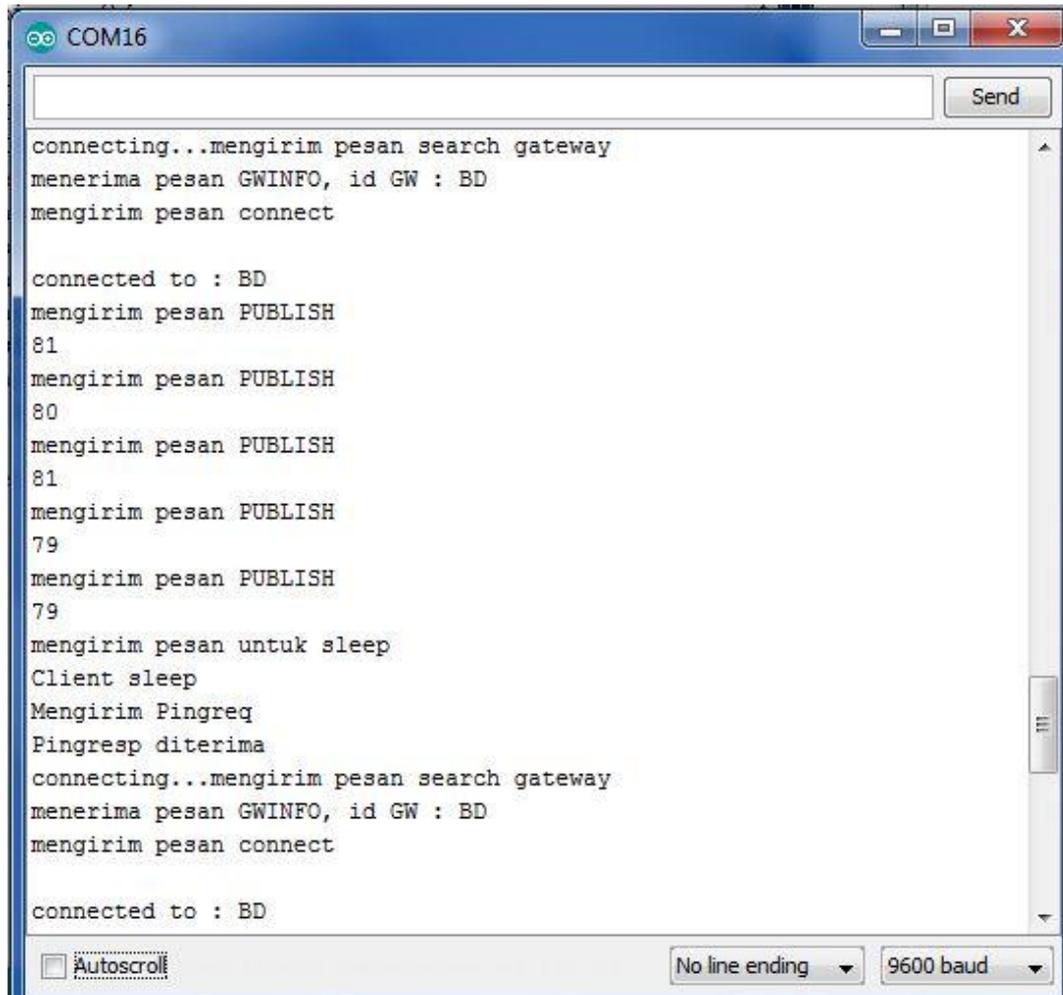
Gambar 6.13 Perangkat client2 berhasil terkoneksi dengan gateway

Pada Gambar 6.12 dan 6.13 merupakan *serial monitor* dari Client1 dan Client2 yang terhubung dengan *gateway* sehingga dapat melakukan pengiriman data sensor dari kedua *client* tersebut.

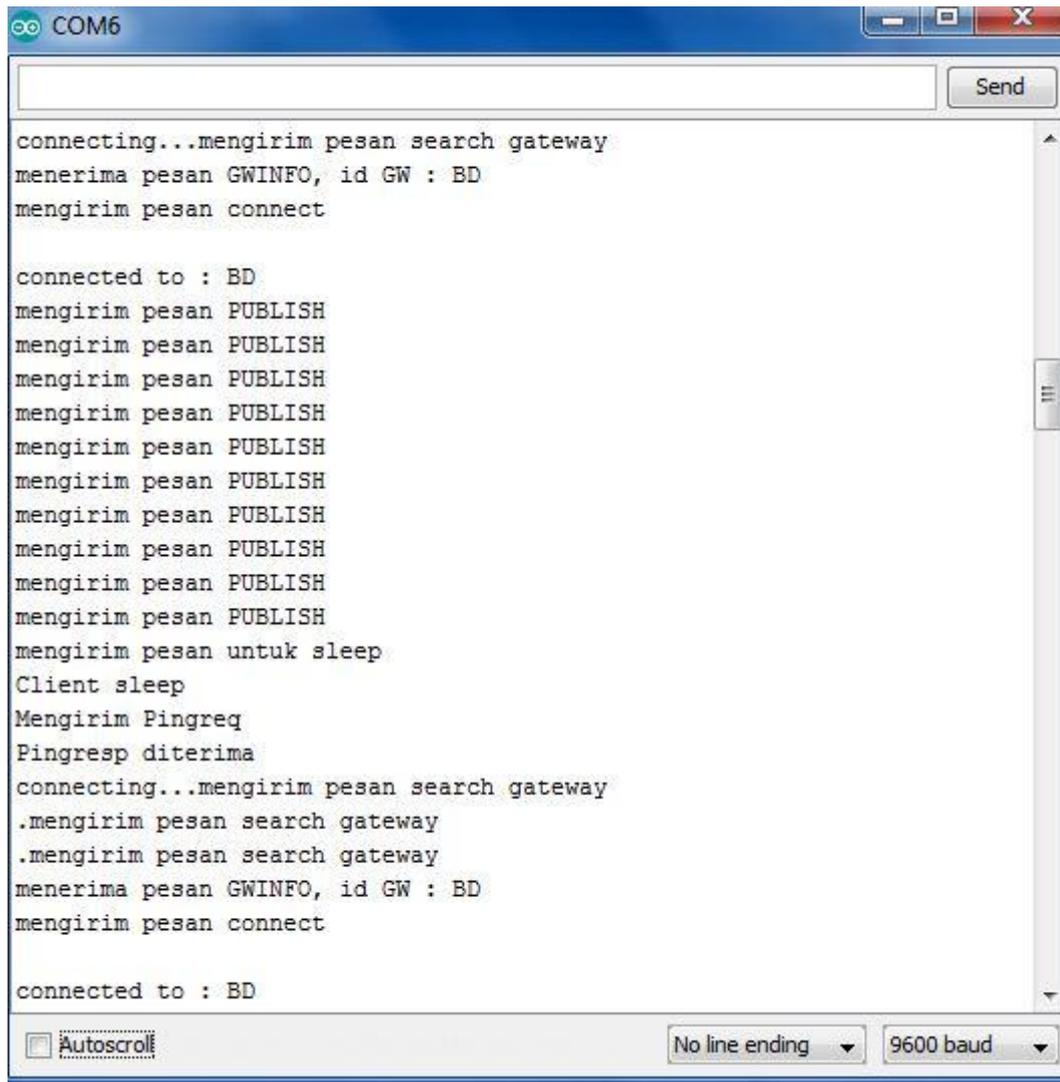


Gambar 6.14 Perangkat Gateway menerima pesan dari kedua client

Gambar 6.14 merupakan perangkat *gateway* menerima pesan dari kedua *client* yang dimulai dengan client2 terkoneksi dahulu kepada *gateway* dan diteruskan oleh client1 dimana *id* client1 yaitu C1 dan client2 yaitu C2.

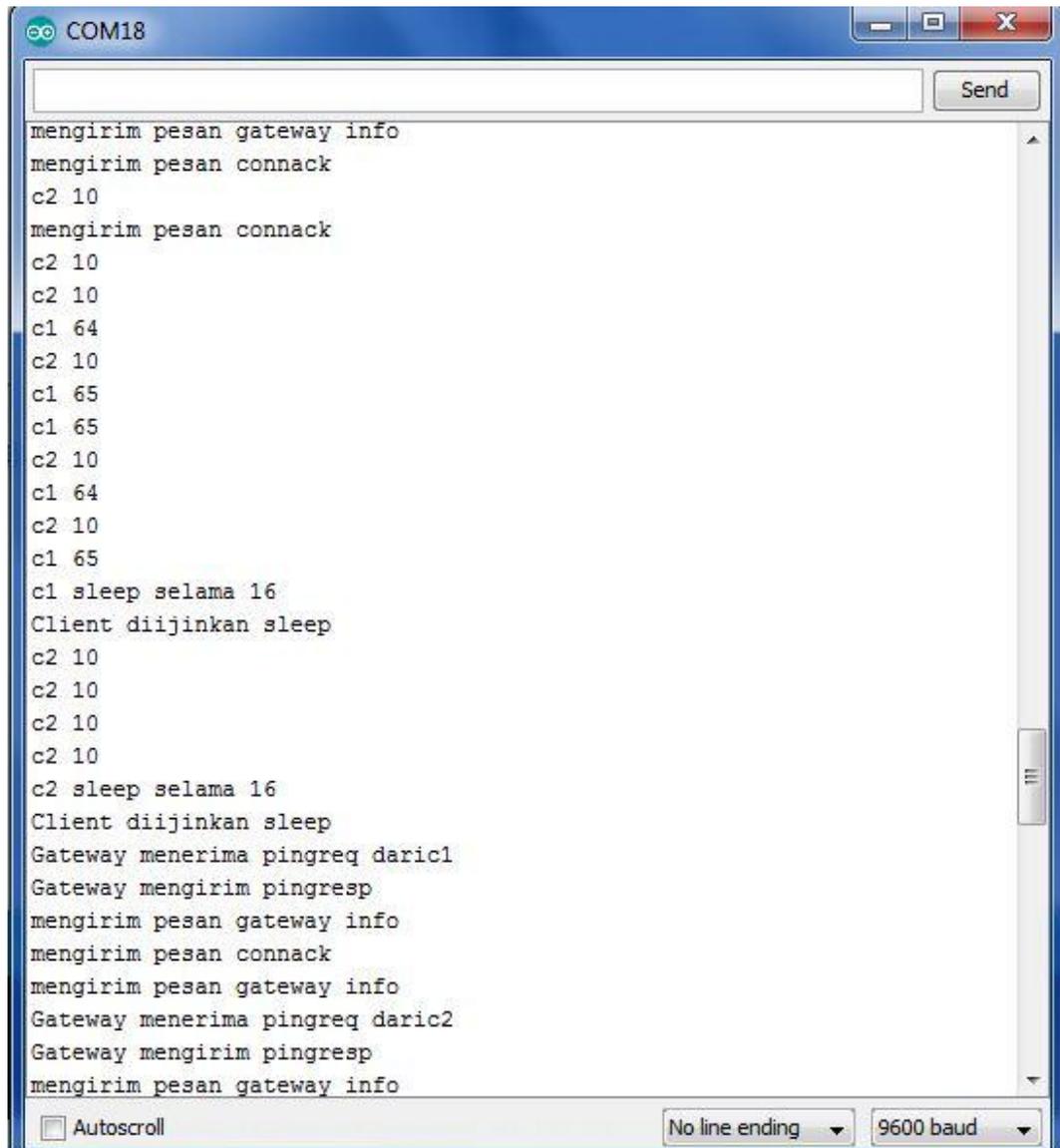


Gambar 6.15 Perangkat client1 melakukan fitur *sleep* dan *awake*



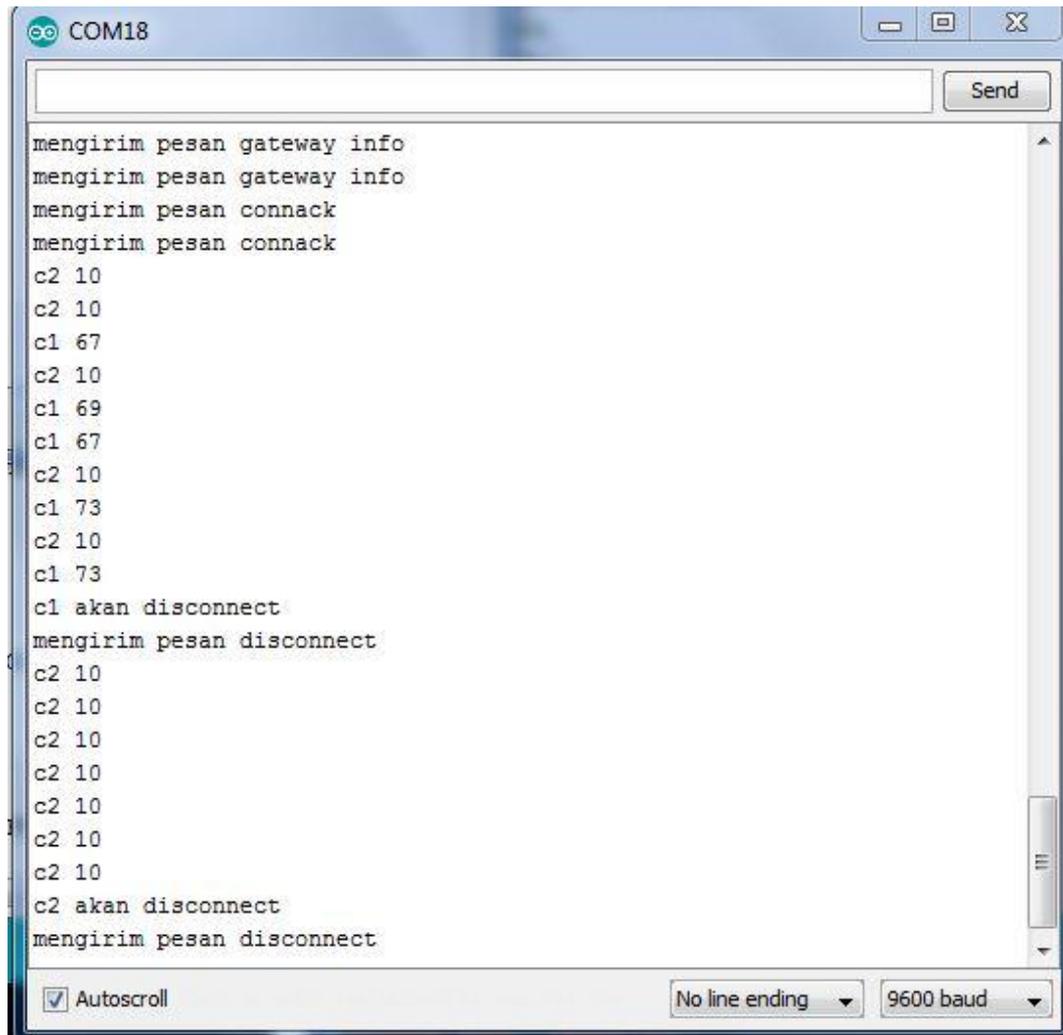
Gambar 6.16 Perangkat client2 melakukan fitur *sleep* dan *awake*

Pada Gambar 6.15 dan 6.16 menjelaskan kedua *client* melakukan fitur *sleep* dengan memulai mengirim pesan *sleep* terlebih dahulu kepada *gateway* dan menunggu ACK dari *gateway*. *Sleep* akan dijalankan dan akan *awake* kembali dengan mengirim pesan pingreq kepada perangkat *gateway*.



Gambar 6.17 Perangkat *gateway* menerima pesan *sleep* dan *pingreq* kedua *client*

Gambar 6.17 menjelaskan perangkat *gateway* menerima pesan *sleep* dan mengirim ACK untuk mengijinkan fitur *sleep* kedua *client* dan menunggu pesan *pingreq* dari kedua *client* untuk mengirim pesan *pingresp* yang agar kedua *client* dapat terhubung kembali.



Gambar 6.18 Perangkat gateway menerima pesan *disconnect* kedua *client*

Gambar 6.18 menjelaskan bahwa kedua *client* berhasil melakukan *disconnect* dengan ijin dari *gateway*. Proses selanjutnya yang dilakukan oleh kedua *client* sama dengan pengujian dengan menggunakan satu *client* pada pengujian mekanisme kondisi *client disconnect*.

```

COM16
67
mengirim pesan PUBLISH
68
mengirim pesan PUBLISH
67
mengirim pesan PUBLISH
67
mengirim pesan untuk sleep
Client sleep
Mengirim Pingreq
Pingresp diterima
connecting...mengirim pesan search gateway
.mengirim pesan search gateway
.mengirim pesan search gateway
.mengirim pesan search gateway
menerima pesan GWINFO, id GW : BD
mengirim pesan connect

connected to : BD
mengirim pesan PUBLISH
67
mengirim pesan PUBLISH
68
mengirim pesan PUBLISH
67
mengirim pesan PUBLISH
66
mengirim pesan PUBLISH
66

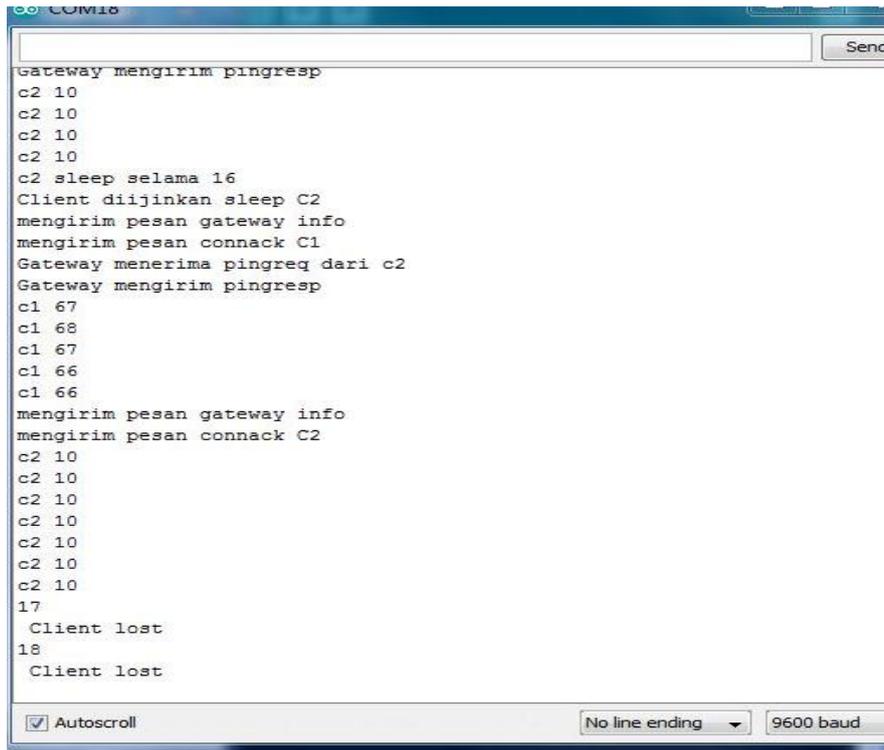
COM17
connected to : BD
mengirim pesan PUBLISH
mengirim pesan untuk sleep
Client sleep
Mengirim Pingreq
Pingresp diterima
connecting...mengirim pesan search gateway
.mengirim pesan search gateway
.mengirim pesan search gateway
menerima pesan GWINFO, id GW : BD
mengirim pesan connect

connected to : BD
mengirim pesan PUBLISH

```

Gambar 6.19 Kedua *client* masuk *state lost*

Gambar 6.19 merupakan kondisi kedua *client* jika dalam *state lost*, dimana kedua *client* tidak mengirim pesan apapun kepada *gateway* yang melebihi durasi *sleep* kedua *client*.



Gambar 6.20 Gateway mendeteksi kedua *client* sedang *lost*

Gambar 6.20 menjelaskan bahwa *gateway* mendeteksi kedua *client* dalam keadaan *lost*, jika salah satu *client* masih berkomunikasi dengan *gateway*, maka *gateway* tidak akan mendeteksi terjadinya kondisi *lost*.

Tabel 6.5 Tabel Pengujian Mekanisme Kerja 2 *Client* dan *Gateway*

No.	Test Name	Test Case	Expected Result	Result	Status
1.	REQ-SLEEPING-K2CG-3400-2 <i>Client</i> mencari <i>Gateway</i>	Pengujian pada <i>client</i> fungsi <code>sendGWMsg()</code> untuk mencari <i>gateway</i>	Perangkat <i>gateway</i> mendapat pesan dari <i>client</i>	Perangkat <i>gateway</i> berhasil menerima pesan pencarian <i>gateway</i> dari <i>client</i>	Valid
2.	REQ-SLEEPING-K2CG-3401- <i>Gateway</i> mengirim	Pengujian pada <i>gateway</i> fungsi <code>sendGWinfo()</code> untuk mengirim <i>id</i> miliknya	Perangkat <i>client</i> mendapat pesan berisi <i>id gateway</i>	Perangkat <i>client</i> berhasil menerima pesan berisi <i>id gateway</i>	Valid

No.	Test Name	Test Case	Expected Result	Result	Status
	<i>id</i> kepada <i>client</i>				
3.	REQ-SLEEPING-K2CG-3402-2 <i>Client</i> mengirim pesan <i>connect</i>	Pengujian pada <i>client</i> fungsi <i>connect()</i> untuk mengirim pesan <i>connect</i> ke <i>gateway</i>	Perangkat <i>gateway</i> mendapat pesan <i>connect</i> dari <i>client</i>	Perangkat <i>gateway</i> berhasil menerima pesan <i>connect</i> dari <i>client</i>	Valid
4.	REQ-SLEEPING-K2CG-3403- <i>Gateway</i> mengirim ACK <i>connect</i>	Pengujian pada <i>gateway</i> pada fungsi <i>sendConnACK()</i> untuk menyetujui pesan <i>connect</i> <i>client</i>	Perangkat <i>client</i> mendapat pesan <i>ack</i> dari <i>gateway</i> dan langsung terkoneksi	Perangkat <i>client</i> berhasil mendapat pesan <i>ack</i> dari <i>gateway</i> dan melakukan koneksi	Valid
5.	REQ-SLEEPING-K2CG-3404-2 <i>Client</i> mengirim data sensor	Pengujian pada <i>client</i> pada fungsi <i>sendData()</i> untuk mengirim data sensor	Perangkat <i>gateway</i> menerima data sensor dari <i>client</i>	Perangkat <i>gateway</i> berhasil menerima data sensor beserta <i>id</i> dari <i>client</i>	Valid
6.	REQ-SLEEPING-K2CG-3405-2 <i>Client</i> mengirim pesan <i>sleep</i>	Pengujian pada <i>client</i> fungsi <i>sleepMsg()</i> untuk mengirim pesan <i>sleep</i> kepada <i>gateway</i>	Perangkat <i>gateway</i> mendapat pesan dari <i>client</i> beserta durasi <i>sleep</i>	Perangkat <i>gateway</i> berhasil menerima pesan <i>sleep</i> dari <i>client</i> yang berisi durasi <i>sleep</i>	Valid
7.	REQ-SLEEPING-K2CG-3406- <i>Gateway</i> mengirim pesan <i>sleep</i> ACK	Pengujian pada <i>gateway</i> fungsi <i>sleepMsgACK()</i> untuk mengirim <i>id</i> miliknya	Perangkat <i>client</i> mendapat pesan persetujuan <i>sleep</i>	Perangkat <i>client</i> berhasil menerima pesan berisi <i>ack</i> tentang <i>sleep</i> <i>client</i>	Valid
8.	REQ-SLEEPING-	Pengujian pada <i>client</i> fungsi	Perangkat <i>client</i> <i>sleep</i> dan <i>count</i>	Perangkat <i>client</i> berhasil <i>sleep</i>	Valid

No.	Test Name	Test Case	Expected Result	Result	Status
	K2CG-3407-2 <i>Client</i> melakukan proses <i>sleep</i>	<code>do_sleep()</code> untuk melakukan proses <i>sleep</i>	<i>down</i> durasi pada perangkat <i>gateway</i> tidak lebih	dan durasi <i>sleep</i> tidak lebih	
9.	REQ-SLEEPING-K2CG-3408-2 <i>Client</i> menuju kondisi <i>awake</i> dan mengirim pesan <i>pingreq</i>	Pengujian pada <i>client</i> fungsi <code>wakeup()</code> untuk mengaktifkan kembali perangkat <i>client</i> dan mengirim pesan <code>pingreq()</code>	Perangkat <i>client</i> kembali aktif dan perangkat <i>gateway</i> menerima pesan <i>pingreq</i>	Perangkat <i>client</i> berhasil aktif dan perangkat <i>gateway</i> menerima pesan <i>pingreq</i>	Valid
10.	REQ-SLEEPING-K2CG-3409 - <i>Gateway</i> mengirim <i>pingresp</i>	Pengujian pada <i>gateway</i> pada fungsi <code>pingresp()</code> menanggapi <i>pingreq</i> dari <i>client</i>	Perangkat <i>client</i> menerima pesan <i>pingresp</i>	Perangkat <i>client</i> berhasil menerima pesan <i>pingresp</i>	Valid
11.	REQ-SLEEPING-K2CG-3410 - 2 <i>Client</i> dapat melakukan koneksi dengan <i>gateway</i>	Pengujian pada <i>client</i> pada fungsi <code>sendGWMsg()</code> untuk membangun koneksi kembali	Perangkat <i>gateway</i> mendapat pesan dari <i>client</i>	Perangkat <i>gateway</i> berhasil menerima pesan pencarian <i>gateway</i> dari <i>client</i>	Valid
12.	REQ-SLEEPING-K2CG-3411-2 <i>Client</i> mengirim pesan <i>disconnect</i>	Pengujian pada <i>client</i> fungsi <code>disMsg()</code> untuk mengirimkan pesan <i>disconnect</i>	Perangkat <i>gateway</i> mendapat pesan dari <i>client</i> berisi <i>disconnect</i> tanpa durasi	Perangkat <i>gateway</i> berhasil menerima pesan <i>disconnect</i> tanpa durasi dari <i>client</i>	Valid
13.	REQ-SLEEPING-	Pengujian pada <i>gateway</i> fungsi	Perangkat <i>client</i> mendapat	Perangkat <i>client</i> berhasil	Valid

No.	Test Name	Test Case	Expected Result	Result	Status
	K2CG-3412– <i>Gateway</i> mengirim pesan ack <i>disconnect</i>	<code>disMsgACK()</code> untuk menyetujui proses <i>disconnect</i>	pesan <i>disconnect</i>	menerima pesan berisi <i>disconnect</i>	
14.	REQ-SLEEPING-K2CG-3413–2 <i>client</i> melakukan <i>disconnect</i> dari <i>gateway</i>	Pengujian pada <i>client</i> fungsi <i>disconnect</i> untuk membuat koneksi <i>client</i> terputus	Perangkat <i>client</i> memutuskan koneksi dari <i>gateway</i>	Perangkat <i>gateway</i> berhasil memutuskan koneksi dair <i>gateway</i>	Valid
15.	REQ-SLEEPING-K2CG-3414–2 <i>client</i> melakukan proses koneksi kembali	Pengujian pada <i>client</i> dimana mendapat <i>interrupt</i> dari serial monitor berupa karakter 'a'	Perangkat <i>client</i> mendapat masukan karakter dan memulai koneksi ke <i>gateway</i>	Perangkat <i>client</i> berhasil mendapat masukan karakter dan memulia koneksi ke <i>gateway</i>	Valid
16.	REQ-SLEEPING-K2CG-3415– <i>Gateway</i> mendeteksi 2 <i>client lost</i>	Pengujian pada <i>gateway</i> dengan fungsi <i>countdown</i> bahwa <i>client</i> tidak memberikan pesan	Tidak ada pesan dari perangkat <i>client</i> hingga melewati <i>durasi</i>	Perangkat <i>client</i> berhasil tidak mengirim pesan hingga melewati durasi	Valid
17.	REQ-SLEEPING-K2CG-3416–2 <i>client</i> melakukan proses koneksi kembali	Pengujian pada <i>client</i> fungsi yang melakukan koneksi kembali	Perangkat <i>client</i> mengirim pesan <code>sendGWMsg()</code>	Perangkat <i>client</i> berhasil mengirim pesan <code>sendGWMsg()</code>	Valid