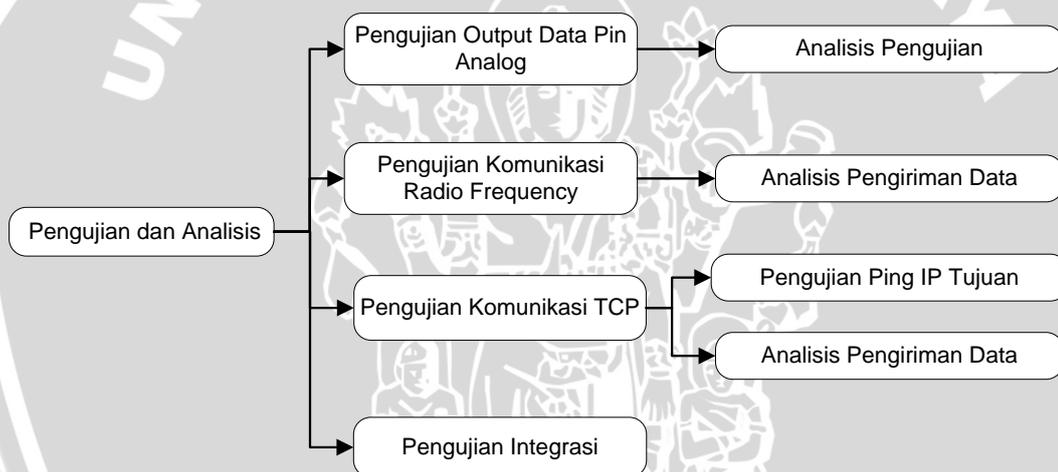


BAB VI PENGUJIAN DAN ANALISIS

Pada bab pengujian dan analisis dilakukan proses pengujian dan analisis terhadap sistem relay routing protocol yang telah dibangun. Proses pengujian dilakukan berdasarkan tujuan sistem semula yaitu sistem dapat berintegrasi keseluruhan. Pengujian difokuskan pada tiap-tiap subsistem, dimana terdapat pengujian komunikasi Radio Frequency, dan pengujian komunikasi TCP. Tiap subsistem ini diuji dan dianalisis performa pengiriman datanya. Jika pengujian antar subsistem selesai maka dilakukan pengujian integrasi, yaitu pengujian yang melibatkan keseluruhan subsistem agar menjadi kesatuan sistem. Berikut merupakan pohon diagram dari pengujian sistem.



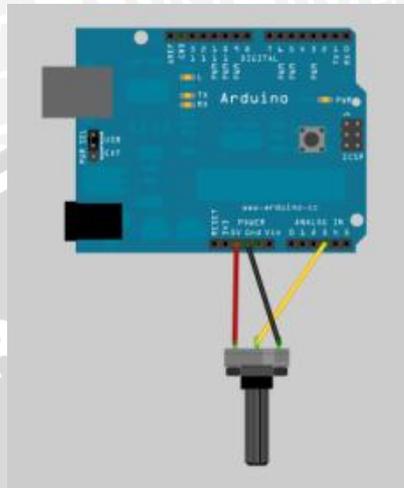
Gambar 6.1 Pohon Pengujian dan Analisis
Sumber : [Pengujian dan Analisis]

6.1 Pengujian Output Data Pin Analog

Pengujian data pin analog adalah pengujian data yang keluar dari pin analog pada papan arduino. Pada pengujian kali ini data yang diuji menggunakan potensiometer seperti pada gambar 6.2. Potensiometer bekerja berdasarkan perubahan resistansi. Ketika dialirkan arus listrik maka akan terjadi pembagian tegangan yang sebanding dengan besarnya resistansi sesuai dengan posisi potensiometer saat itu. Potensiometer yang terhubung dengan tegangan 5 volt akan memberikan keluaran tegangan antara 0 dan 5 volt pada pin bagian



tengahpotensiometer menuju pin analog yang ada pada arduino, tergantung pada sudut diputarnya potensiometer tersebut. Kemudian tegangan keluaran ini dapat digunakan sebagai sebuah variabel di dalam program.



Gambar 6.2 Konfigurasi Perancangan Hardware Pengujian Data Pin Analog
Sumber : [Pengujian dan Analisis]

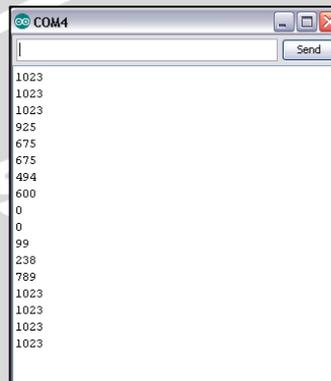
1. Hubungkan kaki 1 Potensiometer dengan pin 5 volt Arduino menggunakan kabel jumper (warna merah).
2. Hubungkan kaki Tengah Potensiometer dengan pin A0 Arduino menggunakan kabel jumper (warna kuning).
3. Hubungkan kaki 2 Potensiometer dengan pin GND Arduino menggunakan kabel jumper (warna hijau).
4. Pasangkan board Arduino dengan port USB komputer menggunakan kabel USB.
5. Bukalah IDE Arduino kemudian ketikkan sketch berikut.

```
pin_analog | Arduino 1.0.3
File Edit Sketch Tools Help
pin_analog $
void setup() {
  Serial.begin(9600);
}
void loop() {
  int valueDigital = analogRead(0);
  Serial.println(valueDigital);
  delay(500);
}
```

Gambar 6.3 Code Pengujian Analog Pin
Sumber : [Pengujian dan Analisis]

6. Upload ke papan Arduino menggunakan tombol "Upload" atau melalui File > Upload.
7. Bukalah jendela komunikasi serial pada IDE arduino caranya, pilih Tools > Serial Monitor.

Tampilan output pada serial monitor seperti pada gambar berikut



Gambar 6.4 Output Pengujian data Pin Analog
Sumber : [Pengujian dan Analisis]

6.1.1 Analisa Pengujian Output Data Pin Analog

Pengujian output data pin analog radio frequency menghasilkan hasil uji. Pada gambar 6.4 terlihat pada serial monitor menghasilkan output berupa data yang keluar dari pin analog. Jika potensiometer diputar maka hasil bacaannya antara 0 (batas bawah 0 volt) sampai 1023 (batas atas 5 volt) tergantung dengan sudut putar potensiometer. Ketika potensiometer diputar, angka-angka ini akan merespon dengan cepat dan menampilkan data input dari potensiometer dengan instan. Angka tersebut berasal dari perhitungan analog to digital converter.

$ADC = (V_{in} / V_{ref}) \times 1023$. Berarti jika inputan tegangan sebesar 2,5 V, maka nilai output yang keluar pada serial monitor adalah

$$\begin{aligned} ADC &= (V_{in} / V_{ref}) * 1023 \\ &= 2,5V / 5V * 1023 \\ &= 512 \end{aligned}$$

Berdasarkan perhitungan diatas dapat diketahui output ADC pada serial monitor berdasarkan inputan dibawah ini.

Tabel 6.1 Input Tegangan dan Nilai ADCnya

Tegangan	Nilai ADC di Arduino	Status
0	0	Berhasil
1	205	Berhasil
2	410	Berhasil
2,5	512	Berhasil
4	818	Berhasil
5	1023	Berhasil

Sumber: [Pengujian dan Analisis]

Berikut merupakan tabel pengujian output data pin analog

Tabel 6.2 Kasus Uji untuk Pengujian Output Data Pin Analog

Nama Kasus Uji	Kasus Uji Output Data Pin Analog
Objek Uji	Output keluaran pin analog 0
Tujuan Pengujian	Pengujian dilakukan untuk mengetahui output data analog yang berasal dari tegangan 5 volt arduinokemudian tegangan tersebut diconvert menjadi data digital pada ADC arduino dan diatur besar kecil tegangannya menggunakan potensiometer
Prosedur Uji	Penguji menjalankan program komunikasi serial, pada main loop dideklarasikan variabel yang menyimpan nilai tegangan analog dari pin analog 0 yang diatur besar hambatanny melalui potensiometer yaitu <code>int valueDigital = analogRead(A0);</code> Kemudian dari data yang diperoleh, ditampilkan dalam nilai desimal melalui perintah <code>Serial.println(valueDigital);</code>
Hasil yang diperoleh	Data keluaran berupa data digital dengan range 0-1023

Sumber: [Pengujian dan Analisis]

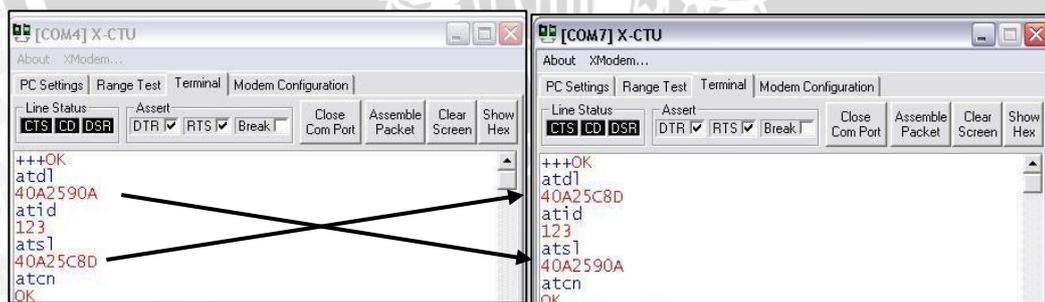
6.2 Pengujian Komunikasi Radio Frequency

Pengujian pada komunikasi *radio frequency* meliputi komunikasi kedua XBee dalam hal melakukan pengiriman dan penerimaan data dan pengiriman data melalui komunikasi serial dan analisisnya

6.2.1 Pengujian Komunikasi Radio Frequency via Terminal Monitor

Pengujian XBee adalah pengujian pengiriman karakter antar serial monitor pada kedua XBee. Pengujian menggunakan program terminal monitor dari program X-CTU. Pertama, atur pin arduino agar berada dalam mode reset, hubungkan pin reset dengan ground menggunakan kabel jumper. Arduino yang telah disusun dengan XBee shield dihubungkan dengan kabel serial ke USB yang ada di PC.

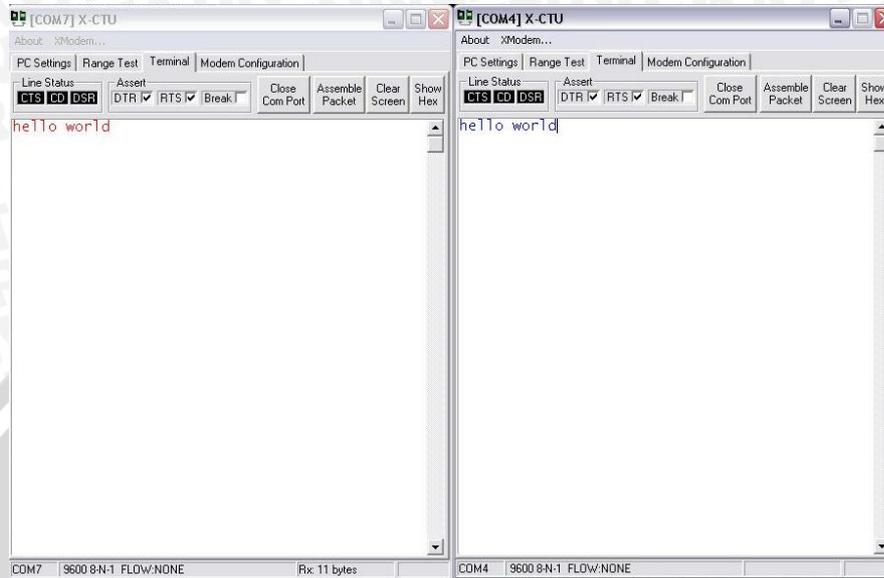
- Buka program X-CTU.
- Klik tab terminal
- Pada terminal monitor, mulai AT Command dengan memulai perintah “+++” tanpa enter maka akan keluar kembalian nilai OK
Ketikkan ATDL untuk melihat Destination Low dari XBee
Ketikkan ATID untuk melihat PAN ID XBee
Ketikkan ATSL untuk melihat Serial Number High XBee
Ketikkan ATCN untuk mengakhiri AT Command
- Masukkan perintah yang sama untuk XBee kedua.
- Settingan ATDL XBee pertama dengan ATSL XBee kedua harusnya memiliki nilai yang sama, begitu juga nilai ATSL XBee pertama dengan ATDL XBee kedua juga memiliki nilai yang sama. Contohnya seperti pada gambar berikut.



Gambar 6.5 Pencocokan ATDL dan ATSL pada kedua XBee
Sumber : [Pengujian dan Analisis]

- Setelah keluar dari AT Command, ketikkan karakter pada terminal monitor. String yang diketik pada terminal XBee pertama akan keluar di XBee kedua, begitu juga sebaliknya. Artinya kedua XBee sudah dapat

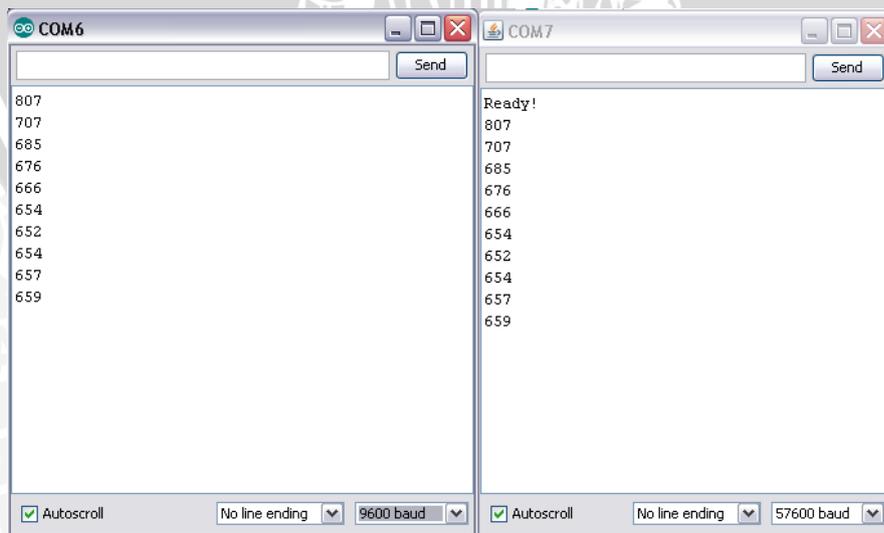
berkomunikasi melalui komunikasi Radio Frequency dengan terminal serial monitor sebagai mediana.



Gambar 6.6 Pengiriman Data Melalui Terminal X-CTU
Sumber : [Pengujian dan Analisis]

6.2.2 Pengujian Komunikasi Data Radio Frequency via Arduino IDE

Pengujian pengiriman data berupa hasil dari pengiriman data dengan dua XBee melalui program Arduino IDE. Arduino disusun dengan XBee melalui perantara XBee shield. Kabel serial dihubungkan untuk komunikasi dengan PC. Adapun tampilan keluaran data yang berhasil dikirim adalah sebagai berikut.



Gambar 6.7 Pengiriman Data antar dua XBee
Sumber:[Pengujian dan Analisis]

6.2.3 Analisa Pengujian Komunikasi antar Radio Frequency

Pengujian komunikasi radio frequency menghasilkan hasil uji. Pada gambar 6.7 terlihat dua serial monitor menampilkan karakter yang sama. COM 6 bertindak sebagai pengirim mengirimkan data pin analog dari arduino pengirim, data tersebut ditransmit dan diterima kemudian ditampilkan di COM 7.

Tabel 6.3 Kasus Uji untuk Pengujian Komunikasi antar Radio Frequency

Nama Kasus Uji	Kasus Uji Komunikasi Data Radio Frequency
Objek Uji	Komunikasi antar radio frequency
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa objek uji komunikasi antar radio frequency dapat bekerja dengan baik
Prosedur Uji	Penguji menjalankan program pengiriman data melalui komunikasi serial pada sisi pengirim. Membuka serial monitor pada sisi pengirim dan penerima.
Hasil yang diperoleh	Data yang dikirim keluar di serial monitor sisi penerima

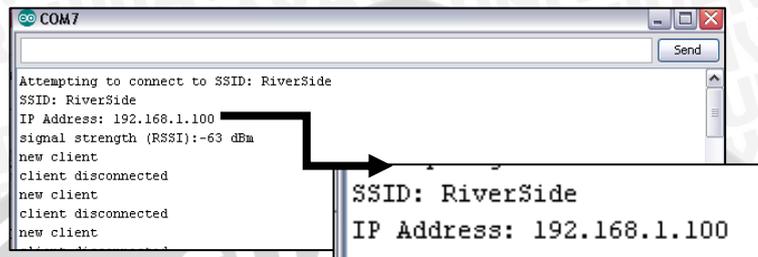
Sumber: [Pengujian dan Analisis]

6.3 Pengujian Komunikasi TCP/IP

Pengujian komunikasi TCP/IP meliputi proses pengiriman data dari arduino menuju komputer menggunakan komunikasi Wifi. Pengujian ini untuk mengetahui apakah Wifi Shield dapat bekerja dengan baik. Selain pengujian tersebut adalah menguji tingkat kecepatan Wifi Shield untuk melakukan koneksi dengan Access Point dan berkomunikasi dengan sink node.. Arduino wifi shield butuh inialisasi IP ke jaringan terlebih dahulu agar dapat berkomunikasi wireless dengan PC. Prosedur pengujian Komunikasi TCP/IP adalah sebagai berikut:

1. Susun Arduino dengan Arduino Wifi Shield dengan menghubungkan kaki wifi shield ke pin arduino
2. Buka aplikasi Arduino IDE
3. Pada menu utama, buka file yang sudah diterapkan pada implementasi middle node. klik File ->Open -> Middle_Node.ino

4. Upload program
5. Klik serial monitor untuk melihat proses koneksi Wifi Shield dengan *Access Point* dan menghitung waktu yang dibutuhkan Wifi Shield untuk melakukan koneksi dengan *Access Point*.



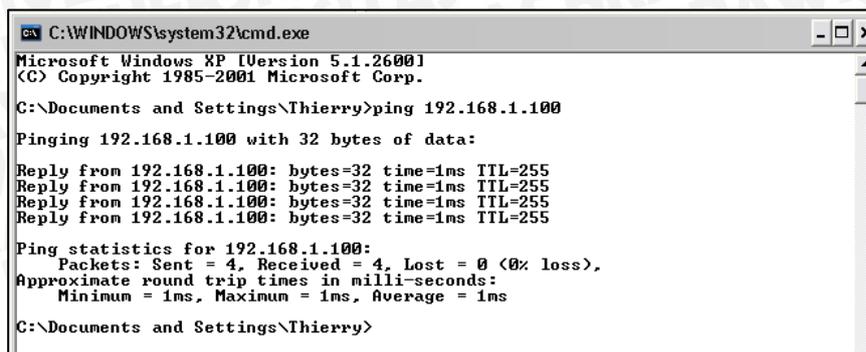
Gambar 6.8 Mendapatkan IP dari Jaringan
Sumber:[Pengujian dan Analisis]

Berdasarkan gambar 6.8 Serial Monitor Arduino, dapat dijelaskan bahwa proses koneksi ke SSID yang bernama “RiverSide” sudah berhasil dan Wifi Shield menerima IP 192.168.1.100.

6.3.1 Pengujian Ping IP Tujuan

Pengujian “PING” digunakan untuk menguji wifi shield benar-benar terkoneksi di jaringan untuk menguji kualitas transmisi yang digunakan. Program diagnosik “PING” bekerja dengan menggunakan protokol ICMP. Sedangkan protokol ICMP sendiri membutuhkan protokol IP serta SLIP untuk membawanya sampai ke tempat tujuan. Protokol ICMP disini diaplikasikan untuk proses ICMP echo. Setiap permintaan (ICMP echo request) dari PC harus dijawab oleh alat yang dibuat dengan menggunakan ICMP echo reply beserta muatan data yang diterima jika ada.

Setelah mendapatkan IP, Buka program command prompt, klik Start->Command Prompt. Ketikkan ping <Alamat IP tujuan> seperti pada gambar dibawah berikut



```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Thierry>ping 192.168.1.100

Pinging 192.168.1.100 with 32 bytes of data:

Reply from 192.168.1.100: bytes=32 time=1ms TTL=255

Ping statistics for 192.168.1.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms

C:\Documents and Settings\Thierry>

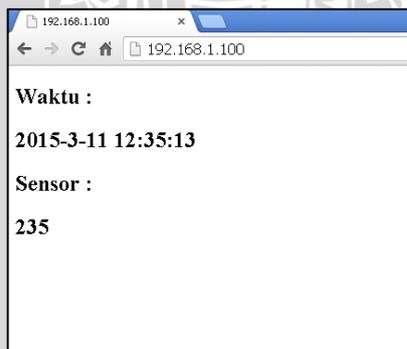
```

Gambar 6.9 Pengujian PING
Sumber:[Pengujian dan Analisis]

Dari hasil pengujian terlihat bahwa empat buah ICMP Echo Request yang dihasilkan program "PING" berhasil dijawab oleh alat dengan baik dan *time* yang dihasilkan relative kecil. Hal tersebut menunjukkan bahwa alat sedang berada di jaringan bersama-sama dengan PC dan alat mampu menangani ICMP Echo Request yang diberikan kepadanya serta waktu pengiriman paket data dari board arduino ke komputer ditempuh dengan cepat

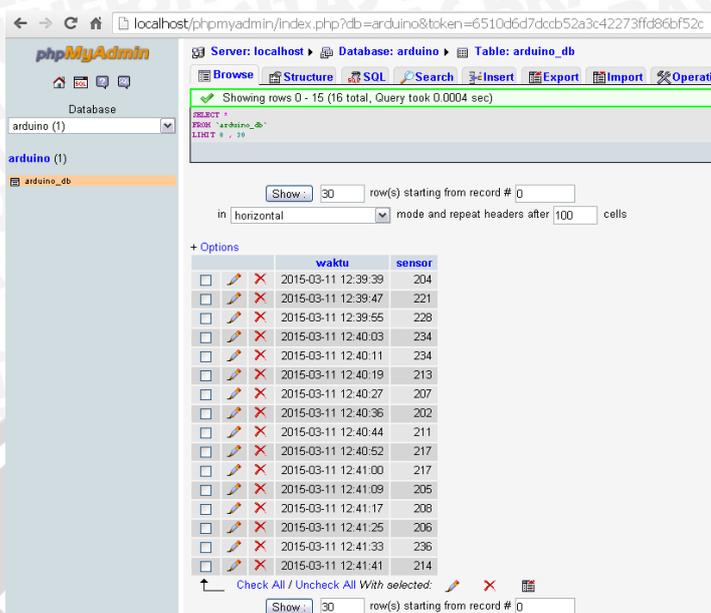
6.3.2 Pengujian Komunikasi Data TCP/IP

Buka web browser, masukkan alamat localhost/phpmyadmin. Pada web browser muncul tampilan waktu dan sensor di database. Data waktu dan sensor tersebut akan masuk ke database tiap 10 detik.



Gambar 6.10 Data yang Diterima pada Web Browser
Sumber:[Pengujian dan Analisis]

Buka localhost/phpmyadmin untuk membuka database yang telah disimpan. Database akan menyimpan data waktu, sensor yang telah dikirim dan akan menampilkan pada web browser yang muncul tiap interval 10 detik tadi



Gambar 6.11 Data disimpan dalam Database
Sumber:[Pengujian dan Analisis]

Pengujian lanjutan adalah mencatat selisih waktu antara pengiriman data dari source node dengan penerimaan data pada database sink node. Selisih waktu dipengaruhi beberapa hal seperti *traffic* jaringan, dan pencocokan waktu upload program pengiriman dan program penerimaan datayangkadang tidak bersamaan. Berikut hasil prosedur uji selisih waktu pengiriman data dengan database dengan 10 sampel pengiriman

Tabel 6.4 Hasil Selisih Waktu Pengiriman Data Middle Node-Sink Node

No	Selisih Waktu	Status
1.	2,46 detik	Berhasil
2.	1,41 detik	Berhasil
3.	3,03 detik	Berhasil
4.	8,52 detik	Gagal
5.	2,52 detik	Berhasil
6.	3,05 detik	Berhasil
7.	3,49 detik	Berhasil
8.	1,2 detik	Berhasil
9.	3,05 detik	Berhasil
10	4,55 detik	Berhasil
Persentase Keberhasilan Pengiriman		90%

Sumber: [Pengujian dan Analisis]

6.3.3 Analisa Pengiriman Paket Data via TCP/IP

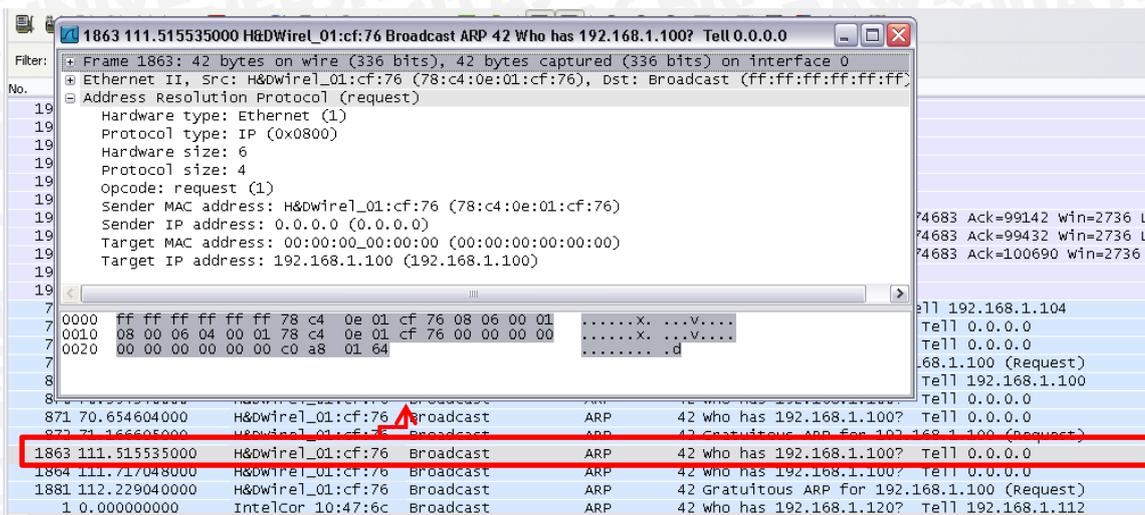
Sebelum memulai proses koneksi HTTP, proses transmisi *Address Resolution Protocol* (ARP) harus dilakukan dahulu. ARP bertanggung jawab dalam melakukan resolusi alamat IP ke dalam alamat *Media Access Control* (MAC Address), karena ketika sebuah aplikasi yang mendukung teknologi TCP/IP yang mencoba untuk mengakses sebuah *host* dengan menggunakan alamat IP, maka alamat IP yang dimiliki ini harus diterjemahkan dahulu ke dalam MAC Address. Hal ini bertujuan agar *frame* data dapat diteruskan ke tujuan dan diletakkan di atas media transmisi. Berikut hasil *capturing* komunikasi data sensor node dengan data sink dengan menggunakan Wireshark :

No.	Time	Source	Destination	Protocol	Length	Info
13	0.931015000	119.235.248.60	192.168.1.117	DNS	139	Standard query response 0x9334 CNAME appspot.1.google.com A 74.125.2
14	0.933797000	192.168.1.117	74.125.200.141	TCP	62	pcia-rxp-b > https [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM=1
15	0.995156000	0.0.0.0	255.255.255.255	DHCP	350	DHCP Request - Transaction ID 0xabcd0002
16	0.995953000	H&Dwirel_01:cf:76	Broadcast	ARP	42	who has 192.168.1.100? Tell 0.0.0.0
17	1.148118000	119.235.248.60	192.168.1.117	DNS	128	Standard query response 0x8b54 CNAME appspot.1.google.com A 74.125.2
18	1.150683000	192.168.1.117	74.125.200.141	TCP	62	passwd-policy > https [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM
19	1.197467000	H&Dwirel_01:cf:76	Broadcast	ARP	42	who has 192.168.1.100? Tell 0.0.0.0
20	1.697331000	184.172.2.99	192.168.1.117	TCP	60	http > pip [FIN, ACK] Seq=561 Ack=2 win=6432 Len=0
21	1.697375000	192.168.1.117	184.172.2.99	TCP	54	pip > http [ACK] Seq=2 Ack=562 win=16240 Len=0
22	1.709442000	H&Dwirel_01:cf:76	Broadcast	ARP	42	Gratuitous ARP for 192.168.1.100 (Request)
23	1.866586000	184.172.2.99	192.168.1.117	TCP	62	http > intersan [SYN, ACK] Seq=0 Ack=1 win=5840 Len=0 MSS=1400 SACK_P
24	1.866629000	192.168.1.117	184.172.2.99	TCP	54	intersan > http [ACK] Seq=1 Ack=1 win=16800 Len=0
25	1.871169000	192.168.1.117	184.172.2.99	HTTP	511	POST /market HTTP/1.1 (application/x-www-form-urlencoded)
26	1.978942000	74.125.200.141	192.168.1.117	TCP	62	https > pcia-rxp-b [SYN, ACK] Seq=0 Ack=1 win=62920 Len=0 MSS=1400 SA
27	1.978984000	192.168.1.117	74.125.200.141	TCP	54	pcia-rxp-b > https [ACK] Seq=1 Ack=1 win=16800 Len=0
28	1.983663000	192.168.1.117	74.125.200.141	SSL	134	Client Hello
29	2.204892000	74.125.200.141	192.168.1.117	TCP	62	https > passwd-policy [SYN, ACK] Seq=0 Ack=1 win=62920 Len=0 MSS=140
30	2.204939000	192.168.1.117	74.125.200.141	TCP	54	passwd-policy > https [ACK] Seq=1 Ack=1 win=16800 Len=0

Gambar 6.12 Hasil Capture Komunikasi Data Wifi Shield dan Sink Node
Sumber: [Pengujian dan Analisis]

Berdasarkan gambar 6.12 hasil capture komunikasi data wifi shield dan sink node, terdapat beberapa protokol yang memproses komunikasi data. Protokol tersebut adalah ARP, TCP dan HTTP. Berikut penjelasannya :

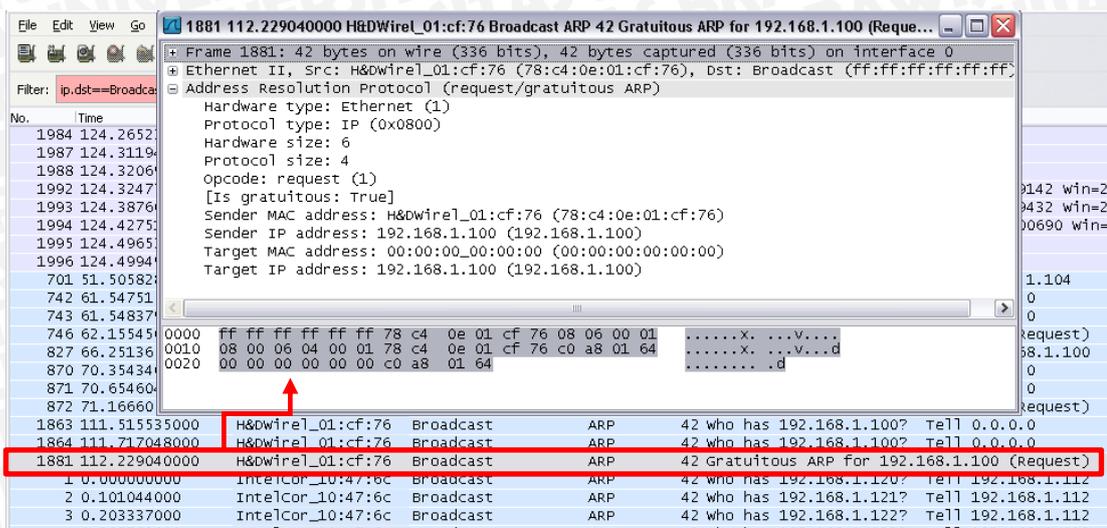
1. Hasil *Capture* Komunikasi Data wifi shield dan sink node, dapat dilihat Source H&Dwirel_01:cf:76 dengan MAC Address seperti di gambar 6.13 bawah ini :



**Gambar 6.13 Hasil Capture ARP Komunikasi Data Wifi Shield dan Sink Node
Sumber: [Pengujian dan Analisis]**

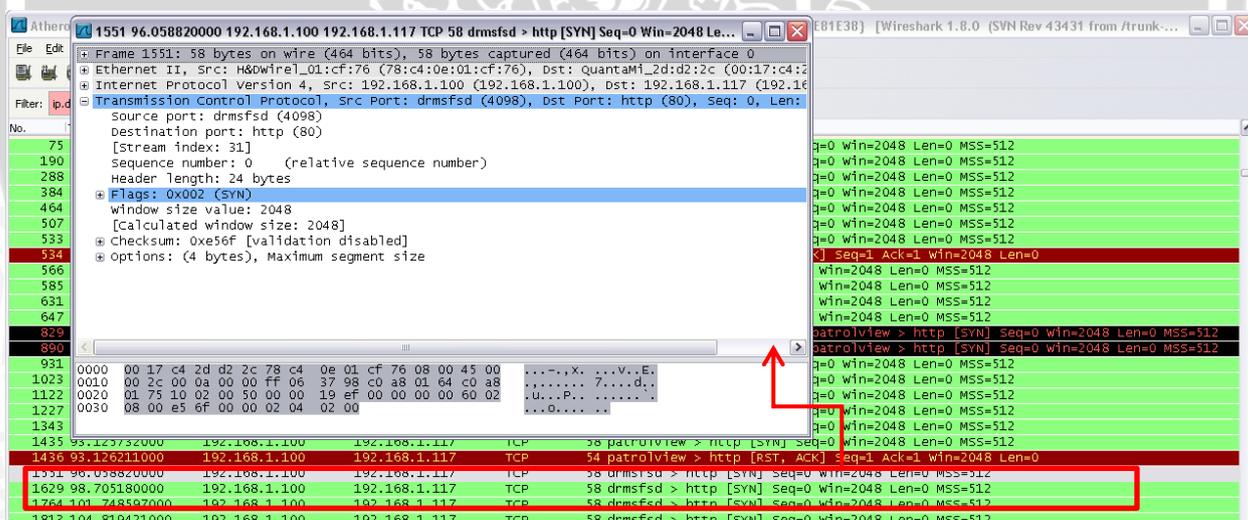
Berdasarkan Gambar 6.13 hasil *capture* ARP komunikasi data wifi shield dan sink node, MAC Address Source adalah (78:c4:0e:01:cf:76). Ini merupakan MAC Address pada wifi shield. Wifi shield akan melakukan komunikasi HTTP ke sink node. Namun sebelumnya wifi shield melakukan koneksi pada *Access Point* dengan cara broadcasting, yaitu mendefinisikan satu persatu IP yang available dalam jaringan secara DHCP, maka dari itu target MAC Address tertulis 00:00:00:00:00:00. Kemudian Access point akan menginformasikan bahwa “Who has 192.168.1.100? Tell 0.0.0.0”, access point mengumumkan bahwa wifi shield memiliki MAC Address (78:c4:0e:01:cf:76) menanyakan kepada access point di jaringan dengan gateway 192.168.1.xx dengan MAC Address berapa yang memiliki IP 192.168.1.100. Jika belum ada yang memiliki IP 192.168.1.100, maka wifi shield dan mendapatkan IP 192.168.1.100 dari access point untuk berkomunikasi dengan sink node seperti pada gambar 6.14 dibawah ini:





Gambar 6.14 Hasil Capture Request IPWifi Shield
Sumber: [Pengujian dan Analisis]

2. Pada gambar 6.15 hasil *capture* komunikasi data sensor node dan data sink dibawah adalah proses request TCP.

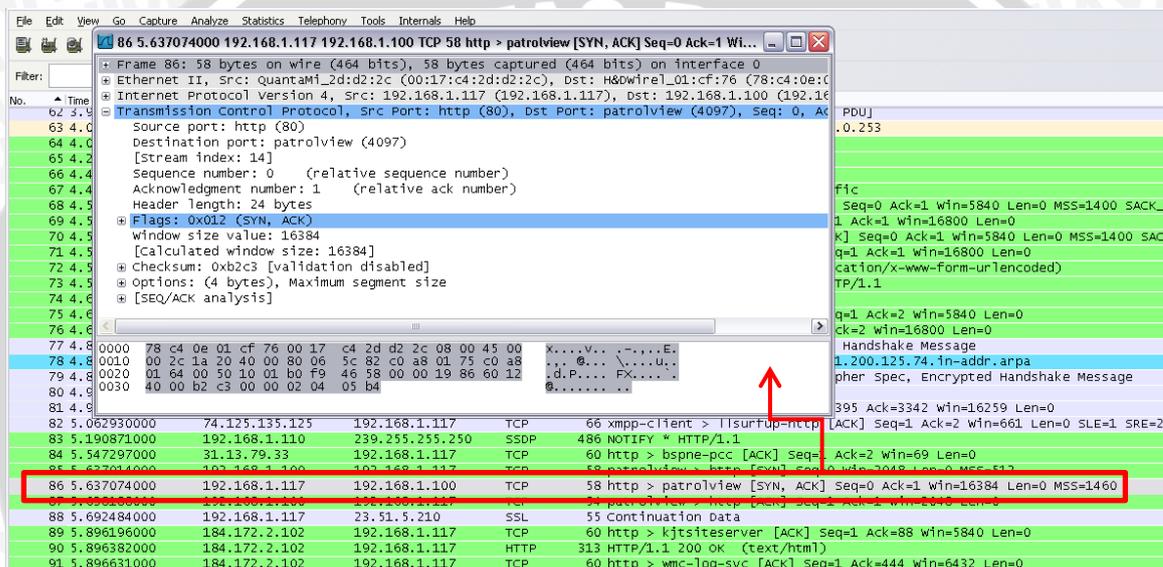


Gambar 6.15 Hasil Capture TCP Komunikasi Data Sensor Node dan Data Sink
Sumber: [Pengujian dan Analisis]

Berdasarkan Gambar 6.15 Hasil *Capture* TCP Komunikasi Data Sensor Node dan Data Sink terjadi proses respond dari request yang berasal dari dari sensor node (192.168.1.100) ke sink node (192.168.1.117) melalui port 4098 menuju port 80 dengan nomor sequence 0, dan acknowledge 1 (Acknowledge Request(1) + Len Request(0)) yang berarti request dari wifi shield sudah diterima oleh sink node secara baik.

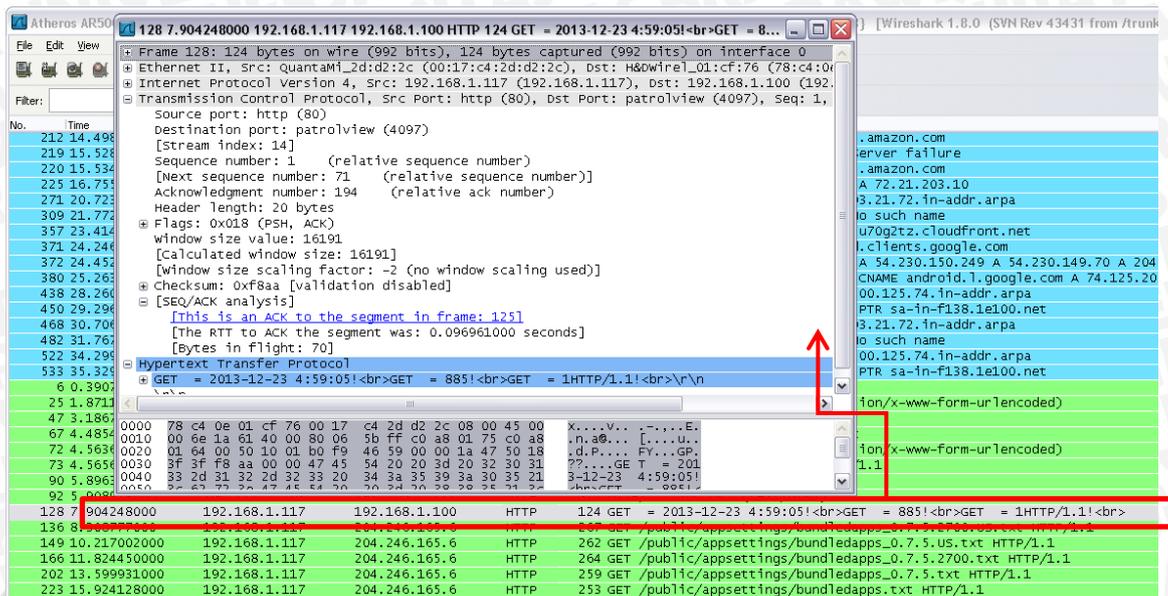


3. Pada gambar 6.16 Hasil Capture Komunikasi Data dari sink node dan wifi shield yang terjadi proses respond dari data sink (192.168.1.117) ke wifi shield (192.168.1.100) melalui port 80 menuju port 4097 dengan nomor sequence 0, dan acknowledge 1 (Acknowledge Request(1) + Len Request(0)) yang berarti request dari wifi shield sudah diterima oleh sink node secara baik dan sink node memberitahukan ke pada wifi shield. Hal ini diperjelas pada gambar 6.16 di bawah ini :



Gambar 6.16 Hasil Capture ACK TCP Komunikasi Data Wifi Shield dan Sink Node
 Sumber: [Pengujian dan Analisis]

4. Pada gambar 6.17 hasil capture komunikasi data sensor node dan data sink adalah proses pengiriman data yang terjadi setelah proses *handshaking connection* antara sensor node dengan data sink. Proses ini biasanya berulang kali terjadi berdasarkan dari besar data yang dikirimkan dari sensor node menuju data sink dan data sink akan melakukan acknowledgment dari setiap data yang dikirimkan oleh sensor node untuk mengecek apakah data yang diterima sesuai dengan data yang dikirimkan (*check sum*). Jika paket data yang dikirimkan terjadi kerusakan maka data sink akan mengirimkan permintaan kepada sensor node untuk mengirimkan ulang data yang rusak atau hilang tersebut.



Gambar 6.17 Hasil Capture Komunikasi Packet Data Unit Wifi Shield dan Sink Node
Sumber: [Pengujian dan Analisis]

Proses yang terjadi pada gambar 6.17 diatas adalah sebagai berikut :

IP wifi shield (192.168.1.100) mengirimkan response ke sink node (192.168.1.117) dengan menggunakan transport protocol TCP dari port 80 menuju port 4097. Informasi yang dikirimkan adalah TCP segment of a reassembled PDU yang berarti sensor node mengirimkan fragment data paket menuju sink node dengan besar fragmen sebesar 70 bytes untuk tiap fragmen data paket .

6.3.4 Analisa Pengujian Komunikasi Data TCP/IP

Dari analisa pengujian dapat dibuat tabel kasus uji pengujian dari komunikasi data TCP/IP

Tabel 6.5 Kasus Uji untuk Komunikasi Data TCP/IP

Nama Kasus Uji	Kasus Uji Komunikasi Data via TCP/IP
Objek Uji	Komunikasi data yang bekerja di layer transport
Tujuan Pengujian	Pengujian dilakukan untuk memastikan komunikasi data yang bekerja pada TCP/IP dapat bekerja dengan baik di sistem
Prosedur Uji	Pengujian dilakukan dengan menanamkan program

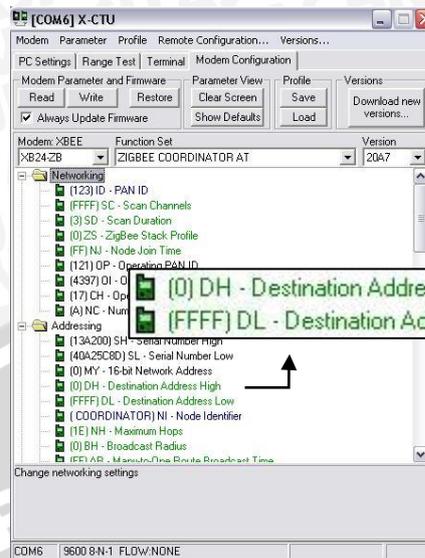
	yang mengfungsikan arduino sebagai web client. Inti dari program tersebut mengirimkan data serial yang diterima dari pengiriman sebelumnya untuk dikirimkan ke user di sink node. Data akan dikirimkan secara wireless melalui TCP/IP menuju database sink node
Hasil yang Diperoleh	Data yang dikirim melalui arduino wifi shield terkirim dan data tersebut dapat masuk database user dengan persentase keberhasilan 90% dari 10 sampel pengiriman

Sumber: [Pengujian dan Analisis]

6.4 Pengujian Integrasi

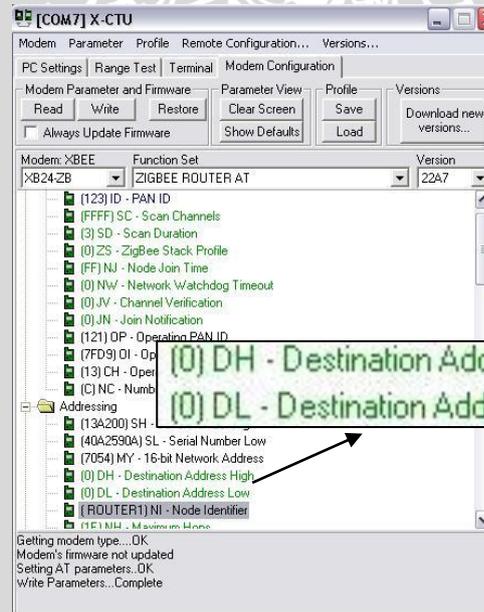
Pengujian integrasi adalah pengujian keseluruhan sistem, mulai dari subsistem *source node*, *middle node*, dan *sink node* diintegrasikan dalam sebuah kesatuan sistem. Topik yang dibahas dalam pengujian integrasi ini adalah menguji *relay node* yang ada pada *sink node*.

Pengujian integrasi dimulai dari subsistem *source node*. *Source node* memulai pengiriman secara *broadcast* melalui komunikasi *radio frequency* menuju node 1 dan node 2. Sebelum melakukan pengiriman data, XBee perlu dikonfigurasi terlebih dahulu untuk melakukan pengiriman data secara *broadcast*. Konfigurasinya hampir sama dengan konfigurasi komunikasi XBee secara *point-to-point*, perbedaannya terletak pada alamat Destination High dan Destination Low pada XBee coordinator (pengirim/*source node*), dimana di setting 0 untuk DH, dan FFFF untuk DL seperti pada gambar dibawah ini.



Gambar 6.18 Setting Broadcast pada Coordinator
Sumber:[Pengujian dan Analisis]

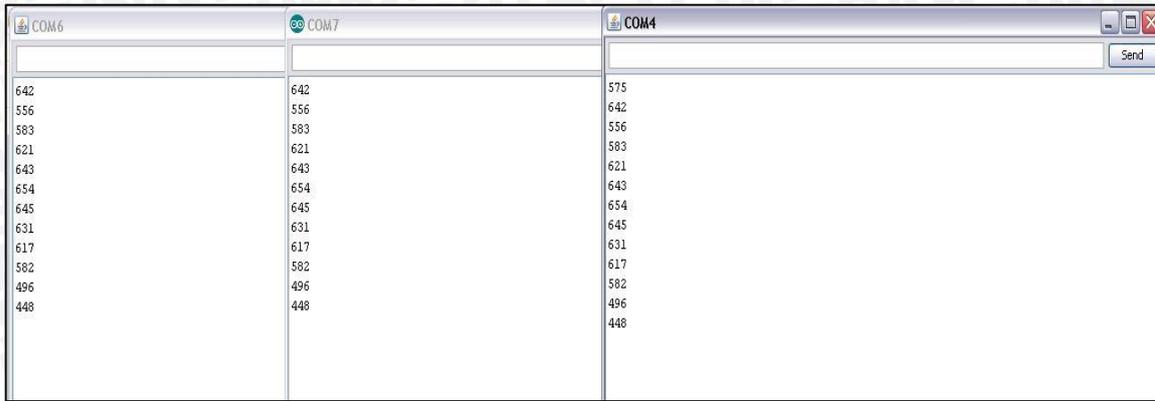
Sedangkan pada kedua XBee penerima (router), pada DH dan DL di konfigurasi dengan memberi value 0 seperti pada gambar dibawah ini



Gambar 6.19 Setting Broadcast pada Router
Sumber:[Pengujian dan Analisis]

Setelah melakukan konfigurasi, pengujian dilanjutkan dengan pengiriman data analog pin yang diterima di kedua XBee penerima. Berikut merupakan hasil transmit data secara broadcast menuju dua XBee yang ada pada sink node.





Gambar 6.20 Hasil Koneksi Broadcast Tiga XBee
Sumber:[Pengujian dan Analisis]

Pada gambar 6.20 diatas dapat dilihat bahwa data dapat diterima pada kedua XBee penerima. Ini menandakan data sudah berhasil dikirim secara broadcast ke dua node. Data tersebut nantinya akan diakses dan wifi shield akan meneruskan datanya melalui komunikasi TCP/IP. Node 1 dan node 2 akan melakukan inialisasi IP pada access point agar IPnya dapat mengakses *sink node*.

Data dari node 1 awalnya akan dikirim secara periodik menuju database *sink node* tiap 10 detik. Data tersebut akan ditampilkan pada antarmuka di sink node menggunakan halaman PHP.

+ Options		waktu	sensor	id	auto
<input type="checkbox"/>		2014-01-22 22:05:39	479	1	1
<input type="checkbox"/>		2014-01-22 22:05:49	914	1	2
<input type="checkbox"/>		2014-01-22 22:05:59	430	1	3
<input type="checkbox"/>		2014-01-22 22:06:09	938	1	4
<input type="checkbox"/>		2014-01-22 22:06:20	352	1	5
<input type="checkbox"/>		2014-01-22 22:06:30	1	1	6
<input type="checkbox"/>		2014-01-22 22:06:40	414	1	7
<input type="checkbox"/>		2014-01-22 22:06:50	585	1	8
<input type="checkbox"/>		2014-01-22 22:07:00	519	1	9
<input type="checkbox"/>		2014-01-22 22:07:10	586	1	10
<input type="checkbox"/>		2014-01-22 22:07:20	257	1	11
<input type="checkbox"/>		2014-01-22 22:07:31	73	1	12
<input type="checkbox"/>		2014-01-22 22:07:41	47	1	13
<input type="checkbox"/>		2014-01-22 22:07:51	662	1	14
<input type="checkbox"/>		2014-01-22 22:08:02	706	1	15
<input type="checkbox"/>		2014-01-22 22:08:12	164	1	16
<input type="checkbox"/>		2014-01-22 22:08:22	829	1	17
<input type="checkbox"/>		2014-01-22 22:08:32	880	1	18

Gambar 6.21 Data Masuk ke Database beserta ID dari Arduino
Sumber:[Pengujian dan Analisis]





Gambar 6.22 Tampilan PHP Interface Sink Node
Sumber:[Pengujian dan Analisis]

Pengujian selanjutnya mencoba menonaktifkan node 1 pada sistem. Seperti pada penjelasan di bab perancangan sebelumnya, selama data dari node 1 masuk halaman php membuat suatu file txt yang berisi string “1” sebagai status yang nantinya diakses oleh node 2. Apabila data berhenti dikirim maka file txt tersebut akan ditulis ulang dengan string “0”. Node 2 pada awal diaktifkan akan mengakses file txt ini. Didalamnya terdapat kondisi apabila isi string “0” maka data dari sink node mulai dikirim, sedangkan apabila isi string masih ”1” maka node 2 akan terus menerus mengakses file txt yang ada di *directory server*. Sehingga data yang dikirim dari source node tidak akan terputus walaupun pada jalurnya jaringan sensornya terdapat node yang mati

Tabel 6.6 Kasus Uji Pengiriman Relay Node

Status Node 1	Status Node 2
Node 1 aktif, Pengiriman dari node 1	Node 2 menerima status string ‘1’
Node 1 aktif, Pengiriman dari node 1	Node 2 menerima status string ‘1’
Node 1 aktif, Pengiriman dari node 1	Node 2 menerima status string ‘1’
Node 1 aktif, Pengiriman dari node 1	Node 2 menerima status string ‘1’
Node 1 aktif, Pengiriman dari node 1	Node 2 menerima status string ‘1’
Node 1 mati	Node 2 menerima status string ‘0’, Pengiriman dari node 2
Node 1 mati	Node 2 menerima status string ‘0’, Pengiriman dari node 2
Node 1 mati	Node 2 menerima status string ‘0’, Pengiriman dari node 2
Node 1 mati	Node 2 menerima status string ‘0’, Pengiriman dari node 2
Node 1 mati	Node 2 menerima status string ‘0’, Pengiriman dari node 2

Persentase Keberhasilan: 100%

Sumber: [Pengujian dan Analisis]



6.4.1 Analisa Pengujian Relay Node

Dari analisa pengujian dapat dibuat tabel kasus uji pengujian dari uji pengiriman relay node. Pada tabel uji relay node diatas terlihat terdapat keadaan dimana saat node 1 dan node 2 aktif maka yang menjadi prioritas adalah pengiriman dari ndoe 1. Kemudian saat node 1 mati maka node 2 yang melakukan request file txt di server akan mulai melakukan pengiriman data.

Tabel 6.7 Kasus Uji untuk Pengiriman Relay Node

Nama Kasus Uji	Kasus Uji Pengiriman Relay Node
Objek Uji	Pergantian jalur komunikasi data yang masuk ke <i>sink node</i>
Tujuan Pengujian	Pengujian dilakukan untuk memastikan pengiriman data dari node 1 tidak terputus dan bekerja dengan baik di sistem
Prosedur Uji	Pengujian dilakukan dengan menanamkan program php pada <i>sink node</i> . Node 1 akan terus mengirim data tiap 10 detik. Pada saat penerimaan data dari node 1, halaman php akan membuat file txt dengan isi string "1" sebagai status yang nantinya diakses oleh node 2. Saat pengiriman berhenti, status pada file txt berubah menjadi "0". Node 2 yang aktif akan terus menerus mengakses file txt yang ada di <i>sink node</i> . Saat status berubah menjadi 0, maka pengiriman selanjutnya dimulai dari node 2
Hasil yang Diperoleh	Data yang dikirimkan dari source node tidak terputus karena pada saat node 1 tidak aktif, node 2 langsung memulai pengiriman data. Hasil ini memperoleh persentase keberhasilan 100% dari 10 sampel pengiriman

Sumber: [Pengujian dan Analisis]