

**LAPORAN PROYEK AKHIR**

**PERBANDINGAN KINERJA ANTARA MODUL XBEE S2 DENGAN MODUL ESP8266 (12E) PADA SISTEM PEMANTAU KUALITAS UDARA**

***PERFORMANCE COMPARISON BETWEEN XBEE S2 MODULE AND ESP8266 (ESP-12E) MODULE ON AIR QUALITY MONITORING SYSTEM***



**Disusun oleh:**

**Ricky Hastomo Erwanto**

13/351005/SV/04069

**PROGRAM SARJANA TERAPAN TEKNOLOGI REKAYASA INTERNET**

**DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA**

**SEKOLAH VOKASI**

**UNIVERSITAS GADJAH MADA**

**YOGYAKARTA**

**2018**



**PERBANDINGAN KINERJA ANTARA MODUL XBEE S2 DENGAN MODUL ESP8266 (12E) PADA SISTEM PEMANTAU KUALITAS UDARA**

RICKY HASTOMO E, Muhammad Arrofiq, S.T., M.T., P.hD

Universitas Gadjah Mada, 2018 | Diunduh dari <http://etd.repository.ugm.ac.id/>

**PERBANDINGAN KINERJA ANTARA MODUL XBEE S2 DENGAN MODUL ESP8266 (12E) PADA SISTEM PEMANTAU KUALITAS UDARA**

**Proyek akhir  
Program Studi Teknologi Rekayasa Internet**

**Diajukan kepada  
Departemen Teknik Elektro dan Informatika  
Sekolah Vokasi Universitas Gadjah Mada  
Sebagai syarat kelengkapan studi jenjang Sarjana Terapan (D-IV)  
Dalam memperoleh derajat Sarjana Terapan Teknologi Rekayasa Internet pada  
Program Studi Teknologi Rekayasa Internet**

**Oleh:  
Ricky Hastomo Erwanto  
13/351005/SV/04069**

**PROGRAM SARJANA TERAPAN TEKNOLOGI REKAYASA INTERNET  
DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA  
SEKOLAH VOKASI  
UNIVERSITAS GADJAH MADA  
YOGYAKARTA  
2018**

Judul : PERBANDINGAN KINERJA ANTARA MODUL XBEE S2 DENGAN  
MODUL ESP8266(12-E) PADA SISTEM PEMANTAUAN KUALITAS  
UDARA  
Nama : Ricky Hastomo Erwanto  
Program Studi : Teknologi Rekayasa Internet  
Pembimbing : Muhammad Arrofiq, S.T., M.T., Ph.D.  
Waktu Ujian : Kamis, 19 Juli 2018, Pukul 07.30 – 09.30, di Ruang 230

---

---

**Telah dipertanggungjawabkan dan diuji oleh tim penguji serta disetujui dan disahkan sebagai syarat kelengkapan Sarjana Terapan ( S.Tr) Program Studi Teknologi Rekayasa Internet Sekolah Vokasi Universitas Gadjah Mada**

**Yogyakarta, 20 Juli 2018  
Diterima dan disetujui oleh,**

**Ketua Penguji**

**Sekretaris Penguji**

Hidayat Nur Isnianto, S.T., M.Eng.

**NIP. 197305282002121001**

Ir. Sri Lestari, M.T.

**NIP. 195908281986022001**

**Anggota Penguji / Dosen Pembimbing**

Muhammad Arrofiq, S.T., M.T., P.hD

**NIP. 197311271999031001**

**Mengetahui,**

**Ketua Departemen  
Teknik Elektro dan Informatika**

**Ketua Program Studi  
Teknologi Rekayasa Internet**

Nur Rohman Rosyid, S.T., M.T., D.Eng.

**NIU. 1120120075**

Muhammad Arrofiq, S.T., M.T., P.hD.

**NIP. 197311271999031001**

## **HALAMAN PERNYATAAN**

Dengan ini, saya Ricky Hastomo Erwanto, menyatakan bahwa proyek akhir ini adalah asli karya sendiri dan belum pernah diajukan sebagai pemenuhan persyaratan untuk memperoleh gelar kesarjanaan Diploma-Empat (D-IV) dari Universitas Gadjah Mada maupun perguruan tinggi lainnya.

Semua informasi yang dimuat dalam proyek akhir ini yang berasal dari karya orang lain, baik yang dipublikasikan atau tidak, telah diberikan penghargaan dengan mengutip nama sumber penulis secara benar dan semua isi dari proyek ini sepenuhnya menjadi tanggung jawab penulis.

Yogyakarta, 20 Juni 2018

Ricky Hastomo Erwanto

13/351005/SV/04069

Puji syukur penulis panjatkan kehadirat Allah SWT karena berkat Rahmat dan Karunia-Nya penulis dapat menyelesaikan Proyek Akhir yang berjudul “Perbandingan Kinerja Antara Modul Xbee S2 Dengan Modul ESP8266(12-E) Pada Sistem Pemantauan Udara” dengan baik.

Proyek akhir ini disusun sebagai syarat untuk mendapatkan gelar Sarjana Terapan (D-IV) Program Studi Teknologi Rekayasa Internet Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada. Untuk itu, perkenankanlah penulis menyampaikan ucapan terimakasih kepada :

1. Ibu, Bude, serta Saudara yang selalu mendoakan dan membantu dalam segalanya.
2. Bapak Nur Rohman Rosyid, S.T., M.T., D.Eng. selaku dosen wali dan Ketua Departemen Teknik Elektro dan Informatika (TEDI) yang telah memberikan bimbingan selama penulis mengikuti masa studi di Program Studi D4 Teknologi Rekayasa Internet Sekolah Vokasi Universitas Gadjah Mada.
3. Bapak Muhammad Arrofiq, S.T., M.T., P.hD selaku dosen pembimbing yang telah memberikan arahan dan dukungan selama penulis menyelesaikan Proyek akhir.
4. Teman-teman TJ Angkatan 2013 yang selalu memberikan semangat, pembelajaran dan dukungan selama masa kuliah hingga akhir kuliah.
5. Seluruh dosen Teknologi Jaringan yang memberikan ilmu pembelajaran dan dukungan selama masa kuliah hingga akhir kuliah.
6. Saudara Amar Fauzi yang telah memberikan bantuan kepada penulis dalam menyelesaikan Proyek akhir.
7. Berbagai pihak yang tidak bisa disebutkan satu persatu yang telah membantu dalam penyelesaian Proyek akhir ini.

Penulis menyadari dalam Proyek akhir ini masih terdapat banyak kekurangan dan kelemahan. Oleh karena itu penulis mengharapkan masukan berupa kiritik dan saran yang



UNIVERSITAS  
GADJAH MADA  
membangun

**PERBANDINGAN KINERJA ANTARA MODUL XBEE S2 DENGAN MODUL ESP8266 (12E) PADA  
SISTEM PEMANTAU KUALITAS  
UDARA**

RICKY HASTOMO E, Muhammad Arrofiq, S.T., M.T., P.hD

Universitas Gadjah Mada, 2018 | Diunduh dari <http://etd.repository.ugm.ac.id/>

untuk memperbaiki setiap kekurangan yang ada sehingga kedepannya akan lebih

baik lagi. Sebuah harapan, semoga proyek akhir ini bermanfaat bagi diri penulis, pembaca dan

semua pihak yang berkepentingan dengan proyek akhir ini.

Yogyakarta, 1 Maret 2018

Penulis

HALAMAN PENGESAHAN.....	iii
HALAMAN PERNYATAAN.....	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL .....	xi
INTISARI .....	xii
ABSTRACT .....	xiii
BAB I.....	1
1.1. Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan.....	3
BAB II.....	5
2.1 Jaringan Sensor Nirkabel.....	5
2.2 Modul <i>Wireless</i> Pada JSN .....	6
2.3 Sistem pemantauan kualitas udara .....	7
2.4 <i>Quality of Service ( QoS)</i> .....	7
a. <i>Throughput</i> .....	8
b. <i>Delay</i> .....	8
2.5 Pengujian kinerja modul <i>wireless</i> .....	9
2.6 Hipotesis.....	12
BAB III.....	13
3.1. Bahan .....	13
3.2. Peralatan .....	17
3.3.1. Metode Penelitian.....	18
3.3.2. Implementasi Sistem Pengujian.....	18
3.4. Perancangan Sistem.....	19
3.4.1. Perancangan Jaringan Sensor Nirkabel .....	20
3.5. Metode Pengujian .....	31
3.5.1 Topologi Jaringan .....	31
3.5.2 Skenario Pengujian .....	32
3.5.3 Penghitungan <i>Delay</i> .....	32



UNIVERSITAS  
GADJAH MADA

**PERBANDINGAN KINERJA ANTARA MODUL XBEE S2 DENGAN MODUL ESP8266 (12E) PADA  
SISTEM PEMANTAU KUALITAS  
UDARA**

RICKY HASTOMO E, Muhammad Arrofiq, S.T., M.T., P.hD

Universitas Gadjah Mada, 2018 | Diunduh dari <http://etd.repository.ugm.ac.id/>

<b>3.5.4</b>	<b>Penghitungan <i>Throughput</i></b> .....	34
<b>3.5.5</b>	<b>Penghitungan <i>Packet loss</i></b> .....	35
<b>BAB IV</b>	.....	36
<b>4.1.</b>	<b>Pengujian Sistem</b> .....	37
<b>4.2.</b>	<b>Pengujian <i>Delay</i></b> .....	38
<b>4.3.</b>	<b>Pengujian <i>Throughput</i></b> .....	40
<b>4.4.</b>	<b>Pengujian <i>Packet loss</i></b> .....	42
<b>4.5.</b>	<b>Pengujian pengiriman data</b> .....	44
<b>BAB V</b>	.....	46
<b>5.1.</b>	<b>Kesimpulan</b> .....	46
<b>5.2.</b>	<b>Saran</b> .....	46
<b>DAFTAR PUSTAKA</b>	.....	47

Gambar 2. 1	Arsitektur Jaringan Sensor Nirkabel signifikan.....	5
Gambar 3. 1	Tampilan Aplikasi Arduino Ide .....	14
Gambar 3. 2	Tampilan Aplikasi X-CTU.....	14
Gambar 3. 3	Tampilan Aplikasi Xampp Control Panel.....	15
Gambar 3. 4	Tampilan Aplikasi Wireshark .....	16
Gambar 3. 5	Tampilan Aplikasi Docklight.....	16
Gambar 3. 6	flowchart penelitian.....	18
Gambar 3. 7	Perancangan Topologi Jaringan Xbee S2 .....	19
Gambar 3. 8	Perancangan Topologi Jaringan ESP8266 .....	19
Gambar 3. 9	Rangkain modul ESP8266 dan sensor mq 135 .....	20
Gambar 3. 10	Konfigurasi Wemos pada Arduino IDE.....	21
Gambar 3. 11	Rangkain modul esp8266 dan sensor mq 135.....	22
Gambar 3. 12	Tampilan awal X-CTU.....	23
Gambar 3. 13	Konfigurasi Port pada Xbee .....	23
Gambar 3. 14	Konfigurasi baud rate pada Xbee.....	24
Gambar 3. 15	Konfigurasi Xbee .....	25
Gambar 3. 16	Konfigurasi Xbee menjadi coordinator dan end device.....	25
Gambar 3. 17	Konfigurasi PAN ID pada Xbee .....	26
Gambar 3. 18	Pengetesan komunikasi Xbee.....	26
Gambar 3. 19	Konfigurasi Arduino pada Xbee .....	27
Gambar 3. 20	Pengaturan xampp.....	27
Gambar 3. 21	Membuat Tabel di Database .....	28
Gambar 3. 22	koneksi ke database untuk modul Xbee.....	29
Gambar 3. 23	koneksi ke database untuk modul ESP8266 .....	29
Gambar 3. 24	Tampilan Website .....	30
Gambar 3. 25	Topologi pengujian QoS .....	31
Gambar 3. 26	Delay Xbee.....	33
Gambar 3. 27	Delay ESP8266 .....	33
Gambar 3. 28	Throughput ESP8266.....	34
Gambar 3. 29	Data pada database.....	35
Gambar 4.1	Tampilan Website .....	36
Gambar 4.2	Tes ping modul ESP8266.....	37
Gambar 4.3	Tes ping modul ESP8266.....	37
Gambar 4.4	Tes pengiriman data Xbee S2 .....	38
Gambar 4.5	Perbandingan <i>Nilai Delay Xbee s2 dan ESP8266</i> .....	39
Gambar 4.6	Pengiriman data ESP8266 pada Wireshark .....	40
Gambar 4.7	Perbandingan <i>Nilai Throughput Xbee s2 dan ESP8266</i> .....	41
Gambar 4.8	Besar data Xbee .....	42
Gambar 4.9	Besar data ESP8266 .....	42



**PERBANDINGAN KINERJA ANTARA MODUL XBEE S2 DENGAN MODUL ESP8266 (12E) PADA SISTEM PEMANTAU KUALITAS UDARA**

RICKY HASTOMO E, Muhammad Arrofiq, S.T., M.T., P.hD

Universitas Gadjah Mada, 2018 | Diunduh dari <http://etd.repository.ugm.ac.id/>

UNIVERSITAS  
GADJAH MADA

Gambar 4.10 Perbandingan Nilai <i>Packet loss</i> Xbee s2 dan ESP8266.....	43
Gambar 4.11 jumlah data yang dikirim ESP8266 dan Xbee .....	44
Gambar 4.12 Jumlah data yang dikirim ESP8266 dan Xbee.....	45



**DAFTAR TABEL**

Tabel 2. 1 Throughput.....	8
Tabel 2. 2 Delay.....	8
Tabel 2. 3 Packet loss.....	9
Tabel 2. 4 Ringkasan Penelitian.....	10
Tabel 2. 5 Ringkasan Penelitian (lanjutan).....	11
Tabel 3. 1 Spesifikasi Laptop Server .....	17
Tabel 3. 2 Spesifikasi Laptop untuk pengambilan data .....	17
Tabel 3. 3 Arduino uno R3 .....	17
Tabel 3. 4 Spesifikasi Xbee S2 .....	17
Tabel 3. 5 Spesifikasi Wemos D1 .....	17

## **PERBANDINGAN KINERJA ANTARA MODUL XBEE S2 DENGAN MODUL ESP8266(12E) PADA SISTEM PEMANTAU KUALITAS UDARA**

Perkembangan teknologi jaringan sensor nirkabel (JSN) mengalami kemajuan yang sangat pesat pada bekangan ini. Perkembangan tersebut juga memicu berkembangnya modul *wireless* yang digunakan dalam JSN. Xbee dan ESP8266 merupakan modul *wireless* yang sering digunakan untuk merancang jaringan sensor nirkabel. Xbee menggunakan standar IEEE 802.15.4 sedangkan ESP8266 menggunakan standar IEEE 802.11 b/g/n. Dalam penelitian ini akan dilakukan perbandingan modul *wireless* antara Xbee dan ESP8266 untuk menguji kinerja dari modul *wireless* tersebut. Perbandingan dilakukan dengan cara menguji nilai *throughput*, *delay packet*, dan *packet lost*. Pengujian akan diterapkan pada sistem pemantauan kualitas udara dengan menggunakan sensor MQ135. Hasil dari penelitian merupakan perbandingan nilai paramater yang diuji dari kedua modul *wireless* tersebut. Berdasarkan hasil pengujian sistem pemantau kualiaas udara dapat dijadikan sumber data pengujian kinerja modul Xbee S2 dan ESP8266(12E). Dari hasil pengujian, nilai *delay* dengan modul Xbee lebih kecil dari pada modul ESP8266. Nilai *throughput* pengiriman data dengan modul ESP8266 lebih besar dari pada modul Xbee. Parameter *packet loss* pengiriman data dengan modul ESP8266 dan Xbee dalam kategori bagus karena nilai *packet loss* tidak lebih dari 3%.

Kata Kunci : Xbee, JSN, ESP8266, modul *wireless*, MQ135

**PERFORMANCE COMPARISON BETWEEN XBEE S2 MODULE AND ESP8266 (12E)  
MODULE ON AIR QUALITY MONITORING SYSTEM**

**838/5000**

*The development of wireless sensor network (WSN) technology has progressed very rapidly. These developments also triggered the development of wireless modules used in WSN. Xbee and ESP8266 are wireless modules that are often used to design wireless sensor networks. Xbee uses the IEEE 802.15.4 standard while ESP8266 uses the IEEE 802.11 b / g / n standard. In this research will be comparison of wireless modules between Xbee and ESP8266 to test the performance of the wireless module. Comparison is performed by testing the value of throughput, delay packet, and packet lost. The test will be applied to the air quality monitoring system using the MQ135 sensor. Air quality monitoring system can be used as data source for performance testing of Xbee S2 and ESP8266 (12E) modules. Delay parameter with Xbee module is smaller than the ESP8266 module. Parameter Throughput with ESP8266 module is bigger than Xbee module. Packet loss parameters with ESP8266 and Xbee modules in good condition because packet loss value is not more than 3%.*

**Keyword:** Xbee, WSN, ESP8266, module wireless, MQ135

## PENDAHULUAN

### 1.1. Latar Belakang

Perkembangan teknologi jaringan sensor nirkabel ( JSN) mengalami kemajuan yang pesat, hal ini dikarenakan dengan menggunakan teknologi nirkabel pada sensor mempunyai banyak keuntungan seperti praktis, sensor bersifat *mobile*, meningkatkan efisiensi secara operasional, dan dapat mengumpulkan data dalam jumlah besar (Dini, et al., 2011). Oleh karena itu, aplikasi JSN sudah banyak digunakan dalam berbagai sistem sensor seperti sensor kebakaran, sensor bencana alam, sistem deteksi tempat parkir dan sebagainya.

JSN merupakan teknologi nirkabel yang terdiri dari kumpulan *node* sensor yang ditempatkan di suatu area tertentu. Teknologi nirkabel yang diterapkan dalam JSN antara lain modul Xbee dan ESP8266. Dua modul ini menggunakan standar yang berbeda, Xbee menggunakan standar Zigbee 802.15.4 sedangkan ESP8266 menggunakan standar *WiFi* 802.11 b/g/n. Pada penelitian (Thaker, 2016) terdapat tabel perbandingan Protokol Zigbee dan *WiFi* disajikan pada Tabel 1.1.

Tabel 1. 1 Perbandingan Protokol Zigbee dan WiFi

	<b>Zigbee</b>	<b>WiFi</b>
<b><i>Range</i></b>	10-100 meter	50-100 meter
<b><i>Data rate</i></b>	20,40 & 250 Kbps	11 & 54 Mbps
<b><i>Topology</i></b>	Ad-hoc, peer to peer,star, mesh	Peer to peer
<b><i>Frekuensi</i></b>	2.4 GHz	2.4-5 GHz
<b><i>Complexity</i></b>	Low	High
<b><i>Power Consumption</i></b>	Very Low	High

Dari data perbandingan tersebut menunjukkan kedua modul *wireless* mempunyai

kelebihan dan kekurangan masing-masing. Dalam penelitian ini akan dilakukan perbandingan kinerja dari modul *wireless* yang menggunakan standar Zigbee dan *WiFi* yaitu Xbee S2 dan ESP8266 pada sistem pemantau kualitas udara yang menggunakan sensor MQ135. Nilai parameter yang akan diuji adalah *throughput*, *packet loss* dan *delay*. Sehingga dari hasil pengujian kinerja kedua modul *wireless* tersebut dapat diketahui manakah modul *wireless* yang lebih baik dalam perancangan jaringan sensor nirkabel.

## 1.2 Rumusan Masalah

Beberapa rumusan masalah yang mengacu pada tujuan penulisan Proyek akhir ini adalah sebagai berikut :

1. Bagaimana cara merancang sistem pemantau kualitas udara dengan menggunakan sensor MQ135?
2. Bagaimana implementasi modul Xbee S2 dan modul ESP8266 untuk pengiriman data pada jaringan sensor nirkabel ?
3. Bagaimana cara menguji kinerja modul Xbee S2 dan modul ESP8266 pada jaringan sensor nirkabel ?

## 1.3 Batasan Masalah

Beberapa batasan masalah pada Proyek akhir ini adalah sebagai berikut :

1. Analisis pengujian kinerja modul *wireless* adalah *throughput*, *packe loss* dan *delay*.
2. Pengambilan data pengujian menggunakan aplikasi *wireshark*, *XCTU*, dan *docklight*.
3. Pengambilan data pengujian berdasarkan jarak transmisi 5, 10, 15, 20 meter.
4. Pengambilan data pengujian modul ESP8266 menggunakan jaringan *Wi-Fi* pada laptop.

## 1.4 Tujuan Penelitian

Tujuan dari penelitian dalam proyek akhir ini adalah sebagai berikut :

1. Merancang jaringan sensor nirkabel pada sistem pemantau kualitas udara dengan menggunakan modul Xbee S2 dan ESP8266.
2. Menguji perbandingan kinerja antara modul Xbee S2 dengan modul ESP8266 pada sistem pemantau kualitas udara.

## 1.5 Manfaat Penelitian

Manfaat yang dapat diambil dari penelitian dan pengerjaan proyek akhir ini adalah sebagai berikut :

1. Memberikan gambaran untuk merancang jaringan sensor nirkabel menggunakan modul Xbee dan Esp8266.
2. Memberikan referensi dalam pemilihan modul *wireless* untuk perancangan jaringan sensor nirkabel.
3. Hasil perbandingan *throughput*, *packet loss* dan *delay* data dari modul Xbee dan modul ESP8266.

## 1.6 Sistematika Penulisan

Untuk menggambarkan secara menyeluruh mengenai masalah yang akan dibahas dalam laporan proyek akhir ini, maka dibuat sistematika penulisan yang terbagi dalam lima bab, BAB I, PENDAHULUAN, memuat latar belakang masalah, perumusan masalah, batasan masalah, tujuan penulisan, kegunaan penulisan, metode penelitian dan sistematika penulisan.

BAB II, TINJAUAN PUSTAKA, memuat uraian sistematis tentang informasi yang relevan dan mutakhir yang terkait dengan lingkup materi penelitian atau teknologi yang akan diterapkan. Uraian dalam tinjauan pustaka ini selanjutnya menjadi dasar teori yang



**PERBANDINGAN KINERJA ANTARA MODUL XBEE S2 DENGAN MODUL ESP8266 (12E) PADA SISTEM PEMANTAU KUALITAS UDARA**

RICKY HASTOMO E, Muhammad Arrofiq, S.T., M.T., P.hD

Universitas Gadjah Mada, 2018 | Diunduh dari <http://etd.repository.ugm.ac.id/>

digunakan oleh penulis dalam melaksanakan penelitian dan menyajikan argumentasi

dalam pembahasan hasil penelitian.

BAB III, BAHAN DAN METODE, Penelitian memuat bahan, peralatan, tahapan penelitian, dan sistem serta analisis data yang ada pada penelitian ini.

BAB IV, ANALISIS HASIL DAN PEMBAHASAN, memuat semua temuan ilmiah yang diperoleh sebagai data hasil penelitian, atau hasil unjuk kerja prototipe yang dibuat. Pada bagian ini peneliti menyusun secara sistematis disertai argumentasi yang rasional tentang hasil unjuk kerja yang diperoleh dari hasil penelitian.

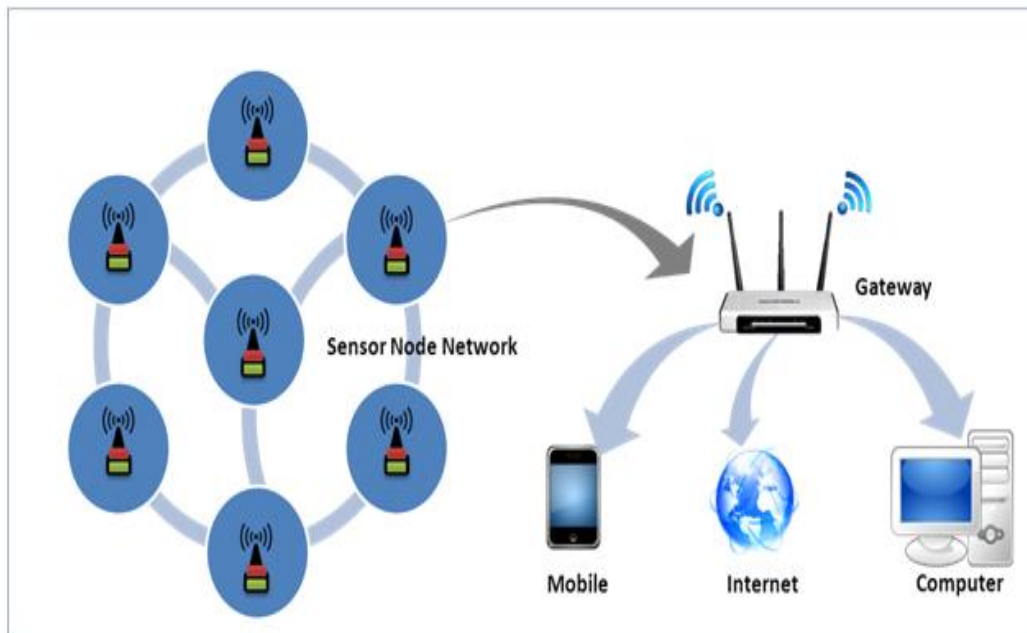
BAB V, PENUTUP, memuat kesimpulan serta saran dari penelitian proyek akhir ini.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Jaringan Sensor Nirkabel

Jaringan nirkabel telah banyak digunakan untuk membangun sensor, yang disebut jaringan sensor nirkabel (JSN) atau bisanya disebut dengan *wireless sensor network* (WSN). jaringan sensor nirkabel (JSN) terdiri dari sejumlah *node* sensor yang dilengkapi dengan modul *wireless* yang bekerja sama untuk memantau kondisi lingkungan tertentu. *Node* Sensor berkomunikasi dan mengirim data melalui jaringan nirkabel. JSN dapat membentuk jaringan yang efisien dan kuat, sehingga meningkatkan kinerja secara signifikan (Jusak, 2013). Gambar 2.1 merupakan contoh Arsitektur Jaringan Sensor Nirkabel.



Gambar 2. 1 Arsitektur Jaringan Sensor Nirkabel signifikan.

Sekarang ini, Teknologi JSN dapat digunakan untuk memonitoring beberapa hal seperti suhu, kelembaban, kondisi cahaya, dan lain sebagainya. Sehingga dapat disimpulkan bahwa JSN didesain sebagai penghubung antara lingkungan fisik dan dunia digital

(Anggraini, et al., 2014). Perancangan teknologi JSN menggunakan modul *wireless* yang dijadikan media transmisi data, contohnya modul ESP8266 dan modul Xbee.

## 2.2 Modul *Wireless* Pada JSN

Dalam perancangan jaringan sensor nirkabel diperlukan modul *wireless* untuk mengirim data dari sensor ke server (pengolah data). Modul *wireless* yang biasa digunakan pada jaringan sensor nirkabel antara lain ESP8266 dan Xbee. ESP8266 merupakan modul *WiFi* yang memiliki standar IEEE 802.11 b/g/n dengan frekuensi 2.4 GHz, sedangkan Xbee merupakan modul zigbee yang memiliki standar IEEE 802.15.4 dengan frekuensi 2.4 GHz. Seiring berkembangnya modul *wireless*, banyak modul yang digunakan untuk merancang jaringan sensor nirkabel antara lain Modul ESP8266(12E) dan Modul Xbee S2.

Modul ESP8266 (12E) terkenal dengan biaya yang murah dengan kualitas yang baik. Modul ini terintegrasi dengan protokol TCP/IP yang memungkinkan untuk mengakses jaringan *WiFi* dari mikrokontroler yang digunakan (Thaker, 2016).

Pada perancangan teknologi JSN dengan modul ESP8266 yang telah dilakukan oleh Thaker dengan judul “*ESP8266 based Implementation of Wireless SensorNetwork with Linux Based Web-Server*” menerapkan modul *wireless* ESP8266 untuk merancang jaringan sensor dengan menggunakan linux sebagai web server. Modul ESP8266 digunakan untuk mengirim data yang terdeteksi dari *node* sensor ke web server dan juga untuk mengirim perintah dari user untuk mengontrol operasi *node*, dari hasil penelitian ini modul disimpulkan bahwa modul ESP8266 cocok untuk memonitor dan mengontrol jaringan sensor secara *real time* (Thaker, 2016).

Xbee S2 merupakan modul *wireless* yang dibuat berdasarkan protokol Zigbee. Modul ini berjalan pada data rate rendah dengan konsumsi daya yang rendah, sehingga cocok untuk menjadi transmisi data untuk jaringan sensor nirkabel. Perangkat Zigbee menggunakan

standar IEEE 802.15.4 dengan frekuensi 2,4 GHz dan kecepatan data 250 kbps (Thaker, 2016).

Pada perancangan teknologi JSN dengan modul Xbee yang telah dilakukan oleh Piyare dan Lee menganalisis kinerja dari modul Xbee ZB, dengan menguji nilai RSSI, *throughput* jaringan, *delay packet*, *mesh routing recovery time*, dan konsumsi energi. Hasil penelitian tersebut bahwa modul Xbee Zb lebih cocok untuk aplikasi dengan *data rate* rendah (Piyare & Lee, 2013).

Pada penelitian proyek akhir ini akan dilakukan pengujian kinerja antara modul Xbee S2 dengan modul esp8266 pada sistem pemantaun kualitas udara.

### 2.3 Sistem pemantauan kualitas udara

Sistem pemantuan kualitas udara merupakan sistem yang dapat memonitoring kualitas udara secara *real time* pada suatu area / tempat, sistem ini biasanya bekerja dengan sebuah sensor yang dapat mendeteksi kandungan udara yang buruk seperti CO<sub>2</sub>. Peralatan yang digunakan adalah arduino uno sebagai perangkat yang mengolah data dari sensor, dan sensor yang digunakan untuk mendekteksi kandungan gas adalah MQ135 (Nur, 2014).

Data yang kirim dari sensor akan dilakukan secara terus menerus untuk memperbarui nilai kandungan CO<sub>2</sub> pada area sensor, sehingga akan terjadi komunikasi data terus menerus antar server dan sensor. Komunikasi data tersebut dapat dijadikan sumber data untuk pengujian QoS pada kedua modul Wireless Xbee 2 dan ESP8266.

### 2.4 *Quality of Service ( QoS)*

Kemampuan suatu jaringan untuk menyediakan layanan yang baik dengan menyediakan bandwith, mengatasi jitter dan delay. QoS merupakan kualitas atau jaminan terhadap layanan yang diberikan kepada pengguna jaringan. Berikut parameter QoS yang akan dianalisis :

**a. Throughput**

*Throughput* yaitu kecepatan (*rate*) transfer data efektif, yaitu diukur dalam bps. *Throughput* merupakan jumlah total kedatangan paket yang sukses yang diamati pada tujuan selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut (Nasihin, et al., 2015).

Tabel 2. 1 *Throughput*

Kategori Throughput	Throughput
Sangat bagus	100 %
Bagus	75 %
Sedang	50 %
Jelek	<25%

**b. Delay**

*Delay* adalah waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. *Delay* dapat dipengaruhi oleh jarak, media fisik kongesti atau juga waktu proses lama (Nasihin, et al., 2015).

Tabel 2. 2 *Delay*

Kategori delay	delay
Sangat bagus	<150 ms
Bagus	150 – 300 ms
Sedang	300 – 450 ms
Jelek	>450 ms

**c. Packet loss**

*Packet loss* adalah suatu parameter yang menggambarkan suatu kondisi jumlah total paket yang hilang pada suatu jaringan (Nasihin, et al., 2015).

Tabel 2. 3 *Packet loss*

Kategori <i>Packet loss</i>	<i>Packet loss</i>
Sangat bagus	0%
Bagus	3 %
Sedang	15 %
Jelek	25 %

## 2.5 Pengujian kinerja modul *wireless*

Pengujian modul *wireless* dilakukan dengan cara membandingkan parameter – parameter yang ditentukan, yaitu adalah *throughput*, *packet loss* dan *delay*. Kedua modul *wireless* akan diuji untuk mendapat nilai parameter tersebut, sehingga dengan membandingkan parameter tersebut diharapkan dapat mengetahui modul *wireless* yang lebih baik yang digunakan pada sistem pemantauan kualitas udara.

Beberapa penelitian yang telah dilakukan pada perancangan jaringan sensor nirkabel banyak menggunakan modul ESP8266 dan Xbee. Pada penelitian yang telah dilakukan oleh Thaker dengan judul “*ESP8266 based Implementation of Wireless SensorNetwork with Linux Based Web-Server*” menerepakan modul *wireless* ESP8266 untuk merancang jaringan sensor dengan menggunakan linux sebagai web server. Modul ESP8266 digunakan untuk mengirim data yang terdeteksi dari *node* sensor ke web server dan juga untuk mengirim perintah dari user untuk mengontrol operasi *node*, dari hasil penelitian ini modul disimpulkan bahwa modul ESP8266 cocok untuk memonitor dan mengontrol jaringan sensor secara real time (Thaker, 2016).

Kedua modul *wireless* tersebut merupakan media transmisi yang sering dipakai dalam membangun jaringan sensor nirkabel (JSN). Banyak penelitian tentang analisis kinerja dari kedua modul tersebut. Penelitian yang telah dilakukan oleh Piyare dan Lee menganalisis kinerja dari modul Xbee ZB, dengan menguji nilai RSSI, *throughput* jaringan, *delay packet*,

*mesh routing recovery time*, dan konsumsi energi. Hasil penelitian tersebut bahwa modul Xbee Zb lebih cocok untuk aplikasi dengan *data rate* rendah (Piyare & Lee, 2013). Sedangkan pada penelitian yang dilakukan oleh Yuliansyah menganalisis kinerja pengiriman paket data Modul ESP8266, pengujian dilakukan dengan cara mengirim data dengan menggunakan tiga metode yaitu AT-Command, Protokol SLIP, dan NodeMCU (Yuliansyah, 2016).

Berdasarkan beberapa uraian hasil penelitian, perancangan jaringan sensor nirkabel banyak menggunakan modul Xbee dan ESP8266 sebagai media untuk mentransmisikan data dari sensor. Oleh karena itu, penelitian ini akan melakukan perbandingan kinerja dari modul kedua modul tersebut yang diterapkan pada sistem pemantauan kualitas udara. Pengujian dilakukan dengan cara membandingkan nilai *throughput*, *packet loss* dan *delay*.

Adapun ringkasan berdasarkan uraian penelitian disajikan pada Tabel 2.3 dan Tabel 2.4.

Tabel 2. 4 Ringkasan Penelitian

No	Judul penelitian	Pokok bahasan	Tipe	Tahun	Penulis	Hasil
1	<i>Performance Analysis of XBee ZB Module Based Wireless Sensor Networks</i>	Menganalisa kinerja Modul Xbee ZB berbasis <i>Wireless Sensor Network</i>	Analisis	2013	Rajeev Piyare, Seong ro Lee	XBee ZB lebih cocok untuk aplikasi dengan rate data rendah
2	Uji Kinerja Pengiriman Data Secara <i>Wireless</i> Menggunakan Modul <i>ESP8266</i> Berbasis <i>Rest Architecture</i>	Menganalisa kinerja pengiriman Data Modul <i>ESP8266</i> berbasis <i>Rest Architecture</i>	Analisis	2016	Harry Yuliansyah	Perbandingan metode – metode pengiriman data dengan Modul <i>ESP8266</i>

Tabel 2. 5 Ringkasan Penelitian (lanjutan)

No	Judul penelitian	Pokok bahasan	Tipe	Tahun	Penulis	Hasil
3	<i>ESP8266 based Implementation of Wireless Sensor Network with Linux Based Web-Server</i>	Merancang sistem <i>Wireless Sensor Network</i> berbasis <i>ESP8266</i> dengan Linux Web Server	Implementasi	2016	Tejas Thaker	ESP8266 lebih cocok untuk memonitor sensor secara <i>real time</i> dan untuk mengontrol jaringan sensor
4	<i>Performance Evaluation Of Ieee 802.15.4 Based Wireless Sensor Network For Monitoring Patients' Pulse Status</i>	Merancang dan menganalisa sistem JSN untuk menonitoring denyut nadi pasien	Analisis	2013	Puput Dani Prasetyo Adi, dkk	Besarnya <i>Throughput</i> ditentukan oleh Jarak, semakin jauh jarak transmisi data maka semakin kecil nilai <i>Throughput</i>
5	Prototype automatic exhaust fan control based arduino uno	Purwarupa Kendali Otomatis Kipas Pembuangan Berbasis Arduino Uno	Implementasi	2014	Aditya Deni Nur R.	Membuat kendali otomatis kipas pembuangan dengan menggunakan sensor MQ 135 berdasarkan kandungan CO2
6	Analisis Dan Simulasi Wireless Sensor Network (Wsn) Untuk Komunikasi Data Menggunakan Protokol Zigbee	Melakukan analisa dan simulasi topologi pada jaringan Sensor nirkabel menggunakan modul Xbee	Analisis	2014	Dwima Anggraini, dkk	Semakin banyak node yang di gunakan maka semakin besar nilai data <i>dropped</i> , <i>delay</i> , <i>packet loss</i>

## **2.6 Hipotesis**

Mengacu pada uraian-uraian sub bab terdahulu, seperti perumusan masalah, dan tinjaun pustaka, maka dapat dikemukakan hipotesisnya yaitu sebagai berikut :

1. Sistem pemantau kualitas udara dapat digunakan sebagai objek sumber data kinerja modul Xbee S2 dan ESP8266.
2. Hasil pengujian nilai parameter dapat dijadikan indikator kualitas kinerja dari kedua modul Xbee S2 dan ESP8266.

## BAB III

### BAHAN DAN METODE PENELITIAN

Pada bab ini dijelaskan proses perancangan perbandingan kinerja antara modul Xbee S2 dan ESP8266(12-E) pada sistem pemantauan udara yang terdiri dari perangkat dan bahan penunjang, metode penelitian, pemodelan sistem pengujian, dan pemodelan skenario pengukuran.

#### 3.1. Bahan

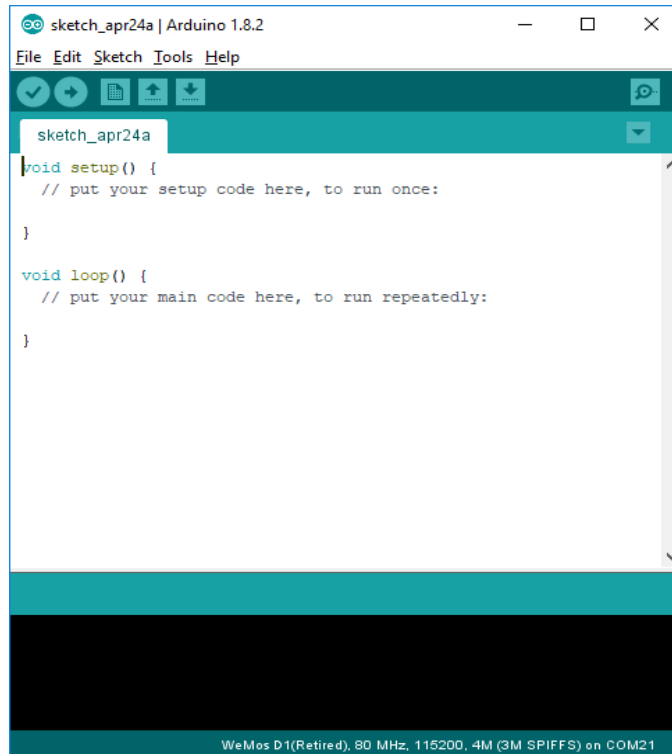
Adapun bahan yang dibutuhkan sebagai berikut:

a. Arduino IDE 1.8.2

Arduino IDE merupakan software yang digunakan untuk memprogram arduino.

Arduino menggunakan bahasa yang menyerupai bahasa C. Gambar 3.1 menunjukkan tampilan awal saat membuka aplikasi Arduino IDE. Suatu program arduino umumnya terdiri atas instruksi void setup () dan void loop (). Instruksi void setup () digunakan untuk mengisialisasi variabel-variabel yang akan digunakan, sedangkan Instruksi void loop () digunakan untuk menjalankan siklus program, yang akan dijalankan terus menerus.

Pada penelitian ini *software* Arduino IDE digunakan untuk memprogram arduino yang berfungsi untuk mengolah data dari sensor sebelum dikirimkan ke server. *Software* Arduino IDE juga digunakan untuk mengkonfigurasi Wemos D1 agar dapat berfungsi untuk mengolah data dari sensor dan mengkonfigurasi modul ESP8266 untuk terhubung ke jaringan *Wi-Fi* yang sama dengan PC server sehingga Wemos D1 dapat mengirimkan data dari sensor ke pc server untuk diolah.



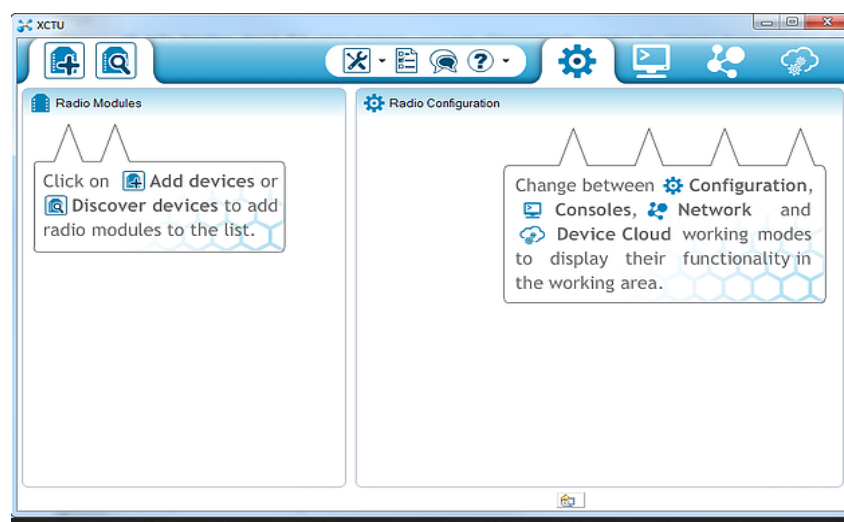
Gambar 3. 1 Tampilan Aplikasi Arduino Ide

b. X-CTU 6.3.5

X-CTU berfungsi untuk mengkonfigurasi Xbee yang dirancang oleh Digi.

Software ini dapat memudahkan pemasangan, konfigurasi dan untuk uji modul Xbee.

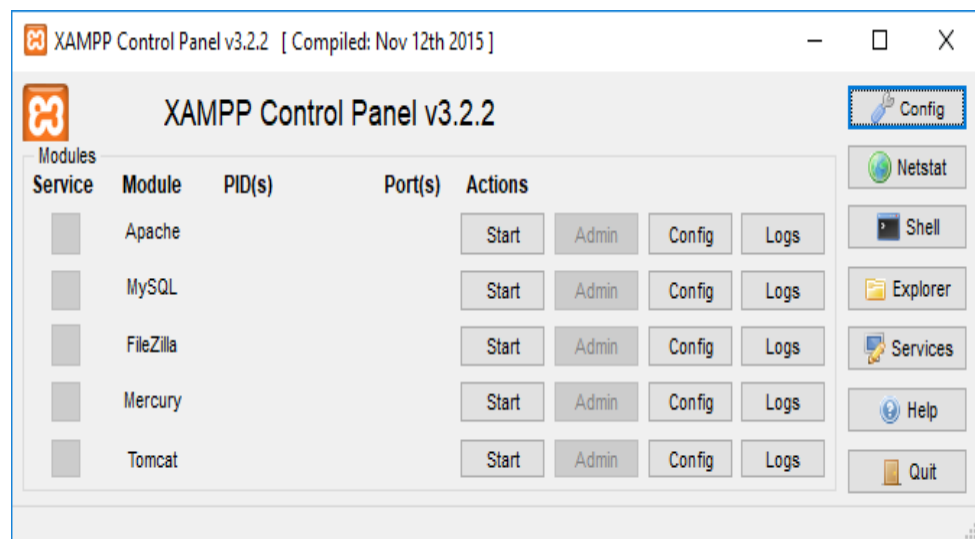
Gambar 3.2 menunjukkan tampilan awal saat membuka aplikasi X-CTU.



Gambar 3. 2 Tampilan Aplikasi X-CTU

c. Xampp Control Panel 3.2.2

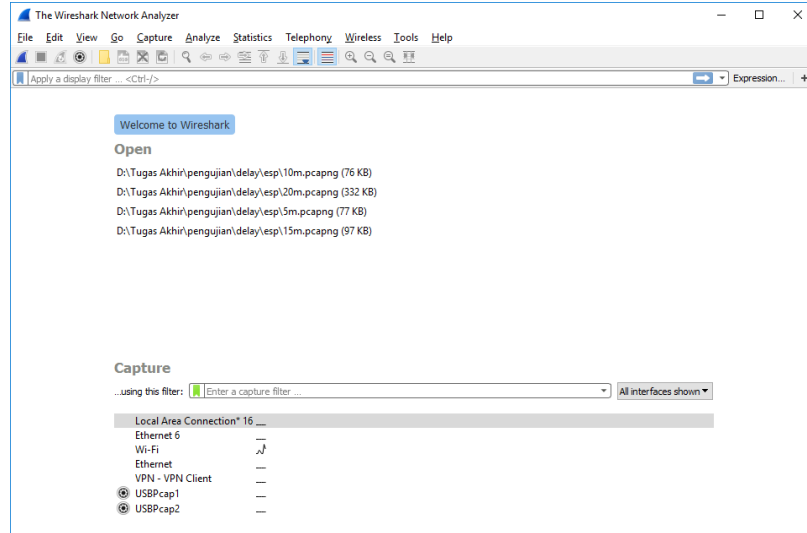
Xampp Control Panel berfungsi sebagai server yang berdiri sendiri (*localhost*) yang terdiri dari Apache HTTP server dan MySQL *Data base*. Gambar 3.3 menunjukkan tampilan awal saat membuka aplikasi Xampp Control Panel. Pada penelitian Xampp digunakan sebagai server untuk menyimpan dan mengolah data dari sensor untuk ditampilkan dalam aplikasi web.



Gambar 3. 3 Tampilan Aplikasi Xampp Control Panel.

d. Wireshark 2.4.5

Wireshark berfungsi untuk menganalisis transmisi paket data dalam jaringan, dalam penelitian ini digunakan untuk mengetahui nilai parameter *throughput* dan *delay* pada modul ESP8266. Gambar 3.4 menunjukkan tampilan awal saat membuka aplikasi Wireshark. Pc server dan modul ESP8266 akan dihungkan pada jaringan yang sama sehingga wireshark dapat melihat transmisi paket – paket antara modul ESP8266 dengan pc server untuk nilai parameter *throughput* dan *delay*.

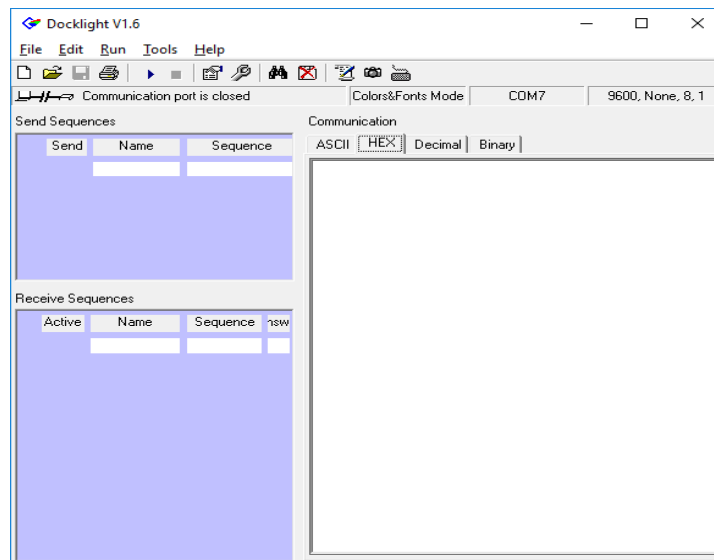


Gambar 3. 4 Tampilan Aplikasi Wireshark

e. Docklight 1.6

Docklight berfungsi untuk menampilkan komunikasi data serial, dalam penelitian digunakan untuk mengetahui nilai parameter *delay* pada modul XBEE S2.

Gambar 3.5 menunjukkan tampilan awal saat membuka aplikasi Docklight.



Gambar 3. 5 Tampilan Aplikasi Docklight

### 3.2. Peralatan

Adapun peralatan yang dibutuhkan sebagai berikut :

- a. Laptop untuk *database server*- 1 unit

Tabel 3. 1 Spesifikasi Laptop Server

Sistem Operasi	Windows 10 <i>Education</i> 64 Bit
RAM	4 GB
CPU	Intel (R) I3-5005U @CPU 2.00 GHz

- b. Laptop untuk Pengambilan data – 1 unit

Tabel 3. 2 Spesifikasi Laptop untuk pengambilan data

Sistem Operasi	Windows 10 <i>Education</i> 32 Bit
RAM	2 GB
CPU	Intel (R) Dual core @CPU 1.60 GHz

- c. Arduino uno R3– 1 unit

Tabel 3. 3 Arduino uno R3

<i>Flash Memori</i>	32 KB
<i>Clock Speed</i>	16 MHz
I/O	15 pin <i>digital</i> , 6 pin <i>analog</i>

- d. Xbee S2 - 2 unit

Tabel 3. 4 Spesifikasi Xbee S2

Frekuensi	2,4 GHz
<i>Data rate</i>	250 kbs
Catu Daya	3,3 V

- e. Xbee *Shield* – 1 unit

- f. Wemos d1 *retired* ( ESP8266 (12E) )

Tabel 3. 5 Spesifikasi Wemos D1

<i>Flash Memori</i>	4 MB
<i>Clock Speed</i>	80 MHz
I/O	15 pin <i>digital</i> , 1 pin <i>analog</i>

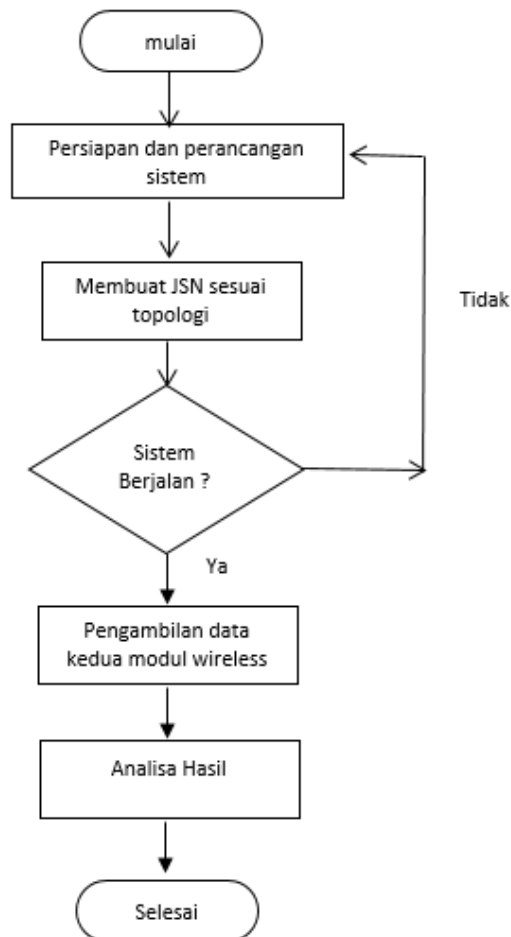
- g. Meteran untuk mengukur jarak pengujian – 1 unit

### 3.3. Prosedur Penelitian

#### 3.3.1. Metode Penelitian

Metode penelitian yang dilakukan penulis dalam penelitian ini dapat dilihat Gambar

3.6.



Gambar 3. 6 *flowchart* penelitian

#### 3.3.2. Implementasi Sistem Pengujian

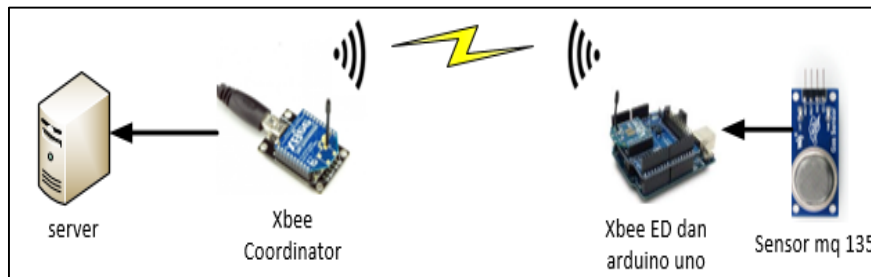
Dalam implementasi sistem pengujian dilakukan beberapa tahapan, diantaranya sebagai berikut:

##### 3.3.2.1. Perancangan Topologi

Topologi yang digunakan untuk menganalisa perbandingan kinerja modul *wireless* pada sistem pemantauan kualitas udara adalah *peer to peer* yang dibagi atas 2 skenario yaitu

topologi menggunakan modul Xbee S2 dan ESP8266. Adapun skenario yang dimaksud disajikan pada Gambar 3.7 dan Gambar 3.8.

Pada konfigurasi perancangan topologi jaringan dengan menggunakan modul Xbee S2, untuk pengiriman data dari sensor menggunakan dua buah Xbee S2. Sensor mq 135 dihubungkan pada arduino yang telah terpasang Xbee *shield* dan Xbee *end device*. Pada server terpasang Xbee *coordinator* untuk menerima data yang dikirim dari sensor.



Gambar 3. 7 Perancangan Topologi Jaringan Xbee S2

Pada konfigurasi perancangan topologi jaringan dengan menggunakan modul ESP8266 (12E), untuk pengiriman data dari sensor menggunakan jaringan *Wi-Fi* yang terpasang pada wemos D1. Perangkat Wemos D1 dan Server akan terkoneksi pada jaringan *Wi-Fi* yang sama, sehingga kedua perangkat dapat berkomunikasi.



Gambar 3. 8 Perancangan Topologi Jaringan ESP8266

### 3.4. Perancangan Sistem

Perancangan sistem pada sub bab ini dibagi menjadi beberapa bagian, mulai dari perancangan jaringan sensor nirkabel, pembuatan *database*, dan pembuatan web sebagai penampil data dari sensor.

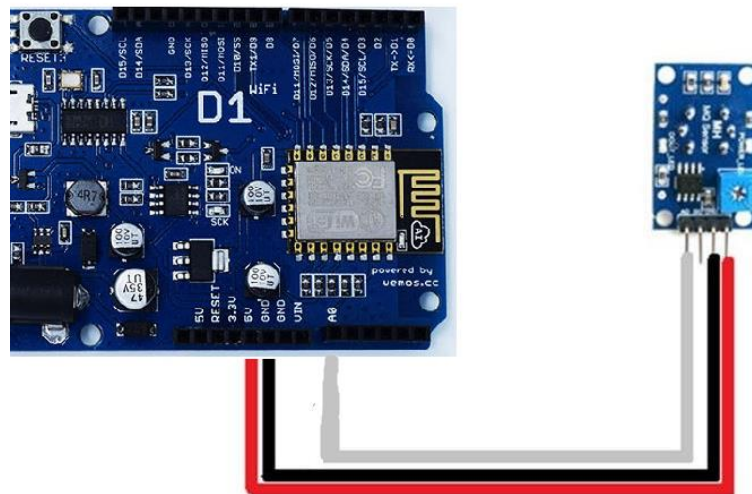
### 3.4.1. Perancangan Jaringan Sensor Nirkabel

Perancangan jaringan sensor nirkabel dibagi menjadi 2, yaitu perancangan dengan modul ESP8266(12E) dan modul Xbee S2.

#### 3.4.1.1. Perancangan Modul ESP8266(12E)

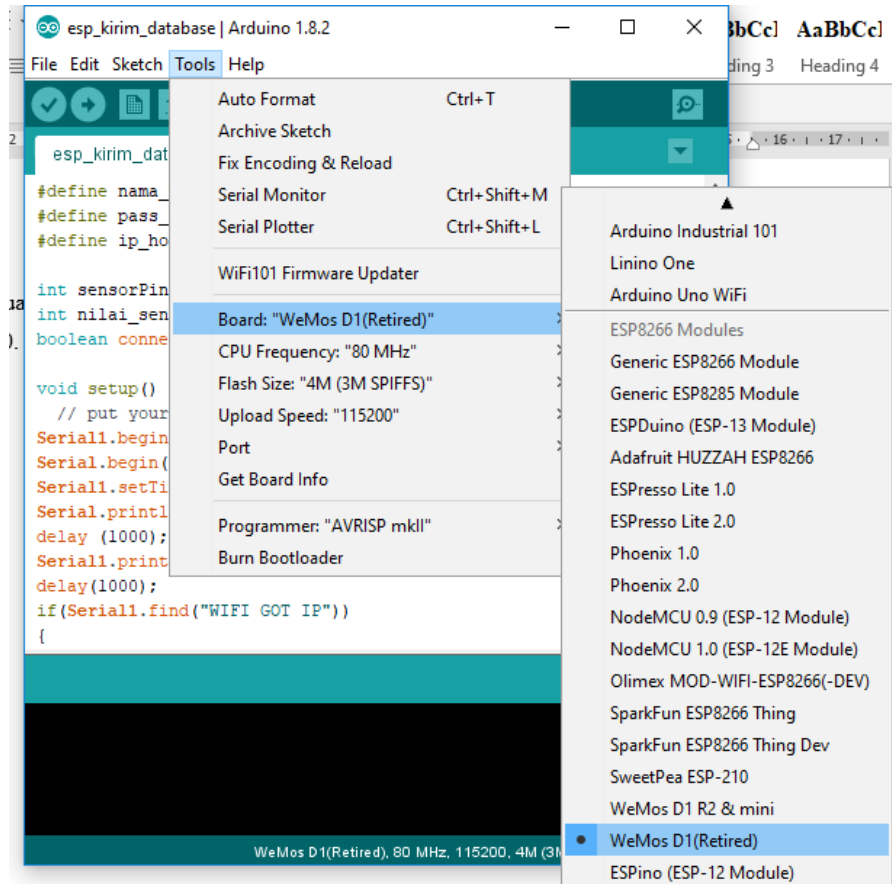
Dalam tahap ini membuat jaringan sensor sensor nirkabel dengan menggunakan modul ESP8266 (12E) seperti topologi pada gambar 3.7 . Perangkat yang digunakan untuk mengolah data dari sensor yaitu wemos d1 yang terintegrasi modul ESP8266 (12E) yang berfungsi untuk mengirim data sensor ke server.

Sensor MQ135 akan dihubungkan ke pin Analog input (A0) , pin *ground* (GND), dan pin power (5V) seperti pada Gambar 3.9. Selanjutnya mengkonfigurasi wemos agar dapat membaca sensor dan mengirimkannya server.



Gambar 3. 9 Rangkaian modul ESP8266 dan sensor mq 135

Konfigurasi pada wemos menggunakan aplikasi yang bersifat *open source* yaitu Arduino IDE dengan versi 1.8.2, setelah membuka aplikasi Arduino, kemudian pilih *board* Wemos D1 untuk mengkonfigurasi perangkat Wemos seperti yang ditunjukkan pada Gambar 3.10.



Gambar 3. 10 Konfigurasi Wemos pada Arduino IDE

Proses deteksi data sensor akan diatur oleh wemos yang kemudian akan dikirim ke server. Modul ESP8266 akan terkoneksi jaringan *Wi-Fi* yang sudah ditentukan, yang kemudian akan menampilkan data dari sensor pada protokol http untuk dikirim ke server.

Untuk mendapatkan nilai PPM (Part Per Milion) sebagai acuan kualitas udara menggunakan sebuah *library*. Sehingga cukup dengan memanggil fungsi `getPPM()` seperti gambar 3.11. Nilai data analog yang didapat sensor dikonversi menjadi nilai PPM dengan menggunakan *library*, sehingga data yang akan dikirim ke server sudah dalam bentuk PPM.

```
MQ135 gasSensor = MQ135(A0);  
float udara = gasSensor.getPPM();  
ppm=udara;  
delay(5000);  
}  
}
```

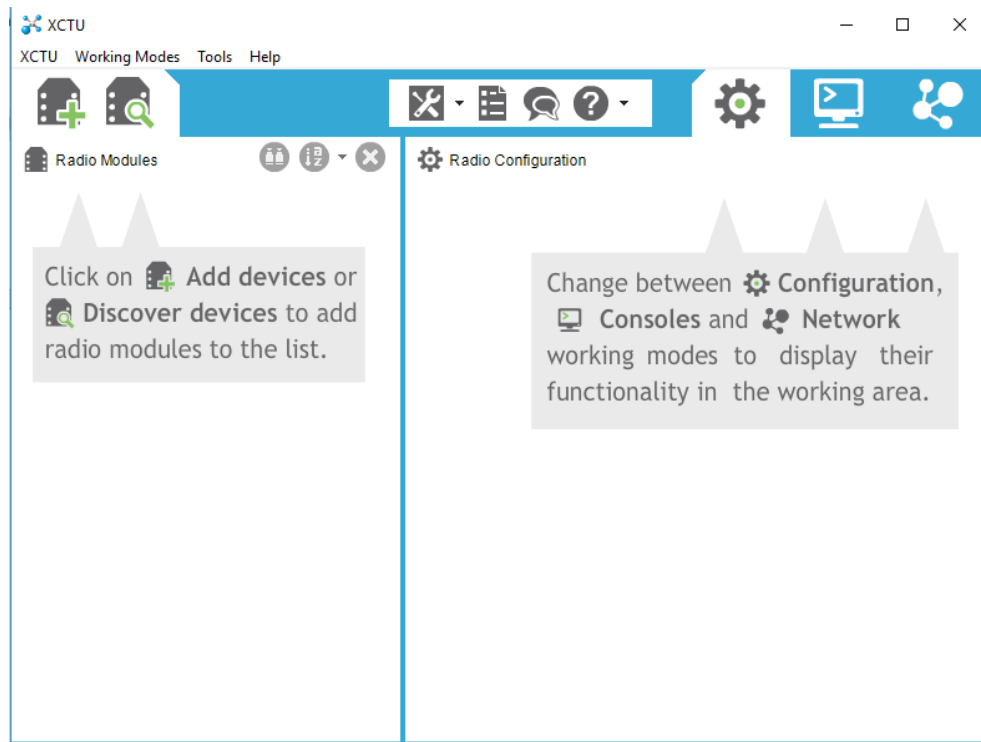
Gambar 3. 11 Rangkain modul esp8266 dan sensor mq 135

### 3.4.1.2. Perancangan Modul XBEE S2

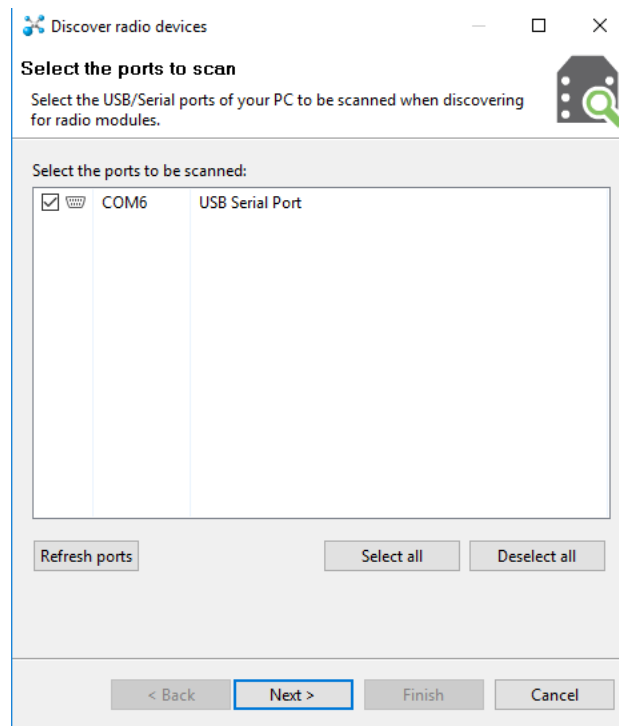
Tahan pertama yang dilakukan dalam perancangan modul Xbee S2 adalah mengkonfigurasi modul menjadi *coordinator* dan *end device*. Untuk mengkonfigurasi xbee menggunakan aplikasi X-CTU dengan versi 6.3.5. Setelah dikonfigurasi kedua Xbee akan dites terlebih dahulu sebelum dipasang pada arduino untuk mengirimkan data sensor,

Gambar 3.12 merupakan tampilan awal saat membuka aplikasi X-CTU, untuk menambah Xbee untuk dikonfigurasi dengan cara mengeklik X-CTU lalu pilih *discover radio modules*, maka nanti akan keluar jendela *pop-up* yang menampilkan Xbee yang ada.

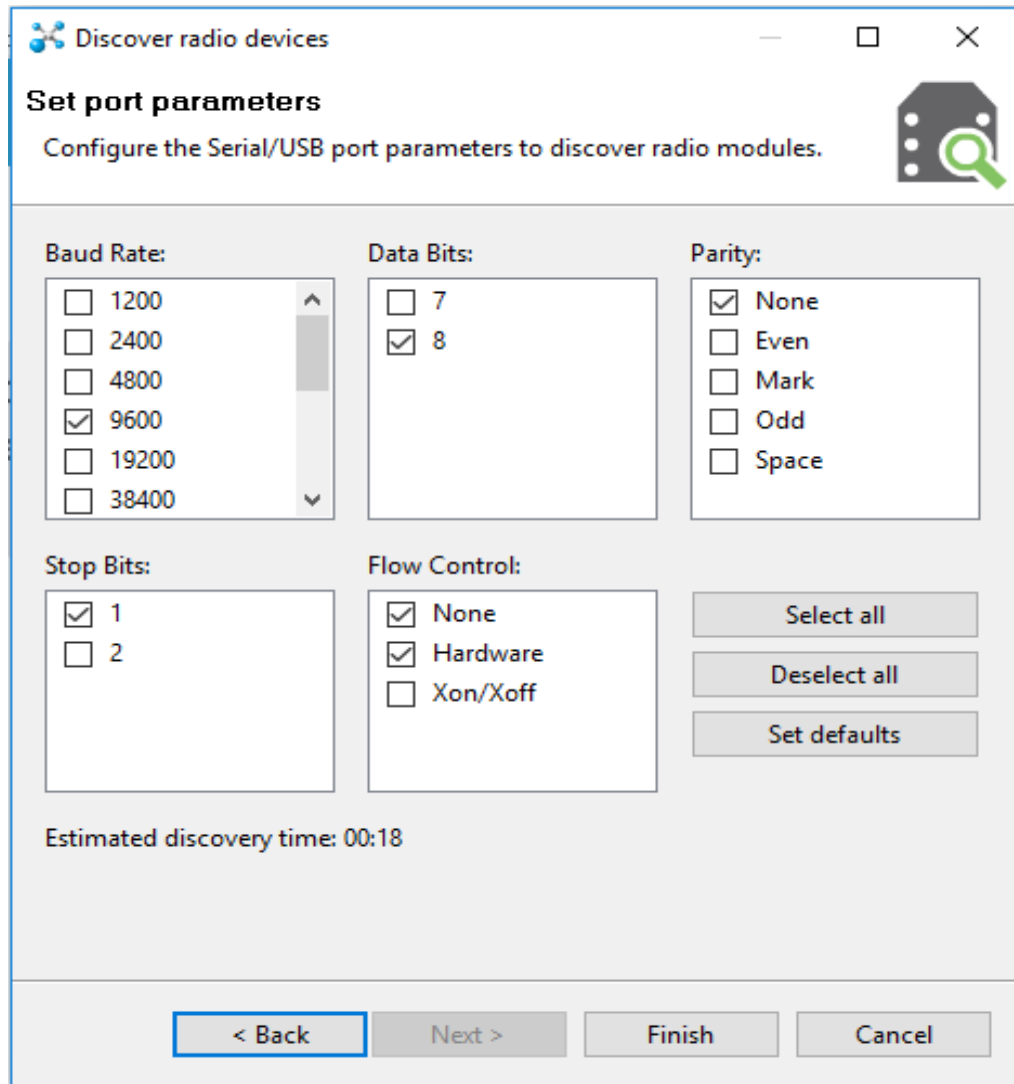
Pada Gambar 3.13 menampilkan jendela *pop-up* yang berisi port-port dari Xbee yang terhubung ke pc. Mengeklik next untuk melanjutkan konfigurasi Xbee. Konfigurasi seperti Gambar 3.14, maka nanti akan muncul halaman untuk mengkonfigurasi Xbee. Kemudian akan mengkonfiguasi kedua Xbee menjadi Xbee *coordinator* dan Xbee *end device*. Xbee *coordinator* akan dihubungkan ke server untuk menerima data dari sensor, sedangkan Xbee *end device* akan dipasangkan ke sensor pada arduino dengan Xbee *shield* untuk mengirim data ke server.



Gambar 3. 12 Tampilan awal X-CTU



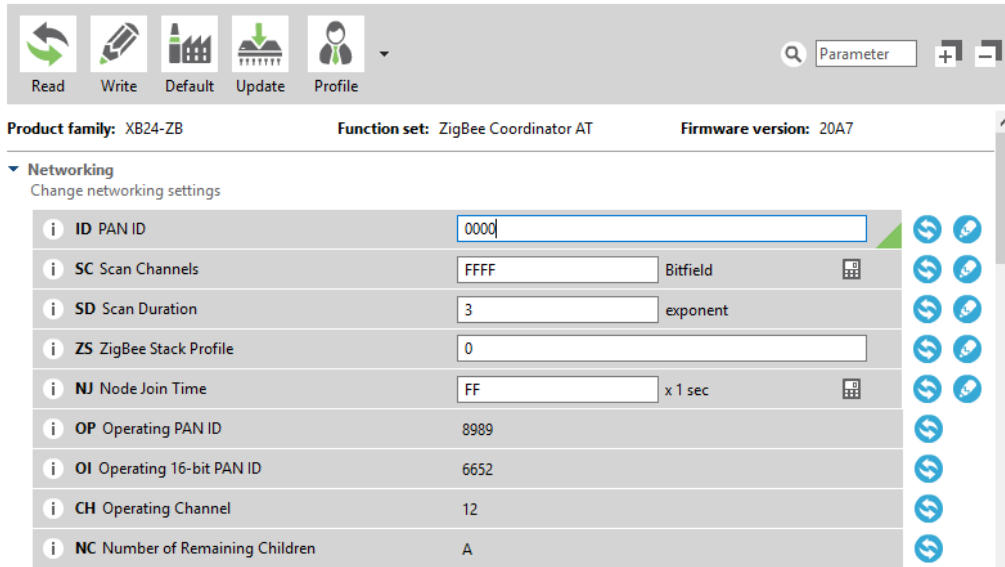
Gambar 3. 13 Konfigurasi *Port* pada Xbee



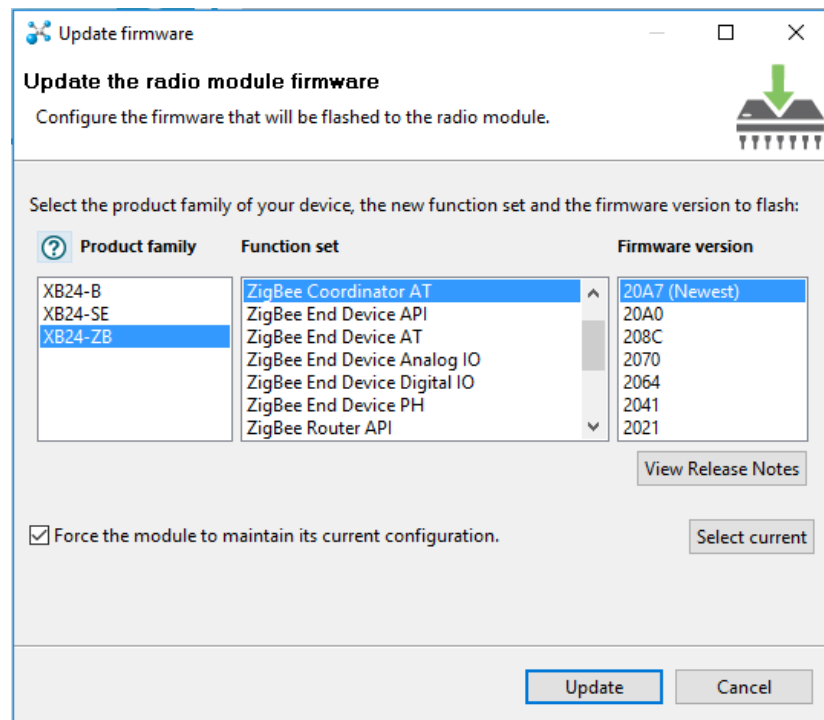
Gambar 3. 14 Konfigurasi *baud rate* pada Xbee

Setelah muncul halaman untuk mengkonfiguasi Xbee seperti Gambar 3.15, selanjutnya mengubah mode dari *Router ( Default mode )* menjadi *coordinator* dan *end Device* seperti yang ditunjukkan pada Gambar 3.16. *End device* berfungsi sebagai perangkat untuk mengirim data dari sensor ke server, *Coordinator* berfungsi untuk menerima data yang dikirimkan oleh *End Device*. Pada tahap proses perubahan Xbee Router ke Xbee *coordinator* atau Xbee *end device*, kadang-kadang akan terjadi kegagalan. Sehingga perlu mencoba beberapa kali untuk mengubah Xbee menjadi *coordinator* maupun *end device*, dan bila masih gagal juga, perangkat Xbee perlu untuk *reset* ulang.

Radio Configuration [ - 0013A20040C0E4BC]

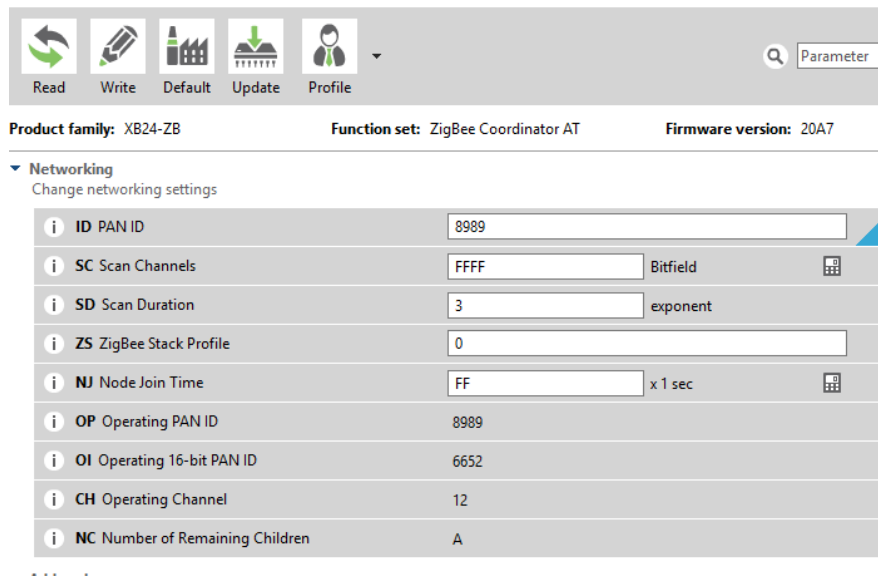


Gambar 3. 15 Konfigurasi Xbee



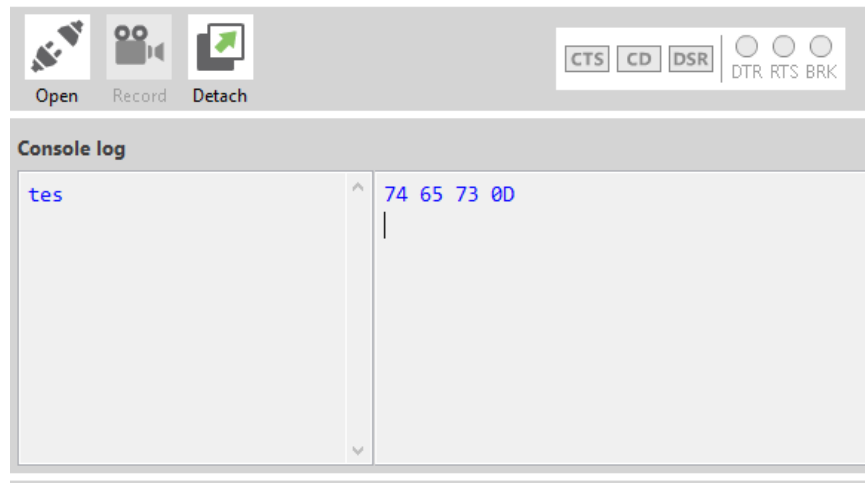
Gambar 3. 16 Konfigurasi Xbee menjadi *coordinator* dan *end device*

Pada Gambar 3.17 menunjukkan tampilan untuk mengkonfigurasi Xbee, agar *coordinator* dan *end device* dapat berkomunikasi dengan mengkonfiguasi Pan Id. Bila Pan Id kedua perangkat Xbee disamakan maka Xbee tersebut akan berada dalam jaringan yang sama.



Gambar 3. 17 Konfigurasi PAN ID pada Xbee

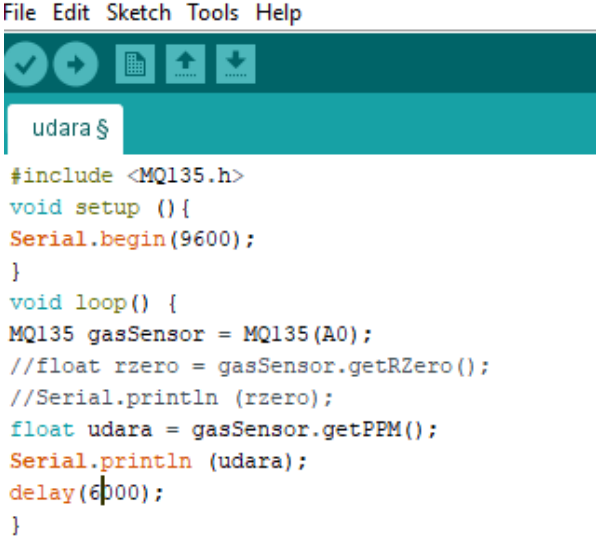
Setelah mengkonfigurasi Xbee tahap selanjutnya mengetes koneksi dari kedua Xbee tersebut seperti yang ditunjukkan pada Gambar 3.18. Bila Xbee sudah terkoneksi satu sama lainnya, kemudian memasang Xbee *end device* ke Xbee Shield yang dipasang pada arduino untuk mengirim data dari sensor.



Gambar 3. 18 Pengetesan komunikasi Xbee

Setelah Xbee terpasang pada arduino, kemudian mengkonfigurasi Arduino untuk mendeteksi kualitas udara. Untuk mendapatkan nilai PPM sebagai acuan kualitas udara

menggunakan sebuah *library*. Sehingga cukup dengan memanggil fungsi `getPPM()` seperti pada Gambar 3.19.



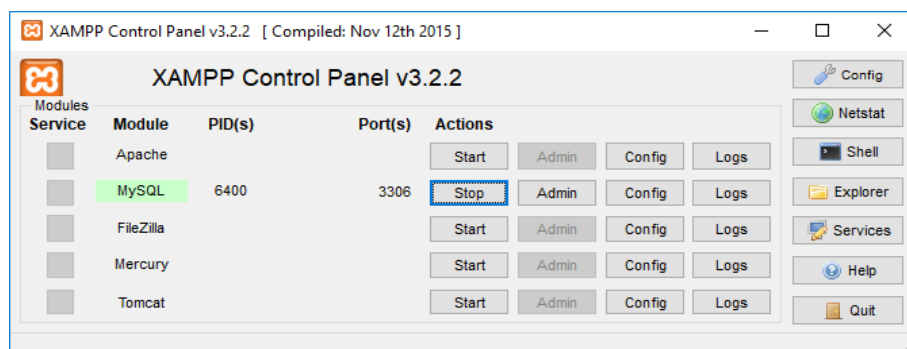
```
File Edit Sketch Tools Help
udara $
#include <MQ135.h>
void setup () {
  Serial.begin(9600);
}
void loop() {
  MQ135 gasSensor = MQ135(A0);
  //float rzero = gasSensor.getRZero();
  //Serial.println (rzero);
  float udara = gasSensor.getPPM();
  Serial.println (udara);
  delay(1000);
}
```

Gambar 3. 19 Konfigurasi Arduino pada Xbee

Xbee End Device akan mengirimkan data dari sensor dengan perintah `Serial.println()` dan diikuti variable yang dibutuhkan untuk dikirim.


### 3.4.1. Perancangan *Database*

Setelah melakukan perancangan kedua modul *wireless*, selanjutnya membuat *database* yang digunakan untuk menyimpan data dari sensor. Pada penelitian ini menggunakan *xampp control panel* dengan versi 3.2.2 seperti pada Gambar 3.20.



Gambar 3. 20 Pengaturan xampp

Setelah mengaktifkan mysql dan apache server pada xampp, kemudian membuka broswer dengan alamat localhost/phpmyadmin/ untuk membuat database. Setelah membuat database kemudian membuat tabel untuk menyimpan data dari sensor seperti pada Gambar 3.21.

#	Name	Type	Collation	Attributes	Null
<input type="checkbox"/>	1	<b>id</b> 	int(11)		No
<input type="checkbox"/>	2	<b>nilai</b>	varchar(20)	latin1_swedish_ci	No
<input type="checkbox"/>	3	<b>waktu</b>	datetime		No

Gambar 3. 21 Membuat Tabel di *Database*

#### 3.4.2. Menyimpan Data Sensor ke *Database*

Setelah membuat *database* untuk menyimpan data sensor, selanjutnya membuat koneksi untuk menghubungkan sensor dengan *database*. Pada penelitian ini untuk menghubungkan modul Xbee dengan *database* menggunakan aplikasi pyton 2.7, sedangkan untuk modul ESP8266 menggunakan Hypertext Preprocessor (PHP).

Pada jaringan sensor dengan modul Xbee menggunakan aplikasi pyton untuk membaca data serial dari *coordinator* yang sudah menerima data dari sensor, seperti yang terlihat pada gambar 3.22.

Untuk menerima data dari sensor menggunakan perintah pada baris “ data = arduino.readline()” yang berfungsi untuk membaca data *serial* yang terima oleh Xbee coordinator, kemudian data tersebut disimpan pada *database server* yang telah dibuat sebelumnya.

```
import serial
import MySQLdb
import time
db = MySQLdb.connect("localhost","root","","udara")

port = 'COM6'
print "Connecting...",port
arduino = serial.Serial(port, 9600)
print "arduino detected"

while True:
    data = arduino.readline()
    time.sleep(5)
    print data
    cursor = db.cursor()
    cursor.execute("INSERT INTO datasensor (nilai) VALUES (%s)", (data))
    db.commit()
    cursor.close()
```

Gambar 3. 22 koneksi ke database untuk modul Xbee

Pada baris “ db = MySQLdb.connect("...") “ merupakan program untuk koneksi ke database, sendang pada baris “ cursor.execute("...”) “ untuk memasukkan data ke tabel pada database yang telah dibuat sebelumnya.

Pada jaringan sensor dengan modul ESP8266 menggunakan PHP atau *Hypertext Preprocessor* untuk mengambil data sensor yang ditampilkan pada protokol HTTP atau web, program seperti Gambar 3.24.

```
1 <head>
2 <meta http-equiv="refresh" content="5">
3 </head>
4 <?php
5
6 include ('koneksi.php');
7
8 $ppm= file_get_contents('http://192.168.137.151/data');
9 $stampil = str_ireplace(" ", "", $ppm);
10 mysqli_query($conn, "insert into $namatabel (nilai) VALUES('" . $stampil . "')");
11 echo $stampil;
12
```

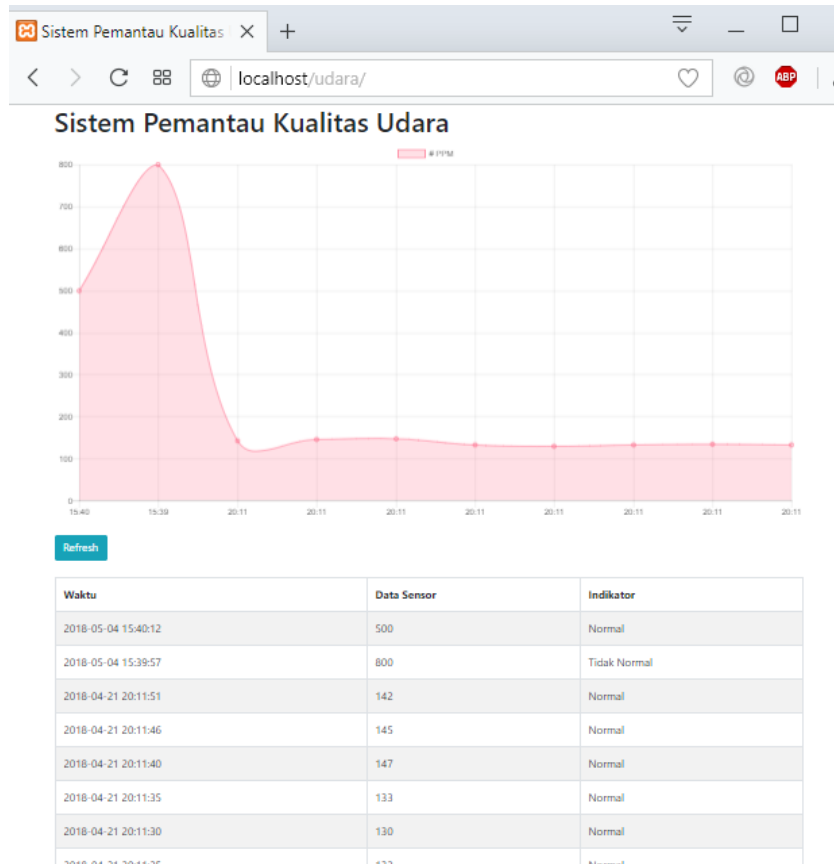
Gambar 3. 23 koneksi ke database untuk modul ESP8266

Pada penelitian ini menggunakan *wireless* pada laptop untuk membuat jaringan *Wi-Fi*, kemudian modul ESP8266 terkoneksi pada jaringan *Wi-Fi* tersebut. Pada baris 8 merupakan perintah program untuk mengambil data sensor dari modul ESP8266 yang terkoneksi pada jaringan *Wi-Fi*. Pada baris 10 merupakan perintah untuk memasukkan data ke tabel pada database yang telah dibuat sebelumnya.

### 3.4.3. Membuat Tampilan WEB

Web pada penelitian ini digunakan untuk menampilkan data dari database yang diperoleh dari sensor. Data akan ditampilkan dalam bentuk grafik dan tabel seperti gambar 3.24 Grafik akan menampilkan data PPM yang didapat dari sensor dan waktu data tersebut tersimpan data di database, sedangkan tabel akan menampilkan data sensor, waktu, dan indikator data dari sensor.

Pada penelitian yang dilakukan oleh Aditya Deni Nur R, nilai polutan yang setara dengan gas karbondioksida (CO<sub>2</sub>) yang didapat ketika udara normal adalah <180 PPM dan nilai polutan yang setara dengan gas karbondioksida (CO<sub>2</sub>) dianggap berbahaya adalah >778 PPM (Nur, 2014). Dengan menggunakan acuan tersebut dapat dijadikan indikator kualitas udara normal atau tidak normal pada sistem ini.



Gambar 3. 24 Tampilan Website

### 3.5. Metode Pengujian

Pada tahap ini penulis akan menjelaskan tahap pengujian dimana akan mendapatkan nilai parameter QoS dari kedua modul *wireless*. Pengambilan nilai QoS berdasarkan kualitas pengiriman data dalam jaringan *wireless*. Nilai QoS ditentukan dari parameter *throughput*, *packet loss* dan *delay*. Nilai QoS yang didapatkan akan dibandingkan untuk mendapat hasil dari perbandingan kinerja kedua modul *wireless* tersebut, sehingga dapat diketahui kinerja layanannya.

#### 3.5.1 Topologi Jaringan

Pengujian dilakukan dengan cara mengukur data yang dikirimkan dari sensor dengan modul *wireless* ke server, untuk mengukur modul Xbee menggunakan aplikasi X-CTU dan docklight, sedangkan untuk mengukur modul ESP8266 menggunakan Wireshark. Perhitungan nilai parameter dilakukan dengan menyalin data nilai QoS yang didapatkan dari aplikasi tersebut, kemudian di tempel ke Microsoft Excel.

Gambar 3.25 menampilkan topologi untuk pengujian QoS, dimana data akan dikirimkan dari sensor ke server dengan menggunakan modul *wireless*. Pada saat pengiriman data tersebut akan diambil nilai parameter QoSnya untuk dicatat kemudian dibandingkan dan dianalisis.



Gambar 3. 25 Topologi pengujian QoS

### 3.5.2 Skenario Pengujian

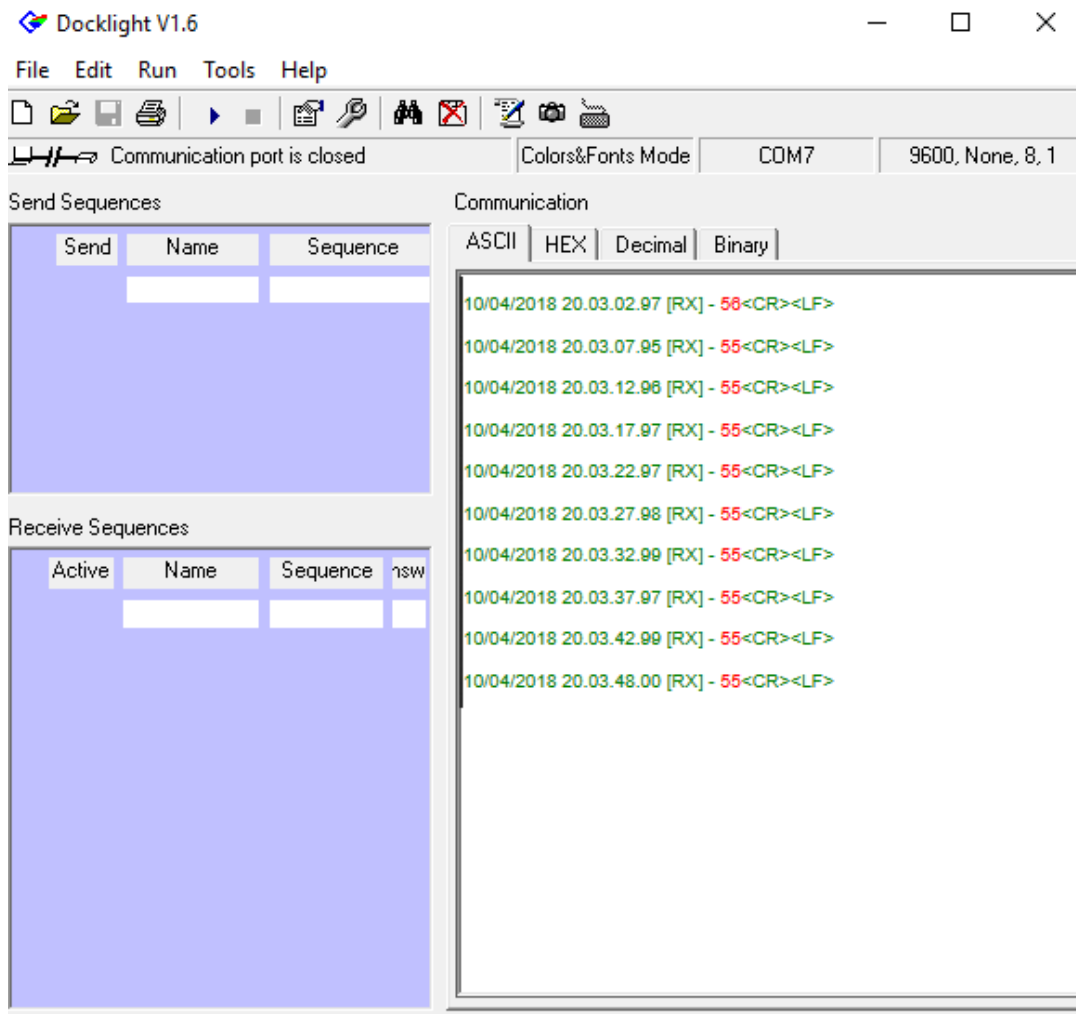
Dalam melakukan skenario pengujian pada penelitian, diawali dengan merancang sistem berdasarkan perancangan topologi yang ada. Setelah sistem berjalan dilakukan percobaan untuk mengirim data dari sensor ke server, dalam percobaan tersebut data QoS yang diambil adalah *delay*, *throughput*, dan *packet loss*.

Pengujian ini dilakukan berdasarkan jarak yang berbeda yaitu 5, 10, 15, dan 20 meter sehingga dapat dilakukan perbandingan kinerja dari kedua modul tersebut. Pengiriman data dilakukan dengan beberapa kali percobaan, data yang dibandingkan merupakan hasil rata-rata dari percobaan tersebut.

### 3.5.3 Penghitungan Delay

Untuk menghitung delay dari suatu pengiriman data dengan modul Xbee dapat menggunakan aplikasi docklight seperti pada Gambar 3.26 untuk mencatat waktu kirim dan terima paket data. nilai *delay* dapat diperoleh dari selisih waktu kirim dan terima suatu paket. Waktu kirim merupakan pencatatan pada *end device*, sedangkan waktu terima pada *coordinator*. Untuk mencatat waktu kirim dan waktu terima menggunakan 2 laptop, laptop pertama digunakan untuk mencatat waktu kirim pada *end device*, laptop kedua digunakan untuk mencatat waktu terima pada *coordinator*.

Untuk menghitung *delay* dari suatu pengiriman data dengan modul ESP8266 menggunakan aplikasi wireshark seperti pada gambar 3.27. Dengan menggunakan aplikasi dapat berfungsi untuk mengetahui waktu kirim dan terima paket sehingga dapat mengetahui waktu *delay* pengiriman data. Aplikasi wireshark akan dipasang pada pc server kemudian ESP8266 akan dikoneksikan pada jaringan *Wi-Fi* yang sama dengan pc server sehingga aplikasi wireshark dapat melihat transmisi data pada pc server dengan modul ESP8266.



Gambar 3. 26 Delay Xbee

9	7.216110	192.168.137.1	192.168.137.43	TCP	66	5.108843000
10	7.217823	192.168.137.43	192.168.137.1	TCP	58	0.001713000
11	7.217945	192.168.137.1	192.168.137.43	TCP	54	0.000122000
12	7.218011	192.168.137.1	192.168.137.43	HTTP	117	0.000066000
13	7.226495	192.168.137.43	192.168.137.1	TCP	54	0.008484000
14	12.223003	192.168.137.43	192.168.137.1	TCP	169	4.996508000
15	12.263962	192.168.137.1	192.168.137.43	TCP	54	0.040959000
16	12.265515	192.168.137.43	192.168.137.1	HTTP	58	0.001553000
17	12.310683	192.168.137.1	192.168.137.43	TCP	54	0.045168000
18	14.315096	192.168.137.43	192.168.137.1	TCP	54	2.004413000
19	14.315173	192.168.137.1	192.168.137.43	TCP	54	0.000077000
20	14.315241	192.168.137.1	192.168.137.43	TCP	54	0.000068000
21	14.316824	192.168.137.43	192.168.137.1	TCP	54	0.001583000

▼ Frame 12: 117 bytes on wire (936 bits), 117 bytes captured (936 bits) on interface 0

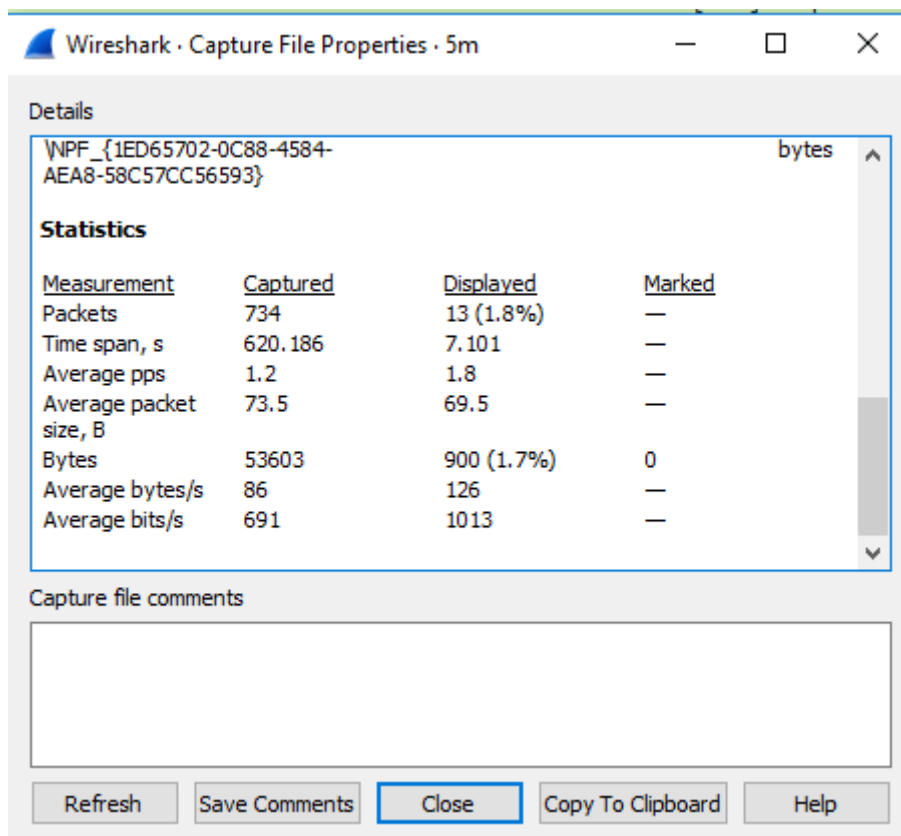
- > Interface id: 0 (\Device\NPF\_{1ED65702-0C88-4584-AEA8-58C57CC56593})
- Encapsulation type: Ethernet (1)
- Arrival Time: Apr 14, 2018 15:40:35.948286000 SE Asia Standard Time
- [Time shift for this packet: 0.000000000 seconds]
- Epoch Time: 1523695235.948286000 seconds
- [Time delta from previous captured frame: 0.000066000 seconds]
- [Time delta from previous displayed frame: 0.000066000 seconds]

Gambar 3. 27 Delay ESP8266

### 3.5.4 Penghitungan *Throughput*

Untuk menghitung *Throughput* dari suatu pengiriman data dengan modul Xbee menggunakan aplikasi xctu untuk mendapatkan nilai besar data yang diterima dan aplikasi docklight untuk mendapat waktu pengiriman. *Throughput* merupakan jumlah total kedatangan paket yang sukses yang diamati pada tujuan selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut.

Untuk menghitung *throughput* dari suatu pengiriman data dengan modul ESP8266 menggunakan aplikasi wireshark seperti pada Gambar 3.28. Nilai *throughput* akan tertampil dengan satuan byte/s.



Gambar 3. 28 *Throughput* ESP8266

### 3.5.5 Penghitungan *Packet loss*

Untuk menghitung *packet loss* dari suatu pengiriman data dengan modul Xbee dan ESP8266 dalam penelitian dengan membandingkan jumlah paket data yang dikirim sensor dan diterima oleh server. Packet loss merupakan paket yang hilang saat pengiriman data, jumlah data yang diterima dapat dilihat pada database server seperti pada Gambar 2.29.



Gambar 3. 29 Data pada *database*

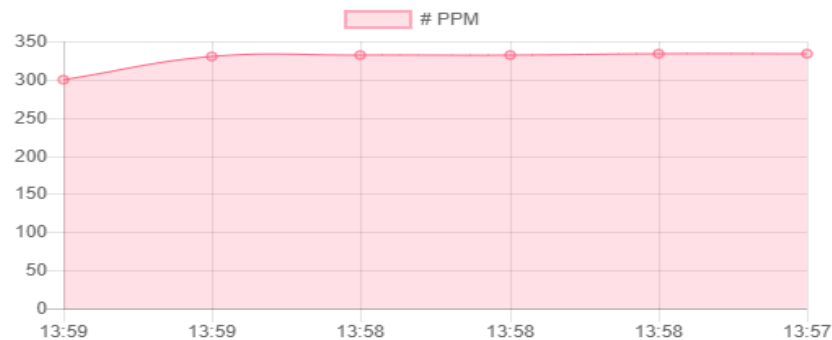
## BAB IV

### HASIL PENELITIAN DAN PEMBAHASAN

Analisis data dilakukan berdasarkan kumpulan data yang telah diambil berdasarkan parameter pada tiap skenario pengujian yang telah ditentukan. Nilai parameter disajikan dalam bentuk grafik kemudian dianalisis.

Sistem yang dibuat akan mengumpulkan informasi kandungan CO<sub>2</sub> dari sensor MQ 135 yang kemudian akan dikirimkan ke server dengan menggunakan modul ESP8266(12e) dan Xbee S2. Sesampainya di server, data tersebut diteruskan ke *database* untuk disimpan. Data dari sensor yang sudah disimpan akan ditampilkan pada sebuah web. Web tersebut akan menampilkan grafik dan tabel yang datanya diambil dari *database* seperti yang terlihat pada Gambar 4.1

## Sistem Pemantau Kualitas Udara



Waktu	Data Sensor (PPM)	Indikator
2018-07-18 13:59:22	300	Normal
2018-07-18 13:59:02	331	Normal
2018-07-18 13:58:46	332	Normal

Gambar 4.1 Tampilan web

Grafik dan Tabel akan diperbarui secara otomatis bila ada data baru yang dikirim dari sensor, sehingga mempermudah untuk melihat perubahan nilai kandungan CO2 pada lingkungan sekitar sensor.

#### 4.1. Pengujian Sistem

Pada tahap ini melakukan pengujian sistem untuk mengecek apakah sistem berjalan dengan baik atau tidak. Pengujian pertama dilakukan pada modul ESP8266 dengan melakukan tes ping pada IP modul ESP8266, hal ini untuk memastikan bahwa koneksi antara server dan modul ESP8266 pada kondisi yang bagus. Hasil tes ping dapat dilihat pada Gambar 4.2

```
C:\Users\Ricky H>ping 192.168.137.43

Pinging 192.168.137.43 with 32 bytes of data:
Reply from 192.168.137.43: bytes=32 time=1ms TTL=64
Reply from 192.168.137.43: bytes=32 time=1ms TTL=64
Reply from 192.168.137.43: bytes=32 time=1ms TTL=64
Reply from 192.168.137.43: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.137.43:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

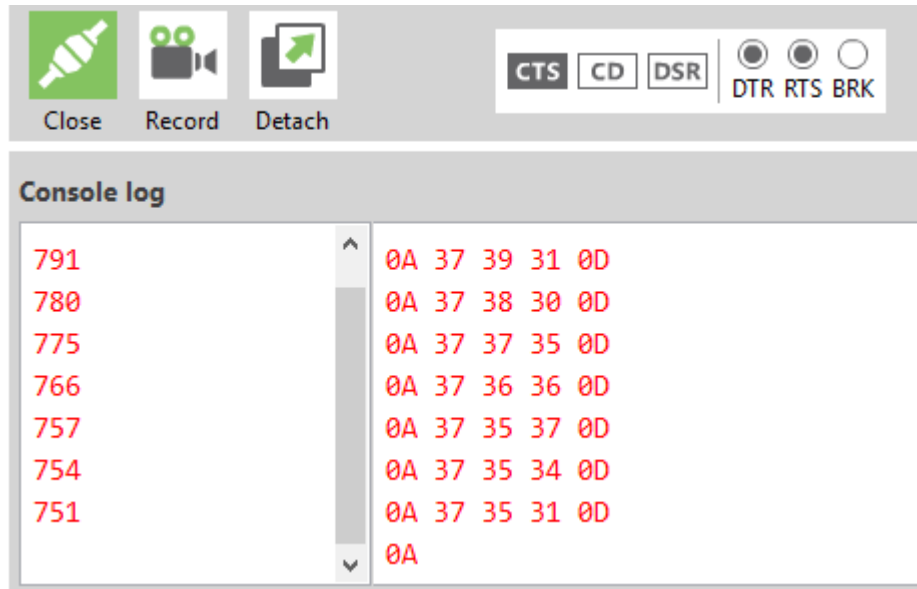
Gambar 4.2 Tes ping modul ESP8266

Setelah melakukan tes ping, kemudian melihat apakah data dari sensor berhasil dikirim atau tidak. Hasil pengujian uji pengiriman data dari sensor dapat dilihat dengan aplikasi wireshark seperti yang terlihat pada gambar 4.3, terjadi komunikasi bolak-balik melalui protokol HTTP antara server dengan modul ESP8266.

→	...	366.429332	192.168.137.1	192.168.137.43	HTTP	117
	...	373.559247	192.168.137.1	192.168.137.43	TCP	54
	...	366.429261	192.168.137.1	192.168.137.43	TCP	54
←	...	371.508883	192.168.137.43	192.168.137.1	HTTP	58
	...	366.429131	192.168.137.43	192.168.137.1	TCP	58
	...	373.587091	192.168.137.43	192.168.137.1	TCP	54

Gambar 4.3 Komunikasi antara server dengan ESP8266

Pengujian kedua dilakukan pada modul Xbee S2 dengan menggunakan aplikasi XTCU untuk melihat komunikasi antara *Coordinator* dengan *End Device*. Gambar 4.4 menunjukkan pengiriman data dari sensor terkirim sukses ke Xbee *Coordinator*.

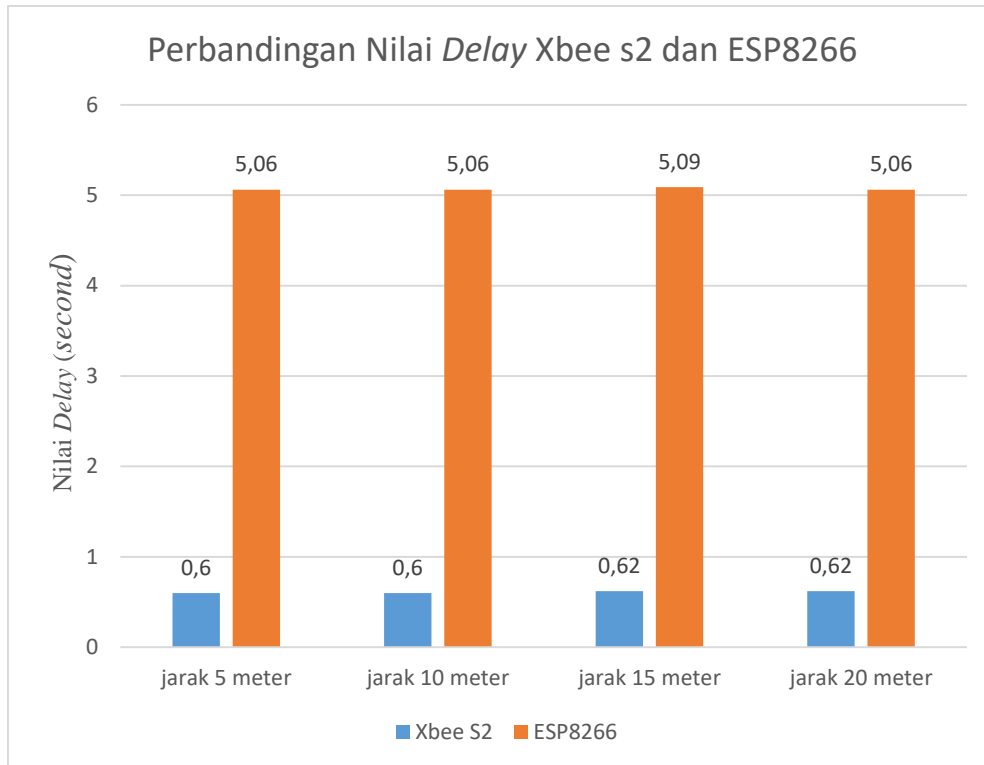


Gambar 4.4 Tes pengiriman data Xbee S2

Setelah melakukan Pengujian Sistem pada pengiriman data kedua modul Xbee S2 dan ESP8266, Selanjutnya melakukan Pengujian QoS berdasarkan parameter *delay*, *throughput*, dan *packet loss*. Nilai parameter akan dibanding pada sebuah Grafik kemudian dianalisis sehingga dapat dilihat perbandingan nilai QoS antara modul Xbee S2 dan ESP8266.

#### 4.2. Pengujian *Delay*

Pengambilan data *delay* dilakukan untuk mengetahui waktu tunda yang disebabkan oleh transmisi dari sensor ke server. Pengambilan data *delay* dilakukan dengan beberapa skenario jarak transmisi data antara sensor dengan server. Penghitungan *delay* modul Xbee S2 dilakukan dengan menggunakan aplikasi Docklight, sedangkan modul ESP8266(12-E) dengan menggunakan Wireshark. Hasil pengujian pada *delay* dihitung dengan satuan detik.



Gambar 4.5 Perbandingan Nilai Delay Xbee s2 dan ESP8266

Pada Gambar 4.5 merupakan hasil pengujian dari *delay* pengiriman data dengan menggunakan modul Xbee S2 dan modul ESP8266(12E). Nilai parameter *delay* yang didapat dari hasil pengujian *delay* pengiriman data dengan menggunakan modul Xbee S2 dan ESP8266(12E) sebagai berikut :

- a. Pada jarak 5 meter modul Xbee S2 mempunyai *delay* sebesar 0,6 detik, sedangkan modul ES8266(12E) mempunyai *delay* sebesar 5,06 detik.
- b. Pada jarak 10 meter modul Xbee S2 mempunyai *delay* sebesar 0,6 detik, sedangkan modul ES8266(12E) mempunyai *delay* sebesar 5,06 detik.
- c. Pada jarak 15 meter modul Xbee S2 mempunyai *delay* sebesar 0,62 detik, sedangkan modul ES8266(12E) mempunyai *delay* sebesar 5,09 detik.
- d. Pada jarak 20 meter modul Xbee S2 mempunyai *delay* sebesar 0,62 detik, sedangkan modul ES8266(12E) mempunyai *delay* sebesar 5,06 detik.

Nilai *delay* pengiriman pada ESP8266 lebih besar dibandingkan Xbee dikarenakan pada proses pengiriman Xbee data dari sensor akan langsung dikirim dari *end device* ke *coordinator*, sedangkan pada modul ESP8266 terjadi beberapa proses komunikasi data yang tertangkap oleh Wireshark seperti yang terlihat pada Gambar 4.6. Pada proses pengiriman data nomor enam terdapat *delay* sebesar 4,9 detik sehingga membuat waktu *delay* lebih besar.

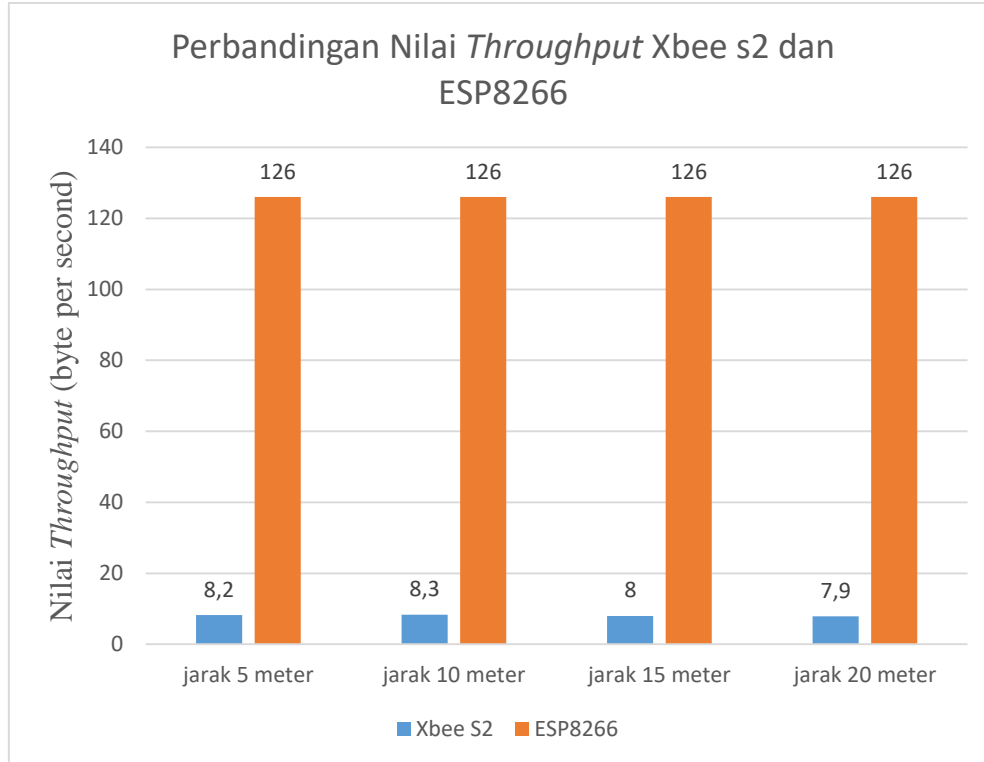
Time	Source	Destination	Protocol	Length	Time delta from previous displayed
1 0.000000	192.168.137.1	192.168.137.43	TCP	66	0.000000000
2 0.003213	192.168.137.43	192.168.137.1	TCP	58	0.003213000
3 0.003308	192.168.137.1	192.168.137.43	TCP	54	0.000095000
4 0.003376	192.168.137.1	192.168.137.43	HTTP	117	0.000068000
5 0.115204	192.168.137.43	192.168.137.1	TCP	54	0.111828000
6 5.016287	192.168.137.43	192.168.137.1	TCP	169	4.901083000
7 5.056610	192.168.137.1	192.168.137.43	TCP	54	0.040323000
8 5.059764	192.168.137.43	192.168.137.1	HTTP	58	0.003154000
9 5.103470	192.168.137.1	192.168.137.43	TCP	54	0.043706000

Gambar 4.6 Pengiriman data ESP8266 pada Wireshark

Dari data di atas membuktikan bahwa *delay* pengiriman Xbee lebih cepat dibandingkan dengan ESP8266 yang mempunyai *delay* lebih dari 5 detik. Baik Xbee dan ESP8266 mempunyai nilai *delay* pengiriman yang hampir sama, untuk Xbee sebesar 0,6 detik, sedangkan ESP8266 sebesar 5,09 detik.

#### 4.3. Pengujian *Throughput*

Pengambilan data *Throughput* dilakukan untuk mengetahui kecepatan transfer data yang sebenarnya pada transmisi data antara sensor dengan server. Pengambilan data *Throughput* dilakukan dengan beberapa skenario jarak transmisi data antara sensor dengan server. Hasil pengujian pada nilai *Throughput* ditampilkan dengan satuan *byte/s*.

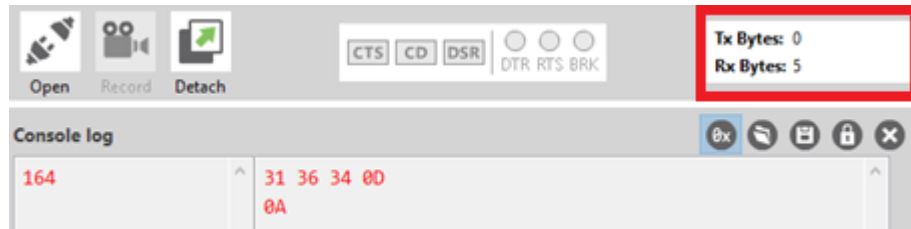


Gambar 4.7 Perbandingan Nilai *Throughput* Xbee s2 dan ESP8266

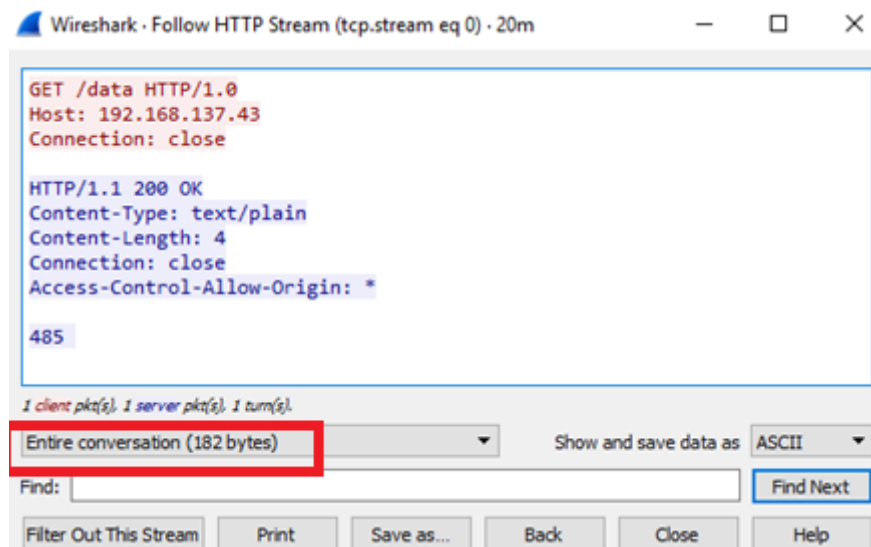
Pada Gambar 4.7 menunjukkan bahwa hasil pengujian *throughput* pengiriman data dengan menggunakan modul Xbee S2 dan modul ESP8266(12E). Nilai parameter *throughput* yang didapat dari hasil pengujian *throughput* pengiriman data dengan menggunakan modul Xbee S2 dan ESP8266(12E) sebagai berikut :

- Pada jarak 5 meter modul Xbee S2 mempunyai *throughput* sebesar 8,2 *byte/s*, sedangkan modul ES8266(12E) mempunyai *throughput* sebesar 126 *byte/s*.
- Pada jarak 10 meter modul Xbee S2 mempunyai *throughput* sebesar 8,3 *byte/s*, sedangkan modul ES8266(12E) mempunyai *throughput* sebesar 126 *byte/s*.
- Pada jarak 15 meter modul Xbee S2 mempunyai *throughput* sebesar 8 *byte/s*, sedangkan modul ES8266(12E) mempunyai *throughput* sebesar 126 *byte/s*.
- Pada jarak 20 meter modul Xbee S2 mempunyai *throughput* sebesar 7,9 *byte/s*, sedangkan modul ES8266(12E) mempunyai *throughput* sebesar 126 *byte/s*.

Nilai *throughput* pengiriman pada ESP8266 lebih besar dibandingkan Xbee dikarenakan besar data xbee lebih kecil dari pada ESP8266, besar data Xbee 5 *byte* sedangkan ESP8266 182 *byte* seperti yang ditampilkan pada Gambar 4.8 dan 4.9.



Gambar 4.8 Besar data Xbee



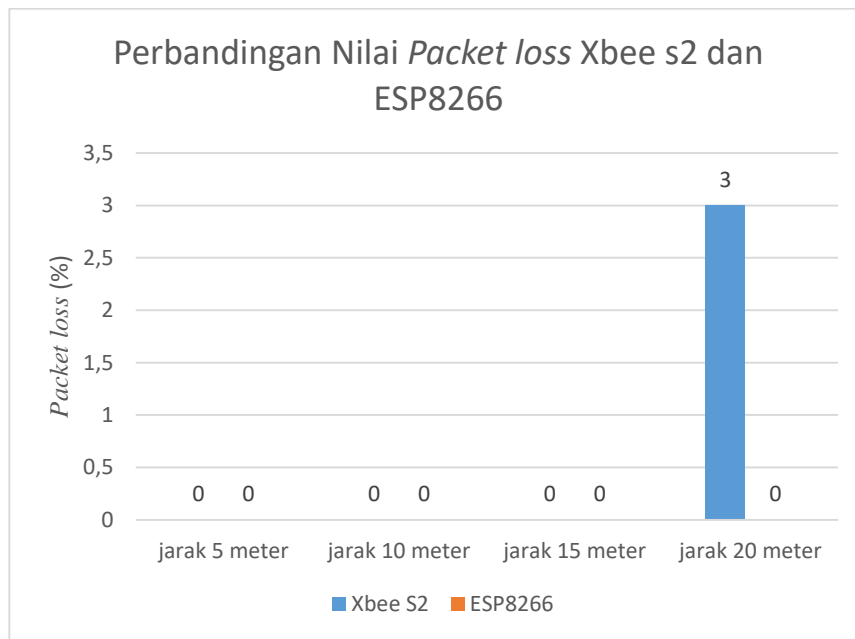
Gambar 4.9 Besar data ESP8266

Dari data di atas membuktikan bahwa *throughput* pengiriman ESP8266 lebih besar dibandingkan dengan Xbee S2. *Throughput* pada Xbee S2 mempunyai nilai yang hampir sama yaitu sebesar 8 *byte/s*, sedangkan EPS8266 mempunyai nilai yang konsisten yaitu sebesar 126 *byte/s*.

#### 4.4. Pengujian *Packet loss*

Pengambilan data *packet loss* dilakukan untuk mengetahui data yang hilang pada saat pengiriman data dari sensor ke server. Pengambilan data *packet loss* dilakukan dengan

beberapa skenario jarak transmisi data antara sensor dengan server. Penghitungan *packet loss* dengan membandingkan jumlah paket data yang dikirim sensor dengan data yang diterima oleh server. Hasil pengujian pada *packet loss* dihitung dengan satuan persen. Dalam pengujian ini dilakukan pengiriman data dari sensor sebanyak 100 kali.



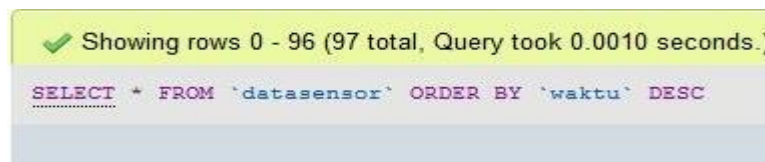
Gambar 4.10 Perbandingan Nilai *Packet loss* Xbee s2 dan ESP8266

Pada Gambar 4.10 merupakan hasil pengujian *packet loss* pengiriman data dengan modul Xbee S2 dan ESP8266. Nilai parameter *packet loss* yang didapat dari hasil pengujian *delay* pengiriman data dengan menggunakan modul Xbee S2 dan ESP8266(12E) sebagai berikut :

- a. Pada jarak 5 meter modul Xbee S2 dan ES8266(12E) mempunyai *packet loss* sebesar 0 %, hal ini membuktikan dengan pada jarak transmisi 5 meter, data dari sensor berhasil sampai ke server tanpa kehilangan paket data.

- b. Pada jarak 10 meter modul Xbee S2 dan ES8266(12E) mempunyai *packet loss* sebesar 0 %, hal ini membuktikan dengan pada jarak transmisi 10 meter, data dari sensor berhasil sampai ke server tanpa kehilangan paket data.
- c. Pada jarak 15 meter modul Xbee S2 dan ES8266(12E) mempunyai *packet loss* sebesar 0 %, hal ini membuktikan dengan pada jarak transmisi 15 meter, data dari sensor berhasil sampai ke server tanpa kehilangan paket data.
- d. Pada jarak 20 meter modul Xbee S2 mempunyai *packet loss* sebesar 3 %, sedangkan modul ES8266(12E) mempunyai *packet loss* sebesar 0 %.

Nilai *packet loss* dengan modul Xbee S2 pada jarak 20 meter terdapat *packet loss* sebesar 3 %, karena ada data yang tidak sampai atau hilang pada saat transmisi data dari ke server. Pada Gambar 4.11 menunjukkan jumlah data yang telah terkirim ke *database server* sebanyak 97 data, sedangkan jumlah data yang dikirim dari sensor sebanyak 100 data. Hal ini berarti ada 3 data yang telah hilang pada saat transmisi data.



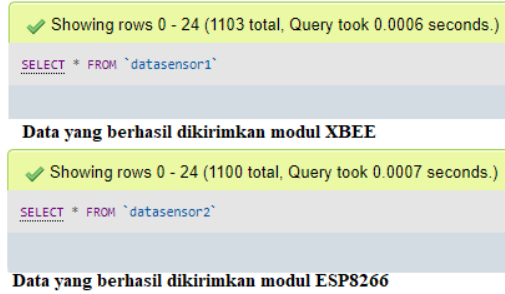
Gambar 4.11 Jumlah data yang dikirim Xbee S2

Dari data di atas membuktikan bahwa pengiriman data dengan modul ESP8266 dan Xbee S2 mempunyai kategori *packet loss* yang bagus, meskipun pada modul Xbee terdapat *packet loss* sebesar 3 % pada jarak transmisi 20 meter (Nasihin, et al., 2015).

#### 4.5. Pengujian pengiriman data

Dalam pengujian dilakukan pengiriman data dari sensor secara bersamaan dalam waktu hampir 2 jam, Xbee dapat mengirimkan 1103 data, sedangkan ESP8266 dapat

mengirimkan 1100 data. Hal ini dapat dilihat pada *database server* seperti yang ditunjukkan pada Gambar 4.12.



Gambar 4.12 Jumlah data yang dikirim ESP8266 dan Xbee

## BAB V

### PENUTUP

#### 5.1. Kesimpulan

Berdasarkan hasil analisis pengujian terhadap QoS dari sistem monitoring kualitas udara, maka dapat diambil kesimpulan sebagai berikut :

1. Sistem pemantau kuliatas udara dapat dijadikan sumber data pengujian kinerja modul Xbee S2 dan ESP8266(12E).
2. Parameter *Delay* pengiriman data dengan modul Xbee lebih kecil dari pada modul ESP8266.
3. Parameter *Throughput* pengiriman data dengan modul ESP8266 lebih besar dari pada modul Xbee.
4. Parameter *Packet loss* pengiriman data dengan modul ESP8266 dan Xbee dalam kondisi bagus karena nilai *packet loss* tidak lebih dari 3 %.

#### 5.2. Saran

Berikut adalah beberapa saran yang dapat digunakan untuk mengembangkan penelitian mengenai perbandingan kinerja antara modul Xbee dan ESP8266 .

1. Melakukan perbandingan pengujian QoS pada area In-door dan Out-door sehingga dapat menambah parameter pengujian untuk dianalis.
2. Melakukan pengujian QoS pada sistem deteksi sensor lainnya seperti pada monitoring suhu, kelembapan udara, dan lainnya.

## DAFTAR PUSTAKA

- Adi, P. D. P., Niswar, M. & Ilham, A. A., 2013. Kinerja Jaringan Sensor Nirkabel Ieee 802.15.4 Untuk Monitoring Denyut Nadi Pasien. *Universitas Satya Wiyata Mandala*.
- Ahamed, D., 2005. *The Role Of Zigbee Technoogy In Future Data Communication System*. S.L., Sathak Institute Of Technology.
- Alif, M. I. N., 2017. *Perancangan Sistem Komunikasi Menggunakan Xbee Pro S2 Ieee 802.15.4*. Yogyakarta, Universitas Gadjah Mada.
- Anggraini, D., Irawati, I. D. & Mayasari, R., 2014. *Analisis Dan Simulasi Wireless Sensor Network (Wsn) Untuk Komunikasi Data Menggunakan Protokol Zigbee*. S.L.:S.N.
- Dini, N. H., Munadi, R. & Suryadi, A., 2011. Analisis Performansi Wireless Sensor Network Dengan Mekanisme Cdma/Ca Pada Standar Ieee 802.15.4/Zigbee.
- Jusak, 2013. Implementasi Zigbee Ieee 802.15.4 Untuk Pemantauan Suhu Dan Kelembaban Udara. *Seminar Nasional Sistem & Teknologi Informasi (Snasti)* .
- Kumbhar, P. H., 2016. *Wireless Sensor Network Using Xbee On Arduino Platform*. Padmabhooshan Vasantdada Patil Institute Of Technology, Pune,India, Ieee.
- Munir, R., 2004. *Advanced Encryption Standard (Aes)*. Bandung: Departemen Teknik Informatika, Institut Teknologi Bandung.
- Nasihin, F. Z., Negara, A. B. P. & Irwansyah, A., 2015. Studi Perbandingan Performa Qos (Quality Of Service) Tunneling Protocol Pptp Dan L2tp Pada Jaringan Vpn Menggunakan
- Nur, A. D., 2014. Purwarupa Kendali Otomatis Kipas Pembuangan Berbasis Arduino Uno.
- Onno, P. W., 2005. *Infrastruktur Wireless Internet*. Yogyakarta: Andi.
- Phan Thi Anh Mai, J. K. N. M. D. F., 2014. *Cloud Databases For Internet-Of-Things Data*. Taipei, Ieee.
- Piyare, R. & Lee, S. R., 2013. Performance Analysis Of Xbee Zb Module Based Wireless Sensor Networks. *IEEE*, Iv(4).
- Thaker, T., 2016. Esp8266 Based Implementation Of Wireless Sensor Network With Linux Based Web-Server. *IEEE*.
- Yuliansyah, H., 2016. Uji Kinerja Pengiriman Data Secara Wireless Menggunakan Modul Esp8266 Berbasis Rest Architecture. *Jurnal Rekayasa Dan Teknologi Elektro*, 10(2).

## Lampiran

### A. Hasil Pengujian Xbee

#### 1. Delay jarak 5 meter

Waktu kirim

```

10/04/2018 20.03.02.35 [RX] - 35 36 0D 0A
10/04/2018 20.03.07.35 [RX] - 35 35 0D 0A
10/04/2018 20.03.12.36 [RX] - 35 35 0D 0A
10/04/2018 20.03.17.36 [RX] - 35 35 0D 0A
10/04/2018 20.03.22.36 [RX] - 35 35 0D 0A
10/04/2018 20.03.27.38 [RX] - 35 35 0D 0A
10/04/2018 20.03.32.38 [RX] - 35 35 0D 0A
10/04/2018 20.03.37.38 [RX] - 35 35 0D 0A
10/04/2018 20.03.42.38 [RX] - 35 35 0D 0A
10/04/2018 20.03.47.40 [RX] - 35 35 0D 0A
10/04/2018 20.03.52.40 [RX] - 35 35 0D 0A
10/04/2018 20.03.57.40 [RX] - 35 35 0D 0A
10/04/2018 20.04.02.40 [RX] - 35 35 0D 0A
10/04/2018 20.04.07.41 [RX] - 35 35 0D 0A
10/04/2018 20.04.12.41 [RX] - 35 35 0D 0A
10/04/2018 20.04.17.41 [RX] - 34 39 0D 0A
10/04/2018 20.04.22.41 [RX] - 36 33 0D 0A
10/04/2018 20.04.27.43 [RX] - 37 34 0D 0A
10/04/2018 20.04.32.43 [RX] - 37 33 0D 0A
10/04/2018 20.04.37.43 [RX] - 37 32 0D 0A
10/04/2018 20.04.42.43 [RX] - 37 32 0D 0A
10/04/2018 20.04.47.44 [RX] - 37 31 0D 0A
10/04/2018 20.04.52.44 [RX] - 37 30 0D 0A
10/04/2018 20.04.57.44 [RX] - 37 30 0D 0A
10/04/2018 20.05.02.45 [RX] - 37 30 0D 0A
10/04/2018 20.05.07.45 [RX] - 37 30 0D 0A

```

Waktu terima

```

10/04/2018 20.03.02.97 [RX] - 35 36 0D 0A
10/04/2018 20.03.07.95 [RX] - 35 35 0D 0A
10/04/2018 20.03.12.96 [RX] - 35 35 0D 0A
10/04/2018 20.03.17.97 [RX] - 35 35 0D 0A
10/04/2018 20.03.22.97 [RX] - 35 35 0D 0A
10/04/2018 20.03.27.98 [RX] - 35 35 0D 0A
10/04/2018 20.03.32.99 [RX] - 35 35 0D 0A
10/04/2018 20.03.37.97 [RX] - 35 35 0D 0A
10/04/2018 20.03.42.99 [RX] - 35 35 0D 0A
10/04/2018 20.03.48.00 [RX] - 35 35 0D 0A
10/04/2018 20.03.53.00 [RX] - 35 35 0D 0A
10/04/2018 20.03.58.01 [RX] - 35 35 0D 0A
10/04/2018 20.04.03.00 [RX] - 35 35 0D 0A
10/04/2018 20.04.08.01 [RX] - 35 35 0D 0A
10/04/2018 20.04.13.02 [RX] - 35 35 0D 0A
10/04/2018 20.04.18.01 [RX] - 34 39 0D 0A
10/04/2018 20.04.23.03 [RX] - 36 33 0D 0A
10/04/2018 20.04.28.04 [RX] - 37 34 0D 0A
10/04/2018 20.04.33.04 [RX] - 37 33 0D 0A
10/04/2018 20.04.38.03 [RX] - 37 32 0D 0A
10/04/2018 20.04.43.04 [RX] - 37 32 0D 0A
10/04/2018 20.04.48.05 [RX] - 37 31 0D 0A
10/04/2018 20.04.53.05 [RX] - 37 30 0D 0A

```

2. *Delay* jarak 10 meter

Waktu kirim

10/04/2018 19.49.31.30 [RX] - 36 33 0D 0A  
10/04/2018 19.49.36.30 [RX] - 36 32 0D 0A  
10/04/2018 19.49.41.32 [RX] - 36 32 0D 0A  
10/04/2018 19.49.46.32 [RX] - 36 32 0D 0A  
10/04/2018 19.49.51.32 [RX] - 36 31 0D 0A  
10/04/2018 19.49.56.32 [RX] - 36 31 0D 0A  
10/04/2018 19.50.01.33 [RX] - 36 31 0D 0A  
10/04/2018 19.50.06.33 [RX] - 36 31 0D 0A  
10/04/2018 19.50.11.33 [RX] - 36 30 0D 0A  
10/04/2018 19.50.16.35 [RX] - 36 30 0D 0A  
10/04/2018 19.50.21.35 [RX] - 36 30 0D 0A  
10/04/2018 19.50.26.35 [RX] - 36 30 0D 0A  
10/04/2018 19.50.31.35 [RX] - 35 39 0D 0A  
10/04/2018 19.50.36.36 [RX] - 35 39 0D 0A  
10/04/2018 19.50.41.36 [RX] - 35 39 0D 0A  
10/04/2018 19.50.46.37 [RX] - 35 39 0D 0A  
10/04/2018 19.50.51.36 [RX] - 35 39 0D 0A  
10/04/2018 19.50.56.38 [RX] - 35 39 0D 0A  
10/04/2018 19.51.01.38 [RX] - 35 38 0D 0A  
10/04/2018 19.51.06.38 [RX] - 35 39 0D 0A  
10/04/2018 19.51.11.40 [RX] - 35 39 0D 0A  
10/04/2018 19.51.16.40 [RX] - 35 38 0D 0A  
10/04/2018 19.51.21.39 [RX] - 35 38 0D 0A  
10/04/2018 19.51.26.39 [RX] - 35 38 0D 0A  
10/04/2018 19.51.31.40 [RX] - 35 38 0D 0A  
10/04/2018 19.51.36.40 [RX] - 35 38 0D 0A  
10/04/2018 19.51.41.41 [RX] - 35 38 0D 0A  
10/04/2018 19.51.46.40 [RX] - 35 38 0D 0A

Waktu terima

10/04/2018 19.49.31.90 [RX] - 36 33 0D 0A  
10/04/2018 19.49.36.91 [RX] - 36 32 0D 0A  
10/04/2018 19.49.41.91 [RX] - 36 32 0D 0A  
10/04/2018 19.49.46.91 [RX] - 36 32 0D 0A  
10/04/2018 19.49.51.91 [RX] - 36 31 0D 0A  
10/04/2018 19.49.56.93 [RX] - 36 31 0D 0A  
10/04/2018 19.50.01.93 [RX] - 36 31 0D 0A  
10/04/2018 19.50.06.94 [RX] - 36 31 0D 0A  
10/04/2018 19.50.11.93 [RX] - 36 30 0D 0A  
10/04/2018 19.50.16.93 [RX] - 36 30 0D 0A  
10/04/2018 19.50.21.95 [RX] - 36 30 0D 0A  
10/04/2018 19.50.26.95 [RX] - 36 30 0D 0A  
10/04/2018 19.50.31.96 [RX] - 35 39 0D 0A  
10/04/2018 19.50.36.97 [RX] - 35 39 0D 0A  
10/04/2018 19.50.41.95 [RX] - 35 39 0D 0A  
10/04/2018 19.50.46.96 [RX] - 35 39 0D 0A  
10/04/2018 19.50.51.97 [RX] - 35 39 0D 0A  
10/04/2018 19.50.56.98 [RX] - 35 39 0D 0A  
10/04/2018 19.51.01.98 [RX] - 35 38 0D 0A  
10/04/2018 19.51.06.98 [RX] - 35 39 0D 0A  
10/04/2018 19.51.11.97 [RX] - 35 39 0D 0A  
10/04/2018 19.51.16.98 [RX] - 35 38 0D 0A  
10/04/2018 19.51.21.99 [RX] - 35 38 0D 0A  
10/04/2018 19.51.27.01 [RX] - 35 38 0D 0A  
10/04/2018 19.51.32.01 [RX] - 35 38 0D 0A  
10/04/2018 19.51.37.01 [RX] - 35 38 0D 0A

3. Delay jarak 15 meter  
Waktu kirim

```
10/04/2018 20.17.03.72 [RX] - 36 38 0D 0A
10/04/2018 20.17.08.74 [RX] - 36 37 0D 0A
10/04/2018 20.17.13.74 [RX] - 36 37 0D 0A
10/04/2018 20.17.18.74 [RX] - 36 37 0D 0A
10/04/2018 20.17.23.74 [RX] - 36 37 0D 0A
10/04/2018 20.17.28.76 [RX] - 36 36 0D 0A
10/04/2018 20.17.33.76 [RX] - 36 36 0D 0A
10/04/2018 20.17.38.76 [RX] - 36 37 0D 0A
10/04/2018 20.17.43.76 [RX] - 36 37 0D 0A
10/04/2018 20.17.48.77 [RX] - 36 37 0D 0A
10/04/2018 20.17.53.77 [RX] - 36 36 0D 0A
10/04/2018 20.17.58.77 [RX] - 36 37 0D 0A
10/04/2018 20.18.03.79 [RX] - 36 36 0D 0A
10/04/2018 20.18.08.79 [RX] - 36 36 0D 0A
10/04/2018 20.18.13.79 [RX] - 36 36 0D 0A
10/04/2018 20.18.18.79 [RX] - 36 37 0D 0A
10/04/2018 20.18.23.80 [RX] - 36 37 0D 0A
10/04/2018 20.18.28.80 [RX] - 36 37 0D 0A
10/04/2018 20.18.33.80 [RX] - 36 37 0D 0A
10/04/2018 20.18.38.80 [RX] - 36 37 0D 0A
10/04/2018 20.18.43.81 [RX] - 36 36 0D 0A
10/04/2018 20.18.48.81 [RX] - 36 37 0D 0A
10/04/2018 20.18.53.81 [RX] - 36 37 0D 0A
10/04/2018 20.18.58.81 [RX] - 36 37 0D 0A
10/04/2018 20.19.03.83 [RX] - 36 37 0D 0A
10/04/2018 20.19.08.83 [RX] - 36 37 0D 0A
10/04/2018 20.19.13.83 [RX] - 36 36 0D 0A
```

Waktu terima

```
10/04/2018 20.17.04.34 [RX] - 36 38 0D 0A
10/04/2018 20.17.09.35 [RX] - 36 37 0D 0A
10/04/2018 20.17.14.36 [RX] - 36 37 0D 0A
10/04/2018 20.17.19.36 [RX] - 36 37 0D 0A
10/04/2018 20.17.24.37 [RX] - 36 37 0D 0A
10/04/2018 20.17.29.36 [RX] - 36 36 0D 0A
10/04/2018 20.17.34.37 [RX] - 36 36 0D 0A
10/04/2018 20.17.39.37 [RX] - 36 37 0D 0A
10/04/2018 20.17.44.38 [RX] - 36 37 0D 0A
10/04/2018 20.17.49.39 [RX] - 36 37 0D 0A
10/04/2018 20.17.54.40 [RX] - 36 36 0D 0A
10/04/2018 20.17.59.39 [RX] - 36 37 0D 0A
10/04/2018 20.18.04.40 [RX] - 36 36 0D 0A
10/04/2018 20.18.09.40 [RX] - 36 36 0D 0A
10/04/2018 20.18.14.40 [RX] - 36 36 0D 0A
10/04/2018 20.18.19.47 [RX] - 36 37 0D 0A
10/04/2018 20.18.24.41 [RX] - 36 37 0D 0A
10/04/2018 20.18.29.41 [RX] - 36 37 0D 0A
10/04/2018 20.18.34.43 [RX] - 36 37 0D 0A
10/04/2018 20.18.39.43 [RX] - 36 37 0D 0A
10/04/2018 20.18.44.43 [RX] - 36 36 0D 0A
10/04/2018 20.18.49.44 [RX] - 36 37 0D 0A
10/04/2018 20.18.54.43 [RX] - 36 37 0D 0A
10/04/2018 20.18.59.44 [RX] - 36 37 0D 0A
10/04/2018 20.19.04.44 [RX] - 36 37 0D 0A
```

4. Delay jarak 20 meter  
Waktu kirim

```
10/04/2018 20.29.01.95 [RX] - 36 37 0D 0A
10/04/2018 20.29.06.96 [RX] - 36 36 0D 0A
10/04/2018 20.29.11.97 [RX] - 36 36 0D 0A
10/04/2018 20.29.16.97 [RX] - 36 36 0D 0A
10/04/2018 20.29.21.97 [RX] - 36 36 0D 0A
10/04/2018 20.29.26.98 [RX] - 36 36 0D 0A
10/04/2018 20.29.31.97 [RX] - 36 35 0D 0A
10/04/2018 20.29.36.97 [RX] - 36 36 0D 0A
10/04/2018 20.29.41.97 [RX] - 36 36 0D 0A
10/04/2018 20.29.46.99 [RX] - 36 35 0D 0A
10/04/2018 20.29.51.99 [RX] - 36 35 0D 0A
10/04/2018 20.29.56.99 [RX] - 36 36 0D 0A
10/04/2018 20.30.01.99 [RX] - 36 35 0D 0A
10/04/2018 20.30.07.01 [RX] - 36 35 0D 0A
10/04/2018 20.30.12.01 [RX] - 36 36 0D 0A
10/04/2018 20.30.17.01 [RX] - 36 35 0D 0A
10/04/2018 20.30.22.02 [RX] - 36 34 0D 0A
10/04/2018 20.30.27.02 [RX] - 36 35 0D 0A
10/04/2018 20.30.32.02 [RX] - 36 35 0D 0A
10/04/2018 20.30.37.02 [RX] - 36 35 0D 0A
10/04/2018 20.30.42.04 [RX] - 36 35 0D 0A
10/04/2018 20.30.47.04 [RX] - 36 35 0D 0A
10/04/2018 20.30.52.04 [RX] - 36 35 0D 0A
10/04/2018 20.30.57.04 [RX] - 36 35 0D 0A
10/04/2018 20.31.02.05 [RX] - 36 35 0D 0A
10/04/2018 20.31.07.05 [RX] - 36 35 0D 0A
```

Waktu terima

```
10/04/2018 20.29.02.58 [RX] - 67<CR><LF>
10/04/2018 20.29.07.58 [RX] - 66<CR><LF>
10/04/2018 20.29.12.58 [RX] - 66<CR><LF>
10/04/2018 20.29.17.59 [RX] - 66<CR><LF>
10/04/2018 20.29.22.60 [RX] - 66<CR><LF>
10/04/2018 20.29.27.61 [RX] - 66<CR><LF>
10/04/2018 20.29.32.60 [RX] - 65<CR><LF>
10/04/2018 20.29.37.61 [RX] - 66<CR><LF>
10/04/2018 20.29.42.61 [RX] - 66<CR><LF>
10/04/2018 20.29.47.62 [RX] - 65<CR><LF>
10/04/2018 20.29.52.62 [RX] - 65<CR><LF>
10/04/2018 20.29.57.61 [RX] - 66<CR><LF>
10/04/2018 20.30.02.62 [RX] - 65<CR><LF>
10/04/2018 20.30.07.63 [RX] - 65<CR><LF>
```

5. *Packet loss jarak 5 meter*

```

✓ Showing rows 0 - 24 (100 total, Query took 0.0010 seconds.)

SELECT * FROM `datasensor` ORDER BY `waktu` DESC
.....

```

6. *Packet loss jarak 10 meter*

```

✓ Showing rows 0 - 24 (100 total, Query took 0.0010 seconds.)

SELECT * FROM `datasensor` ORDER BY `waktu` DESC
.....

```

7. *Packet loss jarak 15 meter*

```

✓ Showing rows 0 - 24 (100 total, Query took 0.0010 seconds.)

SELECT * FROM `datasensor` ORDER BY `waktu` DESC
.....

```

8. *Packet loss jarak 20 meter*

```

✓ Showing rows 0 - 24 (100 total, Query took 0.0010 seconds.)

SELECT * FROM `datasensor` ORDER BY `waktu` DESC
.....

```

B. Hasil pengujian EPS8266

1. Pengujian jarak 5 meter

9	7.216110	192.168.137.1	192.168.137.43	TCP	66	0.000000000	56264 → 80	[SYN] Seq=0 Win=17520 Len=0 MSS=1460 WS=256 SACK_PERM=1
10	7.217823	192.168.137.43	192.168.137.1	TCP	58	0.001713000	80 → 56264	[SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
11	7.217945	192.168.137.1	192.168.137.43	TCP	54	0.000122000	56264 → 80	[ACK] Seq=1 Ack=1 Win=17520 Len=0
12	7.218011	192.168.137.1	192.168.137.43	HTTP	117	0.000066000	GET /data HTTP/1.0	
13	7.226495	192.168.137.43	192.168.137.1	TCP	54	0.000484000	80 → 56264	[ACK] Seq=1 Ack=64 Win=5777 Len=0
14	12.223803	192.168.137.43	192.168.137.1	TCP	169	4.996508000	80 → 56264	[PSH, ACK] Seq=1 Ack=64 Win=5777 Len=115 [TCP segment of a reassembled ...
15	12.263962	192.168.137.1	192.168.137.43	TCP	54	0.040959000	56264 → 80	[ACK] Seq=64 Ack=116 Win=17405 Len=0
16	12.265515	192.168.137.43	192.168.137.1	HTTP	58	0.001553000	HTTP/1.1 200 OK (text/plain)	
17	12.310683	192.168.137.1	192.168.137.43	TCP	54	0.045168000	56264 → 80	[ACK] Seq=64 Ack=120 Win=17401 Len=0

Frame 12: 117 bytes on wire (936 bits), 117 bytes captured (936 bits) on interface 0  
 Interface id: 0 (Device\NPF\_{1ED65702-0C88-4584-AE48-58C57CC56593})  
 Encapsulation type: Ethernet (1)  
 Arrival Time: Apr 14, 2018 15:40:35.948286000 SE Asia Standard Time  
 [Time shift for this packet: 0.000000000 seconds]  
 Epoch Time: 1523695235.948286000 seconds  
 [Time delta from previous captured frame: 0.000066000 seconds]  
 [Time delta from previous displayed frame: 0.000066000 seconds]  
 [Time since reference or first frame: 7.218011000 seconds]  
 Frame Number: 12  
 Frame Length: 117 bytes (936 bits)  
 Capture Length: 117 bytes (936 bits)

Wireshark · Capture File Properties · 5m

Details

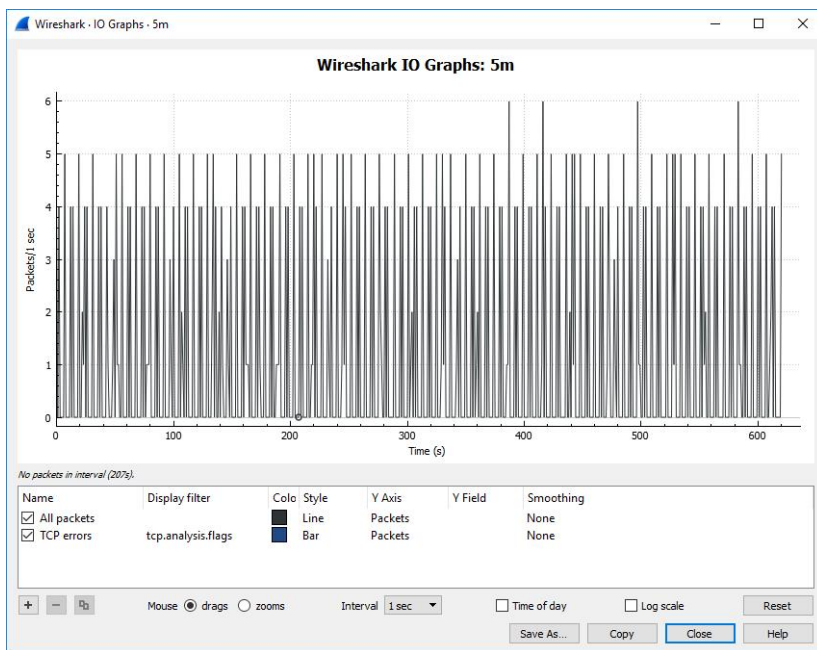
\NPF\_{1ED65702-0C88-4584-AEAB-58C57CC56593} bytes

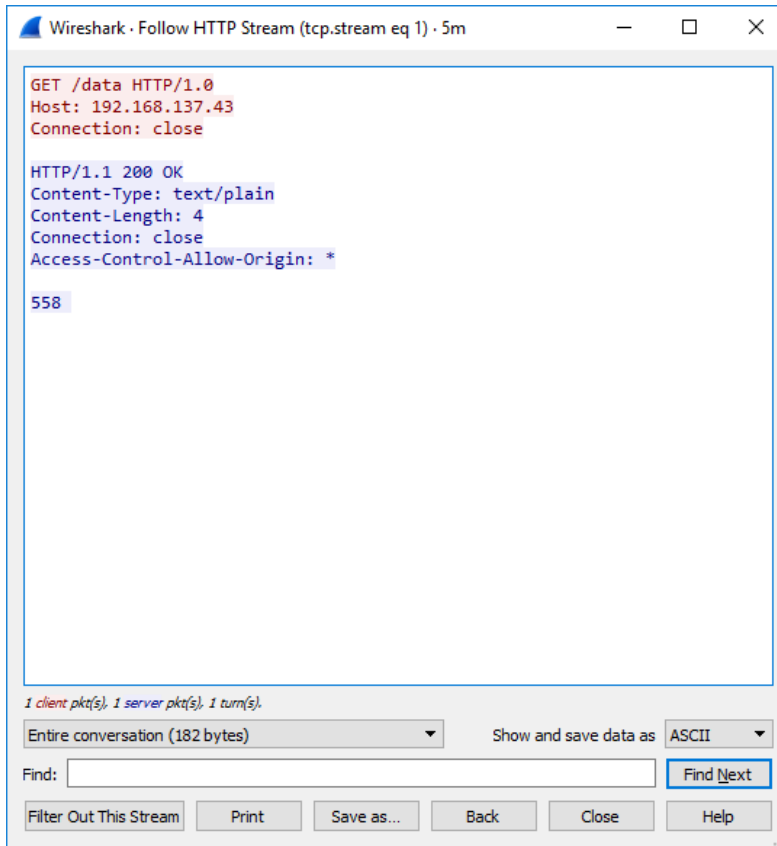
**Statistics**

Measurement	Captured	Displayed	Marked
Packets	734	13 (1.8%)	—
Time span, s	620.186	7.101	—
Average pps	1.2	1.8	—
Average packet size, B	73.5	69.5	—
Bytes	53603	900 (1.7%)	0
Average bytes/s	86	126	—
Average bits/s	691	1013	—

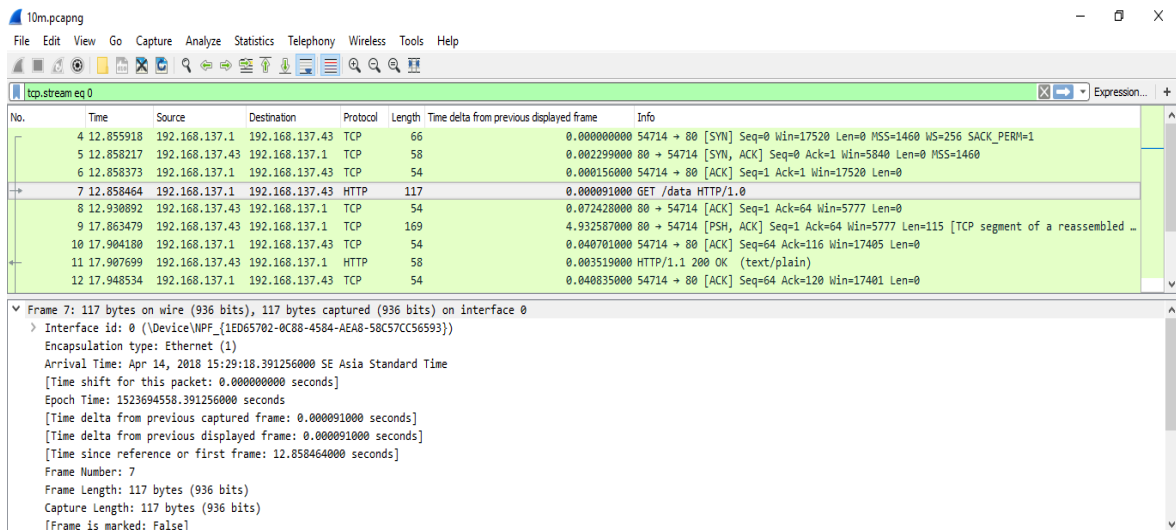
Capture file comments

Refresh Save Comments Close Copy To Clipboard Help





## 2. Pengujian jarak 10 meter



Wireshark · Capture File Properties · 10m

Details

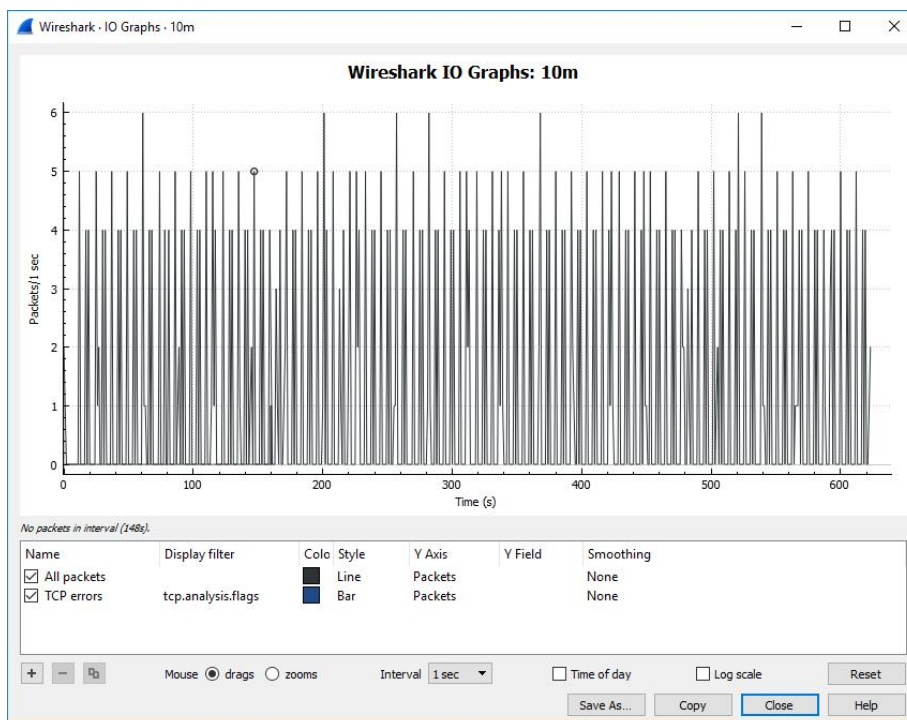
WPF\_{1ED65702-0C88-4584-AEA8-58C57CC56593} bytes

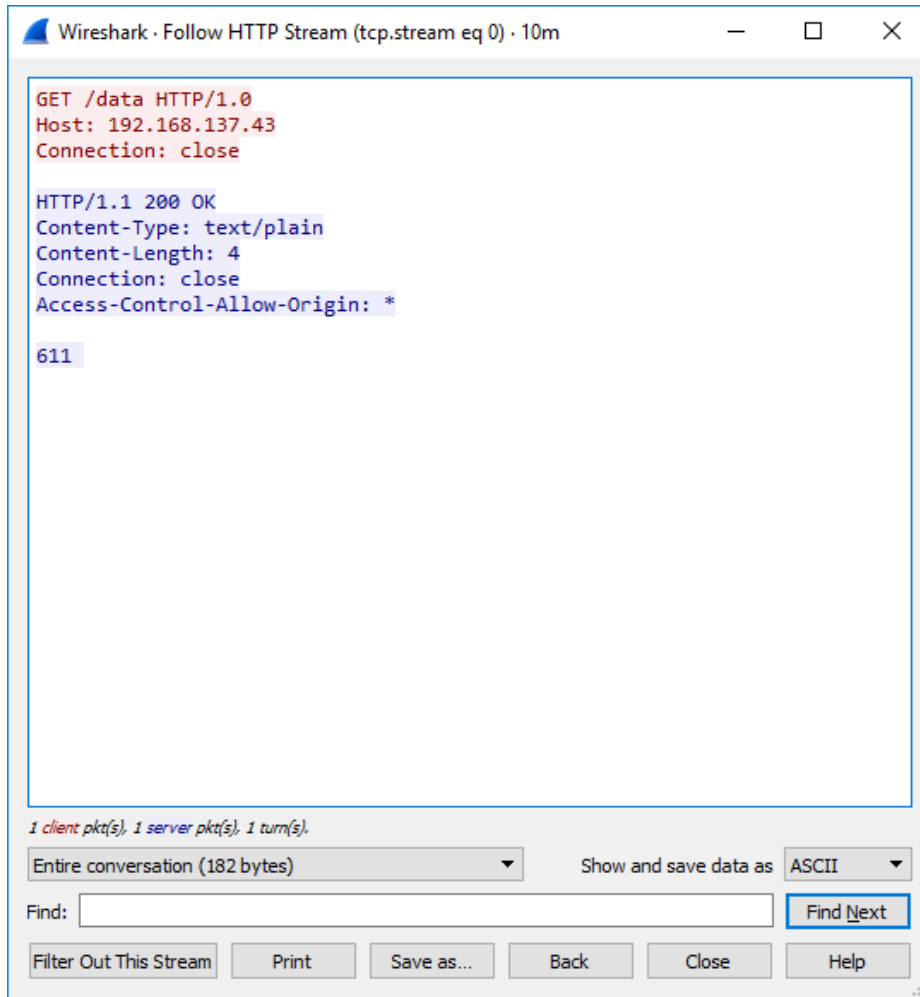
**Statistics**

Measurement	Captured	Displayed	Marked
Packets	725	13 (1.8%)	—
Time span, s	623.844	7.099	—
Average pps	1.2	1.8	—
Average packet size, B	73.5	69.5	—
Bytes	53096	900 (1.7%)	0
Average bytes/s	85	126	—
Average bits/s	680	1014	—

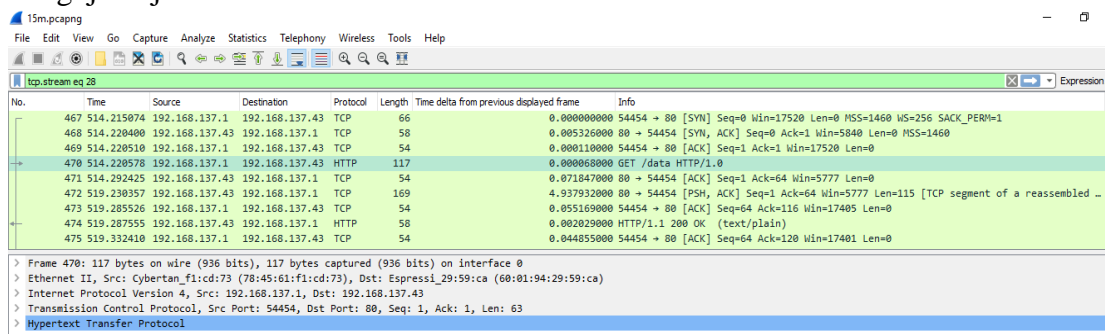
Capture file comments

Refresh Save Comments **Close** Copy To Clipboard Help





### 3. Pengujian jarak 15 meter



Wireshark · Capture File Properties · 15m

Details

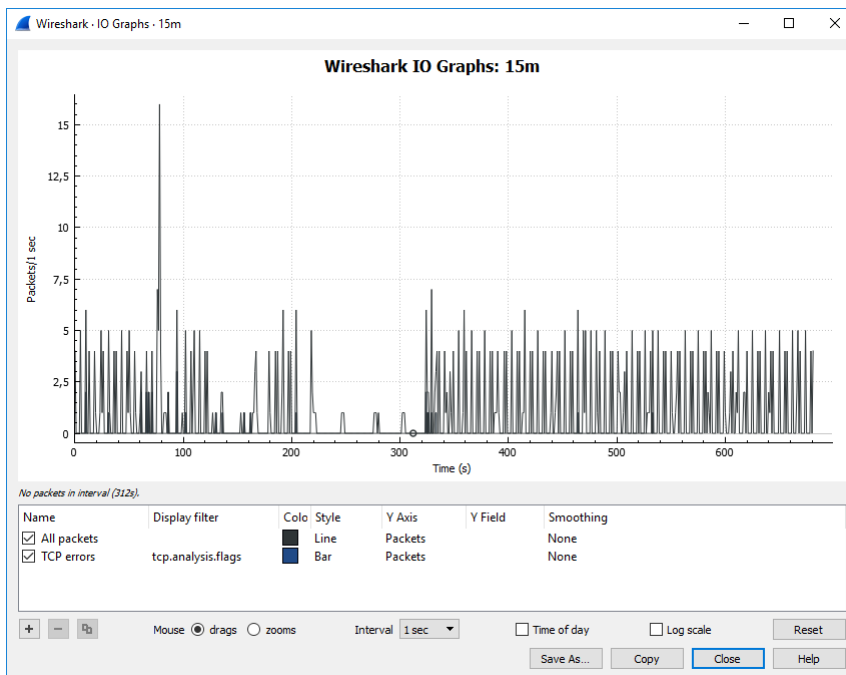
\NPF\_{1ED65702-0C88-4584-AEA8-58C57CC56593} bytes

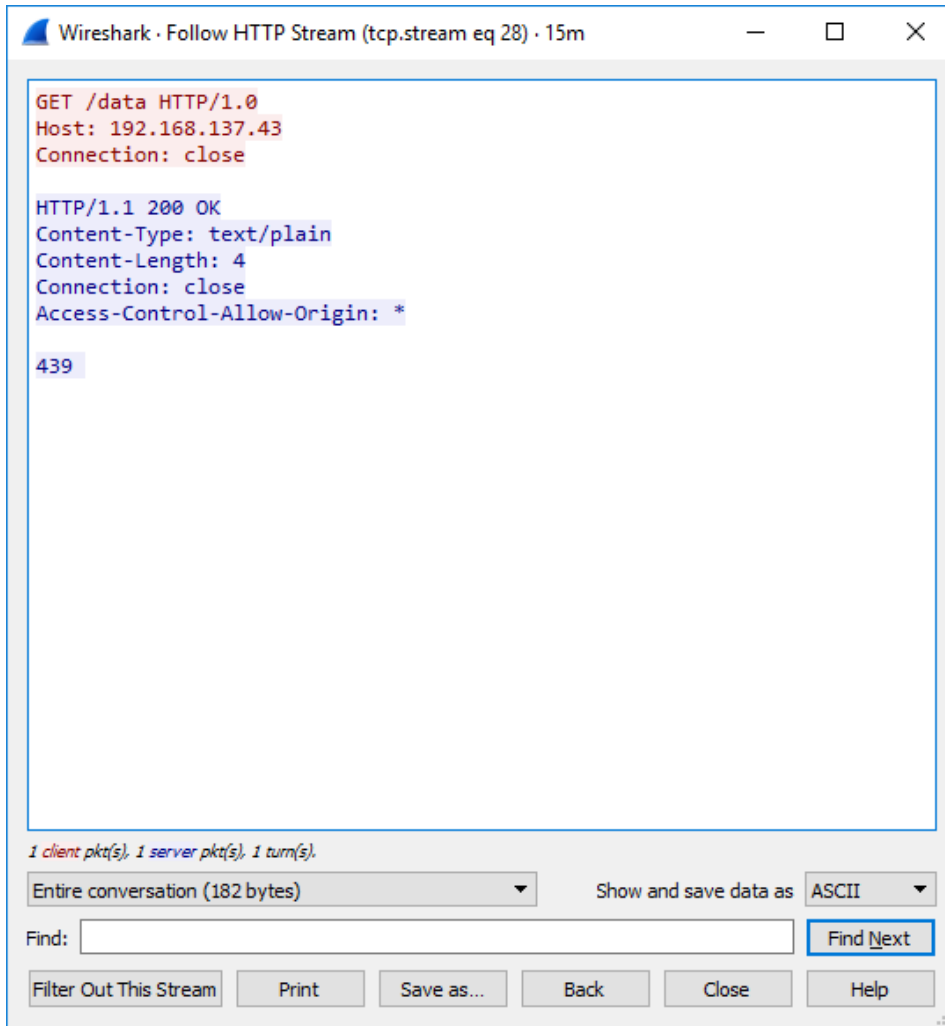
**Statistics**

Measurement	Captured	Displayed	Marked
Packets	667	13 (1.9%)	—
Time span, s	681.702	7.136	—
Average pps	1.0	1.8	—
Average packet size, B	114.5	69.5	—
Bytes	76314	900 (1.2%)	0
Average bytes/s	111	126	—
Average bits/s	895	1008	—

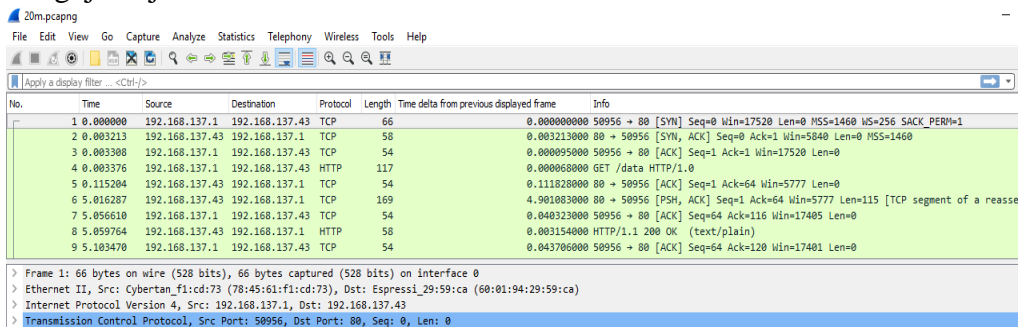
Capture file comments

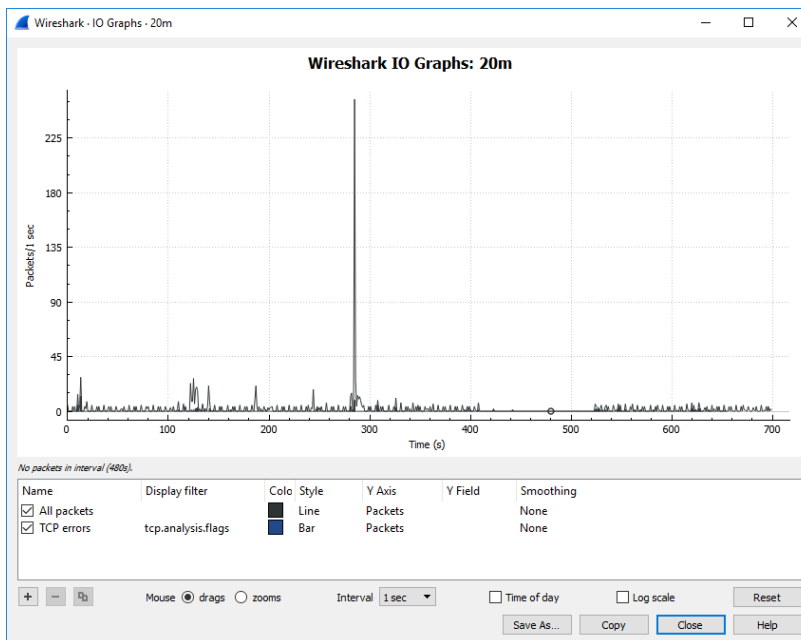
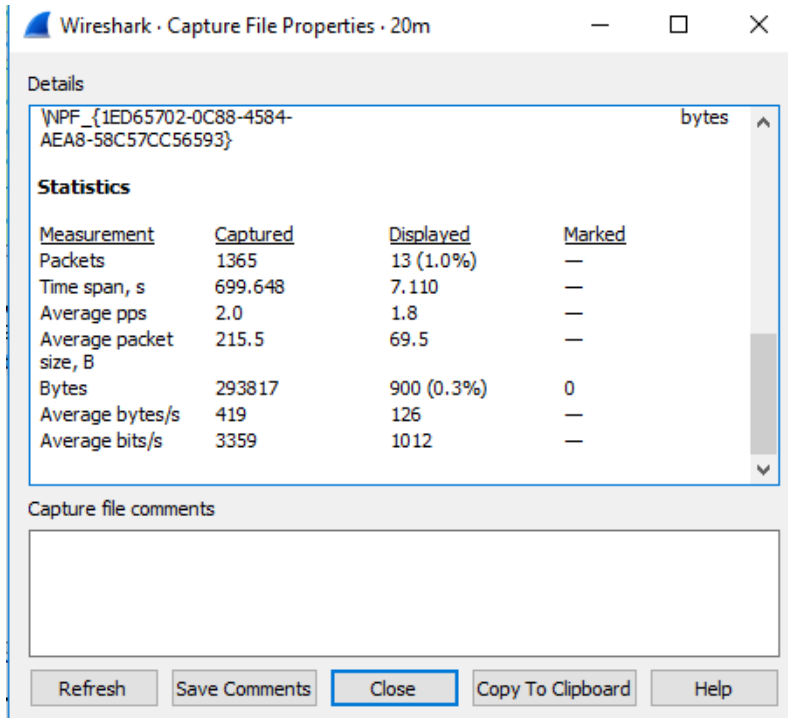
Refresh Save Comments Close Copy To Clipboard Help

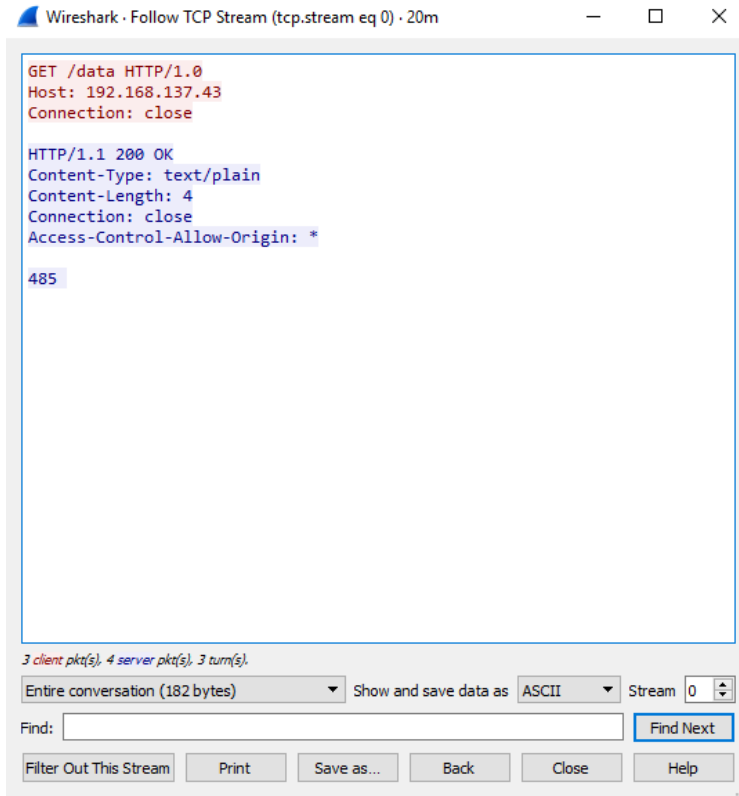




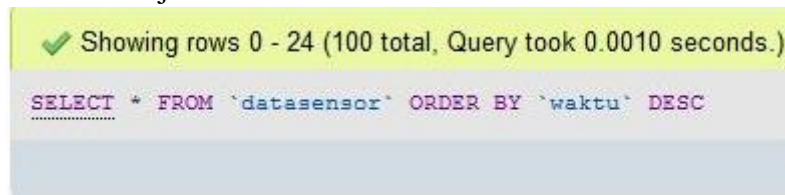
#### 4. Pengujian jarak 20 meter



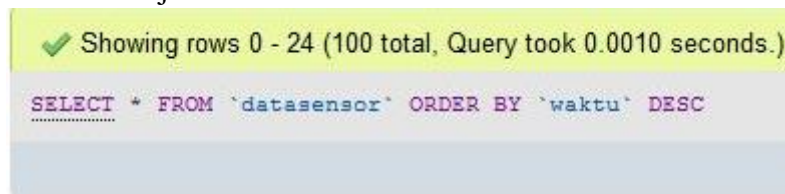




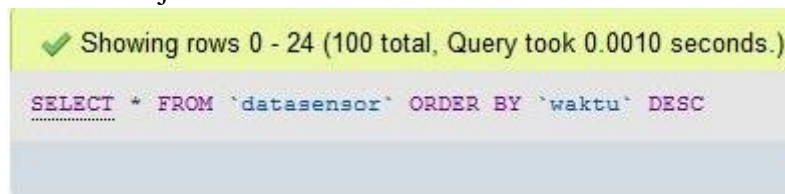
5. *Packet loss jarak 5 meter*



6. *Packet loss jarak 10 meter*



7. *Packet loss jarak 15 meter*



## 8. *Packet loss* jarak 20 meter

```
Showing rows 0 - 24 (100 total, Query took 0.0010 seconds.)  
  
SELECT * FROM `datasensor` ORDER BY `waktu` DESC
```

## C. Pembuatan *database*

```
Showing rows 0 - 24 (100 total, Query took 0.0010 seconds.) [waktu: 2018-04-21 20:00:47... - 2018-04-21 19:58:44...]  
  
SELECT * FROM `datasensor` ORDER BY `waktu` DESC
```

Profiling [ Ed

1 > >> |  Show all | Number of rows: 25 | Filter rows: Search this table | Sort

+ Options

	id	nilai	waktu
<input type="checkbox"/> Edit Copy Delete	6823	251	2018-04-21 20:00:47
<input type="checkbox"/> Edit Copy Delete	6822	252	2018-04-21 20:00:42
<input type="checkbox"/> Edit Copy Delete	6821	253	2018-04-21 20:00:37
<input type="checkbox"/> Edit Copy Delete	6820	255	2018-04-21 20:00:32
<input type="checkbox"/> Edit Copy Delete	6819	255	2018-04-21 20:00:27
<input type="checkbox"/> Edit Copy Delete	6818	256	2018-04-21 20:00:22
<input type="checkbox"/> Edit Copy Delete	6817	255	2018-04-21 20:00:17
<input type="checkbox"/> Edit Copy Delete	6816	256	2018-04-21 20:00:12
<input type="checkbox"/> Edit Copy Delete	6815	257	2018-04-21 20:00:06
<input type="checkbox"/> Edit Copy Delete	6814	258	2018-04-21 20:00:01

## D. Program pada Sensor

### 1. Modul Xbee

```
#include <MQ135.h>  
  
void setup () {  
  Serial.begin(9600);  
}
```

```
void loop() {  
  
  MQ135 gasSensor = MQ135(A0);  
  
  float udara = gasSensor.getPPM();  
  
  Serial.println (udara);  
  
  delay(6000);  
  
}
```

## 2. Modul ESP8266

```
#include "MQ135.h"  
  
#include <ESP8266WiFi.h>  
  
#include <WiFiClient.h>  
  
#include <ESP8266WebServer.h>  
  
//#define ANALOGPIN A0  
  
const char* ssid = "maniacyl";  
  
const char* password =  
"0987654321";  
  
ESP8266WebServer server(80);  
  
float val;  
  
float udara;  
  
float ppm;  
  
String webString="";  
  
void handle_root() {  
  
  server.send(200, "text/plain",  
  "monitoring kualitas udara ");
```

```
delay(100);  
  
}  
  
void ambiludara() {  
  
  val = analogRead(A0);  
  
  MQ135 gasSensor = MQ135(A0);  
  
  float udara = gasSensor.getPPM();  
  
  ppm=udara;  
  
  float rzero =  
  gasSensor.getRZero();  
  
  delay(5000);  
  
}  
  
void setup(void)  
{  
  
  Serial.begin(9600);  
  
  WiFi.begin(ssid, password);
```

```
Serial.print("\n\r\n\rWorking to
connect");

while (WiFi.status() !=
WL_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("loading to server");

Serial.print("Connected to ");

Serial.println(ssid);

Serial.print("IP address: ");

Serial.println(WiFi.localIP());

server.on("/", handle_root);

server.on("/data", []){
```

```
ambiludara(); // baca data dari
sensor

webString="" +String((int)ppm)+
";

server.send(200, "text/plain",
webString);

});

server.begin();

Serial.println("HTTP server
started");

}

void loop(void){

server.handleClient();

}
```

## E. Program menyimpan data ke *database*

### 1. Modul Xbee

```
import serial

import MySQLdb

db =
MySQLdb.Connect("localhost","root","","udara")

cursor = db.cursor()

port = 'COM3'
```

```
print "Connecting...",port

arduino = serial.Serial(port, 9600)

print "arduino detected"

while True:

    data = arduino.readline()

    cursor=db.cursor()

    cursor.execute("INSERT INTO datasensor
(nilai) VALUES (%s)",(data))

    db.commit()

    cursor.close()
```

## 2. Modul ESP8266

```
<head>

<meta http-equiv="refresh" content="5">

</head>

<?php

include ('koneksi.php');

$data=
file_get_contents('http://192.168.137.155/data');

$stampil = str_ireplace(,$data);

mysqli_query($conn, "insert into $namatabel
(nilai) VALUES (" . $stampil . ")");

echo $stampil;
```

## F. Program Website

### 1. Koneksi ke database

```
<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "udara";

$namatabel = "datasensor";

$conn = mysqli_connect($servername,

$username, $password, $dbname);
```

### 2. Grafik dan tabel

```
<!DOCTYPE html>
<html>
<head>
<title>Sistem Pemantau Kualitas
Udara</title>
<meta name="viewport"
content="width=device-width,
initial-scale=1.0"/>
<link rel="stylesheet"
href="css/bootstrap.min.css">
<meta name="viewport"
content="width=device-width,
initial-scale=1.0">
</head>
<body>
<div class="container ">
<h1>Sistem Pemantau Kualitas
Udara</h1>
<script
src="chartjs/Chart.min.js"></script>
<canvas
id="grafikBatang"></canvas>

<div class="row">
<div class="col-md-12">
```

```
<button type="button" class="btn
btn-info pull-right"
onClick="getData()">Refresh</butto
n>
</div>
</div>
<?php
include "koneksi.php";
$nilai=mysqli_query($conn,"select *
from datasensor order by waktu desc
limit 10");
?>
<table class="table table-bordered
table-striped table-hover">
<thead>
<tr class="danger">
<th>Waktu</th>
<th>Data Sensor</th>
<th>Indikator</th>
</tr>
</thead>
<tbody id="datatable">
<?php
include "koneksi.php";
```

```

        $sql=mysqli_query($conn,"select * from datasensor order by waktu desc limit 10");
        foreach ($sql as $row){
            if($row['nilai']>150){
                $indikator="tidak normal";
            }
            else {
                $indikator="normal";
            }
            echo "<tr class='info'>
            <td>".$row['waktu']."</td>
            <td>".$row['nilai']."</td>
            <td>".$indikator."</td>
            </tr>";
        }
    ?>

</tbody>
</table>
</div><br /><br />
<script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
var ctx =
document.getElementById("grafikBatang").getContext('2d');
var myChart = new Chart(ctx, {
type: 'line',
data: {
labels: ["1s", "2s", "3s", "4s", "5s", "6s", "7s", "8s", "9s", "10s"],
datasets: [{label: '# PPM',data:
[<?php while ($b =
mysqli_fetch_array($nilai)) { echo ""
. $b['nilai'] . ""';}>],
        backgroundColor: [
            'rgba(255, 99, 132, 0.2)',
            'rgba(54, 162, 235, 0.2)',
            'rgba(255, 206, 86, 0.2)',
            'rgba(75, 192, 192, 0.2)',
            'rgba(153, 102, 255, 0.2)',
            'rgba(255, 159, 64, 0.2)'
        ],
        borderColor: [
            'rgba(255,99,132,1)',
            'rgba(54, 162, 235, 1)',
            'rgba(255, 206, 86, 1)',

```

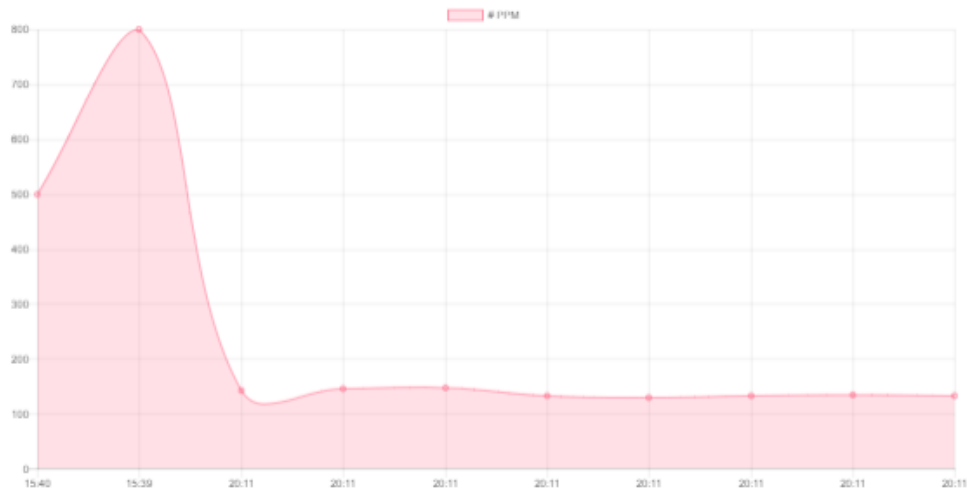
```

            'rgba(75, 192, 192, 1)',
            'rgba(153, 102, 255, 1)',
            'rgba(255, 159, 64, 1)'
        ],
        borderWidth: 1}}
    },options: {
        scales: {yAxes: [{
            ticks: {beginAtZero:true
            }}}}});
function getData(){
$.ajax({type : 'GET', url :
'api/data.php',
dataType : 'json',success :
function(data){
myChart.data.datasets[0].data = [];
myChart.data.labels =
[];$('#datatable').html("");
for(var i =0; i<data.length;i++){
myChart.data.datasets[0].data.push(
data[i].nilai);
myChart.data.labels.push(data[i].waktu);
$('#datatable').append('<tr><td>'+data[i].waktu+
'</td><td>'+data[i].nilai+
'</td><td>'+data[i].indikator+'</td></tr>');
myChart.update();}});
$(document).ready(function(){
getData();
setInterval(function(){ getData(); },
6000);
});
</script>
</body>
</html>

```

## G. Tampilan Web

### Sistem Pemantau Kualitas Udara



Refresh

Waktu	Data Sensor	Indikator
2018-05-04 15:40:12	500	Normal
2018-05-04 15:39:57	800	Tidak Normal
2018-04-21 20:11:51	142	Normal
2018-04-21 20:11:46	145	Normal
2018-04-21 20:11:40	147	Normal
2018-04-21 20:11:35	133	Normal
2018-04-21 20:11:30	130	Normal
2018-04-21 20:11:25	131	Normal