IMPLEMENTASI DAN ANALISIS PROTOKOL MODBUS TCP PADA SMART BUILDING BERBASIS OPENMTC

MODBUS TCP IMPLEMENTATION AND ANALYZE FOR SMART BUILDING BASED ON OPENMTC

[1] Ichsan Edi Putranto [2] Dr. Maman Abdurohman, S.T., M.T. [3] Anton Herutomo ST, M.Eng
[1 2 3] Fakultas Informatika, Universitas Telkom, Bandung
[1]ichsan.edi@gmail.com [2] m_abdurohman@yahoo.com [3] anton.herutomo@gmail.com

Abstrak

Modbus TCP merupakan protokol yang saat ini telah digunakan dalam industri besar untuk menangani komunikasi antara sensor dengan *PLC (Programmable Logic Controller)*. Protokol dalam penelitian ini di implementasikan dalam jaringan *smart building* dengan menggunakan media transmisi kabel dan nirkabel. Parameter penelitian yang digunakan yaitu *round trip time delay, protocol overhead,* dan *packet loss*.

Kata kunci : Modbus TCP, Smart Building, OpenMTC, *round trip time delay, packet loss, protocol overhead*

Abstract

Modbus TCP is a protocol that is currently used in large industries to handle communication between the sensor with a PLC (Programmable Logic Controller). In the study protocol implemented in smart building networks using wired and wireless transmission media. Parameters used in this research are round trip time delay, protocol overhead, and packet loss.

Keywords: Modbus TCP, Smart Building, OpenMTC, round trip time delay, packet loss, protocol overhead

1 Pendahuluan

1.1 Latar Belakang

Internet of things merupakan teknologi penerus dari M2M yang dikembangkan sekitar tahun 2000-an. Teknologi ini kedepannya akan menjadi sebuah hal yang umum digunakan.Ada beberapa negara yang telah mengaplikasikan teknologi ini di negaranya, contohnya adalah negara Eropa. Dalam perkembangannya internet of things ini juga dikembangkan dalam area Smart Building, Smart City, dan berbagai bidang yang lain.

Pada konsep Smart Building, penggunaan energi yang seefisien dan sehemat mungkin merupakan salah satu syaratnya, sehingga penggunaan protokol komunikasi juga harus yang mempunyai spesifikasi yang seperti itu. Protokol yang sudah sering digunakan dalam menangani manajemen sensor ini yaitu Modbus. Modbus ini merupakan singkatan dari Modicon Bus. Protokol ini dikembangkan pertama kali pada akhir tahun 1970-an oleh perusahaan Modicon, yang sekarang telah berubah menjadi Schneider Automation. Teknologi ini diberikan secara cuma-cuma oleh perusahaan itu, sehingga tingkat pemakaian saat itu sangat tinggi untuk sensor yang digunakan di bidang industri. Karena kehandalannya, protokol ini dikembangkan untuk digunakan dalam bidang Smart Building.

2 Dasar Teori

2.1 Platform

Platform adalah sebuah jembatan penghubung antara perangkat lunak dengan perangkat keras. Contoh yang paling sering ditemui adalah pada penggunaan PC (*Personal Computer*) atau laptop, yang merupakan sebuah jembatan penghubung adalah sistem operasi yang digunakan. Untuk pengembangan sistem *Smart Building*, *platform* yang sering digunakan adalah OpenMTC. Salah satu keunggulan OpenMTC ini adalah fitur *openness* yang dimiliki setiap sistem terdistribusi, yang dapat dikembangkan dengan menggunakan bahasa pemrograman *Javascript, Java,* atau *Ruby*.

2.2 Smart Building

Salah satu pilot proyek yang telah dilakukan dalam bidang *Smart Building* ini adalah proyek milik Microsoft. Microsoft bekerja sama dengan Accenture, membangun sebuah bangunan yang digunakan oleh salah satu anak perusahaan milik Microsoft yaitu pada bagian *Real Estate and Facilities*.

Proyek ini diterapkan pada salah satu gedung milik Microsoft, dalam tahap awal proyek ini, program awalnya yaitu :

1. *Fault Detection and Diagnosis*

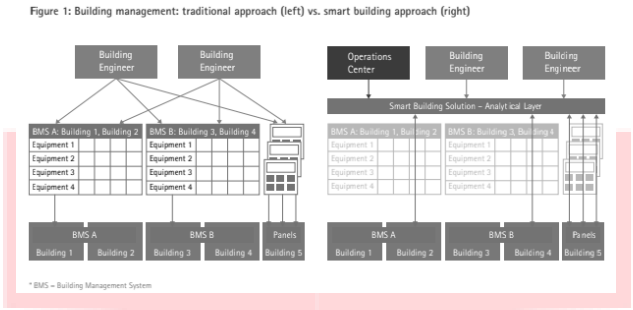
Program ini bertujuan untuk mengatasi suatu masalah yang terjadi secara tepat waktu dan tepat sasaran.

2. *Alarm Management*

   Program ini bertujuan untuk memberikan notifikasi pada bagian perawatan untuk memilah-milah masalah mana saja yang mempunyai dampak paling besar dari sistem sehingga harus segera untuk ditindak lanjuti.

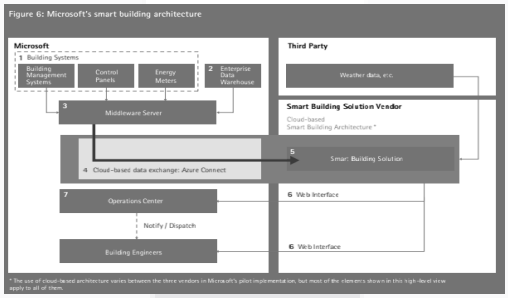
3. *Energy Management*

   Program ini bertujuan untuk memberikan sistem *tracking* dan optimisasi dalam penggunaan energi gedung tersebut.



Gambar 2- 1 Pendekatan antara manajemen gedung tradisional dengan Smart Building

Gambar tersebut menjelaskan perbandingan dari manajemen bangunan dari pendekatan tradisional dan pendekatan *Smart Building*. Berdasarkan gambar tersebut, menurut saya dapat diambil kesimpulan yaitu dengan menggunakan pendekatan *Smart Building* ada bagian yang dapat dihilangkan dan menambahkan bagian yang baru yaitu pada gambar bagian tengah pada pendekatan yang tradisional yang dapat dihilangkan yaitu bagian BMS, dan pada pendekatan *Smart Building* ditambahkan layer baru yaitu *Smart Building Solution* dan *Operation Center* sebagai tambahan pengguna.



Gambar 2-2 Arsitektur Smart Building menurut Microsoft

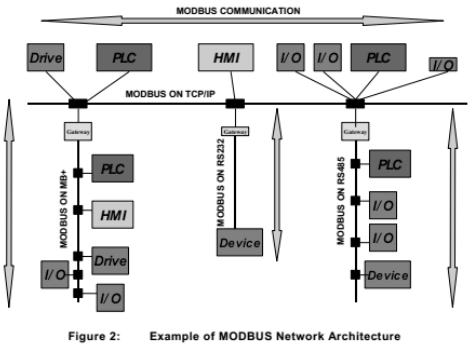
Gambar 2-2 adalah arsitektur yang digunakan dalam pembuatan aplikasi tersebut[4]

2.3 NodeJS

NodeJS adalah sebuah *platform* yang dibangun menggunakan Javascript untuk membangun dan mengembangkan aplikasi berbasis jaringan. Nodejs digunakan karena mempunyai beberapa kelebihan yaitu menggunakan non-blocking I/O sehingga membuat NodeJS ringan dan efisien, kemudian baik untuk digunakan dalam penggunaan data intensif secara *real time* yang berjalan dalam sistem terdistribusi.[9]

2.4 Protokol Serial Modbus

Protokol Modbus adalah struktur komunikasi yang dikembangkan oleh Modicon pada tahun 1979. Protokol ini digunakan untuk memberikan standar komunikasi *master/slave* atau *client/server* diantara beberapa alat pintar. Protokol ini telah dipakai oleh banyak industri, khususnya pabrik-pabrik yang menggunakan mesin otomatis. [8]



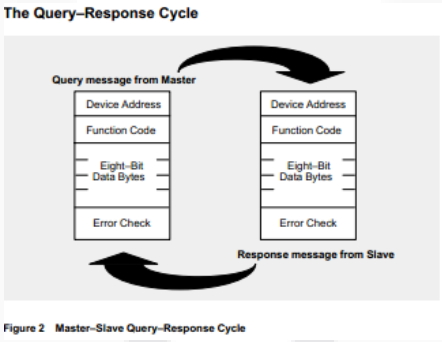
Gambar 2-3 Arsitektur Protokol Modbus

Gambar diatas merupakan arsitektur yang digunakan untuk protokol Modbus.

Dalam format pengiriman *master* ke *slave* terdapat 2 mode pengalamatan yang digunakan yaitu :

- *Unicast mode*
  Dalam mode ini, *master* akan mengirimkan *query* ke satu *slave*, setelah diterima oleh *slave* dan diproses, *slave* tersebut akan memberikan respon kepada *master*.
- *Broadcast mode*
  Dalam mode ini, *master* akan mengirimkan *query* ke setiap *slave*, dan *slave* tersebut tidak memberikan respon, sehingga hanya menerima *query* saja.

Kemudian untuk siklus *Query-Response* seperti gambar berikut :



Gambar 2-4 Master – Slave Query Response Cycle

2.5 Machine to Machine (M2M)

Machine to machine adalah komunikasi yang dilakukan oleh berbagai barang atau alat-alat yang mempunyai kemampuan untuk saling berkomunikasi melalu media internet atau media koneksi yang lainnya. Teknologi dikembangkan untuk memberikan kecerdasan buatan ke dalam suatu alat atau barang sehari-hari, sehingga bisa berkomunikasi dengan manusia melalui bantuan aplikasi yang berhubungan langsung dengan manusia. [3]

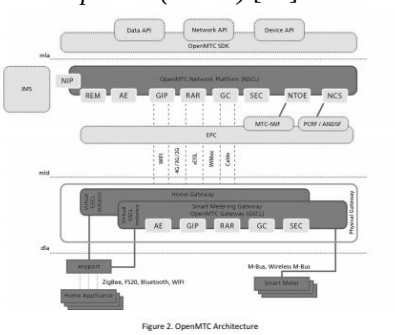
2.6 OpenMTC

OpenMTC adalah sebuah prototipe *platform* yang dibuat untuk memenuhi kebutuhan komunikasi *M2M* dan *Smart City*. *Platform* ini berfungsi sebagai *middleware*, sehingga diharapkan akan melayani berbagai macam *vendor* pembuat alat *M2M* atau sensor-sensor dan yang lain.

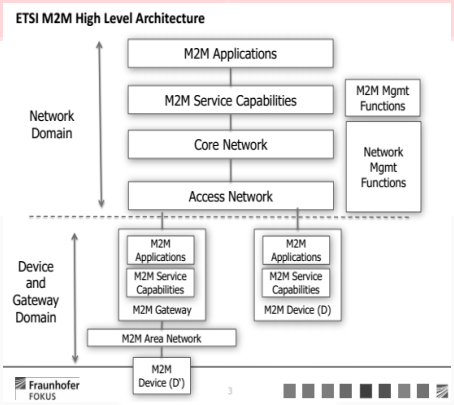
OpenMTC terdiri dari 2 lapisan kapabilitas, yaitu *Gateway Service Capability Layer (GSCL)* dan *Network Service Capability Layer (NSCL).* Kemudian OpenMTC juga telah mendukung kapabilitas ETSI yaitu :

- *Generic Communication (xGC)*
- *Application Enablement (xAE)*
- *Reachability, Addressing and Repository (xRAR)*
- *Remote Entity Management (xREM)*

- *Interworking Proxy (xIP)*
- *Security Capability (xSC)*
- *Network Communication Selection (NCS)*
- *Network Telco Operator Exposure (NTOE)* [12]



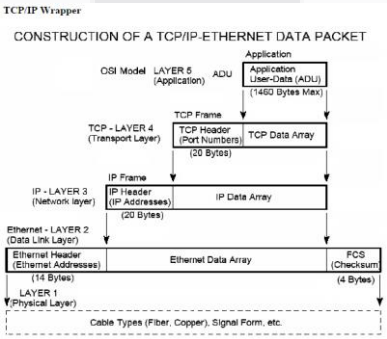
Gambar 2-5 Arsitektur OpenMTC



Gambar 2-6 Arsitektur ETSI untuk M2M [13]

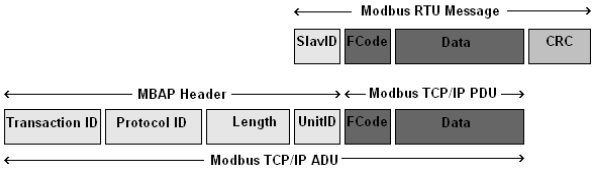
2.7 Modbus TCP

Modbus *TCP* adalah protokol Modbus yang sistem pengirimannya tidak menggunakan komunikasi serial tetapi menggunakan pembungkus *TCP IP* dan dikirimkan melalui jaringan. Sehingga dengan menggunakan protokol ini, paket yang dikirimkan dalam jaringan akan berubah.



Gambar 2-7 TCP IP Wrapper (source http://www.simplymodbus.ca/TCP.htm)

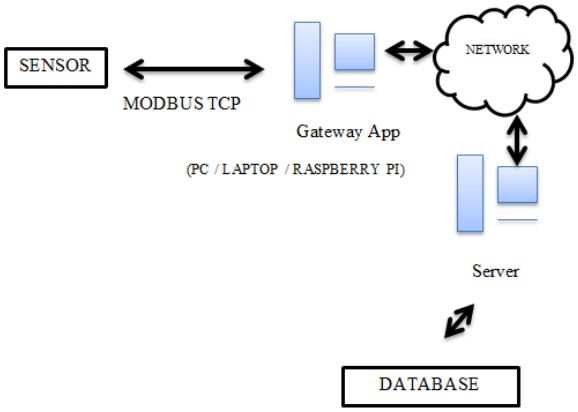
Dengan menghilangkan *slave ID* dan *CRC*, maka didapatkan *PDU (Packet Data Unit)*. Dengan menggunakan *PDU* tersebut kemudian ditambahkan 7 *byte header* yang disebut *MBAP (Modbus Application Header)* seperti pada gambar dibawah.



Gambar 2-8 Paket yang akan dikirimkan melalui internet (source http://www.simplymodbus.ca/TCP.htm)

Panjang dari *Transaction ID* yaitu 2 *bytes*, kemudian *Protocol ID* 2 *bytes* yang diset oleh klien, dan selalu bernilai = 00 00. Kemudian untuk *Length 2 bytes*. Untuk *Unit ID* 1 *byte* yang diset oleh klien kemudian di jawab oleh *server* untuk identifikasi.

3 Perancangan dan Implementasi



Gambar 3-1 Gambaran Umum Sistem Pengujian

Pada tugas akhir ini menerapkan protokol Modbus dengan menggunakan server OpenMTC untuk membangun sebuah komunikasi *Smart Building*. Protokol standar dari OpenMTC adalah HTTP. Kemudian sensor yang akan digunakan adalah sensor virtual.

Rancangan sistem yang akan dibangun yaitu terdiri dari 2 bagian, yaitu OpenMTC (*server*) dan jaringan sensor. Seperti pada gambar di atas, protokol Modbus merupakan penghubung antara sensor dengan *gateway app*. *Gateway app* ini merupakan pembungkus data dari sensor yang akan dikirimkan dengan pembungkus format data HTTP, yang merupakan protokol default yang digunakan oleh OpenMTC.

4 Pengujian dan Analisis

Pada bab ini akan dilanjutkan dengan penjelasan tentang aktifitas yang telah dilakukan yaitu dengan mengujicobakan VNODE tersebut untuk mengirimkan data ke *server* OpenMTC. Jumlah data sensor yang telah diuji cobakan adalah sebagai berikut :

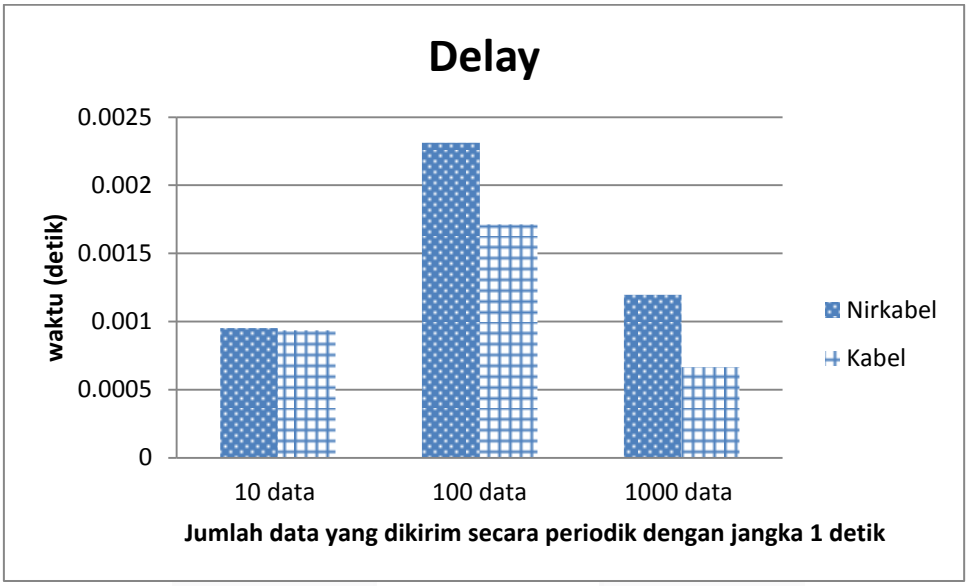
Tabel4-1 Skenario Uji

No	Media	Jumlah Sensor	Skenario
1	Jaringan Kabel	10	1
2	Jaringan Kabel	100	2
3	Jaringan Kabel	1000	3

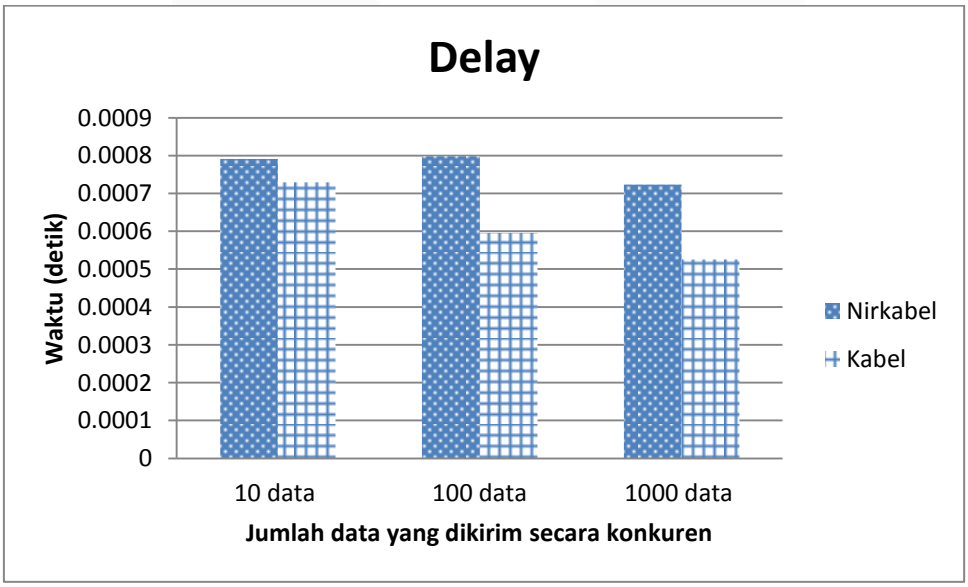
4	Jaringan Nirkabel	10	4
5	Jaringan Nirkabel	100	5
6	Jaringan Nirkabel	1000	6

4.1 Pengujian

- ***Round Trip Time Delay***
  Dalam grafik 4.1 *Delay*, data *round trip time delay* yang didapatkan sebagai berikut.



Grafik 4-1 Delay pengiriman data secara periodik



Grafik 4-2 Delay Delay pengiriman ketika data dikirim secara konkuren

- ***Protocol Overhead***

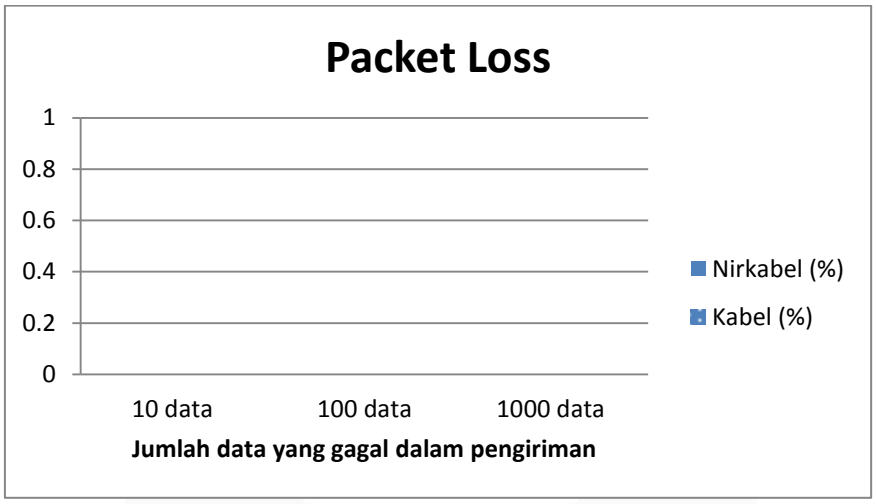
  Protocol Overhead yang tercatat di dalam pengujian yaitu sebagai berikut :

Tabel 4-2 Protocol Overhead data Modbus TCP

	Ukuran data (bytes)	Total Header (bytes)	Total Packet (bytes)	Protocol Overhead (%)
1 data	83	371	454	77.62803235

- ***Packet Loss***

  Packet Loss yang tercatat selama pengujian yaitu seperti pada grafik di bawah ini.



Grafik 4-2 Packet Loss

5 Kesimpulan dan Saran

5.1 Kesimpulan

Kesimpulan yang dapat diambil dalam percobaan ini sangat terpengaruhi dengan lingkungan pengujian, sehingga didapatkan kesimpulan :

1. Protokol Modbus TCP dengan *gateway app*,dan menggunakan *server* OpenMTC dapat mendukung untuk pembuatan *smart building*, dengan menggunakan paramater pengujian *delay*, *packet loss*, dan *protocol overhead.* Kode fungsi (*function code*) yang digunakan dalam penelitian tugas akhir ini yaitu *Read Holding Register* (03H).
2. Parameter *round trip time delay* dalam pengujian didapatkan bahwa ketika jumlah data sensor yang dikirimkan sebanyak 1000 sensor lebih baik dibandingkan dengan data jumlah sensor sebanyak 10 sensor dengan melihat rentang *delay* yang masih bisa ditoleransi yaitu kurang dari 1290 *milisecond* menurut EUROCOM Institut dalam publikasi LOLA Project.[16]
3. Parameter *protocol overhead* dalam pengujian didapatkan sebesar 77 %. Nilai ini memang masih cukup tinggi, karena menggunakan protokol TCP, yang perlu menggunakan header tambahan.
4. Parameter *packet loss* dalam pengujian didapatkan nilai 0 %., batas yang masih bisa ditoleransi yaitu tidak boleh lebih dari 0,1 % menurut standar ITU-T Y.1541.[17]
5. Protokol Modbus mempunyai keunggulan jika mengirimkan data dari *gateway app* dengan jumlah data sensor lebih dari 1000 data untuk setiap *gateway app* nya.

5.2 Saran

Saran yang dapat diberikan untuk penelitian lebih lanjut yaitu :

1. Skenario pengujian yang dapat dilakukan dengan menggunakan lebih dari satu *GSCL* dan *gateway app*.
2. Penggunaan *router* dalam infrastruktur jaringan, sehingga akan nampak bagaimana pengaruhnya terhadap parameter *round trip time delay, packet loss,* dan *protocol overhead*.
3. Perbandingan dengan menggnunakan protokol yang lain seperti Bacnet, dan yang lain.

.

Daftar Pustaka

[1] “Machine-to-Machine (M2M) Communication Challenges Established (U)SIM Card Technology" - GD". Gi-de.com. Retrieved 2014-01-21.
[2] "Machine to Machine (M2M) Technology in Demand Responsive Commercial Buildings" (PDF). Retrieved 2014-01-21.
[3] Wikipedia, http://en.wikipedia.org/wiki/Machine_to_machine diakses pada 28 September 2014.
[4] Accenture. energy-smart-buildings-whitepaper-1
[5] Wikipedia. http://en.wikipedia.org/wiki/Smart_city diakses pada 8 Oktober 2014
[6] ITS, Zen Samsono Hadi ,Muhammad. *Performance and Monitoring Network.*
[7] OpenMTC, http://open-mtc.org. Diakses tanggal 21 Oktober 2014
[8] MODBUS, http://www.modbus.org/faq.php. Diakses pada tanggal 21 Oktober 2014
[9] NodeJS. http://nodejs.org/. Diakses pada 21 Oktober 2014.
[10] Modbus. Modbus Protocol Reference Guide Rev J.
[11] Tiyono,Agus,dkk. 2007. *Sistem Telekontrol dengan Fungsi Dasar Modbus Menggunakan Mikrokontroller AT89S51 dan Komunikasi Serial RS485*. Universitas Diponegoro Semarang.
[12] WhitePaper-OpenMTC. 2012. OpenMTC Platform. Berlin
[13] OpenMTC . 2013. *The OpenMTC framework…interworking*. Berlin.
[14] Simply Modbus. http://www.simplymodbus.ca/TCP.htm. diakses pada 29 Januari 2015
[15] Zul Fahmi ,Fitra,2015. *ANALISIS STABILITAS SERVER OPENMTC UNTUK KOMUNIKASI MESIN KE MESIN (M2M) MENGGUNAKAN STRESS TEST*.Pembimbing Dr. Maman Abdurohman, ST., MT. Bandung.
[16] Knopp, Raymond. *Latency Requirement in M2M Application Scenarios.* Eurocom. Eropa.
[17] ITU-T Y.1541. *SERIES Y: GLOBAL INFORMATIONINFRASTRUCTURE, INTERNET PROTOCOL ASPECTSAND NEXT-GENERATION NETWORKS Internet protocol aspects – Quality of service and network performance*.2011.