

**APLIKASI SISTEM MONITORING JARINGAN
BERBASIS WEBSITE****NETWORK MONITORING SYSTEM APPLICATION
BASED ON WEBSITE**

Risang Suryadi Saputra, Hafidudin, Dadan Nur Ramadan

^{1,2,3}Diploma 3 Teknik Telekomunikasi, Fakultas Ilmu Terapan, Universitas Telkom
risang.pro@gmail.com, hafid@telkomuniversity.ac.id, dadannr@telkomuniversity.ac.id**Abstrak**

Dalam melakukan pengamatan terhadap kualitas jaringan, masih banyak yang menggunakan cara *manual*, seperti melakukan *ping* dari satu *host* ke *host* yang lain. Penelitian ini bertujuan membuat aplikasi monitoring berbasis *website* untuk mengamati kondisi *traffic*, *delay* dan *jitter* pada jaringan, aplikasi dibuat menggunakan bahasa pemrograman PHP, *python*, *javaScript*, dan MySQL. Kemampuan website ini diantaranya, dapat melihat *traffic* pada jaringan, mengetahui kondisi perangkat yang terhubung ke jaringan, informasi *traffic* jaringan akan ditampilkan pada google Map.

Berdasarkan hasil pengujian aplikasi monitoring, yang mengacu pada standar QoS, diperoleh rata-rata *delay* 1,97581 *millisecond* dan *jitter* 1,36925 *millisecond*, menandakan kualitas *traffic* data sangat bagus berdasarkan standar G.1010 untuk *telnet application*. Persentase error pengujian aplikasi monitoring terhadap pengujian dengan aplikasi Wireshark sebesar 16,82% untuk *delay* dan untuk *jitter* sebesar 23,46%.

Kata kunci: Sistem monitoring, Qos, Traffic.**Abstract**

To observe a network quality, there are still many who use manual methods, using ping from one host to another. This research aims create a website-based monitoring application to observe traffic delay and jitter on the network, this application made using PHP, python, JavaScript, and MySQL programming languages. The ability of this website can monitor a traffic and discover condition of the devices that connected to network computer, and the network traffic information displayed on Google Map.

The test results of monitoring application, that refers to the QoS standard. Show the average delay is 1.97581 millisecond and 1.36925 millisecond for jitter, indicating the quality of data traffic is very good, according to G.1010 standard for telnet application. The percentage error for testing monitoring application against the Wireshark testing, is 16.82% for delay and 23.46% for jitter.

Keywords: Monitoring Sistem, Qos, Traffic.**1. PENDAHULUAN**

Jaringan komputer dapat digunakan sebagai media informasi, semakin luas jaringannya menyebabkan sistem yang digunakan menjadi semakin kompleks, dan dibutuhkan penanganan dan pemeliharaan yang baik, agar jaringan komputer tersebut dapat berfungsi dengan optimal [1].

Saat ini, masih banyak proses pemeliharaan dan monitoring jaringan, yang menggunakan metode sederhana, yaitu dengan cara *ping* ke *Internet Protocol Address* (IP) setiap *client* atau *host* yang terhubung di pada jaringan, yang menyebabkan proses pemantauan menjadi kurang efisien, dikarenakan untuk mengetahui informasi *traffic* dan gangguan pada jaringan [2-4], harus dipantau secara langsung.

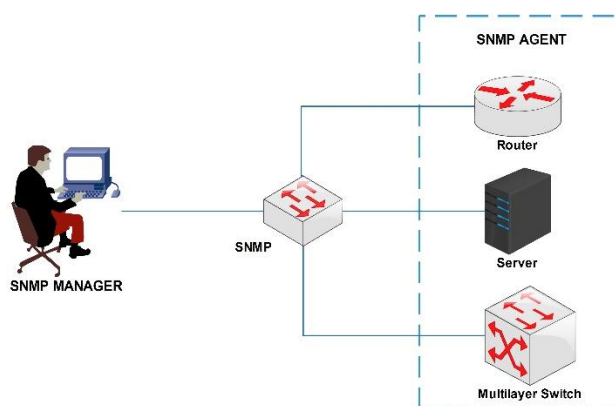
Beberapa penelitian sebelumnya mengenai monitoring jaringan telah dilakukan, diantaranya menggunakan Zigbee [5-9], tetapi area monitoring menjadi terbatas dikarenakan cakupan Zigbee yang pendek. Penelitian berikutnya menggunakan mikrotik, akan tetapi informasi masih ditampilkan dalam *Graphic user Interfaces* (GUI), sehingga proses pemantauan hanya dapat dilakukan di suatu lokasi.

Pada penelitian ini, dirancang aplikasi berupa *website* yang berfungsi untuk melakukan sistem informasi [10-11], untuk melakukan pengamatan pada *traffic* dan gangguan jaringan. Dengan menempatkan mikrotik di beberapa titik (*node*) jaringan, mikrotik tersebut akan memberikan informasi tentang *traffic* dan gangguan pada jaringan, informasi tersebut kemudian diolah di *web-server*, lalu ditampilkan pada halaman *website* untuk mempermudah proses pemantauan *traffic* dan gangguan pada jaringan.

2. DASAR TEORI

2.1 *Simple Network Management Protocol* (SNMP)

SNMP adalah *Internet Protocol* (IP) yang digunakan untuk mengumpulkan dan mengatur informasi tentang perangkat yang dikelola pada jaringan IP [12-13]. Jaringan yang dikelola SNMP terdiri dari tiga komponen utama, yaitu; *Manager* SNMP, *Agent* SNMP dan *Management Information Base* (MIB), konfigurasi sederhana dari komponen SNMP dapat terlihat pada Gambar 1 berikut.



Gambar 1. Konfigurasi SNMP

2.2 *Hyper Text Markup Language* (HTML)

HTML adalah sebuah bahasa pemrograman yang digunakan untuk membuat halaman *website* [14-15], dengan menggunakan kode dalam format ASCII disimpan dalam sebuah file. HTML dapat berfungsi sebagai dasar dan desain tampilan halaman *website*, mempublikasi dokumen atau data-data secara online sehingga dapat diakses menggunakan *browser* internet secara online, dengan ketentuan *delay* tidak lebih dari 10 detik [16].

2.3 MySQL

MySQL adalah salah satu perangkat lunak, yang memiliki fungsi sebagai *database management system* (DBMS) [17-19]. MySQL digunakan untuk mengatur data-data suatu *website*. MySQL dapat digunakan secara bersamaan oleh beberapa pengguna, serta dapat menangani *query* sederhana dengan cepat. Mendukung perintah dalam bentuk *query*, serta dapat menampilkan data dengan menggunakan *Application Programming Interface* (API) [20].

2.4 Quality of Service (QoS)

QoS merupakan metode pengukuran kualitas jaringan dalam menyediakan layanan yang baik, pengukuran QoS dapat berupa kualitatif atau kuantitatif [16], kualitas QoS berdasarkan G.1010 dapat dilihat seperti pada tabel 1, dimana rentang nilai QoS menentukan kualitas suatu jaringan.

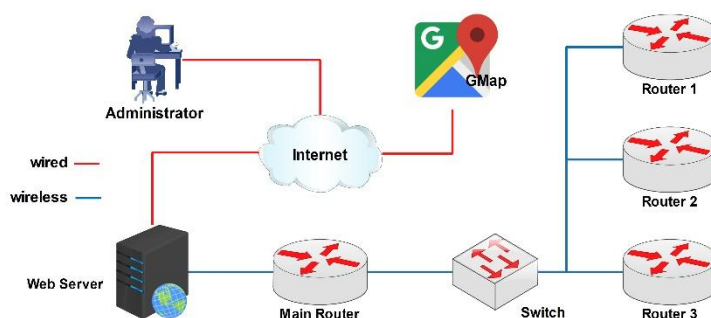
Table 1. Indeks parameter QoS berdasarkan G.1010 untuk Aplikasi[16]

Medium	Application	Degree of symmetry	Typical amount of data	Key performance parameters and target values		
				One-way delay (Note)	Delay variation	Information loss
Data	Web-browsing – HTML	Primarily one-way	~10 KB	Preferred < 2 s /page Acceptable < 4 s /page	N.A.	Zero
Data	Telnet	Two-way (asymmetric)	< 1 KB	< 200 ms	N.A.	Zero
Data	Low priority transactions	Primarily one-way	< 10 KB	< 30 s	N.A.	Zero
Data	Usenet	Primarily one-way	Can be 1 MB or more	Can be several minutes	N.A.	Zero

3. PERANCANGAN DAN PENGUJIAN

3.1 Perancangan

Untuk membuat *website* monitoring, digunakan bahasa pemrograman *Hypertext Preprocessor* (PHP) dan Python yang berfungsi sebagai backend. Bahasa pemrograman python berfungsi untuk pengambilan data *traffic* dari Mikrotik dan menyimpannya ke dalam *database* MySQL, sedangkan bahasa pemrograman PHP digunakan untuk pengambilan data dari *database* dan menampilkannya pada halaman *website*, pada Gambar 2 diperlihatkan blok diagram sistem aplikasi monitoring.



Gambar 2. Blok diagram sistem

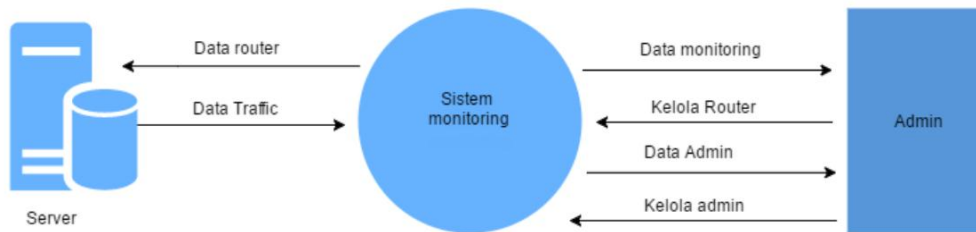
Pembuatan sistem monitoring yang memanfaatkan Google map (GMap) untuk mengambil titik koordinat lokasi penempatan setiap mikrotik *router*, informasi dari mikrotik *router* akan diambil oleh *python* dan disimpan ke dalam *database*, kemudian data tersebut akan di tampilkan pada halaman *website*, sehingga *user* dapat mengakses *website* dan dapat memantau kondisi jaringan dari jarak jauh.

Database MySQL digunakan untuk menyimpan semua data yang dibutuhkan, didalamnya terdapat beberapa tabel untuk memudahkan dalam menyimpan dan mengolah data. Pada Gambar 3 adalah desain dari *database* yang telah dibuat.

monitoring router	monitoring admin	monitoring traffic
id_router : int(200)	id_admin : int(255)	id_traffic : int(255)
lokasi : varchar(200)	nama : varchar(255)	ip : text
lat : varchar(200)	email : varchar(255)	jitter : varchar(255)
lng : varchar(200)	username : varchar(255)	delay : varchar(255)
cabang : varchar(200)	password : varchar(255)	tanggal : datetime
ip : varchar(200)	gambar : varchar(255)	
keterangan : text	level : int(255)	

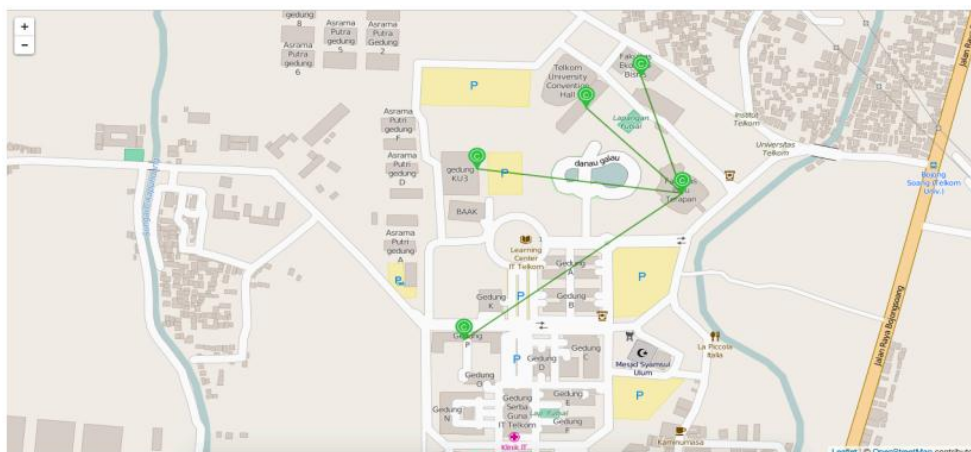
Gambar 3. Relasi pada Database MySQL

Pada Gambar 4 adalah *data-flow* bagaimana sistem dapat komunikasi dengan server, proses pertukaran informasi dari *router* dan *server* untuk mengirimkan data *traffic* jaringan, sehingga sistem dapat menampung data *router* dan data *traffic*, kemudia *user* atau *admin* dapat mengelola data tersebut.



Gambar 4. Data Flow Diagram

Pada Gambar 5, terlihat ada 1 *node router* berwarna merah, yang menandakan bahwa *node* tersebut mengalami gangguan jaringan, sehingga dengan aplikasi *website* ini, *user* jaringan dapat dengan cepat mengetahui kondisi jaringannya, baik dalam keadaan normal atau pada saat mengalami gangguan, serta dapat dengan cepat mngetahui dimanakah posisi gangguan itu terjadi.



Gambar 5. Tampilan website monitoring jaringan

3.1 Pengujian

Pada pengujian koneksi *router*, data *router* yang sudah disimpan pada *database* akan ditampilkan pada aplikasi gmap, sesuai dengan titik koordinat masing-masing *router*. Pada gmap akan menampilkan *pin point router*, dimana *pin* tersebut terdapat dua warna, yaitu hijau dan

merah, seperti pada gambar 6. Ketika pin berwarna hijau, menandakan bahwa *router* sedang terkoneksi dengan *server*, sedangkan apabila pin berwarna merah, menandakan router tidak terkoneksi ke server atau tidak aktif.



Gambar 6. Terdapat 2 pin warna sebagai indikator jaringan

Proses monitoring dilakukan menggunakan *script* python akan pengambilan informasi mengenai nilai *traffic delay* dan *jitter* dari setiap mikrotik *router* yang terpasang secara LAN atau WAN, dengan cara melakukan *ping* atau dengan mengirimkan paket ICMP dengan besar paket ICMP 64 *bytes*, proses tersebut akan berulang setiap 5 menit untuk mendapatkan data secara *real-time*, tanpa mengganggu *traffic* jaringan karena menggunakan protokol SNMP, seperti diperlihatkan pada Gambar 7.

```

send packet 1
Done
send packet 2
Done
send packet 3
Done
send packet 4
Done

                delay packet 1                = 0.00466990470886 s
                delay packet 2                = 0.00156998634338 s
                delay packet 3                = 0.00153708457947 s
                delay packet 4                = 0.00170803070068 s

rata rata delay= 2.3712515831 ms
Statistic from 192.168.1.2
Location GKU
Inter-arrival Delay Average = 0.00056653 ms
Jitter = 1.10125542 ms
    
```

Gambar 7. Proses pengiriman paket ICMP sebesar 64 bytes.

Setelah dilakukan proses monitoring, maka data *jitter* dan *delay* yang diperoleh akan di database, kemudian akan diolah dan ditampilkan dalam bentuk grafik pada halaman website seperti pada Gambar 8.

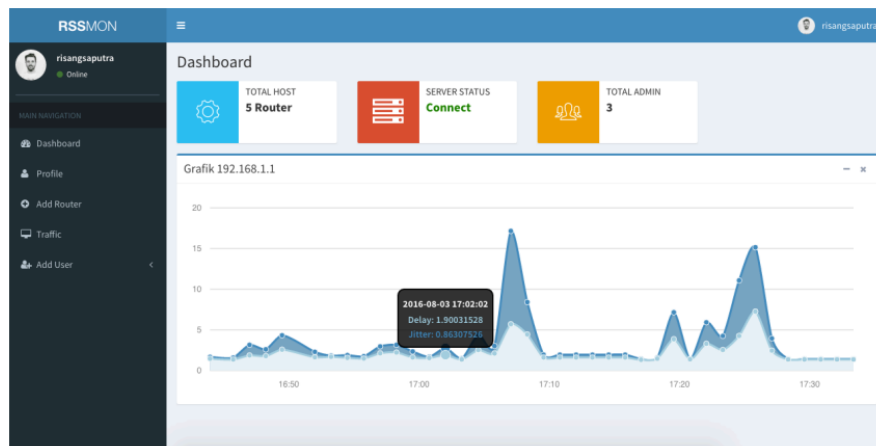
Data Traffic

Show 10 entries Search:

No	Ip	Jitter/ms	Delay/ms	Tanggal	Action
1	192.168.2.1	0.02495448	0.39052963	2016-08-03 17:33:22	🗄 🔗
2	192.168.1.1	0.05602837	1.36119127	2016-08-03 17:33:22	🗄 🔗
3	192.168.2.1	0.02495448	0.39052963	2016-08-03 17:32:06	🗄 🔗
4	192.168.1.1	0.05602837	1.36119127	2016-08-03 17:32:06	🗄 🔗
5	192.168.2.1	0.02495448	0.39052963	2016-08-03 17:30:51	🗄 🔗
6	192.168.1.1	0.05602837	1.36119127	2016-08-03 17:30:51	🗄 🔗
7	192.168.2.1	0.02495448	0.39052963	2016-08-03 17:29:36	🗄 🔗
8	192.168.1.1	0.05602837	1.36119127	2016-08-03 17:29:36	🗄 🔗

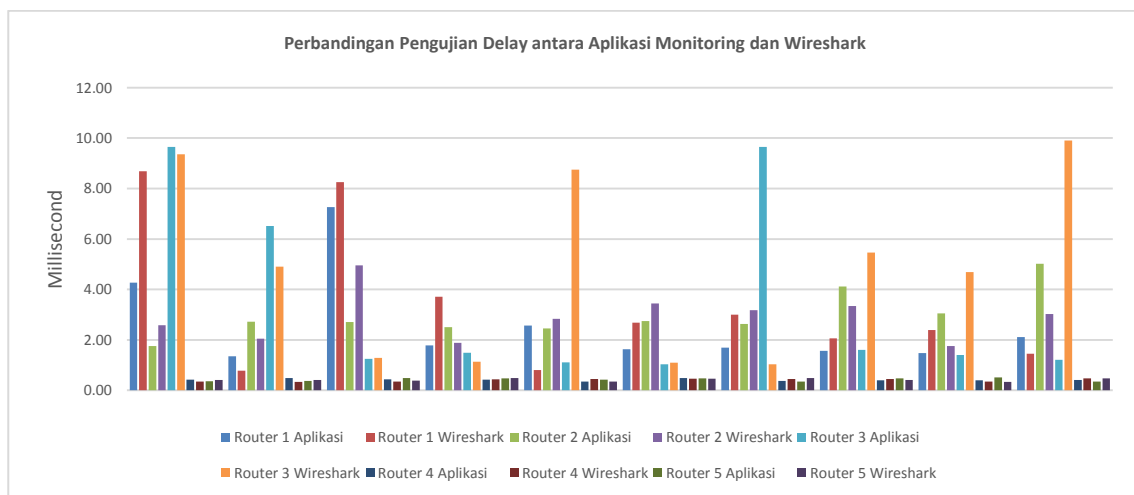
Gambar 8. Data monitoring traffic seluruh node

Untuk informasi yang lebih lengkap dari node jaringan, ditampilkan dalam grafik *history*, seperti pada gambar 9, yang menampilkan semua data dari *note* yang ingin dipantau.



Gambar 9. Tampilan grafik informasi pada halaman website monitoring

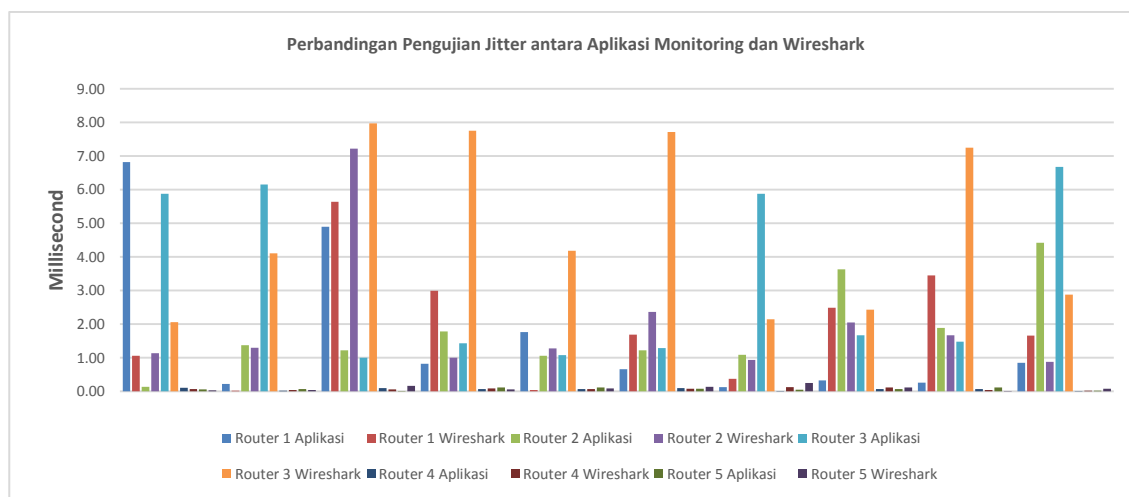
Pengujian berikutnya adalah pengujian nilai *delay* dan *jitter* dari setiap *node*, dengan menggunakan aplikasi Wireshark sebagai pembandingan. Pada gambar 10 dapat dilihat grafik perbandingan nilai *delay* antara aplikasi monitoring dan Wireshark, dimana diagram *router 1* aplikasi adalah nilai yang peroleh oleh aplikasi monitoring untuk *router 1*, dan diagram *router 1* Wireshark adalah nilai yang peroleh berdasarkan aplikasi Wireshark pada Router 1.



Gambar 10. Perbandingan Pengujian *delay* antara Aplikasi Monitoring dan Wireshark

Berdasarkan hasil pengujian yang ditampilkan pada gambar 10, nilai *delay* rata-rata yang diperoleh adalah 1,97581 *millisecond* untuk perhitungan dengan menggunakan aplikasi monitoring, dan 2,37548 *millisecond* untuk nilai rata-rata *delay* yang diperoleh aplikasi Wireshark.

Pada gambar 11 adalah grafik perbandingan nilai *jitter* antara aplikasi monitoring dan wireshark. Berdasarkan hasil pengujian yang ditampilkan pada gambar 10, nilai *jitter* rata-rata yang diperoleh adalah 1,36925 *millisecond* untuk perhitungan dengan menggunakan aplikasi monitoring, dan 1,78884 *millisecond* untuk nilai rata-rata *jitter* yang diperoleh aplikasi Wireshark.



Gambar 11. Perbandingan Pengujian *Jitter* antara Aplikasi Monitoring dan Wireshark

Berdasarkan tabel kategori *jitter* 0 - 75 *millisecond* dan *delay* < 150 *millisecond* dengan mengacu pada standar ITU-T, maka hasil pengujian rata-rata yang diperoleh keseluruhan sebanyak 5 buah node adalah 1,97581 *millisecond* untuk *delay*, dan *jitter* sebesar 1,36925 *millisecond*, yang artinya *traffic* yang diperoleh dari aplikasi yang dibuat sangat bagus.

4. KESIMPULAN

Dari perancangan dan pengujian aplikasi monitoring yang telah dilakukan, terhadap fitur-fitur pada sistem monitoring berbasis *website*, semua fitur dapat berjalan sebagaimana mestinya. Aplikasi sistem monitoring berbasis *website* ini dapat membantu *administrator* dalam mengawasi jaringannya, berdasarkan hasil pengujian *traffic* yang didapatkan setelah di analisis dan dilakukan percobaan dengan mendapatkan *delay* 1,97581 *millisecond* dan *jitter* 1,36925 *millisecond* artinya kualitas *traffic* data sangat bagus berdasarkan standar ITU-T G.1010 untuk *telnet application*. Persentase *error* pengujian aplikasi *monitoring* terhadap pengujian dengan aplikasi Wireshark sebesar 16,82% untuk *delay* dan untuk *jitter* sebesar 23,46%.

DAFTAR PUSTAKA

- [1] IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks - Amendment 23: Application Virtual Local Area Network (VLAN) Type, Length, Value (TLV). 2015. INSPEC Accession Number: 14998209
- [2] A, Puri. 2002, Optimizing traffic flow in fixed wireless networks, Wireless Communications and Networking Conference Record, vol.2, Pages: 904 - 907

- [3] L, Chi-Wai. K, Tsz-Mei. 1998. A multi-rate channelized wireless LAN system with fixed channel assignment, IEEE International Symposium on Circuits and Systems (Cat. No.98CH36187). Vol 6. pp: 533 - 536
- [4] B, Anup. S, Uma, CK Jha. 2014. Traffic analysis between two neighbor ad hoc network under channel interference. Recent Advances in Engineering and Computational Sciences (RAECS). pp: 1 – 5
- [5] Y, Wu. J, Han-hong. W, Jin. Z, Chao-liang. 2010. Research on the hybrid network technology of industrial ethernet and Zigbee for monitoring the ship power system. 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010).
- [6] P, A, Manoj. A, Pravin. 2011 Pervasive ambient intelligence system: A ZigBee based sensor networks for ambient monitoring. Conference on Signal Processing, Communication, Computing and Networking Technologies. pp: 619 - 622
- [7] Z, Chunlong. Z, Min. S, Yongsheng. W, Weilian. 2012. Smart home design based on ZigBee wireless sensor network. Conference on Communications and Networking.
- [8] Z, Lihong. S, Lei. L, Weina. A Temperature Monitoring System of Power Cable Joints Based on the Combining of CAN Wired Transmission and ZigBee Wireless Network. Conference on Information Engineering and Computer Science
- [9] N, Kaushik. N, Kiran. C, Shrikrishna. R, Ramana. S, David. P, Haribabu. 2012. Web based monitoring and control of WSN using WINGZ (Wireless IP network gateway for Zigbee). Sixth International Conference on Sensing Technology (ICST). pp: 666 – 67
- [10] R, Dadan Nur. P, Agus Ganda. M, Galuh. P, Dyah. 2016. Rancang Bangun Dan Implementasi Alat Ukur Dan Sistem Informasi Pada Listrik Satu Fasa. Jurnal Elektro dan Telekomunikasi Terapan (JETT). Vol 2. No 2. pp:166-176
- [11] W, Aditya Alif. R, Tengku Ahmad. P, Hasanah. 2016. Perancangan Dan Implementasi Sarana Informasi Sman 3 Madiun Berbasis Framework Codeigniter. Jurnal Elektro dan Telekomunikasi Terapan (JETT). Vol 3 No 1. pp: 269-277
- [12] P, Kasula Chaithanya. N, Iluri. J, Ranjith Singh. 2017. Implementation of SNMP-JSON translator and integrating SNMP agents with JSON based network management system. Communication Systems and Network Technologies (CSNT). pp: 67 – 73
- [13] Z, Wenxian. W, Yue. 2019. Design and Implementation of Server Monitoring System Based on SNMP. International Joint Conference on Artificial Intelligence. Pp: 680 - 682
- [14] R, Burget. 2004. Hierarchies in HTML documents: linking text to concepts. International Workshop on Database and Expert Systems Applications, 2004. pp: 186 - 190
- [15] A, Kandel. M, M, El-Khouly. M, AbdEl-Hakeem. 2004. Tutoring system for teaching HTML through WWW. International Conference on Information and Communication Technologies: From Theory to Applications, 2004. pp: 107 - 108
- [16] ITU-T, “TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS Quality of service and performance.” [Online]. Available: <https://www.itu.int/rec/T-REC-G.1010-200111-I/en>.
- [17] Z, Ivan. 2006. MYSQL Security Best Practices. IET Conference on Crime and Security. pp: 183 - 198
- [18] G, Steven A. 2007. Creating Database-Driven Websites with PHP/MySQL. Wiley-IEEE Press eBook Chapters.
- [19] P, Yu. W, Hu Hong. N, Zhou. 2014. Design and implementation of a MySQL database backup and recovery system. World Congress on Intelligent Control and Automation. Pp: 5410-5415

- [20] K, Dong Inn. J, M, Squyres. A, Lumsdaine. 2006. The Introduction of the OSCAR Database API (ODA). International Symposium on High-Performance Computing in an Advanced Collaborative Environment (HPCS'06). pp: 39