



SISTEM *REAL TIME MONITORING* KECEPATAN ANGIN, ARAH ANGIN, DAN SUHU BERBASIS *WEB* MENGGUNAKAN *PROXY RESERVE* PADA PROTOKOL *TRANSMISSION CONTROL PROTOCOL (TCP)*

SKRIPSI

Oleh

Romadhoni Ibnu Fadhil

NIM 111910201107

**PROGRAM STUDI STRATA SATU (S1)
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2016**



SISTEM *REAL TIME MONITORING* KECEPATAN ANGIN, ARAH ANGIN, DAN SUHU BERBASIS *WEB* MENGGUNAKAN *PROXY RESERVE* PADA PROTOKOL *TRANSMISSION CONTROL PROTOCOL (TCP)*

SKRIPSI

diajukan guna melengkapi skripsi dan memenuhi salah satu syarat untuk menyelesaikan Program Studi S1 Teknik Elektro dan mencapai gelar Sarjana Teknik

Oleh

Romadhoni Ibnu Fadhil

NIM 111910201107

**PROGRAM STUDI STRATA SATU (S1)
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2016**

PERSEMBAHAN

Dengan menyebut nama Allah SWT, skripsi ini saya persembahkan untuk:

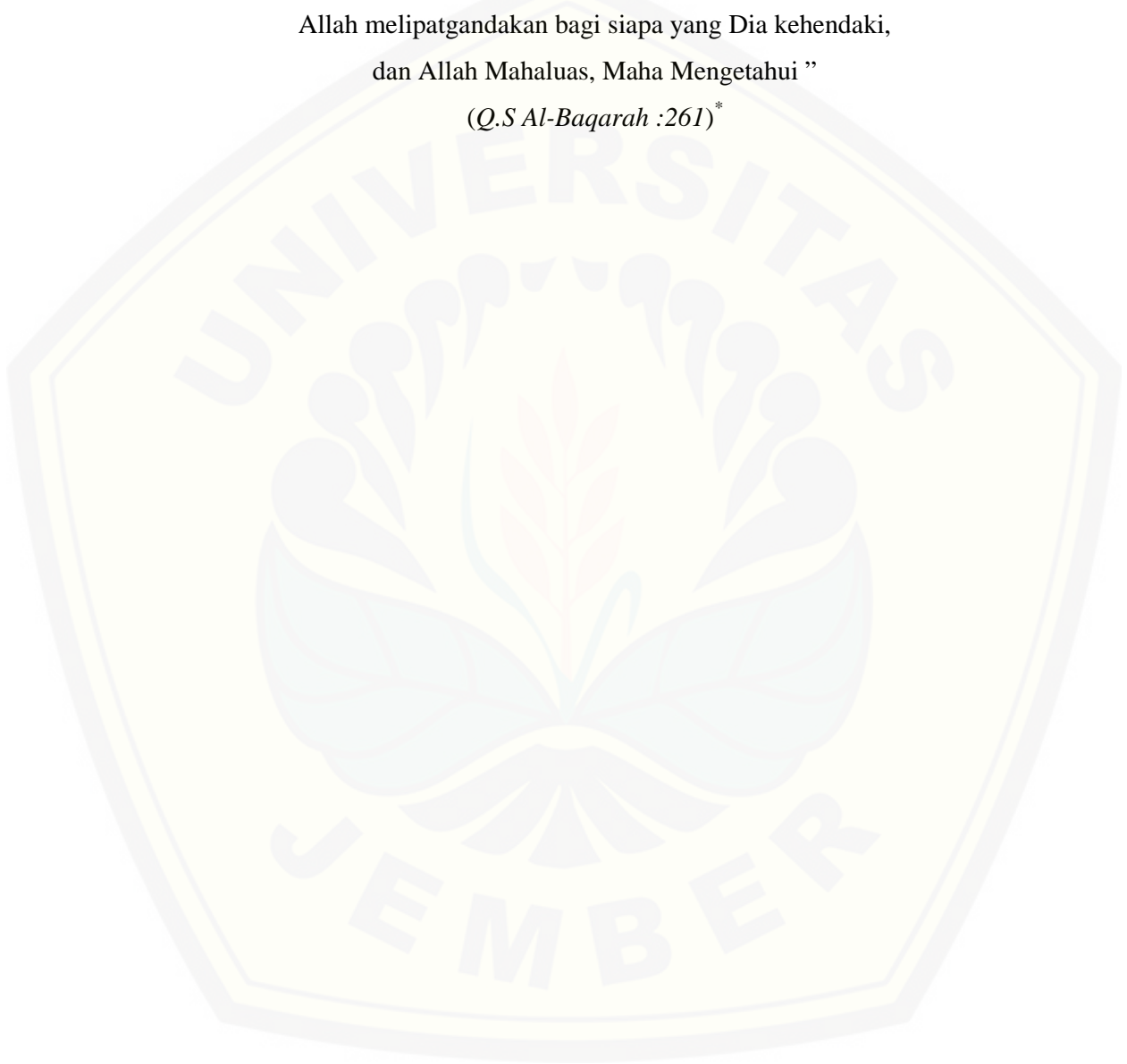
1. Keluarga besar tercinta, yaitu: Ibuku Hj Siti Asiyah, Ayahku Alm. H. Djalal Suyuthi, dan kakak-kakakku Anis Rudiana, SE., Faida Halifah, Spd., Adib Muttaqin, SE., serta Fitroni Hariadi, SE. atas doa, dukungan, motivasi, dan kasih sayang yang diberikan dalam iringan langkahku untuk menuntut ilmu;
2. Guru-guruku sejak Sekolah Dasar, Sekolah Menengah Pertama, Sekolah Menengah Atas sampai dengan Perguruan Tinggi;
3. Almamater Fakultas Teknik Universitas Jember.

MOTTO

“Perumpamaan orang yang menginfakkan hartanya di jalan Allah seperti sebutir biji yang menumbuhkan tujuh tangkai, pada setiap tangkai ada seratus biji.

Allah melipatgandakan bagi siapa yang Dia kehendaki,
dan Allah Mahaluas, Maha Mengetahui ”

(*Q.S Al-Baqarah :261*)*



*Departemen Agama Republik Indonesia. 2006. *Al Qur'an dan Terjemahan*. Bandung: Penerbit Sinar Baru Algensindo.

PERNYATAAN

Saya yang bertanda tangan di bawah:

Nama : Romadhoni Ibnu Fadhil

Nim : 111910201107

Menyatakan dengan sesungguhnya bahwa skripsi yang berjudul **SISTEM REAL TIME MONITORING KECEPATAN ANGIN, ARAH ANGIN, DAN SUHU BERBASIS WEB MENGGUNAKAN PROXY RESERVE PADA PROTOKOL TRANSMISSION CONTROL PROTOCOL (TCP)** adalah benar-benar karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, dan belum pernah diajukan pada institusi manapun, serta bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapatkan sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, April 2016

Yang menyatakan,

Romadhoni Ibnu Fadhil
NIM 111910201107

SKRIPSI

SISTEM *REAL TIME MONITORING* KECEPATAN ANGIN, ARAH ANGIN, DAN SUHU BERBASIS *WEB* MENGGUNAKAN *PROXY RESERVE* PADA PROTOKOL *TRANSMISSION CONTROL PROTOCOL (TCP)*

Oleh

**Romadhoni Ibnu Fadhil
NIM 111910201107**

Pembimbing

Dosen Pembimbing 1 : Catur Suko Sarwono, S.T.

Dosen Pembimbing 2 : Dr. Triwahju Hardianto, S.T., M.T.

PENGESAHAN

Skripsi berjudul "Sistem *Real Time Monitoring* Kecepatan Angin, Arah Angin, dan Suhu Berbasis *Web* Menggunakan *Proxy Reserve* pada Protokol *Transmission Control Protocol* (TCP)" telah diuji dan disahkan oleh Fakultas Teknik Universitas Jember pada:

Hari, tanggal : Selasa, 05 April 2016

Tempat : Fakultas Teknik Universitas Jember.

Menyetujui,

Pembimbing Utama,

Pembimbing Anggota,

Catur Suko Sarwono S.T.
NIP. 19680119 199702 1 001

Dr. Triwahju Hardianto S.T., M.T.
NIP. 19700826 199702 1 001

Penguji I,

Penguji II,

Widjonarko, S.T., M.T.
NIP. 19710908 199903 1 001

Satryo Budi Utomo, S.T., M.T.
NIP. 19850126 200801 1 002

Mengesahkan,
Dekan Fakultas Teknik
Universitas Jember,

Dr.Ir. Entin Hidayah M.U.M
NIP 19661215 199503 2 001

Sistem *Real Time Monitoring* Kecepatan Angin, Arah Angin, dan Suhu
Berbasis *Web* Menggunakan *Proxy Reserve* pada Protokol
Transmission Control Protocol (TCP)

Romadhoni Ibnu Fadhil

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember

ABSTRAK

Keperluan sistem *monitoring* saat ini semakin mempermudah pengguna dalam mendapatkan kebutuhan informasi dari tempat yang sulit dijangkau. Sistem *monitoring* dapat diterapkan untuk *memonitoring* kondisi angin di suatu tempat untuk mendapatkan kebutuhan informasi dalam perencanaan pembangunan Pembangkit Listrik Tenaga Angin. Pada penelitian ini dilakukan pembuatan sistem *real time monitoring* kecepatan angin, arah angin, dan suhu berbasis *web* menggunakan *proxy reserve* pada *protocol transmission control protocol* (TCP). Bentuk *monitoring* akan ditampilkan pada halaman *web* dalam bentuk angka, grafik, dan gambar. *Client* dapat melakukan *monitoring* melalui halaman *web* yang dibuat dengan perangkat yang tersambung dengan jaringan internet. Dari hasil pengujian didapatkan analisis data dengan rata-rata waktu respon *server* terhadap pengiriman data pada *proxy reserve* sebesar 0.395 s. Pada pengukuran QoS data protokol TCP pada *client* didapatkan nilai paling baik pada kategori waktu malam dengan nilai rata-rata RTT 0.079 s, rata-rata *time out* 0.418 s, dan rata-rata *throughput* 607.873 KBps. Sedangkan presentase kesalahan pengiriman terbesar terjadi pada kategori waktu siang dengan nilai RTO sebesar 4.37 %. Klasifikasi QoS pada parameter RTT menghasilkan kategori sangat baik pada waktu pagi, siang, dan malam dengan nilai rata-rata sebesar 104 ms. Klasifikasi QoS pada parameter RTO menghasilkan kategori sangat baik pada waktu pagi dengan nilai RTO sebesar 1.25%, sedangkan menghasilkan kategori baik pada waktu siang dengan RTO sebesar 4.37% dan waktu malam dengan RTO sebesar 3.75%.

Kata kunci: sistem *monitoring*, protokol TCP, QoS, *proxy reserve*, RTT, *time out*, *throughput*, RTO.

RINGKASAN

Sistem *Real Time Monitoring* Kecepatan Angin, Arah Angin, dan Suhu Berbasis Web Menggunakan *Proxy Reserve* pada Protokol *Transmission Control Protocol* (TCP); Romadhoni Ibnu Fadhil, 111910201107; 2016: 79 halaman; Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Angin merupakan sumber energi terbarukan yang potensial untuk menghasilkan energi listrik. Menurut informasi yang dilansir oleh ESDMMAG edisi 2 (2012:12) Indonesia merupakan negara dengan kondisi wilayah yang berpotensi untuk pembangunan sumber energi listrik berbasis angin. Perencanaan pembangunan sumber energi angin membutuhkan informasi tentang potensi angin yang berada pada wilayah yang direncanakan. Data informasi yang dibutuhkan adalah kecepatan angin, arah angin, dan suhu. Untuk mendapatkan informasi tersebut diharuskan mengambil data angin langsung ketempat objek pengambilan data angin. Seperti penelitian sebelumnya yang dilakukan oleh Gunawan (2015), data angin dapat diketahui setelah pergi ketempat objek penelitian yaitu di Pantai Pancer Kecamatan Puger Kabupaten Jember. Oleh karena itu, diperlukan sistem *monitoring* jarak jauh secara *real time* untuk mempermudah mendapatkan informasi data angin tersebut. Sistem *real time monitoring* yang dibangun yaitu dengan memanfaatkan komunikasi *client/server* berbasis web pada jaringan internet publik *proxy reserve*. Tujuan penelitian untuk membangun sistem *monitoring* kecepatan angin, arah angin, dan suhu secara jarak jauh berbasis *web* menggunakan *proxy reserve* dan mengetahui hasil kualitas komunikasi data sistem *monitoring* pada protokol *Transmission Control Protocol* (TCP). Penelitian ini dapat dimanfaatkan untuk pengukuran jarak jauh potensi angin sebagai kajian data untuk pembangunan pembangkit listrik energi terbarukan.

Penelitian dilaksanakan di Pantai Pancer Kecamatan Puger Kabupaten Jember sebagai *server* dan di kawasan Universitas Jember sebagai *client*. Alat yang digunakan adalah anemometer berbasis kecepatan angin, arah angin, dan suhu. Sensor yang digunakan untuk anemometer yaitu sensor *hall effect* A3144, sensor kompas CMPS03, dan sensor suhu DS1621. *Server* untuk menyimpan dan

publikasi data pada penelitian ini menggunakan Raspberry Pi 2 tipe B. *Proxy reserve* sebagai jembatan publikasi ke internet publik menggunakan aplikasi *ngrok*.

Analisis yang dilakukan setelah membuat sistem *real time monitoring* adalah melakukan pengujian terhadap keandalan sistem *server* dan pengujian terhadap pengukuran kualitas komunikasi data pada *client*. Pengujian tersebut akan menghasilkan data dalam penelitian untuk mengetahui bahwa sistem *monitoring* yang dibuat bekerja dengan baik dan optimal. Penagambilan data dan penyimpanan pada *database server* memberikan hasil data dan waktu yang sama dengan yang dihasilkan oleh alat anemometer. Hasil yang didapat dari pengujian keandalan sistem *server* dalam mengirimkan data terbukti bekerja dengan sangat baik. *Server* mampu merespon dan mengirimkan data ke *proxy reserve* dengan waktu respon rata-rata sebesar 0.395 detik. Hasil tersebut didapatkan dengan mengambil 10 pencuplikan data kirim *server* satu waktu secara acak. Hasil dari pengujian kualitas komunikasi data dari sisi *client* memberikan hasil yang sangat baik. Pengujian dilakukan dengan perhitungan beberapa parameter QOS pada protokol TCP yaitu *Round Trip Time (RTT)*, *time out*, *Throughput*, dan *Request Time Out (RTO)*. Skenario pengujian adalah dengan dilakukan 10 pencuplikan pengiriman data setiap satu jam secara acak pada tiga kategori waktu yaitu pagi (01.00-09.00), siang (09.00-17.00), dan malam (17.00-01.00). Hasil rata-rata yang didapat pada kategori pagi RTT 0.134s, *time out* 0.723s, *throughput* 499.768 KBps, dan RTO 1.25%. kategori siang menghasikan rata-rata RTT 0.088s, *time out* 0.546, *throughput* 578.884 KBps, dan RTO 4.37%. Kategori malam menghasilkan rata-rata RTT 0.079s, *time out* 0.418, *throughput* 607.873 KBps, dan RTO 3.75%.

Kesimpulan yang diperoleh dari penelitian ini yaitu keandalan *server* memberikan respon pengiriman data yang baik pada jaringan internet publik menggunakan *proxy reserve*. Kualitas komunikasi data pada sisi *client* dipengaruhi oleh kondisi jaringan *Internet Service Provider (ISP)* yang digunakan. Hal tersebut dibuktikan dengan hasil pengukuran kualitas pengiriman data yang berubah-ubah setiap waktu.

PRAKATA

Puji syukur penulis panjatkan ke hadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi ini yang berjudul sistem *real time monitoring kecepatan* angin, arah angin, dan suhu berbasis *web* menggunakan *proxy reserve* pada protokol *transmission control protocol* (TCP). Skripsi ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan strata satu (S1) pada Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak, oleh karena itu penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Ibu Dr. Ir. Entin Hidayah M.U.M selaku Dekan Fakultas Teknik Universitas Jember.
2. Bapak Dr. Triwahju Hardianto S.T., M.T selaku Ketua jurusan Teknik Elektro.
3. Bapak Catur Suko Sarwono S.T. selaku Dosen Pembimbing Skripsi I yang telah meluangkan waktu, tenaga, pikiran, dan perhatian dalam penulisan skripsi ini.
4. Bapak Dr. Triwahju Hardianto S.T., M.T. selaku Dosen Pembimbing Skripsi II dan telah membantu memberikan kepercayaan untuk mengerjakan proyek penelitian yang digunakan pada skripsi ini.
5. Bapak Widjonarko, S.T., M.T. selaku Dosen Penguji Skripsi I.
6. Bapak Satryo Budi Utomo, S.T., M.T. selaku Dosen Penguji Skripsi II.
7. Bapak Suprihadi Prasetyono, ST., MT. selaku Dosen Pembimbing Akademik yang telah membimbing penulis selama menjadi mahasiswa.
8. Seluruh Bapak dan Ibu dosen yang telah memberikan bekal ilmu selama menyelesaikan studi di Jurusan Teknik Elektro.
9. Semua pihak yang telah memberikan bantuan dalam penyelesaian skripsi ini.

Penulis juga menerima segala kritik dan saran yang bersifat membangun demi kesempurnaan penulisan selanjutnya. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat. Amin

Jember, April 2016

Penulis

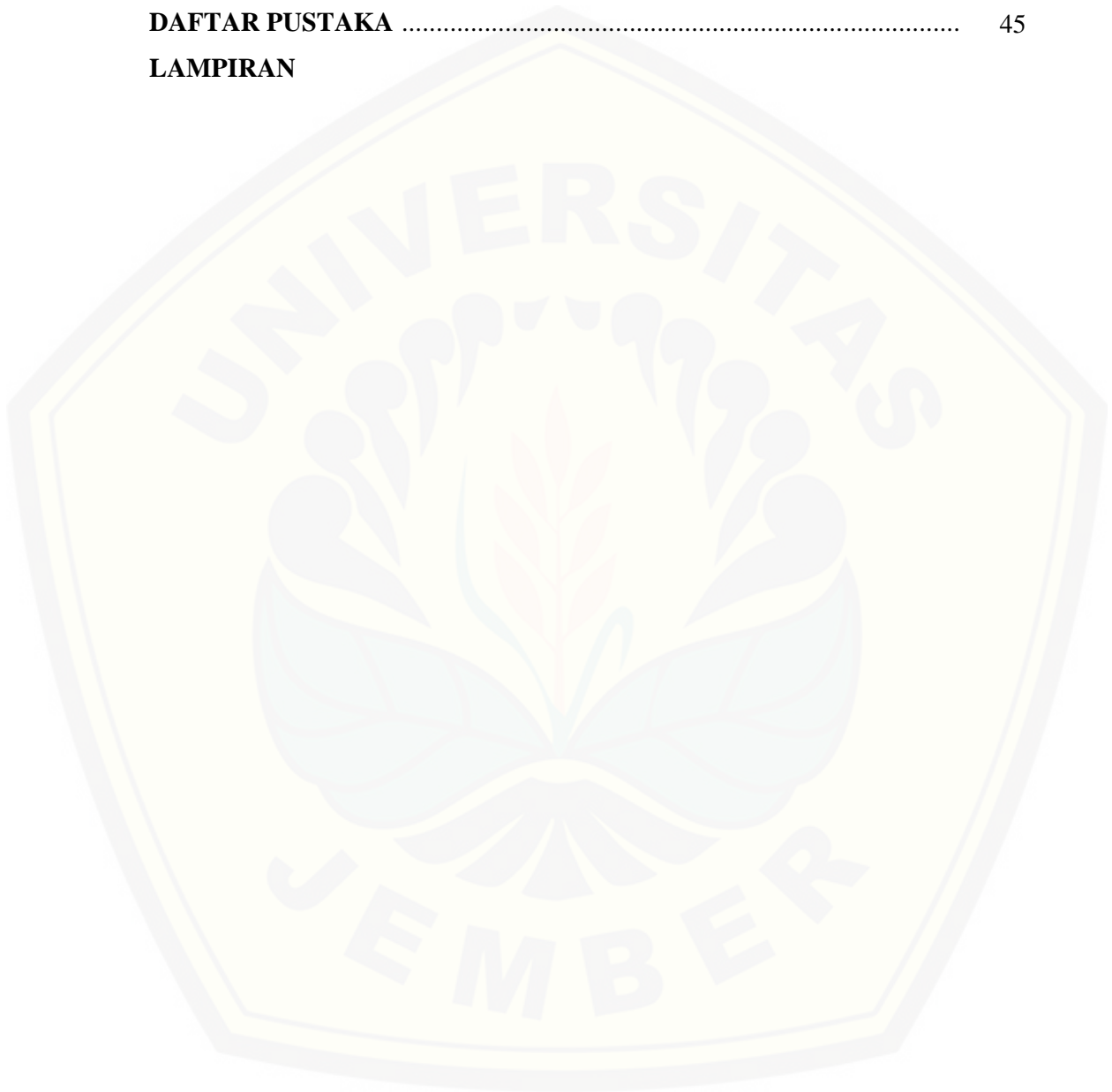


DAFTAR ISI

	Halaman
HALAMAN JUDUL	ii
HALAMAN PERSEMBAHAN	iii
HALAMAN MOTTO	iv
PERNYATAAN	v
PEMBIMBINGAN	vi
HALAMAN PENGESAHAN	vii
ABSTRAK	viii
RINGKASAN.....	ix
PRAKATA	xi
DAFTAR ISI	xiii
DAFTAR TABEL	xvi
DAFTAR GAMBAR.....	xvii
DAFTAR LAMPIRAN.....	xviii
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
BAB 2. TINJAUAN PUSTAKA	5
2.1 Sistem Telemetry	5
2.2 Internet	5
2.3.1 Pengertian Internet.....	5
2.3.2 <i>World Wide Web (WWW)</i>	5
2.3 GSM (<i>Global System for Mobile Communication</i>)	6
2.4 Protokol Jaringan.....	7
2.5 Protokol TCP (<i>Transmission Control Protocol</i>).....	8
2.6 Komunikasi Serial.....	9

2.7 Model <i>Client/Server</i>	11
2.8 <i>Localhost</i>	11
2.9 <i>Reverse Proxy</i>	12
2.10 Sistem Operasi	12
2.11.1 Linux	12
2.11 Bahasa Pemrograman Python	13
2.12 Raspberry Pi 2 B	14
2.13 Arduino Uno R3	15
2.14 <i>Data Logger</i>	16
2.15 <i>SD Card</i>	17
2.16 Sensor <i>Hall Effect A3144</i>	18
2.17 Sensor CMPS03	19
2.18 Sensor Suhu DS1621	20
2.19 <i>Quality of Service (QoS)</i>	20
BAB 3. METODE PENELITIAN	22
3.1 Jenis Penelitian	22
3.2 Tempat dan Waktu Penelitian	22
3.3 Prosedur Penelitian	23
3.4 Alat dan Bahan Penelitian	24
3.4.1 Anemometer Berbasis Kecepatan Angin, Arah Angin, dan Suhu	24
3.4.2 Raspberry Pi 2	26
3.4.2 Portable Router TP-Link MR3020	27
3.4.2 <i>Software Wireshark</i>	28
3.4.2 Laptop	30
3.4.2 <i>Localhost Tunnel Ngrok</i>	30
3.5 Diagram Perancangan Alat	31
3.6 Diagram Alir Sistem Monitoring	32
BAB 4. HASIL DAN PEMBAHASAN	35
4.1 Pengujian Respon <i>Server Raspberry Pi</i>	35
4.2 Pengukuran QOS (<i>Quality Of Service</i>)	38

BAB 5. PENUTUP	44
5.1 Kesimpulan	44
5.2 Saran	44
DAFTAR PUSTAKA	45
LAMPIRAN	



DAFTAR TABEL

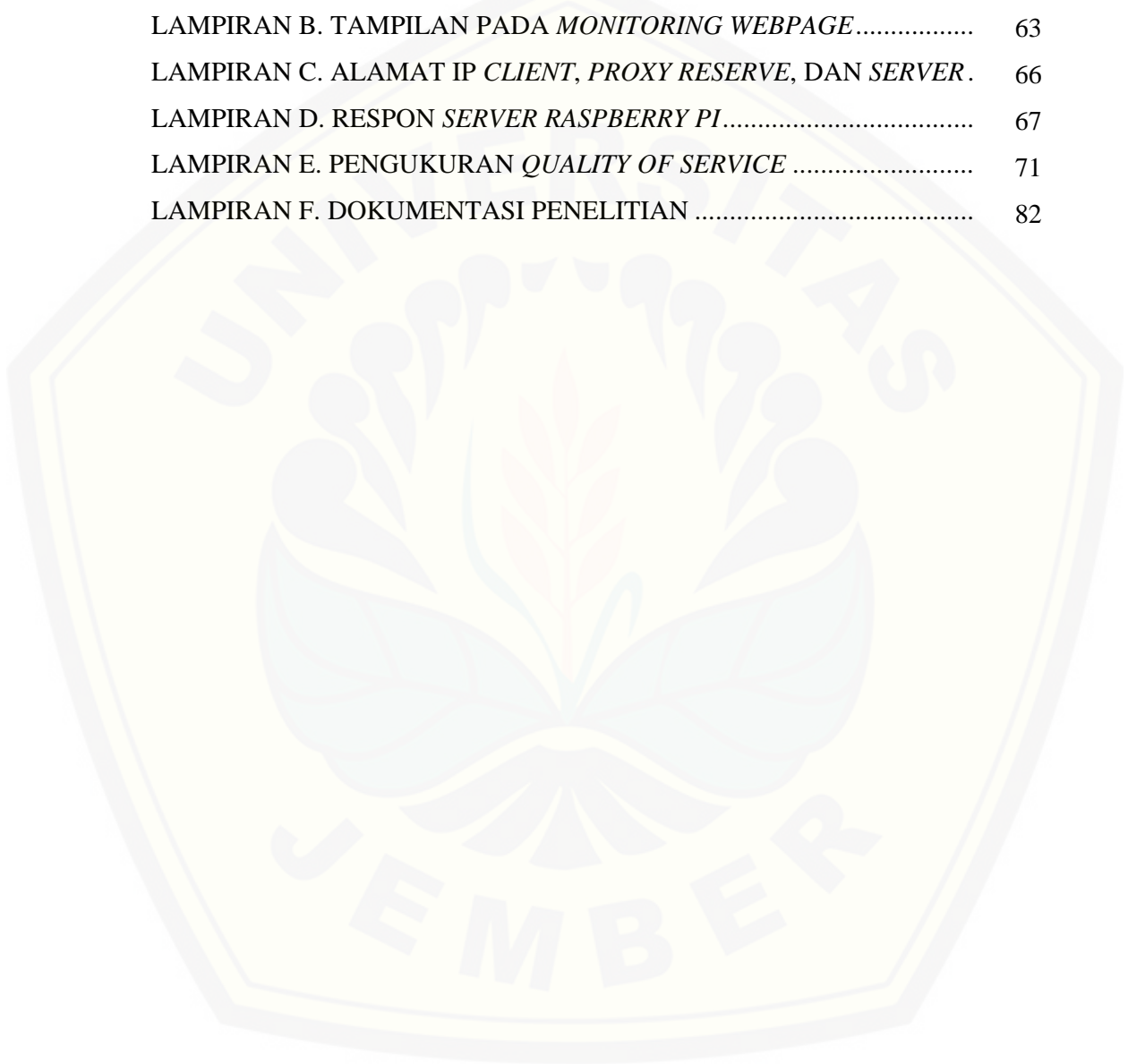
	Halaman
2.1 Spesifikasi <i>Raspberry Pi 2 B</i>	15
2.2 Klasifikasi QoS parameter <i>latency end-to-end</i>	21
2.3 Klasifikasi QoS parameter <i>packet loss</i>	21
3.1 Jadwal pelaksanaan penelitian.....	22
4.1 Waktu respon <i>server</i> pada pengiriman data kecepatan angin, suhu, arah angin pada raspberry pi	37
4.2 Hasil pengukuran rata-rata RTT, rata-rata <i>time out</i> , dan <i>throughput</i>	39
4.3 Hasil pengukuran rata-rata RTT, rata-rata <i>time out</i> , dan <i>throughput</i> setiap kategori waktu	41

DAFTAR GAMBAR

	Halaman
2.1 Proses <i>three-way handshaking</i>	9
2.2 <i>Raspberry Pi 2 B</i>	14
2.3 <i>Arduino Uno R3</i>	16
2.4 <i>Micro SD Card</i>	17
2.5 Pin Sensor <i>Hall Effect A3144</i>	18
2.6 Sensor Kompas <i>CMPS03</i>	19
2.7 Deskripsi PIN <i>DS1621</i>	20
3.1 Diagram blok sistem alat anemometer berbasis kecepatan angin, arah angin, dan suhu.....	24
3.2 Anemometer berbasis kecepatan angin, arah angin, dan suhu	25
3.3 Sistem arduino dan <i>server</i>	26
3.4 Tampilan desktop raspberry pi <i>remote session VNC</i>	26
3.5 Tampilan <i>phpMyAdmin</i>	27
3.6 Tampilan <i>management router</i>	28
3.7 Tampilan <i>Software Wireshark</i>	28
3.9 <i>Port forwarding http</i> pada ngrok	30
3.10 Diagram blok <i>transmitter</i> sistem <i>monitoring</i>	31
3.11 Diagram blok <i>receiver</i>	32
3.12 Diagram Alir Sistem <i>Monitoring</i>	34
4.1 Grafik perbandingan <i>RTT</i> dan <i>time out</i> bersarkan kategori waktu	42

DAFTAR LAMPIRAN

	Halaman
LAMPIRAN A. <i>LISTING</i> PROGRAM <i>RASPBERRY PI</i>	47
LAMPIRAN B. TAMPILAN PADA <i>MONITORING</i> <i>WEBPAGE</i>	63
LAMPIRAN C. ALAMAT IP <i>CLIENT</i> , <i>PROXY</i> <i>RESERVE</i> , DAN <i>SERVER</i> .	66
LAMPIRAN D. RESPON <i>SERVER</i> <i>RASPBERRY PI</i>	67
LAMPIRAN E. PENGUKURAN <i>QUALITY OF SERVICE</i>	71
LAMPIRAN F. DOKUMENTASI PENELITIAN	82



BAB 1. PENDAHULUAN

1.1 Latar Belakang

Aktivitas kehidupan penduduk di Indonesia tidak bisa lepas dari energi listrik. Produksi dan kebutuhan terhadap energi listrik semakin hari semakin meningkat. Skenario dasar produksi listrik nasional diperkirakan mencapai 976 TWh atau tumbuh sebesar 7.1% pertahun selama periode 2012-2035 dengan dominasi energi fosil sebagai bahan bakar pembangkit (BPPT, 2014:79). Eksploitasi terhadap energi fosil yang tidak dapat diperbarui dapat menyebabkan berkurangnya ketersediaan sumber energi pembangkit listrik. Energi terbarukan menjadi solusi yang tepat untuk mengatasi pemenuhan kebutuhan energi listrik, salah satunya adalah dengan memanfaatkan energi angin.

Menurut informasi yang dilansir oleh ESDMMAG edisi 2 (2012:12) indonesia merupakan negara dengan kondisi wilayah yang berpotensi untuk pembangunan sumber energi listrik berbasis angin. Perencanaan pembangunan sebuah PLTB (Pembangkit Listrik Tenaga Bayu) memerlukan analisa potensi energi angin dan estimasi energi listrik yang dihasilkan pada wilayah yang ditunjuk. Informasi parameter yang dibutuhkan untuk perencanaan pembangunan PLTB harus akurat dan cepat. Namun dalam mendapatkan informasi parameter tidak semua kondisi memungkinkan dilakukan dengan datang secara langsung ke lokasi dikarenakan faktor geografis dan jarak, hal tersebut dapat menghambat dalam memperoleh informasi yang diinginkan. Pemanfaatan teknologi telekomunikasi sangat dibutuhkan untuk proses pengiriman informasi ukur jarak jauh (telemetry) dengan cepat tanpa harus pergi langsung ke tempat pengambilan data informasi.

Sistem telemetry digunakan untuk pengukuran jarak jauh dari daerah yang sulit dijangkau oleh manusia seperti wilayah kelautan atau pegunungan. Sistem telemetry tanpa kabel (*wireless*) menambah efisiensi kerja dengan hanya meletakkan alat ukur di lokasi pengukuran dan memonitoring hasil ukur dari tempat lain. Penelitian sebelumnya (Husumardiana, 2015) tentang perancangan sistem telemetry pada alat pengukur kecepatan angin berbasis X-Bee Pro

menggunakan *Kalman Filter* mendapatkan hasil yang perlu dikembangkan lagi. Hasil yang didapat adalah jarak pengiriman data pada saat kondisi *loss space* sekitar 100 meter, dan hasil data setelah proses *kalman filter* adalah kurang maksimal. Desain alat ukur pada penelitian tersebut mengalami perkembangan dengan membuat desain anemometer berbasis kecepatan angin, arah angin, dan suhu secara *real time* oleh gunawan (2015). Penelitian tersebut menghasilkan data kecepatan, arah angin dan suhu yang disimpan pada data logger dan ditampilkan langsung pada LCD dengan selang waktu pengiriman data yaitu 2 detik.

Masyarakat modern saat ini tidak bisa lepas dari fasilitas komunikasi data internet untuk memperoleh informasi yang diinginkan. Kebutuhan informasi tersebut salah satunya disajikan dalam bentuk *webpage* dalam sebuah *website*. Pembangunan sebuah *website* memerlukan *web server* yang handal untuk merespon *request* dari *user*. Salah satu kebutuhan untuk meningkat performa *web server* adalah menggunakan *proxy reserve*. *Proxy reserve* mampu mengurangi presentase *page load time* pada sebuah *website* (Sirajjudin dkk, 2012).

Komunikasi data internet memiliki model jaringan agar suatu data dapat dikirim dan diterima. Salah satu protokol dalam model tersebut adalah *Transmission Control Protocol* (TCP). TCP memberikan jaminan pengiriman data sampai diterimanya data tersebut.

Berdasarkan pada latar belakang dan merujuk pada penelitian sebelumnya penulis akan membangun sistem *monitoring* kecepatan angin, arah angin, dan suhu berbasis *web* secara *realtime*. Pengguna dapat melakukan monitoring data dimana saja melalui *website* selama mendapat akses internet. Data diperoleh dari alat anemometer berbasis kecepatan angin, arah angin, dan suhu yang dikirim menggunakan komunikasi serial USB ke *server*. Data akan dipublikasikan menggunakan *proxy reserve* dengan memanfaatkan jaringan internet *mobile wireless Global System for Mobile communication* (GSM). Server dibangun menggunakan *mini computer raspberry pi 2*. *Proxy reserve* yang diterapkan menggunakan aplikasi *ngrok* sebagai perantara *user* dengan *webserver* (Permana, 2014:3). Protokol pengiriman data internet yang dianalisa adalah pada protokol TCP.

1.2 Rumusan Masalah

Berdasarkan latar belakang, dapat dirumuskan suatu permasalahan diantaranya :

1. Bagaimana membangun sistem *monitoring* jarak jauh kecepatan angin, arah angin, dan suhu berbasis *web* menggunakan *proxy reserve*?
2. Bagaimana kualitas komunikasi data sistem *monitoring* pada protokol TCP ?

1.3 Batasan Masalah

Pembahasan masalah pada penelitian ini dibatasi oleh beberapa hal, antara lain :

1. Lokasi yang digunakan untuk objek monitoring adalah Pantai Puger Kabupaten Jember.
2. Raspberry Pi digunakan sebagai *local server*.
3. Tidak membahas sisi elektronika secara keseluruhan tapi lebih kearah komunikasi.
4. Protokol pengiriman data yang digunakan adalah protocol *Transmission Control Protocol* (TCP).
5. Bahasa pemrograman yang digunakan adalah Python, PHP, dan SQL.
6. Kualitas komunikasi data yang diukur adalah *Round Trip Time* (RTT), *time out*, *throughput*, dan *Request Time Out* (RTO).
7. Analisa kualitas komunikasi data menggunakan *software wireshark*.
8. Konfigurasi *proxy reserve* menggunakan *ngrok*.
9. Anemometer yang digunakan dilengkapi dengan pengukuran arah angin dan suhu yang terdiri dari Arduino Uno R3, data logger, LCD, dan tiga buah sensor yaitu sensor *hall effect* A3144, sensor kompas CMPS03, dan sensor suhu DS1621.
10. Pembuatan dan pengkalibrasian alat anemometer telah dilakukan pada penelitian sebelumnya (Gunawan, 2015).
11. Objek *monitoring* adalah kecepatan angin, arah angin, dan suhu pada ketinggian 43.2 meter.

1.4 Tujuan Penelitian

Tujuan dilakukannya penelitian ini antara lain :

1. Membangun sistem *monitoring* kecepatan angin, arah angin, dan suhu secara jarak jauh berbasis *web* menggunakan *proxy reserve*.
2. Mengetahui hasil kualitas komunikasi data sistem *monitoring* pada protokol *Transmission Control Protocol* (TCP).

1.5 Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini adalah sebagai berikut:

1. Menciptakan teknologi *monitoring* potensi angin jarak jauh berbasis *web* secara *real time*.
2. Menyajikan data laporan kecepatan angin, arah angin, dan suhu untuk dimanfaatkan sebagai pembangkit listrik *renewable energi*.
3. Memberikan manfaat bagi perkembangan ilmu pengetahuan dan teknologi khususnya dalam bidang telekomunikasi.

BAB 2. TINJAUAN PUSTAKA

2.1 Sistem Telemetry

Telemetry adalah penggunaan telekomunikasi untuk merekam dan mengirimkan sinyal pengukuran secara otomatis dari suatu alat ukur yang berada pada jarak jauh. Selanjutnya informasi hasil pengukuran tersebut dikirimkan dengan berbagai cara menuju user (Sayood, 2003). Sistem telemetry bertujuan untuk mengambil suatu data dari tempat yang lokasinya jauh dan mengirimkannya ke stasiun pusat untuk diolah. Penggunaan sistem telemetry banyak dijumpai dalam kehidupan sehari – hari pada pemantauan cuaca, *tracking* satelit, monitoring kendaraan , dan sebagainya.

2.2 Internet

2.3.1 Pengertian Internet

Menurut pakar telekomunikasi Onno. W. Purbo, "Internet dengan berbagai aplikasinya seperti *Web*, *VoIP*, *E-Mail* pada dasarnya merupakan media yang digunakan untuk mengefesiesikan proses komunikasi.

Sedangkan menurut tim penelitian dan pengembangan wahana komputer internet adalah suatu metode untuk menghubungkan berbagai komputer ke dalam satu jaringan global, melalui protokol yang disebut *Transmission Control Protocol / Internet Protocol (TCP / IP)*.

Berdasarkan kedua pendapat diatas, maka dapat disimpulkan bahwa internet adalah suatu jaringan komunikasi antara komputer yang besar, yang mencakup seluruh dunia dan berbasis pada sebuah protokol yang disebut TCP / IP (*Transmission Control Protocol / Internet Protocol*). Selain itu, internet dapat disebut sebagai sumber daya informasi yang dapat digunakan oleh seluruh dunia dalam mencari informasi (Awaludin, 2015).

2.3.2 *World Wide Web (WWW)*

World Wide Web merupakan jaringan dokumentasi yang sangat besar yang saling berhubungan antara satu dengan yang lainnya. Satu set protokol yang mendefinisikan bagaimana sistem bekerja dan mentransfer data, dan sebuah

software yang membuatnya bekerja dengan mulus. *Web* menggunakan tehnik *hypertext* dan multimedia yang membuat internet mudah digunakan dijelajahi dan didistribusikan.

Web merupakan sistem hypermedia yang berarea luas yang ditujukan untuk akses secara universal. Salah satu kuncinya adalah kemudahan tempat seseorang atau perusahaan dapat menjadi bagian dari *web* dan berkontribusi pada *web*.

Web merupakan sistem yang menyebabkan pertukaran data di internet menjadi mudah dan efisien. *Web* terdiri atas 2 komponen dasar:

- a. *Server web* : sebuah komputer dan *software* yang menyimpan dan mendistribusikan data ke komputer lainnya melalui internet.
- b. *Browser web* : *software* yang dijalankan pada komputer pemakai (*client*) yang meminta informasi dari *server web* yang menampilkannya sesuai dengan *file* data itu sendiri.

Menurut Hardjono (2006:2), *web* merupakan fasilitas hiperteks untuk menampilkan data berupa teks, gambar, suara, animasi, dan data multimedia lainnya. Ada 2 kategori dalam perograman *web*, yaitu: pemrograman *Server Web* dan *Client Web*. Pada pemrograman *Server Side*, perintah-perintah program (*script*) dijalankan di *server web*, kemudian hasil dikirimkan ke *browser* dalam bentuk HTML biasa. Adapun pada *Client Side*, perintah program dijalankan pada *browser web* sehingga ketika *client* meminta dokumen *script*, maka *script* dapat di-*download* dari *server* kemudian dijalankan pada *browser* yang bersangkutan.

2.3 GSM (Global System for Mobile Communication)

GSM berasal dari bahasa perancis yaitu Groupe Speciale Mobile, dan sekarang lebih dikenal dengan sebutan *Global Sistem for Mobile Communications*. GSM adalah generasi kedua dari *cellular mobile system* yang diinovasi di eropa oleh ETSI (*European telecommunications Standardization Institute*).

GSM modem bekerja tanpa kabel (*wireless*) dan bekerja dengan jaringan GSM. Cara kerja GSM hampir mirip seperti cara kerja dial-up modem. Perbedaan utamanya adalah dial-up modem mengirim dan menerima data melalui jalur

telepon, sedangkan GSM melalui gelombang radio. Untuk itu GSM memerlukan pembawa datanya untuk mengirim dan menerima data.

GSM tidak akan bisa berkomunikasi bila tidak menggunakan SIM-Card (*Subscriber Identity Module-Card*) karena kartu SIM tersebut yang akan memberikan *network profile* sebagai carrier, serta memiliki algoritma untuk *encrypt* data yang dikirim dan diterima. Untuk berkomunikasi dengan modem GSM bisa dilakukan melalui kabel USB untuk PC dan kabel serial untuk *device* lain dengan jenis komunikasinya adalah UART.

Sistem telemetri menggunakan jaringan GSM membuat pengguna dapat mengakses data informasi yang diberikan dengan jarak jangkauan yang luas. Pengguna menerima informasi langsung dari server dengan waktu pengiriman yang sangat efisien. Selain efisiensi waktu, pembangunan sistem telemetri menggunakan jaringan GSM juga relatif murah (Pambudi dkk, 2012).

2.4 Protokol Jaringan Komunikasi

Sebuah komunikasi yang berhasil dapat terjadi karena memiliki protokol komunikasi. Komunikasi data mirip dengan percakapan manusia. Manusia dan komputer melakukan komunikasi formal untuk tujuan tertentu. Keduanya, baik manusia atau komputer mengikuti aturan – aturan tertentu yang memungkinkan para pelaku untuk bertukar informasi dengan cara yang teratur dan bebas dari kesalahan. Protokol merupakan suatu himpunan aturan yang mengatur komunikasi data. Secara umum protokol pada jaringan mendefinisikan tiga hal utama, yaitu apa yang dikomunikasikan, kapan terjadinya komunikasi, dan bagaimana komunikasi antar terminal (Suntana, 2005).

Protokol – protokol dipatuhi untuk membentuk dan mengakhiri komunikasi dengan sedemikian rupa sehingga tidak ada satu pesan pun yang tertinggal dalam keadaan yang tidak diinginkan. Data yang dikirim pada prosesnya masih terdapat permasalahan yang harus diselesaikan. Permasalahannya adalah pengiriman data harus tepat pada tujuan. Masalah akan bertambah ketika komputer tujuan berada di luar wilayah lokal. Usaha dalam mengatasi masalah tersebut adalah dengan memecah paket yang dikirim menjadi lebih kecil. Protokol komunikasi dibuat

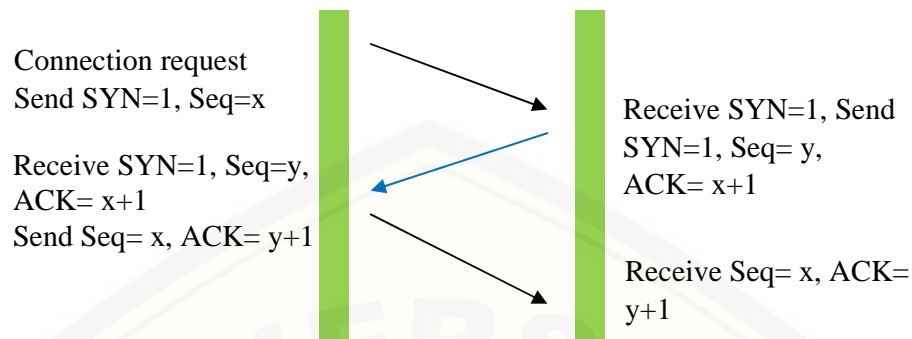
untuk menangani permasalahan komunikasi dengan menerapkan aturan – aturan komunikasi.

Protokol TCP/IP menjadi standart protokol yang digunakan pada jaringan internet. Protokol TCP/IP terdiri atas dua protokol yaitu protokol *transmission control protocol* (TCP) dan *Internet Protocol* (IP). Protokol TCP mengatur pengiriman data dari sumber ke tujuan yang benar. *Internet Protocol* berwenang untuk pengaturan indentifikasi data agar data dapat dikenal dan dapat dikirim dari sumber ke tujuan (Muarso dan Suryono, 2014).

2.5 Protokol TCP (*Transmission Control Protocol*)

Protocol TCP merupakan protokol yang bekerja pada layer *transport protocol* TCP/IP. Protokol pada layer *transport* membentuk *logical communication* antara aplikasi yang berjalan pada sebuah *host* menuju ke *host* lain. *logical communication* adalah proses berjalannya aplikasi pada sebuah terminal dengan membentuk koneksi sedemikian sehingga seolah-olah langsung terhubung dengan terminal lain, padahal pada kenyataanya kedua terminal terpisah oleh banyak router dan perangkat komunikasi lainnya.

Protokol TCP bertanggung jawab terhadap pengiriman data sampai diterimanya data oleh tujuan. Protokol TCP adalah protokol yang bersifat *connection oriented*. Protocol TCP mengharuskan komunikasi *three-way handshake* untuk melanjutkan proses pertukaran data antar *host*. Protokol TCP juga bersifat *reliable*. TCP mampu mendeteksi kerusakan data pada saat pengiriman dan melakukan pengiriman ulang (*retransmission*) terhadap kerusakan data tersebut. *Byte stream* pada protokol TCP mampu mengirimkan paket data secara berurutan (*sequence*).



Gambar 2.1 Proses *three-way handshaking*

(Sumber : Grace, 2014:352)

2.6 Komunikasi Serial

Perkembangan komunikasi di masa sekarang sangat cepat. Dimulai dari teknik komunikasi data secara paralel sampai pengembangan teknik komunikasi data serial yang dilakukan pengembangan sangat cepat. Pada awalnya yang kita kenal port paralel (DB25) sebagai piranti komunikasi komputer dengan printer dan berbagai alat lainnya. Akan tetapi dilihat dari perkembangan kedepannya, teknik ini mengalami kesulitan dan kendala, hal ini dipandang dari sudut ekonomisnya. Dengan jumlah jalur komunikasi yang banyak, menjadikan komunikasi paralel ini mulai ditinggalkan dan beralih ke teknik komunikasi serial.

Komunikasi data serial dimulai dari port serial (DB9) sampai saat ini dikenal dengan teknologi USB, SATA, dan Wi-Fi menjadikan komunikasi serial sebagai teknik komunikasi yang mengalami perkembangan yang sangat cepat. Pada prinsipnya komunikasi serial merupakan komunikasi dimana pengiriman data dilakukan per bit sehingga lebih lambat dibandingkan komunikasi paralel, atau dengan kata lain komunikasi serial merupakan salah satu metode komunikasi data di mana hanya satu bit data yang dikirimkan melalui seuntai kabel pada suatu waktu tertentu. Dipandang cukup ekonomis hanya membutuhkan dua jalur komunikasi yaitu transmit (Tx) dan receive (Rx), dan jika menggunakan piranti kabel maka cukup membutuhkan minimal 3 kabel yaitu transmit, receive, dan ground. Prinsip komunikasi serial adalah pengiriman data secara serial dengan

menggunakan karakter – karakter didalam ASCII. Karakter ini nantinya akan diubah menjadi sinyal digital oleh hardware transmitter (Tx), dan akan diterjemahkan lagi menjadi data karakter oleh hardware receiver (Rx). Komunikasi data serial mikrokontroler sangatlah sederhana, dikarenakan sudah memiliki instruksi – instruksi pemrograman yang standart.

Antarmuka kanal serial lebih kompleks atau sulit dibandingkan dengan antarmuka melalui kanal paralel, hal ini disebabkan karena:

- a. Dari segi perangkat keras: adanya proses konversi data paralel menjadi serial atau sebaliknya menggunakan piranti tambahan yang disebut UART (Universal Asynchronous Receiver/Transmitter).
- b. Dari segi perangkat lunak: lebih banyak register yang digunakan atau terlibat. Namun di sisi lain antarmuka kanal serial menawarkan berapa kelebihan dibandingkan secara paralel, antara lain:
 - 1) Kabel untuk komunikasi serial bisa lebih panjang dibandingkan dengan paralel: data - data dalam komunikasi serial dikirimkan untuk logika '1 sebagai tegangan -3 s/d -25 volt dan untuk logika '0 sebagai tegangan +3 s/d +25 volt, dengan demikian tegangan dalam komunikasi serial memiliki ayunan tegangan maksimum 50 volt, sedangkan pada komunikasi paralel hanya 5 volt. Hal ini menyebabkan gangguan pada kabel - kabel panjang lebih mudah diatasi dibandingkan pada paralel.
 - 2) Jumlah kabel serial lebih sedikit: kita dapat menghubungkan dua perangkat komputer yang berjauhan dengan hanya 3 kabel untuk konfigurasi null modem, yaitu TXD (saluran kirim), RXD (saluran terima) dan Ground, bayangkan jika digunakan teknik paralel akan terdapat 20 – 25 kabel. Namun pada masing - masing komputer dengan komunikasi serial harus dibayar biaya antarmuka serial yang agak lebih mahal.
 - 3) Banyaknya piranti saat ini (palmtop, organizer, handphone, dan lain - lain) menggunakan teknologi infra merah untuk komunikasi data, dalam hal ini pengiriman datanya dilakukan secara serial.
 - 4) Untuk teknologi embedded system, banyak mikrokontroler yang dilengkapi dengan komunikasi serial (baik seri RISC maupun CISC) atau Serial

Communication Interface (SCI): dengan adanya SCI yang terpadu pada IC mikrokontroler akan mengurangi jumlah pin keluaran, sehingga hanya dibutuhkan 2 pin utama TxD dan RxD (di luar acuan ground).

2.7 Model Client/Server

Server adalah sebuah komputer yang berisi program baik sistem operasi maupun program aplikasi yang menyediakan pelayanan kepada komputer atau program lain yang sama atau berbeda. Komputer server adalah komputer yang dikhususkan untuk menyimpan data yang akan digunakan bersama, atau sebagai basis data. Jenis server yang banyak digunakan adalah *Disk Server*, *File Server*, *Printer Server*, dan *Terminal Server*. *Client* adalah yang meminta layanan dari server. Sebuah Aplikasi komputer bisa memiliki bagian server dan bagian *client*, dimana dapat berjalan bersama – sama dalam satu sistem. server merupakan program yang dapat menerima permintaan (*request*), melakukan pelayanan yang diminta, kemudian mengembalika sebagai *reply*. Server dapat melayani *muti request* bersamaan.

2.8 Localhost

Localhost merupakan istilah dalam komputer jaringan yang berarti “komputer ini”. *Localhost* adalah nama standar yang diberika sebagai alamat loopback network interface. Localhost selalu menerjemahkan *loopback ip address* 127.0.0.1 dalam IPv4 atau IPv6. *Localhost* digunakan untuk mengantarkan web browser pada HTTP server yang terpasang di komputer lokal. Alamat <http://localhost> akan menampilkan website lokal pada komputer pribadi. Jadi kita membuat komputer kita menjadi *localserver*, kemudian mempostingkan web kita kedalamnya (*localhost*) untuk dijadikan tempat membangun website sementara yang kemudian akan dihosting ke internet publik. Dengan menjadika komputer kita sebagai *localserver*, kita dapat bekerja secara *offline* tanpa ada permasalahan biaya, waktu, dan ketidaknyamanan.

Semua komputer yang terhubung dengan internet itu harus menggunakan alamat IP secara unik, untuk megakses suatu server, maka harus memasukan

alamat IP nya. Alamat IP tersebut akan diterjemahkan dan dijadikan DNS (*Domain Name Sistem*) agar lebih mudah dalam pengaksesanya. Contoh DNS adalah www.unej.ac.id.

2.9 Reserve proxy

Metode lain yang bisa digunakan adalah dengan cara mencache data-data static seperti gambar dan teks kedalam sebuah server lain yang disebut reverse proxy. Cache adalah sebuah penyimpanan data sementara, sebagai contoh, ketika user mengakses sebuah halaman yang terdapat data berupa gambar maka gambar tersebut tidak perlu diambil dari web server tetapi langsung dikirim oleh reverse proxy, sehingga bisa akan mengurangi beban web server

Ngrok merupakan sebuah *proxy reverse*, yaitu menjadi perantara user-user di internet terhadap akses ke *webservice* yang berada pada *local area network*, sehingga seolah-olah *user* di internet mengakses langsung *web server* yang dimaksud padahal sesungguhnya *user* di internet mengakses *web server* yang terdapat di *local area network* melalui *proxy reverse* tersebut.

2.10 Sistem Operasi

Sistem operasi adalah suatu software sistem yang bertugas untuk melakukan control dan manajemen hardware serta operasi-operasi dasar sistem, termasuk menjalankan software aplikasi seperti program-program pengolahan kata dan browser web. Secara umum, sistem operasi adalah software pada lapisan pertama yang ditaruh dalam memori komputer pada saat komputer dijalankan. Sedangkan software lainnya dijalankan setelah sistem operasi berjalan, dan sistem operasi akan melakukan layanan inti umum untuk software tersebut.

2.10.1 Linux

Linux merupakan sistem operasi yang diciptakan oleh Linus Torvalds. Linux merupakan salah satu sistem operasi yang dapat diperbanyak tanpa harus mengeluarkan biaya pembelian software, karena bersifat open source. Sistem

operasi yang akan digunakan pada penelitian ini adalah Raspbian yang merupakan salah satu sistem operasi dari platform linux.

a. Raspbian

Raspbian adalah sistem operasi *open source* berbasis Debian GNU / Linux dan dioptimalkan untuk perangkat keras Raspberry Pi. Raspbian dilengkapi dengan lebih dari 35.000 paket atau perangkat lunak pre-compiled paket dalam format yang bagus untuk kemudahan instalasi pada Raspberry Pi. Awal dirilis sejak juni 2012, menjadi distribusi yang terus aktif dikembangkan dengan penekanan pada peningkatan stabilitas dan kinerja sebanyak mungkin. Meskipun Debian menghasilkan distribusi untuk arsitektur lengan, Raspberry Pi hanya kompatibel dengan versi yang lebih baru dari yang digunakan pada Raspberry Pi (ARMv7 CPU-A dan Raspberry Pi ARMv6 CPU yang lebih tinggi).

2.11 Bahasa Pemrograman Python

Guido van Rossum menciptakan Python sebagai bahasa pemrograman untuk penghitungan. Python merupakan bahasa pemrograman tingkat tinggi yang umum digunakan oleh para engineer dalam pembuatan aplikasi, bahkan beberapa perusahaan menggunakan python sebagai dasar pembuatan aplikasi komersial. Python merupakan *interpreted language* yang mana dapat menulis dan mengeksekusi program secara langsung pada mesin. Raspberry pi mendukung bahasa pemrograman python karena merupakan bahasa yang jelas dan mudah serta banyak pengguna yang saling berbagi kode program (Richardson and Wallace, 2013:33). The integrated development environment (IDE) yang tergabung dengan python dan raspberry pi adalah program IDLE. Raspberry memiliki dua versi python yaitu IDLE untuk python versi 2 dan IDLE 3 untuk python versi 3. Selain penulisan pada IDLE pengguna juga dapat menulis program pada *command line*. Proses *running* pada IDLE terkadang sangat lambat, maka pengguna dapat menulis dan menyimpan program dalam bentuk *.py* dan dijalankan menggunakan terminal. Python pada raspberry pi memiliki banyak pengaplikasian, diantaranya adalah pengaksesan basis data, penggunaan API mekanisme client/server terintegrasi, dan juga beberapa skrip internet pada

antarmuka HTTP. Selain pada Raspberri Pi dengan sistem operasi Raspbian yang dihuni oleh python terdapat beberapa sistem operasi lain yang dihuni oleh python yaitu Win 9x/NT/200, Unix, Macintosh, OS/2, DOS, PalmOS, Nokia mobile phones, Windows CE, Acorn/RISC OS, BeOS, Amiga, VMS/OpenVMS/ QNX, VxWorks., Psion, dan .NET Virtual Machine (Septian, 2013:1).

2.12 Raspberry Pi 2 B

Raspberry Pi merupakan *device embedded* sistem dalam jenis *single board computer*. Raspberry pi memiliki ukuran sebesar kartu kredit dengan harga yang sangat terjangkau sebagai mini komputer. Komputer mini ini telah mengeluarkan 3 versi sampai saat ini, yaitu raspberry pi model A, raspberry pi model B, dan raspberry pi 2. Berikut ini merupakan gambar dari Raspberry Pi 2 B.



Gambar 2.2 Raspberry Pi 2 B

(Sumber : Wikipedia, 2016)

Raspberry Pi 2 model B memiliki kapasitas RAM sebesar 1GB, Network Interface Card (NIC), slot SD Card yang berfungsi sebagai hardisk, pin GPIO sebagai *low level* peripheral dan fasilitas lainnya yang terdapat pada table berikut ini :

Tabel 2.1 Spesifikasi Raspberry Pi 2 B

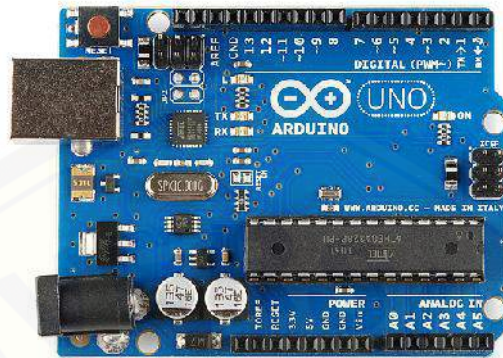
Spesifikasi Raspberry Pi 2 B	
<i>Sistem On Chip (SoC)</i>	Broadcom BCM2836
<i>Central Processing Unit (CPU)</i>	900 MHz Quad Core ARM Cortex-A7
GPU	Dual Core VideoCore IV
Memory	1GB LPDDR2
Port USB 2.0	4
<i>Video input</i>	Konektor CSI Input
<i>Video output</i>	HDMI (rev 1.3 & 1.4)
<i>Audio output</i>	HDMI 3.5 mm Jack
<i>Onboard Storage</i>	Micro SDIO
<i>Onboard network</i>	10/100 Ethernet (8P8C)
<i>Low-level peripheral</i>	40 pin 2.54 mm expansion header : 2x20 strip providing 27 GPIO pins as well as +3.3 V, +5V and GND supply lines
<i>Camera Connector</i>	15-pin MIPI Camera Serian Interface (CSI-2)
Suplai daya	5V melalui MicroUsb atau GPIO Header
Ukuran	85mm x 56mm x 17 mm
Berat	45 gram
Sistem operasi	Linux dan windows 10 IoT

(Sumber: Wikipedia,2016)

2.13 Arduino Uno R3

Arduino Uno R3 adalah *board system minimum* berbasis mikrokontroler ATmega328P jenis AVR. Arduino Uno R3 memiliki 14 digital input/output (6 diantaranya dapat digunakan untuk PWM output), 6 analog input, 16 MHz osilator kristal, USB connection, power jack, ICSP header dan tombol reset. Beberapa karakteristik dari Arduino uno R3 yaitu :

- a. Operating voltage 5 VDC
- b. Rekomendasi input voltage 7-12 VDC
- c. Batas input voltage 6-20 VDC
- d. Memiliki 14 buah input/output digital
- e. Memiliki 6 buah input analog
- f. DC current setiap I/O pin sebesar 40mA
- g. DC current untuk 3.3 V pin sebesar 50mA
- h. Flash memory 32 KB
- i. SRAM sebesar 1 KB
- j. EEPROM sebesar 1 KB



Gambar 2.3 Arduino Uno R3
(Sumber : Evans, 2011:2)

2.14 Data Logger

Data logger menyimpan data teknis dan sensor. Sebuah *data logger* adalah perangkat elektronik yang mencatat data dari waktu ke waktu atau dalam kaitannya dengan lokasi baik dengan *built in instrumen* atau sensor atau melalui instrumen eksternal dan sensor. Namun tidak sepenuhnya *data logger* didasarkan pada prosesor digital (atau komputer). *Data logger* umumnya adalah kecil, bertenaga baterai, portabel, dan dilengkapi dengan mikroprosesor, memori internal untuk penyimpanan data, dan sensor. Beberapa *data logger* antarmuka dengan komputer pribadi dan memanfaatkan perangkat lunak untuk mengaktifkan *data logger* dan melihat serta menganalisis data yang dikumpulkan, sementara yang lain memiliki perangkat antarmuka lokal (*keypad*, LCD) dan dapat digunakan sebagai perangkat yang berdiri sendiri.

Data logger bervariasi antara jenis tujuan umum untuk berbagai aplikasi pengukuran untuk perangkat yang sangat spesifik untuk mengukur dalam satu lingkungan atau jenis aplikasi saja. Hal ini umum untuk jenis tujuan umum harus diprogram, namun, masih banyak sebagai mesin statis dengan hanya sejumlah

terbatas atau tidak ada parameter berubah. *Data logger* elektronik telah menggantikan perekam grafik dalam banyak aplikasi. Salah satu manfaat utama menggunakan *data logger* adalah kemampuan untuk secara otomatis mengumpulkan data pada basis 24 jam. Setelah aktivasi, *data logger* biasanya digunakan dan ditinggalkan untuk mengukur dan merekam informasi selama periode pemantauan. Hal ini memungkinkan untuk gambar, komprehensif akurat dari kondisi lingkungan yang dipantau, seperti kecepatan angin (Pribadi, 2011).

2.15 SD Card

MicroSD adalah kartu memori *non-volatile* yang dikembangkan oleh SD Card Association yang digunakan dalam perangkat *portable*. Saat ini, teknologi microSD sudah digunakan oleh lebih dari 400 merek produk serta dianggap sebagai standar industri *defacto*.



Gambar 2.4 Micro SD Card

Keluarga microSD yang lain terbagi menjadi SDSC yang kapasitas maksimum resminya sekitar 2GB, meskipun beberapa ada yang sampai 4GB. SDHC (*High Capacity*) memiliki kapasitas dari 4GB sampai 32GB. Dan SDXC (*Extended Capacity*) kapasitasnya di atas 32GB hingga maksimum 2TB. Keberagaman kapasitas seringkali membuat kebingungan karena masing-masing protokol komunikasi sedikit berbeda.

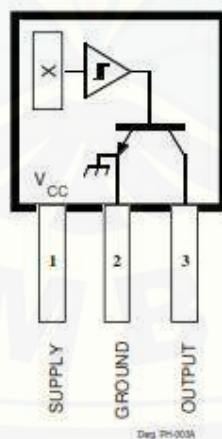
Dari sudut pandang perangkat, semua kartu ini termasuk kedalam keluarga SD. SD adapter memungkinkan konversi fisik kartu SD yang lebih kecil untuk bekerja di slot fisik yang lebih besar dan pada dasarnya ini adalah alat pasif yang

menghubungkan pin dari microSD yang kecil ke pin adaptor microSD yang lebih besar (Dharmawan, 2014).

2.16 Sensor *Hall Effect* A3144

Sensor *hall effect* yang digunakan penelitian ini merupakan sensor yang memiliki spesifikasi *on/off* atau *pulse*. Dimana kondisi yang diambil untuk mendapatkan data kecepatan adalah saat sensor *hall effect* mengenai magnet. *Hall effect* akan mengalami kondisi *low* saat menyentuh magnet dan kondisi *high* saat sensor *hall effect* tidak menyentuh magnet. Dari kondisi inilah akan membuat *pulse* dari kecepatan angin yang nantinya akan dirubah dari frekuensi ke dalam kecepatan dengan satuan meter *per second* (m/s). sensor *hall effect* A3144 memiliki tiga buah pin. Satu buah pin untuk masukan positif, satu untuk ground, dan satu untuk output data ke arduino.

Hasil keluaran dari sensor ini berupa nilai digital, jadi tidak diperlukan converter ADC untuk proses menghasilkan keluaran digital. Pada prosesnya hanya diperlukan pemrograman dan kalibrasi terhadap alat anemometer konvensional untuk mendapatkan nilai kecepatan angin yang presisi.



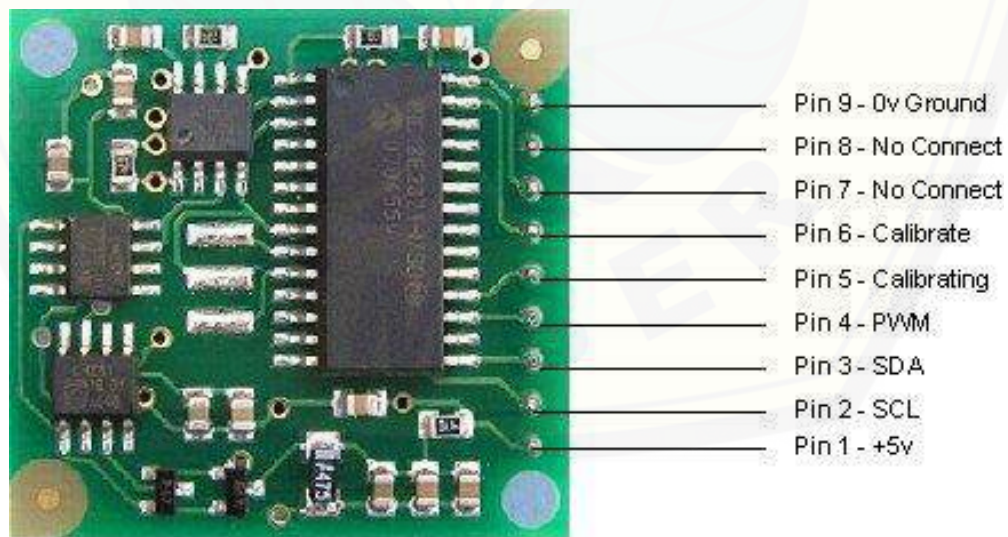
Gambar 2.5 Pin Sensor *Hall Effect* A3144
(Sumber : Allegro, 2005)

2.17 Sensor Kompas CMPS03

CMPS03 adalah Magnetic Compass buatan Devantech Ltd. CMPS03 yang berukuran 4x4 cm ini menggunakan sensor medan magnet Philips KMZ51 yang cukup sensitif untuk mendeteksi medan magnet bumi.

Kompas digital ini hanya perlu catu tegangan sebesar 5Vdc dengan konsumsi arus 15mA. Pada CMPS03 arah mata angin dibagi dalam bentuk derajat yaitu : Utara (0), Timur (90), Selatan (180) dan barat (270). Sensor ini menggunakan jalur komunikasi I2C ke mikrokontroler. Dengan adanya jalur komunikasi data I2C ini CMPS03 dapat dihubungkan langsung ke mikrokontroler yang telah memiliki jalur komunikasi I2C seperti ATmega328 yang sudah menjadi Arduino Uno.

Sensor CMPS03 memiliki jumlah pin sebanyak 9 dimana pin-pin yang digunakan adalah pin 1 untuk tegangan +5V, pin 2 SCL, pin 3 SDA dan pin 9 untuk tegangan GND. Dalam pemasangannya pin SDA dan SCL harus dalam kondisi *Pull Up* dimana harus mendapatkan tegangan positif yang telah diparalelkan dengan resistor 4k7. Sedangkan untuk melakukan kalibrasi bisa menggunakan pin 6 yang dihubungkan dengan pin GND.

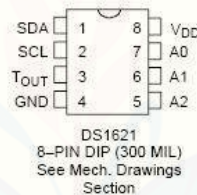


Gambar 2.6 Sensor kompas CMPS03
(Sumber : Soebhakti, 2008)

2.18 Sensor Suhu DS1621

IC DS1621 memungkinkan untuk besaran *temperature* yang diukur langsung diubah menjadi bentuk digital tanpa menggunakan lagi converter ADC (*Analog to Digital Converter*). Sensor produksi Dallas Semikonduktor ini cukup bagus karena bisa membaca range suhu mulai -55° sampai dengan 125° C. *Temperature* dibaca oleh DS1621 maka untuk komunikasi data dari digital sensor *temperature* dengan mikrokontroler menggunakan komunikasi data serial sinkron I2C Bus, yang menggunakan dua jalur data yaitu serial clock atau SCL dan serial data atau SDA sebagai jalur data untuk input dan output yang nantinya dihubungkan secara serial port pada mikrokontrol.

Pada sensor DS1621 secara keseluruhan memiliki delapan *wire pin connector*. Satu buah pin *power supply*, satu pin *ground*, satu pin SDA, satu pin SCL, satu pin *output thermostat*, dan 3 buah pin *chip address input*. Masing- masing pin memiliki fungsi masing-masing sesuai kegunaannya.



PIN DESCRIPTION

SDA	- 2-Wire Serial Data Input/Output
SCL	- 2-Wire Serial Clock
GND	- Ground
T _{OUT}	- Thermostat Output Signal
A0	- Chip Address Input
A1	- Chip Address Input
A2	- Chip Address Input
V _{DD}	- Power Supply Voltage

Gambar 2.7 Deskripsi PIN DS1621
(Sumber : Santosa, 2008:180)

2.19 Quality Of Service (QoS)

Berdasarkan pada rekomendasi CCITT E.800, Quality of Service (QoS) adalah efek kolektif dari kinerja layanan yang menentukan derajat kepuasan seorang pengguna terhadap suatu layanan. QoS menentukan performansi suatu

jaringan yang diukur. Pengukuran QoS menjadi dasar oleh para *professional network engineer* dalam membangun dan memperbaiki suatu jaringan komunikasi.

Pada penelitian ini menggunakan beberapa parameter pengukuran QoS pada layer transport (protokol TCP) dengan standarisasi TIPHON yaitu *Round Trip Time* (RTT) dan *throughput* TCP. RTT merupakan waktu estimasi yang dibutuhkan untuk mengirimkan paket data sampai diterimanya kembali ACK (*acknowledgment*) pada protokol TCP. RTT pada parameter QoS menggunakan satuan detik (*second*). Throughput adalah kecepatan pengiriman data yang *real* dari proses sebuah pengiriman data. Perhitungan throughput didapat dengan melakukan pembagian antara besar jumlah data yang berhasil dikirim dengan durasi waktu yang dibutuhkan untuk pengiriman. Satuan parameter *throughput* TCP adalah *bytes per second*. Data Parameter lain yang diukur adalah variasi RTT yaitu *time out*. *Time out* merupakan batas waktu pengiriman data sampai terjadinya proses *retransmission* pada protokol TCP. Parameter lain yang diukur adalah *Request Time Out* (RTO). RTO merupakan parameter untuk mengetahui persentase kegagalan transmisi segmen data antara *client* dan *server* sehingga akan menjalankan *retransmission time out*. RTO diukur dalam bentuk persen (%).

Berikut tabel kategori QoS berdasarkan standarisasi TIPHON pada parameter *latency end-to-end* dan *packet loss* :

Tabel 2.2 Klasifikasi QoS parameter *latency end-to-end*

Kategori <i>Latency</i>	Besar <i>Latency</i> (ms)	Indeks
Sangat Baik	< 150 ms	4
Baik	150 ms s/d 300 ms	3
Sedang	300 s/d 450 ms	2
Buruk	> 450 ms	1

(Sumber : TIPHON, 1999)

Tabel 2.3 Klasifikasi QoS parameter *packet loss*

Kategori <i>Packet Loss</i>	<i>Packet Loss</i>	Indeks
Sangat Baik	0-3 %	4
Baik	4-15 %	3
Sedang	16-25 %	2
Buruk	> 26 %	1

(Sumber : TIPHON, 1999)

BAB 3. METODE PENELITIAN

Dalam bab ini akan dijelaskan mengenai beberapa hal pokok penelitian, yaitu parameter atau objek penelitian, cara pengamatan variabel atau parameter, tempat dan waktu penelitian, langkah-langkah dalam pengumpulan data dan manajemen penelitian di lapangan, dan pengolahan data serta analisis data yang dipakai. Adapun uraian dari metode penelitian studi analisis ini adalah sebagai berikut :

3.1 Jenis Penelitian

Jenis penelitian yang akan dilakukan yaitu membangun sistem *real time monitoring* kecepatan angin, arah angin, dan suhu berbasis *web* menggunakan *proxy reserve* pada protokol *Transmission Control Protocol (TCP)*.

3.2 Tempat dan Waktu Penelitian

Penelitian ini dilakukan di Fakultas Teknik Universitas Jember dan di daerah Puger, tepatnya di pantai Pancer Puger Kabupaten Jember. Pengguna atau *client* berada di Fakultas Teknik Universitas Jember dan *server* berada di wilayah pantai Pancer Puger Kabupaten Jember. Pembangunan sistem *monitoring* ini dilakukan mulai bulan november 2015 sampai dengan februari.

Tabel 3.1 Jadwal pelaksanaan penelitian

No.	Kegiatan	Bulan/Minggu																			
		November				Desember				Januari				Februari				Maret			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	Tahap Persiapan																				
2.	Studi Literatur																				
3.	Perencanaan Alat																				
4.	Pembuatan Alat																				

f. Pengambilan Kesimpulan dan Saran

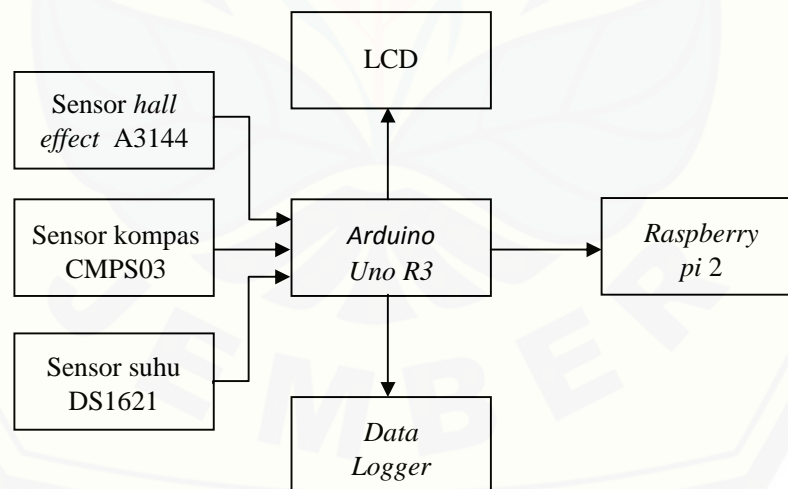
Pengambilan kesimpulan dan saran merupakan tahap terakhir dari penelitian ini, pengambilan kesimpulan ini didasarkan pada hasil analisa data yang ada. Sementara untuk saran digunakan untuk perbaikan-perbaikan yang mungkin terjadi, atau kemungkinan pengembangan dan aplikasi dengan metode maupun cara yang berbeda.

3.4 Alat dan Bahan Penelitian

Alat dan bahan yang digunakan pada penelitian ini adalah sebagai berikut.

3.4.1 Anemometer berbasis kecepatan angin, arah angin, dan suhu

Alat yang digunakan untuk memberikan data sebagai data yang dimonitoring adalah anemometer berbasis kecepatan angin, arah angin, dan suhu. Alat tersebut terdiri dari *arduino* uno R3 sebagai unit pengontrol, LCD sebagai penampil nilai, data logger sebagai penyimpan data, dan menggunakan tiga sensor yaitu sensor hall effect A3144, sensor CMPS03, dan sensor DS1621.

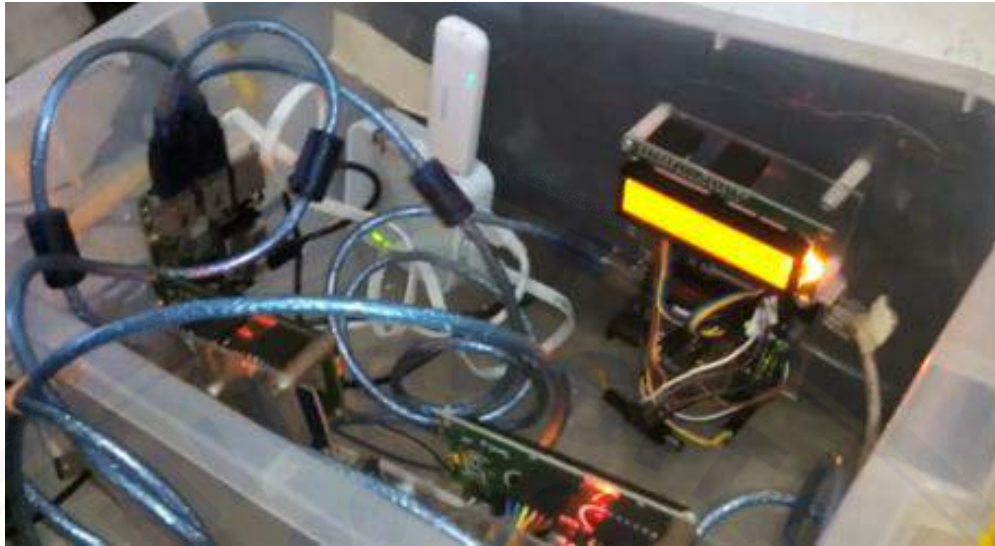


Gambar 3.1 Diagram blok sistem alat anemometer berbasis kecepatan angin, arah angin, dan suhu

Diagram blok pada Gambar 3.1 menjelaskan tentang rangkaian pada sistem alat anemometer. Sistem tersebut digunakan sebagai sistem masukan dan pengukuran awal dari sensor. Ketika mesin bekerja sensor A3144 (*hall effect*), Sensor CMPS03 (arah angin), dan sensor DS1621 (suhu) akan bekerja sesuai tugasnya masing-masing. Data yang dihasilkan adalah data digital dengan menggunakan komunikasi serial USB (*Universal Serial Bus*) pada *Arduino Uno*. Data digital yang terkumpul akan diolah oleh *Arduino Uno* agar dapat direkam dan disimpan oleh *data logger*. Data digital dari *arduino* juga ditampilkan pada LCD. Secara bersamaan data yang didapat oleh *arduino* langsung dikirim ke *server raspberry pi* melalui komunikasi serial USB.



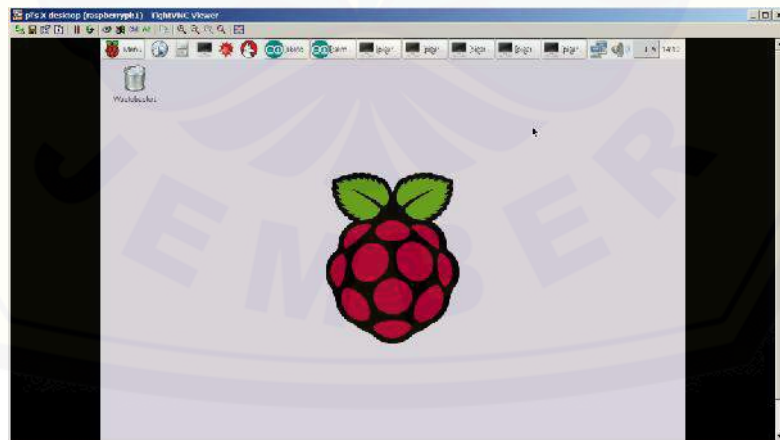
Gambar 3.2 Anemometer berbasis kecepatan angin, arah angin, dan suhu



Gambar 3.3 Sistem *arduino* dan *server*

3.4.2 *Raspberry pi 2*

Raspberry pi yang digunakan adalah *raspberry pi 2* versi B tahun 2104 dengan menggunakan sistem operasi Raspbian versi *Debian Wheezy*. *Interface* desktop *raspberry pi* menggunakan monitor dengan bantuan kabel HDMI to VGA. Tampilan *raspberry pi* juga dapat ditampilkan melalui *remote access* antara lain menggunakan SSH dan TightVNC dengan model *client/server*.

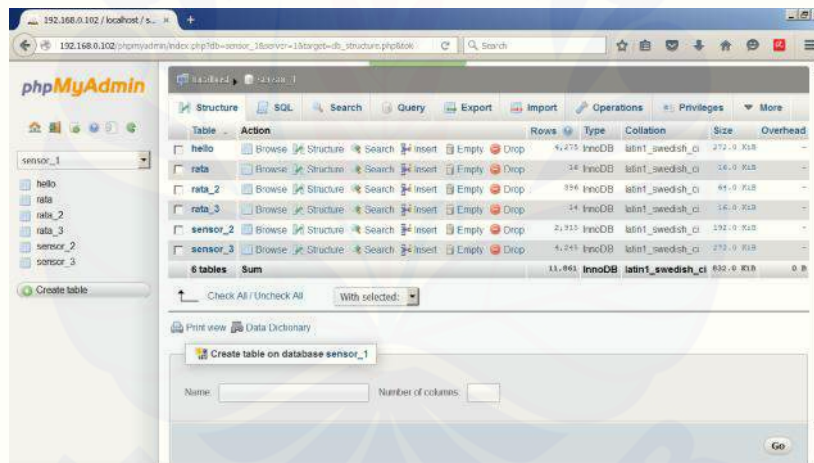


Gambar 3.4 Tampilan desktop *raspberry pi* remote session VNC

Raspberry pi 2 digunakan sebagai *server* untuk menyimpan dan mempublikasikan informasi yang dihasilkan dari alat anemometer ke internet

publik. Data digital yang dikirim ke *server raspberry pi* berasal dari *arduino uno* menggunakan komunikasi serial USB pada beudrate 9600. Port *arduino* pada *raspberry pi* adalah *ttyACM0*, *ttyACM1*, dan *ttyACM2* sesuai dengan urutan terhubungnya USB *arduino* ke port *raspberry pi*.

Raspberry pi membutuhkan beberapa aplikasi dalam membangun sebuah *server*. Aplikasi yang harus terinstal yaitu *web server apache*, PHP, *database MySQL*, dan *phpMyAdmin*. Apache digunakan untuk mengaktifkan *web lokal server* dengan alamat *localhost 127.0.0.1*. PHP digunakan untuk membangun *web server* dinamis yang terhubung dengan *database server*. MySQL merupakan *database server* yang digunakan untuk menyimpan data yang dikirim dari *arduino*. Aplikasi *phpMyAdmin* digunakan sebagai *management configuration* dari *database MySQL*.



Gambar 3.5 Tampilan *phpMyAdmin*

3.4.3 Portable Router TP-Link MR3020

Router TP-link MR3020 digunakan untuk *management routing internet protocol* pada *server raspberry pi*. Router dan *raspberry pi* terhubung menggunakan kabel LAN melalui port kedua *device*. *Raspberry pi* akan mendapatkan alamat IP lokal pada jaringan dibuat oleh router. Alamat IP lokal tersebut hanya dapat diakses oleh user yang berada pada jaringan yang sama. Pada

Berikut ini adalah rumus yang akan digunakan dalam perhitungan beberapa parameter tersebut:

a. *Round Trip Time* (RTT)

$$RTT_i = a * RTT_{i-1} + (1 - a) * rtt_i \dots\dots\dots (3.1)$$

Dimana :

RTT_i = Estimasi rata-rata nilai RTT ke-i

rtt_i = Waktu pengiriman paket ke-i sampai diterimanya *acknowledgement* paket ke-i

a = 0.875 (algoritma *Jacobson*)

RTT_0 = 0 (asumsi)

b. *Time out*

$$time\ out = RTT_{i-1} + 4 * MDEV_{i-1} \dots\dots\dots (3.2)$$

$$MDEV_i = (1 - \beta) * MDEV_{i-1} + \beta * |rtt_i - RTT_{i-1}| \dots\dots\dots (3.3)$$

Dimana :

$MDEV_i$ = Rata-rata deviasi

RTT_i = Estimasi rata-rata nilai RTT ke-i

rtt_i = Waktu pengiriman paket ke-i sampai diterimanya *acknowledgement* paket ke-i

β = 0.25 (algoritma *Jacobson*)

$MDEV_0$ = 0 (asumsi)

c. *TCP Throughput*

$$TCP\ Throughput = \frac{TCP\ RWND}{RTT} \dots\dots\dots (3.4)$$

Dimana:

$TCP\ RWND$ = *TCP receiver window size value*

RTT = Waktu pengiriman paket sampai diterimanya ACK

d. *Request Time Out* (RTO)

$$RTO = \frac{G}{B} \times 100\% \dots\dots\dots (3.5)$$

G = Jumlah kesalahan pengiriman data

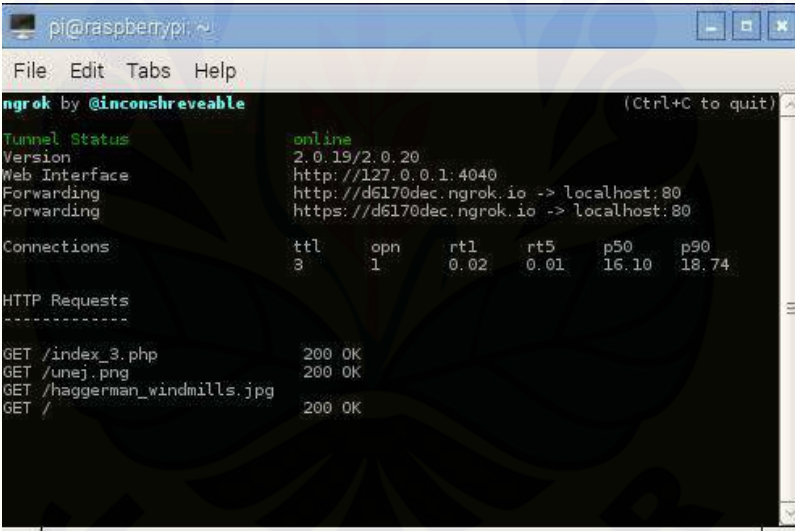
B = Jumlah total data yang dikirim

3.4.5 Laptop

Laptop digunakan untuk menampilkan *interface* dari *raspberry pi* menggunakan *remote session*. Laptop juga digunakan untuk *maintenance* program, menjalankan program, dan mengecek tampilan *web monitoring*.

3.4.6 Localhost Tunnel Ngrok

Ngrok merupakan aplikasi yang digunakan untuk mempublikasikan *local web server* yang telah dibuat melalui jaringan tunnel pada alamat *xxxxx.ngrok.io*. *Ngrok* memberikan *proxy reserve* untuk digunakan pada *localhost server* sebagai IP publik. *Localhost* pada *server* dapat diakses melalui alamat HTTP *port forwarding* yang diberikan oleh *ngrok* pada port 80.



```
pi@raspberrypi: ~  
File Edit Tabs Help  
ngrok by @inconsheveable (Ctrl+C to quit)  
Tunnel Status      online  
Version            2.0.19/2.0.20  
Web Interface      http://127.0.0.1:4040  
Forwarding         http://d6170dec.ngrok.io -> localhost:80  
                   https://d6170dec.ngrok.io -> localhost:80  
Connections  
-----  
t1l  opn  rt1  rt5  p50  p90  
3    1    0.02 0.01 16.10 18.74  
HTTP Requests  
-----  
GET /index_3.php      200 OK  
GET /unej.png         200 OK  
GET /haggerman_windmills.jpg 200 OK  
GET /                 200 OK
```

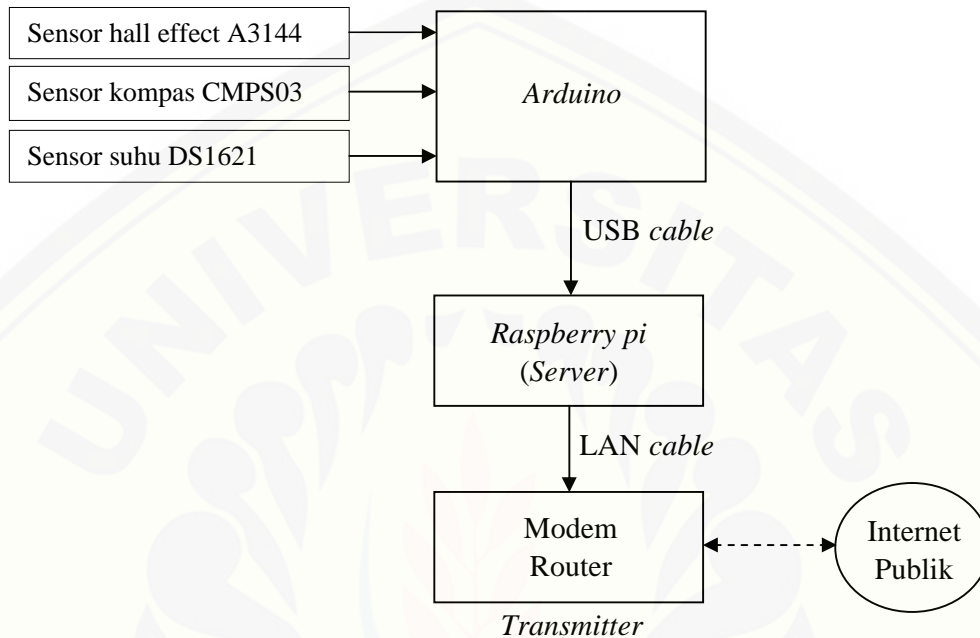
Gambar 3.8 Port forwarding http pada Ngrok

Reserve proxy ngrok dalam kondisi aktif jika *tunnel status* pada *interface ngrok* dalam mode *online* yang berwarna hijau. *Ngrok* dalam kondisi pada saat sinyal internet lemah jika *tunnel status* menunjukkan tulisan *reconnecting* berwarna merah. *Web interface ngrok* yang sudah terinstal berada pada alamat <http://127.0.0.1:4040>. *Web interface* tersebut menyajikan trafik *web* yang terjadi pada alamat yang diberikan *ngrok* kepada *client*.

3.5 Diagram Blok Sistem

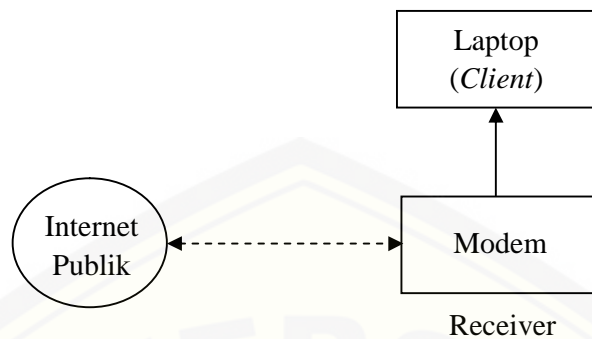
Diagram blok sistem pada penelitian ini adalah sebagai berikut.

3.5.1 Diagram blok Sistem



Gambar 3.9 Diagram blok *transmitter* sistem *monitoring*

Blok sistem pada gambar 3.9 menjelaskan proses *transmitter* sistem *realtime monitoring* yang dibangun. Proses awal adalah ketika alat dihidupkan, sensor pada *arduino* mulai bekerja. Data hasil pembacaan sensor hall effect A3144 untuk kecepatan angin, sensor suhu DS1621, dan sensor kompas CMPS03 pada *arduino* dikirimkan ke *server raspberry pi*. Komunikasi serial yang digunakan untuk menghubungkan antara *arduino* dengan *raspberry pi* adalah USB. Setelah data terkirim ke *raspberry pi*, kemudian *server* akan menyimpannya dalam *database* dan mengunggah data yang dikirim dari *arduino* pada tampilan *website local server*. Modem router digunakan sebagai media koneksi *remote access* antara laptop dan *raspberry pi*. Modem router juga digunakan sebagai *transmitter* untuk mengirim data ke internet publik.



Gambar 3.10 Diagram blok *receiver*

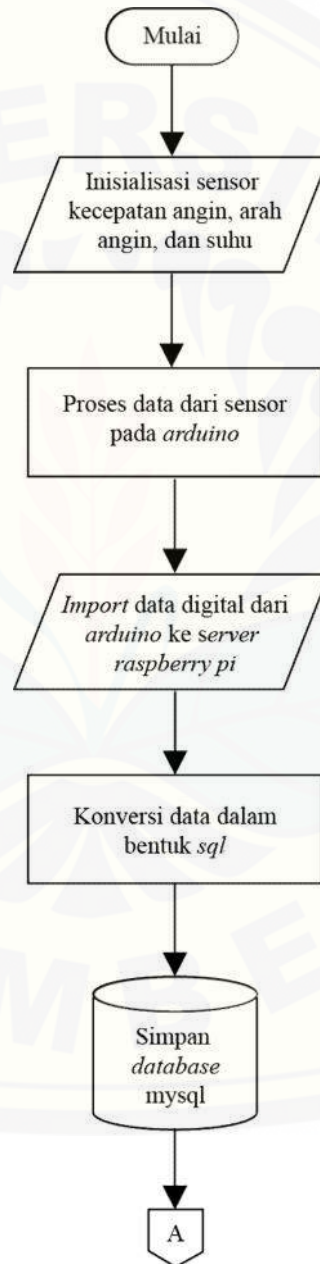
Gambar 3.10 menjelaskan Diagram blok *receiver*. Setelah lokal *web server* berhasil dikirim maka *client* dapat menerima data data dari *server* menggunakan modem sebagai *receiver* data dari internet publik. *Client* dapat melakukan *monitoring* kecepatan angin, arah angin, dan suhu secara jarak jauh menggunakan *web* dimanapun *client* berada selama mendapatkan akses data internet.

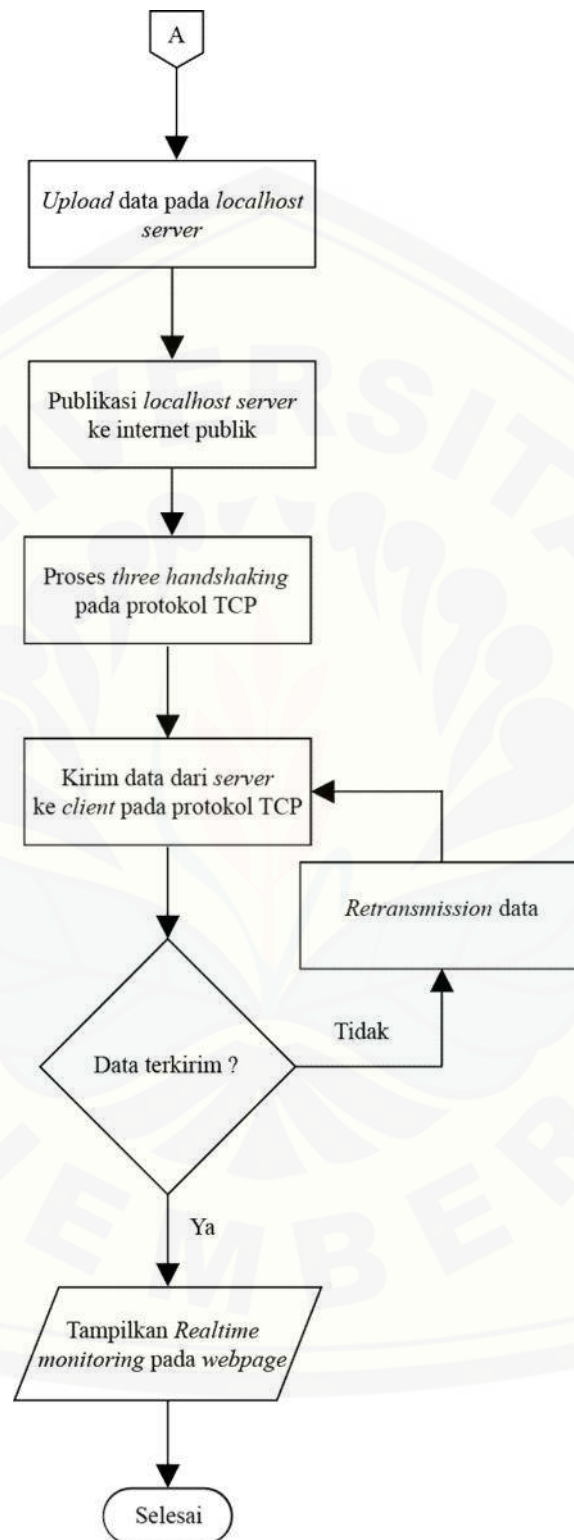
3.6 Diagram Alir Sistem *Monitoring*

Flowchart pada Gambar 3.11 menjelaskan alur pengiriman data dari *arduino* ke *client*. Proses awal adalah pengolahan data dari sensor pada *arduino*. Selanjutnya *server* menjalankan program import data dari *arduino* ke *raspberry pi*. Setelah program berjalan, data yang diterima oleh *raspberry pi* akan ditampilkan pada *command prompt*. *Raspberry pi* akan mengubah format data tersebut dalam format *sql* sehingga data dapat disimpan pada *database mysql*. Setelah data berhasil disimpan data akan langsung diupload ke dalam *localhost server*.

Server akan mempublikasikan ke internet publik *localhost* tersebut menggunakan *proxy reserve ngrok*. *Server* akan mendapatkan alamat *port forwarding* yang digunakan untuk publikasi *localhost server*. *Client* dapat mengakses alamat *port forwarding* tersebut pada internet publik. Sebelum *client* mendapat data yang diinginkan *client/server* harus melewati proses *three handshaking* pada protokol TCP agar terjadi koneksi antara *client* dan *server*.

Setelah proses *three handshaking*, *server* dan *client* dapat saling mengirimkan data. Apabila data tidak terkirim maka akan terjadi *retransmission*/pengiriman ulang data pada protokol TCP sampai data terkirim. Pada akhirnya *Client* akan dapat mengakses *web* lokal *server* dan melakukan *monitoring*.



Gambar 3.11 Diagram Alir Sistem *Monitoring*

BAB 5. PENUTUP

5.1 Kesimpulan

Setelah melakukan perencanaan dan pembuatan sistem kemudian dilakukan pengujian dan analisa, maka dari hasil tersebut dapat diambil beberapa kesimpulan sebagai berikut:

1. Penggunaan *proxy reserve* sebagai media publikasi lokal *web server/localhost* pada penelitian ini bisa dikatakan *real time* karena memiliki respon waktu dengan rata-rata 0.395 s.
2. Hasil analisa protokol TCP pada penelitian ini menghasilkan nilai paling baik pada kategori waktu malam dengan nilai rata-rata RTT 0.079 s, rata-rata *time out* 0.418 s, dan rata-rata *throughput* TCP 607.873 KBps. Sedangkan persentase kesalahan pengiriman terbesar terjadi pada kategori waktu siang dengan nilai RTO sebesar 4.37 persen.
3. Klasifikasi QoS pada parameter RTT menghasilkan kategori sangat baik pada waktu pagi, siang, dan malam dengan nilai rata-rata sebesar 104 ms. Klasifikasi QoS pada parameter RTO menghasilkan kategori sangat baik pada waktu pagi dengan nilai RTO sebesar 1.25%, sedangkan menghasilkan kategori baik pada waktu siang dengan RTO sebesar 4.37% dan waktu malam dengan RTO sebesar 3.75%.

5.2 Saran

Pada penelitian ini masih terdapat beberapa kekurangan atau kendala, berikut ini merupakan saran untuk pengembangan yang lebih lanjut:

1. *Server raspberry pi* yang digunakan memiliki kelemahan untuk proses *running program* ulang fungsi *server* pada saat sistem *restart*, disarankan memperbaiki pada sisi otomatisasi *running program*.
2. Sistem *reserve proxy ngrok* memiliki kelemahan dalam memberikan alamat *Domain Name System (DNS)* yang berubah-ubah, disarankan menggunakan *server hosting* yang berbayar dan membuat *web app* sendiri agar mempermudah dari sisi *client*.

DAFTAR PUSTAKA

- Badan Pengkajian dan Penerapan Teknologi. 2014. *Outlook Energi Indonesia 2014 : Pengembangan Energi untuk Mendukung Program Substitusi BBM*, Jakarta : Pusat Teknologi Pengembangan Sumberdaya Energi (PTPSE) dan Badan Pengkajian dan Penerapan Teknologi (BPPT).
- Dhewiberta Hardjono. 2006. *Seri Panduan Lengkap Menguasai Pemrograman Web dengan PHP 5*. Yogyakarta: Andi.
- Evans Brian. 2011. *Beginning Arduino Programming*. New York : Springer Science + Bussines Media, LLC and Apress.
- Richardson, M., and Wallace, S. 2013. *Getting Started with Raspberry Pi*. California : O'Reilly Media. Inc.
- Sayood, K. 2003. *Lossless Compression Handbook*. New York : Academic Press.
- Septian, R.F. 2013. *Belajar Pemrograman python Dasar*. Bandung : POSS – UPI.
- Suntana, E. 2005. *Komunikasi Data & Jaringan Komputer*. Yogyakarta : Graha Ilmu.
- TIPHON, 1999, “*Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) General aspects of Quality of Service (QoS)*”, DTR/TIPHON05006. France : ETSI.
- Constantine, *et al.* 2011. Framework for TCP Throughput Testing. *Internet Engineering Task Force (IETF)*, ISSN 2070-1721 : 1-27.
- Jaloun, M dan Zouhair, G. 2010. Wireless Mobile Evolution to 4G Network. *Wireless Sensor Network*, 2(1) : 309-317
- Lukin Stephanie. 2010. *A Comparison of Round-Trip Time Estimation Algorithms. Research Experiences for Undergraduates Site, NSF grant CNS-0852099*, UCSC SURF-IT.
- Pambudi, I.R., Yudi, N., Mitra, D. 2012. Sistem Telemetri Pemantau Gempa Menggunakan Jaringan GSM. *J. Auto.Ctrl.Inst*, 4(1) : 45-50.
- Permana, T. D. 2014. Sistem Monitoring Menggunakan Mini PC Raspberry Pi. *Jurnal Teknik computer Unikom*, 3(1) : 1-6.

- Permatasari, G.K., Sukiswo, dan Santoso Imam. 2014. Analisis Kinerja *TCP Westwood* untuk Pencegahan Kongesti pada Jaringan LTE dengan Menggunakan *Network Simulator 2.33 (NS2.33)*. *TRANSIENT*, 3(3) : 350-357.
- Sirajuddin, Affandi Achmad, dan Setijadi Eko. 2012. Rancang Bangun Server *Learning Management System* Menggunakan *Load Balancer* dan *Reserve Proxy*. *Jurnal Teknik ITS*, ISSN 2301-9271 : A50-A51.
- Susandi Herman dan Pinem Maksum. 2014. Analisis Kualitas layanan Data pada Jaringan Telekomunikasi Berbasis CDMA EVDO Rev.A. *Singuda Ensikom*, 6(2) : 93-98.
- Gunawan. 2015. “Analisa Potensi Energi Angin dan Estimasi Energi Listrik Berdasarkan Perbedaan Ketinggian di Pantai Puger Kabupaten Jember”. Tidak Diterbitkan, Skripsi. Jember: Universitas Jember.
- Husumardiana, D. 2015. “Analisa Packetloss Sistem Telemetry pada Perangkat Pengukur Kecepatan Angin Berbasis X-Bee Pro Menggunakan Kalman Filter”. Tidak Diterbitkan, Skripsi. Jember: Universitas Jember.
- Wicaksono, A.W, 2015. “Desain Sistem *Monitoring* Debit Air Menggunakan Protokol HTTP”. Tidak Diterbitkan, Skripsi. Jember: Universitas Jember.
- Pemimpin Redaksi. “Potensi Energi Indonesia”. *ESDMMAG*. Edisi 2 2012. Halaman 12.
- Allegro. 2005. *A3141, A3142, A3143, and A3144*. <http://www.allegromicro.com/~media/Files/Datasheets/A3141-2-3-4-Datasheet.ashx?la=en>. [23 Oktober 2015].
- Soebhakti Hendawan. 2008. *Digital Compass CMPS03*. <https://hendawan.files.wordpress.com/2009/01/digital-compass-application3.pdf>.
- Wikipedia. 2016. *Raspberry Pi*. https://en.wikipedia.org/wiki/Raspberry_Pi. diakses pada 15 maret 2016 : 06.52.

LAMPIRAN

LAMPIRAN A. LISTING PROGRAM RASPBERRY PI SERVER

A.1 Listing Program Terima Data Server

```
import time
import serial
import MySQLdb as mdb
ser = serial.Serial('/dev/ttyACM0', 9600)
while True:
    data = ser.readline()
    a,b,c = data.split(",")
    print(a,b,c)
    con = mdb.connect('localhost', 'pi_insert', 'fadhil', 'sensor_1');
    cur = con.cursor()
    cur.execute("""INSERT INTO hello(hellopi, arah, ok) VALUES(%s,
%s, %s)""", (a,b,c))
    con.commit()
    time.sleep(2)
con.close()
print('end')
```

A.2 Listing Program index.html

```
<html>
<head>
<title>TEKNIK ELEKTRO UNEJ</title>
<link rel="icon" type="image/png" href="unej.png">
<style type="text/css">
#footer {
    bottom:0;
    height:20px; /* tinggi dari footer */
```

```
    }
</style>

</head>
<body>
<h1 align="center"><font color="black"> Web Based System of Realtime
Monitoring Wind Speed, Wind Direction, and Temperature in The Pancer
Beach</font></h1>
<hr>
<div id="menu">
<a href="index_3.php?p=index_kecepatan3">KECEPATAN</a> /
<a href="index_3.php?p=index_suhu3">SUHU</a> /
<a href="index_3.php?p=tabel_3">DATA TABEL</a> /
<a href="index_3.php?p=tabel_rata3">DATA RATA TABEL</a> /
<hr>
<script type='text/javascript'>
    <!--
    var months = ['Januari', 'Februari', 'Maret', 'April', 'Mei', 'Juni', 'Juli', 'Agustus',
'September', 'Oktober', 'November', 'Desember'];
    var myDays = ['Minggu', 'Senin', 'Selasa', 'Rabu', 'Kamis', 'Jumat', 'Sabtu'];
    var date = new Date();
    var day = date.getDate();
    var month = date.getMonth();
    var thisDay = date.getDay(),
        thisDay = myDays[thisDay];
    var yy = date.getYear();
    var year = (yy < 1000) ? yy + 1900 : yy;
    document.write(thisDay + ', ' + day + ' ' + months[month] + ' ' + year);
    //-->
</script>
```

```
<br>
<div id="clock" align="left"></div>
<script type="text/javascript">
<!--
function showTime() {
    var a_p = "";
    var today = new Date();
    var curr_hour = today.getHours();
    var curr_minute = today.getMinutes();
    var curr_second = today.getSeconds();
    if (curr_hour < 12) {
        a_p = "AM";
    } else {
        a_p = "PM";
    }
    if (curr_hour == 0) {
        curr_hour = 12;
    }
    if (curr_hour > 12) {
        curr_hour = curr_hour - 12;
    }
    curr_hour = checkTime(curr_hour);
    curr_minute = checkTime(curr_minute);
    curr_second = checkTime(curr_second);
    document.getElementById('clock').innerHTML=curr_hour + ":" +
curr_minute + ":" + curr_second + " " + a_p;
}
function checkTime(i) {
    if (i < 10) {
```

```
        i = "0" + i;
    }
    return i;
}
setInterval(showTime, 500);
//-->
</script></br>
</div>
<div id="konten">
    <?php
        $pages_dir = 'pages';
        if(!empty($_GET['p'])){
            $pages = scandir($pages_dir, 0);
            unset($pages[0], $pages[1]);

            $p = $_GET['p'];
            if(in_array($p.'.php', $pages)){
                include($pages_dir.'/'.$p.'.php');
            } else {
                echo 'Halaman tidak ditemukan! :(';
            }
        } else {
            include($pages_dir.'/blank.php');
        }
    ?>

<br>
<hr>
<div id="footer">
 <br>
```

```
<hr>
</div>
</div>
</body>
</html>
```

A.3 Listing program index_3.php

```
<html>
<head>
  <title>TEKNIK ELEKTRO UNEJ</title>
  <link rel="icon" type="image/png" href="unej.png">

  <style type="text/css">

  #footer {
    bottom:0;
    height:20px; /* tinggi dari footer */

  }
</style>

</head>
<body>

  <h1>Anemometer pada Ketinggian 43.2 Meter</h1>
<hr>
  <div id="menu">
    <a href="index_3.php?p=index_kecepatan3">KECEPATAN</a>
  /
  <a href="index_3.php?p=index_suhu3">SUHU</a> /
  <a href="index_3.php?p=tabel_3">DATA TABEL</a> /
```



```
<a href="index_3.php?p=tabel_rata3">DATA RATA TABEL</a> /
<hr>
</div>
<div id="konten">
    <?php
    $pages_dir = 'pages';
    if(!empty($_GET['p'])){
        $pages = scandir($pages_dir, 0);
        unset($pages[0], $pages[1]);

        $p = $_GET['p'];
        if(in_array($p.'.php', $pages)){
            include($pages_dir.'/'.$p.'.php');
        } else {
            echo 'Halaman tidak ditemukan! :(';
        }
    } else {
        include($pages_dir.'/blank2.php');
    }
    ?>

<br>
<hr>
<div id="footer">
 <br>
TEKNIK ELEKTRO UNIVERSITAS JEMBER
<hr>
</div>
</div>
</body>
</html>
```

A.4 Listing program link kecepatan

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
  <title></title>
</head>
<body>
  <div id="chartContainer" style="width: 800px; height: 380px;"></div><br>
  <div id="chartContainer2" style="width: 800px; height: 380px;"></div>
  <script src="jquery-1.10.2.js"></script>
  <script src="jquery.canvasjs.min.js"></script>
  <script type="text/javascript">
$(document).ready(function () {
  var waktu = setInterval(cek, 1000);
  var waktu2 = setInterval(cek2, 5000);
  var ceka = false;
  var ceka2 = false;
  jalankan();
  jalankan2();
  function cek () {
    if (ceka) {
      jalankan();
    }
  }
  function cek2 () {
    if (ceka2) {
      jalankan2();
    }
  }
  function jalankan() {
```

```
ceka = false;

$.getJSON("hasil_kecepatan3.php", function (result) {
    ceka = true;

    var chart = new CanvasJS.Chart("chartContainer", {
        title: {
            text: "loading"
        },
        axisX: {
            labelFontSize: 10
        },
        data: [
            {
                'type': "spline",
                'showInLegen': true,
                'name': "Loading",
                'legendText' : "Loading",
                dataPoints: result
            }
        ]
    });

    chart.render();
}).error(function() { ceka = true; });
}
```

```
function jalankan2() {
    ceka2 = false;
```

```
$.getJSON("rata_utama_kecepatan3.php", function (result)
{
    ceka2 = true;

    var chart = new CanvasJS.Chart("chartContainer2",
    {
        title: {
            text: "loading"
        },
        axisX: {
            labelFontSize: 10
        },
        data: [
            {
                'type':"spline",
                'showInLegen':true,
                'name': "Loading",
                'legendText' : "Loading",
                dataPoints: result
            }
        ]
    });

    chart.render();
}).error(function() { ceka2 = true; });
}

});
</script>
```

```
</body>
```

```
</html>
```

A.5 Listing Program Link Suhu

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml" >
```

```
<head>
```

```
  <title></title>
```

```
</head>
```

```
<body>
```

```
  <div id="chartContainer" style="width: 800px; height: 380px;"></div><br>
```

```
  <div id="chartContainer2" style="width: 800px; height: 380px;"></div>
```

```
  <script src="jquery-1.10.2.js"></script>
```

```
  <script src="jquery.canvasjs.min.js"></script>
```

```
  <script type="text/javascript">
```

```
    $(document).ready(function () {
```

```
      var waktu = setInterval(cek, 1000);
```

```
      var waktu2 = setInterval(cek2, 5000);
```

```
      var ceka = false;
```

```
      var ceka2 = false;
```

```
      jalankan();
```

```
      jalankan2();
```

```
      function cek () {
```

```
        if (ceka) {
```

```
          jalankan();
```

```
        }
```



```
}
function cek2 () {
    if (ceka2) {
        jalankan2();
    }
}
function jalankan() {
    ceka = false;

    $.getJSON("hasil_suhu3.php", function (result) {
        ceka = true;
        var chart = new CanvasJS.Chart("chartContainer", {
            title: {
                text: "loading"
            },
            axisX: {
                labelFontSize: 10
            },
            data: [
                {
                    'type': "spline",
                    'showInLegen': true,
                    'name': "Loading",
                    'legendText' : "Loading",
                    dataPoints: result
                }
            ]
        });
    });
};
```

```
        chart.render();
    }).error(function() { ceka = true; });

}

function jalankan2() {
    ceka2 = false;

    $.getJSON("rata_suhu3.php", function (result) {
        ceka2 = true;

        var chart = new CanvasJS.Chart("chartContainer2",
        {
            title:{
                text: "loading"
            },
            axisX:{
                labelFontSize: 10
            },
            data: [
                {
                    'type':"spline",
                    'showInLegen':true,
                    'name': "Loading",
                    'legendText' : "Loading",
                    dataPoints: result
                }
            ]
        });
    });
}
```

```
                chart.render();
            }).error(function() { ceka2 = true; });
        }
    });
</script>
</body>
</html>
```

A.6 Listing Program Link Data Tabel

```
<?php
$dbHost = "localhost";
$dbUser = "pi_select";
$dbPass = "fadhil";
$dbName = 'sensor_1';

// membuat koneksi mysql
$conn = mysql_connect($dbHost, $dbUser, $dbPass, $dbName);

//membuat koneksi database
$dbSelected = mysql_select_db($dbName, $conn);

// Mengecek koneksi database
if (!$dbSelected) die ('Koneksi Gatabase Gagal: ' . mysql_error());
else echo "DATA RECORD PADA TABEL<br/>";

//membuat query membaca record dari tabel User
$query="SELECT * FROM sensor_3";

//menjalankan query
```

```
if (mysql_query($query)) {
$result=mysql_query($query);
} else die ("Error menjalankan query". mysql_error());

//mengecek record kosong
if (mysql_num_rows($result) > 0)
{
//membuat tabel dan heading
echo "<table border=1>";
echo "<tr>";
echo "<th>KECEPATAN (m/s)</th>";
echo "<th>SUHU (C)</th>";
echo "<th>ARAH (degree)</th>";
echo "<th>DATE</th>";
echo "</tr>";
//menampilkan hasil query
while($row = mysql_fetch_array($result)) {
echo "<tr>";
echo "<td>".$row["kecepatan"]."</td>";
echo "<td>".$row["suhu"]."</td>";
echo "<td>".$row["arah"]."</td>";
echo "<td>".$row["dtg"]."</td>";
echo "</tr>";
}
echo "</table>";
}
else echo "Tidak ada Record didalam tabel";

//menutup koneksi mysql
mysql_close($conn);
?>
```

A.7 Listing Program Link Data Rata Tabel

```
<?php
$dbHost = "localhost";
$dbUser = "pi_select";
$dbPass = "fadhil";
$dbName = 'sensor_1';

// membuat koneksi mysql
$conn = mysql_connect($dbHost, $dbUser, $dbPass, $dbName);

//membuat koneksi database
$dbSelected = mysql_select_db($dbName, $conn);

// Mengecek koneksi database
if (!$dbSelected) die ('Koneksi Gatabase Gagal: ' . mysql_error());
else echo "DATA RECORD RATA-RATA PER 10 MENIT  PADA TABEL
<br/>";

//membuat query membaca record dari tabel User
$query="SELECT * FROM rata_3";

//menjalankan query
if (mysql_query($query)) {
$result=mysql_query($query);
} else die ("Error menjalankan query". mysql_error());

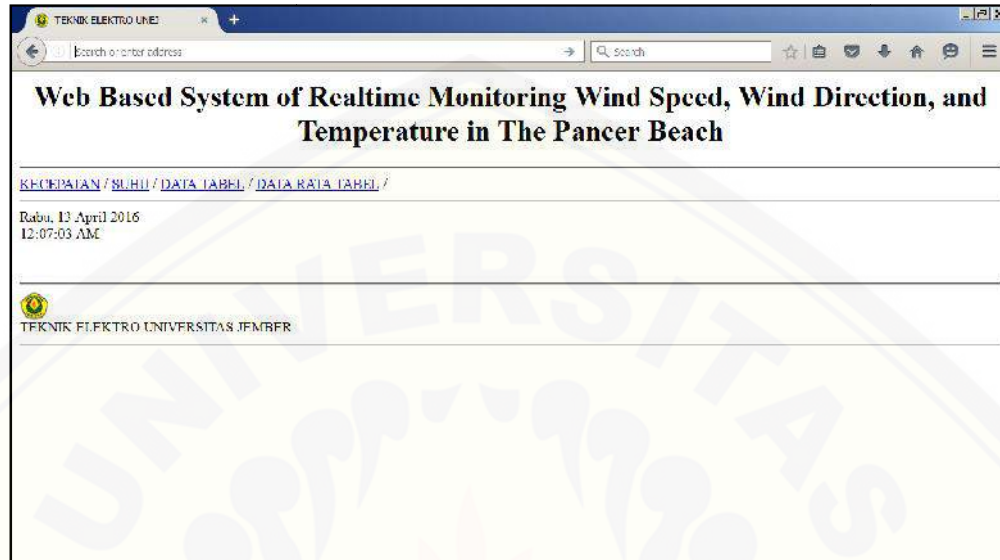
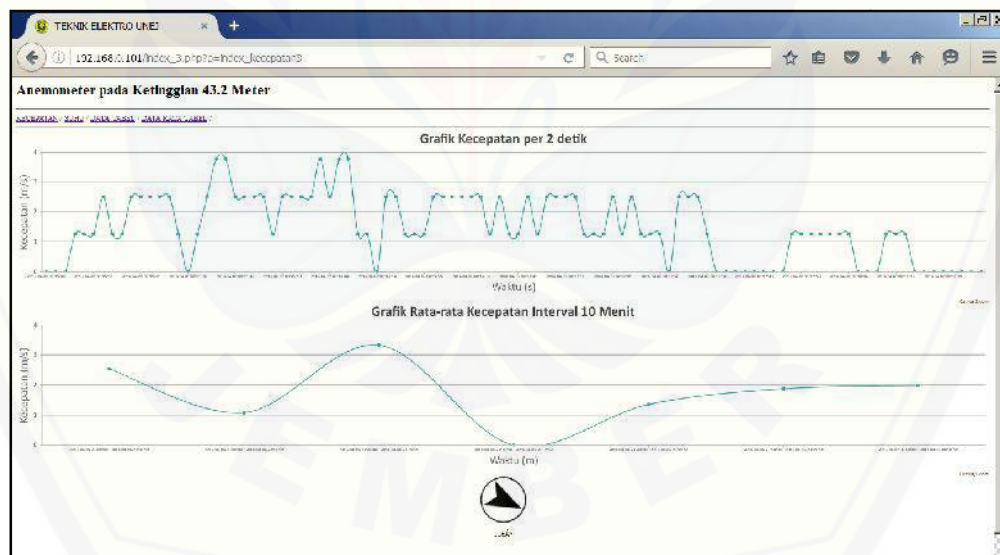
//mengecek record kosong
if (mysql_num_rows($result) > 0)
{
//membuat tabel dan heading
```

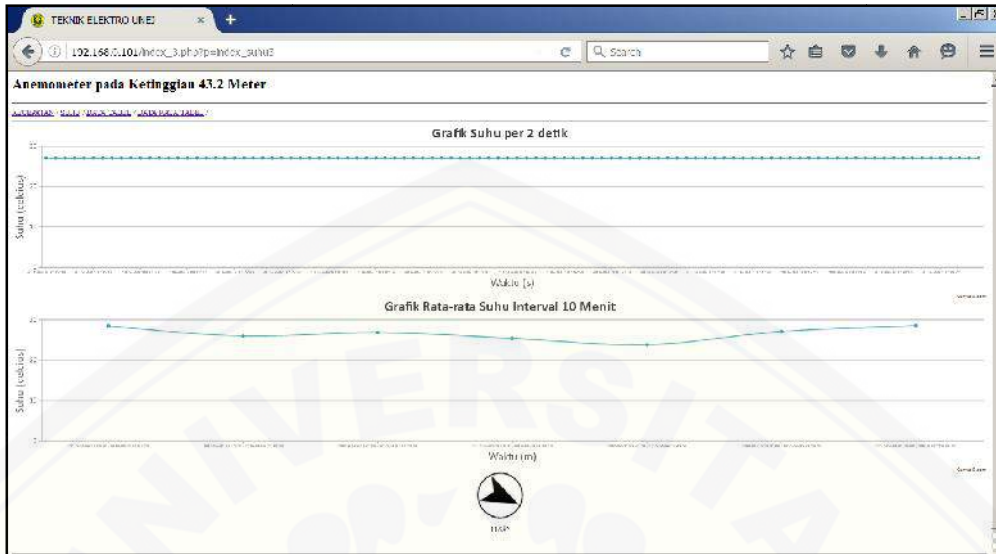


```
echo "<table border=1>";
echo "<tr>";
echo "<th>KECEPATAN (m/s)</th>";
echo "<th>SUHU (C)</th>";
echo "<th>ARAH (degree)</th>";
echo "<th>DATE1</th>";
echo "<th>DATE2</th>";
echo "</tr>";

//menampilkan hasil query
while($row = mysql_fetch_array($result)) {
    echo "<tr>";
    echo "<td>".$row["kecepatan"]."</td>";
    echo "<td>".$row["suhu"]."</td>";
    echo "<td>".$row["arah"]."</td>";
    echo "<td>".$row["dtg1"]."</td>";
    echo "<td>".$row["dtg2"]."</td>";
    echo "</tr>";
}
echo "</table>";
}
else echo "Tidak ada Record didalam tabel";

//menutup koneksi mysql
mysql_close($conn);
?>
```

LAMPIRAN B. TAMPILAN PADA *MONITORING WEBPAGE*Gambar B.1 Tampilan index.html pada *monitoring webpage*Gambar B.2 Tampilan link kecepatan pada *monitoring webpage*

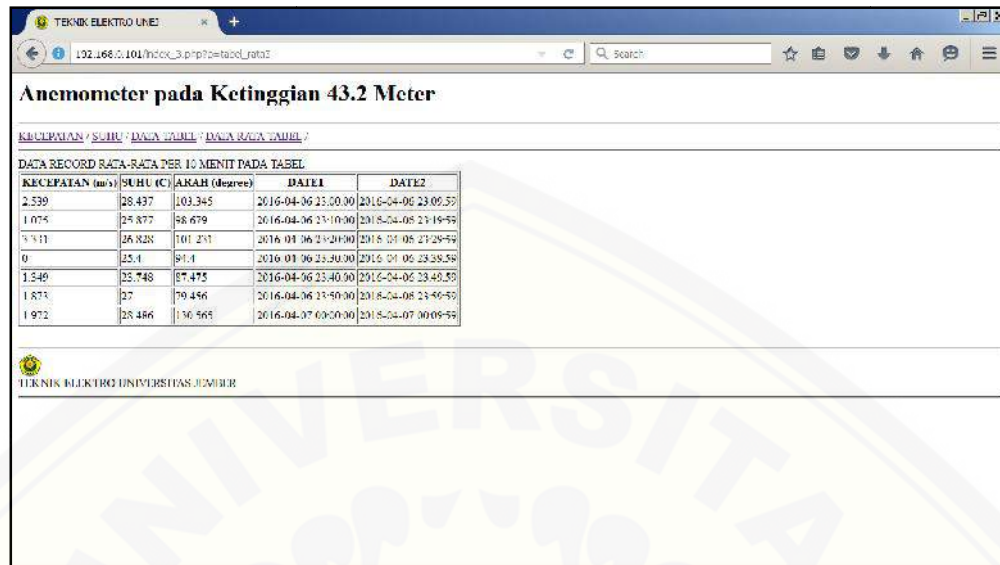


Gambar B.3 Tampilan link suhu pada *monitoring webpage*

The figure shows a data table from a web monitoring interface. The table is titled "DATA RECORD PADA TABEL" and has four columns: "KECEPATAN (m/s)", "SUHU (C)", "ARAH (degree)", and "DATE". The data rows show various readings over time, with speed values ranging from 0 to 2.51 m/s, temperature values from 75 to 78°C, and direction values from 75 to 79 degrees. The dates are all from 2016-04-06.

KECEPATAN (m/s)	SUHU (C)	ARAH (degree)	DATE
0	27	75	2016-04-06 23:57:58
0	27	75	2016-04-06 23:58:00
1.77	27	76	2016-04-06 23:58:02
2.51	27	80	2016-04-06 23:58:05
2.51	27	81	2016-04-06 23:58:07
1.77	27	87	2016-04-06 23:58:09
1.77	27	70	2016-04-06 23:58:11
2.51	27	77	2016-04-06 23:58:13
2.51	27	77	2016-04-06 23:58:15
0	27	76	2016-04-06 23:58:17
0	27	78	2016-04-06 23:58:19
1.28	27	77	2016-04-06 23:58:21
0	27	77	2016-04-06 23:58:23
0	27	79	2016-04-06 23:58:25
1.28	27	79	2016-04-06 23:58:28
2.51	27	76	2016-04-06 23:58:30
2.51	27	77	2016-04-06 23:58:32
2.51	27	77	2016-04-06 23:58:34
2.51	27	77	2016-04-06 23:58:36
2.51	27	75	2016-04-06 23:58:38

Gambar B.4 Tampilan link data tabel pada *monitoring webpage*



TEKNIK ELEKTRO UJNE

192.168.0.101/index_3.php?table=tabel_data2

Anemometer pada Ketinggian 43.2 Meter

[KECEPATAN](#) / [SUHU](#) / [ARAH TABEL](#) / [DATA RATA RATA](#) /

DATA RECORD RATA-RATA PER 10 MENIT PADA TABEL

KECEPATAN (m/s)	SUHU (C)	ARAH (degree)	DATE1	DATE2
2.539	28.437	103.345	2016-04-06 23:00:00	2016-04-06 23:09:59
1.075	25.877	98.679	2016-04-06 23:10:00	2016-04-06 23:19:59
3.311	26.828	101.231	2016-04-06 23:20:00	2016-04-06 23:29:59
0	25.4	81.4	2016-04-06 23:30:00	2016-04-06 23:39:59
1.549	25.748	87.475	2016-04-06 23:40:00	2016-04-06 23:49:59
1.873	27	79.456	2016-04-06 23:50:00	2016-04-06 23:59:59
1.972	28.486	136.565	2016-04-07 00:00:00	2016-04-07 00:09:59

TEKNIK ELEKTRO UNIVERSITAS JEMBER

Gambar B.5 Tampilan link data rata tabel pada *monitoring webpage*

LAMPIRAN C. ALAMAT IP *CLIENT*, *PROXY RESERVE*, DAN *SERVER*

```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Fadhil>ipconfig

Windows IP Configuration

Ethernet adapter Network Bridge:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::c923:8129:aa83:b5f9%16
    IPv4 Address. . . . . : 192.168.43.15
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.43.1
```

Gambar C.1 Alamat IP *client*

```
pi@raspberrypi:~$ ipconfig
File Edit Tools Help
-----
pi@raspberrypi:~$ ipconfig
Linux eth0: eth0: 192.168.0.152  Netif: 192.168.0.152  Netif: 255.255.255.0
  IP address: 192.168.0.152  Netif: 255.255.255.0  Netif: 255.255.255.0
  RX packets: 112301 errors: 0 dropped: 0 overruns: 0 carrier: 0
  collisions: 0 transmit: 0
  TX bytes: 53841216 (50.4 MiB)

Linux wlan0: wlan0: 192.168.0.1  Netif: 192.168.0.1  Netif: 255.255.255.0
  IP address: 192.168.0.1  Netif: 255.255.255.0  Netif: 255.255.255.0
  RX packets: 6079 errors: 0 dropped: 0 overruns: 0 frame: 0
  TX packets: 619 errors: 0 dropped: 0 overruns: 0 carrier: 0
  collisions: 0 transmit: 0
  RX bytes: 11029701 (10.5 MiB)  TX bytes: 11222701 (10.5 MiB)
```

Gambar C.1 Alamat IP *server raspberry pi*

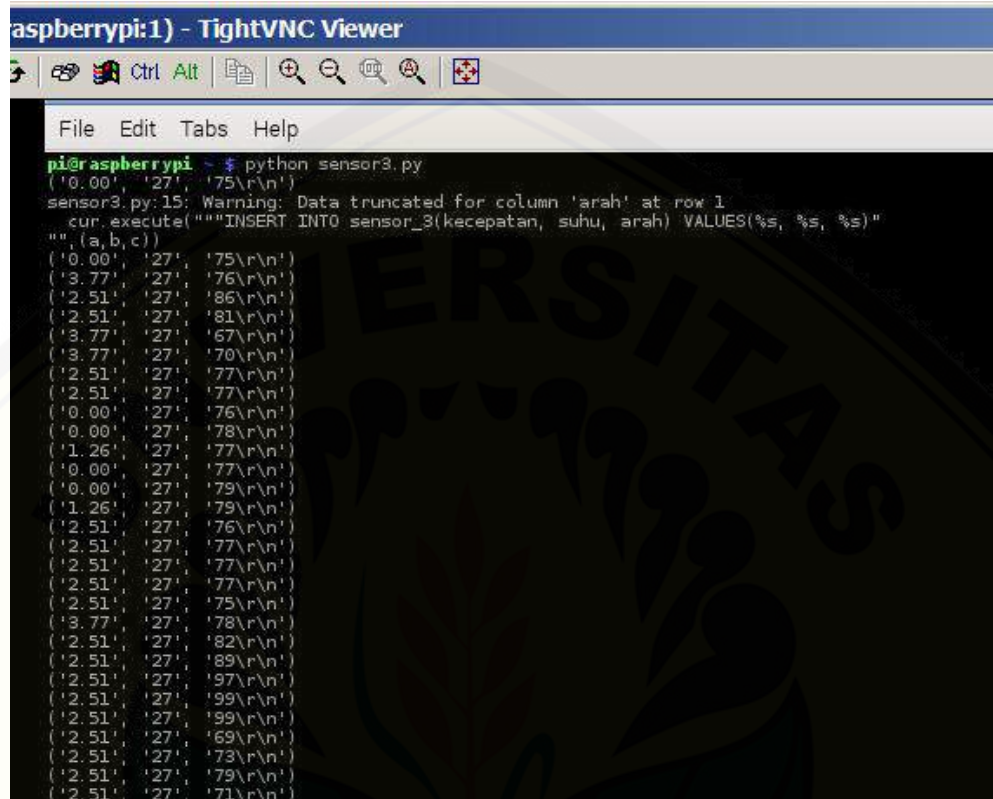
```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Fadhil>ping e5467177.ngrok.io

Pinging e5467177.ngrok.io [173.255.197.142] with 32 bytes of data:
Reply from 173.255.197.142: bytes=32 time=1176ms TTL=48
Reply from 173.255.197.142: bytes=32 time=2070ms TTL=48
Reply from 173.255.197.142: bytes=32 time=1373ms TTL=48
Reply from 173.255.197.142: bytes=32 time=282ms TTL=48

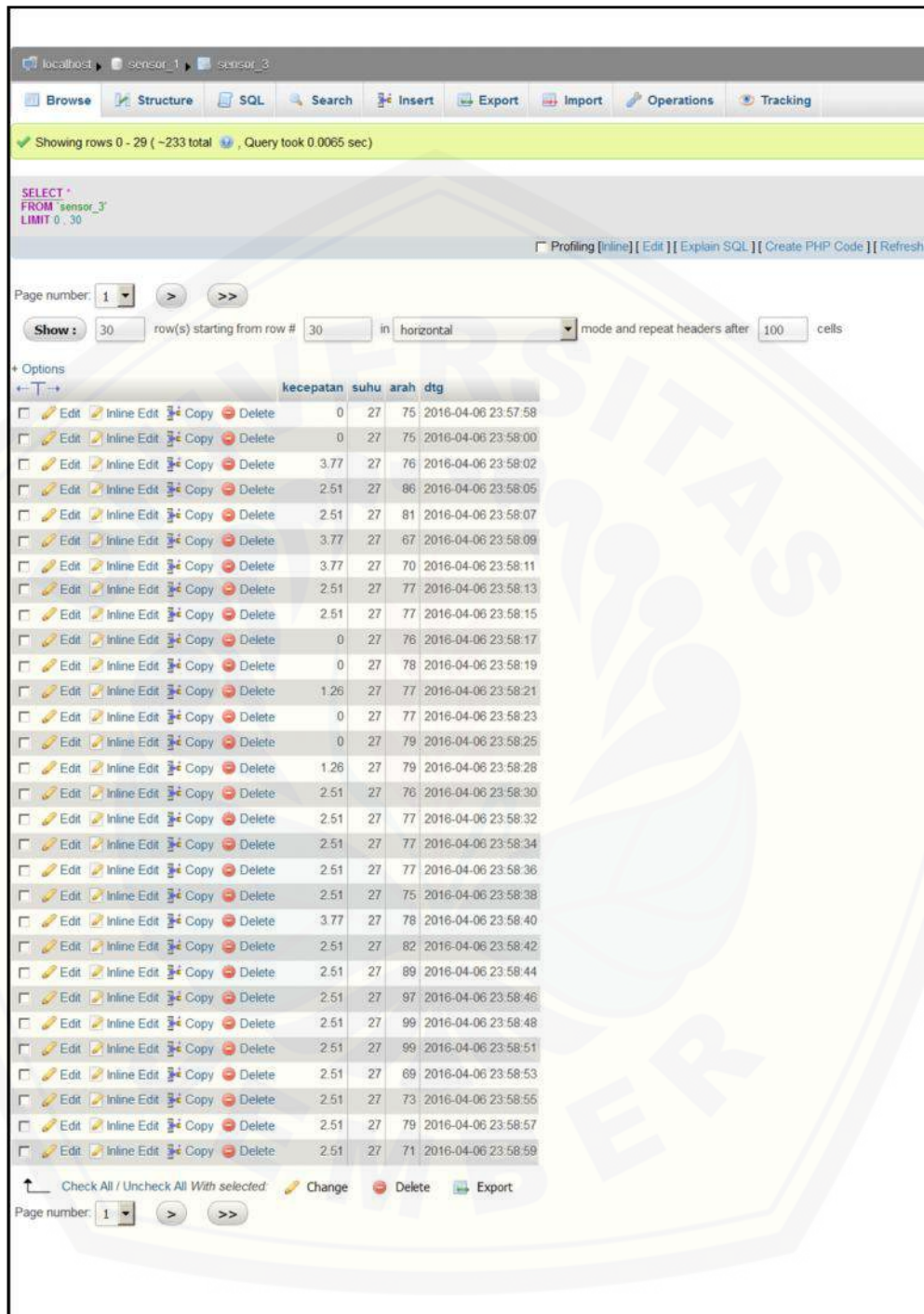
Ping statistics for 173.255.197.142:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 282ms, Maximum = 2070ms, Average = 1225ms
```

Gambar C.3 Alamat IP *proxy reserve*

LAMPIRAN D. RESPON *SERVER RASPBERRY PI*

```
raspberrypi:1) - TightVNC Viewer
File Edit Tabs Help
pi@raspberrypi ~$ python sensor3.py
('0.00', '27', '75\r\n')
sensor3.py:15: Warning: Data truncated for column 'arah' at row 1
cur.execute("""INSERT INTO sensor_3(kecepatan, suhu, arah) VALUES(%s, %s, %s)""
", (a, b, c))
('0.00', '27', '75\r\n')
('3.77', '27', '76\r\n')
('2.51', '27', '86\r\n')
('2.51', '27', '81\r\n')
('3.77', '27', '67\r\n')
('3.77', '27', '70\r\n')
('2.51', '27', '77\r\n')
('2.51', '27', '77\r\n')
('0.00', '27', '76\r\n')
('0.00', '27', '78\r\n')
('1.26', '27', '77\r\n')
('0.00', '27', '77\r\n')
('0.00', '27', '79\r\n')
('1.26', '27', '79\r\n')
('2.51', '27', '76\r\n')
('2.51', '27', '77\r\n')
('2.51', '27', '77\r\n')
('2.51', '27', '77\r\n')
('2.51', '27', '75\r\n')
('3.77', '27', '78\r\n')
('2.51', '27', '82\r\n')
('2.51', '27', '89\r\n')
('2.51', '27', '97\r\n')
('2.51', '27', '99\r\n')
('2.51', '27', '99\r\n')
('2.51', '27', '69\r\n')
('2.51', '27', '73\r\n')
('2.51', '27', '79\r\n')
('2.51', '27', '71\r\n')
```

Gambar D.1 Tampilan keberhasilan *import* data *raspberry pi* dari *arduino*



Showing rows 0 - 29 (-233 total), Query took 0.0065 sec

```
SELECT *  
FROM 'sensor_3'  
LIMIT 0, 30
```

Page number: 1

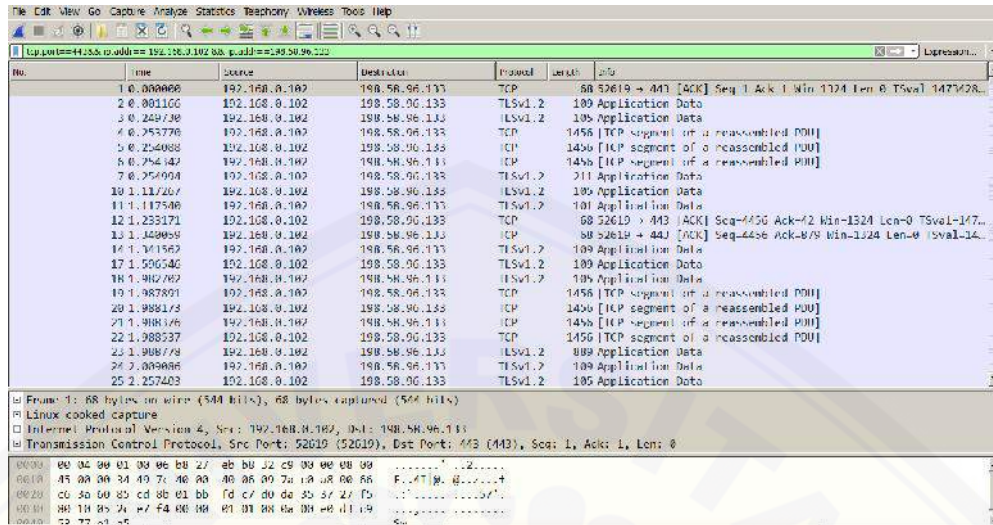
Show: 30 row(s) starting from row # 30 in horizontal mode and repeat headers after 100 cells

	kecepatan	suhu	arah	dtg
<input type="checkbox"/>	0	27	75	2016-04-06 23:57:58
<input type="checkbox"/>	0	27	75	2016-04-06 23:58:00
<input type="checkbox"/>	3.77	27	76	2016-04-06 23:58:02
<input type="checkbox"/>	2.51	27	86	2016-04-06 23:58:05
<input type="checkbox"/>	2.51	27	81	2016-04-06 23:58:07
<input type="checkbox"/>	3.77	27	67	2016-04-06 23:58:09
<input type="checkbox"/>	3.77	27	70	2016-04-06 23:58:11
<input type="checkbox"/>	2.51	27	77	2016-04-06 23:58:13
<input type="checkbox"/>	2.51	27	77	2016-04-06 23:58:15
<input type="checkbox"/>	0	27	76	2016-04-06 23:58:17
<input type="checkbox"/>	0	27	78	2016-04-06 23:58:19
<input type="checkbox"/>	1.26	27	77	2016-04-06 23:58:21
<input type="checkbox"/>	0	27	77	2016-04-06 23:58:23
<input type="checkbox"/>	0	27	79	2016-04-06 23:58:25
<input type="checkbox"/>	1.26	27	79	2016-04-06 23:58:28
<input type="checkbox"/>	2.51	27	76	2016-04-06 23:58:30
<input type="checkbox"/>	2.51	27	77	2016-04-06 23:58:32
<input type="checkbox"/>	2.51	27	77	2016-04-06 23:58:34
<input type="checkbox"/>	2.51	27	77	2016-04-06 23:58:36
<input type="checkbox"/>	2.51	27	75	2016-04-06 23:58:38
<input type="checkbox"/>	3.77	27	78	2016-04-06 23:58:40
<input type="checkbox"/>	2.51	27	82	2016-04-06 23:58:42
<input type="checkbox"/>	2.51	27	89	2016-04-06 23:58:44
<input type="checkbox"/>	2.51	27	97	2016-04-06 23:58:46
<input type="checkbox"/>	2.51	27	99	2016-04-06 23:58:48
<input type="checkbox"/>	2.51	27	99	2016-04-06 23:58:51
<input type="checkbox"/>	2.51	27	69	2016-04-06 23:58:53
<input type="checkbox"/>	2.51	27	73	2016-04-06 23:58:55
<input type="checkbox"/>	2.51	27	79	2016-04-06 23:58:57
<input type="checkbox"/>	2.51	27	71	2016-04-06 23:58:59

Check All / Uncheck All With selected Change Delete Export

Page number: 1

Gambar D.2 Tampilan data yang tersimpan didatabase mysql di raspberry pi



Gambar D.3 Capture Waktu Respon Pengiriman Data Server

Tabel D.1 Pengukuran response time pada server

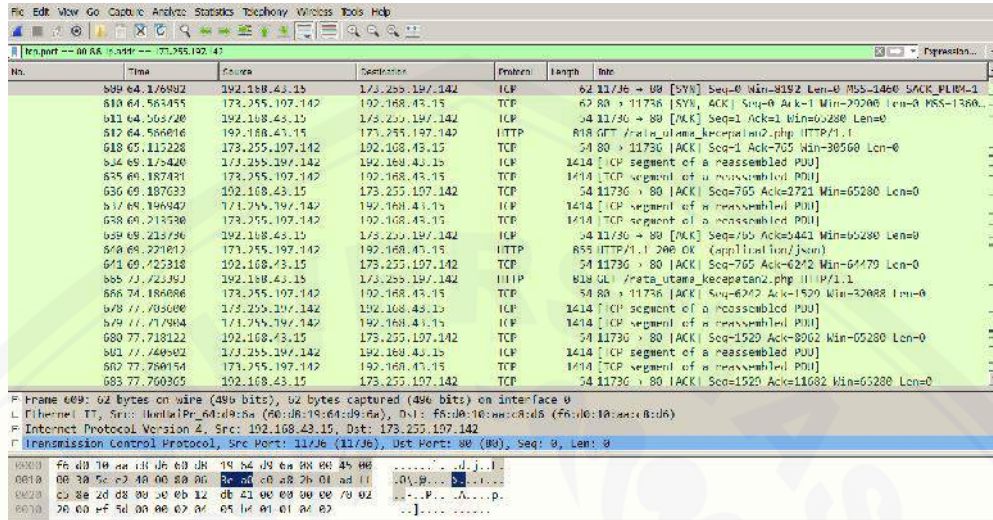
No.	Waktu Record Data (yy-mm-dd hh:mm:ss)	Waktu Pengiriman (T1)	Waktu Terkirim (T2)	Response time (T2 – T1)
1	2016-04-06 23:57:58	57:58.621	57:58.966	0.345 s
2	2016-04-06 23:58:00	58:00.561	58:00.994	0.433 s
3	2016-04-06 23:58:02	58:03.616	58:03.982	0.366 s
4	2016-04-06 23:58:05	58:06.004	58:06.420	0.416 s
5	2016-04-06 23:58:07	58:08.676	58:08.966	0.290 s
6	2016-04-06 23:58:09	58:10.708	58:11.802	0.374 s
7	2016-04-06 23:58:11	58:12.756	58:13.162	0.454 s
8	2016-04-06 23:58:13	58:14.750	58:15.126	0.376 s
9	2016-04-06 23:58:15	58:16.423	58:16.866	0.423 s
10	2016-04-06 23:58:17	58:18.342	58:18.986	0.644 s
11	2016-04-06 23:58:19	58:20.728	58:21.161	0.433 s
12	2016-04-06 23:58:21	58:22.928	58:23.325	0.397 s
13	2016-04-06 23:58:23	58:24.836	58:25.381	0.545 s
14	2016-04-06 23:58:25	58:26.736	58:27.101	0.364 s
15	2016-04-06 23:58:28	58:28.875	58:29.341	0.466 s
16	2016-04-06 23:58:30	58:30.708	58:31.241	0.533 s
17	2016-04-06 23:58:32	58:33.078	58:33.501	0.423 s
18	2016-04-06 23:58:34	58:35.656	58:36.425	0.769 s
19	2016-04-06 23:58:36	58:37.732	58:37.820	0.088 s
20	2016-04-06 23:58:38	58:38.041	58:38.323	0.282 s
21	2016-04-06 23:58:40	58:41.201	58:41.533	0.332 s
22	2016-04-06 23:58:42	58:42.708	58:42.819	0.111 s
23	2016-04-06 23:58:44	58:44.689	58:44.885	0.196 s
24	2016-04-06 23:58:46	58:46.714	58:46.825	0.111 s
25	2016-04-06 23:58:48	58:48.442	58:48.961	0.519 s

Lanjutan.

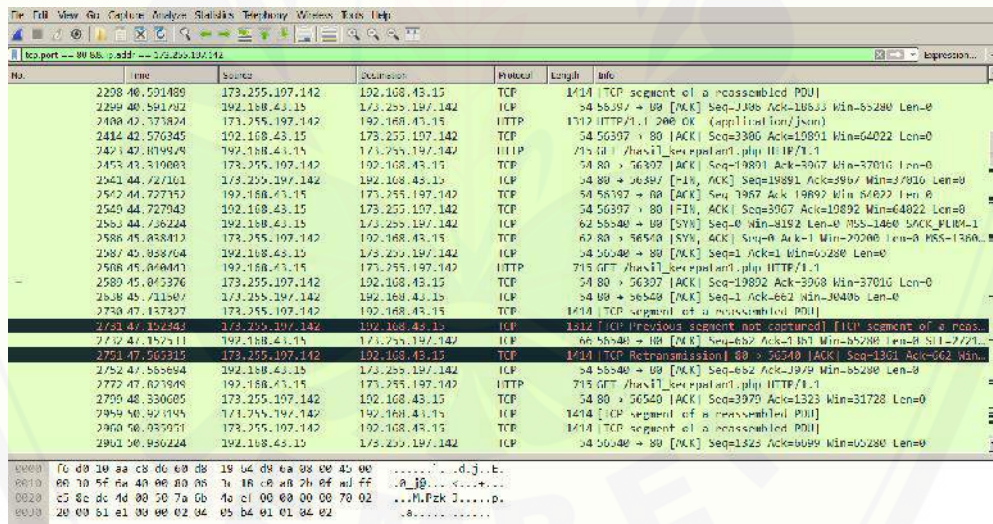
No.	Waktu <i>Record Data</i> (yy-mm-dd hh:mm:ss)	Waktu Pengiriman (T1)	Waktu Terkirim (T2)	Response time (T2 – T1)
26	2016-04-06 23:58:51	58:51.187	58:51.615	0.428 s
27	2016-04-06 23:58:53	58:53.121	58:53.621	0.500 s
28	2016-04-06 23:58:55	58:55.753	58:55.824	0.070 s
29	2016-04-06 23:58:57	58:57.748	58:58.820	1.072 s
30	2016-04-06 23:58:59	58:59.705	58:59.807	0.102 s



LAMPIRAN E. PENGUKURAN QUALITY OF SERVICE



Gambar E.1 Capture pengiriman data protokol TCP



Gambar E.2 Capture kesalahan pengiriman data protokol TCP

E.A Data pengukuran komunikasi TCP (*Transmission Control Protocol*)**1) Data jam 01.00-02.00**

No	rtt _i	Length	RTT _i	rtt _i -RTT _{i-1}	MDEV	Time out
1	0.347	62	0.043	0.347	0.086	0
2	0.0002	54	0.062	0.042	0.075	0.387
3	0.437	818	0.108	0.375	0.150	0.362
4	0.0002	54	0.094	0.107	0.139	0.708
5	0.0003	54	0.082	0.093	0.174	0.650
6	0.199	54	0.096	0.117	0.159	0.778
7	0.926	813	0.199	0.830	0.326	0.732
8	0.0002	54	0.174	0.198	0.294	1.503
9	0.202	54	0.177	0.028	0.227	1.354
10	0.579	813	0.227	0.402	0.270	1.085

2) Data jam 02.00-03.00

No	rtt _i	Length	RTT _i	rtt _i -RTT _{i-1}	MDEV	Time out
1	0.0002	54	0	0.0002	0	0
2	0.292	54	0.036	0.292	0.073	0
3	1.018	813	0.158	0.982	0.300	0.328
4	0.195	54	0.162	0.037	0.234	1.358
5	0.0001	54	0.141	0.161	0.215	1.098
6	0.0003	66	0.123	0.140	0.196	1.001
7	0.199	54	0.132	0.067	0.163	0.907
8	0.478	813	0.175	0.346	0.208	0.784
9	0.0003	54	0.153	0.174	0.199	1.007
10	0.448	818	0.189	0.295	0.223	0.949

3) Data jam 03.00-04.00

No	rtt _i	Length	RTT _i	rtt _i -RTT _{i-1}	MDEV	Time out
1	0.221	54	0.027	0.221	0.055	0
2	0.529	813	0.089	0.502	0.166	0.247
3	0.0002	54	0.077	0.088	0.146	0.753
4	0.0002	54	0.067	0.076	0.128	0.661
5	0.198	54	0.083	0.131	0.128	0.579
6	0.0002	54	0.072	0.082	0.116	0.595
7	0.198	54	0.087	0.126	0.133	0.536

No	rtt_i	Length	RTT_i	$ rtt_i-RTT_{i-1} $	MDEV	Time out
8	0.732	813	0.167	0.645	0.261	0.619
9	0.0002	54	0.146	0.166	0.237	1.211
10	0.153	813	0.146	0.007	0.179	1.094

4) Data jam 04.00-05.00

No	rtt_i	Length	RTT_i	$ rtt_i-RTT_{i-1} $	MDEV	Time out
1	0.35	813	0.043	0.35	0.087	0
2	0.0002	54	0.037	0.042	0.075	0.391
3	0.2	54	0.057	0.163	0.097	0.337
4	0.343	813	0.092	0.286	0.144	0.445
5	0.0003	54	0.080	0.091	0.130	0.668
6	0.198	54	0.094	0.118	0.127	0.600
7	0.642	818	0.107	0.548	0.232	0.602
8	0.702	813	0.181	0.595	0.322	1.035
9	0.0003	54	0.158	0.180	0.286	1.469
10	0.0002	54	0.138	0.157	0.253	1.304

5) Data jam 05.00-06.00

No	rtt_i	Length	RTT_i	$ rtt_i-RTT_{i-1} $	MDEV	Time out
1	0.315	62	0.039	0.315	0.078	0
2	0.0002	54	0.034	0.033	0.066	0.351
4	0.0002	54	0.186	0.212	0.359	1.845
5	0.198	54	0.187	0.012	0.272	1.622
6	0.574	813	0.235	0.387	0.300	1.283
7	0.315	54	0.245	0.080	0.245	1.435
8	0.205	54	0.240	0.040	0.193	1.225
9	0.481	813	0.270	0.241	0.205	1.012
10	0.0002	54	0.236	0.269	0.221	1.120

6) Data jam 06.00-07.00

No	rtt_i	Length	RTT_i	$ rtt_i-RTT_{i-1} $	MDEV	Time out
1	0.386	62	0.048	0.386	0.096	0
2	0.0002	54	0.042	0.041	0.082	0.432
3	0.549	818	0.105	0.507	0.188	0.370

No	rtt_i	Length	RTT_i	$ rtt_i - RTT_{i-1} $	MDEV	Time out
4	0.0002	54	0.091	0.104	0.167	0.857
5	0.0002	54	0.079	0.090	0.147	0.667
6	0.204	54	0.094	0.125	0.104	0.667
7	0.462	818	0.140	0.368	0.170	0.510
8	0.0002	54	0.122	0.139	0.162	0.820
9	0.0002	54	0.106	0.121	0.151	0.770
10	0.243	54	0.123	0.137	0.147	0.710

7) Data jam 07.00-08.00

No	rtt_i	Length	RTT_i	$ rtt_i - RTT_{i-1} $	MDEV	Time out
1	0.0002	54	0	0	0	0
2	0.0002	54	0	0	0	0
3	0.201	54	0.025	0.201	0.05	0
4	0.0002	54	0.021	0.024	0.043	0.225
5	0.119	813	0.034	0.099	0.057	0.193
6	0.0002	54	0.029	0.033	0.051	0.257
7	0.197	54	0.050	0.168	0.080	0.233
8	0.599	813	0.119	0.549	0.197	0.370
9	0.0001	54	0.104	0.118	0.177	0.907
10	0.206	54	0.117	0.102	0.158	0.812

8) Data jam 08.00-09.00

No	rtt_i	Length	RTT_i	$ rtt_i - RTT_{i-1} $	MDEV	Time out
1	0.0001	54	0	0	0	0
2	0.0002	54	0	0	0	0
3	0.192	54	0.024	0.192	0.048	0
4	0.205	54	0.046	0.181	0.081	0.216
5	0.352	54	0.084	0.306	0.137	0.370
6	0.351	813	0.117	0.267	0.169	0.632
7	0.0003	54	0.102	0.117	0.156	0.846
8	0.325	54	0.131	0.233	0.175	0.726
9	0.346	813	0.157	0.215	0.185	0.731
10	0.0001	54	0.137	0.157	0.131	0.897

9) Data jam 09.00-10.00

No	rtt_i	<i>Length</i>	RTT_i	$ rtt_i-RTT_{i-1} $	MDEV	<i>Time out</i>
1	0.488	62	0.061	0.488	0.122	0
2	0.0002	54	0.053	0.0608	0.106	0.549
3	0.426	715	0.099	0.373	0.173	0.477
4	0.0002	54	0.087	0.098	0.154	0.791
5	0.192	54	0.100	0.105	0.141	0.703
6	0.469	715	0.146	0.369	0.198	0.664
7	0.0002	54	0.127	0.145	0.184	0.938
8	0.638	715	0.190	0.511	0.265	0.863
9	0.0002	54	0.166	0.189	0.246	1.250
10	0.198	54	0.170	0.032	0.192	1.150

10) Data jam 10.00-11.00

No	rtt_i	<i>Length</i>	RTT_i	$ rtt_i-RTT_{i-1} $	MDEV	<i>Time out</i>
1	0.0002	54	0	0	0	0
2	0.0002	54	0	0	0	0
3	0.193	54	0.024	0.193	0.048	0
4	0.0001	54	0.021	0.023	0.041	0.216
5	0.0003	54	0.018	0.200	0.080	0.185
6	0.698	715	0.103	0.680	0.230	0.338
7	0.0002	54	0.090	0.102	0.198	1.023
8	0.0002	54	0.078	0.890	0.371	0.882
9	0.479	715	0.128	0.401	0.378	1.562
10	1.744	720	0.330	1.616	0.687	1.640

11) Data jam 11.00-12.00

No	rtt_i	<i>Length</i>	RTT_i	$ rtt_i-RTT_{i-1} $	MDEV	<i>Time out</i>
1	0.315	62	0.039	0.315	0.078	0
2	0.0002	54	0.034	0.038	0.068	0.351
3	0.563	720	0.1142	0.529	0.183	0.306
4	0.0002	54	0.099	0.114	0.165	0.846
5	0.001	66	0.086	0.098	0.148	0.759
6	0.019	54	0.077	0.067	0.127	0.678
7	0.383	720	0.115	0.306	0.171	0.585
8	0.0002	54	0.100	0.114	0.136	0.799
9	0.0002	54	0.087	0.099	0.141	0.724

No	rtt_i	Length	RTT_i	$ rtt_i-RTT_{i-1} $	MDEV	Time out
10	0.203	54	0.101	0.116	0.134	0.651

12) Data jam 12.00-13.00

No	rtt_i	Length	RTT_i	$ rtt_i-RTT_{i-1} $	MDEV	Time out
1	0.0002	54	0	0.0002	0	0
2	0.0002	54	0	0.0002	0	0
3	0.0002	54	0	0.0002	0	0
4	0.0001	54	0	0.0001	0	0
5	0.268	54	0.033	0.268	0.067	0
6	0.323	720	0.069	0.290	0.122	0.301
7	0.537	715	0.127	0.468	0.208	0.557
8	0.0002	54	0.111	0.126	0.187	0.959
9	0.011	54	0.098	0.100	0.165	0.859
10	0.0002	54	0.085	0.097	0.148	0.677

13) Data jam 13.00-14.00

No	rtt_i	Length	RTT_i	$ rtt_i-RTT_{i-1} $	MDEV	Time out
1	0.565	62	0.07	0.565	0.141	0
2	0.0002	54	0.061	0.069	0.123	0.634
3	0.413	715	0.105	0.352	0.180	0.553
4	0.0002	54	0.090	0.104	0.161	0.825
5	0.0002	54	0.078	0.089	0.143	0.734
6	0.0002	54	0.068	0.077	0.126	0.650
7	0.511	715	0.123	0.443	0.205	0.572
8	0.0002	54	0.107	0.122	0.184	0.943
9	0.0001	54	0.090	0.106	0.164	0.843
10	0.38	715	0.126	0.290	0.195	0.746

14) Data jam 14.00-15.00

No	rtt_i	Length	RTT_i	$ rtt_i-RTT_{i-1} $	MDEV	Time out
1	0.241	62	0.03	0.241	0.060	0
2	0.0002	54	0.026	0.029	0.117	0.27
3	0.297	715	0.059	0.271	0.155	0.494
4	0.0002	54	0.051	0.058	0.260	0.679

No	rtt_i	Length	RTT_i	$ rtt_i-RTT_{i-1} $	MDEV	Time out
5	0.205	54	0.070	0.154	0.233	1.091
6	0.242	715	0.091	0.172	0.217	1.002
7	0.0002	54	0.79	0.09	0.185	0.947
8	0.356	715	0.113	0.277	0.208	0.819
9	0.0002	54	0.098	0.112	0.184	0.945
10	0.162	715	0.106	0.064	0.154	0.834

15) Data jam 15.00-16.00

No	rtt_i	Length	RTT_i	$ rtt_i-RTT_{i-1} $	MDEV	Time out
1	0.0001	54	0	0.0001	0	0
2	0.0001	54	0	0.001	0	0
3	0.228	54	0.028	0.228	0.057	0
4	0.0002	54	0.024	0.027	0.049	0.256
5	0.274	54	0.055	0.250	0.990	0.220
6	0.334	720	0.089	0.279	0.144	0.451
7	0.358	715	0.122	0.296	0.175	0.665
8	0.0002	54	0.106	0.121	0.158	0.822
9	0.202	54	0.118	0.096	0.142	0.738
10	0.251	715	0.134	0.133	0.139	0.686

16) Data jam 16.00-17.00

No	rtt_i	Length	RTT_i	$ rtt_i-RTT_{i-1} $	MDEV	Time out
1	0.0001	54	0	0.0001	0	0
2	0.203	54	0.02	0.203	0.05	0
3	0.267	720	0.055	0.242	0.098	0.225
4	0.298	715	0.085	0.243	0.134	0.447
5	0.0002	54	0.074	0.084	0.121	0.621
6	0.135	715	0.081	0.061	0.106	0.558
7	0.0001	54	0.070	0.080	0.099	0.505
8	0.286	54	0.097	0.216	0.128	0.466
9	0.209	54	0.111	0.112	0.124	0.609
10	0.25	715	0.128	0.139	0.127	0.607

17) Data jam 17.00-18.00

No	rtt_i	<i>Length</i>	RTT_i	$ rtt_i - RTT_{i-1} $	MDEV	<i>Time out</i>
1	0.0001	54	0	0.0001	0	0
2	0.204	54	0.025	0.204	0.051	0
3	0.27	55	0.056	0.251	0.101	0.229
4	0.241	720	0.079	0.185	0.122	0.460
5	0.25	715	0.100	0.171	0.134	0.567
6	0.248	55	0.118	0.148	0.1375	0.636
7	0.242	55	0.133	0.124	0.134	0.660
8	0.001	54	0.116	0.132	0.126	0.652
9	0.205	54	0.127	0.089	0.116	0.620
10	0.255	715	0.143	0.128	0.119	0.591

18) Data jam 18.00-19.00

No	rtt_i	<i>Length</i>	RTT_i	$ rtt_i - RTT_{i-1} $	MDEV	<i>Time out</i>
1	0.0001	54	0	0.0001	0	0
2	0.211	54	0.026	0.211	0.052	0
3	0.254	715	0.060	0.228	0.096	0.234
4	0.0001	54	0.052	0.059	0.086	0.444
5	0.112	715	0.059	0.060	0.079	0.396
6	0.231	54	0.080	0.172	0.102	0.375
7	0.189	715	0.093	0.109	0.103	0.488
8	0.0001	54	0.081	0.920	0.100	0.505
9	0.0001	54	0.070	0.800	0.095	0.481
10	0.2	54	0.086	0.130	0.103	0.450

19) Data jam 19.00-20.00

No	rtt_i	<i>Length</i>	RTT_i	$ rtt_i - RTT_{i-1} $	MDEV	<i>Time out</i>
1	0.0002	54	0	0.0002	0	0
2	0.242	54	0.03	0.242	0.06	0
3	0.286	715	0.062	0.250	0.109	0.27
4	0.0002	54	0.054	0.061	0.097	0.498
5	0.249	54	0.078	0.195	0.121	0.442
6	0.0001	54	0.068	0.077	0.111	0.200
7	0.242	54	0.089	0.174	0.126	0.508
8	0.248	715	0.108	0.159	0.134	0.593
9	0.264	720	0.127	0.156	0.139	0.644

No	rtt_i	Length	RTT_i	$ rtt_i - RTT_{i-1} $	MDEV	Time out
10	0.0001	54	0.111	0.126	0.135	0.683

20) Data jam 20.00-21.00

No	rtt_i	Length	RTT_i	$ rtt_i - RTT_{i-1} $	MDEV	Time out
1	0.0002	54	0	0.0002	0	0
2	0.0002	54	0	0.0002	0	0
3	0.204	54	0.025	0.204	0.051	0
4	0.379	818	0.069	0.354	0.126	0.229
5	0.012	54	0.061	0.057	0.108	0.573
6	0.208	54	0.079	0.147	0.117	0.493
7	0.532	813	0.135	0.453	0.201	0.547
8	0.0002	54	0.118	0.134	0.184	0.939
9	0.395	813	0.202	0.277	0.207	0.854
10	0.0001	54	0.176	0.201	0.205	1.030

21) Data jam 21.00-22.00

No	rtt_i	Length	RTT_i	$ rtt_i - RTT_{i-1} $	MDEV	Time out
1	0.328	64	0.041	0.328	0.082	0
2	0.0003	54	0.035	0.04	0.071	0.368
3	0.617	818	0.107	0.582	0.198	0.319
4	0.0001	54	0.093	0.106	0.175	0.899
5	0.0002	54	0.081	0.092	0.142	0.793
6	0.193	54	0.095	0.112	0.134	0.663
7	0.482	818	0.143	0.387	0.197	0.631
8	0.0002	54	0.125	0.142	0.183	0.931
9	0.0002	54	0.109	0.124	0.168	0.857
10	0.194	54	0.119	0.085	0.147	0.781

22) Data jam 22.00-23.00

No	rtt_i	Length	RTT_i	$ rtt_i - RTT_{i-1} $	MDEV	Time out
1	0.299	62	0.037	0.299	0.074	0
2	0.0003	54	0.032	0.036	0.064	0.33
3	0.408	813	0.079	0.376	0.142	0.288
4	0.0002	54	0.069	0.078	0.126	0.647

No	rtt_i	Length	RTT_i	$ rtt_i - RTT_{i-1} $	MDEV	Time out
5	0.195	54	0.084	0.126	0.126	0.573
6	0.5	813	0.136	0.416	0.198	0.588
7	0.0003	54	0.119	0.135	0.182	0.928
8	0.298	62	0.141	0.179	0.181	0.847
9	0.0002	54	0.123	0.140	0.170	0.865
10	0.41	813	0.158	0.287	0.199	0.803

23) Data jam 23.00-24.00

No	rtt_i	Length	RTT_i	$ rtt_i - RTT_{i-1} $	MDEV	Time out
1	0.0002	54	0	0.0002	0	0
2	0.397	54	0	0.0002	0	0
3	0.0001	637	0.049	0.397	0.099	0
4	0.258	54	0.042	0.048	0.086	0.445
5	0.41	54	0.069	0.216	0.118	0.386
6	0.0002	637	0.111	0.341	0.173	0.541
7	267	54	0.097	0.110	0.157	0.803
8	0	54	0.118	0.170	0.160	0.725
9	0.634	637	0.182	0.516	0.249	0.758
10	0.0001	54	0.159	0.181	0.232	1.178

24) Data jam 24.00-00.00

No	rtt_i	Length	RTT_i	$ rtt_i - RTT_{i-1} $	MDEV	Time out
1	0.0002	54	0	0.0002	0	0
2	0.0002	54	0	0.0002	0	0
3	0.194	54	0.024	0.194	0.048	0
4	0.378	642	0.068	0.354	0.124	0.216
5	0.0002	54	0.059	0.067	0.109	0.564
6	0.0002	54	0.051	0.058	0.096	0.495
7	0.199	54	0.069	0.148	0.109	0.435
8	0.431	637	0.110	0.362	0.172	0.505
9	0.0002	54	0.096	0.110	0.156	0.798
10	0.204	54	0.109	0.108	0.144	0.720

E.B Data Pengukuran *throughput* TCP dan Request Time Out (RTO)Tabel E.1 Pengukuran *throughput* TCP

Kategori waktu	Jam	TCP RWND (KBps)	RTT (s)	Throughput (KBps)
Pagi	01.00-02.00	46.848	0.126	371.809
	02.00-03.00	46.848	0.127	368.881
	03.00-04.00	46.848	0.096	488.000
	04.00-05.00	46.848	0.098	478.040
	05.00-06.00	46.848	0.188	249.191
	06.00-07.00	46.848	0.095	493.136
	07.00-08.00	46.848	0.049	956.081
	08.00-09.00	46.848	0.079	593.012
Siang	09.00-10.00	46.848	0.119	393.680
	10.00-11.00	46.848	0.079	593.012
	11.00-12.00	46.848	0.085	551.152
	12.00-13.00	46.848	0.052	900.923
	13.00-14.00	46.848	0.091	514.813
	14.00-15.00	46.848	0.143	327.608
	15.00-16.00	46.848	0.067	699.223
	16.00-17.00	46.848	0.072	650.666
Malam	17.00-18.00	46.848	0.089	526.382
	18.00-19.00	46.848	0.060	780.800
	19.00-20.00	46.848	0.072	650.666
	20.00-21.00	46.848	0.086	544.744
	21.00-22.00	46.848	0.094	498.382
	22.00-23.00	46.848	0.097	482.969
	23.00-24.00	46.848	0.082	571.317
	24.00-01.00	46.848	0.058	807.724

Tabel E.2 Pengukuran persentase *Request Time Out* (RTO)

Waktu	Segmen data dikirim	<i>retransmission time out</i>	RTO (%)
Pagi	160	2	1.25
Siang	160	7	4.37
Malam	160	6	3.75

LAMPIRAN F. DOKUMENTASI PENELITIAN



Gambar F.1 Tempat pemasangan alat monitoring sebagai objek penelitian



Gambar F.2 Pengambilan data



Gambar F.3 *Maintenance* dan perbaikan alat