

**RANCANG BANGUN *MAIL SERVER* DALAM JARINGAN LOKAL  
MENGUNAKAN *EMBEDDED SYSTEM***

**SKRIPSI**

**Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer**



**Disusun oleh :**

**MOH. HALIMI**

**NIM. 0910680041**

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN**

**UNIVERSITAS BRAWIJAYA**

**PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER**

**MALANG**

**2013**

**LEMBAR PERSETUJUAN**

**RANCANG BANGUN *MAIL SERVER* DALAM JARINGAN LOKAL  
MENGUNAKAN *EMBEDDED SYSTEM***

**SKRIPSI**

**Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer**

**Disusun oleh :**

**MOH. HALIMI**

**NIM. 0910680041**

Skripsi ini telah disetujui dosen pembimbing pada  
tanggal 13 Desember 2013

Pembimbing I

Pembimbing II

**Barlian Henryranu Prasetyo, S.T., M.T.**  
**NIK. 821024 06 1 1 0254**

**Gembong Edhi Setyawan, S.T., M.T.**  
**NIK. 761201 16 1 1 0373**

**LEMBAR PENGESAHAN**

**RANCANG BANGUN *MAIL SERVER* DALAM JARINGAN LOKAL  
MENGUNAKAN EMBEDDED SYSTEM**

**SKRIPSI**

**Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer**

**Disusun oleh :**

**MOH. HALIMI**

**NIM. 0910680041**

Skripsi ini diuji dan dinyatakan lulus pada

Tanggal 7 Januari 2014

Penguji I

Penguji II

**Eko Setiawan, S.T., M.Eng.**  
**NIK. 870610 06 1 1 0256**

**Wijaya Kurniawan, S.T., M.T**  
**NIK. 820 125 16 1 1 0418**

Penguji III

**Sabriansyah Rizqika Akbar, S.T., M.Eng.**  
**NIK. 820 809 06 1 1 0084**

Mengetahui,

Ketua Program Studi Informatika/Ilmu Komputer

**Drs. Marji., M.T.**

**NIP. 19670801 199203 1 001**

**PERNYATAAN  
ORISINALITAS SKRIPSI**

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

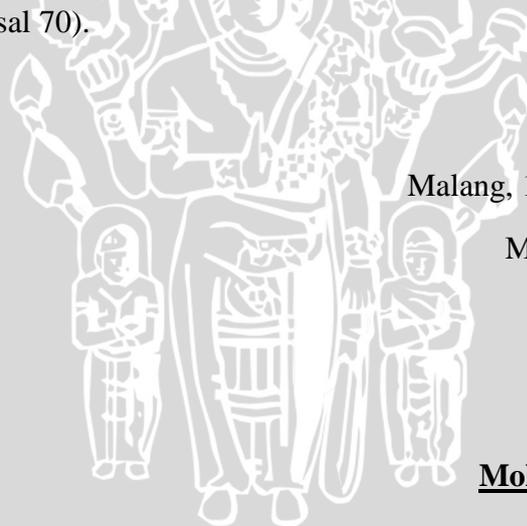
Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 12 November 2013

Mahasiswa,

**Moh, Halimi**

**NIM. 0910680041**



## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, karena berkat rahmat dan hidayah-Nya, penulis dapat menyelesaikan penyusunan skripsi ini. Tak lupa penulis haturkan shalawat serta salam kepada Rasulullah Nabi Muhammad SAW, keluarga, sahabat-sahabat, dan para pengikutnya, semoga rahmat Allah SWT selalu tercurah untuk kita semua. Aamiin.

Skripsi ini disusun dengan judul “**RANCANG BANGUN MAIL SERVER DALAM JARINGAN LOKAL MENGGUNAKAN EMBEDDED SYSTEM**”.

Penulisan skripsi ini disusun untuk memenuhi syarat menjadi Sarjana Komputer. Dalam penyusunan skripsi ini telah banyak mendapat bantuan dari berbagai pihak. Ucapan terima kasih penulis sampaikan kepada:

1. Bapak Sholichin, Ibu Nurul Chusniyah, Anisah, Habibah dan M. Zamroni selaku Ayah, Bunda dan Kakak – Kakak tercinta atas segenap do’a, dukungan dan kasih sayang yang telah diberikan.
2. Bapak Barlian Henryranu P. S.T., M.T. dan Bapak Gembong Edhi Setyawan S.T., M.T. selaku dosen pembimbing yang telah membimbing penulis dalam pengerjaan dan penulisan skripsi ini.
3. Bapak Ir. Sutrisno, M.T. selaku Ketua Program Teknologi Informasi dan Ilmu Komputer dan selaku dosen penasihat akademik penulis.
4. Bapak Drs. Mardji, M.T. selaku Ketua Program Studi Informatika/Ilmu Komputer Program Teknologi Informasi dan Ilmu Komputer.
5. Segenap dosen Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya atas segenap ilmu pengetahuan dan perhatian yang diberikan.

6. Segenap staff dan pegawai Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya atas segala bantuan yang bersifat administratif.
7. Ardhan Dharma Yudha Handoyo S.Kom, Antarangga Bhawika Manggala, Cantika nur Previana S.Kom, Delis Sukma, Dwinta Ayu Pramita, Gerry Wisudawan, M. Rizki Gus Sofwan, Rizky Kurnia Aziz Mustofa atas bantuan dan persahabatan yang tak ternilai selama penulis menjalani kuliah.
8. Adin Faisol Habib, Bintang Karunia Akbar, Yoga Pradana Pamungkas, Putra Fajar Romadhoni, Om Alizar Erwinsyah Uber, dan Lik Yahya Ansori serta segenap Keluarga Besar Parkiran atas hiburan yang membantu penulis untuk melepas penat semasa kuliah.
9. I Putu Arie Bayu Antara atas fasilitas dan ijinnya yang membolehkan penulis bertempat tinggal di kediamannya.
10. Anita Pratiwi, Galuh Prasetyanita Wulandari, Henidar Sekarningtiyas Putri, Yuriska Lintang Kesuma yang telah memberikan dukungan dan bantuan selama penulis mengerjakan skripsi.
11. Seluruh teman-teman TIF B UB 09 tercinta atas dukungan dan kebersamaannya dari awal perkuliahan sampai akhir.
12. Teman-teman Program Studi Informatika/Ilmu Komputer angkatan 2009 tercinta yang selalu memberikan semangat, dorongan, dan bantuan pikiran.
13. Serta semua pihak yang telah membantu dan memberikan pengalaman berharga bagi penulis selama penulis menjalani masa perkuliahan.

Akhirnya atas segala bantuan semua pihak semoga mendapat balasan yang setimpal dari Allah SWT. Harapan penulis semoga skripsi ini dapat memberikan manfaat pada semua pihak, terutama kepada penulis dan para pengembang yang nantinya akan mengembangkan penelitian dari penulis.

Malang, 12 Desember 2013

Penulis

## ABSTRAK

**Mohammad Halimi. 2013. : Rancang Bangun *Mail Server* dalam Jaringan Lokal Menggunakan *Embedded system*. Penelitian Program Studi Informatika/Illmu Komputer, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.**

**Dosen Pembimbing: Barlian Henryranu Prasetyo, ST, M.T. dan Gembong Edhi Setyawan ST., M.T**

Metode pengiriman surat maupun dokumen secara elektronik sudah menjadi kebutuhan bagi setiap orang guna mengikuti perkembangan teknologi. *Mail Server* merupakan *server* yang melayani layanan pengiriman dan pengambilan *e-mail*. *Mail server* merupakan salah satu fungsi *server* yang paling banyak digunakan oleh perusahaan maupun instansi. Hal ini mengingat fungsi email sendiri yang bisa mengurangi biaya surat-menyurat, lebih efisien dibandingkan komunikasi manual dan dapat menyertakan *attachment* yang berguna sebagai pelengkap dan dokumen tambahan terkait dengan isi email. Guna meringankan beban kerja dari *mail server* maka dilakukan penelitian mengenai implementasi dan analisa performansi *mail server* menggunakan *embedded system*. Mini komputer yang digunakan adalah Raspberry Pi model B dan sebagai tampilan antar muka pengguna *mail server* menggunakan perangkat lunak Citadel. Hasil yang didapatkan adalah *mail server* mampu melayani komunikasi email hingga besar data keseluruhan 20MB dengan nilai *transfer rate* rata – rata 505.7 KBps. Dalam satu kali pengiriman email menggunakan *mail server embedded system*, didapatkan nilai integritas data 100% untuk setiap parameter pengujian. *Mail server* ini nantinya diharapkan dapat melayani permintaan dari pengguna atau *user* yang hanya memerlukan komunikasi email pada jaringan lokal sehingga tidak perlu mengakses *mail server master* yang ada.

**Kata kunci :** *Mail server, Embedded system, Performansi, Raspberry Pi, Citadel*

## ABSTRACT

**Mohammad Halimi. 2013. : Design and Implementation of Mail Server in Local Area Network Using Embedded System. Penelitian Program Studi Informatika/Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.**

**Dosen Pembimbing: Barlian Henryranu Prasetyo, ST, M.T. dan Gembong Edhi Setyawan ST., M.T**

Method of mail delivery and electronic documents has become a necessity for everyone to keep up with technology and a server that serves service delivery and retrieval of e-mail. Mail server is a server that most widely used in the company. It is given its own email function that can reduce the cost of correspondence, communication and more efficient than the manual and it's can include attachments that useful as a complementary and additional documents related to the content of the email. Raspberry Pi type B was used for mail server implementation and for user interface of the mail server used Citadel application. This research has two results. First, mail server was capable of serving an email communication until amount of data is 20 MB with average transfer rate value is 505.7 KBps. Second, In once delivery email process using mail server embedded system obtained 100% value of data integrity for each parameter testing. The implementation of the mail server using the embedded system has purpose to ease the workload of mail server's master. The mail server will serve requests from the user who only require an email communication on a local network so it does not need to access the mail server's master.

**Keyword :** Mail server, Embedded system, Performance, Raspberry Pi, Citadel

DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>i</b>
<b>ABSTRAK .....</b>	<b>iii</b>
<b>ABSTRACT.....</b>	<b>iv</b>
<b>DAFTAR ISI.....</b>	<b>v</b>
<b>DAFTAR GAMBAR.....</b>	<b>vii</b>
<b>DAFTAR TABEL .....</b>	<b>viii</b>
<b>LAMPIRAN - LAMPIRAN .....</b>	<b>ix</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Sistematika Penulisan.....	3
<b>BAB II KAJIAN PUSTAKA DAN DASAR TEORI .....</b>	<b>6</b>
2.1 Kajian Pustaka.....	6
2.2 Dasar Teori.....	7
2.2.1 Email.....	7
2.2.2 Mail Server.....	8
2.2.3 Citadel.....	9
2.2.4 Embedded System.....	11
<b>BAB III METODOLOGI PENELITIAN DAN PERANCANGAN .....</b>	<b>13</b>
3.1 Metode Penelitian.....	13

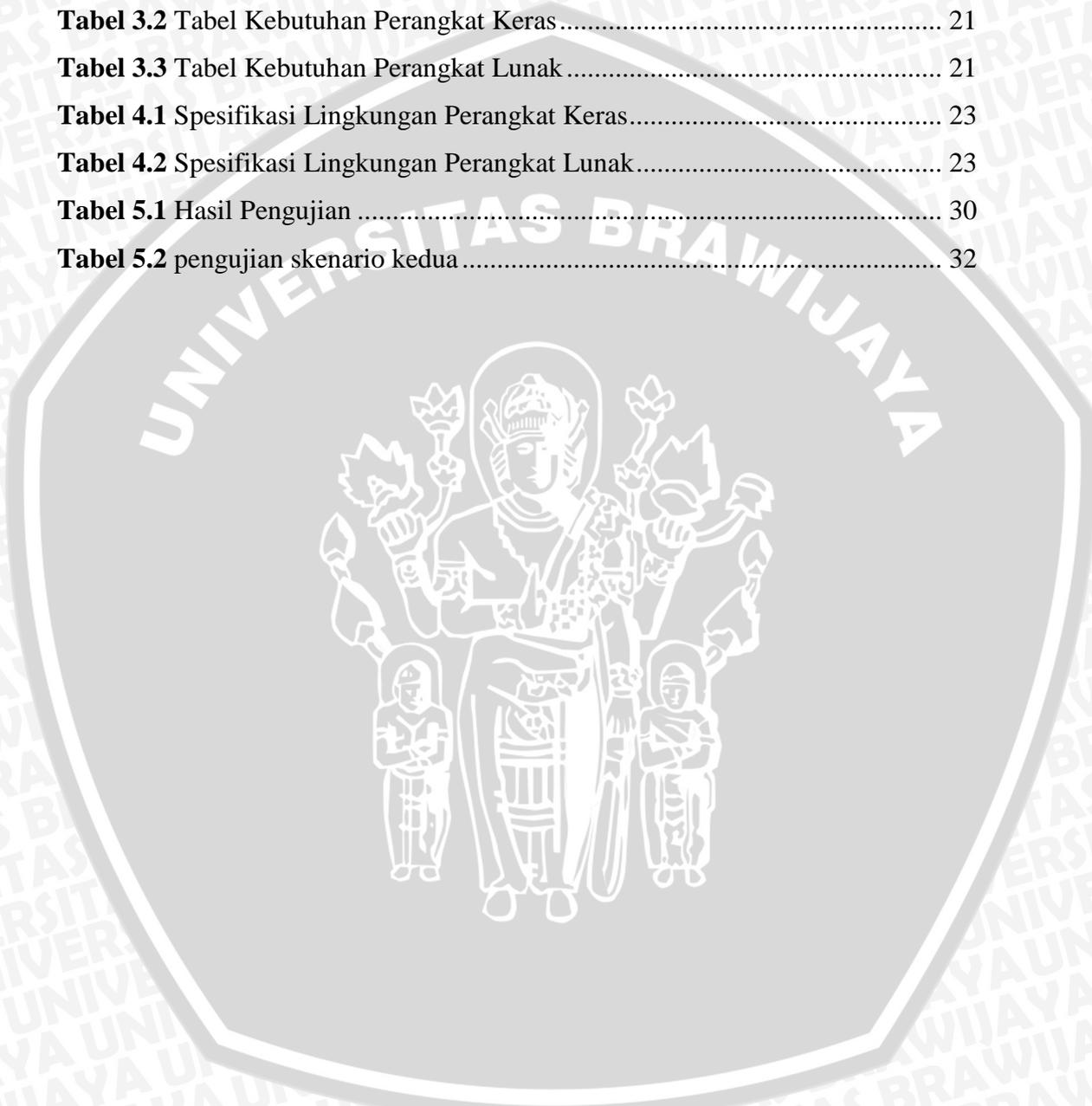
3.1.1	Analisa Kebutuhan.....	14
3.1.2	Spesifikasi Infrastruktur.....	15
3.1.3	Perancangan Sistem.....	19
3.1.4	Implementasi.....	19
3.1.5	Pengujian dan Analisis.....	19
3.1.6	<i>Maintenance</i> .....	20
3.2	Perancangan.....	20
3.2.1	Analisa Kebutuhan Perangkat Lunak dan Perangkat Keras.....	20
3.2.2	Perancangan Perangkat Lunak dan Perangkat Keras.....	21
3.2.3	Topologi Perancangan.....	21
<b>BAB IV</b>	<b>IMPLEMENTASI.....</b>	<b>23</b>
4.1	Spesifikasi Lingkungan Infrastruktur.....	23
4.2	Implementasi Raspberry Pi.....	23
4.3	Implementasi Citadel.....	24
4.4	Implementasi infrastruktur.....	25
<b>BAB V</b>	<b>PENGUJIAN DAN ANALISIS.....</b>	<b>27</b>
5.1	Pengujian.....	27
5.1.1	Skenario Pengujian Pertama.....	27
5.1.2	Skenario Pengujian Kedua.....	31
5.2	Analisis.....	32
<b>BAB VI</b>	<b>PENUTUP.....</b>	<b>36</b>
6.1	Kesimpulan.....	36
6.2	Saran.....	36
<b>DAFTAR PUSTAKA</b>	<b>.....</b>	<b>38</b>
<b>LAMPIRAN - LAMPIRAN</b>	<b>.....</b>	<b>40</b>

## DAFTAR GAMBAR

<b>Gambar 2.1</b> Arsitektur mail server .....	9
<b>Gambar 3.1</b> Diagram Alir Metode Penelitian .....	13
<b>Gambar 3.2</b> Raspberry Pi .....	16
<b>Gambar 3.3</b> Blok Diagram Raspberry Pi.....	17
<b>Gambar 3.4</b> Topologi Perancangan .....	22
<b>Gambar 4.1</b> Tampilan Raspbian Wheezy .....	24
<b>Gambar 4.2</b> Tamplinan UI Citadel .....	25
<b>Gambar 4.3</b> Skema implementasi .....	26
<b>Gambar 5.1</b> Skenario Pengujian pertama .....	28
<b>Gambar 5.2</b> Pengamatan menggunakan aplikasi Wireshark .....	29
<b>Gambar 5.3</b> <i>Screenshot</i> Pengujian.....	30
<b>Gambar 5.4</b> Skenario Pengujian kedua .....	31
<b>Gambar 5.5</b> Grafik Hasil Pengujian Skenario 1 .....	33
<b>Gambar 5.6</b> Grafik Waktu Pengiriman.....	34
<b>Gambar 5.7</b> Grafik hasil pengujian skenario 2 .....	34

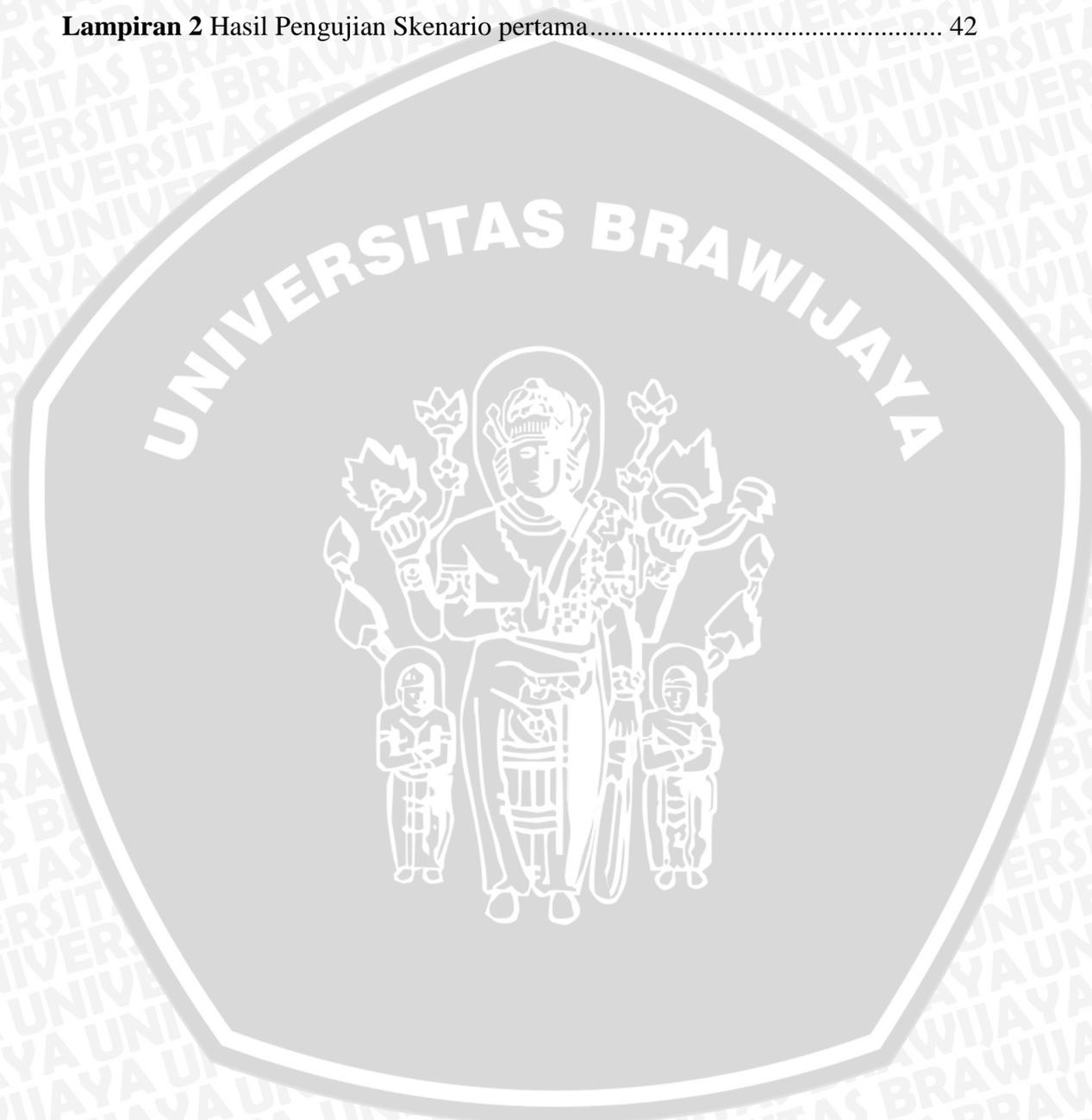
**DAFTAR TABEL**

<b>Tabel 3.1</b> Spesifikasi raspberry pi .....	18
<b>Tabel 3.2</b> Tabel Kebutuhan Perangkat Keras .....	21
<b>Tabel 3.3</b> Tabel Kebutuhan Perangkat Lunak .....	21
<b>Tabel 4.1</b> Spesifikasi Lingkungan Perangkat Keras .....	23
<b>Tabel 4.2</b> Spesifikasi Lingkungan Perangkat Lunak .....	23
<b>Tabel 5.1</b> Hasil Pengujian .....	30
<b>Tabel 5.2</b> pengujian skenario kedua .....	32



## LAMPIRAN - LAMPIRAN

<b>Lampiran 1</b> Konfigurasi <i>file</i> webcit.css .....	40
<b>Lampiran 2</b> Hasil Pengujian Skenario pertama.....	42



## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Seiring dengan kemajuan teknologi, *email* merupakan salah satu media komunikasi sebagai pengganti surat. Menurut Narendro Arifia [NAE-13], dalam tulisannya menyebutkan *electronic mail* (surat elektronik) sering disebut e-mail atau email, merupakan metode *store and forward* dari menulis, mengirim, menerima dan menyimpan surat melalui sebuah sistem komunikasi elektronik. Surat dipertukarkan antar host melalui *Simple Mail Transfer Protocol* dengan software program : *Mail Transport Agents* (MTA). *User* dapat men-download surat mereka dari server dengan protokol standard seperti POP atau IMAP, atau di lingkungan perusahaan besar dengan protokol khusus seperti *Lotus Notes* atau *Microsoft Exchange Servers*. Surat dapat disimpan pada *client*, pada *server*, atau keduanya. Sebagai penunjang untuk melakukan komunikasi melalui *email* dibutuhkan sebuah *server* yang khusus untuk melayani proses pengiriman surat elektronik yang disebut dengan *mail server*. Dalam salah satu tulisannya I. E. S. W. Mangunkusumo [MIE-13], menyatakan *Mail server* merupakan salah satu fungsi *server* yang paling banyak digunakan di perusahaan. Hal ini mengingat fungsi *email* sendiri yang bisa mengurangi biaya surat-menyurat, lebih efisien dibandingkan komunikasi manual dan dapat menyertakan *attachment* yang berguna sebagai pelengkap dan dokumen tambahan terkait dengan isi email. Tuntutan teknologi juga yang menyebabkan setiap perusahaan maupun universitas harus mempunyai *mail server* sendiri sehingga menjadi lebih efisien.

Untuk meminimalkan penggunaan sumber daya saat ini terdapat beberapa sistem yang mampu menjalankan tugas yang bersifat khusus atau spesifik sesuai dengan kebutuhan yang diharapkan. Salah satu sistem yang memiliki fungsi tersebut adalah *embedded system*. Menurut Steve Heath [HSE-03], Sebuah *embedded system* adalah sistem berbasis mikroprosesor yang dibangun untuk

mengontrol fungsi atau jangkauan fungsi dan tidak dirancang untuk diprogram oleh pengguna akhir dengan cara yang sama pada penggunaan PC .

Pada kesempatan ini penulis akan membangun serta menganalisa sistem *mail server* berbasis *embedded system* dengan menggunakan *raspberry pi*. Perangkat *raspberry pi* digunakan karena memiliki beberapa keunggulan yakni hemat biaya, hemat energy dan *low-noise* karena memiliki ukuran yang kecil [RPT-13]. Penelitian ini bertujuan untuk menghasilkan sebuah *mail server* untuk komunikasi surat elektronik guna membantu performansi sebuah *mail server master*. Dengan adanya *mail server* berbasis *embedded system* ini, *mail server master* tidak perlu menangani komunikasi *email* antar pengguna yang berada pada sebuah jaringan lokal. Hal ini diharapkan dapat mengurangi beban kerja *mail server master*. Parameter yang digunakan untuk menganalisa performansi *mail server* adalah nilai *transfer rate* dan besar kapasitas data yang bisa dikirimkan melalui *mail server* tersebut.

Berdasarkan tujuan yang telah diuraikan pada paragraf diatas, penulis berharap penelitian ini dapat diimplementasikan secara langsung dilapangan. Guna menunjang proses tersebut maka perlu dilakukan proses sinkronisasi antara *mail server embedded system* dan *mail server master* yang ada. Proses tersebut menggunakan metode replikasi dan sinkronisasi *database LDAP*. Data dari pengguna yang ada pada *mail server master* diduplikasi atau disalin pada server *LDAP* yang ada pada *mail server embedded system*.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang dipaparkan, maka rumusan masalah yang perlu diperhatikan adalah :

1. Bagaimana perancangan dan implementasi *mail server* pada *Embedded System*.
2. Pengujian ditekankan pada *transfer rate*, besar data dan integritas data yang dapat dikirimkan menggunakan *mail server* pada *embedded system*.

### 1.3 Batasan Masalah

Batasan masalah dapat diklasifikasikan sebagai berikut :

1. Pembahasan difokuskan mengenai analisa performansi *mail server* menggunakan *embedded system* yang berkaitan dengan kecepatan pengiriman data, besar data dan integritas data yang mampu dikirimkan.
2. Perancangan *mail server* menggunakan perangkat lunak Citadel.
3. Perangkat keras yang digunakan dalam penelitian ini adalah mini komputer *raspberry pi* model B.
4. Jumlah pengguna yang terdaftar pada *mail server* adalah 50 pengguna.

### 1.4 Tujuan

Tujuan penulisan tugas akhir ini adalah untuk merancang dan menganalisa *mail server* pada *embedded system* menggunakan *raspberry pi* guna membantu performansi *mail server master*.

### 1.5 Manfaat

Penelitian mengenai *mail server* pada *embedded system* mempunyai beberapa manfaat yakni mengurangi beban kerja dari *mail server master* dan hanya memerlukan konsumsi daya yang sedikit. Hasil penelitian ini diharapkan dapat dijadikan sebagai bahan referensi dan tambahan pengetahuan mengenai *mail server* menggunakan *embedded system Raspberry Pi*.

### 1.6 Sistematika Penulisan

Sistematika penulisan dalam skripsi ini adalah sebagai berikut :

#### BAB I

#### Pendahuluan

Memuat latar belakang permasalahan, identifikasi dan pembatasan masalah, rumusan masalah, tujuan dan manfaat, dan sistematika penulisan.

#### BAB II

#### Kajian Pustaka dan Dasar teori

Menguraikan kajian pustaka yang digunakan pada proses penelitian. Serta menguraikan teori dasar dan teori penunjang yang berkaitan dengan *mail server*.

**BAB III****Metode Penelitian dan Perancangan**

Membahas metode yang digunakan dalam penelitian yang terdiri dari analisa kebutuhan, spesifikasi infrastruktur, perancangan sistem, implementasi, pengujian dan analisis, serta *maintenance*.

**BAB IV****Implementasi**

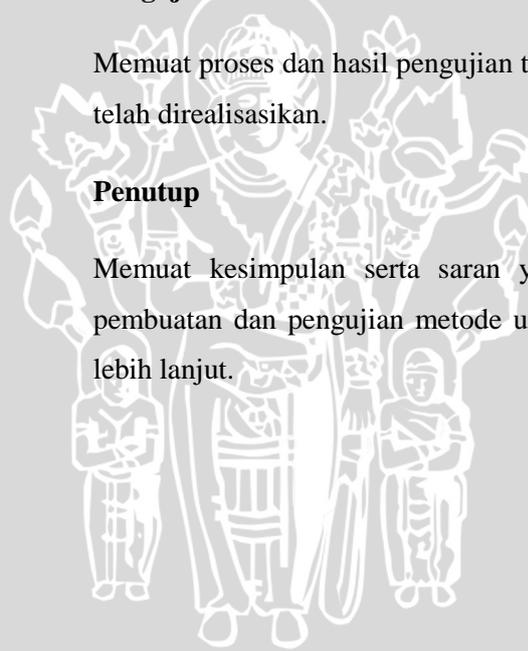
Membahas tentang implementasi dari *system mail server* pada *embedded system raspberry pi*.

**BAB V****Pengujian dan Analisis**

Memuat proses dan hasil pengujian terhadap sistem yang telah direalisasikan.

**BAB VI****Penutup**

Memuat kesimpulan serta saran yang diperoleh dari pembuatan dan pengujian metode untuk pengembangan lebih lanjut.





## BAB II

### KAJIAN PUSTAKA DAN DASAR TEORI

#### 2.1 Kajian Pustaka

Salah satu penelitian yang membahas mengenai implementasi perangkat citadel sebagai *mail server* dilakukan oleh Fahmi Rajab Febrian [RFF-11], berjudul “Building a Mail Server Using Mail Citadel Transfer Agent on Linux Ubuntu 10:10 and accessing With Mozilla Thunderbird”. Penelitian ini membahas tentang implementasi Citadel sebagai *mail server* pada sistem operasi Linux dan menggunakan aplikasi Mozilla Thunderbird pada sisi pengguna. Hasil dari penelitian ini adalah sebuah layanan komunikasi *email* pada sebuah perusahaan, dimana hanya pihak admin yang berwenang untuk menentukan pengguna yang berhak menggunakan layanan ini.

Penelitian lain membahas mengenai performansi dari *mail server* dilakukan oleh Jun Wan dan Yiming Hu [WJY-04], berjudul “A Performance Study on Internet Server Provider Mail Servers”. Penelitian ini membahas tentang performansi dari sebuah *mail server*. Masalah yang dibahas pada penelitian ini dibagi pada tiga hal, pertama adalah mengamati *session* dari protokol SMTP dan POP mengenai respon *server* pada awal koneksi ketika mengakses sebuah *mail server*. Kedua adalah mengenai pengaruh dari perangkat I/O atau *Input/ output* terhadap total waktu transfer ketika proses permintaan *email* terutama pada iserveri dengan pengguna yang besar. Ketiga adalah hal yang dapat membuat sebuah *server* menjadi kelebihan beban. Penelitian ini menghasilkan beberapa kesimpulan yakni pada *session* protokol SMTP dan POP ketika proses koneksi awal menuju *mail server* membutuhkan waktu yang lama. Kesimpulan selanjutnya adalah *latency* dari perangkat I/O memiliki kontribusi sebesar 40-55% terhadap proses transfer data pada proses permintaan *email*. Terakhir adalah sebuah kelompok *email* dapat dengan mudah membuat *server* penerima jarak jauh menjadi kelebihan beban.

## 2.2 Dasar Teori

### 2.2.1 Email

*Electronic mail* atau *email* adalah metode pengiriman surat melalui jaringan internet. Sistematis penggunaan *email* tidak jauh berbeda dengan surat konvensional. Surat terdiri dari nama pengirim, alamat yang dituju dan isi surat. Perbedaan menggunakan *email* terletak pada media pengiriman yang digunakan yakni menggunakan jaringan internet [RRA-06].

Sebelum pengguna dapat memanfaatkan *email*, terlebih dahulu seorang pengguna harus mendaftarkan dirinya pada sebuah *server* atau *administrator*. Setelah mendaftarkan dirinya pengguna akan mendapatkan *account* yang digunakan untuk masuk pada sebuah layanan *email* dimana pengguna dapat menulis, menerima dan mengirimkan *email* [RRA-06]. Kelebihan yang dimiliki oleh *email* adalah menghemat biaya karena *email* tidak perlu menggunakan kertas sebagai media penulisannya. *Email* hanya memerlukan sebuah komputer yang tersambung dalam jaringan internet.

Seorang pengguna tidak hanya bisa mengirimkan sebuah surat melalui *email*, pengguna juga dapat mengirimkan *file* berupa gambar atau dokumen sesuai dengan kebutuhan pengguna. *File* yang akan dikirimkan dapat dilampirkan pada *email* yang akan dikirimkan dan akan dikirim bersamaan dengan pengiriman *email*. Seorang pengguna dapat menggunakan *email* selama 24 jam dalam sehari dan 7 hari dalam seminggu.

*Email* dianggap kurang formal jika dibandingkan dengan surat konvensional bagi sebagian orang, meskipun pengaturan *email* sudah berkembang dalam beberapa tahun terakhir. Tidak ada pedoman khusus dalam penulisan *email* dan juga tidak ada panjang maksimum yang dianjurkan dalam penulisan *email*. Pengguna juga harus mempertimbangkan besar *file* yang akan dilampirkan pada *email* yang dikirim. *File* yang dilampirkan dapat berasal dari komputer dimana pengguna mengirimkan *email* atau berasal dari sumber yang ada di internet [TUY-10].

### 2.2.2 Mail Server

Pada awalnya *mail server* dirancang sebagai alat komunikasi untuk mentransfer pesan teks biasa dari satu komputer ke komputer yang lain. *Mail server* berjalan pada satu komputer dan hanya mampu melayani beberapa pengguna dalam skala kecil. Pada pertengahan tahun 1990-an beberapa perusahaan merintis layanan *email* gratis untuk pengguna umum dengan menggunakan teknologi *web* [LMJ-11].

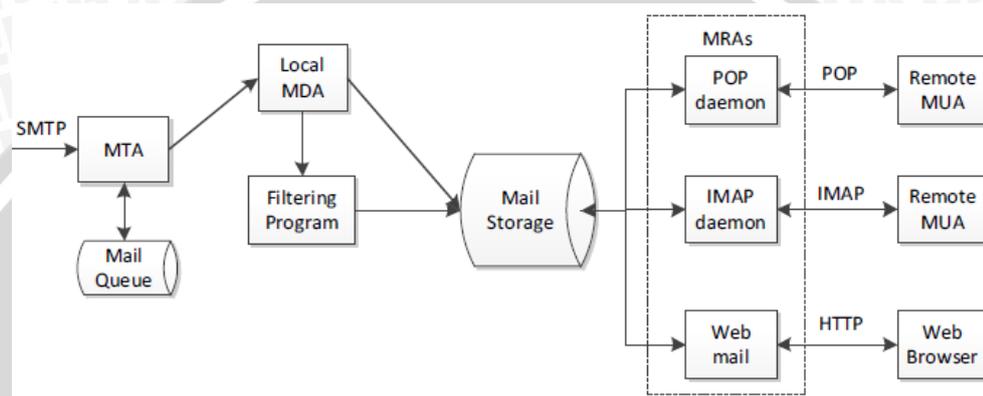
Format dasar pengiriman pesan melalui *internet* didefinisikan dalam RFC 5322. Sebagian besar proses pengiriman pesan dari komputer satu ke komputer yang lain menggunakan SMTP (*Simple Mail Transfer Protocol*). Pada awalnya protokol SMTP hanya mendukung 7 *bit* karakter dengan format ASCII yang membatasi isi pesan menjadi urutan baris yang berisi karakter ASCII saja. Sebuah *email* terdiri dari 2 komponen yaitu *header* dan *body* [LMJ-11]. *Header* terdiri dari beberapa *field*, yaitu:

1. Alamat pengirim *email*
2. Alamat penerima *email*
3. Hari, tanggal dan jam pengiriman *email*
4. Id dari *email* yang dikirimkan
5. *Subject* atau judul *email*

Komponen *body email* merupakan bagian opsional sebuah *email*. Pengguna dapat mengisi bagian dari *body* email atau membiarkannya tetap kosong. Bagian tersebut biasanya berisi mengenai isi pesan yang akan dikirimkan oleh pengguna baik berupa tulisan maupun *file* yang dilampirkan dalam sebuah *email*.

Sebuah *mail server* terdiri dari beberapa komponen yakni MTA (*Mail Transfer Agent*), MDA (*Mail Delivery Agent*) dan MRA (*Message Retrieval Agent*). MTA bertindak sebagai pengirim dan penerima pesan yang melalui sebuah *mail server* melalui SMTP. *Email* yang diterima oleh MTA akan disimpan dalam sebuah antrian kemudian MDA dipanggil untuk mengirimkan dan menyimpan *email* sesuai alamat penerima yang dituju.

Ketika pengguna ingin membaca *email* dia dapat masuk atau *log in* menggunakan akun yang sudah terdaftar. MRA dirancang sebagai MUA (*Message User Agent*) dimana pengguna dapat megakses akunnya dari jarak jauh melalui internet. Sementara IMAP membuat akses yang mungkin sehingga pengguna dapat mengakses *email* yang disimpan dalam sebuah server [LMJ-11].



**Gambar 2.1** Arsitektur mail server

Sumber : [LMJ-11]

### 2.2.2.1 Parameter pengujian mail server

Untuk mengetahui apakah sebuah *mail server* telah bekerja secara optimal, maka diperlukan pengujian mengenai performansi dari *mail server* tersebut. Uji performansi pada umumnya membahas tentang performansi dari *simple mail transfer prtocol* (SMTP) dan POP3 (*Post office protocol version 3*). Parameter uji mengenai dua hal tersebut adalah jumlah pengguna yang terdaftar pada *mail server* dan besar data yang mampu dikirimkan melalui *mail server* [UME-13]. Selain itu dapat juga digunakan parameter mengenai kecepatan pengiriman data untuk mengetahui *mail server* telah berfungsi secara optimal [TOM-05].

### 2.2.3 Citadel

Citadel merupakan perangkat dari SpamExile yang menyediakan kita solusi untuk semua persyaratan email yang kita butuhkan, teknologi email menyatu dengan layanan SpamExile yang unik. SpamExile menggantikan

semua teknologi yang ada pada saat ini dan merevolusi industri anti-*spam*. Menghentikan *spam* pada sumber yang benar, menghentikan *spam* yang berasal dari pengirim *spam* sejak awal terdeteksi *spam*. Citadel secara dinamis membangun blacklist server sendiri dan memeriksa sumber yang tidak diketahui terhadap server RBL (*Real-time Block Lists*) pada setiap sambungan eksternal. Server yang sudah terdaftar yang mencoba untuk koneksi ke citadel server akan ditolak secara aktif, karena semua pengguna citadel menerima update otomatis melalui topologi jaringan komunitas [EXS].

Fitur–fitur yang dimiliki oleh *citadel mail server* adalah sebagai berikut :

- a. SMTP/POP3/IMAP
- b. Mendukung akun tidak terbatas
- c. Mendukung otentikasi SMTP
- d. Ini adalah *server* yang berfungsi penuh: *server* tersebut meneruskan pesan langsung ke penukar Mail, melewati ISP Anda
- e. Memiliki *built-in* antarmuka web, yang memungkinkan pengguna untuk memeriksa surat menggunakan *web browser* dari manapun.
- f. modus administrasi dilindungi oleh *password*
- g. Mendukung responden otomatis
- h. ukuran pesan tidak terbatas
- i. Memiliki *built-in web server*. Artinya, *server* dapat diakses dari mana saja di dunia, menggunakan *browser*.
- j. Administrator dapat mengontrol server, dan melakukan pemeliharaan pengguna
- k. Pengguna dapat mengakses *account* mereka: membaca dan mengirim *email*, mengubah pengaturan mereka sendiri
- l. Antarmuka *web* dapat disesuaikan
- m. Mendukung IMAP
- n. Memiliki dukungan beberapa domain dimana anda dapat membuat *account* dengan nama yang sama pada *domain* yang berbeda
- o. Mendukung teknologi mobile (*WAP interface*)

#### 2.2.4 Embedded System

*Embedded system* adalah sistem komputer yang didesain guna menjalankan pekerjaan khusus. Berbeda dengan fungsi dari sebuah komputer yang didesain fleksibel untuk memenuhi kebutuhan penggunanya, *embedded system* dapat mengontrol *hardware* untuk melakukan tugas yang spesifik. Selain itu proses eksekusi sebuah pekerjaan yang dilakukan oleh *embedded system* lebih cepat jika dibandingkan dengan PC (*Personal Computer*). Oleh karena itu, pada umumnya *embedded system* merupakan bagian dari sebuah sistem yang besar [ESY-13].

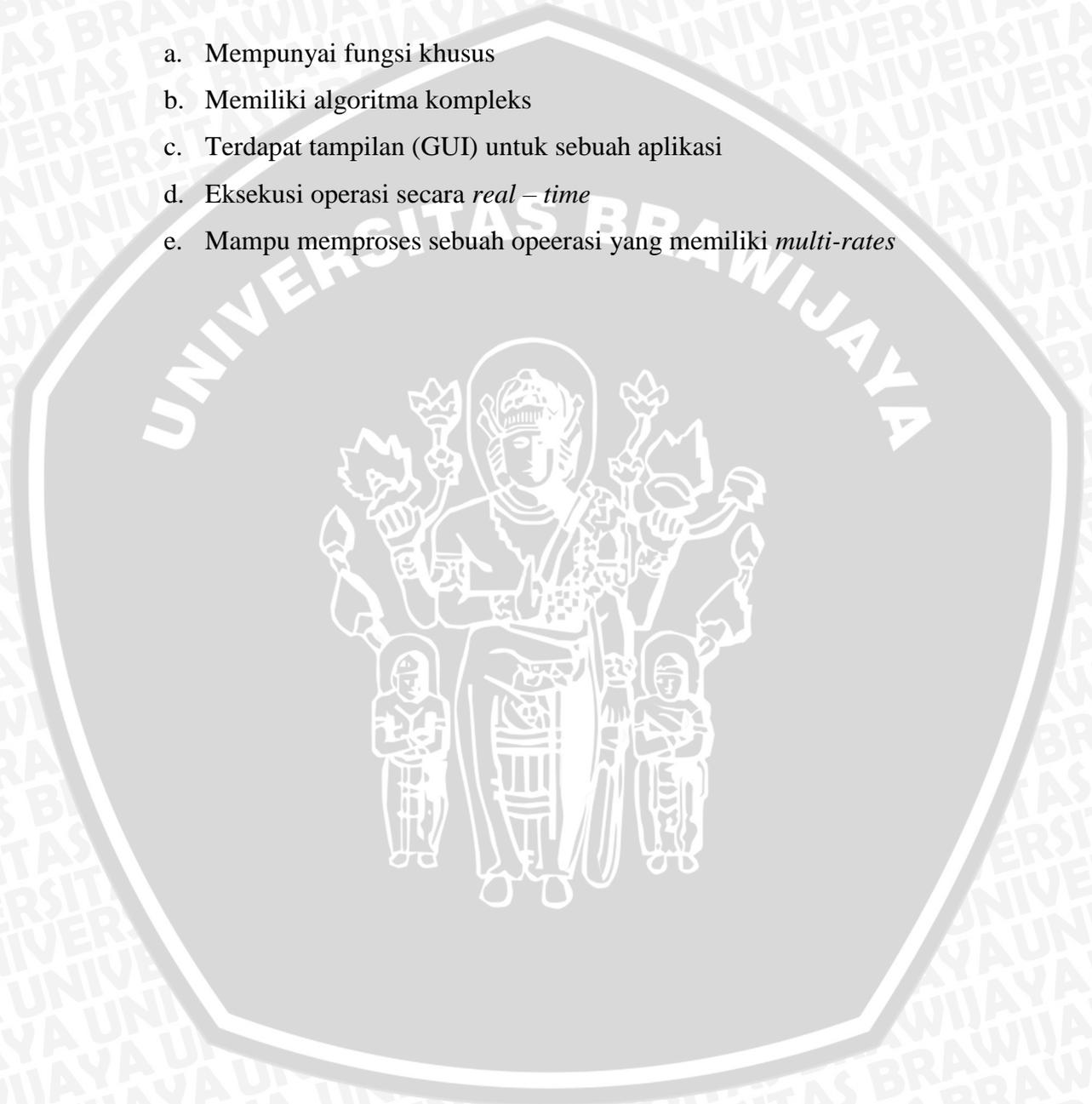
Raj kamal [KAR-08], dalam bukunya yang berjudul *Embedded System* menerangkan bahwa *Embedded system* adalah sebuah sistem memiliki perangkat lunak tertanam pada *hardware* atau perangkat keras dan dapat didedikasikan untuk sebuah aplikasi atau menjadi bagian tertentu dari sebuah aplikasi. *Embedded system* juga dapat berfungsi sebagai bagian dari sebuah sistem yang besar. Perangkat lunak dapat ditanamkan pada ROM (*Read Only Memory*) atau sebuah *removable disk* seperti *flashdisk* dan *SD card memory*. *Embedded system* memiliki tiga komponen utama yakni:

1. *Hardware* atau perangkat keras yang memiliki fungsi seperti layaknya komputer pada umumnya
2. Melekatkan perangkat lunak aplikasi utama ke dalam flash atau ROM dan aplikasi perangkat lunak memproses beberapa tugas.
3. Memiliki sistem operasi real time (RTOS), yang mengawasi tugas software aplikasi yang berjalan pada perangkat keras dan mengatur akses ke sumber daya sistem sesuai dengan prioritas dan waktu kendala tugas dalam sistem.

Dengan RTOS yang dimilikinya, *embedded system* mampu mengeksekusi beberapa proses baik berupa *thread* atau tugas secara bersama – sama. RTOS juga menyediakan mekanisme dimana prosesor dapat menjalankan setiap proses sesuai dengan penjadwalan dan mampu melakukan *context-switch* atau pergantian antar berbagai proses yang dieksekusi. Hal tersebut memungkinkan untuk dilakukan karena RTOS menetapkan beberapa aturan

selama proses eksekusi aplikasi sehingga aplikasi dapat diselesaikan dalam selang waktu yang telah ditetapkan dan sesuai dengan prioritas yang diberikan. Selain RTOS, *embedded system* yang canggih dapat dilihat berdasarkan karakteristiknya, diantaranya adalah :

- a. Mempunyai fungsi khusus
- b. Memiliki algoritma kompleks
- c. Terdapat tampilan (GUI) untuk sebuah aplikasi
- d. Eksekusi operasi secara *real – time*
- e. Mampu memproses sebuah operasi yang memiliki *multi-rates*

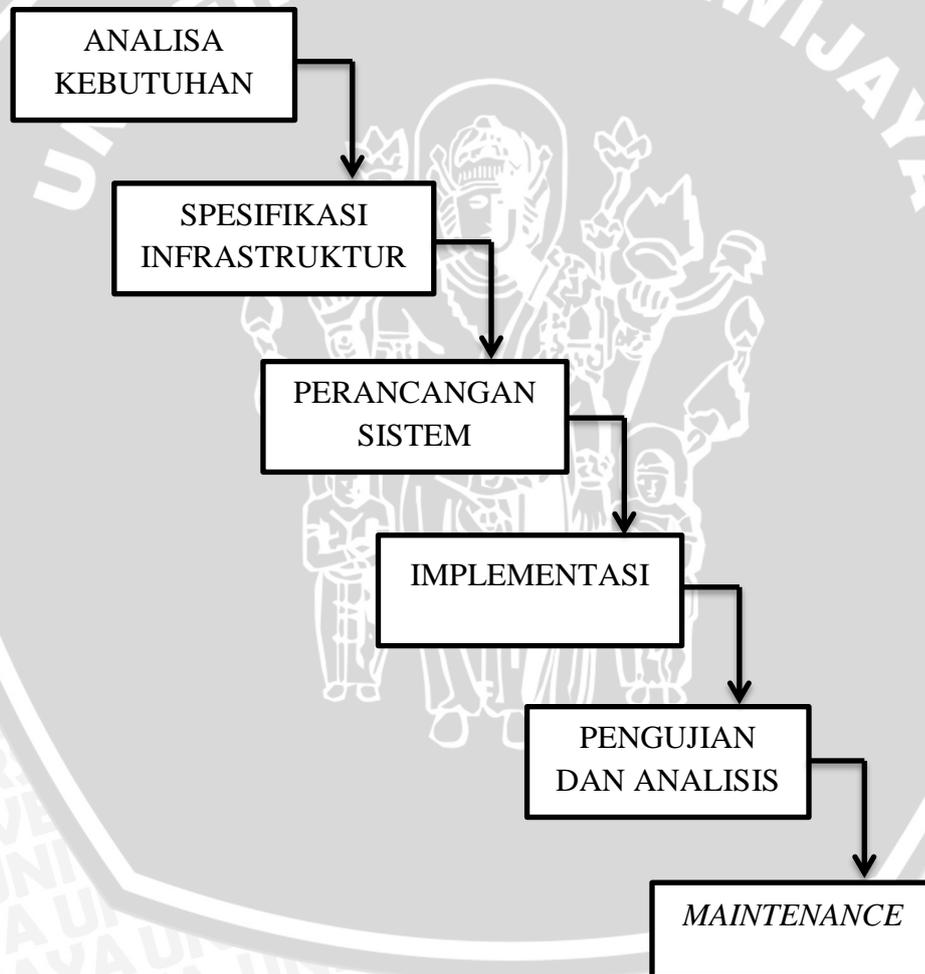


## BAB III

### METODOLOGI PENELITIAN DAN PERANCANGAN

#### 3.1 Metode Penelitian

Bab ini menjelaskan langkah-langkah yang akan ditempuh dalam penyusunan skripsi, yaitu perancangan, implementasi dan pengujian dari aplikasi perangkat lunak yang akan dibuat. Kesimpulan dan saran disertakan sebagai catatan atas aplikasi dan kemungkinan arah pengembangan aplikasi selanjutnya.



Gambar 3.1 Diagram Alir Metode Penelitian

### 3.1.1 Analisa Kebutuhan

Analisa kebutuhan dilakukan dengan cara mengumpulkan dan mempelajari segala macam informasi yang berhubungan dengan *citadel mail server*, mini komputer *raspberry pi* dan segala hal yang berhubungan dengan hal – hal mengenai *mail server* pada *embedded system*.

#### 3.1.1.1 Mail server

Dalam penelitian ini, dibutuhkan sebuah *mail server* yang berbasis *open source* atau gratis. Dalam penelitian ini juga dibutuhkan sebuah aplikasi *black box*, dimana aplikasi tersebut tidak bergantung pada sekelompok dependensi dalam sebuah sistem. Sehingga aplikasi tersebut diharapkan dapat berjalan secara portabel atau mandiri. Selain itu, penulis juga menawarkan pengaturan konten yang memungkinkan untuk menambah aturan pengamanan sesuai dengan kebijakan peraturan yang sudah ada [EXS]. Berdasarkan kebutuhan yang telah disebutkan, maka penulis memilih aplikasi Citadel untuk digunakan sebagai *mail server*.

#### 3.1.1.2 Mini Komputer

Untuk meminimalkan sumber daya dan menghemat biaya, penulis menggunakan mini komputer sebagai objek implementasi. Mini komputer dipilih karena memiliki ukuran yang lebih kecil serta harga yang lebih murah jika dibandingkan dengan komputer personal, namun memiliki fungsi yang hampir sama dengan komputer personal. Pengguna hanya perlu menambahkan sebuah SDCard sebagai media penyimpanan pada mini komputer. Guna menghindari kurangnya ketersediaan media penyimpanan pada proses penelitian, penulis memilih menggunakan sebuah SDCard dengan kapasitas 8 GB. Mini komputer yang digunakan juga harus memiliki 10/100 MB *Ethernet port* sehingga memungkinkan komunikasi melalui sebuah jaringan. Meskipun perangkat yang digunakan memiliki ukuran yang relatif kecil serta harga yang lebih terjangkau, namun memiliki kemampuan yang tidak kalah tangguh jika dibandingkan

perangkat *embedded system* yang lain. Setelah mempertimbangkan beberapa hal tersebut, maka penulis menggunakan perangkat Raspberry Pi untuk penelitian ini.

### 3.1.2 Spesifikasi Infrastruktur

Pada tahap spesifikasi infrastruktur ini dilakukan proses pemilihan perangkat yang akan digunakan dalam implementasi *mail server* pada *embedded system*. Spesifikasi infrastruktur meliputi pemilihan perangkat lunak dan perangkat keras yang akan digunakan.

#### 3.1.2.1 Kebutuhan Perangkat Lunak atau *Software*

Pada penelitian ini diperlukan beberapa perangkat lunak untuk proses implementasi. Perangkat lunak yang diperlukan adalah sebuah sistem operasi yang dapat diimplementasikan pada mini komputer dan perangkat lunak yang berfungsi sebagai mail server. *Raspbian wheezy* adalah sistem operasi yang digunakan untuk mengoperasikan *embedded system raspberry pi*. *Raspbian* adalah sistem operasi berbasis *linux* debian yang dioptimalkan untuk perangkat keras raspberry pi.

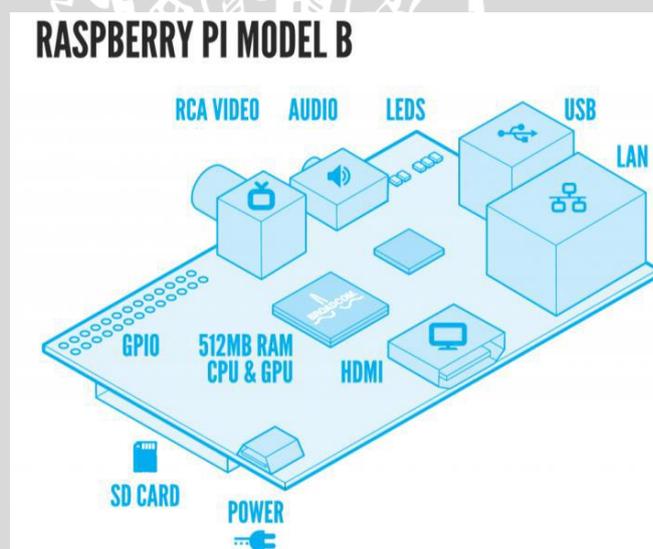
Perangkat yang digunakan sebagai *mail server* adalah perangkat *citadel*. Didalam buku panduan yang diterbitkan oleh *Spamexile* [EXS], menerangkan bahwa *Citadel* adalah sebuah perangkat *open source* yang berfungsi sebagai *email server* berbasis *web*. Keunggulan menggunakan perangkat ini adalah sebagai berikut:

- a. Mengurangi overhead
- b. Menghemat kapasitas jaringan
- c. Mengurangi kebutuhan hardware
- d. Mengurangi persyaratan software
- e. Menghapus persyaratan filter
- f. Meningkatkan produktivitas
- g. Menghemat waktu

- h. Mengurangi kewajiban perusahaan dengan memblokir konten yang menyinggung dan seksual secara eksplisit
- i. Kebutuhan Perangkat Keras atau Hardware

### 3.1.2.2 Kebutuhan Perangkat Keras atau Hardware

Perangkat keras yang dibutuhkan dalam proses penelitian ini adalah mini komputer. Mini komputer yang digunakan juga telah memiliki kapasitas memori RAM sebesar 512 MB guna menunjang perangkat lunak yang akan digunakan. Selain itu, mini komputer yang digunakan harus memiliki *port ethernet* guna proses koneksi dengan sebuah jaringan. Perangkat mini komputer yang sesuai dengan kebutuhan penelitian ini adalah perangkat Raspberry Pi. Terdapat dua jenis perangkat *raspberry pi* yaitu *raspberry pi* model A dan *raspberry pi* model B.

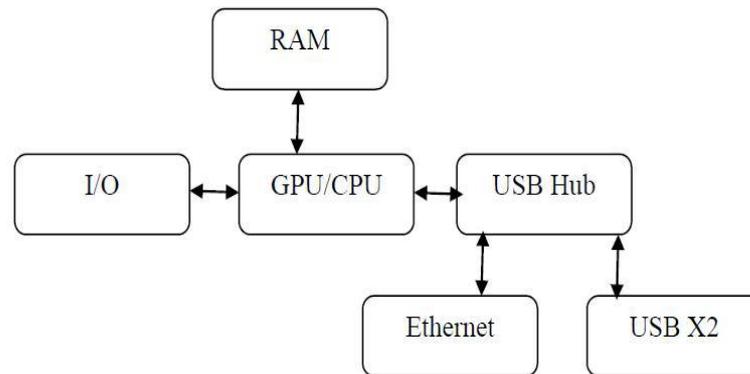


Gambar 3.2 Raspberry Pi

Sumber: [RPF-13]

Perangkat yang digunakan adalah *raspberry pi* model B karena memiliki beberapa keunggulan dibandingkan dengan perangkat *raspberry pi* model A yakni :

1. *Raspberry pi* model A hanya memiliki RAM sebesar 256 MB, pada model B memiliki RAM sebesar 512 MB.
2. Terdapat *port Ethernet* pada *raspberry pi* model B.
3. Memiliki 2 *port USB*, dimana pada *raspberry pi* model A hanya terdapat satu *port*.



**Gambar 3.3** Blok Diagram Raspberry Pi

Sumber : [PNP-13]

Seperti yang telah dijelaskan diatas bahwa perangkat *raspberry pi* model B memiliki beberapa kelebihan jika dibandingkan dengan perangkat *raspberry pi* model A. Secara lebih rinci mengenai spesifikasi yang dimiliki oleh perangkat *raspberry pi* model B dapat dilihat melalui tabel spesifikasi sebagai berikut.

**Tabel 3.1** Spesifikasi raspberry pi

Sumber : [RPI-13]

Chip	Broadcom BCM2835 SoC full HD multimedia applications processor
CPU	700 MHz Low Power ARM1176JZ-F Applications Processor
GPU	Dual Core VideoCore IV® Multimedia Co-Processor
Memory	512MB SDRAM
Ethernet	onboard 10/100 Ethernet RJ45 jack
USB 2.0	Dual USB Connector
Video Output	HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC)
Audio Output	3.5mm jack, HDMI
Onboard Storage	SD, MMC, SDIO card slot
Operating System	Linux
Dimension	8.6cm x 5.4cm x 1.7cm

*Raspberry pi* merupakan perangkat yang mudah digunakan tetapi memiliki performansi yang baik, terjangkau dan tidak mudah rusak. Perangkat ini juga cocok digunakan oleh pengguna yang sedang mempelajari bagaimana sebuah komputer dapat bekerja. *Raspberry pi* bekerja seperti layaknya komputer pada umumnya, tetapi perangkat ini memiliki prosesor yang berbeda sehingga hanya memungkinkan untuk dioperasikan menggunakan sistem operasi berbasis linux [CAS-12].

### 3.1.3 Perancangan Sistem

Tahap ini dilakukan guna merancang sistem yang akan dibuat berdasarkan hasil dari analisa kebutuhan mengenai *mail server* pada *embedded system*. Setelah menentukan jenis kebutuhan yang diperlukan, tahap selanjutnya adalah melakukan perancangan *mail server* pada *embedded system* menggunakan perangkat mini komputer Raspberry Pi. Perancangan diawali dengan proses penentuan perangkat lunak dan perangkat keras yang digunakan. Perangkat lunak yang digunakan adalah sistem operasi Windows 7, sistem operasi Raspbian Wheezy dan Citadel sebagai *mail server*. Perangkat keras yang digunakan adalah komputer personal sebagai pengujian dan mini komputer Raspberry Pi. Hasil dari tahap perancangan ini akan dijadikan bahan acuan pada tahap implementasi.

### 3.1.4 Implementasi

Dalam tahap ini, dilakukan implementasi berdasarkan studi literatur dan analisa kebutuhan yang telah dibuat pada tahap sebelumnya. Implementasi *mail server* meliputi:

1. Instalasi sistem operasi *raspbian wheezy* sebagai *user interface* dalam menggunakan perangkat *raspberry pi*.
2. Instalasi dan konfigurasi perangkat *citadel* yang digunakan sebagai *mail server*.

### 3.1.5 Pengujian dan Analisis

Pengujian pada sistem dilakukan untuk menunjukkan bahwa sistem mampu bekerja sesuai dengan spesifikasi kebutuhan. Pada tahap ini dilakukan pengujian mengenai performansi dari *mail server* dengan dua skenario. Skenario pertama dilakukan guna mengetahui nilai *transfer rate* dari *mail server* dengan menggunakan aplikasi sebagai alat bantu uji. Skenario kedua berfungsi untuk mendapatkan nilai integritas data yang dikirimkan menggunakan *mail server* pada *embedded system*.

### 3.1.6 Maintenance

Pada tahap ini dilakukan proses *maintenance* atau pemeliharaan pada sistem. Pemeliharaan yang dilakukan adalah proses perubahan konfigurasi dari sistem guna mendapatkan sebuah sistem yang dikehendaki. Tahap ini berjalan seiring dengan proses implementasi *mail server* pada *embedded system*. Proses yang dilakukan yakni mengenai perubahan konfigurasi *webcit* yang berfungsi sebagai tampilan antar muka pengguna pada *web browser* dan proses konfigurasi *IP address* dari mini komputer Raspberry Pi.

Sebelum *mail server embedded system* diimplementasikan pada sebuah sistem dilapangan, maka perlu dilakukan proses sinkronisasi data dengan *mail server master* yang ada. Proses tersebut menggunakan metode replikasi dan sinkronisasi *database LDAP*. Data dari pengguna yang ada pada *mail server master* diduplikasi atau disalin pada server LDAP yang ada pada *mail server embedded system*, namun hanya sebatas data pengguna pada jaringan lokal tersebut. Kemudian dilakukan proses sinkronisasi sehingga data dapat diperbaharui secara kontinu.

## 3.2 Perancangan

Tahap perancangan merupakan tahap dimana penulis menentukan langkah – langkah yang akan dilakukan selama proses penelitian. Tahap ini terdiri dari dua bagian yakni mengenai analisis kebutuhan perangkat lunak dan kebutuhan perangkat keras. Perancangan bertujuan untuk merancang komponen yang telah diperoleh sesuai analisa kebutuhan.

### 3.2.1 Analisa Kebutuhan Perangkat Lunak dan Perangkat Keras

Dalam tahap ini penulis akan menganalisa seluruh kebutuhan mengenai perangkat lunak dan perangkat keras yang diperlukan untuk merancang sistem. Analisa kebutuhan dilakukan guna mengetahui komponen yang akan digunakan pada perancangan *mail server* pada *embedded system*. Berikut adalah tabel kebutuhan perangkat keras dan perangkat lunak yang diperlukan.

**Tabel 3.2** Tabel Kebutuhan Perangkat Keras

Komputer Penguji	<i>Mail Server Embedded System</i>	Komponen Jaringan
Processor Intel Dual Core	Raspberry Pi Processor ARM	Kabel UTP Cat 5
2 GB RAM	512 KB RAM	Konektor RJ 45
320 GB Hardisk Drive	8 GB SD Card	

**Tabel 3.3** Tabel Kebutuhan Perangkat Lunak

Komputer Penguji	<i>Mail Server Embedded System</i>
Sistem Operasi Windows 7 Ultimate	Sistem Operasi Raspbian Wheezy
ServerTool 10.4.0	Citadel

### 3.2.2 Perancangan Perangkat Lunak dan Perangkat Keras

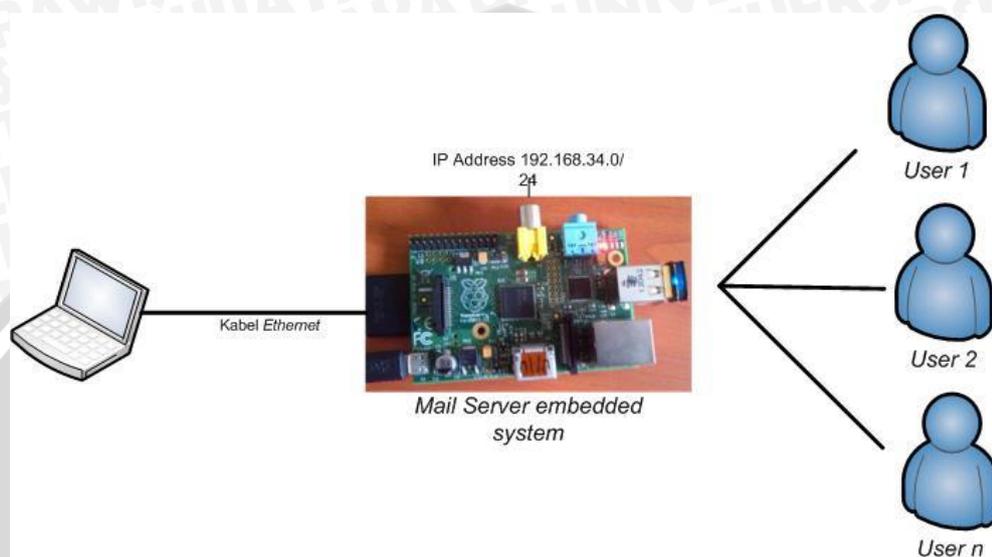
Perancangan *mail server* pada *embedded system* dilakukan setelah proses analisa kebutuhan selesai dilakukan. Perangkat lunak yang dipakai adalah sistem operasi Raspbian Wheezy dan Citadel sebagai *mail server*. Penulis menggunakan layanan tampilan melalui desktop yang disediakan oleh sistem operasi Raspbian Wheezy guna mempermudah proses penelitian.

Citadel *mail server* yang digunakan memerlukan proses konfigurasi pada bagian webcit yang digunakan sebagai tampilan antar muka dengan pengguna. Perubahan dilakukan dengan menambahkan *file* webcit.css yang berisi konfigurasi halaman utama *mail server* berbasis *web* pada direktori `/usr/local/webcit`.

### 3.2.3 Topologi Perancangan

Pada perancangan *mail server* yang menggunakan *embedded system*, system yang dirancang bertujuan untuk membantu performansi dari *mail server master*. Sehingga ketika diperlukan komunikasi surat elektronik pada jaringan lokal tidak perlu mengakses *mail server master* yang ada. Hal ini

bertujuan untuk mengoptimalkan proses komunikasi *email* pada sebuah jaringan lokal.



**Gambar 3.4** Topologi Perancangan

Komputer pengujian digunakan untuk mengakses *mail server embedded system* sehingga dapat melihat tampilan antar muka dari *mail server* guna mempermudah proses pengujian. *Mail server* yang ditanamkan pada mini komputer Raspberry Pi diakses menggunakan kabel *Ethernet* pada skala jaringan lokal.

## BAB IV IMPLEMENTASI

Pada tahap ini membahas mengenai langkah – langkah implementasi *mail server embedded system*. Sesuai dengan yang telah dijelaskan pada tahap perancangan, implementasi mengenai perangkat mini komputer *raspberry pi*. Perangkat tersebut akan digunakan sebagai *mail server embedded system* dengan menanamkan perangkat lunak *citadel*.

### 4.1 Spesifikasi Lingkungan Infrastruktur

Spesifikasi mengenai perangkat keras dan perangkat lunak yang digunakan dalam perancangan *mail server embedded system* menggunakan mini komputer *raspberry pi* adalah sebagai berikut.

**Tabel 4.1** Spesifikasi Lingkungan Perangkat Keras

Perangkat keras	Spesifikasi
CPU / Processor	ARM1176JZFS 700 Mhz
Memory (RAM)	512 Mb
Storage	SDHC Card 8 Gb
Network Interface	Ethernet LAN
USB 2.0	Dual USB Connector

**Tabel 4.2** Spesifikasi Lingkungan Perangkat Lunak

Perangkat Lunak	Spesifikasi
Sistem Operasi	Linux Raspbian wheezy
Perangkat Lunak <i>mail server</i>	Citadel

### 4.2 Implementasi Raspberry Pi

Tahapan penelitian selanjutnya yang dilakukan adalah proses implementasi pada perangkat *raspberry pi* yang digunakan sebagai *mail server*. Langkah pertama yang dilakukan adalah instalasi sistem operasi pada perangkat *raspberry pi*. Sistem operasi yang digunakan adalah Linux *raspbian wheezy*. *Raspbian*

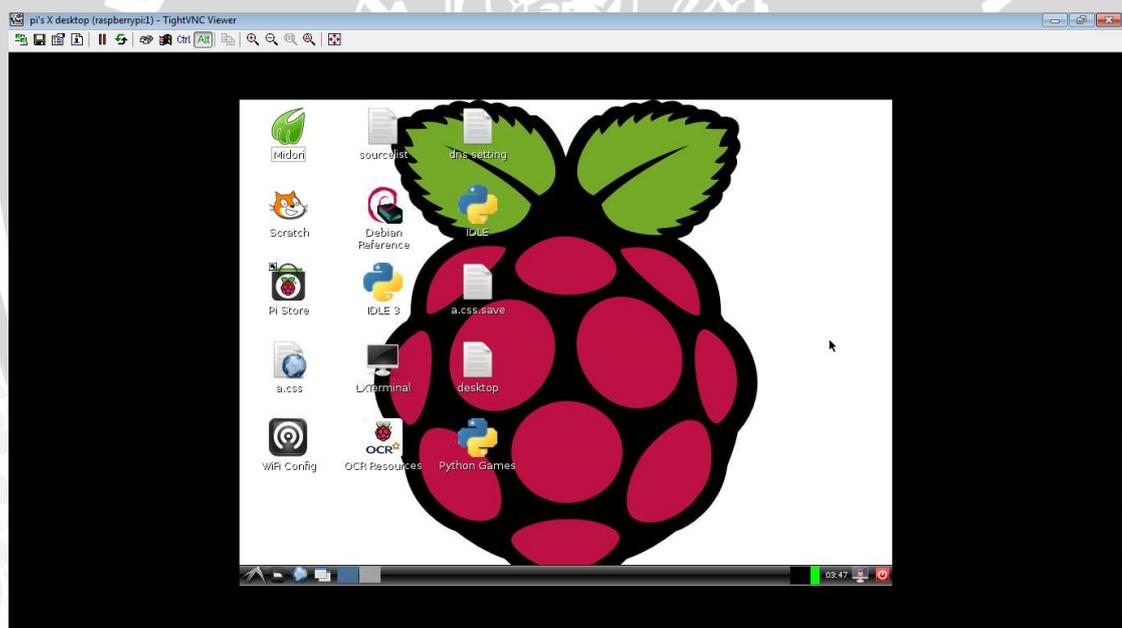
wheezy digunakan karena sistem operasi ini *compatible* atau sesuai digunakan pada perangkat mini komputer raspberry pi. Sistem operasi digunakan untuk menampilkan GUI atau tampilan tatap muka pengguna guna proses instalasi dan konfigurasi *mail server*. Berikut adalah tampilan sistem operasi raspbian wheezy menggunakan aplikasi *tightvnc*. Berikut adalah konfigurasi *tightvnc* pada raspberry pi :

```
$ vncserver -geometry 800x600 -dpi 24
```

Penjelasan mengenai *syntax* diatas adalah:

-*geometry 800x600* : membuat tampilan yang dimunculkan pada aplikasi *tightvnc viewer* dengan resolusi 800x600 *pixel*.

-*dpi 24* : konfigurasi ketajaman warna pada layar tampilan, nilainya mulai dari 2 - 36

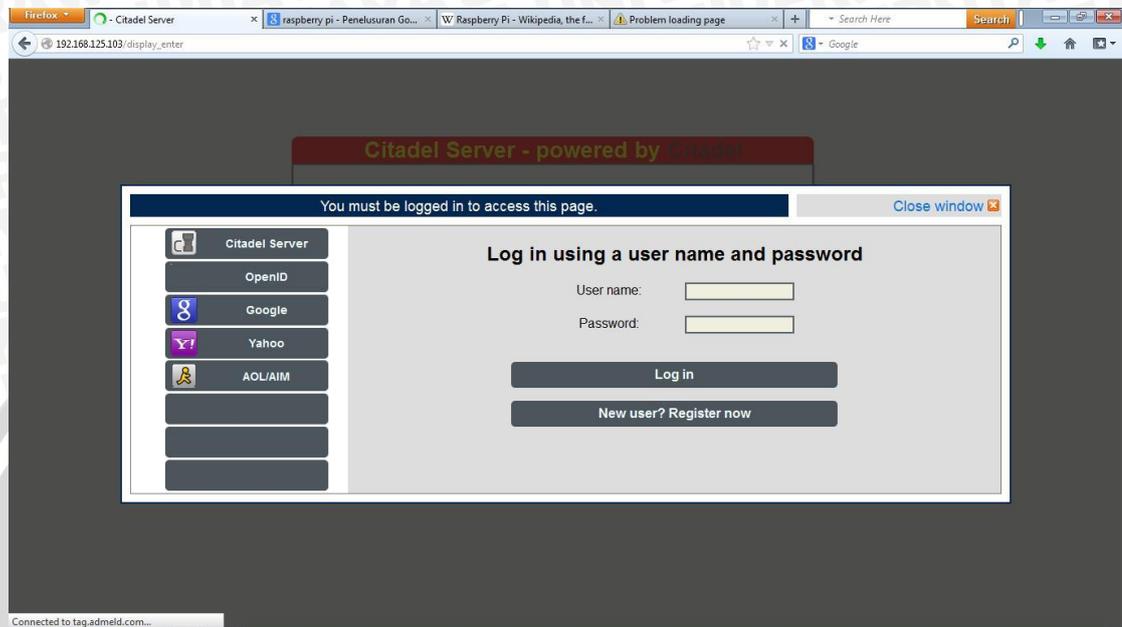


Gambar 4.1 Tampilan Raspbian Wheezy

### 4.3 Implementasi Citadel

Citadel adalah aplikasi yang digunakan sebagai *mail server* sehingga pengguna dapat memanfaatkannya sebagai sarana komunikasi *email*. Citadel mempunyai tampilan tatap muka atau *user interface* berbasis *web*, sehingga pengguna dapat mengakses *mail server* dengan menggunakan *web browser* yang

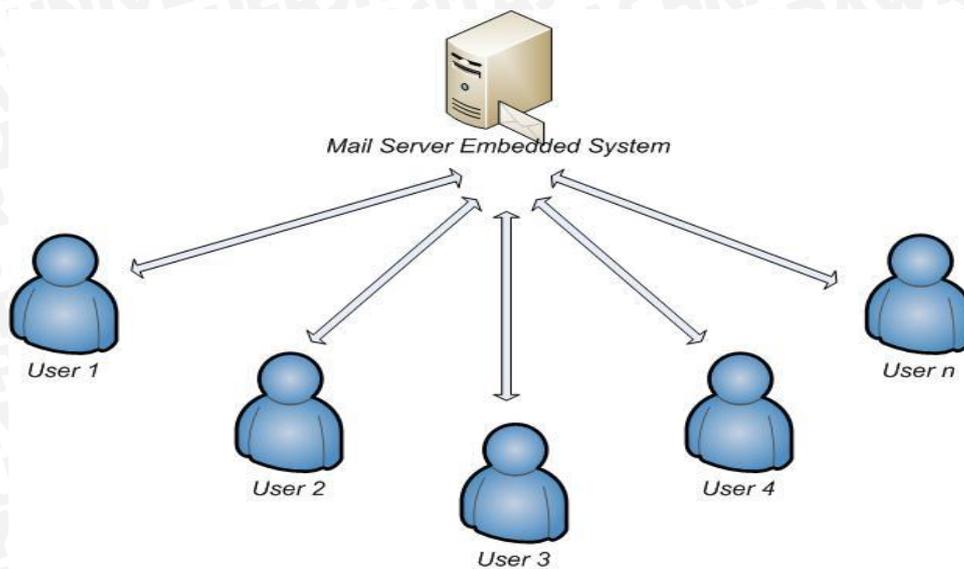
dimiliki. Citadel merupakan aplikasi *open source* sehingga kita dapat mengunduh dan menggunakannya secara gratis. Selain itu kita juga dapat melakukan konfigurasi pada aplikasi ini sesuai dengan kebutuhan.



Gambar 4.2 Tamplinan UI Citadel

#### 4.4 Implementasi infrastruktur

Setelah melalui tahap analisa kebutuhan dan perancangan, maka tahap selanjutnya adalah tahap implementasi sistem. Implementasi yang dilakukan juga mengacu pada kedua tahap tersebut. *Mail server* akan difungsikan sebagai komunikasi *email* oleh pengguna yang berada pada sebuah jaringan lokal.



**Gambar 4.3** Skema implementasi

Skenario pada gambar diatas dapat dijelaskan sebagai berikut :

1. Pengguna yang akan melakukan komunikasi *email* kepada pengguna lain yang berada pada jaringan lokal dapat mengakses *mail server embedded system* melalui *browser* sesuai yang dimiliki oleh pengguna.
2. Setiap pengguna diwajibkan mendaftarkan diri untuk memperoleh sebuah akun sebelum dapat memanfaatkan fasilitas *mail server embedded system*.
3. Setelah masing – masing pengguna telah memperoleh akun, maka pengguna dapat masuk ke *mail server embedded system* dan dapat menggunakannya sebagai sarana komunikasi *email* dengan pengguna lain pada jaringan lokal.

## BAB V

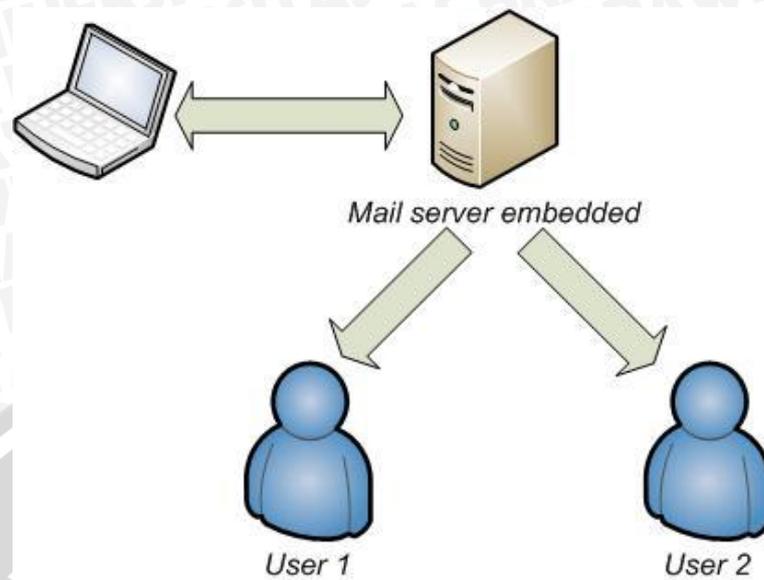
### PENGUJIAN DAN ANALISIS

#### 5.1 Pengujian

Tahap pengujian *mail server* ini melalui dua skenario pengujian. Skenario pengujian dibedakan berdasarkan parameter yang digunakan. Pada skenario pertama menggunakan beberapa parameter yakni besar data, *transfer rate* dan waktu pengiriman. Sedangkan pada skenario kedua menggunakan besar data, jumlah pengguna dan integritas data yang dikirimkan sebagai parameter pengujianya.

##### 5.1.1 Skenario Pengujian Pertama

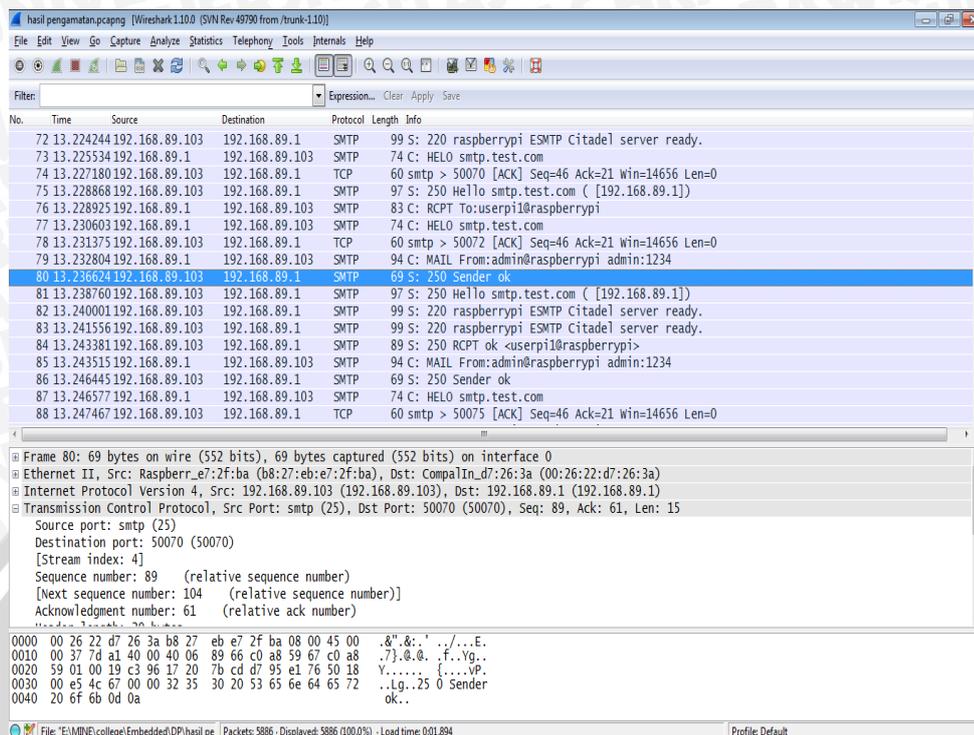
Skenario pertama dibagi menjadi sepuluh tahap berdasarkan besar *file* atau dokumen yang dikirimkan. Besar *file* yang dikirimkan adalah 1MB sampai dengan 10MB. Setiap tahapan pengujian dilakukan sebanyak tiga kali, kemudian didapatkan nilai rata - rata dari proses pengujian. Pada pengujian skenario pertama digunakan sebuah aplikasi yang bernama *ServerTool* 10.4.0. Aplikasi ini berfungsi untuk mengetahui kinerja SMTP (*Simple Mail Transfer Protocol*) dari sebuah *mail server*. Terdapat beberapa parameter pengujian yang disediakan oleh aplikasi ini yakni besar data yang dikirimkan, *transfer rate* pada saat pengujian dan waktu yang diperlukan untuk setiap pengujian.



**Gambar 5.1** Skenario Pengujian pertama

*Mail* server diakses melalui sebuah komputer personal dengan menggunakan jaringan LAN. Personal komputer tersebut telah diinstal aplikasi *ServerTool* sebagai alat bantu uji dari *mail server embedded* yang diterapkan. Aplikasi ini memiliki beberapa *field* atau kolom yang harus diisi oleh pengguna sebelum melakukan proses pengujian. Kolom *Host* diisi dengan alamat *domain* dari *server* yang akan diuji, kolom *From* mengenai alamat dari pengirim *email* dan kolom *To* dapat diisi dengan alamat *email* yang akan menerima *email* tersebut. Pengguna dapat memilih untuk mengirimkan pesan yang disediakan oleh aplikasi atau pengguna juga dapat memilih *file* yang akan dikirimkan

Aplikasi ini mengirimkan data berupa *SMTP-request* kepada *mail server*, kemudian data dikirimkan kepada alamat email yang dituju sesuai dengan alamat yang diisikan pada kolom tujuan. Pada saat pengujian komputer penguji berkomunikasi dengan *mail server* menggunakan *port* yang telah disediakan secara acak. Sedangkan *Mail server* melakukan komunikasi dengan komputer penguji menggunakan *port 25* yakni *port* dari *SMTP*. Setelah data terkirim kepada email yang dituju, *mail server* mengirimkan pesan kepada *ServerTool* bahwa data berhasil dikirimkan. Proses tersebut dilakukan berulang sesuai dengan jumlah tahapan skenario.

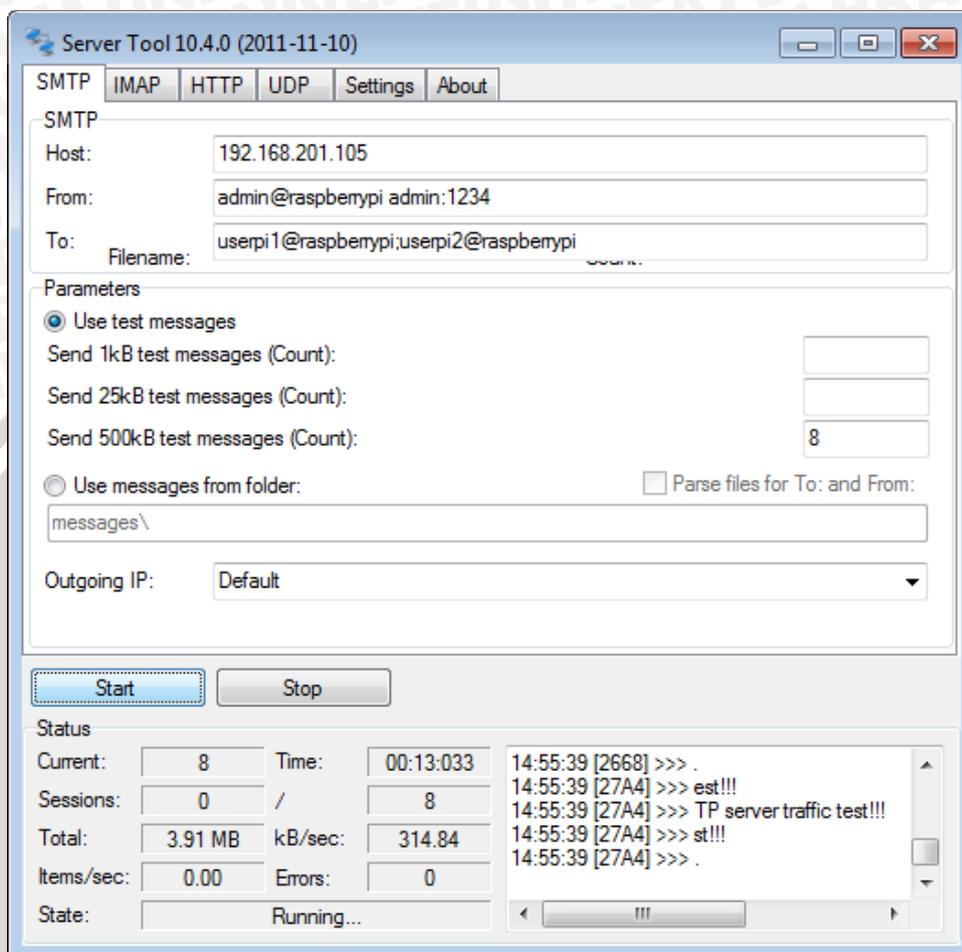


**Gambar 5.2** Pengamatan menggunakan aplikasi Wireshark

Pada gambar diatas dapat kita lihat bahwa IP *address* sumber adalah 192.168.89.103 dimana IP tersebut adalah IP dari perangkat Raspberyy Pi yang digunakan sebagai *mail server*. IP *address* tujuan adalah 192.168.89.1 merupakan IP dari komputer penguji. *Port* yang digunakan oleh Raspberyy Pi untuk mengirimkan data adalah *port* 25 atau dikenal sebagai *port* SMTP. Pada komputer penguji menerima komunikasi dari perangkat Raspberyy Pi menggunakan *registered port* secara acak, berdasarkan gambar diatas adalah *port* 50070.



Berikut adalah hasil pengujian pada skenario pertama:



Gambar 5.3 Screenshot Pengujian

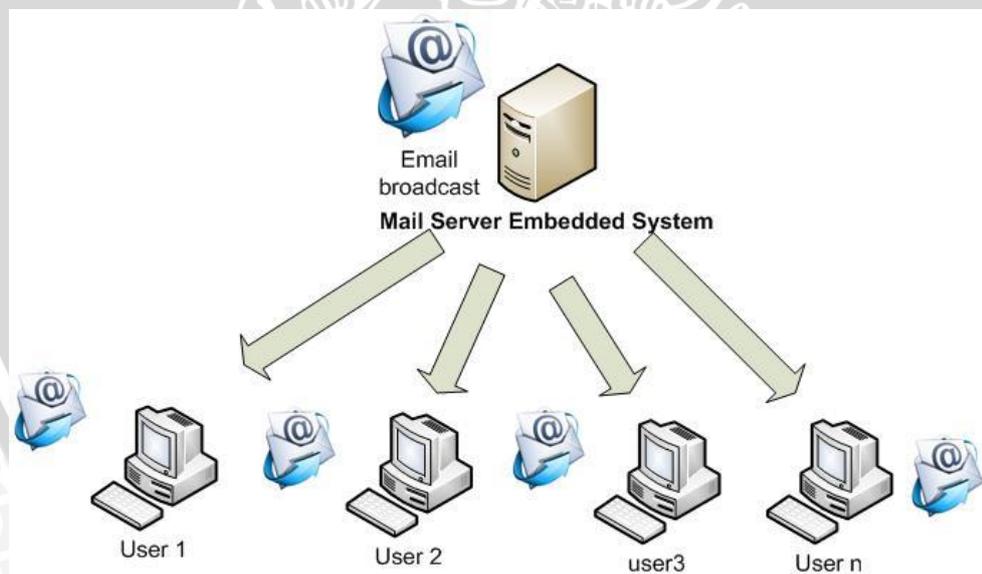
Tabel 5.1 Hasil Pengujian

Besar file (MB)	Transfer rate Rata – rata (KBps)	Waktu pengiriman rata – rata (s)
5	275.83	18.14
10	389.1	25.7
15	450.03	33.4
20	505.7	39.6
25	491.4	51
30	482.6	62.3

Pengujian ini dilakukan dalam lingkungan jaringan yang sama untuk setiap proses pengujian. Pengujian dilakukan sebanyak tiga kali percobaan untuk setiap parameter uji yang digunakan. Selanjutnya diambil nilai rata-rata dari tiga nilai yang didapat untuk setiap parameter uji (Lampiran 2). Hal ini dilakukan untuk meminimalkan varian sehingga *error* atau kesalahan yang dilakukan dapat diminimalisir [RAA-11].

### 5.1.2 Skenario Pengujian Kedua

Pada skenario kedua memiliki perbedaan dengan pengujian pada skenario pertama yakni mengenai parameter yang digunakan dan tahapan yang dilakukan pada saat pengujian. Pengujian yang dilakukan pada skenario kedua melalui sepuluh tahap. Tahapan tersebut dibagi berdasarkan besar *file* yakni 1Mb, 2Mb, 3Mb, 4Mb dan 5Mb. Jumlah pengguna yang terdaftar pada saat pengujian sebanyak 50 pengguna. Berikut adalah hasil pengujian pada skenario kedua :



Gambar 5.4 Skenario Pengujian kedua

Tabel 5.2 pengujian skenario kedua

Jumlah User	Besar File	Integritas Data
50	1 x 1 MB	100%
50	2 x 1 MB	100%
50	3 x 1MB	100%
50	4 x 1 MB	100%
50	5 x 1 MB	100%

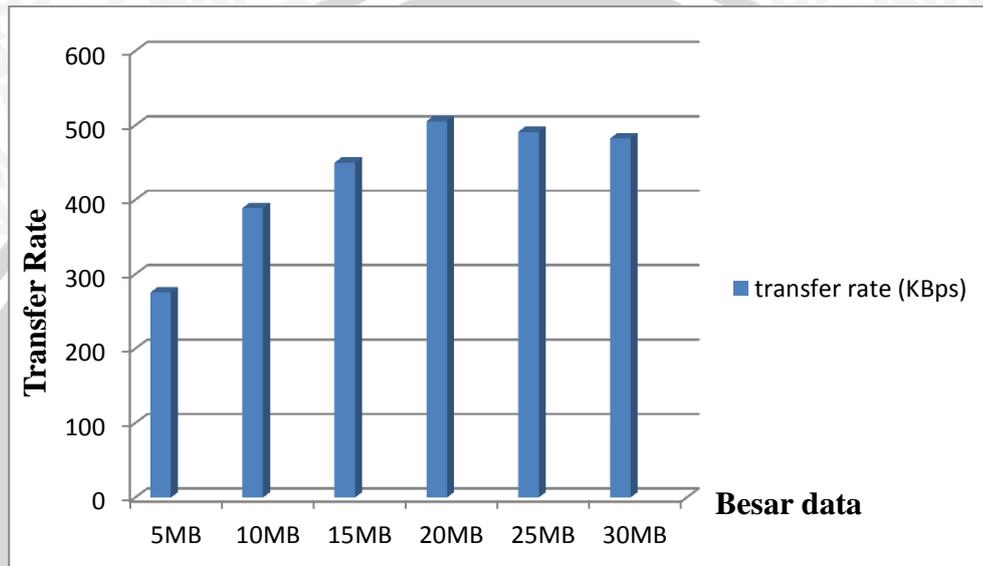
*File* yang digunakan pada pengujian ini adalah satu *file* yang memiliki kapasitas 1MB. Sehingga jika *file* yang dikirimkan memiliki besar lebih dari 1MB, maka *file* tersebut diduplikasi atau dilampirkan sesuai dengan besar parameter pengujian yang digunakan. Jadi ketika pengujian menggunakan parameter kedua yakni 2MB, maka *file* tersebut dilampirkan sebanyak dua kali sehingga total *file* yang dikirimkan sebesar 2MB dan begitu juga seterusnya.

Keseluruhan skenario pengujian yang dilakukan pada penelitian ini dilakukan dengan menggunakan jaringan lokal yakni antara komputer dengan mini komputer *raspberry pi*. Dua perangkat tersebut dihubungkan melalui *Ethernet* menggunakan kabel UTP (*Unshielded Twisted Pair*).

## 5.2 Analisis

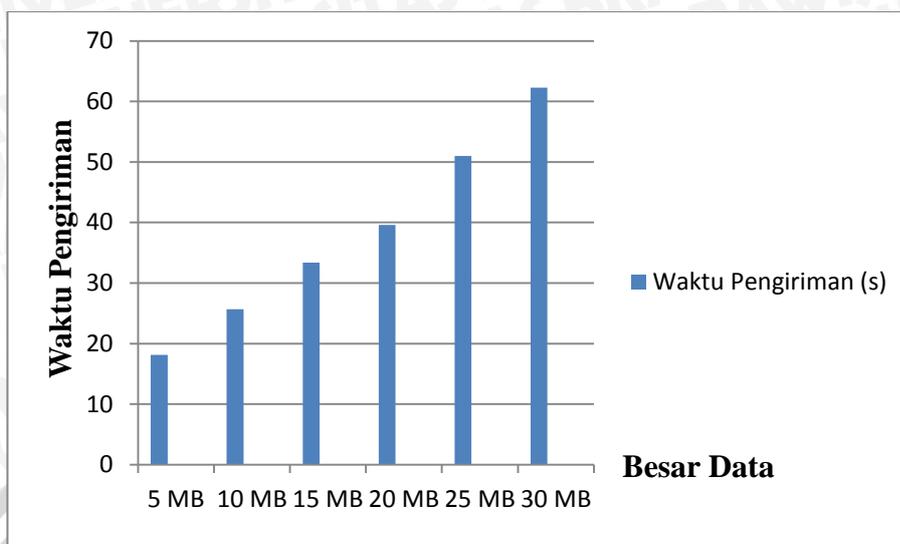
Pada sub bab ini akan membahas mengenai analisis hasil pengujian yang telah dilakukan. Analisis yang dilakukan meliputi dua bagian yakni mengenai skenario pengujian pertama dan skenario pengujian kedua. Pengujian skenario pertama dilakukan menggunakan bantuan sebuah aplikasi sebagai alat bantu uji. Sedangkan pada pengujian skenario kedua tanpa menggunakan aplikasi sebagai alat bantu uji.

Analisis pada pengujian skenario pertama dengan menggunakan aplikasi ServerTool 10.4.0 dapat dilihat pada gambar 5.4 dibawah ini. Pengujian ini bertujuan untuk mengetahui beberapa hal yakni *transfer rate* pada saat mengirim *file* atau data. Gambar dibawah ini didapatkan berdasarkan hasil yang ada pada Tabel 5.1.



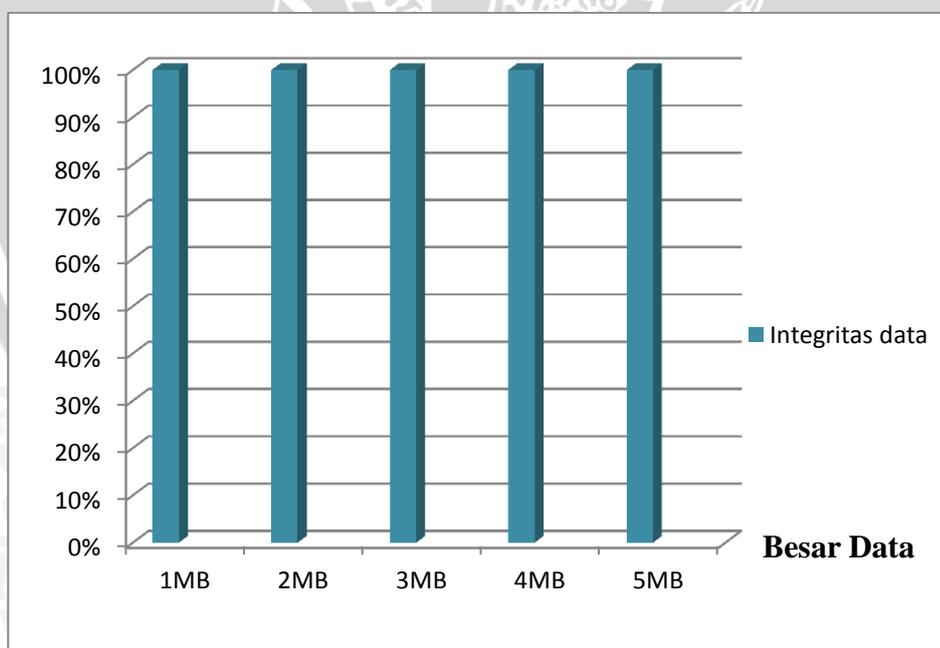
**Gambar 5.5** Grafik Hasil Pengujian Skenario 1

Hasil pengujian pada skenario pertama khususnya pada parameter pengujian ke – 1 sampai ke – 4 menghasilkan nilai *transfer rate* yang selalu meningkat atau sebanding dengan besar *file* yang dikirimkan. Namun nilai tersebut menurun pada parameter pengujian ke – 5, dimana besar *file* yang dikirimkan sebesar 25MB. *Mail server embedded system* mengalami penurunan performa ketika dibebani pekerjaan untuk mengirimkan *file* sebanyak 50 dengan besar data 500 KB kepada dua pengguna. Hal ini dipengaruhi oleh terbatasnya kapasitas *buffer* atau tempat penyimpanan data sementara yang dimiliki oleh perangkat mini komputer Raspberry Pi.



**Gambar 5.6** Grafik Waktu Pengiriman

Analisis selanjutnya dilakukan pada skenario pengujian yang kedua dimana pada pengujian ini tidak memerlukan aplikasi sebagai alat bantu pada saat pengujian. Hasil pengujian pada skenario ini dapat dilihat pada gambar 5.6 dibawah ini.

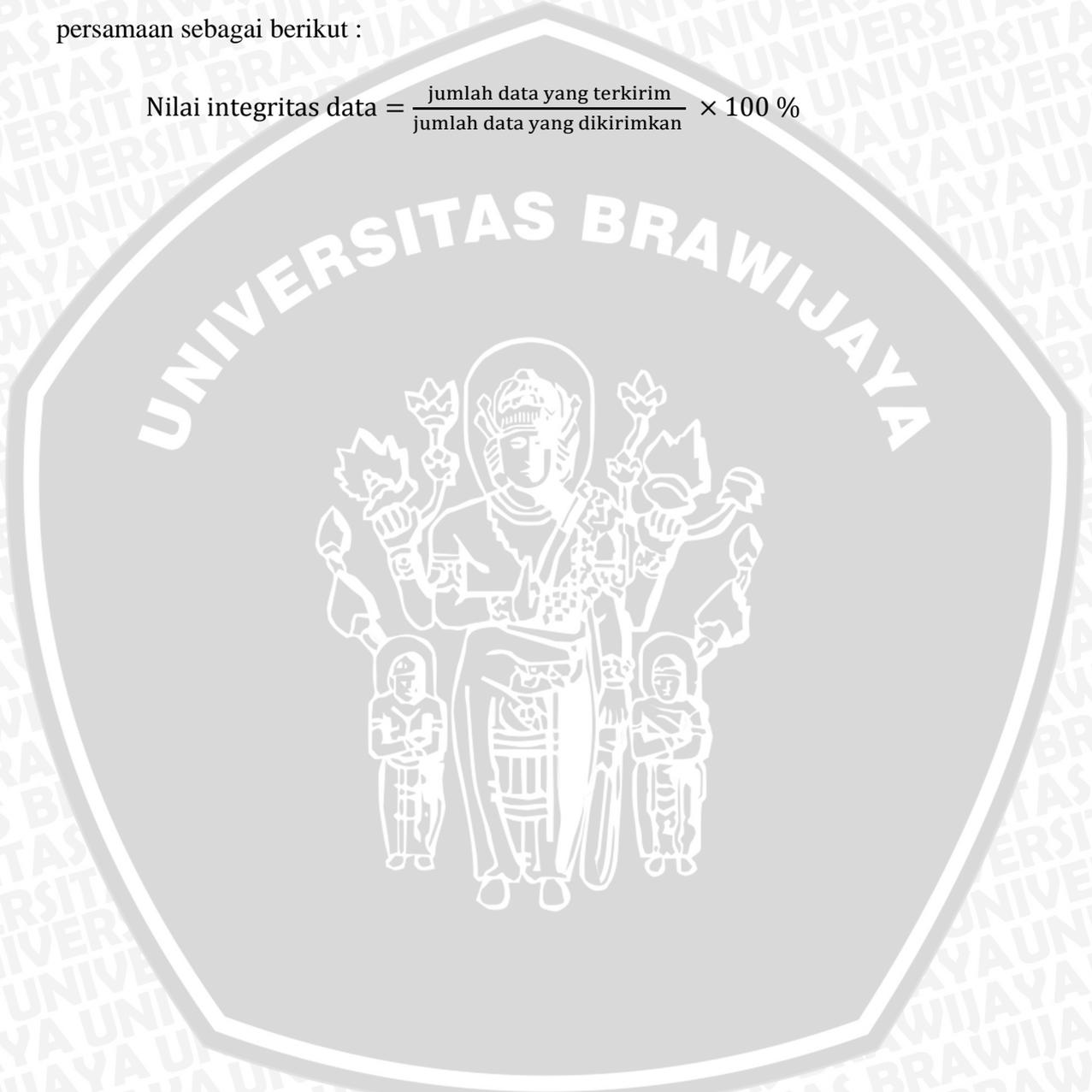


**Gambar 5.7** Grafik hasil pengujian skenario 2

Pengujian pada skenario kedua bertujuan untuk mendapatkan nilai integritas data yang dikirimkan melalui *mail server embedded system*. Hasil yang

didapat pada saat pengujian adalah nilai integritas data yang dikirimkan mencapai 100% untuk setiap parameter pengujian yang ada. Hal ini didapatkan berdasarkan pengecekan pada akun email yang menjadi tujuan pada waktu pengujian. nilai presentase untuk setiap parameter pengujian didapatkan dengan menggunakan persamaan sebagai berikut :

$$\text{Nilai integritas data} = \frac{\text{jumlah data yang terkirim}}{\text{jumlah data yang dikirimkan}} \times 100 \%$$



## BAB VI PENUTUP

### 6.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, pengujian dan analisis yang telah dilakukan, maka diambil kesimpulan sebagai berikut :

1. Perangkat mini komputer Raspberry Pi dapat digunakan sebagai *mail server embedded system*.
2. Perangkat lunak Citadel dapat digunakan sebagai *mail server* pada spesifikasi default perangkat mini komputer Raspberry Pi ( CPU 700MHz + RAM 512MB).
3. Pengujian mengenai integritas data pada *mail server embedded system* menggunakan lima parameter uji. Nilai presentase data yang didapat untuk setiap parameter pengujian yang ada adalah 100%. Dengan kata lain data yang terkirim pada alamat email yang dituju sesuai dengan data yang dikirimkan oleh email pengirim.
4. *Mail server embedded system* menggunakan perangkat mini komputer Raspberry Pi telah diuji menggunakan aplikasi ServerTool 10.4.0. Hasil yang didapat adalah *mail server embedded system* secara optimal dapat melakukan pengiriman *file* dengan *transfer rate* rata – rata 505.7 KBps pada besar *file* 20MB. Sedangkan ketika besar *file* yang dikirimkan > 20MB nilai *transfer rate* yang didapatkan lebih kecil dibandingkan dengan pengiriman *file* sebesar 20MB.

### 6.2 Saran

1. Infrastruktur ini dapat dimodifikasi dan dikembangkan dengan merubah perangkat lunak sesuai dengan kebutuhan pengguna guna pengembangan lebih lanjut.
2. Perlu dilakukan uji coba lapangan untuk mengetahui bagaimana infrastruktur ini berinteraksi dengan pengguna secara langsung.

3. Perlu dilakukan penelitian lebih lanjut terkait jumlah pengguna yang dapat menggunakan *mail server embedded system*, sehingga dapat diketahui banyaknya jumlah pengguna secara pasti untuk skala pengguna besar.



## DAFTAR PUSTAKA

- [CAS-12] Computing at school. 2012, "The Raspberry Pi Education Manual".
- [EXS] Exile, Spam. "Citadel by SpamExile The E-mail System for Small to Medium Enterprise", SpamExile Ltd.
- [ESY-13] Embedded System, 2010. Diakses melalui [Http://adiyasan.wordpress.com](http://adiyasan.wordpress.com). Tanggal akses: 23-09-2013.
- [HSE-03] Heath, S. 2003. "Embedded-Systems Design, Second Edition", Newnes.
- [KAR-08] Kamal, Raj. 2008, "Embedded System", 2<sup>nd</sup> edition, McGraw-Hill Education.
- [LMJ-11] Li, Y., Mingzhe Z., dan Jiaqi Gu. 2011, "An Email Server Optimized for Storage Issues", Fudan University.
- [MIE-13] Mangunkusumo, I.E.S.W. 2013, "Analisa dan Perancangan Keamanan Mail Server Zimbra pada Sistem Operasi Ubuntu 8.04". Hal 1.
- [NAE-13] Narendro Arifia. "E-mail", Universitas Gunadarma. Hal 2 & 13.
- [PNP-13] Padhye, N., Preet Jain. 2013. "Implementation of ARM Embedded Web Server for DAS Using Raspberry Pi". Madhya Pradesh. India
- [RAA-11] Raupong., Anisa. 2011. "Bahan Ajar Mata Kuliah Perancangan Percobaan". Universitas Hasanuddin. Makassar
- [RPF-13] Raspberry pi Foundation, 2013. Diakses melalui <http://www.raspberrypi.org/faq>. Tanggal akses : 07-10-2013
- [RPI-13] Raspberry Pi. 2013. <http://downloads.element14.com>. Tanggal akses: 16-10-2013

- [RPT-13] Raspberry Pi: The Perfect Home Server, 2013. Diakses melalui <http://www.linuxjournal.com/>. Tanggal akses: 16-10-2013
- [RRA-06] Rafiudin, rahmat. 2006, "Sistem Komunikasi Data Mutakhir", CV Andi offset. Yogyakarta.
- [RFF-11] Rajab, Fahmi Febrian. 2011. " Building a Mail Server Using Mail Citadel Transfer Agent on Linux Ubuntu 10:10 and accessing WITH Mozilla Thunderbird". Universitas Gunadarma. Depok.
- [TUY-10] The University of York. 2010, "Email standards, protocol and guidance".
- [TOM-05] Transfer rate Outgoing mail, 2005. Diakses melalui <http://www.mailarchive.com/mdaemon1@dutaint.com/msg07528>. Tanggal akses: 21-10-2013.
- [WJY-04] Wan, Jun., Yiming Hu. 2004. " A Performance Study on Internet Server Provider Mail Servers".

## LAMPIRAN - LAMPIRAN

Lampiran 1 Konfigurasi *file* webcit.css

```
#login_screen .boxlabel {
    font-size: 20pt;
    background-color: red;
    color: yellow;
}

.login_hello, .login_infos {
    font-size: 14pt;
    margin-left: 0;
    width: 60%;
    text-align: left;
}

.login_message, #login_form {
    position: absolute;
    display: block;
    left: 80%;
    padding: 10px;
    width: 150px;
}

.login_message { top: 80px; }
```

```
#login_form { top: 100px; }

#login_form .logbutton, #login_form input,
#login_form select, #login_form br {

    float: none;

    width: 130px;

    margin-top: 0;

    margin-left: auto;

    margin-right: auto;

    margin-bottom: 0;

    padding: 0

}

#login_form label {

    float: none;

    background-color: red;

    color: yellow;

    font-weight: bold;

    width: 80px;

    font-size: 9px;

    text-transform: uppercase;

    margin: 0;

    padding: 2px;

}
```

**Lampiran 2 Hasil Pengujian Skenario pertama**

Besar Data	Trasnfer Rate	Waktu Pengiriman
5MB	17.5	285.71
5MB	18.1	276.243
5MB	18.83	265.53

Besar Data	Trasnfer Rate	Waktu Pengiriman
10MB	25.72	388.8
10MB	25.95	385.4
10MB	25.44	393.1

Besar Data	Trasnfer Rate	Waktu Pengiriman
15MB	34.9	429.8
15MB	32.4	462.96
15MB	32.8	457.32

Besar Data	Trasnfer Rate	Waktu Pengiriman
20MB	39.1	511.51
20MB	39.72	503.53
20MB	39.84	502.1

Besar Data	Trasnfer Rate	Waktu Pengiriman
25MB	48.81	512.2
25MB	54.6	457.9
25MB	49.6	504.03

Besar Data	Trasnfer Rate	Waktu Pengiriman
30MB	58.3	514.6
30MB	64.1	468.02
30MB	64.5	465.12