

**PEMANFAATAN KLASTERISASI PADA PERANGKAT
RASPBERRY PI UNTUK MELAKUKAN AKUISISI MULTI
PERANGKAT**



N a m a : Aulia Rahman Ganimeda
NIM : 16523014

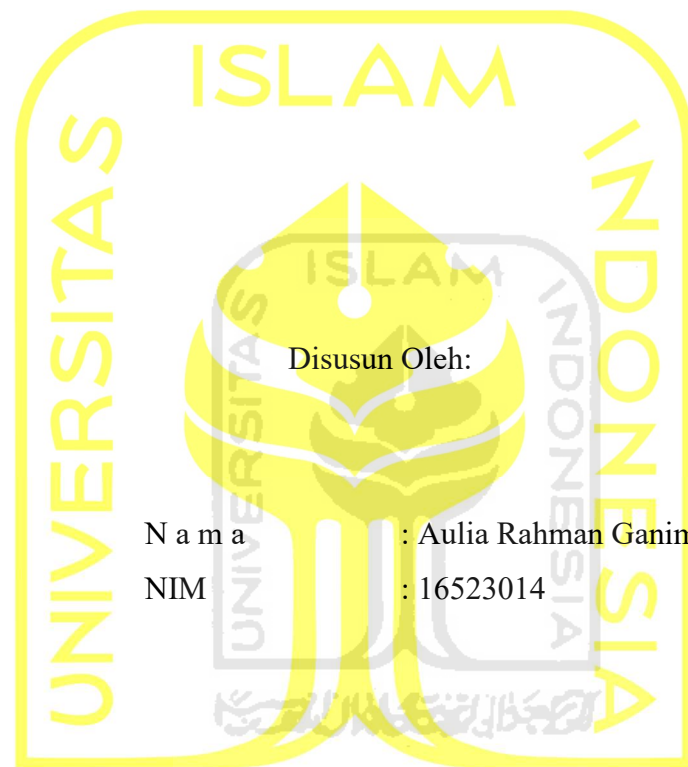
**PROGRAM STUDI TEKNIK INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2020

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**PEMANFAATAN KLASTERISASI PADA PERANGKAT
RASPBERRY PI UNTUK MELAKUKAN AKUISISI MULTI
PERANGKAT**

TUGAS AKHIR



Disusun Oleh:

N a m a : Aulia Rahman Ganimeda

NIM : 16523014

الجمعة الاستدائية
الاستدائية

Yogyakarta, 10 Juli 2020

Pembimbing

(Fietyata Yudha, S.Kom., M.Kom.)

HALAMAN PENGESAHAN DOSEN PENGUJI

**PEMANFAATAN KLASTERISASI PADA PERANGKAT
RASPBERRY PI UNTUK MELAKUKAN AKUISISI MULTI
PERANGKAT**

TUGAS AKHIR

Telah dipertahankan di depan sidang pengujian sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Teknik Informatika

di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 19 Agustus 2020

Tim Penguji

Fietyata Yudha, S.Kom., M.Kom.

Anggota 1

Syarif Hidayat, Dr., S.Kom., M.I.T.

Anggota 2

Ahmad Fathan, S.T., M.Cs.





Mengetahui,

Ketua Program Studi Teknik Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Aulia Rahman Ganimeda

NIM : 16523014

Tugas akhir dengan judul:

**PEMANFAATAN KLASIFIKASI PADA PERANGKAT
RASPBERRY PI UNTUK MELAKUKAN AKUISISI MULTI
PERANGKAT**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 10 Juli 2020



(Aulia Rahman Ganimeda)

HALAMAN PERSEMBAHAN

Alhamdulillah rabbil ‘alamin, atas berkat rahmat Allah SWT dan ridho-Nya dapat menyelesaikan tugas akhir yang sekaligus merupakan tanggung jawab dalam proses perkuliahan ini. Tugas akhir ini saya persembahkan kepada

1. Bapak Kendra Hartaya dan Ibu Ani Listiani sebagai orang tua yang telah mendidik, membesarkan serta mendukung penuh melalui usaha dan do’a.
2. Adik saya, Mirza Mahdi Andromeda yang senantiasa memberi semangat.
3. Keluarga, saudara dan teman teman yang selalu memberikan motivasi.
4. Dan pembaca yang telah meluangkan waktunya dalam membaca karya ini.



HALAMAN MOTO

"Ya Allah, tidak ada kemudahan kecuali yang Engkau buat mudah. Dan Engkau menjadikan kesedihan (kesulitan), jika Engkau kehendaki pasti akan menjadi mudah."

(HR: Ibnu Hibban)

"(Ingatlah) ketika pemuda-pemuda itu berlindung ke dalam gua lalu mereka berdoa, "Ya Tuhan kami. Berikanlah rahmat kepada kami dari sisi-Mu dan sempurnakanlah petunjuk yang lurus bagi kami dalam urusan kami."

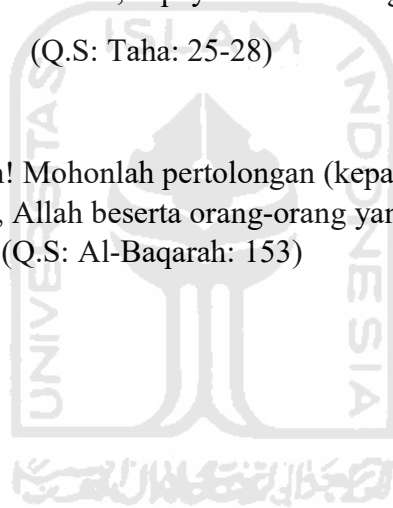
(Q.S: Al-Kahfi :10)

"Ya Tuhanku, lapangkanlah untukku dadaku, dan mudahkanlah untukku segala urusanku, dan lepaskanlah kekakuan dari lidahku, supaya mereka mengerti perkataanku."

(Q.S: Taha: 25-28)

"Wahai orang-orang yang beriman! Mohonlah pertolongan (kepada Allah) dengan sabar dan sholat. Sungguh, Allah beserta orang-orang yang sabar".

(Q.S: Al-Baqarah: 153)



KATA PENGANTAR

Puji syukur saya panjatkan kehadiran Allah SWT, atas limpahan berkat, rahmat dan hidayahnya, sehingga penelitian dengan judul “PEMANFAATAN KLASSTERISASI PADA PERANGKAT RASPBERRY PI UNTUK MELAKUKAN AKUISISI MULTI PERANGKAT” dapat diselesaikan. Shalawat serta salam selalu tercurah pada junjungan Nabi Muhammad SAW yang syafa’atnya senantiasa kita nantikan pada hari kiamat kelak.

Laporan tugas akhir ini disusun dengan tujuan memenuhi syarat mencapai gelar Sarjana Komputer di Universitas Islam Indonesia, serta sebagai tahapan akhir dalam proses studi di Teknik Informatika Universitas Islam Indonesia. Kurangnya pengetahuan dan pengalaman yang dimiliki oleh peneliti, menuntut peneliti untuk meminta dukungan dari berbagai pihak. Oleh karena itu, peneliti memberikan ucapan terimakasih kepada pihak-pihak yang telah memberikan dukungan baik secara moril maupun materiil kepada :

1. Kedua orang tua, Bapak Kendra Hartaya dan Ibu Ani Listiani yang telah memberikan banyak dukungan baik secara moril dan materiil, serta do’a dan motivasi yang senantiasa diberikan.
2. Bapak Fietyata Yudha, S.Kom., M.Kom. selaku Dosen Pembimbing yang telah meluangkan waktu untuk memberikan bimbingan, pengarahan dan masukan demi terselesainya tugas akhir ini.
3. Seluruh Dosen Teknik Informatika yang telah memberikan ilmu dan pengetahuan kepada peneliti.
4. Rekan-rekan Informatika angkatan 2016 yang selalu memberikan motivasi dan dukungan.
5. Dan semua pihak yang tidak bisa disebutkan satu per satu dan telah memantu terlaksananya tugas akhir ini.

Semua bantuan yang diberikan semoga mendapatkan balasan dari Allah SWT. Pada penelitian ini diharapkan mendapatkan kritik dan saran guna meningkatkan untuk mencapai kesempurnaan. Semoga tugas akhir ini dapat bermanfaat bagi pembaca, khususnya pada bidang Jaringan dan Keamanan Komputer.

Yogyakarta, Juli 2020



(Aulia Rahman Ganimeda)



SARI

Cyber crime atau kejahatan dunia maya merupakan kejahatan dengan jaringan komputer sebagai alat, sasaran dan tempat terjadinya. Dari tahun ke tahun *cyber crime* terus mengalami peningkatan dengan berbagai perkembangan kasus. Data dari Kaspersky mengatakan bahwa Indonesia menduduki peringkat 6 berdasarkan negara dengan target serangan melalui *malware cellular*. Seluruh aktivitas *cyber crime* tentu akan meninggalkan barang bukti. Barang bukti tersebut akan digunakan dalam proses penyelidikan dan forensika *digital*. Namun untuk mendapatkan barang bukti tersebut dibutuhkan sebuah sistem yang mampu mengakuisisi seluruh data yang ada pada barang bukti.

Berdasarkan uraian diatas, peneliti membuat sebuah sistem klaster menggunakan perangkat Raspberry Pi untuk melakukan akuisisi terhadap banyak barang bukti yang didapatkan. Pembuatan sistem klaster diharapkan mampu mengakuisisi banyak barang bukti dengan waktu yang lebih cepat. Pengecekan *hostname* dilakukan untuk memastikan seluruh *node* pada sistem klaster telah terhubung dengan baik. Selain membuat sistem klaster, penelitian ini juga membuat sistem monitor menggunakan Cacti. Sistem ini ditujukan untuk mengetahui beban sumber daya ketika digunakan untuk mengakuisisi banyak barang bukti digital. Sehingga dapat membantu dalam menganalisis hasil pengembangan sistem klaster.

Kata kunci: Akuisisi, Forensika digital, Klaster, Raspberry Pi

GLOSARIUM

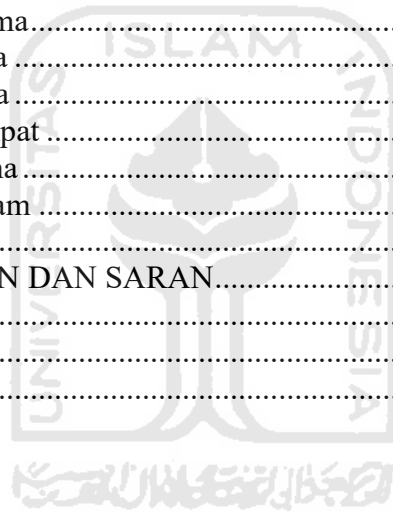
Idle	kondisi dimana sebuah perangkat sedang tidak digunakan atau menganggur.
Memory	tempat penyimpanan data sementara sebelum nantinya diproses ke dalam CPU.
CPU	unit pemrosesan dalam sebuah komputer.
Router	perangkat pengarah lalu lintas data internet
Traffic	lalu lintas data pada jaringan internet
Database	kumpulan data yang tersimpan di komputer secara terstruktur



DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI.....	ix
GLOSARIUM.....	x
DAFTAR ISI	xi
DAFTAR TABEL	xiii
DAFTAR DIAGRAM.....	xiv
DAFTAR GAMBAR.....	xv
BAB I PENDAHULUAN	2
1.1 Latar Belakang Masalah.....	2
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
1.6 Metode Penelitian	4
1.6.1 Identifikasi Kebutuhan	4
1.6.2 Perakitan Sistem	4
1.6.3 Pengujian	4
1.6.4 Analisis Hasil Pengujian	5
1.6.5 Penulisan laporan	5
1.7 Sistematika Penulisan	5
1.7.1 BAB I PENDAHULUAN	5
1.7.2 BAB II LANDASAN TEORI	5
1.7.3 BAB III METODOLOGI PENELITIAN	5
1.7.4 BAB IV HASIL DAN PEMBAHASAN.....	5
1.7.5 BAB V KESIMPULAN DAN SARAN	5
BAB II LANDASAN TEORI.....	6
2.1 Penelitian terkait	6
2.2 Landasan teori	6
2.2.1 Raspberry Pi	6
2.2.2 Forensika <i>Digital</i>	7
2.2.3 Klaster	8
2.2.4 Topologi Star.....	9
2.2.5 Messsage Passing Interface	9
2.2.6 Samba	10
2.2.7 Python.....	10
2.2.8 Internet Protocol version 4 (IPv4).....	10
2.2.9 Mikrotik.....	11
2.2.10 Cacti.....	13
BAB III METODOLOGI PENELITIAN	14
3.1 Identifikasi Kebutuhan	14
3.2 Perakitan Sistem.....	14

3.2.1	Kebutuhan Alat dan Bahan.....	14
3.2.2	Instalasi Jaringan Klaster.....	15
3.2.3	Konfigurasi MPI.....	16
3.2.4	Konfigurasi Cacti	16
3.2.5	Konfigurasi SNMP	17
3.2.6	Konfigurasi SMB	17
3.3	Pengujian.....	17
3.4	Analisis Hasil Pengujian	18
3.5	Pembuatan Laporan.....	18
	BAB IV KONFIGURASI, IMPLEMENTASI DAN PENGUJIAN.....	19
4.1	Konfigurasi <i>Router</i> Mikrotik dan <i>Hostname</i> Raspberry Pi.....	19
4.2	Pengujian Jaringan dan Sistem Klaster.....	21
4.3	Konfigurasi MPI	22
4.4	Konfigurasi Cacti	26
4.5	Konfigurasi SNMP.....	28
4.6	Konfigurasi SMB	29
4.7	30
4.8	Pengujian Sistem Akuisisi	30
4.8.1	Skenario Pertama.....	32
4.8.2	Skenario Kedua	35
4.8.3	Skenario Ketiga	38
4.8.4	Skenario Keempat	41
4.8.5	Skenario Kelima.....	44
4.8.6	Skenario Keenam	47
4.9	Pembahasan.....	50
	BAB V KESIMPULAN DAN SARAN.....	52
5.1	Kesimpulan	52
5.2	Saran.....	52
	DAFTAR PUSTAKA	54



DAFTAR TABEL

Tabel 2.1.1 Rincian Penelitian Terkait	6
Tabel 3.2.1.1 Kebutuhan Alat	14
Tabel 3.2.1.2 Kebutuhan Sistem Raspberry Pi	15
Tabel 3.3.1 Skenario Pengujian	17
Tabel 4.1.1 Daftar Alamat IP	20
Tabel 4.8.1 Tabel Spesifikasi Sistem Independen	31



DAFTAR DIAGRAM

Diagram 4.8.1.1 <i>Idle CPU Utilization</i> Skenario Pertama.....	32
Diagram 4.8.1.2 <i>Free Memory</i> Skenario Pertama	33
Diagram 4.8.1.3 <i>Load Average</i> Skenario Pertama.....	33
Diagram 4.8.1.4 <i>Process</i> Skenario Pertama.....	34
Diagram 4.8.1.5 <i>Data Traffic</i> Skenario Pertama.....	34
Diagram 4.8.2.1 <i>Idle CPU Utilization</i> Skenario Kedua.....	35
Diagram 4.8.2.2 <i>Free Memory</i> Skenario Kedua	36
Diagram 4.8.2.3 <i>Load Average</i> Skenario Kedua	36
Diagram 4.8.2.4 <i>Process</i> Skenario Kedua	37
Diagram 4.8.2.5 <i>Data Traffic</i> Skenario Kedua	37
Diagram 4.8.3.1 <i>Idle CPU Utilization</i> Skenario Ketiga	38
Diagram 4.8.3.2 <i>Free Memory</i> Skenario Ketiga.....	39
Diagram 4.8.3.3 <i>Load Average</i> Skenario Ketiga	39
Diagram 4.8.3.4 <i>Process</i> Skenario Ketiga	40
Diagram 4.8.3.5 <i>Data Traffic</i> Skenario Ketiga	40
Diagram 4.8.4.1 <i>Idle CPU Utilization</i> Skenario keempat	41
Diagram 4.8.4.2 <i>Free Memory</i> Skenario Keempat	42
Diagram 4.8.4.3 <i>Load Average</i> Skenario Keempat	43
Diagram 4.8.4.4 <i>Processes</i> Skenario Keempat.....	43
Diagram 4.8.4.5 <i>Data Traffic</i> Skenario Keempat	44
Diagram 4.8.5.1 <i>Idle CPU Utilization</i> Skenario Kelima	45
Diagram 4.8.5.2 <i>Free Memory</i> Skenario Kelima.....	45
Diagram 4.8.5.3 <i>Load Average</i> Skenario Kelima	46
Diagram 4.8.5.4 <i>Processes</i> Skenario Kelima.....	46
Diagram 4.8.5.5 <i>Data Traffic</i> Skenario Kelima	47
Diagram 4.8.6.1 <i>Idle CPU Utilization</i> Skenario Keenam.....	48
Diagram 4.8.6.2 <i>Free Memory</i> Skenario Keenam	48
Diagram 4.8.6.3 <i>Load Average</i> Skenario Keenam	49
Diagram 4.8.6.4 <i>Process</i> Skenario Keenam	49
Diagram 4.8.6.5 <i>Data Traffic</i> Skenario Keenam.....	50

DAFTAR GAMBAR

Gambar 2.2.1.1 Raspberry Pi 3	7
Gambar 2.2.3.1 Topologi Sistem Klaster.....	9
Gambar 2.2.9.1 Mikrotik RB951Ui-2nd.....	12
Gambar 2.2.9.2 Tampilan Winbox	13
Gambar 3.1 Alur Penelitian	14
Gambar 3.2.2.1 Topologi Sistem Klaster.....	16
Gambar 4.1.1 Konfigurasi Dasar Mikrotik.....	19
Gambar 4.1.2 <i>Hostname</i>	20
Gambar 4.2.1 Sistem yang Digunakan	21
Gambar 4.2.2 Hasil Pengujian Sistem Klaster.....	22
Gambar 4.3.1 Konfigurasi MPICH.....	22
Gambar 4.3.2 <i>Path</i> Hasil Instalasi MPI.....	23
Gambar 4.3.3 Konfigurasi Python	23
Gambar 4.3.4 SD Card Kloning.....	24
Gambar 4.3.5 <i>Machinefile</i>	24
Gambar 4.3.6 SSH Pi01	25
Gambar 4.3.7 SSH Pi02.....	25
Gambar 4.3.8 SSH Pi03	25
Gambar 4.3.9 SSH Pi04.....	25
Gambar 4.3.10 Otorisasi SSH Pi01.....	26
Gambar 4.4.1 Konfigurasi MariaDB	26
Gambar 4.4.2 Konfigurasi <i>Timezone</i>	27
Gambar 4.4.3 Tampilan Cacti.....	27
Gambar 4.5.1 Konfigurasi SNMP <i>Client</i>	28
Gambar 4.5.2 Konfigurasi SNMP Mikrotik	28
Gambar 4.5.3 Konfigurasi SNMP Server	29
Gambar 4.6.1 Konfigurasi Samba <i>File Sharing</i>	30
Gambar 4.7.1 Grafik Hasil Monitoring.....	30
Gambar 4.8.1 Perintah Akuisisi	31
Gambar 4.8.2 File Hasil Akuisisi.....	32

Gambar 4.9.1 Hasil Pengujian Penelitian Sebelumnya51



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Kejahatan atau tindak kriminal merupakan salah satu bentuk dari perilaku menyimpang yang selalu ada dan melekat pada tiap bentuk masyarakat. Namun pada era saat ini seiring dengan pesatnya perkembangan pada dunia *digital*, kejahatan yang sebelumnya beraksi pada dunia nyata juga merambah pada dunia *digital*. Negara Indonesia menduduki peringkat 6 berdasarkan negara dengan target serangan melalui *malware cellular* (Kaspersky, 2019). Dalam menjalankan aksinya pelaku menggunakan perangkat *digital*. Dengan menggunakan perangkat *digital* inilah pelaku dapat dengan mudah memodifikasi atau bahkan menghilangkan barang bukti agar sulit ditemukan oleh orang awam. Hal ini memunculkan kajian yang dikenal dengan istilah forensika *digital*. Forensika *digital* merupakan disiplin ilmu yang mempelajari tentang serangkaian kegiatan dengan tujuan menganalisis dan menyelesaikan kasus yang terkait dengan kejahatan dunia maya (Amato et al., 2019).

Salah satu tahapan forensika *digital* yang dilakukan yaitu proses *Imaging*. *Imaging* adalah menggandakan isi dari barang bukti *harddisk* tersebut secara *physical* (sektor per sektor atau *bit-stream copy*) sehingga hasil *imaging* akan sama persis dengan barang bukti secara *physical*. Derajat kesamaan ini dapat dipastikan melalui proses *hashing* yang diterapkan pada keduanya (Saputra & Widiyasono, 2017). Banyaknya barang yang menjadi bukti kejahatan tentunya akan membuat proses *imaging* memakan waktu yang cukup lama. Untuk menunjang proses *imaging* diperlukan sebuah perangkat yang cukup *powerfull*. Untuk itu pada penelitian ini kami mencoba merancang sebuah alat dengan menggunakan Raspberry Pi. Raspberry Pi (RPi) adalah komputer papan tunggal berukuran kecil (berukuran kartu kredit) yang murah dan mampu menjalankan linux dan sistem operasi ringan lainnya yang berjalan pada *processor* ARM (Kiepert, 2013). Raspberry Pi digunakan dalam penelitian karena merupakan perangkat yang memiliki harga cukup murah dan dengan performa yang cukup tinggi sehingga ini menjadi faktor penting yang mendorong untuk menggunakan perangkat Raspberry Pi dalam proyek ini.

Raspberry Pi tersebut nantinya akan disusun menjadi satu kesatuan untuk menjalankan komputasi secara bersamaan. Istilah ini biasa disebut dengan klusterisasi. Klusterisasi adalah jenis sistem pemrosesan paralel atau terdistribusi. Terdiri dari kumpulan komputer yang berdiri sendiri yang saling berhubungan yang bekerja bersama sebagai satu sumber daya komputasi (López & Baydal, 2018). Tujuan dengan menggunakan teknologi klusterisasi adalah menghemat biaya untuk mendapatkan kinerja tinggi. Adanya fleksibilitas konfigurasi untuk memenuhi berbagai kebutuhan (López & Baydal, 2017).

Dalam perancangan kluster tersebut nantinya akan menggunakan standar *interface* yang sangat familiar yakni MPI (*Message Passing Interface*). MPI digunakan karena mendukung heterogenitas perangkat, sehingga dalam merancang kluster dapat menggunakan Raspberry Pi dengan tipe yang berbeda.

Selama menjalankan proses *imaging* nantinya perangkat pada kluster akan dimonitoring sumber daya yang digunakan, menggunakan sebuah *tool* bernama cacti. *Tool* ini digunakan dengan tujuan dapat dengan mudah diamati karena sudah menggunakan GUI dan adanya grafik yang menandakan seberapa besar penggunaan sumber daya selama proses *imaging*.

1.2 Rumusan Masalah

Berdasarkan latar belakang dapat dirumuskan permasalahan akan diselesaikan dalam penelitian ini adalah

1. Bagaimana merancang sebuah sistem kluster dengan perangkat Raspberry Pi untuk meningkatkan kecepatan proses akuisisi?

1.3 Batasan Masalah

Pembahasan pada penelitian ini dibatasi pada:

1. Sistem yang dibuat terdiri dari 2 bagian, sistem kluster dan sistem untuk monitoring.
2. Sistem kluster untuk akuisisi terdiri dari 3 *node* Raspberry Pi.
3. Pembuatan kluster menggunakan MPI.

1.4 Tujuan Penelitian

Sesuai dengan masalah yang telah dirumuskan, maka tujuan dari penelitian ini adalah:

1. Merancang sebuah sistem klaster menggunakan perangkat Raspberry Pi dengan standar *interface Message Passing Interface* (MPI).

1.5 Manfaat Penelitian

Dengan adanya penelitian ini diharapkan dapat memberikan manfaat antara lain :

1. Meningkatkan efisiensi waktu dibandingkan perangkat sebelumnya dengan banyaknya barang bukti.
2. Meningkatkan efektivitas kerja dengan menggunakan satu Raspberry Pi sebagai *controller*.

1.6 Metode Penelitian

1.6.1 Identifikasi Kebutuhan

Identifikasi kebutuhan dilakukan untuk menentukan kebutuhan dari sistem yang nantinya akan dibangun. Sehingga sistem nantinya dapat menjalankan fungsinya dengan tepat.

1.6.2 Perakitan Sistem

Perakitan sistem dilakukan dengan dua tahapan inti, yaitu perancangan sistem dan konfigurasi sistem. Pada perancangan sistem, semua perangkat keras yang dibutuhkan dikumpulkan dan disusun sedemikian rupa sehingga menjadi sebuah sistem yang siap dikonfigurasi. Pada tahapan konfigurasi sistem, semua perangkat lunak yang dibutuhkan dalam membangun sistem dilakukan instalasi dan dikonfigurasi.

1.6.3 Pengujian

Pengujian dilakukan dengan dua tahap, tahap pertama pengujian dilakukan untuk memastikan bahwa seluruh konfigurasi yang dilakukan untuk membangun sistem telah berjalan seperti yang diharapkan. Pengujian tahap kedua dilakukan sesuai dengan skenario pengujian yang telah ditetapkan oleh peneliti hingga mendapatkan data yang nantinya akan digunakan sebagai bahan untuk dianalisis.

1.6.4 Analisis Hasil Pengujian

Setelah mendapatkan data dari hasil pengujian tahap kedua, peneliti menganalisis hasil pengujian tersebut. Hasil analisis nantinya akan digunakan oleh peneliti sebagai data pembuatan laporan.

1.6.5 Penulisan laporan

Pada tahap terakhir penelitian, peneliti menyusun laporan, menarik kesimpulan berdasarkan rumusan masalah penelitian dan memberi saran sebagai dasar penelitian selanjutnya.

1.7 Sistematika Penulisan

1.7.1 BAB I PENDAHULUAN

Bab ini membahas tentang hal-hal yang mendasari adanya penelitian ini. Bab ini terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, serta metode penelitian.

1.7.2 BAB II LANDASAN TEORI

Bab ini membahas landasan teori yang digunakan sebagai dasar penelitian untuk membangun sebuah sistem kluster dengan perangkat Raspberry Pi

1.7.3 BAB III METODOLOGI PENELITIAN

Pada bab ini memuat tentang analisis sistem dan perancangan alat yang digunakan untuk membangun sistem kluster Raspberry Pi. Selain itu pada bab ini membahas tentang konfigurasi dalam membangun sistem kluster Raspberry Pi.

1.7.4 BAB IV HASIL DAN PEMBAHASAN

Bab ini membahas tentang skenario pengujian dan grafik pengujian hasil monitoring dengan cacti dan hasil dari grafik akan dikaji yang nantinya digunakan untuk menarik kesimpulan dan saran

1.7.5 BAB V KESIMPULAN DAN SARAN

Pada bab ini menyimpulkan apa yang sudah didapat pada penelitian ini dan memberi saran untuk penelitian yang akan dilakukan selanjutnya.

BAB II

LANDASAN TEORI

2.1 Penelitian terkait

Penelitian ini berdasar pada dua penelitian sebelumnya terkait penggunaan *single board computer* untuk melakukan akuisisi forensika *digital*. (Olson & Shashidhar, 2016) melakukan penelitian dengan membuat alat duplikator untuk keperluan forensik dengan *budget* yang murah, dibawah \$100. (Maulida, 2018) melakukan penelitian dengan merancang aplikasi *forensic imaging* berbasis GUI. Dua penelitian sebelumnya dirangkum ke dalam sebuah tabel berikut

Tabel 2.1.1 Rincian Penelitian Terkait

No	Judul	Penulis	Pencapaian
1	<i>Forensic Imaging Application Using Raspberry Pi</i>	Razan Maulida	Merancang sebuah aplikasi <i>forensic imaging</i> berbasis GUI menggunakan <i>single board computer</i> Raspberry Pi dan diuji menggunakan <i>flash drive</i> dengan kapasitas yang sama namun dengan merk yang berbeda
2	<i>Low Budget Forensic Drive Imaging Using Arm Based Single Board Computer</i>	Eric Olson dan Narasimha Shashidhar	Menggunakan <i>single board computer</i> untuk melakukan kegiatan forensik dengan harga yang murah. Karena alat duplikator yang ada dipasaran terlalu mahal, tidak cukup hanya dengan mini-pc, tablet atau laptop yang masih berharga sekitar \$200. Karena menurutnya angka \$200 masih tergolong tinggi

2.2 Landasan teori

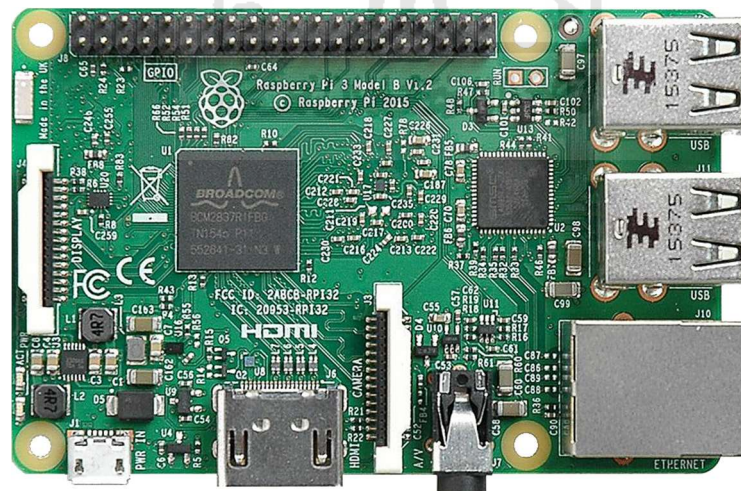
2.2.1 Raspberry Pi

Raspberry Pi atau yang biasa disebut dengan Raspi merupakan komputer papan tunggal (*Single Board Computer*) yang berukuran kecil atau sebesar kartu kredit. Raspberry Pi dikembangkan oleh yayasan nirlaba dari Inggris yang bertujuan untuk mempromosikan studi

dasar ilmu komputer di sekolah sekolah (Bhaskar, 2015). Berikut spesifikasi Raspberry Pi 3 yang digunakan pada penelitian ini :

- Soc: Broadcom BCM2837
- CPU: 4X ARM Cortex-A53, 1.2GHz
- GPU: Broadcom VideoCore IV
- RAM: 1GB LPDDR2 (900 MHz)
- *Networking*: 10/100 Ethernet, 2.4 GHz 802.11 n wireless
- Bluetooth: Bluetooth 4.1 *Classic*, Bluetooth *Low Energy*
- *Storage*: microSD
- GPIO: 40-pin *header*, *populated*
- *Ports*: HDMI, 3.5 mm *analogue* audio-video *jack*, 4x USB 2.0, Ethernet, *Camera Serial Interface* (CSI), *Display Serial Interface* (DSI).

Raspberry Pi umumnya digunakan untuk berbagai *project*, salah satunya yaitu *Internet of Things* (IoT). Raspberry Pi menjalankan sistem operasi Raspbian. Raspbian adalah sistem operasi berbasis Debian yang dioptimalkan untuk perangkat keras Raspberry Pi yang diluncurkan pada Juli 2012 (Sari et al., 2015). Dengan demikian Raspberry Pi dapat menjalankan *library* yang ada di Linux.



Gambar 2.2.1.1 Raspberry Pi 3

2.2.2 Forensika Digital

Forensika *digital* adalah ilmu memperoleh, mengambil, melestarikan dan penyajian data yang telah diproses secara elektronik dan disimpan pada media komputer (Hartanto et al.,

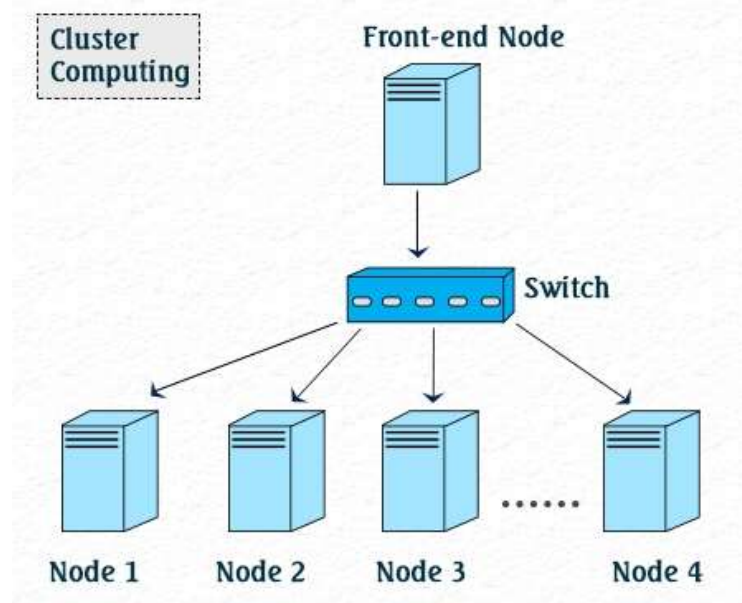
2011). Dalam forensik digital terdapat barang bukti dapat dibagi menjadi dua, barang bukti elektronik dan barang bukti *digital*. Barang bukti elektronik bersifat fisik dan dapat dikenali secara visual seperti, *hard drive, flash drive, router*, dll. Barang bukti digital yaitu barang bukti yang bersifat *digital* yang dapat diekstrak atau di-*recover* dari barang bukti elektronik seperti *logical file, lost file, encrypted file*, dll.

2.2.3 Klaster

Klaster adalah jenis sistem pemrosesan paralel atau terdistribusi. Terdiri dari kumpulan komputer yang berdiri sendiri yang saling berhubungan yang bekerja sama sebagai satu sumber daya (López & Baydal, 2018). Di dalam sebuah klaster terdapat beberapa komputer yang disebut sebagai *node*. Pada masing-masing *node* ini saling terintegrasi satu sama lain dengan sebuah jaringan. Jaringan yang menghubungkan antar *node* tersebut harus memiliki *bandwidth* yang tinggi dan *latency* yang sangat rendah agar mendapatkan performa yang optimal pada klaster tersebut. Tiap *node* memiliki sistem operasi dan sistem I/O. Jika *node* tersebut memiliki spesifikasi perangkat keras dan perangkat lunak yang identik maka dapat disebut sebagai klaster yang homogen.

Dalam membangun sebuah sistem klaster diperlukan dua komponen, *hardware* dan *software*. Pada *hardware* diperlukan mesin komputasi atau *node* dan jaringan interkoneksi yang menghubungkan antar *node*. Pada *software* diperlukan sebuah *software* untuk pengembangan aplikasi paralel dan aplikasi yang digunakan untuk memonitor proses yang berjalan di dalam klaster.

Klaster digunakan untuk melakukan komputasi dengan performa yang tinggi. Komputasi dengan performa tinggi tersebut dicapai dengan melakukan komputasi secara paralel yaitu menyelesaikan sebuah permasalahan komputasi dengan membagi permasalahan untuk diselesaikan secara simultan oleh masing masing *node* di dalam klaster (Widyaputra, 2008).



Gambar 2.2.3.1 Topologi Sistem Klaster

2.2.4 Topologi Star

Topologi *star* merupakan sebuah topologi dimana setiap *client / user* terhubung langsung ke *switch/hub* secara *point to point*. *Switch/hub* dalam topologi ini bertujuan mengatur lalu lintas pada jaringan dan juga bertindak sebagai *repeater*. Topologi ini sangat mudah diaplikasikan, selain itu jika ada *client/user* baru atau yang tidak terkoneksi maka jaringan tidak mengganggu. Tetapi jika *switch/hub* terjadi malfungsi maka seluruh jaringan akan terganggu dan tidak bisa terkoneksi.

2.2.5 Message Passing Interface

Message Passing Interface (MPI) adalah spesifikasi *library* yang digunakan untuk melewatkan pesan. Dengan menambahkan model penyimpanan pesan paralel dari komputasi, dimana proses dengan ruang alamat terpisah dan berkomunikasi dengan mengirim dan menerima pesan (Muttill et al., 2007). Dengan mengimplementasikan *message passing interface*, setiap proses yang terlibat dalam penyelesaian masalah secara paralel dapat berkomunikasi dengan mengirim pesan (*message passing*). *Message passing interface* berisi sekumpulan *Application Programming Interface* (API) yang dapat digunakan oleh pengembang untuk dapat diterapkan pada bahasa pemrograman apapun (*language independent*).

2.2.6 Samba

Samba adalah aplikasi yang berjalan di Sistem Operasi Linux, Unix maupun BSD dan dapat bertindak sebagai layanan berbagi berkas (*file sharing*) maupun layanan berbagi alat cetak (*Printer Sharing*) (Smb et al., 2000). Samba memiliki dua program yang dijalankan di *background*, yakni *Service Message Block Daemon* (SMBD) yang bertindak sebagai program yang menghasilkan proses baru untuk klien yang aktif dan *NetBIOS Name Block Daemon* (NMBD) yang bertindak sebagai konversi nama komputer menjadi alamat IP. Samba memiliki program bantu sehingga Sistem Operasi Linux, Unix maupun BSD dapat mengakses sumber daya yang ada pada Sistem Operasi Windows.

2.2.7 Python

Python merupakan bahasa pemrograman modern yang mendukung orientasi objek. Python adalah bahasa *scripting* yang pertama dan utama, serta dapat dikompilasi ke dalam biner yang bisa dibaca oleh komputer (Kelly, 2016). Python pertama kali dikembangkan tahun 1990 oleh Guido van Rossum. Selain pemrograman modern, Python juga dapat dijalankan banyak *platform* dan mudah digunakan, sehingga banyak orang menggunakan bahasa pemrograman ini.

2.2.8 Internet Protocol version 4 (IPv4)

Internet Protocol versi 4 (IPv4) merupakan sistem pengalamatan pada jaringan yang direpresentasikan berupa kombinasi angka sebanyak empat deret, dimana setiap deretnya terdiri antara bilangan 0 sampai 255 (Levin & Schmidt, 2014). IP memiliki 2 jenis berdasarkan penggunaannya yakni, IP *Public* yang digunakan untuk menghubungkan ke jaringan internet dan IP *Private* yang digunakan sebagai jaringan lokal.

Pada IPv4 memiliki beberapa kelas yakni, kelas A, B, C, D dan E. Tiap kelas memiliki besar kecil *Network* yang terpasang, *Network* tersebut meliputi *Network ID* dan *Host ID*. Umumnya hanya kelas A, B dan C yang digunakan dalam sebuah jaringan, sedangkan kelas D dan E digunakan untuk keperluan *multicasting* maupun eksperimental.

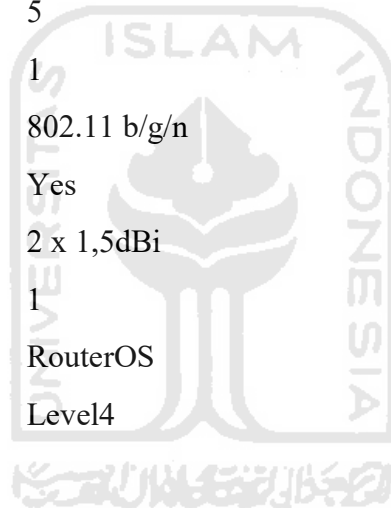
IP kelas A menggunakan *octet* pertama untuk menunjukkan *Network ID* dan tiga *octet* terakhir sebagai *Host ID*. Pada IP kelas A memiliki IP *private* pada alamat IP 10.0.0.0/8. IP kelas B menggunakan *octet* pertama dan kedua untuk menunjukkan *Network ID* dan dua *octet* terakhir sebagai *Host ID*. IP kelas B menggunakan alamat IP 172.16.0.0 – 172.168.31.255 sebagai IP *privatenya*. Dan IP kelas C menggunakan tiga *octet* pertama untuk menunjukkan

Network ID dan satu *octet* terakhir untuk *Host ID*. Pada kelas C menggunakan alamat IP 192.168.0.0/16 sebagai IP *privatenya*

2.2.9 Mikrotik

Penelitian ini menggunakan perangkat *router* Mikrotik tipe RB951ui 2nd. *Router* ini memiliki spesifikasi sebagai berikut

- *Product Code* RB951Ui-2nD
- *Architecture* MIPS-BE
- CPU QCA9531-BL3A-R 650MHz
- *Main Storage/NAND* 16MB
- RAM 64MB
- LAN Ports 5
- *Integrated Wireless* 1
- *Wireless Standarts* 802.11 b/g/n
- *Integrated Antenna* Yes
- *Antenna Gain* 2 x 1,5dBi
- USB 1
- *Operating System* RouterOS
- *RouterOS License* Level4





Gambar 2.2.9.1 Mikrotik RB951Ui-2nd

Mikrotik

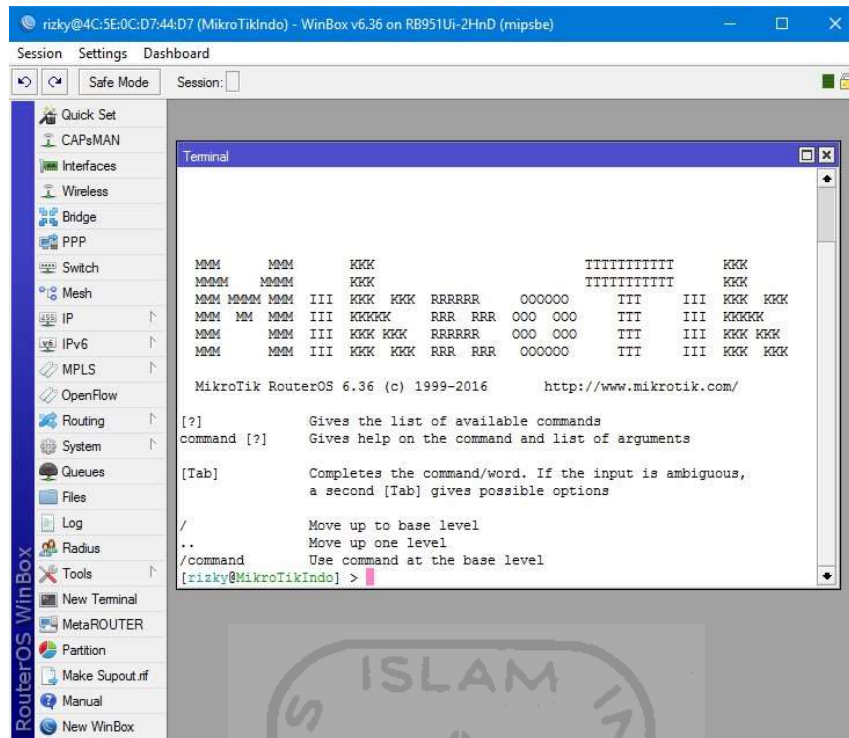
Mikrotik merupakan perusahaan baik hardware maupun software yang mengembangkan perangkat jaringan, mulai dari *router*, *switch* hingga *access point*. Perangkat Mikrotik banyak digunakan dalam membangun sebuah jaringan, karena fitur yang disediakan sangatlah banyak. Salah satu fitur tersebut adalah *Samba file sharing* yang digunakan pada penelitian ini.

Router OS

Router OS adalah sistem operasi yang ditanamkan pada perangkat Mikrotik atau dapat *diinstall* pada perangkat komputer sehingga perangkat tersebut dapat digunakan untuk keperluan jaringan. Router OS memiliki berbagai tingkat lisensi mulai dari level 0 sampai 6 sesuai dengan kebutuhan dalam membangun jaringan. Di dalam Router OS terdapat banyak *plug in*, dimana setiap *plug in* tersebut berisi fitur fitur yang disediakan oleh Mikrotik.

Winbox

Dalam melakukan *remote* terhadap perangkat Mikrotik banyak metode yang dilakukan, salah satunya dengan *tool* Winbox. *tool* ini dikembangkan secara resmi oleh Mikrotik dengan tujuan *remote* perangkat Mikrotik ke dalam mode GUI.



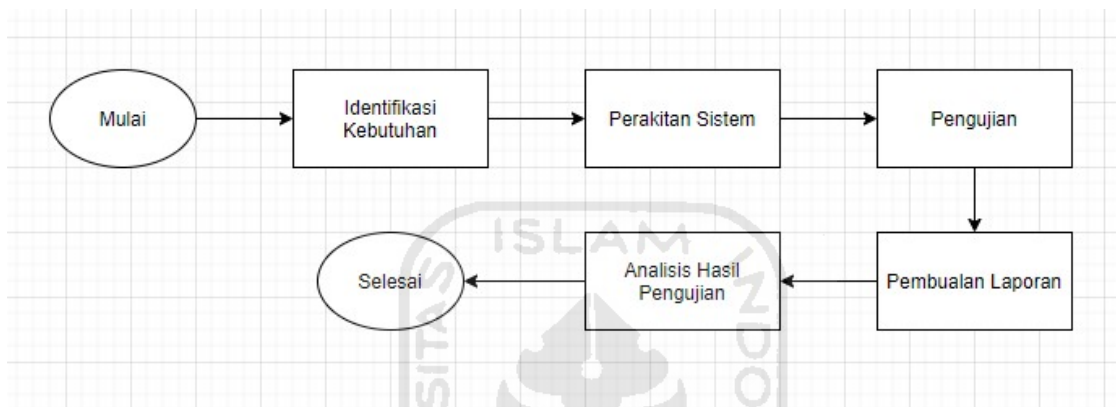
Gambar 2.2.9.2 Tampilan Winbox

2.2.10 Cacti

Cacti merupakan *tool* monitoring dalam bentuk grafis (*graphing*) untuk melihat aktivitas yang terjadi dalam sebuah sistem seperti, *traffic* data, sumber daya sistem, dll. Cacti merupakan pengembangan dari RRD *tool* untuk menggambarkan sebuah grafik dengan menggunakan My Sql sebagai basis datanya. Cacti menggunakan protokol *Simpel Network Management Protocol* (SNMP) sebagai media komunikasi, sehingga semua fungsi yang ada pada SNMP bisa digunakan pada Cacti.

BAB III METODOLOGI PENELITIAN

Dalam penelitian "Pemanfaatan Klasterisasi pada Perangkat Raspberry Pi untuk Melakukan Akuisisi Multi Perangkat", peneliti menggunakan alur penelitian yang menggambarkan tahapan selama penelitian ini. Alur penelitian yang digunakan sebagai berikut



Gambar 3.1 Alur Penelitian

3.1 Identifikasi Kebutuhan

Dalam penelitian ini, peneliti menggunakan MPI sebagai media pembuatan sistem kluster Raspberry Pi. Untuk mengatasi perbedaan performa antara satu sistem dengan sistem lain (*bottleneck*) maka peneliti harus menggunakan sistem yang identik, mulai dari perangkat keras hingga perangkat lunak yang dipakai. Sehingga nantinya hasil keluaran dari sistem kluster ini diharapkan tidak memiliki perbedaan performa yang signifikan.

3.2 Perakitan Sistem

3.2.1 Kebutuhan Alat dan Bahan

Untuk keperluan membangun sistem kluster ini dibutuhkan beberapa alat dan bahan. Alat dan bahan ini terdiri dari dua komponen yaitu *hardware* dan *software*. Berikut daftar alat dan bahan yang diperlukan untuk kebutuhan membangun sistem kluster ini

Tabel 3.2.1.1 Kebutuhan Alat

Nama	Jumlah

Raspberry Pi 3 Model B	4 Buah
Micro SD	4 Buah
Kabel Micro USB	4 Buah
Flash drive	4 buah
Spacer	20 Buah
Kabel Lan	2 Meter
Konektor RJ 45	12 Buah
Crimping Tool	1 Set
Mikrotik RB951ui-2nd	1 Buah
Power Supply	1 Set
Card Reader	1 Buah

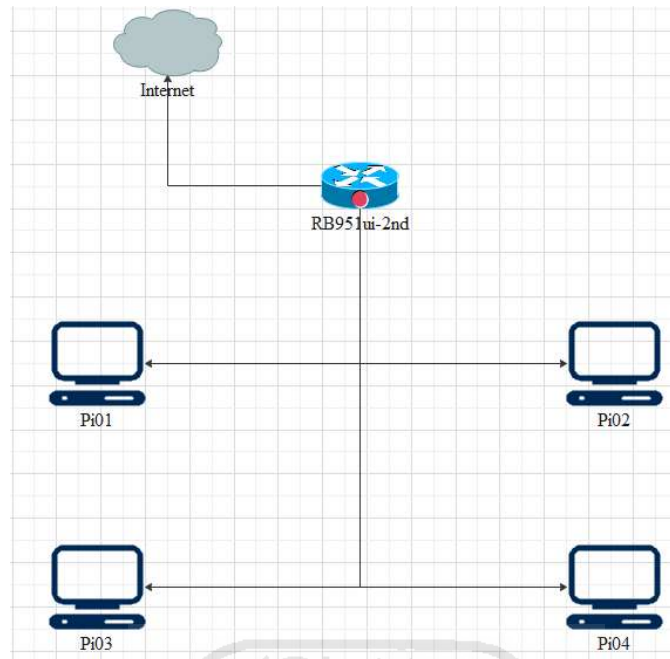
Tabel 3.3.2.1.2 Kebutuhan Sistem Raspberry Pi

Nama
Raspbian Buster
BalenaEtcher

Semua alat dan bahan tersebut dirancang dan dilakukan instalasi hingga siap digunakan untuk membuat sistem klaster. Mulai dari instalasi Sistem Operasi Raspbian ke dalam Micro SD dengan bantuan aplikasi BalenaEthcer. Hingga *crimping* kabel lan sesuai dengan ukuran, agar tidak terlalu panjang atau pendek dan juga penyiapan *power supply*. Dan kemudian sistem siap dirancang berdasarkan topologi jaringan yang telah dibuat

3.2.2 Instalasi Jaringan Klaster

Topologi yang digunakan yaitu topologi *star*. Topologi ini dipilih karena instalasinya yang mudah. Selain itu topologi ini dipilih karena semua perangkat Raspberry Pi membutuhkan satu *broadcast domain* agar nantinya sistem klaster dapat berjalan dengan lancar.



Gambar 3.2.2.1 Topologi Sistem Kluster

Pada penelitian ini peneliti menggunakan Mikrotik sebagai router karena salah satu fiturnya dapat dimanfaatkan pada penelitian ini, yaitu Samba.

3.2.3 Konfigurasi MPI

Untuk menjalankan sistem kluster pada penelitian ini menggunakan metode *Message Passing Interface* (MPI). Agar MPI dapat dijalankan maka terlebih dahulu melakukan instalasi MPICH, MPICH ini merupakan *library* dari metode MPI. Berikutnya melakukan instalasi Python, untuk menjalankan perintah akuisisi yang berbasis bahasa Python agar nantinya dapat berjalan dengan lancar.

Setelah kebutuhan utama penelitian ini *terinstall* dengan baik, untuk mengefisienkan kerja maka dilakukan kloning semua isi dari Micro SD, sehingga sistem operasi dan konfigurasi di dalamnya akan sama persis. Hasil kloning tersebut akan digunakan untuk membuat *node-node* selanjutnya. Setelah mengkloning Micro SD sesuai dengan jumlah *node*, maka pada *master node* diberikan inisiasi berupa alamat IP untuk semua sistem yang akan masuk ke dalam kluster.

3.2.4 Konfigurasi Cacti

Sistem monitoring pada penelitian ini menggunakan *tool* Cacti. Agar Cacti dapat berjalan dengan baik perlu melakukan instalasi Apache sebagai *web server*, karena Cacti menggunakan

antarmuka web. MariaDB sebagai basis data yang nantinya digunakan untuk menyimpan data hasil monitoring Cacti. RRD *-tool* karena Cacti merupakan pengembangan dari RRD. Cacti sebagai *tool* utamanya. Dan SNMP sebagai protokol komunikasi Cacti.

3.2.5 Konfigurasi SNMP

Sebagai protokol komunikasi, Cacti menggunakan SNMP. Konfigurasi ini dilakukan pada dua sisi. Pertama pada sisi *client (node)* yang nantinya akan dimonitor. Dan kedua pada sisi *server (master node)* yang berisi hasil instalasi Cacti, yang nantinya akan digunakan untuk monitoring *node*.

3.2.6 Konfigurasi SMB

Tujuan dari *file* hasil akuisisi nantinya akan disimpan dalam *flash drive* yang dipasangkan pada *router* Mikrotik. Untuk menghubungkan antara *flash drive* di Mikrotik dan *node* pada kluster, menggunakan fitur Samba yang ada pada *router* Mikrotik. Untuk mengaktifkan fitur tersebut perlu mengetikkan beberapa perintah melalui *terminal* pada Winbox Mikrotik.

3.3 Pengujian

Pengujian dilakukan dengan enam skenario pengujian. Setiap menjalankan skenario pengujian peneliti akan mengamati sumber daya yang digunakan selama proses pengujian dengan Cacti. Parameter yang akan diamati yakni, CPU, *memory*, *load*, *process*, dan besaran data yang ditransmisikan.

Tabel 3.3.1 Skenario Pengujian

Skenario	<i>Flash drive</i>	Raspberry Pi
1	3 Buah	3 Buah
2	2 Buah	2 Buah
3	1 Buah	1 Buah
4	2 Buah	1 Buah
5	3 Buah	1 Buah
6	3 Buah	2 Buah

- Skenario pertama masing masing Raspberry Pi dipasangkan dengan satu buah *flash drive* dan total ada tiga *node* yang dijalankan.

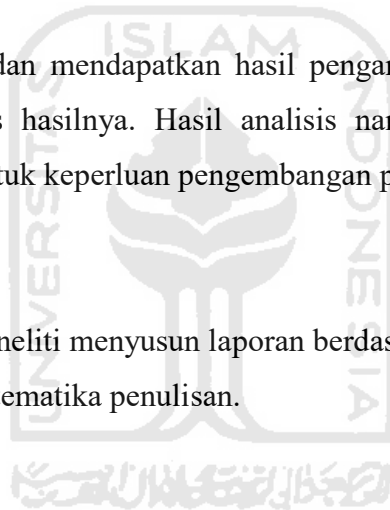
- Skenario kedua tiap Raspberry Pi dipasangkan dengan satu buah *flash drive* dan total ada dua *node* yang dijalankan.
- Skenario ketiga satu buah Raspberry Pi dipasangkan dengan satu buah *flash drive* dan hanya satu *node* yang dijalankan.
- Skenario keempat satu buah Raspberry Pi dipasangkan dengan dua buah *flash drive* dan menggunakan satu *node*.
- Skenario kelima tiga buah *flash drive* dipasangkan pada satu buah Raspberry Pi dan menjalankan satu buah *node*.
- Skenario keenam menggunakan dua Raspberry Pi dan dipasangkan dengan tiga buah *flash drive* tetapi salah satu Raspberry Pi dipasangkan dua buah *flash drive* dengan menjalankan total dua buah *node*.

3.4 Analisis Hasil Pengujian

Setelah pengujian dilakukan dan mendapatkan hasil pengamatan sumber daya dengan Cacti, peneliti akan menganalisis hasilnya. Hasil analisis nantinya akan ditarik sebuah kesimpulan dan diberikan saran untuk keperluan pengembangan penelitian kedepannya.

3.5 Pembuatan Laporan

Pada tahap akhir penelitian, peneliti menyusun laporan berdasarkan rumusan masalah dan tujuan penelitian sesuai dengan sistematika penulisan.



BAB IV

KONFIGURASI, IMPLEMENTASI DAN PENGUJIAN

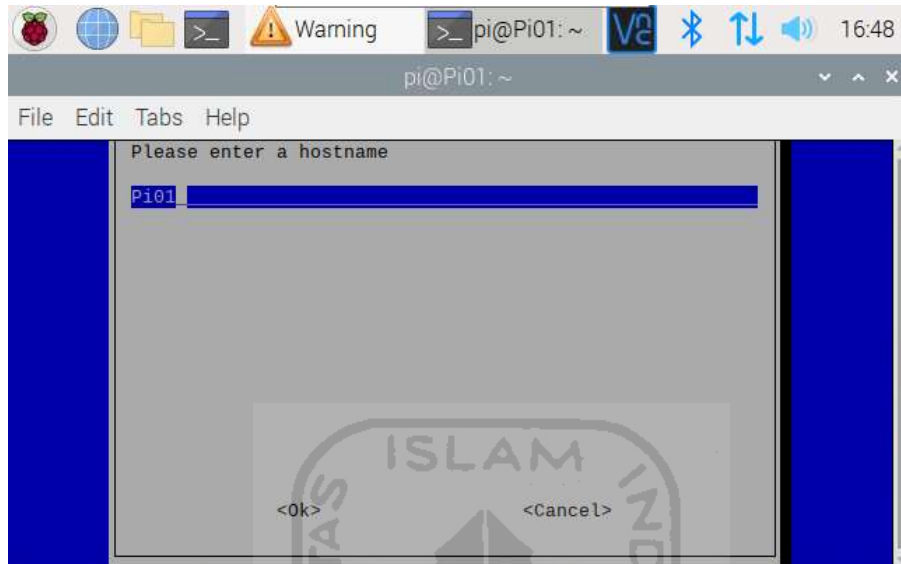
4.1 Konfigurasi *Router* Mikrotik dan *Hostname* Raspberry Pi

Karena penggunaan Mikrotik sebagai router maka diperlukan konfigurasi awal hingga *router* ini mampu terkoneksi ke internet dan memberikan alamat IP ke Raspberry Pi yang terhubung pada *router* ini. Untuk melakukan konfigurasi *router* Mikrotik, *remote router* menggunakan aplikasi Winbox dan pilih “*terminal*” yang terdapat pada menu Winbox.

```
/interface bridge
add name=bridge1
/interface bridge port
add bridge=bridge1 interface=ether2
add bridge=bridge1 interface=ether3
add bridge=bridge1 interface=ether4
add bridge=bridge1 interface=ether5
/ip address
add address=192.168.11.1/24 interface=bridge1 network=192.168.11.0
/ip pool
add name=dhcp_pool2 ranges=192.168.11.2-192.168.11.254
/ip dhcp-server network
add address=192.168.11.0/24 gateway=192.168.11.1
/ip dhcp-server
add address-pool=dhcp_pool2 disabled=no interface=bridge1 name=dhcp1
/ip dhcp-client
add dhcp-options=hostname,clientid disabled=no interface=ether1
/ip firewall nat
add action=masquerade chain=srcnat out-interface=ether1
```

Gambar 4.1.1 Konfigurasi Dasar Mikrotik

Setelah konfigurasi Mikrotik, kemudian menghubungkan Raspberry Pi dengan *router* Mikrotik. Dan ubah *hostname* pada Raspberry Pi agar mudah diidentifikasi antara *node* dengan *master node*.



Gambar 4.1.2 *Hostname*

Setelah *hostname* Raspberry Pi diubah kemudian didapatkan Alamat IP Raspberry Pi sebagai berikut

Tabel 4.1.1 Daftar Alamat IP

<i>Hostname</i>	<i>IP Address</i>
Pi01	192.168.11.254
Pi02	192.168.11.251
Pi03	192.168.11.252
Pi04	192.168.11.253

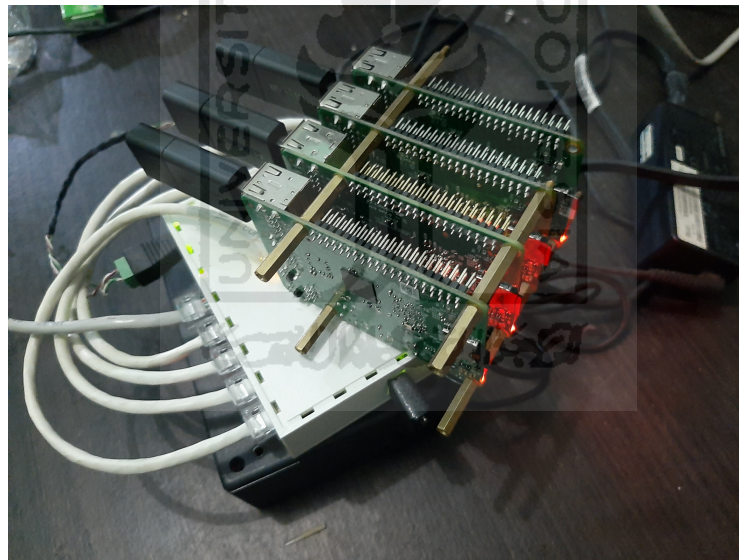
Pi01 nantinya akan digunakan sebagai *master node* dan sistem monitoring. Sedangkan Pi02 sampai Pi04 akan digunakan untuk *node* klaster yang nantinya akan mengeksekusi perintah akuisisi dari Pi01 dan dimonitoring.

4.2 Pengujian Jaringan dan Sistem Kluster

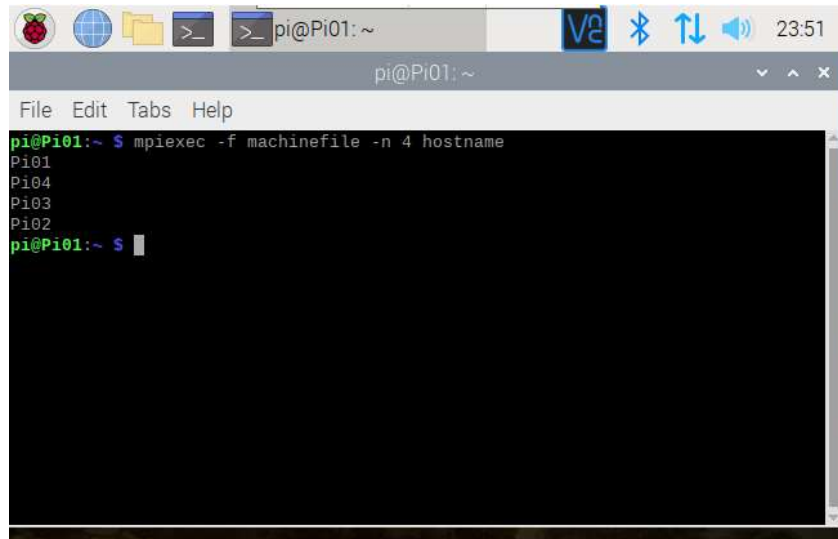
Sistem kluster yang dibangun pada penelitian ini menggunakan topologi jaringan *star*. Dimana topologi star akan menghubungkan *client* langsung menuju pusat jaringan, sehingga topologi ini sangat cocok diimplementasikan pada penelitian ini.

Sebagai pengganti *switch*, penelitian ini memanfaatkan *port* ethernet yang ada pada *router* Mikrotik yang berjumlah lima buah, satu buah dihubungkan menuju internet dan empat buah lainnya di *bridge* untuk menggantikan posisi *switch*, sehingga semua perangkat Raspberry Pi yang tersambung ke *router* Mikrotik berada pada satu *broadcast* domain.

Ketika semua Raspberry Pi telah terhubung pada satu *broadcast* domain sistem kluster dapat dijalankan sesuai dengan yang diharapkan. Peneliti menguji sistem kluster apakah berjalan atau tidak dengan mengirimkan pesan ke setiap *node* yang terhubung ke sistem kluster melalui *master node*, ketika mendapat respon nama *hostname* dari masing masing *node* maka pembuatan sistem kluster telah berjalan sesuai yang diharapkan.



Gambar 4.2.1 Sistem yang Digunakan



Gambar 4.2.2 Hasil Pengujian Sistem Kluster

4.3 Konfigurasi MPI

Pada kluster ini memanfaatkan *library* MPI dan python yang dijalankan diatas MPI, namun untuk menjalankan MPI dan python perlu beberapa tahapan.

Instalasi MPICH

Untuk menjalankan MPI diperlukan instalasi MPICH. Berikut konfigurasi MPICH yang dieksekusi melalui *terminal* di linux

- `sudo apt-get update`
- `sudo mkdir mpich2`
- `cd ~/mpich2`
- `sudo wget http://www.mpich.org/static/downloads/3.2/mpich-3.2.tar.gz`
- `sudo tar xzf mpich-3.2.tar.gz`
- `sudo mkdir /home/rpimpi/`
- `sudo mkdir /home/rpimpi/mpi-install`
- `mkdir /home/pi/mpi-build`
- `sudo apt-get install gfortran`
- `sudo /home/pi/mpich2/mpich-3.2/configure -prefix=/home/rpimpi/mpi-install`
- `sudo make`
- `sudo make install`
- `cd ~`

Gambar 4.3.1 Konfigurasi MPICH

Tambahkan *path* hasil instalasi mpi pada folder “.bashrc” dengan mengetikkan nano “.bashrc” dan tambahkan path berikut pada baris paling bawah

```
PATH=$PATH:/home/rpimpi/mpi-install/bin
```

Gambar 4.3.2 *Path* Hasil Instalasi MPI

Instalasi Python

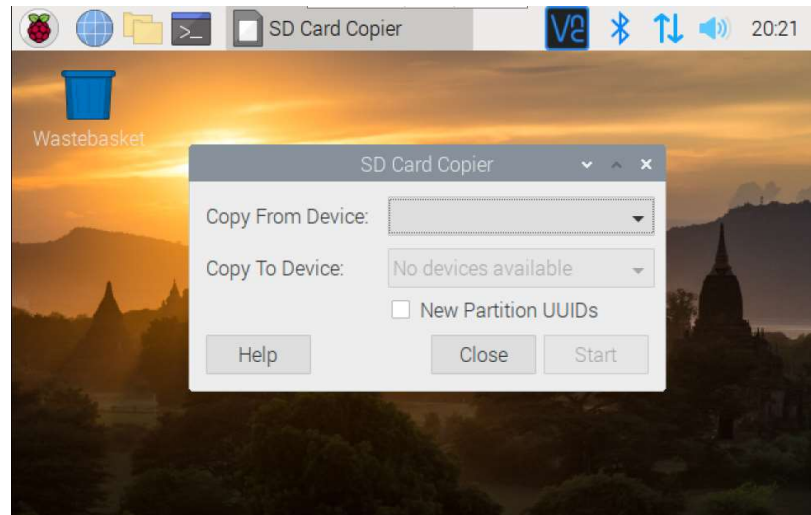
Pada penelitian ini diperlukan Python, Python tersebut digunakan untuk mengeksekusi sebuah *file* yang nantinya digunakan untuk proses akuisisi. Namun Python tersebut akan berjalan diatas MPI, sehingga diperlukan instalasi Python untuk MPI. Berikut instalasi Python

- wget <https://bitbucket.org/mpi4py/mpi4py/downloads/mpi4py-2.0.0.tar.gz>
- `sudo tar -zxf mpi4py-2.0.0.tar.gz`
- `cd mpi4py-2.0.0`
- `sudo aptitude install python-dev`
- `python setup.py build`
- `sudo python setup.py install`
- `export PYTHONPATH=/home/pi/mpi4py-2.0.0`

Gambar 4.3.3 Konfigurasi Python

Kloning *Micro SD*

Penelitian ini memerlukan empat Raspberry Pi yang perlu dikonfigurasi. Namun jika mengonfigurasi satu per satu tentu akan memerlukan waktu dan tenaga yang cukup banyak. Sehingga harus mencari alternatif lain untuk mengefektifkan waktu dan mengefisienkan kerja, salah satu nya dengan memanfaatkan fitur “SD Card Copier” yang ada pada raspbian. Sehingga hanya cukup mengonfigurasi satu sistem secara umum dan kemudian *Micro SD* hasil konfigurasi tersebut dikloning ke tiga *Micro SD* lainnya.



Gambar 4.3.4 SD Card Kloning

Machinefile

Untuk mengendalikan *node-node* dalam klaster, *master node* harus memiliki control mengenali alamat IP yang terdapat pada setiap Raspberry Pi yang akan dijadikan *node*. Untuk mengonfigurasi cukup membuat *file* bernama “*machinefile*” dengan mengetikkan perintah “`sudo nano machinefile`” dan mengisikan alamat *master node* dan *node-node* yang ada ke dalam *file* “*machinefile*”

- 192.168.11.254
- 192.168.11.251
- 192.168.11.252
- 192.168.11.253

Gambar 4.3.5 Machinefile

Secure Shell

Setelah *master node* mengenali *node-node* yang ada pada klaster, kemudian *master node* maupun *node* harus memiliki akses agar bisa saling terhubung. Untuk keperluan koneksi antar *master node* dan *node* memakai protokol SSH. Konfigurasi dilakukan pada seluruh Raspberry Pi baik *master node* maupun *node*

1. Membuat *keygen* ssh untuk Pi01 (*master node*)

```

• ssh-keygen
• cd ~
• cd .ssh
• cp id_rsa.pub Pi01

```

Gambar 4.3.6 SSH Pi01

2. Membuat *keygen* ssh untuk Pi02 (*node*) dan mengotorisasi ke Pi01

```

• ssh-keygen
• cd .ssh
• cp id_rsa.pub Pi02
• scp 192.168.11.254:/home/pi/.ssh/Pi01 .
• cat Pi01 >> authorized_keys

```

Gambar 4.3.7 SSH Pi02

3. Membuat *keygen* ssh untuk Pi03 (*node*) dan mengotorisasi ke Pi01

```

• ssh-keygen
• cd .ssh
• cp id_rsa.pub Pi03
• scp 192.168.11.254:/home/pi/.ssh/Pi01 .
• cat Pi01 >> authorized_keys

```

Gambar 4.3.8 SSH Pi03

4. Membuat *keygen* ssh untuk Pi04 (*node*) dan mengotorisasi ke Pi01

```

• ssh-keygen
• cd .ssh
• cp id_rsa.pub Pi04
• scp 192.168.11.254:/home/pi/.ssh/Pi01 .
• cat Pi01 >> authorized_keys

```

Gambar 4.3.9 SSH Pi04

5. Mengotorisasi Pi01 ke seluruh *node*

- scp 192.168.11.251:/home/pi/.ssh/Pi02 .
- cat Pi02 >> authorized_keys
- scp 192.168.11.252:/home/pi/.ssh/Pi03 .
- cat Pi03 >> authorized_keys
- scp 192.168.11.253:/home/pi/.ssh/Pi04 .
- cat Pi04 >> authorized_keys

Gambar 4.3.10 Otorisasi SSH Pi01

4.4 Konfigurasi Cacti

Cacti memerlukan sebuah database untuk menyimpan penulisan hasil monitoring. Pada penelitian kali ini menggunakan MariaDB sebagai *database*. Terlebih dahulu melakukan instalasi MariaDB *server* dan *client*. Setelah instalasi selesai, dilakukan *update* pada *file* “50-server.cnf”. Tambahkan atau perbaharui pengaturan yang ada pada *file* tersebut.

```

collation-server = utf8mb4_unicode_ci
character-set-server = utf8mb4
max_heap_table_size = 128M
tmp_table_size = 64M
join_buffer_size = 64M
innodb_file_format = Barracuda
innodb_large_prefix = 1
innodb_buffer_pool_size = 1GB
innodb_buffer_pool_instances = 10
innodb_flush_log_at_timeout = 3
innodb_read_io_threads = 32
innodb_write_io_threads = 16
innodb_io_capacity = 5000
innodb_io_capacity_max = 10000

```

Gambar 4.4.1 Konfigurasi MariaDB

Setelah instalasi MariaDB selesai, kemudian menyiapkan web *server* dan PHP *extensions* karena Cacti berbasis web *based*. Sehingga nantinya konfigurasi Cacti melalui web. Terlebih dahulu lakukan instalasi Apache dan PHP *extensions*. Setelah instalasi

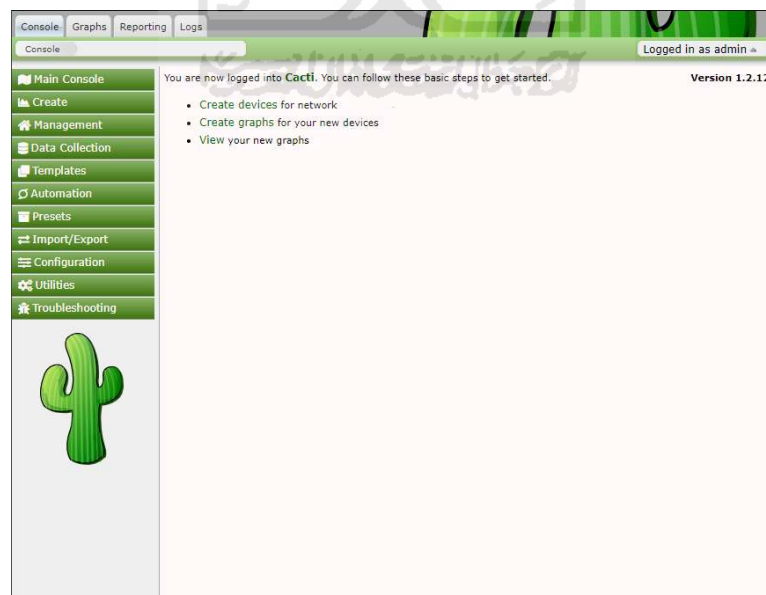
Apache dan PHP *extensions* kemudian atur *Timezone* pada *file* “php.ini” di dalam folder “apache2” dan “cli”. Lakukan pada kedua *file* tersebut.

```
date.timezone = US/Central
memory_limit = 512M
max_execution_time = 60
```

Gambar 4.4.2 Konfigurasi *Timezone*

Setelah konfigurasi web *server* dan instalasi PHP *extension*, kemudian melakukan instalasi SNMP dan RRD *Tool*. Karena cacti merupakan aplikasi hasil pengembangan dari RRD dan menggunakan protokol SNMP untuk media komunikasinya. Setelah melakukan instalasi SNMP dan RRD *Tool*. Kemudian membuat *database* yang nantinya digunakan untuk basis data Cacti. Basis data tersebut dibuka *permission*nya sehingga nantinya Cacti bisa mendapatkan akses ke *database* secara penuh.

Ketika semua sudah berhasil diinstall dan dikonfigurasi, selanjutnya mengonfigurasi cacti melalui *browser* dengan mengetikkan “ip_address cacti_server/cacti”. Ketika pertama kali mengakses cacti akan disuguhkan halaman *login*, secara *default username* dan *password* menggunakan “admin”.



Gambar 4.4.3 Tampilan Cacti

4.5 Konfigurasi SNMP

Agar perangkat Raspberry Pi (*node*) dapat dimonitoring ke dalam Cacti perlu mengonfigurasi SNMP *client* terlebih dahulu, namun sebelum itu melakukan instalasi SNMP *client*. Setelah melakukan instalasi SNMP *client* kemudian mengatur konfigurasi SNMP pada *file* “snmpd.conf”.

```
agentAddress udp:161
rocommunity public 192.168.11.0/24
```

Gambar 4.5.1 Konfigurasi SNMP *Client*

Setelah konfigurasi SNMP *client* pada Raspbery Pidilakukan, kemudian melakukan konfigurasi SNMP pada Mikrotik dengan mngetikkan perintah pada *terminal* melalui Winbox.

```
/snmp community
set [ find default=yes ] addresses=0.0.0.0/0 write-access=yes
/snmp
set enabled=yes trap-version=2
```

Gambar 4.5.2 Konfigurasi SNMP Mikrotik

Setelah semua perangkat yang akan dimonitoring dilakukan konfigurasi SNMP *client*. Kemudian melakukan konfigurasi SNMP pada Cacti agar nantinya kedua sistem dapat berkomunikasi untuk kebutuhan monitoring sumber daya.

Pi02 (192.168.11.251)
SNMP Information
 System: Linux Pi02 4.19.75-v7+ #1270 SMP Tue Sep 24 18:45:11 BST 2019 armv7l
 Uptime: 2301143 (0days, 6hours, 23minutes)
 Hostname: Pi02
 Location: Unknown
 Contact: root

*Create New Device
 *Create Graphs for this Device
 *Re-Index Device
 *Enable Device Debug
 *Data Source List
 *Graph List

Ping Results
 ICMP Ping Success (0.628 ms)
 Device [edit: Pi02]

General Device Options

Description: Pi02
 Hostname: 192.168.11.251
 Location: None
 Poller Association: Main Poller
 Device Site Association: Edge
 Device Template: Net-SNMP Device
 Number of Collection Threads: 1 Thread (default)
 Disable Device:

SNMP Options

SNMP Version: Version 2
 SNMP Community String: public
 SNMP Port: 161
 SNMP Timeout: 500
 Maximum OIDs Per Get Request: 10

Availability/Reachability Options

Downed Device Detection: Ping and SNMP Uptime
 Ping Method: ICMP Ping
 Ping Timeout Value: 400
 Ping Retry Count: 1

Additional Options

Notes

Gambar 4.5.3 Konfigurasi SNMP Server

4.6 Konfigurasi SMB

Untuk kebutuhan *file sharing*, penelitian ini memanfaatkan fitur Samba yang ada pada perangkat Mikrotik. Peneliti menggunakan fitur Samba pada perangkat Mikrotik karena praktis dan konfigurasinya tidak rumit pada Raspbian melalui Raspberry Pi. Untuk mengaktifkan fitur Samba tersebut perlu mengetikkan perintah berikut pada *terminal* Mikrotik melalui Winbox.

```

/ip smb
set allow-guests=no enabled=yes
/ip smb shares
add directory=/disk3 name=disk3
/ip smb users
add name=test password=test read-only=no

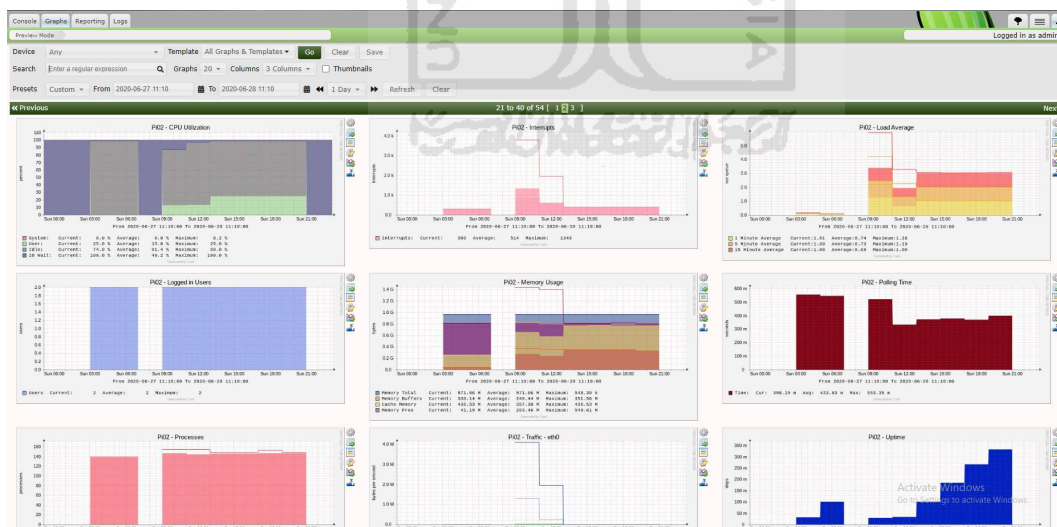
```

Gambar 4.6.1 Konfigurasi Samba *File Sharing*

Setelah fitur Samba aktif kemudian *dimountkan* pada Raspberry Pi (node) yang nantinya akan digunakan sebagai tujuan hasil akuisisi.

4.7 Pengujian Sistem Monitoring

Setelah konfigurasi Cacti dan SNMP selesai dilakukan, kemudian dilakukan pengujian pada sistem monitoring. Pengujian dilakukan melalui web *browser* kemudian masuk *webfig* Cacti untuk memastikan bahwa grafik pada Cacti sudah berjalan sesuai yang diharapkan.



Gambar 4.7.1 Grafik Hasil Monitoring

4.8 Pengujian Sistem Akuisisi

Proses pengujian pada penelitian ini akan menggunakan sebuah sistem yang terdiri dari

Tabel 4.88.1 Tabel Spesifikasi Sistem Independen

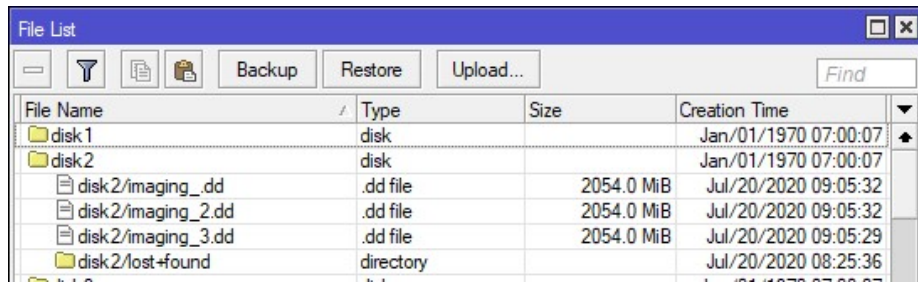
Raspberry Pi 3 model B
Raspbian 10.1 (Buster)
Micro SD APACER 16GB Class 10
Lan ±20 cm AMP CAT 5E

Sistem tersebut dirancang menjadi tiga sistem independen yang identik agar mengantisipasi terjadinya perbedaan performa (*bottleneck*) yang signifikan ketika dilakukan pengujian. Dari tiga sistem independen tersebut dirancang menjadi sistem kluster yang akan digunakan untuk mengakuisisi tiga buah *flash drive* Kingston DT100G3. Agar tidak memerlukan waktu yang sangat lama, *flash drive* yang semula 16GB *resize* dan dipartisi sebesar 2GB berlaku untuk semua *flash drive*. Setelah menjadi sistem kluster, selanjutnya dilakukan tahap pengujian dengan enam skenario. Untuk menjalankan perintah akuisisi menggunakan kode program berbasis python.

```
with open("source",'rb') as f:
    with open("destination_path.dd", "wb") as i:
        while True:
            if i.write(f.read(512)) == 0:
                break
```

Gambar 4.8.1 Perintah Akuisisi

Dari hasil menjalankan perintah akuisisi tersebut nantinya setiap pengujian akan menghasilkan *file* sesuai dengan jumlah *flash drive* yang digunakan setiap pengujian. *File* tersebut nantinya akan berada pada *flash drive* yang dipasangkan pada *Router* Mikrotik.



File Name	Type	Size	Creation Time
disk 1	disk		Jan/01/1970 07:00:07
disk 2	disk		Jan/01/1970 07:00:07
disk 2/imaging_dd	.dd file	2054.0 MiB	Jul/20/2020 09:05:32
disk 2/imaging_2.dd	.dd file	2054.0 MiB	Jul/20/2020 09:05:32
disk 2/imaging_3.dd	.dd file	2054.0 MiB	Jul/20/2020 09:05:29
disk 2/lost+found	directory		Jul/20/2020 08:25:36

Gambar 4.8.2 File Hasil Akuisisi

4.8.1 Skenario Pertama

Pada skenario pengujian yang pertama, seluruh *node* dijalankan dan masing masing *node* dipasangkan satu buah *flash drive* yang *diresize* menjadi 2 GB. *File* hasil akuisisi tersebut akan masuk melalui Samba *file sharing* menuju Mikrotik. Sedangkan penyimpanan yang digunakan untuk *file sharing* tersebut menggunakan *flash drive* sebesar 16 GB.

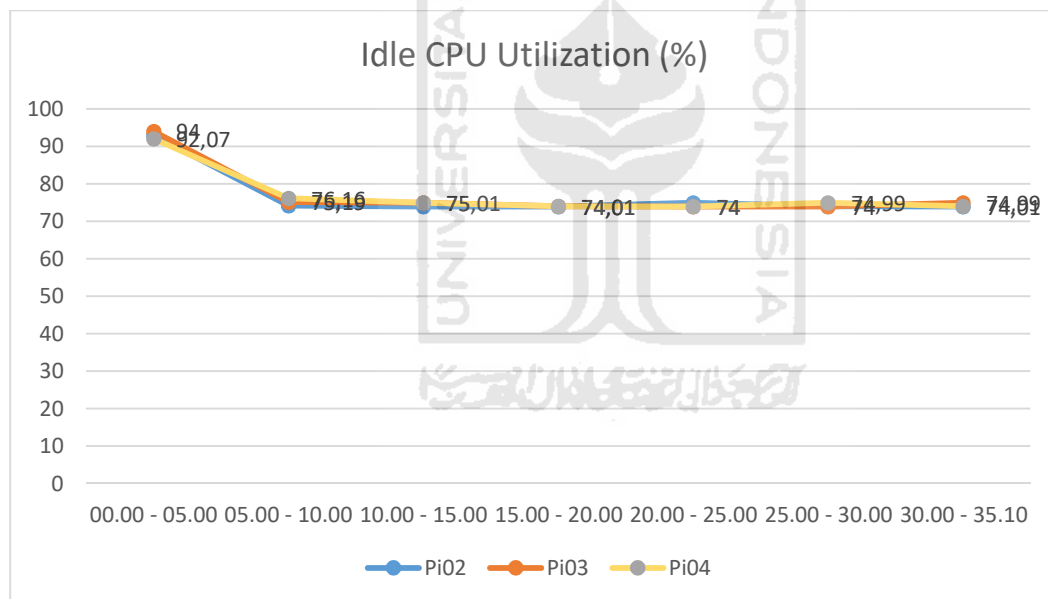


Diagram 4.8.1.1 Idle CPU Utilization Skenario Pertama



Diagram 4.8.1.2 *Free Memory* Skenario Pertama

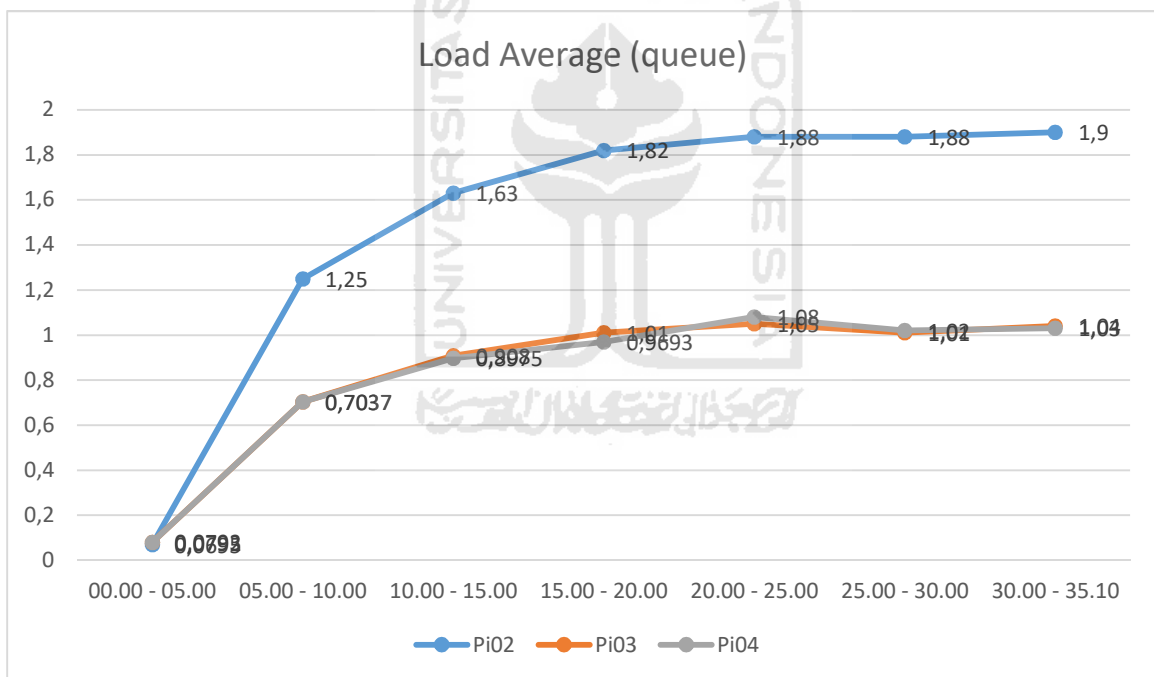


Diagram 4.8.1.3 *Load Average* Skenario Pertama

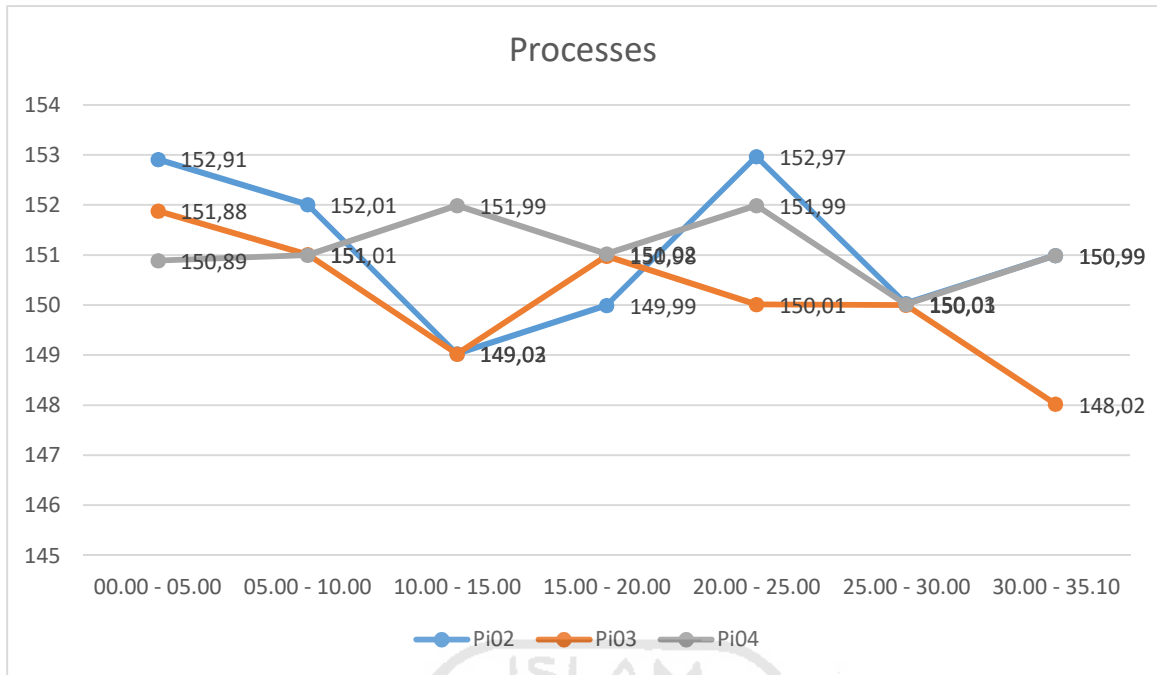


Diagram 4.8.1.4 *Process* Skenario Pertama

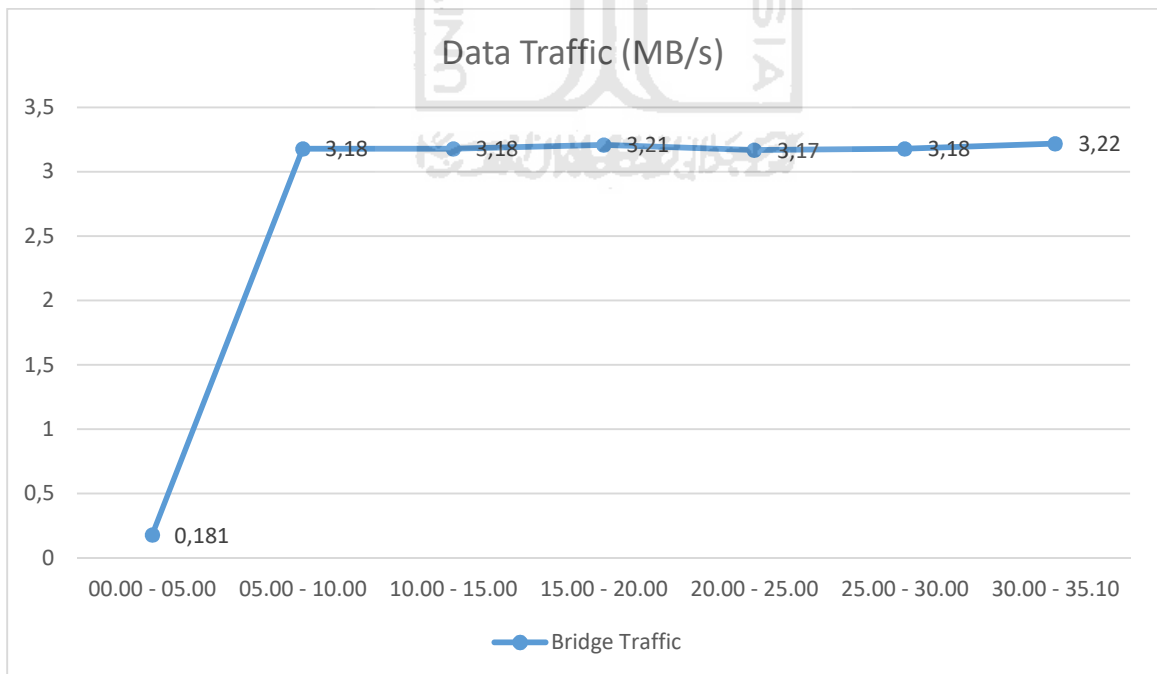


Diagram 4.8.1.5 *Data Traffic* Skenario Pertama

Parameter *idle CPU utilization* dan *free memory* mengalami penurunan untuk ketiga *node* ketika diberikan beban untuk akuisisi. Tetapi tidak ada perbedaan signifikan hingga selesai proses akuisisi. Pada parameter *load average* “Pi02” mendapatkan angka yang hampir dua kali dibanding “Pi03” dan “Pi04”. Pada parameter *processes* semua *node* mencatatkan angka yang dinamis, tetapi pada 5 menit ke-enam sempat mendapatkan nilai yang sama pada ketiga *node*. Pada skenario pertama ini mencatatkan lalu lintas data rata rata 3MB/s dan dengan total waktu akuisisi 35 menit 10 detik.

4.8.2 Skenario Kedua

Skenario kedua dijalankan dengan dua Raspberry Pi (*node*) yang dipasangkan dua buah *flash drive* pada masing masing Raspberry Pi.

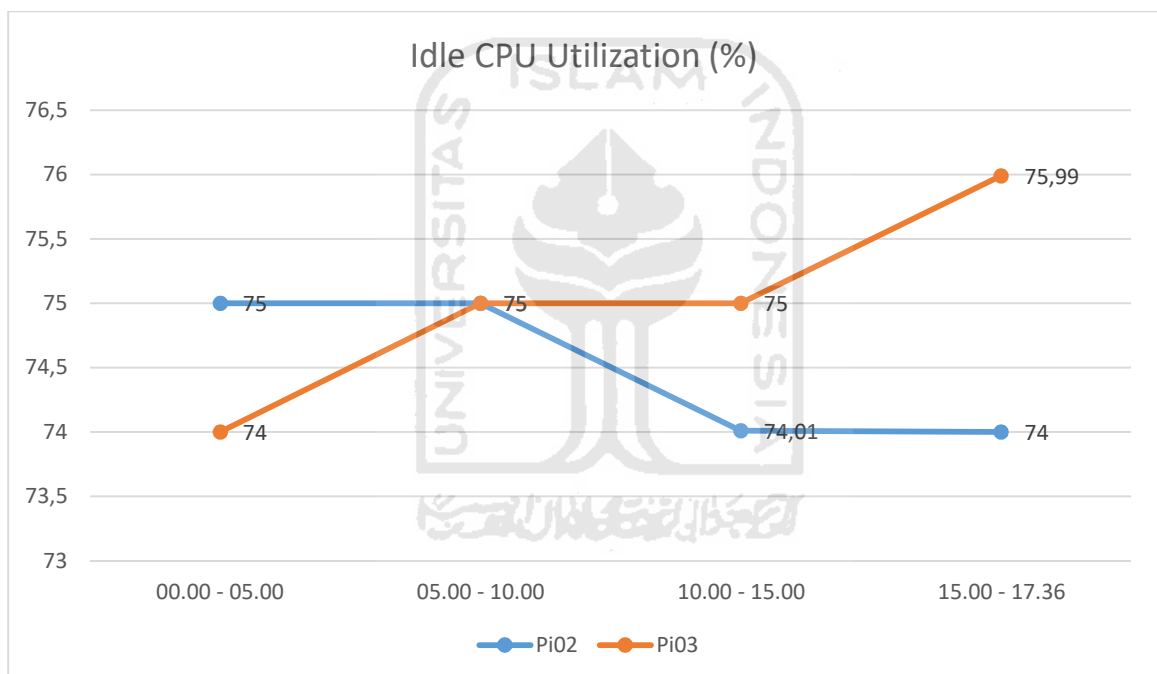


Diagram 4.8.2.1 *Idle CPU Utilization* Skenario Kedua

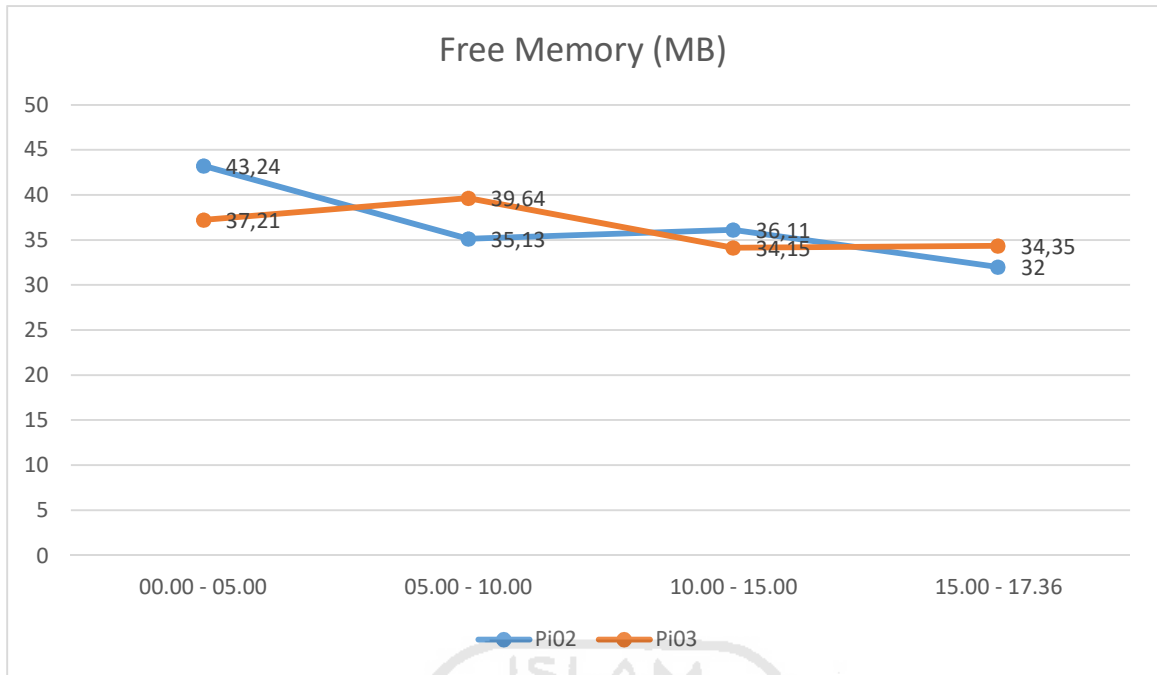


Diagram 4.8.2.2 *Free Memory* Skenario Kedua

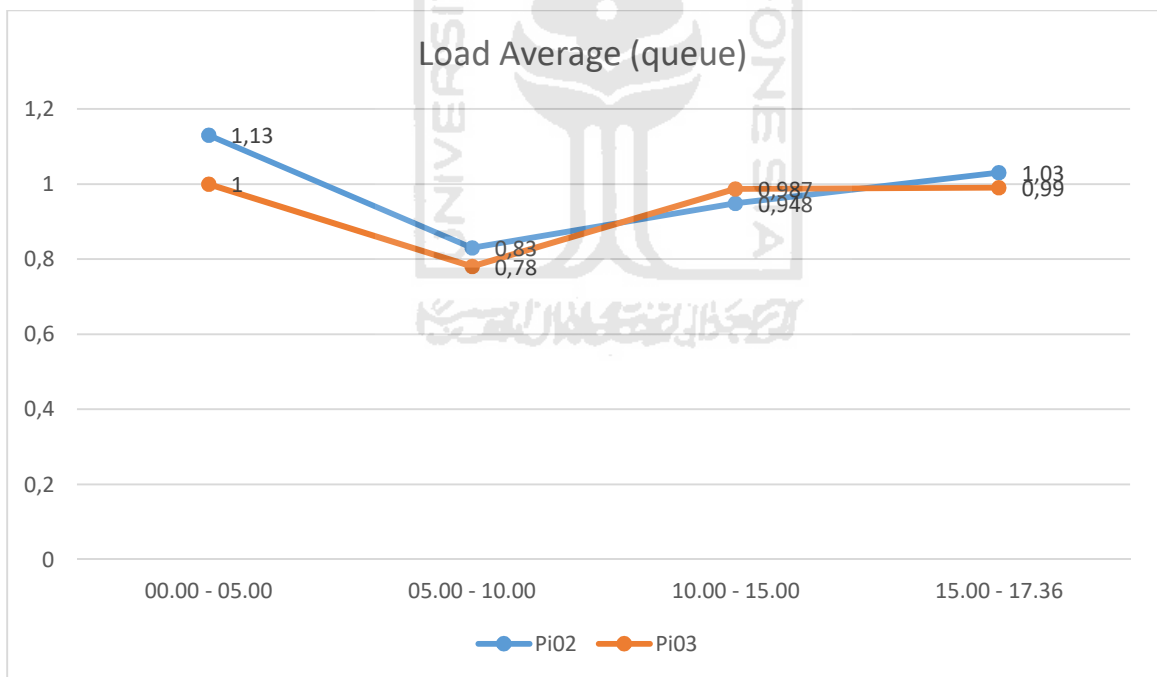


Diagram 4.8.2.3 *Load Average* Skenario Kedua

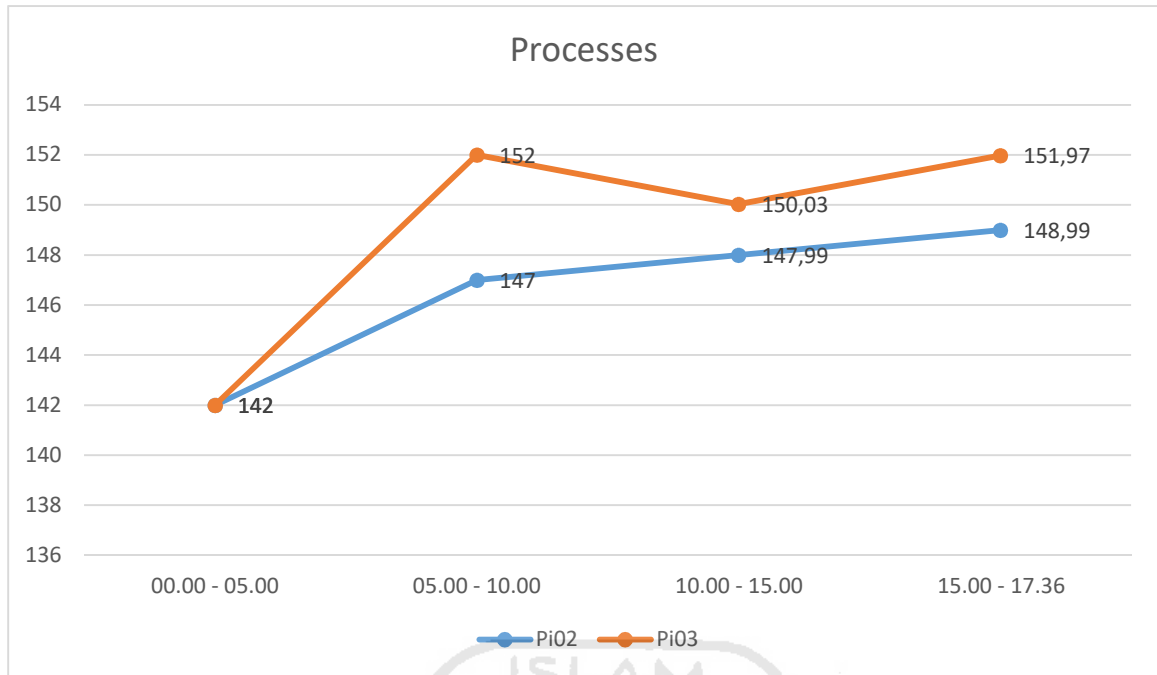


Diagram 4.8.2.4 *Process* Skenario Kedua

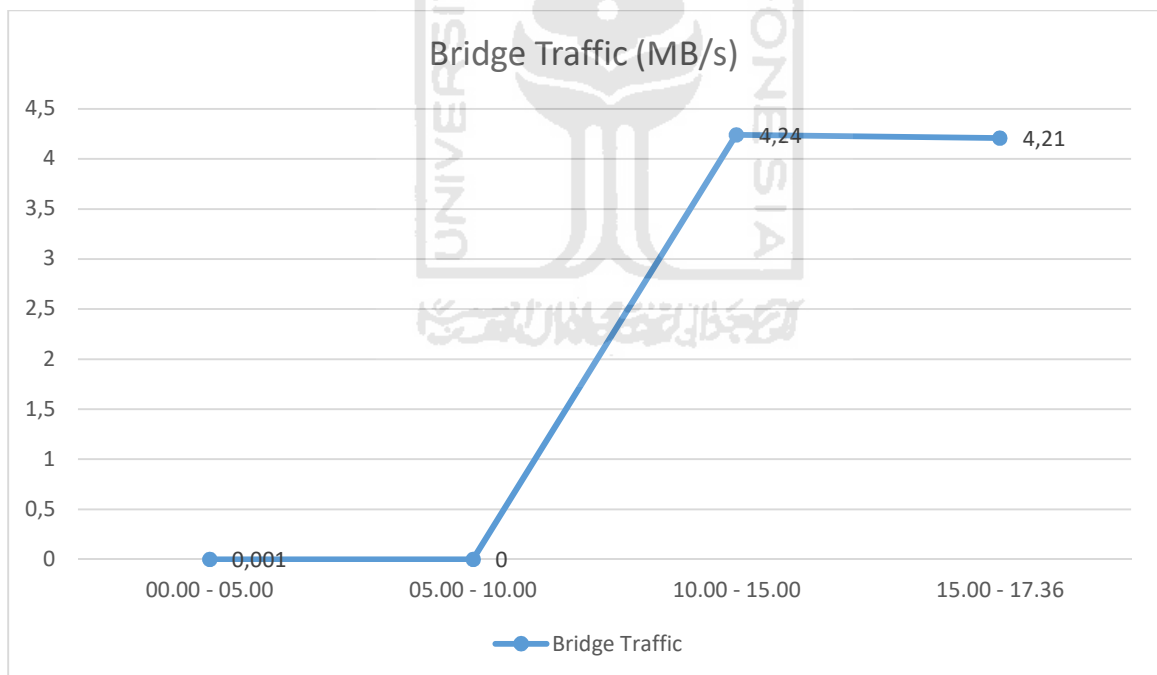


Diagram 4.8.2.5 *Data Traffic* Skenario Kedua

Skenario ini pada parameter *idle CPU utilization*, *free memory*, *load average* dan *processe* tidak mengalami perbedaan yang signifikan. Tetapi pada lalu lintas data untuk lima menit pertama dan lima menit kedua mencatatkan tidak ada lalu lintas data yang terjadi, namun pada lima menit ketiga mengalami perubahan yang sangat signifikan ke angka 4,24 MB/s dan

4,21 MB/s hingga selesai pengujian ini. Waktu yang dibutuhkan untuk menyelesaikan pengujian ini sebesar 17 menit 36 detik.

4.8.3 Skenario Ketiga

Pada skenario ini hanya menggunakan satu buah Raspberry Pi dan dipasangkan satu buah *flash drive*

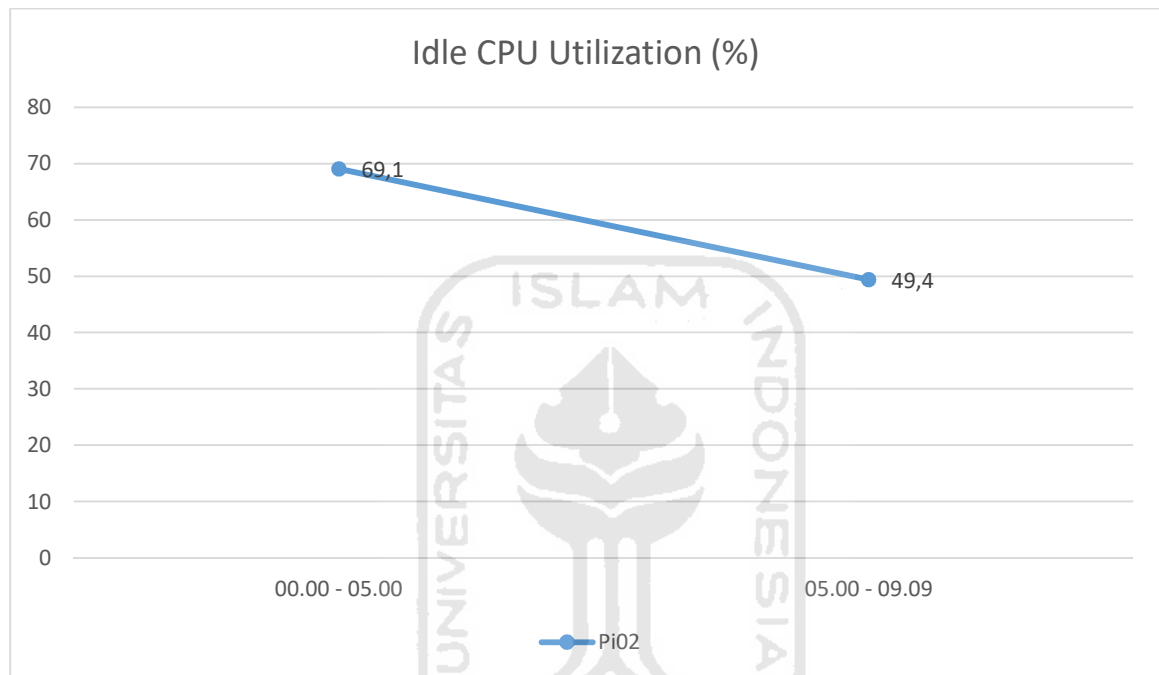
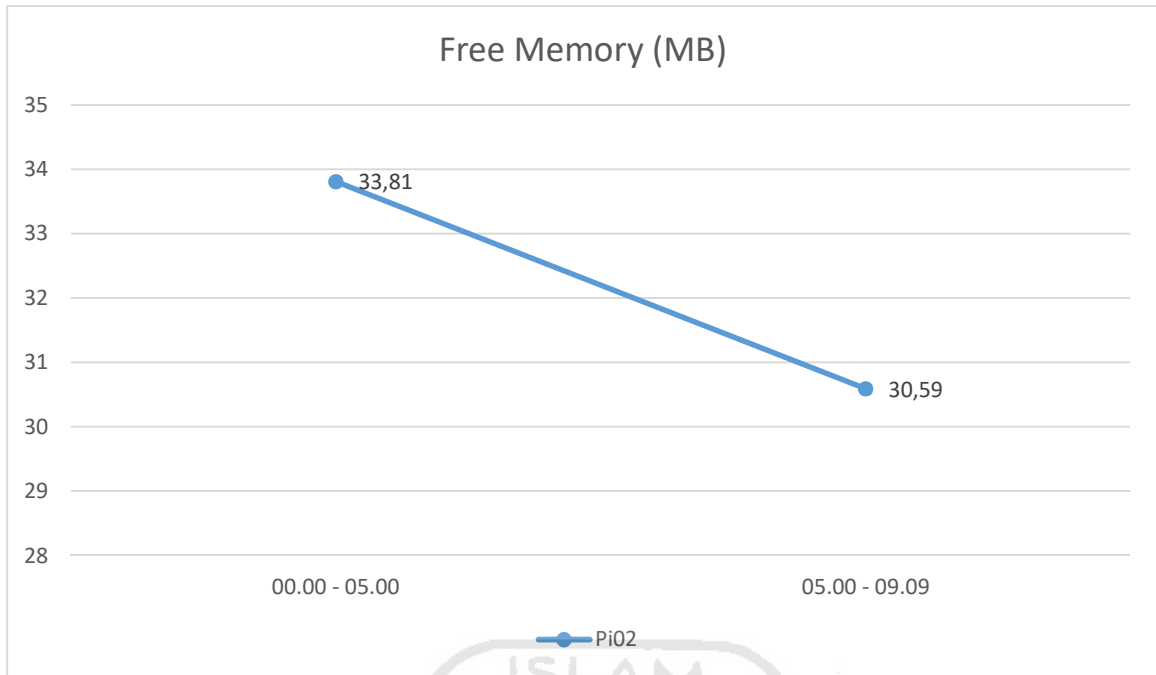
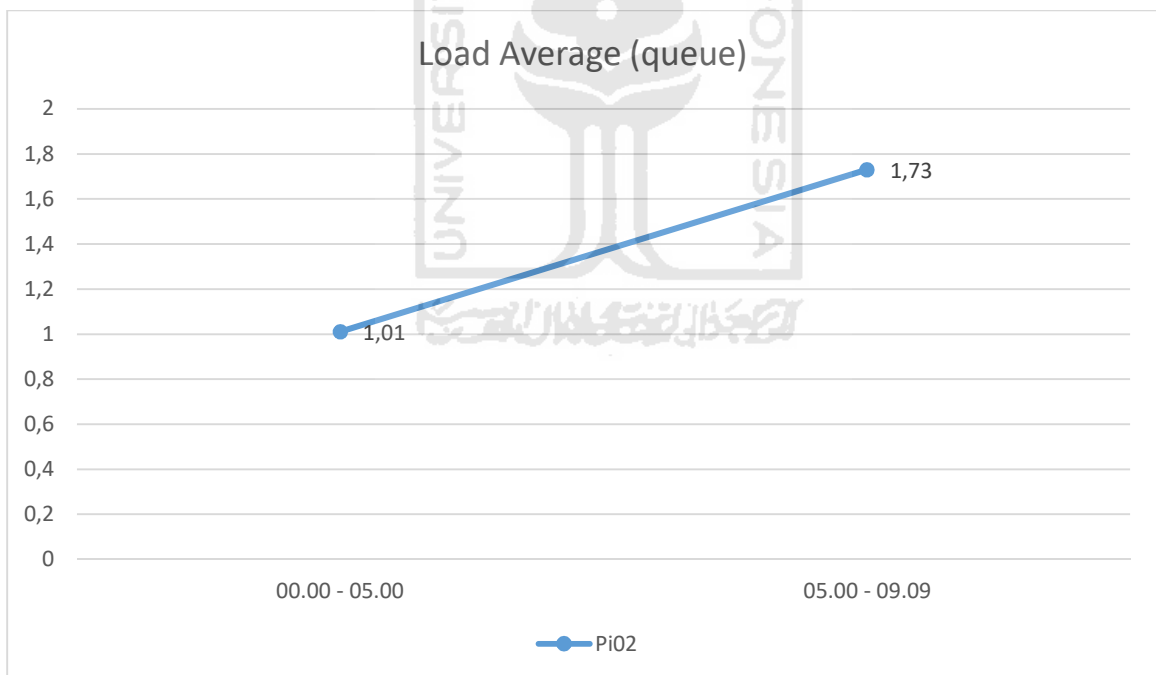


Diagram 4.8.3.1 *Idle CPU Utilization* Skenario Ketiga

Diagram 4.8.3.2 *Free Memory* Skenario KetigaDiagram 4.8.3.3 *Load Average* Skenario Ketiga

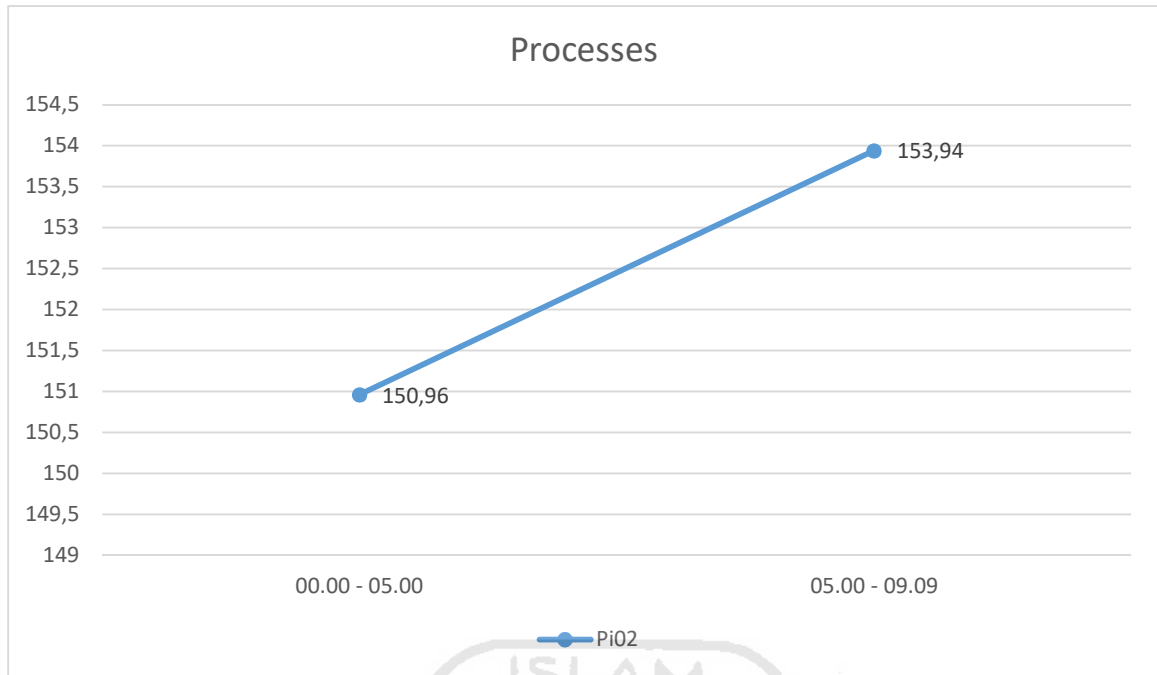


Diagram 4.8.3.4 *Process* Skenario Ketiga

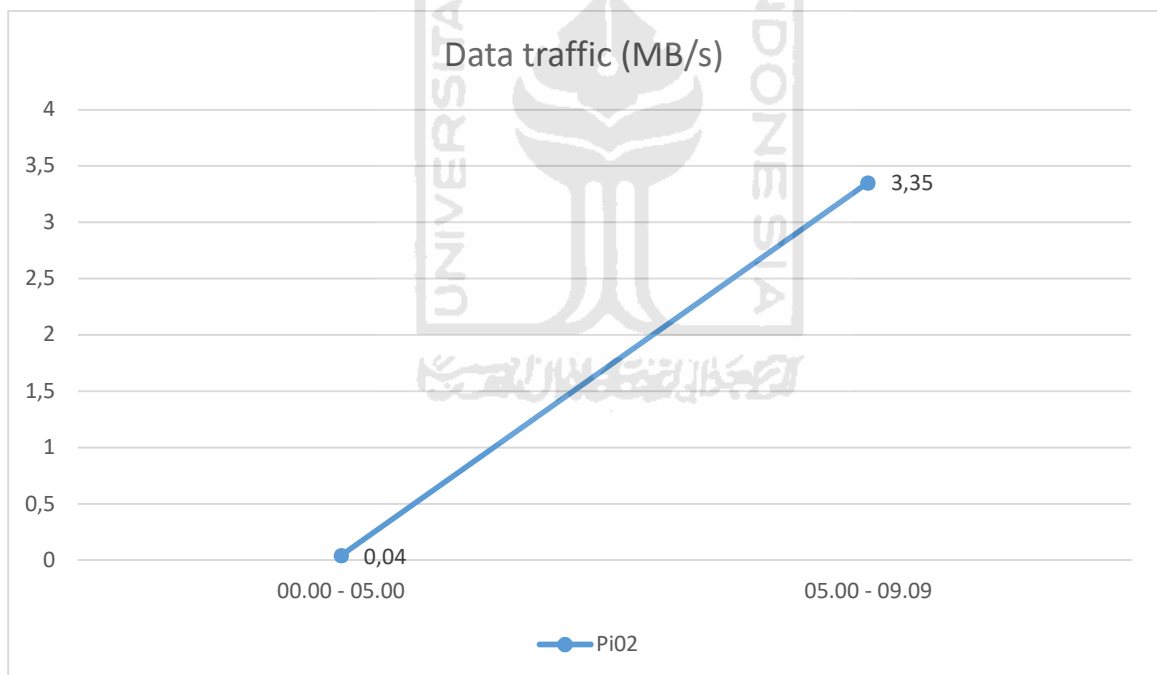


Diagram 4.8.3.5 *Data Traffic* Skenario Ketiga

Pada skenario ketiga ini, parameter *idle CPU utilization* turun secara disignifikan yang semula 69,1 % menjadi 49,4 %. Parameter *free memory* tidak mencatatkan hasil perubahan yang signifikan. Namun pada parameter *load average*, terjadi lonjakan yang awalnya 1,01 *queue* naik menjadi 1,73 *queue*. Pada parameter *processes* tidak mengalami kenaikan yang

signifikan. Dan lalu lintas data yang terjadi naik dari 0,04 MB/s menjadi 3,35 MB/s. Waktu yang dibutuhkan untuk menyelesaikan pengujian ini cukup singkat, yaitu 9 menit 9 detik.

4.8.4 Skenario Keempat

Pada skenario ini menggunakan “Pi03” sebagai *nodenya* dikarenakan terjadi masalah ketika menggunakan lebih dari satu *flash drive* pada “Pi02”. Karena pada skenario ini menggunakan konfigurasi satu buah Raspberry Pi dan dua buah *flash drive* yang dipasangkan.

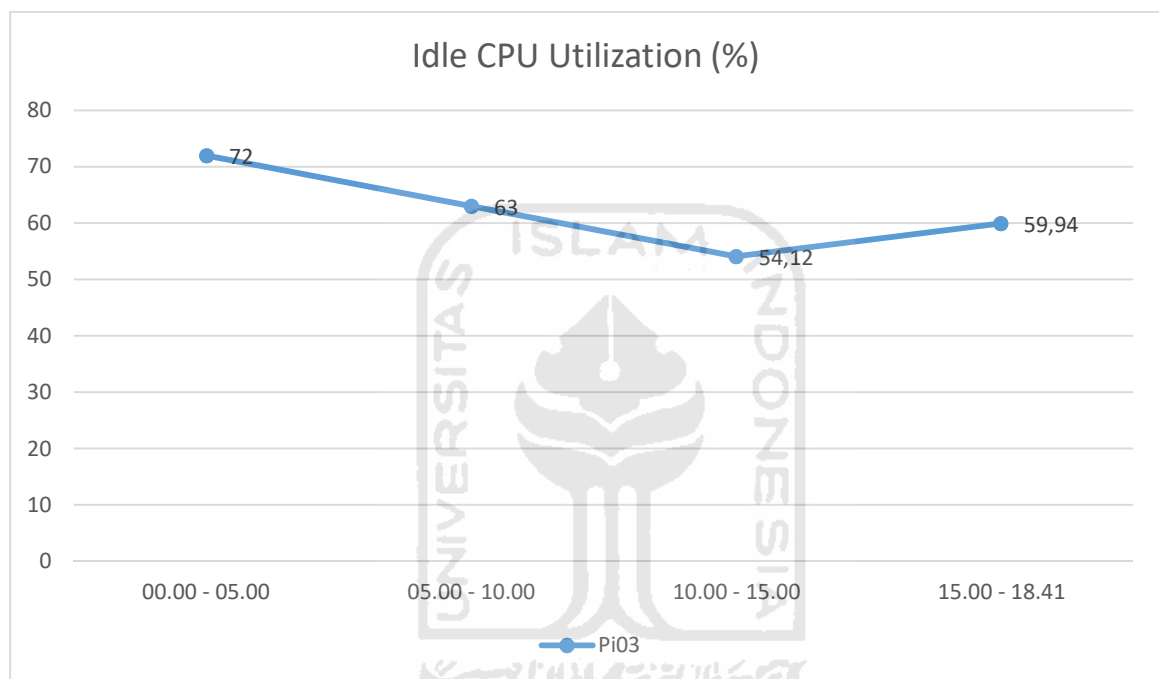


Diagram 4.8.4.1 *Idle CPU Utilization* Skenario keempat

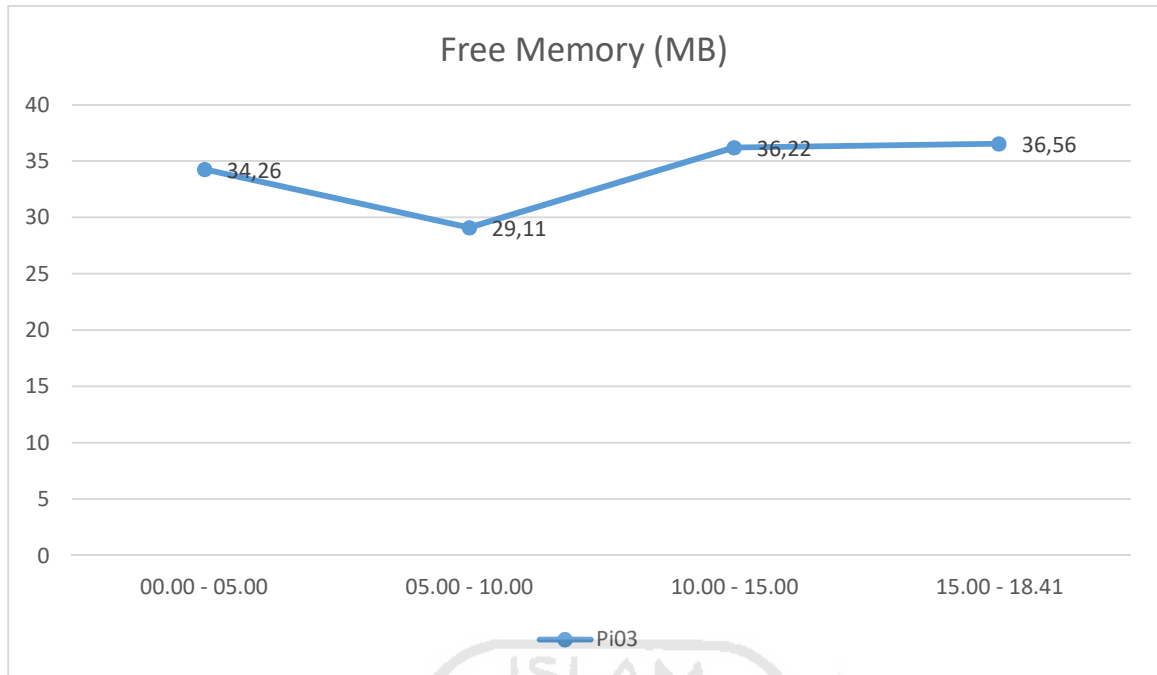


Diagram 4.8.4.2 *Free Memory* Skenario Keempat



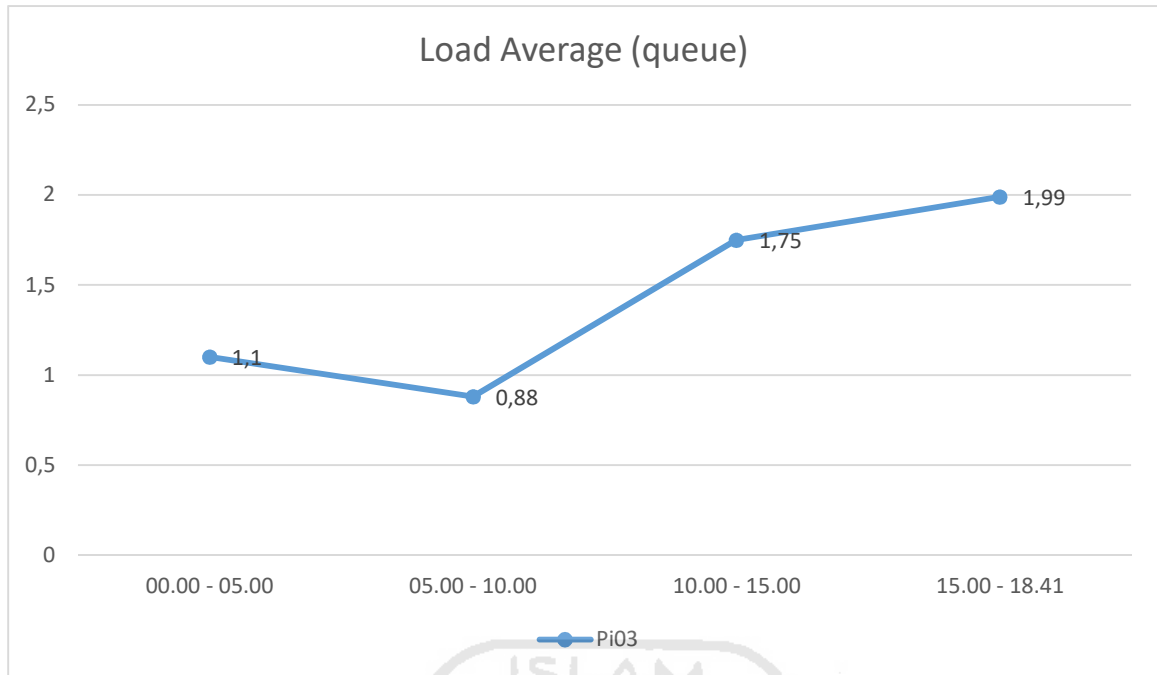


Diagram 4.8.4.3 Load Average Skenario Keempat

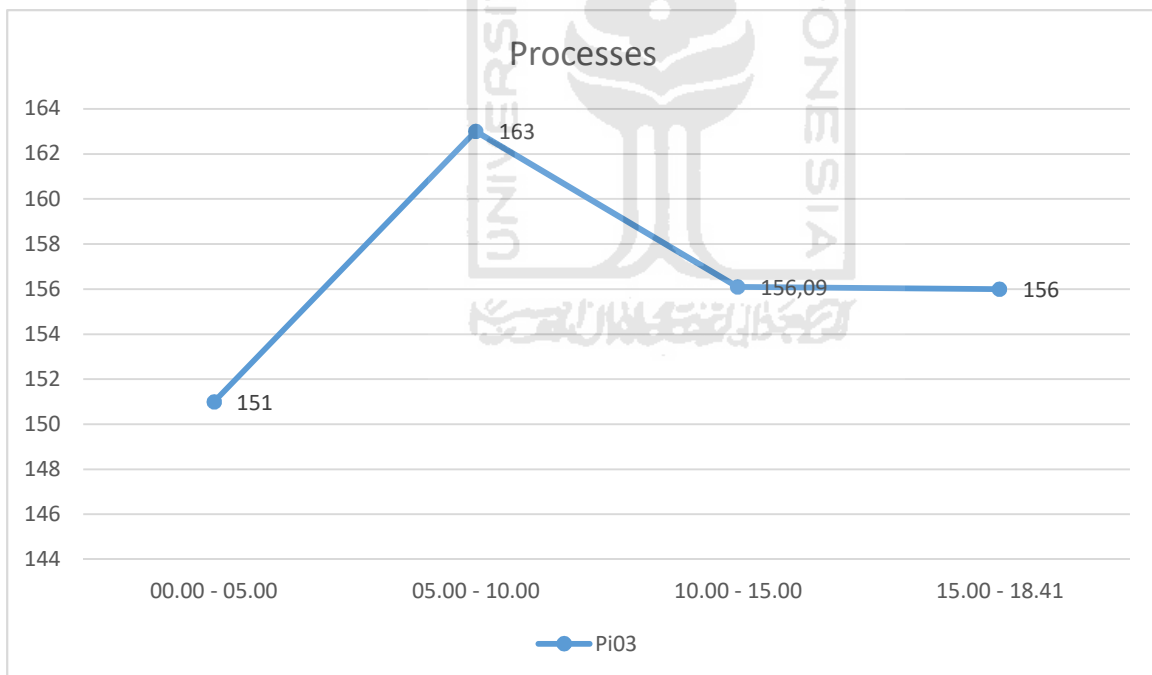


Diagram 4.8.4.4 Processes Skenario Keempat

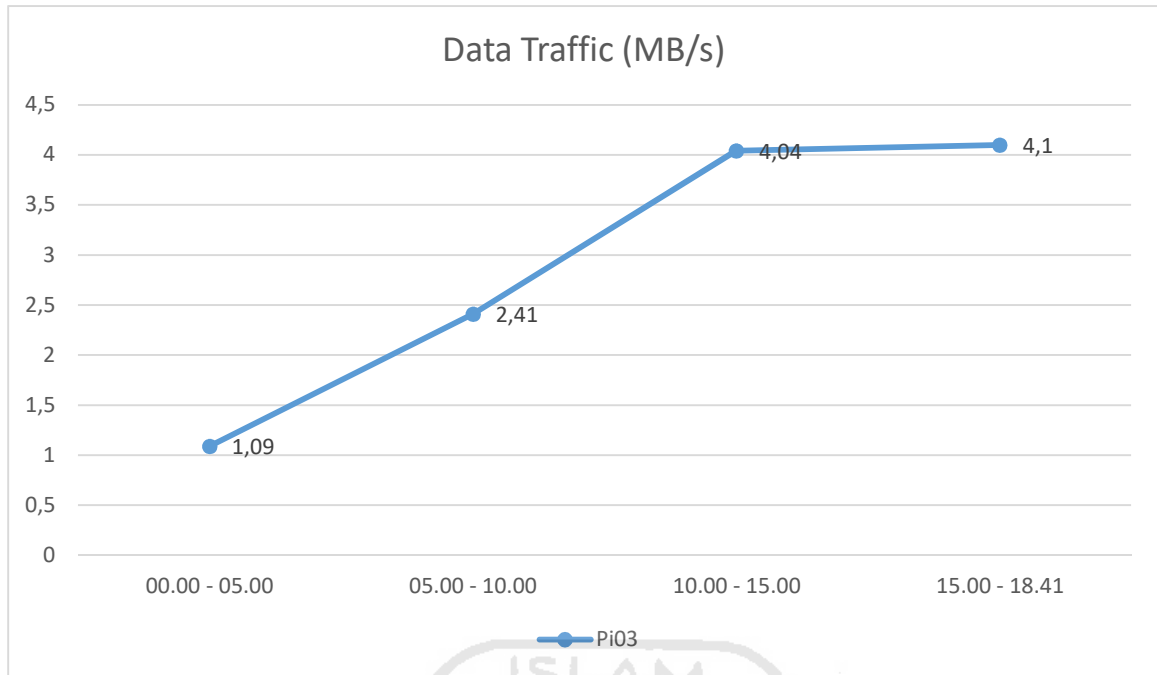


Diagram 4.8.4.5 Data *Traffic* Skenario Keempat

Ketika skenario keempat dijalankan, *idle CPU utilization* sempat mengalami perubahan yang signifikan dari lima menit pertama sebesar 72 % hingga lima menit ketiga menjadi 54,12 % dan terjadi kenaikan lagi pada lima menit ke empat menjadi 59,94 %. Parameter *free memory* tidak mengalami perubahan yang signifikan. Namun parameter *load average* mengalami kenaikan yang sangat signifikan, ketika skenario dijalankan mencatatkan hasil 1,1 *queue* dan terjadi penurunan pada lima menit berikutnya, kemudian naik hingga mencapai 1,99 *queue*. Processes yang terjadi sempat mengalami kenaikan yang signifikan sebesar 12 *process* antara lima menit pertama hingga lima menit kedua, namun lima menit ketiga turun dan stabil hingga selesai pengujian. Lalu lintas data yang terjadi cukup besar, semula 1,09 MB/s kemudian naik menjadi sekitar 4 MB/s. Pada skenario ini membutuhkan waktu sebesar 18 menit 41 detik.

4.8.5 Skenario Kelima

Skenario pengujian ini hanya menggunakan satu buah Raspberry Pi dan dipasangkan dengan tiga buah *flash drive*.

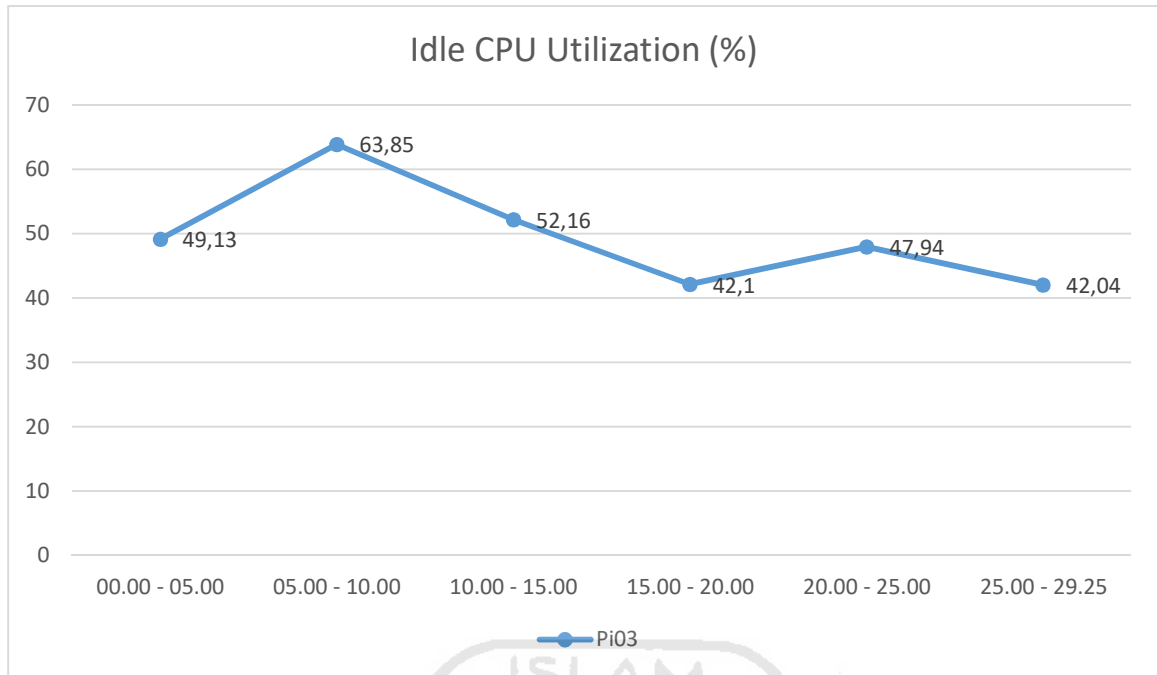


Diagram 4.8.5.1 *Idle CPU Utilization* Skenario Kelima

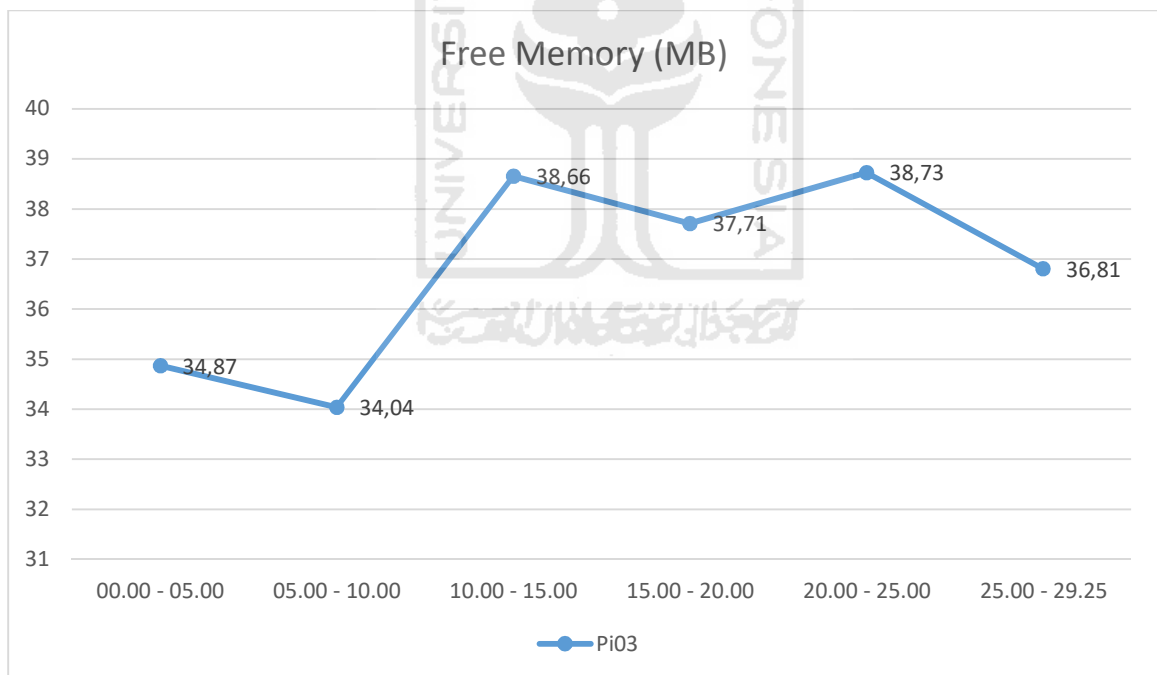


Diagram 4.8.5.2 *Free Memory* Skenario Kelima

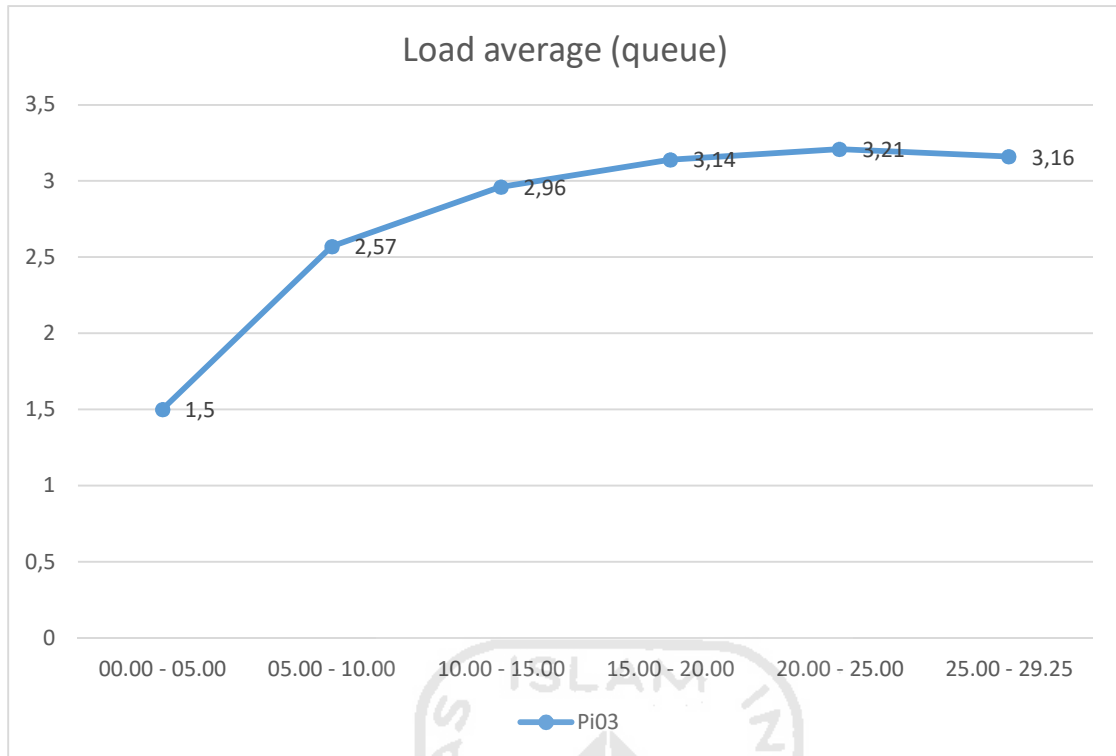


Diagram 4.8.5.3 *Load Average* Skenario Kelima

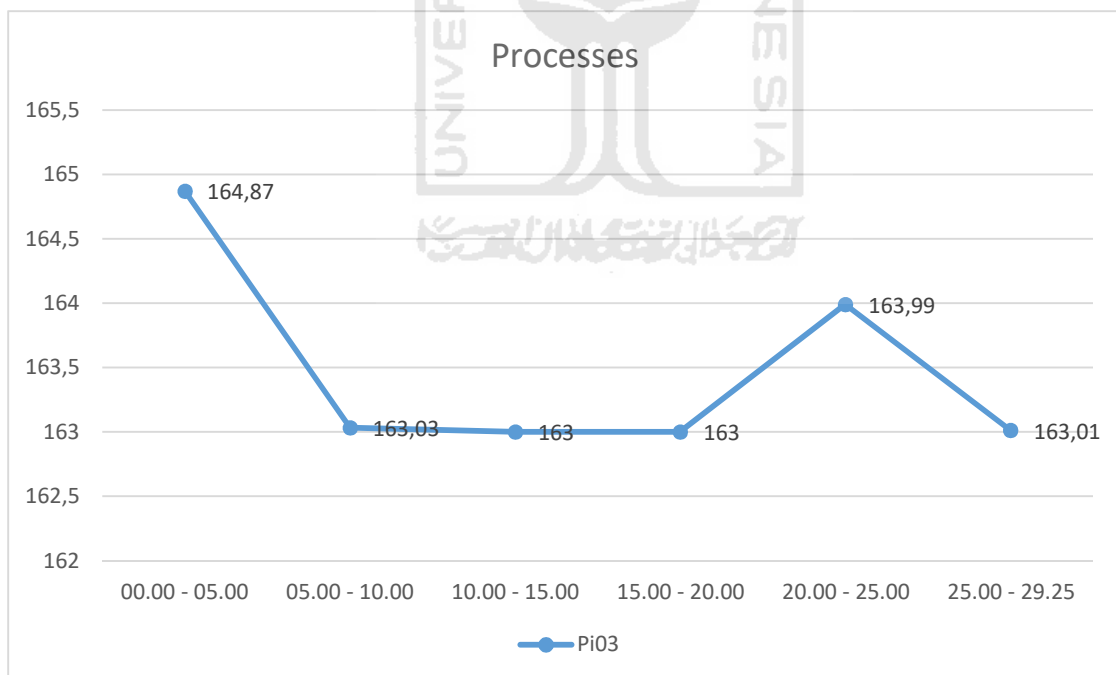


Diagram 4.8.5.4 *Processes* Skenario Kelima

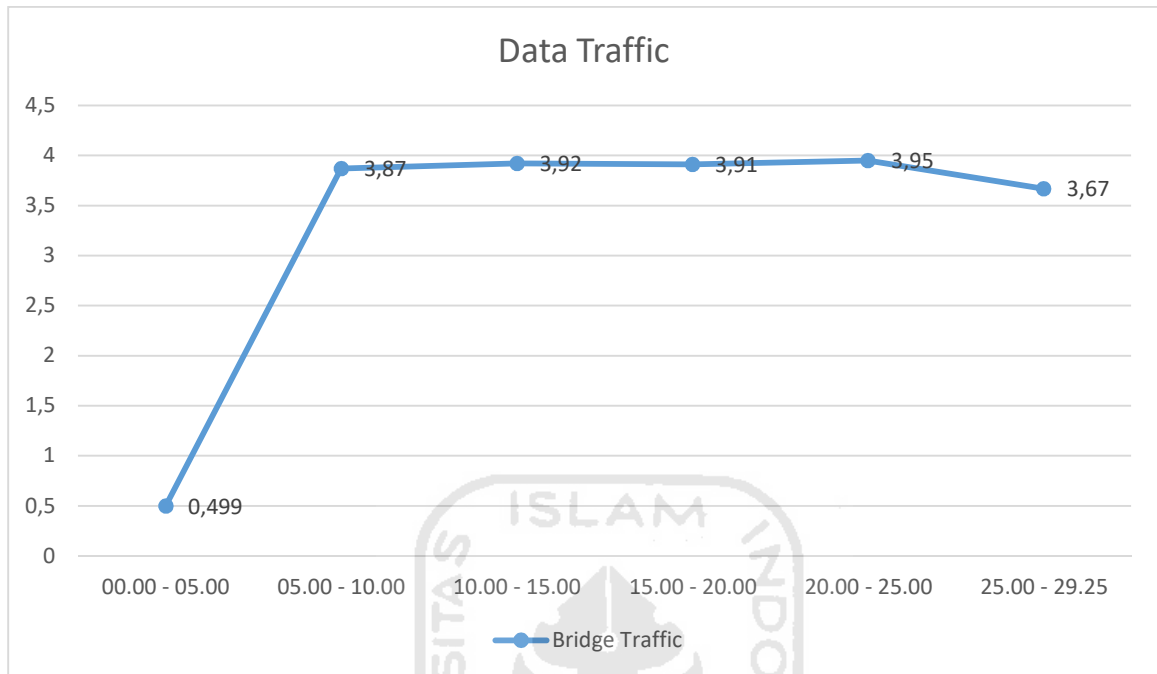


Diagram 4.8.5.5 Data *Traffic* Skenario Kelima

Pada skenario kelima ini, ketika diberikan beban tiga *flash drive*, grafik *idle CPU utilization* menjadi sangat dinamis. Namun *free memory* tidak mengalami perubahan yang signifikan. Tetapi pada parameter *load average* mencatatkan hasil yang sangat tinggi sekali namun hal itu tetap membuat nilai yang dicatatkan pada parameter *process* terbilang stabil, walaupun nilai yang dicatatkan juga terbilang cukup tinggi. Lalu lintas data yang dicatatkan pada skenario ini rata rata hampir menyentuh angka 4MB/s dan dengan waktu 29 menit 25 detik.

4.8.6 Skenario Keenam

Pada skenario terakhir pengujian sistem kluster ini menggunakan konfigurasi dua Raspberry Pi dengan dipasangkan tiga *flash drive*, namun pada *node* Pi03 dipasangkan dua buah *flash drive*.

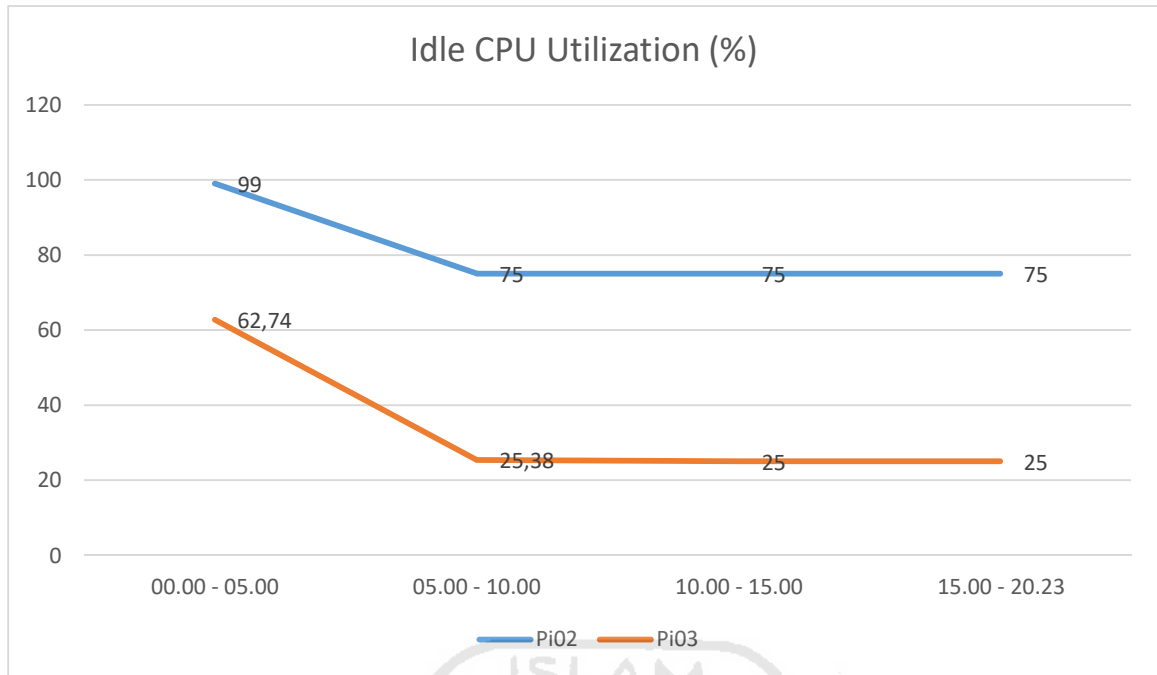


Diagram 4.8.6.1 *Idle CPU Utilization* Skenario Keenam

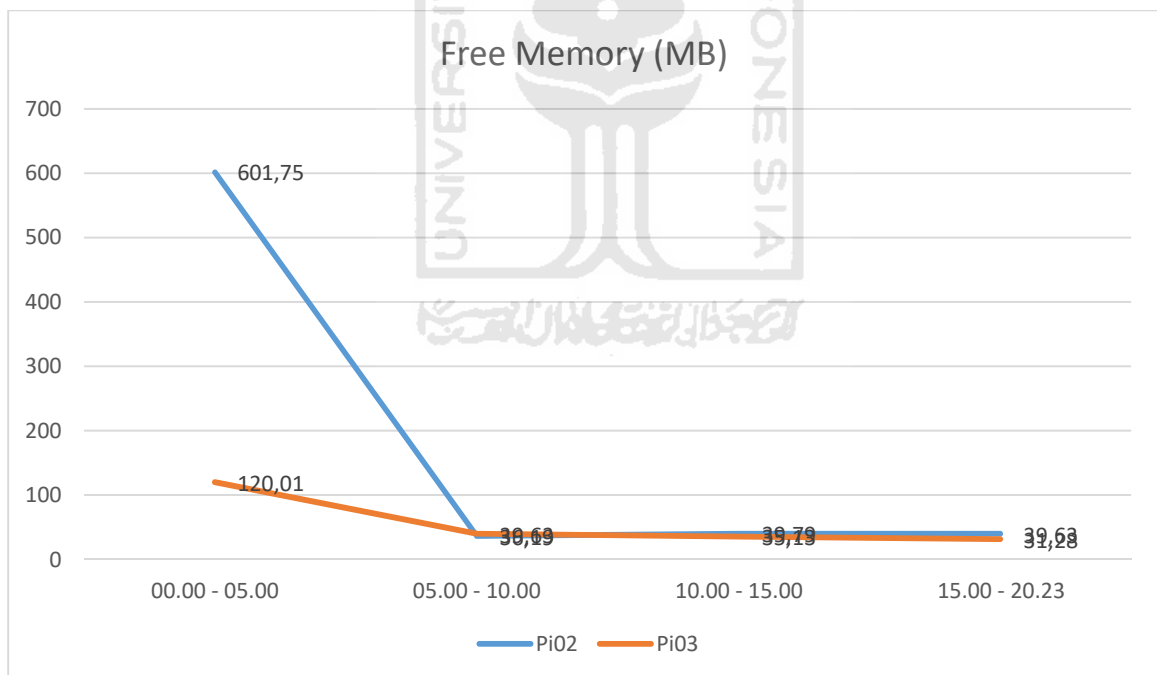


Diagram 4.8.6.2 *Free Memory* Skenario Keenam

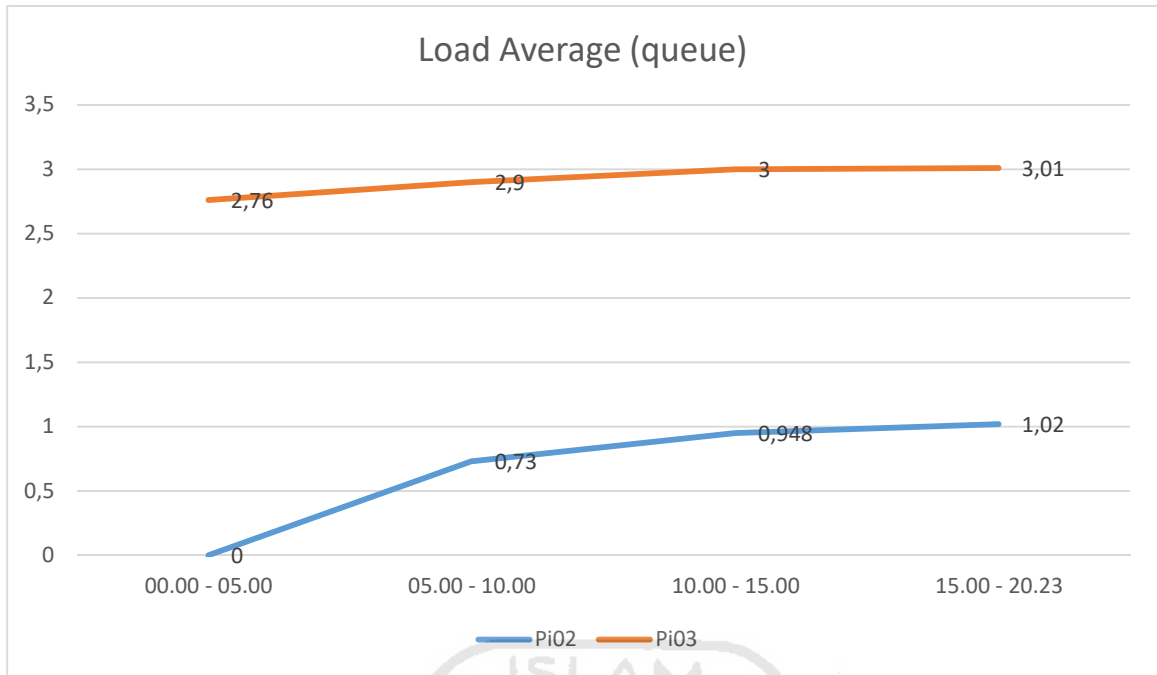


Diagram 4.8.6.3 Load Average Skenario Keenam

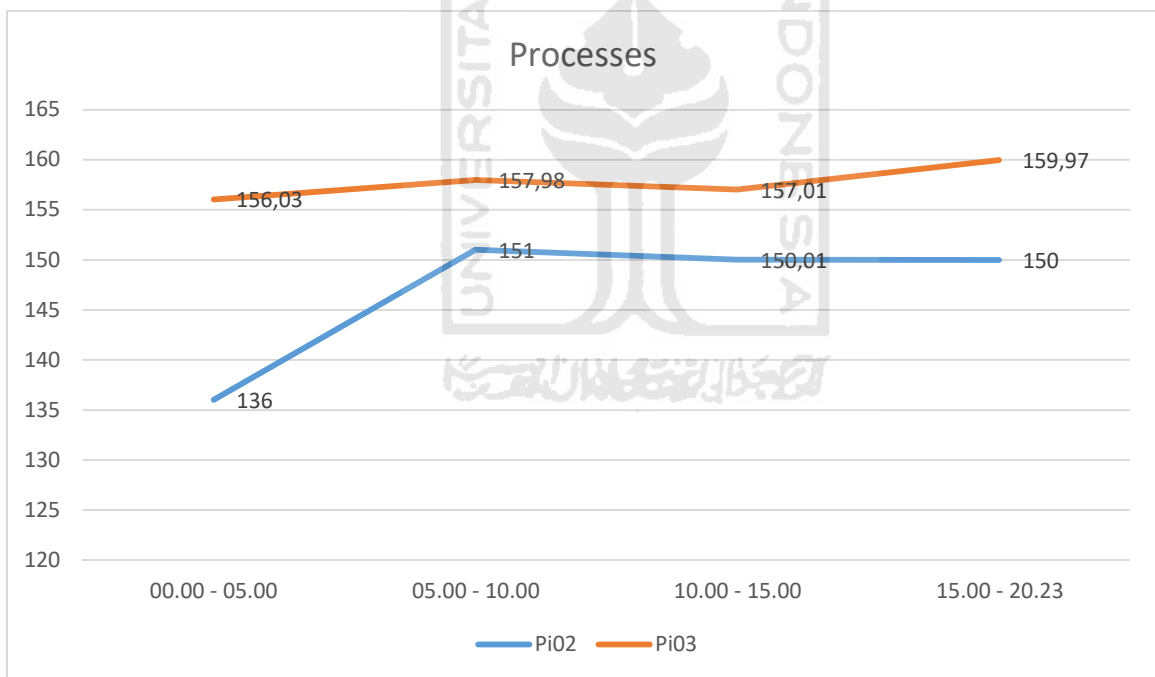


Diagram 4.8.6.4 Process Skenario Keenam

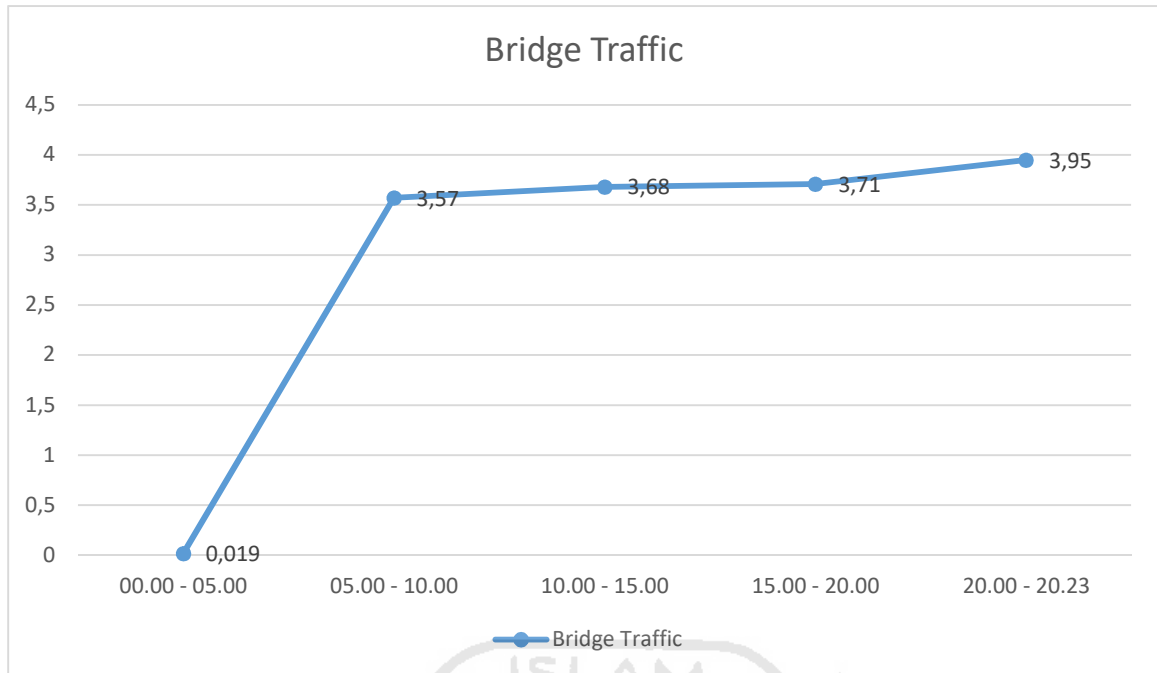


Diagram 4.8.6.5 Data *Traffic* Skenario Keenam

Skenario keenam ini mencatatkan perbedaan yang cukup tinggi tiap parameter pada dua *node* terkecuali pada *free memory*. Pada parameter *idle CPU utilization*, lima menit pertama Pi02 mendapatkan nilai 99 % dan mencatatkan rata rata 75 % hingga selesai pengujian sedangkan Pi03 mendapatkan 62,74 % pada lima menit pertama dan mencatatkan rata rata 25 % hingga pengujian selesai. Pada parameter *free memory*, kedua *node* mencatatkan nilai yang hampir mirip, walau pada lima menit pertama terjadi perbedaan nilai yang cukup tinggi hal ini dikarenakan Pi02 sebelumnya dalam kondisi *idle*. Pada load average memiliki selisih yang cukup tinggi antar *node*, dan pada Pi03 mencatatkan nilai yang sangat tinggi. Pada parameter processes kedua *node* terbilang cukup stabil walau Pi02 pada lima menit pertama merupakan hasil rata rata dengan kondisi *idle*. Untuk lalu lintas data rata rata berkisar antara 3,5MB/s hingga 3,9 MB/s. Dan waktu yang dibutuhkan sebanyak 20 menit 23 detik.

4.9 Pembahasan

Penelitian ini memiliki kesamaan pada *interface flash drive* yang digunakan sebagai bahan pengujian, yakni semua masih menggunakan USB 2.0 walau dengan merk yang berbeda. Tetapi pada penelitian sebelumnya, skenario pengujian dilakukan dengan menggunakan dua buah *flash drive*, satu buah sebagai sumber atau bahan untuk akuisisi dan satu buah sebagai

tujuan penyimpanan. Dengan mekanisme tersebut mendapatkan rata rata kecepatan transfer data sebesar 1,85 MB/s. Berikut hasil pengujian pada penelitian sebelumnya.

sebagai Dari semua skenario pengujian yang dilakukan, sistem kluster ini berhasil meningkatkan kecepatan akuisisi hingga dua kali lipat dibandingkan dengan penelitian sebelumnya. Dimana penelitian sebelumnya memiliki rata rata 1,85MB/s.

Nomor Percobaan	Input	Output	Durasi Proses Akuisisi	Kecepatan Transfer Data
1	Kingston 4GB USB 2.0	Sandisk 64GB USB 2.0	35 Menit	1,75 MB/s
2	ICRC 4GB USB 2.0	Sandisk 64GB USB 2.0	34 Menit	1,91 MB/s
3	Sandisk 8GB USB 2.0	Sandisk 64GB USB 2.0	73 Menit	1,69 MB/s
4	Toshiba 8GB USB 2.0	Sandisk 64GB USB 2.0	64 Menit	1,92 MB/s
5	Sandisk 16GB USB 2.0	Sandisk 64GB USB 2.0	129 Menit	1,93 MB/s
6	Toshiba 16GB USB 2.0	Sandisk 64GB USB 2.0	131 Menit	1,88 MB/s
Kecepatan Rata-rata Transfer Data				1,85 MB/s

Gambar 4.9.1 Hasil Pengujian Penelitian Sebelumnya

Dengan semua hasil pengujian pada penelitian sebelumnya, sistem kluster ini berhasil meningkatkan kecepatan transfer hingga dua kali lipat. Dimana rata rata sebelumnya 1,85 MB/s dan penelitian ini mencapai 3,5 MB/s hingga 4 MB/s.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melakukan pengujian dengan enam skenario pengujian dan memaparkan hasil serta pembahasan pada BAB IV, maka tahap berikutnya adalah penarikan kesimpulan terhadap apa yang diperoleh selama proses pengujian serta memberikan saran terhadap pengembangan penelitian selanjutnya.

- Penelitian ini berhasil merancang sebuah sistem kluster dengan menggunakan metode MPI.
- Dengan menggunakan MPI tersebut sistem kluster mampu mendistribusi beban ketika dilakukan akuisisi secara paralel. Sehingga ketika digunakan untuk mengakuisisi banyak perangkat menggunakan sistem kluster ini, maka dapat mempercepat waktu yang dibutuhkan untuk akuisisi dan juga dapat memberikan kecepatan transfer data yang lebih tinggi.
- Jika dibandingkan dengan penelitian sebelumnya, penelitian ini mampu memberikan peningkatan sebesar dua kali lipat.

5.2 Saran

Dalam serangkaian tahapan penelitian ini masih terdapat kekurangan, dimana kekurangan tersebut nantinya dapat digunakan untuk keperluan pengembangan penelitian yang akan datang. Adapaun kekurangan tersebut yang akan dijadikan saran untuk penelitian selanjutnya yaitu

1. Pengembangan penelitian kedepannya diharapkan mampu mengakuisisi dari satu sumber dan di proses secara paralel, kemudian disimpan pada satu tujuan penyimpanan.
2. Penelitian ini masih terbatas pada sumber daya perangkat yang digunakan, sehingga penelitian kedepannya bisa menggunakan sumber daya yang lebih baik
3. Agar memudahkan saat dioperasikan dan meningkatkat *user experience*, diharapkan mampu diimplementasikan ke dalam aplikasi berbasis GUI.



DAFTAR PUSTAKA

- Amato, F., Cozzolino, G., Moscato, V., & Moscato, F. (2019). Analyse digital forensic evidences through a semantic-based methodology and NLP techniques. *Future Generation Computer Systems*, 98, 297–307. <https://doi.org/10.1016/j.future.2019.02.040>
- Bhaskar, P. (2015). Raspberry Pi Home Automation With Wireless Sensors Using Smart Phone. *International Journal of Computer Science and Mobile Computing*, 45(5), 797–803.
- Hartanto, A. D., Utami, E., & Al, H. F. (2011). Penerapan teknik komputer forensik untuk pengembalian dan penghapusan berkas. *Teknologi*, 4(1), 36–44.
- Maulia, R. (2018). *Forensic Imaging Application Forensic Imaging Application*.
- Kelly, S. (2016). Python, PyGame and Raspberry Pi Game Development. *Python, PyGame and Raspberry Pi Game Development*, 3–5. <https://doi.org/10.1007/978-1-4842-2517-2>
- Kiepert, J. (2013). Creating a Raspberry Pi-Based Beowulf Cluster. *Boise State University*, 16.
- Levin, S. L., & Schmidt, S. (2014). IPv4 to IPv6: Challenges, solutions, and lessons. *Telecommunications Policy*, 38(11), 1059–1068. <https://doi.org/10.1016/j.telpol.2014.06.008>
- López, P., & Baydal, E. (2017). On a course on computer cluster configuration and administration. *Journal of Parallel and Distributed Computing*, 105, 127–137. <https://doi.org/10.1016/j.jpdc.2017.01.009>
- López, P., & Baydal, E. (2018). Teaching high-performance service in a cluster computing course. *Journal of Parallel and Distributed Computing*, 117, 138–147. <https://doi.org/10.1016/j.jpdc.2018.02.027>
- Muttil, N., Liong, S. Y., & Nesterov, O. (2007). A parallel shuffled complex evolution model calibrating algorithm to reduce computational time. *MODSIM07 - Land, Water and Environmental Management: Integrated Systems for Sustainability, Proceedings*, 1940–1946.
- Kaspersky. (2019). *IT Threat Evolution : Q3 2019*.
- Olson, E., & Shashidhar, N. (2016). Low Budget Forensic Drive Imaging Using Arm Based Single Board Computers. *Journal of Digital Forensics, Security and Law*, 11(1). <https://doi.org/10.15394/jdfsl.2016.1373>
- Saputra, A. P., & Widiyasono, N. (2017). Analisis Digital Forensik pada File Steganography (Studi kasus : Peredaran Narkoba). *Jurnal Teknik Informatika Dan Sistem Informasi*, 3(1), 179–190. <https://doi.org/10.28932/jutisi.v3i1.594>
- Sari, A. C., Rahayu, A., & Budiharto, W. (2015). Developing Information System of Attendance and Facebook Status for Binus University's Lecturer Using Raspberry Pi Architecture. *Procedia Computer Science*, 59(Iccsci), 178–187. <https://doi.org/10.1016/j.procs.2015.07.546>
- Smb, U., Sharing, D., & Samba, E. (2000). *Chapter 7 - - SMB for Linux and OpenVMS Using SMB with OpenVMS and Linux for Disk Sharing*.