

**PERANCANGAN, PEMBUATAN DAN ANALISIS
PURWARUPA *DISASTER RECOVERY DATABASE SERVER* DI
UNIVERSITAS NEGERI JAKARTA**



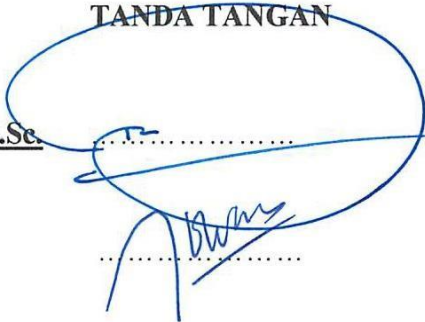
NUGROHO SAPUTRA

5235117081




**Skripsi ini Ditulis untuk Memenuhi Sebagian Persyaratan
dalam Memperoleh Gelar Sarjana**

**PENDIDIKAN TEKNIK INFORMATIKA DAN KOMPUTER
JURUSAN TEKNIK ELEKTRO FAKULTAS TEKNIK
UNIVERSITAS NEGERI JAKARTA
2015**

HALAMAN PENGESAHAN

NAMA DOSEN	TANDA TANGAN	TANGGAL
<u>M. Ficky Duskarnaen, M.Sc.</u> (Dosen Pembimbing I)		5/1/2016
<u>M. Djaohar, M.Sc</u> (Dosen Pembimbing II)		5/1/2016

PENGESAHAN PANITIA UJIAN SKRIPSI

NAMA DOSEN	TANDA TANGAN	TANGGAL
<u>Dr. Yuliatri Sastrawijaya, M.Pd</u> (Ketua Penguji)		17/12/2015
<u>Bambang Prasetya Adhi, M.Kom</u> (Sekretaris Penguji)		17/12/2015
<u>Lipur Sugianta, Phd</u> (Dosen Ahli)		9/1/2016

Tanggal Lulus:

HALAMAN PERNYATAAN

Dengan ini saya menyatakan bahwa:

1. Karya tulis saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik sarjana, baik di Universitas Negeri Jakarta maupun di perguruan tinggi lain.
2. Karya tulis ini adalah murni gagasan, rumusan dan penelitian saya sendiri dengan arahan dosen pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpanan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma yang berlaku di Universitas Negeri Jakarta.

Jakarta,

Yang membuat pernyataan



NUGROHO SAPUTRA

5235117081

Perancangan, Pembuatan dan Analisis Purwarupa *Disaster Recovery Database Server* Di Universitas Negeri Jakarta

NUGROHO SAPUTRA

ABSTRAK

Sebagai sebuah lembaga pendidikan yang mengelola banyak data penting seperti data akademik dan keuangan Universitas Negeri Jakarta belum memiliki sistem *realtime* yang dapat mem-*backup database*. Penelitian ini bertujuan untuk membangun dan menganalisis sebuah purwarupa dari *disaster recovery server database*. Penelitian ini dilakukan di ruang 102 PUSTIKOM UNJ dan ruang *server* gedung D PUSTIKOM UNJ, pada bulan Maret – Oktober 2015. Metode yang digunakan dalam penelitian ini adalah metode *Research and Development* dengan model pengembangan purwarupa atau *prototyping* dan model pengujian hasil purwarupa *disaster recovery database server* dengan metode *blackbox*. Pada penelitian ini dibatasi hanya pada *database* yang menggunakan MySQL. Setelah melakukan perancangan pada sistem, sistem diuji dengan melakukan simulasi peng-*input-an* data menggunakan *web server* yang terhubung dengan *database server*. Untuk menguji keandalan sistem *recovery* maka pada saat pengujian diujikan beberapa kriteria seperti peng-*input-an* biodata melalui *web server*, pengungahan 5 *file* yang berbeda ukuran dan *stress test* dengan jumlah data yang beragam pengujian juga dilakukan bertahap. Dengan tahapan pertama dengan menggunakan 5 *user* dan selanjutnya meningkat dengan *user* menjadi 20 *user*. Konfigurasi *my.cnf default* dengan modifikasi pada *server-id=1 bind-address=192.168.16.250* untuk *server master* dan modifikasi pada *server-id = 2 bind-address=192.168.16.240* untuk *server slave*, berdasarkan hasil pengujian dengan kondisi menggunakan 1 *user* yang melakukan peng-*input-tan*, peng-*update-an* dan pen-*delete-an* 5 buah biodata yang bertipe *text* tidak memiliki *delay* waktu dan memiliki tingkat keakuratan 100%. Dengan kondisi menggunakan 5 *user* yang melakukan peng-*input-tan*, peng-*update-an* dan pen-*delete-an* 10 buah biodata yang bertipe *text* database ada beberapa yang tiba bersamaan, ada satu *record* yang mengalami *delay*. Serta untuk melakukan peng-*upload-an file* 1kb tidak memiliki *delay* waktu, peng-*upload-an file* 1mb tidak memiliki *delay* waktu, peng-*upload-an file* 5mb memiliki *delay* waktu 1 detik, peng-*upload-an file* 10mb memiliki *delay* waktu 2 detik, peng-*upload-an file* 15mb memiliki *delay* waktu 3 detik, peng-*upload-an file* 20mb memiliki *delay* waktu 4 detik, peng-*upload-an file* 30mb memiliki *delay* waktu 6 detik dan peng-*upload-an file* 50mb memiliki *delay* waktu 10 detik saat melakukan replikasi *database* dan memiliki tingkat keakuratan 100% yang sama antara *database master* dan *slave* Delay replikasi database terpengaruh pada besarnya data dan banyaknya data yang di replikasi. Serta banyaknya *user* yang melakukan peng-*input-tan*, peng-*update-an* dan pen-*delete-an* pada suatu database dan *server slave* dapat langsung menggantikan peran *server master* ketika *server master* mengalami masalah atau *down*.

Kata kunci: *Disaster recovery server database*, MySQL, Replikasi

ANALYSIS AND PROTOTYPE DESIGN *DISASTER RECOVERY* *DATABASE SERVER* AT STATE UNIVERSITY OF JAKARTA

NUGROHO SAPUTRA

ABSTRACT

As an education institution which manage enormous important data such as academic and financial, State University of Jakarta hasn't yet own realtime system that can backup database. This research meant to build and analyzea prototype from disaster recovery server database. This research was conducted at room 102

PUSTIKOM UNJ and server room Building D PUSTIKOM UNJ, from March – October 2015. Research method that was used on is Research and Development with prototyping development model. This research was limited only with database that use MySQL. After conducting design on system, it was tested by doing data input simulation using web server that's connected to database server. To test the durability recovey system then during testing, some criteria such as inputing data, uploading 5 different file with different size and stress test with various amount of data. Testing was conducted on several steps, first using 5 users then increase into

20 users. My.cnf configuration default with modification on server-id=1 bind-address=192.168.16.250 for master server and modification server-id=2 bind-address=192.168.16.240 for slave server, based on test result with condition using

1 user which input, update, and delete 5 text-type biodata has no delay with accuracy 100%. with condition using 5 user which input, update, and delete 10 text-type there are some that arrive at the same time, there's one has delay. For uploading

1 kb file has no delay time, upload 1 mb has no delay time, upload 5 mb delay 1 second, upload 10 mb delay 2 second, upload 15 mb delay 3 second, upload 20 mb delay 4 second, upload 30 mb delay 6 second and upload 50 mb delay 10 second when doing database replication and has 100% accuracy between master and slave database. Replication delay was affected by the size of data and how many replication that was initiated. The amount of users that input, update, delete on a database and slave server can directly replace master server role upon facing problem or down.

Key words: Disaster recovery server database, MySQL, Replicatio

KATA PENGANTAR

Segala puji dan syukur kehadirat Allah SWT atas rahmat dan karunia-Nya yang telah diberikan kepada penulis, sehingga penulis dapat menyelesaikan laporan Penelitian dengan judul: “Perancangan, Pembuatan dan Analisis Purwarupa Disaster *Recovery Database Server* Di Universitas Negeri Jakarta” ini dapat terselesaikan dengan baik, dan tidak lupa shalawat dan salam untuk junjungan Nabi Muhammad SAW yang membawa umatnya menuju jalan yang lurus.

Penelitian ini merupakan wujud dari penyempurnaan Tri Dharma Perguruan Tinggi dan merupakan persyaratan untuk meraih gelar Sarjana Pendidikan Teknik Informatika dan Komputer pada Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Jakarta.

Saya mengucapkan terima kasih yang sebesar-besarnya kepada pihak-pihak yang telah membantu dalam kelancaran penelitian ini, diantaranya:

1. Rubiyo dan Yenny Agustina selaku orang tua yang selama ini telah membesarkan, memberi semangat, memberi motivasi, mendidik, menasehati, dan mendoakan sehingga skripsi ini dapat terselesaikan.
2. Drs. Wisnu Djatmiko, M.T. selaku Ketua Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Jakarta dan juga yang telah membantu banyak sehingga penulis bisa seperti saat ini.
3. Dr. Yuliatr Sastrawijaya, M.Pd. selaku Ketua Program Studi Teknik Informatika dan Komputer, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Jakarta.
4. Muhammad Ficky D., M.Sc selaku dosen pembimbing dan Kepala Pusat Teknik Informasi dan Komputer UNJ.
5. M. Djaohar, M.Sc selaku dosen pembimbing dan Wakil Ketua Pusat Teknik Informasi dan Komputer UNJ bidang pengembangan infrastruktur dan jaringan.
6. Hamidillah Ajie, M.T selaku dosen pembimbing akademik penulis dan Sekertaris Pusat Teknik Informasi dan Komputer UNJ. serta dosen yang

sangat banyak membantu perjalanan kuliah penulis, dengan arahan beliau sehingga penulis dapat seperti ini.

7. Mas Arya Adipurwa selaku staf PUSTIKOM yang banyak membantu dalam pemasangan, desain maupun konfigurasi sistem ini.
8. Sahabat – Sahabat ku tercinta yang tergabung dalam Tim langit Riza Danu Triatmoko, Yanuar Dwi Pramana, Septian Ricki Permana dan Didik Arianto Siswoyo yang telah banyak bekerja sama selama perkuliahan 4 Tahun Terakhir.
9. Sahabat – Sahabat ku tercinta yang tergabung dalam Tim Pejabat PTIK Andiaastika, Meilisa, Yunia, Jessy, Aditya. Serta sahabat ku Sherli yang selalu mendengarkan cerita ku baik dalam suka maupun duka.
10. Teman – teman satu angkatan TKJ Putra, Imade, Eko, Zaeri, Bedul dan seluruh tim poyo –poyo.
11. Rekan – rekan mahasiswa PTIK UNJ yang turut membantu dalam penyelesaian skripsi ini khusus nya yang telah membantu proses pengambilan data dalam skripsi ini.
12. Adik, kakak, dan teman – teman yang senantiasa mendoakan dan memberikan semangat hingga terselesaikannya skripsi ini.

Penulis mengharapkan saran dan kritik yang bermanfaat dan bersifat membangun sehingga penulis dapat mengembangkan pengetahuan dan memperbaiki kesalahan.

Akhir kata, penulis memohon maaf atas segala kesalahan yang mungkin ada dalam skripsi ini dan berharap semoga skripsi ini dapat bermanfaat bagi semua pihak serta dapat dikembangkan menjadi lebih baik lagi dikemudian hari.

Jakarta,

Penulis

Nugroho Saputra

DAFTAR ISI

	Halaman
HALAMAN PENGESAHAN	ii
HALAMAN PERNYATAAN	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiv
DAFTAR LAMPIRAN	xvi
 BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	4
1.3 Pembatasan Masalah	5
1.4 Perumusan Masalah Tujuan Penelitian	5
1.5 Tujuan Penelitian.....	6
1.6 Manfaat Penelitian	6
 BAB II KERANGKA TEORETIK, KERANGKA BERPIKIR, DAN HIPOTESIS	
2.1 Kerangka Teoretik.....	7
2.1.1 Definisi Jaringan Komputer	7
2.1.2 Jenis Jaringan Komputer	7
2.1.3 <i>Server</i>	9
2.1.4 Layanan <i>Server</i>	11
2.1.5 Disaster Recovery	17
2.1.6 <i>Database</i>	19

2.1.7 Sistem Basis Data Terdistribusi	21
2.1.7.1 Karakteristik Basis Data Terdistribusi	23
2.1.7.2 Keuntungan Sistem Basis Data Terdistribusi	23
2.1.8.3 Kekurangan Sistem Basis Data Terdistribusi	24
2.1.8.4 Prinsip Dasar Basis Data Terdistribusi	25
2.1.9 Metode Replikasi Basis Data Terdistribusi	26
2.1.9.1 Teknik <i>Single Master Replicated</i>	27
2.1.9.2 Teknik <i>Multi Master Replicated</i>	27
2.1.10 MySQL	28
2.2 Kerangka Berpikir	29
2.3 Hipotesis Penelitian	30

BAB III METODOLOGI PENELITIAN

3.1 Tujuan Penelitian	31
3.2 Tempat dan Waktu Penelitian.....	31
3.3 Metode Penelitian	31
3.3.1 Metode Pengembangan.....	33
3.3.2 Studi Pustaka	33
3.3.3 Analisis	34
3.3.4 Perancangan.....	34
3.3.5 Pengujian	34
3.3.6 Implementasi	34
3.4 Definisi Kebutuhan.....	35
3.5 Instrumen Penelitian	36
3.6 Tahapan Penelitian.....	37
3.6.1 Analisis Requirement Singkat	37
3.6.2 Perancangan <i>Web</i> untuk Pengujian	37

3.6.3 Rancangan <i>Database Website</i>	39
3.6.4 Topologi Pemasangan Perangkat Jaringan	43
3.7 Pengujian Produk.....	45
3.7.1 Tahapan Pengujian Fungsi Replikasi <i>Database</i>	45
3.7.1.1 Pengujian dengan Melakukan Pengisian Biodata	45
3.7.1.2 Prosedur Pengujian dengan Melakukan Pengunggahan Dokumen.....	47
3.7.1.3 Prosedur Pengujian dengan Melakukan <i>Stress Test</i>	48
3.7.2 Tahapan Pengujian Fungsi <i>Recovery database</i>	49
3.8 Teknik Analisis Data	52

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

4.1 Hasil Penelitian.....	53
4.1.1 Pengidentifikasian Masalah.....	53
4.1.2 Perancangan.....	54
4.1.2.1 <i>Use Case Diagram</i>	55
4.1.2.2 <i>Activity Diagram</i>	56
4.1.2.3 Instalasi <i>Server</i>	60
A. Sistem Operasi <i>Server</i>	60
B. Program	60
4.1.2.4 Pengaturan <i>Iptables</i>	61
4.1.2.5 Pengaturan <i>Mysql</i>	63
4.1.3 Uji Fungsional	79
4.1.4 Uji Replikasi	80
4.1.4.1 Uji Replikasi Kelompok Kecil	81
A. Pengisian Biodata.....	81
B. Pengunggahan Dokumen.....	82
C. <i>Stress Test</i>	84
4.1.4.2 Uji Replikasi Kelompok Besar	86

a. Pengisian Biodata	86
b. Pengunggahan Dokumen.....	87
c. <i>Stress test</i>	92
4.1.5 Uji Fungsi <i>Recovery Database</i>	97
4.1.6 Uji Waktu Replikasi <i>Database</i>	104
4.2 Pembahasan	106
BAB V KESIMPULAN DAN SARAN	
5.1 Kesimpulan	108
5.2 Saran	109
DAFTAR PUSTAKA	111
LAMPIRAN	112

DAFTAR TABEL

Tabel 3.1 Struktur Data Tbl_Users pada <i>Server Master</i>	40
Tabel 3.2 Struktur Data Tabel <i>File</i> pada <i>Server Master</i>	40
Tabel 3.3 Struktur Data Table Logdata pada <i>Server Master</i>	40
Tabel 3.4 Struktur Data Tabel Logfile pada <i>Server Master</i>	41
Tabel 3.5 Struktur Data Tabel Tbl_Users pada <i>Server Slave</i>	41
Tabel 3.6 Struktur Data Tabel <i>File</i> pada <i>Server Slave</i>	42
Tabel 3.7 Struktur Data Tabel Logdata pada <i>Server Slave</i>	42
Tabel 3.8 Struktur Data Tabel Logfile pada <i>Server Slave</i>	42
Tabel 3.9 Struktur Data Trigger Tdlogdata pada <i>Server Slave</i>	43
Tabel 3.10 Struktur Data Trigger Tglogfile pada <i>Server Slave</i>	43
Tabel 3.11 Tabel Instrumen pada Tes Isi Biodata.....	46
Tabel 3.12 Tabel Instrumen pada Tes Pengunggahan <i>File</i>	47
Tabel 3.13 Tabel Instrumen pada Tes <i>Stress Test</i>	49
Tabel 3.14 Tabel Isian pada Tes Pengujian Fungsi <i>Recovery Server</i> Isi Biodata	51
Tabel 3.16 Tabel Isian pada Tes Pengujian Fungsi <i>Recovery Server</i> pada Pengunggahan	51
Tabel 4.1 Hasil Uji Fungsional	80
Tabel 4.2 Hasil Uji Pengisian Biodata Kelompok Kecil.....	81
Tabel 4.3 Hasil Uji Pengunggahan Dokumen kelompok kecil	82
Tabel 4.4 Hasil Uji <i>Stress Test</i> Kelompok Kecil	84
Tabel 4.5 Hasil Uji Pengisian Biodata Kelompok Besar	86

Tabel 4.6 Hasil Uji Pengunggahan Dokumen Kelompok Besar.....	87
Tabel 4.7 Hasil Uji <i>Stress Test</i> Kelompok Besar.....	92
Tabel 4.8 Hasil Uji Waktu Replikasi Database saat Peng-input-an Biodata 5 dengan 1 User.....	97
Tabel 4.9 Hasil Uji Waktu Replikasi Database saat Peng-edit-an Biodata 5 dengan 1 User.....	98
Tabel 4.10 Hasil Uji Waktu Replikasi Database saat Peng-delete-an Biodata 5 dengan 1 User.....	98
Tabel 4.11 Hasil Uji Waktu Replikasi Database saat Peng-input-an 10 Biodata dengan 5 User.....	99
Tabel 4.12 Hasil Uji Waktu Replikasi Database saat Peng-edit-an Biodata 10 dengan 5 User.....	100
Tabel 4.13 Hasil Uji Waktu Replikasi Database saat Pen-delete-an Biodata 10 dengan 5 User.....	102
Tabel 4.14 Hasil Uji Waktu Replikasi Database saat Pengunggahan File....	103
Tabel 4.8 Hasil Uji Fungsi <i>Recovery</i> saat Pengisian Biodata	104
Tabel 4.9 Hasil Uji Fungsi <i>Recovery</i> saat Mengunggah <i>File</i>	105
Tabel 4.10 Hasil Uji Fungsi <i>Recovery</i> saat <i>Stress Test</i>	105

DAFTAR GAMBAR

Gambar 2.1 Gambar <i>Server Acer</i>	10
Gambar 2.2 Topologi <i>DHCP Server</i>	11
Gambar 2.3 Topologi <i>DNS Server</i>	12
Gambar 2.4 Topologi <i>FTP Server</i>	13
Gambar 2.5 Topologi <i>Web Server</i>	14
Gambar 2.6 Topologi <i>Mail Server</i>	14
Gambar 2.7 Topologi <i>Proxy Server</i>	15
Gambar 2.8 Topologi <i>Database Server</i>	16
Gambar 2.9 Topologi <i>SSH Server</i>	16
Gambar 2.10 Teknik <i>Single Master Replication</i>	27
Gambar 2.11 Teknik <i>Multi Master Replication</i>	28
Gambar 2.12 Kerangka Berpikir	29
Gambar 3.1 <i>Diagram Alir Metode Penelitian</i>	32
Gambar 3.2 Antarmuka <i>Web</i> pengujian	38
Gambar 3.3 Topologi Pemasangan Perangkat Jaringan.....	43
Gambar 4.1 <i>Use Case Diagram Website Pengujian Database</i>	55
Gambar 4.2 <i>Use Case Diagram Replikasi Database</i>	56
Gambar 4.3 <i>Activity Diagram Mengisi Biodata</i>	56
Gambar 4.4 <i>Activity Diagram Mengubah Biodata</i>	57
Gambar 4.5 <i>Activity Diagram Menghapus Biodata</i>	57
Gambar 4.6 <i>Activity Diagram Mengunduh File</i>	58
Gambar 4.7 <i>Activity Diagram Mengunggah File</i>	58

Gambar 4.8 <i>Activity Diagram</i> Replikasi <i>Database</i>	59
Gambar 4.9 Kode Perintah Konfigurasi <i>Iptables</i>	61
Gambar 4.10 <i>Iptables</i> pada <i>Server Database Master</i>	62
Gambar 4.11 <i>Iptables</i> pada <i>Server Database Slave</i>	62
Gambar 4.12 Konfigurasi <i>my.conf</i> <i>Server Database Master</i>	64
Gambar 4.13 Konfigurasi <i>my.conf</i> <i>Server Database Slave</i>	69
Gambar 4.14 <i>Master Status</i> Replikasi <i>Server Database Master</i>	74
Gambar 4.15 <i>Slave Status</i> Replikasi <i>Server Database Slave</i>	75

DAFTAR LAMPIRAN

Lampiran 1 : Profil Universitas Negeri Jakarta.....	91
Lampiran 2 : Profil PUSTIKOM UNJ	92
Lampiran 3 : Borang Pengujian Analisis Data.....	94
Lampiran 4 : Instalasi Sistem Operasi	96
Lampiran 5 : Pemasangan Program Pendukung	97
Lampiran 6 : Pemasangan dan Pengaturan MySQL	99
Lampiran 7 : Surat Izin Penelitian Skripsi di Pustikom.....	101
Lampiran 8 : Borang Peserta Pengujian Kelompok Kecil Replikasi	102
Lampiran 9 : Borang Peserta Pengujian Kelompok Besar Replikasi.....	105
Lampiran 10 : Tampilan <i>Website</i> Pengujian	110
Lampiran 11 : Pengecekan Kecocokan data Antara <i>Server Master</i> dan <i>Slave</i> ..	114
Lampiran 12 : Bukti Ping dan Input Data saat Pengujian <i>Recovery server</i>	116

BAB I

PENDAHULUAN

1.1 Latar Belakang

Manusia adalah makhluk sosial yang sangat membutuhkan komunikasi, komunikasi adalah hal yang pasti dilakukan setiap harinya. Hal itu telah berlaku sejak zaman dahulu dan seiring berjalannya waktu, alat komunikasi terus mengalami perkembangan, bisa dikatakan bahwa komunikasi merupakan hal yang terpenting atau vital bagi manusia. Tanpa komunikasi maka manusia bisa dikatakan tersesat dalam belantara kehidupan ini.

Dewasa ini perkembangan alat telekomunikasi semakin berkembang dan berubah bentuknya. Media komunikasi sudah makin maju, dan mampu memberikan pelayanan dan fungsi-fungsi yang lebih efektif dan efisien dalam berkomunikasi. Internet merupakan suatu ruang komunikasi terbaru yang hadir di awal abad 21, Internet kepanjangan dari *Interconnected Network*. Internet merupakan jaringan global yang menghubungkan komputer yang satu dengan lainnya di seluruh dunia. Dengan Internet, komputer dapat saling terhubung untuk berkomunikasi, berbagi dan memperoleh informasi. Dengan begitu maraknya informasi dan kegiatan di Internet, menjadikan Internet seakan-akan sebagai dunia tersendiri yang tanpa batas.

Membahas masalah Internet tak lepas dari kita berbicara tentang komputer yang terhubung menjadi satu jaringan besar yang terhubung satu sama lain, ada komputer yang berperan sebagai *server* dan ada juga yang bertindak sebagai *client*.

Server adalah suatu komputer yang merupakan sentral dari sebuah jaringan dalam sebuah jaringan yang berperan sebagai penyedia sumber informasi (*resource*) serta

layanan-layanan atau *service* dan *client* yang berperan sebagai penerima informasi yang telah disediakan oleh *server*.

Universitas Negeri Jakarta merupakan salah satu perguruan tinggi di Jakarta. Universitas Negeri Jakarta memiliki 23.000 mahasiswa S1, 7000 mahasiswa S2, 900 dosen dan 3000 karyawan mereka merupakan *client* yang harus dilayani oleh sebuah sistem komputer.

Data yang diolah dan dikelola oleh Universitas Negeri Jakarta menjadi hal yang sangat vital dikarenakan data tersebut berisi layanan akademik mahasiswa dan layanan keuangan universitas, sehingga membutuhkan sebuah sistem pengolah data yang handal, untuk itu harus adanya sistem *realtime* yang otomatis dapat mem-*backup database* yang tersimpan dalam sebuah *server*.

Selain itu dengan adanya *Peraturan Pemerintah Republik Indonesia Nomor 82 Tahun 2012 Tentang Penyelenggaraan Sistem Dan Transaksi Elektronik pada pasal 17 ayat (2) dan (3)* dengan Bunyi lengkap sebagai berikut:

(2) Penyelenggara Sistem Elektronik untuk pelayanan publik wajib menempatkan pusat data dan pusat pemulihan bencana di wilayah Indonesia untuk kepentingan penegakan hukum, perlindungan, dan penegakan kedaulatan negara terhadap data warga negaranya.

(3) Ketentuan lebih lanjut mengenai kewajiban penempatan pusat data dan pusat pemulihan bencana di wilayah Indonesia sebagaimana dimaksud pada ayat (2) diatur oleh Instansi Pengawas dan Pengatur Sektor terkait sesuai dengan

ketentuan peraturan perundang-undangan setelah berkoordinasi dengan Menteri.¹

Dikarenakan Universitas Negeri Jakarta merupakan suatu penyelenggara yang termasuk dalam kategori penyelenggara sistem elektronik pelayanan publik, maka pelayanan yang dilayani oleh PUSTIKOM Universitas Negeri Jakarta tersebut wajib memiliki pusat pemulihan bencana atau *data recovery center* memulihkan kembali data atau informasi serta fungsi-fungsi penting sistem elektronik yang terganggu atau rusak akibat terjadinya bencana yang disebabkan oleh alam atau manusia.

Saat *downtime* layanan terjadi ada kemungkinan *client* tidak mendapatkan balasan dari *request* dikarenakan *service* yang bersangkutan mengalami *error*. Bukan hanya itu, ada kalanya *server* mengalami *crash* yang menyebabkan konfigurasi dari *server* tersebut menjadi *error* dan juga memungkinkan sebagian data dan *files* yang tersimpan dalam *server* menjadi *corrupt* atau rusak. Tentu saja selalu ada seorang atau beberapa *administrator* jaringan yang bertugas untuk mengelola *server* agar dapat berfungsi dengan baik dan komunikasi dalam jaringan tetap lancar. Namun terkadang *admin* juga tak luput dari kesalahan sehingga data sebelum *downtime* tidak sesuai dengan data sesudahnya. Selain itu *data recovery center server* juga berguna jika suatu saat terjadi bencana seperti kebakaran, kerusakan *server* maka data yang tersimpan pada *server master* tidak hilang karena sudah dilakukan sinkronisasi *backup* secara otomatis dengan *server slave*.

¹ Peraturan Pemerintah Republik Indonesia Nomor 82 Tahun 2012 Tentang Penyelenggaraan Sistem Dan Transaksi Elektronik

Kegunaan atau manfaat *backup data* yaitu kita masih mempunyai cadangan data dari data yang hilang, rusak dan terhapus, baik yang disebabkan oleh kesalahan kita sendiri atau faktor lain di luar kemampuan kita seperti terkena *virus*, terkena *hacker*, *file* rusak, perangkat *server error* , bermasalah mengalami *trouble* , mati listrik, dan bencana alam.

Dengan adanya *data recovery center server* tersebut, diharapkan dapat menjaga ketersediaan jaringan dan data pada *server master*. Serta dapat mengurangi *downtime* jika suatu saat terjadi *error*, kerusakan dan kesalahan pada *server master*. Sehingga *admin* jaringan dapat segera memperbaiki kerusakan pada *server* dan data pada *server* tetap aman karena sudah ter-*backup*. Dengan begitu cadangan data yang sudah kita simpan tersebut dapat kita gunakan kembali sebagai pengganti data yang telah hilang, rusak dan terhapus. Fungsi *backup* data lebih mengacu pada faktor keamanan dan kenyamanan.

1.2 Identifikasi Masalah

Berdasarkan latar belakang diatas, masalah dapat diidentifikasi sebagai berikut:

1. Belum adanya *disaster recovery database server* yang ada di Universitas Negeri Jakarta.
2. Bagaimana mendesain dan menganalisis *disaster recovery database server* di Universitas Negeri Jakarta.
3. Cara pembuatan dan konfigurasi *server* untuk melakukan pembuatan *disaster recovery database server* pada *server database master* untuk tetap menjaga keutuhan data dan ketersediaan *server* di Universitas Negeri Jakarta.

1.3 Pembatasan Masalah

Melihat luasnya lingkup permasalahan yang diidentifikasi pada pembahasan sebelumnya, pembatasan masalah sangat penting untuk dilakukan. Penelitian dibatasi pada:

1. *Server* yang digunakan berada di pusat data PUSTIKOM UNJ.
2. Asumsi matinya *server master* akan dilakukan secara manual dengan cara memantikan *server* secara manual.
3. *Database* yang digunakan pada penelitian ini hanya *database* yang menggunakan MySQL.
4. Penelitian menggunakan tiga buah *server* yaitu *web server*, *server database master (dummy)* sebagai target yang akan di-*backup* dan *server database slave* merupakan *server slave* sebagai data *recovery center* yang digunakan untuk menyimpan data *backup* dari *server database master*.
5. *Server database master* sudah terpasang dan dikonfigurasi program untuk menentukan replikasi secara *realtime* serta sinkronisasi *database*.

1.4 Perumusan Masalah

Berdasarkan proses latar belakang, identifikasi, dan pembatasan masalah, maka perumusan masalah yang akan dibahas pada penelitian ini adalah:

1. Bagaimana cara membuat dan menganalisa purwarupa *disaster recovery database server* di Universitas Negeri Jakarta?

1.5 Tujuan Penelitian

Berdasarkan rumusan masalah diatas, maka tujuan yang hendak dicapai dalam penelitian ini adalah menghasilkan sebuah purwarupa *disaster recovery database server* di Universitas Negeri Jakarta.

1.6 Manfaat Penelitian

Adapun manfaat penelitian yaitu:

1. Memberikan kemudahan kepada admin jaringan dan orang-orang terkait dalam menghemat waktu jika terjadi masalah pada *server database master*.
2. Mengurangi resiko kehilangan data pada *server database master* apabila suatu saat terjadi kehilangan atau kerusakan data pada *server database master*.
3. Jika diterapkan di lokasi yang berbeda *disaster recovery database server* ini akan memberi keamanan data jika suatu saat terjadi hal yang tidak diinginkan seperti kebakaran. Maka data akan selamat sudah ter-*backup* otomatis di tempat lain.

BAB II

KERANGKA TEORETIK, KERANGKA BERPIKIR DAN HIPOTESIS

PENELITIAN

2.1. Kerangka Teoretik

2.1.1 Definisi Jaringan Komputer

Jaringan komputer adalah serangkaian beberapa komputer yang saling terhubung satu sama lain dengan tujuan berbagi sumber daya maupun informasi antara satu dengan yang lain. Jaringan komputer dibentuk dengan minimal 2 komputer yang terhubung satu sama lain dengan berbagai macam bentuk topologi yang bisa digunakan¹. Jaringan komputer bisa hanya berupa kumpulan beberapa komputer yang dikumpulkan di lokasi tertentu, dan dihubungkan menggunakan media tertentu (seperti kabel jaringan atau teknologi *wireless*). Bahkan hingga jaringan besar seperti Internet yang menggunakan berbagai media koneksi, seperti *mikrowave* dan teknologi satelit. Jaringan komputer yang dibentuk akan bisa digunakan untuk memindahkan data, suara atau bahkan *video* antar komputer.

2.1.2 Jenis Jaringan Komputer

1. Berdasarkan Luas Area

Berdasarkan luas areanya jaringan komputer terbagi menjadi 3 jenis, yaitu²:

1. *Local Area Network* (LAN)

LAN adalah jaringan yang dibatasi oleh area yang relatif kecil, umumnya dibatasi oleh area lingkungan, seperti sebuah kantor pada sebuah gedung,

¹ Iwan Sofana, Membangun Jaringan Komputer, (Bandung: Informatika, 2013), hlm. 3.

² Melwin Syafrizal, Pengantar Jaringan computer, (Yogyakarta, ANDI Yogyakarta, 200) hlm 16-17

atau tiap-tiap ruangan pada sebuah sekolah. Biasanya jarak antar *node* tidak jauh dari sekitar 200 m.

2. *Metropolitan Area Network* (MAN)

MAN biasanya meliputi area yang lebih besar dari LAN, misalnya antar gedung dalam suatu daerah (wilayah seperti provinsi atau negara bagian). Dalam hal ini jaringan menghubungkan beberapa buah jaringan kecil ke dalam lingkungan area yang lebih besar. Sebagai contoh, jaringan beberapa kantor cabang sebuah *bank* di dalam sebuah kota besar yang dihubungkan antara satu dengan lainnya.

3. *Wide Area Network* (WAN)

WAN adalah jaringan yang biasanya sudah menggunakan media *wireless*, sarana satelit, ataupun kabel serat *optic*, karena jangkauannya yang lebih luas, bukan hanya meliputi satu kota atau antarkota dalam suatu wilayah, tetapi mulai menjangkau area/wilayah otoritas negara lain. Biasanya WAN lebih rumit dan sangat kompleks bila dibandingkan LAN maupun MAN.

2. Berdasarkan Fungsi

Berdasarkan pola pengoperasian atau fungsi masing-masing komputer maka jaringan komputer dapat dibagi meliputi ³ :

1. *Peer to peer*

Peer to per adalah jenis jaringan komputer dimana setiap komputer bisa menjadi *server* sekaligus *client*. Setiap komputer dapat menerima dan memberikan *access* dari/ke komputer lain.

³ iwan sofana, CISCO CCNA dan Jaringan Komputer ,(Bandung, Informatika, 2009) hlm 101

Peer to peer banyak diimplementasikan pada LAN. Walaupun dapat juga diimplementasikan pada MAN dan WAN atau internet namun hal ini kurang lazim. Salah satu alasannya adalah masalah manajemen dan *security*. Cukup sulit mengawasi *security* pada jaringan *peer to peer* manakala pengguna jaringan komputer sudah sangat banyak.

2. *Client Server*

Client server adalah jaringan komputer yang salah satu (boleh lebih) komputernya difungsikan sebagai *server* untuk melayani komputer lain. Komputer yang dilayani *server* disebut *client*. Layanan yang diberikan bisa berupa *web*, *email*, *file* atau lainnya. *client server* banyak di pakai oleh Internet dan *intranet*.

2.1.3 *Server*

Sesuai dengan namanya, *server* bisa diartikan sebagai pelayan pada suatu jaringan komputer, dengan kata lain *server* adalah komputer yang berfungsi untuk melayani, membatasi, dan mengontrol akses terhadap *client-client* yang terhubung kepadanya dan sebagai pusat sumber daya yang dapat diakses melalui jaringan komputer, gambar dari perangkat *server* dapat di lihat pada gambar 2.1. *server* didukung spesifikasi/kemampuan *hardware* yang besar (berbeda dengan komputer biasa), seperti ⁴:

1. Ukuran dari memori atau *RAM* yang cukup besar untuk menampung jumlah *query* yang dijalankan oleh komputer yang terhubung. Hal ini dikarenakan komputer *server* memberikan layanan kepada sejumlah besar komputer maka dibutuhkan memori yang besar untuk mendukung tugas utamanya.

⁴ Fungsi dan jenis – jenis komputer server diakses di <http://jaringankomputer.org/server-fungsi-pengertian-jenis-jenis-komputer-server/>, tanggal 21 Oktober 2015 Pukul 15:25

2. Untuk mengelola data, dibutuhkan *prosesor* dengan kecepatan tinggi, kecepatan *prosesor* biasanya diukur dalam satuan *gigahertz*, tugas dari *prosesor* adalah untuk mengolah perintah yang diberikan dari komponen-komponen *server*. Sebuah *server* harus memiliki *prosesor* dengan kecepatan tinggi agar dapat bekerja secara optimal, oleh karena itu *prosesor* yang digunakan harus mampu memberikan fasilitas *multitasking*.
3. Kapasitas penyimpanan *harddisk* dari komputer *server* haruslah besar, agar dapat menyimpan semua data. Dalam sebuah jaringan komputer, *server* akan menyimpan data-data dari komputer *client*.



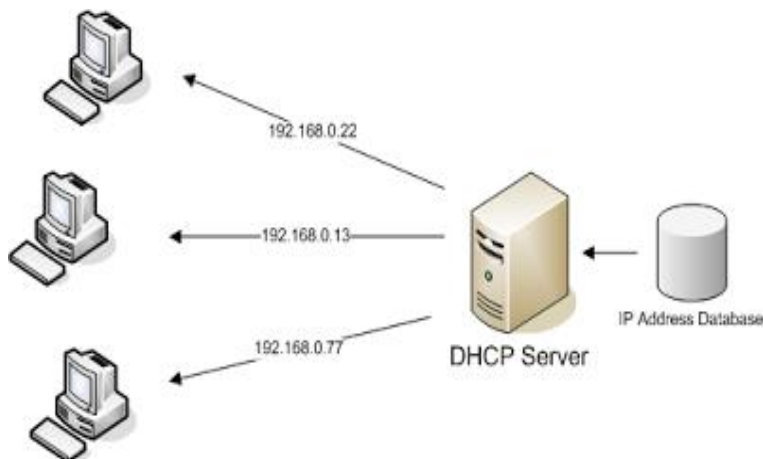
Gambar 2.1 Gambar Server Acer

2.1.4 Layanan Server

Berikut merupakan macam macam layanan / *service* yang biasanya dilayani oleh sebuah *server* yaitu ⁵:

1. DHCP Server

Sebuah *server* yang memberikan layanan pada DHCP atau yang merupakan singkatan dari *Dynamic Host Configuration Protocol*. DHCP merupakan suatu protokol yang mengatur pemberian alamat IP, *subnet mask*, *default router*, dan beberapa parameter lain pada komputer *client*, dengan pemberian alamat IP tersebut dilakukan secara otomatis (diberikan oleh suatu mekanisme dari *server*). Dengan terdaftar dan memiliki IP *address* tersebut maka komputer *client* tersebut dapat terhubung dengan jaringan lokal yang memberikan akses tersebut, sehingga dapat berkomunikasi dengan komputer-komputer lain yang berada dalam jaringan tersebut, untuk lebih jelasnya mengenai gambar topologi DHCP *server* dapat dilihat pada gambar 2.2.

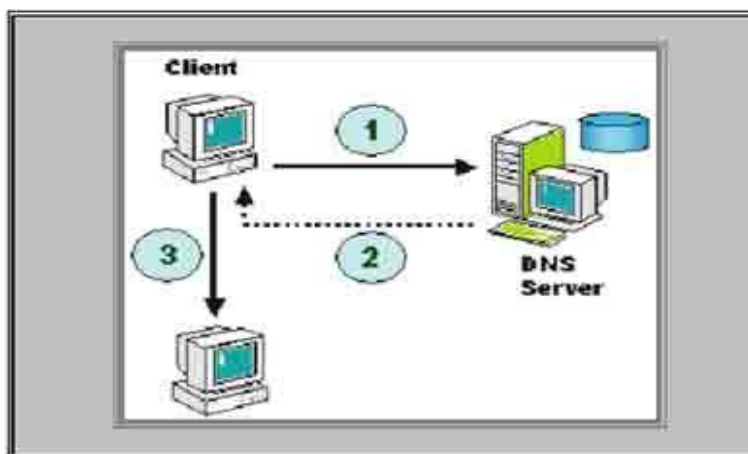


Gambar 2.2 Topologi DHCP Server

⁵ ahmad dkk, linux system administrator,(bandung,2010)hal 135

2. DNS Server

Merupakan sebuah *server* yang menyediakan layanan *Domain Name System*, suatu teknik untuk mengingat *IP address* yang terkadang sulit diingat karena terdiri dari sederetan angka. *Routing* paket IP yang berbasis TCP/IP sebenarnya tidak memerlukan teknik DNS, cukup dengan *IP address*. Teknik DNS diperlukan karena yang melakukan *routing* tidak lain adalah manusia, yang pada umumnya lebih mudah menghafalkan nama dibanding sederet angka. Untuk itu, agar Internet lebih mudah digunakan oleh manusia, diperlukan DNS untuk memetakan *IP address* menjadi nama *host/komputer* dan juga sebaliknya, untuk lebih jelasnya mengenai gambar topologi DNS *server* dapat dilihat pada gambar 2.3.

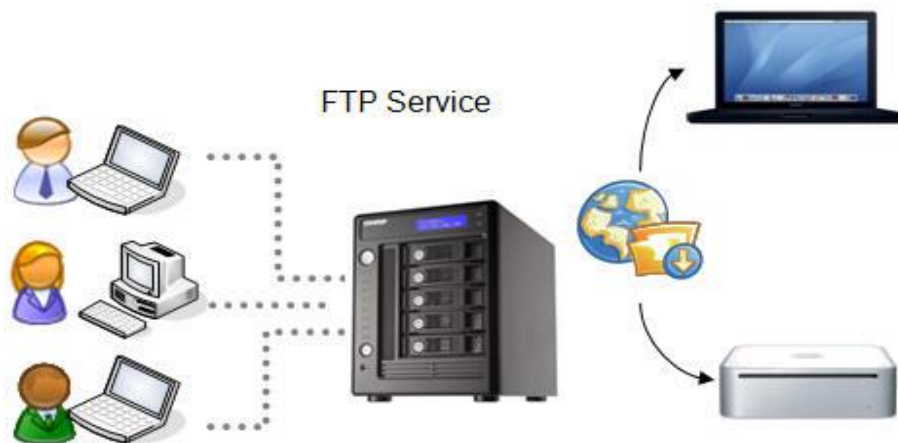


Gambar 2.3 Topologi DNS Server

3. FTP Server

FTP atau *File Transfer Protocol* adalah suatu *protokol* yang digunakan untuk mengirimkan atau memperoleh *file* dari *server* ke komputer lokal, dengan menggunakan *username* dan *password* sebagai informasi yang diperlukan untuk melakukan pengaksesan ke *server*. Untuk melakukan koneksi ke *server*,

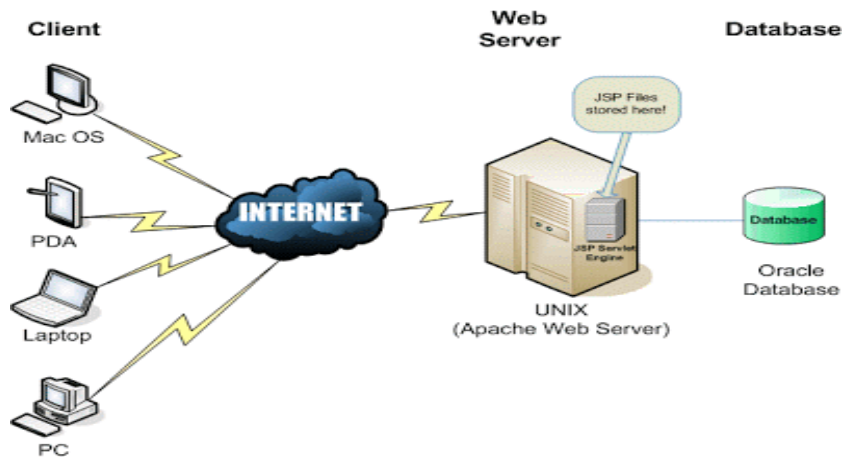
terbagi dengan dua tipe yaitu melalui *console* atau *shell*, dan juga melalui GUI atau *Graphic User Interface*. Layanan inilah yang disediakan oleh *FTP server*, untuk lebih jelasnya mengenai gambar topologi *FTP server* dapat dilihat pada gambar 2.4.



Gambar 2.4 Topologi FTP Server

4. Web Server

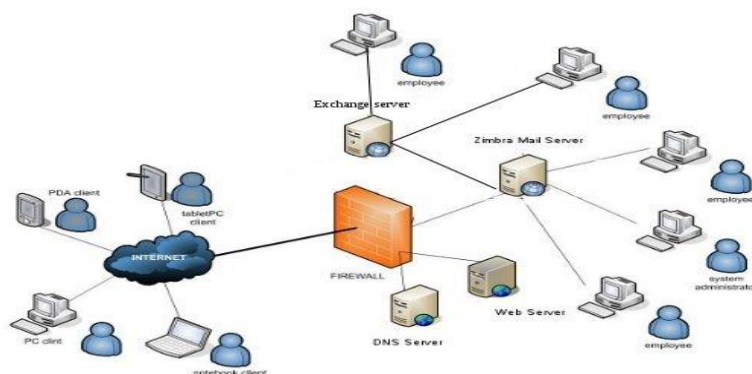
Web server adalah sebuah perangkat lunak *server* yang berfungsi sebagai penerima permintaan HTTP atau HTTPS dari *client* yang dikenal, dengan menggunakan *web browser* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman *web* yang umumnya berbentuk dokumen HTML. *Web server* yang terkenal diantaranya adalah *Apache* yang merupakan *web server multi-platform*, dan *Internet information Service (IIS)* yang hanya dapat beroperasi di sistem operasi berbasis *windows*, untuk lebih jelasnya mengenai gambar topologi *web server* dapat dilihat pada gambar 2.5.



Gambar 2.5 Topologi Web Server

5. Mail Server

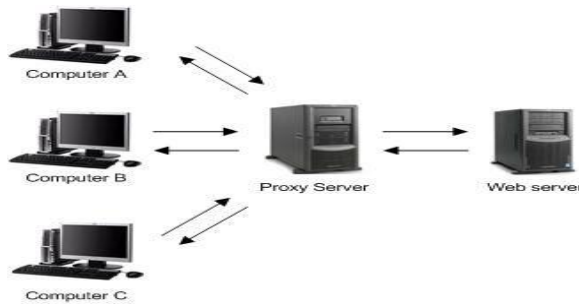
Mail server adalah suatu entitas berupa komputer yang bertindak sebagai sebuah penyedia layanan dalam jaringan komputer sendiri atau jaringan global (Internet). *server* ini berfungsi untuk melakukan penyimpanan dan distribusi yang berupa pengiriman, penjaluran, dan penerimaan *e-mail*. *Mail server* berjalan dengan beberapa *protokol* pada TCP/IP, yaitu SMTP(port 25), POP3(port 110), dan IMAP(port 143). *Mail server* memiliki tiga komponen utama yang menyusunnya, yaitu *Mail Transfer Agent (MTA)*, *Mail Delivery Agent (MDA)*, dan *Mail User Agent (MUA)*, untuk lebih jelasnya mengenai gambar topologi mail *server* dapat dilihat pada gambar 2.6.



Gambar 2.6 Topologi Mail Server

6. Proxy Server

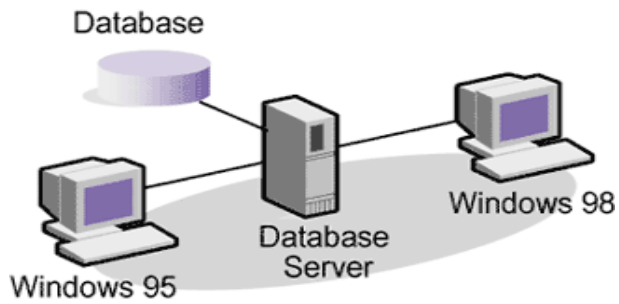
Aplikasi *server* yang dapat bertindak sebagai untuk melakukan *request* terhadap konten dari Internet atau *intranet*. *Proxy server* bertindak sebagai *gateway* terhadap dunia Internet untuk setiap komputer *client*. *Proxy server* tidak terlihat oleh komputer *client*. Seorang pengguna yang berinteraksi dengan Internet melalui sebuah *proxy server* tidak akan mengetahui bahwa sebuah *proxy server* sedang menangani *request* yang dilakukannya, untuk lebih jelasnya mengenai gambar topologi *proxy server* dapat dilihat pada gambar 2.7.



Gambar 2.7 Topologi Proxy Server

7. Database Server

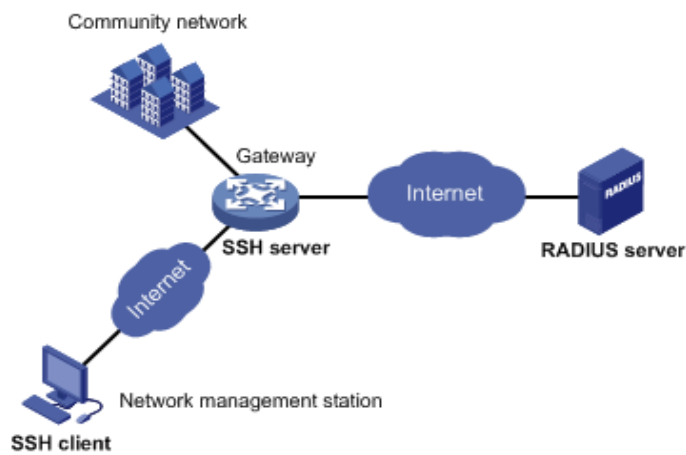
Sebuah program komputer yang menyediakan layanan pengelolaan basis data dan melayani komputer atau program aplikasi basis data yang menggunakan model *client/server*. Istilah ini juga merujuk kepada sebuah komputer (umumnya merupakan *server*) yang didedikasikan untuk menjalankan program yang bersangkutan. Sistem manajemen basis data (SMBD) pada umumnya menyediakan fungsi-fungsi *server* basis data, dan beberapa SMBD (seperti halnya MySQL atau *Microsoft SQL server*) sangat bergantung kepada model *client-server* untuk mengakses basis datanya, untuk lebih jelasnya mengenai gambar topologi *database server* dapat dilihat pada gambar 2.8.



Gambar 2.8 Topologi Database Server

8. SSH Server

SSH *server* merupakan aplikasi *server* yang berfungsi untuk melakukan *remote* atau pengendalian komputer dari jarak jauh, untuk lebih jelasnya mengenai gambar topologi *ssh server* dapat dilihat pada gambar 2.9.



Gambar 2.9 Topologi SSH Server

2.1.5 *Disaster Recovery*

Disaster recovery adalah perencanaan untuk pengelolaan secara rasional dan *cost-effective* bencana terhadap sistem informasi yang akan dan telah terjadi. Didalamnya terdapat aspek *catastrophe in information systems*⁶. Seperti halnya polis asuransi, suatu perencanaan preventif terhadap bencana pada sistem informasi dan pemulihan pasca bencana yang efektif harus dirasakan manfaatnya walaupun bencana tak pernah akan terjadi justru karena efektivitas sistem informasi tersebut. Namun runtuhnya sistem informasi itu sendiri merupakan bencana, terhentinya kegiatan sehari-hari karena kehilangan informasi.

Disaster recovery merupakan fasilitas yang berfungsi untuk mengambil alih fungsi suatu unit ketika terjadi gangguan serius yang menimpa satu atau beberapa unit kerja penting di perusahaan atau instansi, seperti pusat penyimpanan dan pengolahan data dan informasi. *disaster recovery* sudah bukan hal yang baru di dunia IT Indonesia, contoh pemanfaatannya adalah ketika terjadi malapetaka yang menimpa suatu perusahaan, seperti kebakaran, banjir dan lain sebagainya perusahaan tersebut akan lebih cepat kembali beroperasi, karena telah mempersiapkan *disaster recovery* untuk mengantisipasi bencana yang tidak dikehendaki tersebut. Secara umum *disaster recovery* berfungsi untuk meminimalisasi kerugian finansial dan nonfinansial dalam meghadapi kekacauan bisnis atau bencana alam meliputi fisik dan informasi berupa data penting perusahaan meningkatkan rasa aman di antara *personel, supplier, investor*, dan

⁶ Jon William Toigo, *Disaster Recovery Planning*, (Inc, Prentice Hall, 1989)

pelanggan.⁷

Penambahan *data center*, di lokasi yang aman dan terpisah digunakan hanya jika terjadi bencana, akan mengurangi efektifitas biaya dari masalah dan akan menjadi kekuatan untuk menduplikat biaya di lokasi, sumber daya, *link* komunikasi dan lainnya. Akan meng-*handle* semua tugas yang berhubungan dalam mengatur profesional *disaster recovery center*, sehingga bisnis dapat terus berjalan. *disaster recovery center* yang kita tawarkan akan mempunyai fungsi keamanan, dalam menjaga *data center* seperti mempunyai asuransi untuk bisnis *continuity*. Dibanyak institusi beberapa membangun *disaster recovery center* di Bali dan sebagian ada di Bandung sedangkan perusahaan yang bertaraf Internasional biasanya di Singapura atau di Hongkong contohnya BCA.

Ada 3 pilihan tipe *disaster recovery center* yang sesuai dengan kondisi alokasi anggaran suatu organisasi, yaitu antara lain ⁸:

1. *Cold Disaster Recovery Center*

Cold disaster recovery center ini menyediakan sistem yang sama seperti di lokasi *data center* di organisasi dimana aplikasi dan data akan di-*upload* sebelum fasilitas *disaster recovery center* bisa digunakan, namun proses pemindahan dari *data center* ke lokasi *disaster recovery center* akan dilakukan secara manual.

⁷ Disaster recovery center diakses di <http://www.te.net.id/v2/produk/disaster-recovery-center.aspx> tanggal 25 oktober 2015 Pukul 12:15

⁸ type disaster recovery center diakses, <http://www.te.net.id/v2/produk/disaster-recovery-center.aspx> tanggal 2 november 2015 Pukul 12:15

2. *Warm Disaster Recovery Center*

Warm disaster recovery center akan menyediakan komputer dengan segala komponennya, aplikasi, *link* komunikasi, serta *backup* data yang paling *update*, dimana *system* tidak otomatis berpindah tetapi masih terdapat proses manual meskipun dilakukan seminimal mungkin.

3. *Hot Disaster Recovery Center*

Hot disaster recovery center ini mengatur secepat mungkin operasional bisnis, sistem dengan aplikasi, *link* komunikasi yang sama sudah di pasang dan sudah tersedia di lokasi *disaster recovery center*, data secara *continuu* di-*backup* menggunakan koneksi *live* antara data *center* dan lokasi *disaster recovery center*, dan operasional bisnis akan berjalan pada saat itu juga, tanpa harus mematikan sistem di data *center* lama.

2.1.6 Database

Basis data (*database*) adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi. Basis data dimaksudkan untuk mengatasi masalah pada sistem yang memakai pendekatan berbasis berkas⁹.

Didalam suatu organisasi yang besar, sistem *database* merupakan bagian penting pada sistem informasi, karena diperlukan untuk mengelola sumber informasi pada organisasi tersebut. Untuk mengelola sumber informasi tersebut yang pertama kali dilakukan adalah merancang suatu sistem *database* agar informasi yang ada pada organisasi tersebut dapat digunakan secara maksimal.

⁹ Abdul Kadir, Pengenalan Sistem Informasi (Yogyakarta, Andi Offset, 2003)

Tujuan awal dan utama dalam pengolahan data pada sebuah basis data adalah agar dapat mencari data dengan mudah dan cepat. Disamping itu, pemanfaatan data untuk pengolahan data juga memiliki tujuan-tujuan tertentu. Pemanfaatan basis data dilakukan untuk memenuhi sejumlah tujuan sebagai berikut:

1. Kecepatan dan kemudahan (*Speed*)

Pemanfaatan basis data memungkinkan untuk dapat menyimpan data atau melakukan perubahan atau manipulasi terhadap data atau menampilkan kembali data tersebut dengan cepat dan mudah.

2. Efisiensi ruang penyimpanan (*Space*)

Penggunaan ruang penyimpanan di dalam basis data dilakukan untuk mengurangi jumlah *redundansi* (pengulangan) data, baik dengan melakukan penerapan sejumlah pengkodean atau dengan membuat relasi-relasi (dalam bentuk *file*) antar kelompok data yang saling berhubungan.

3. Keakuratan (*Accuracy*)

Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan atau batasan tipe data, *domain* data, keunikan data dan sebagainya diterapkan dalam basis data, sangat berguna untuk menentukan keakuratan pemasukan atau penyimpanan data.

4. Ketersediaan (*Availability*)

Pertumbuhan data (baik dari jumlah maupun jenisnya) sejalan dengan waktu akan semakin membutuhkan ruang penyimpanan yang besar. Data yang sudah jarang atau bahkan tidak pernah lagi digunakan dapat diatur untuk dilepaskan dari sistem basis data dengan cara penghapusan atau dengan memindahkannya ke media penyimpanan.

5. Kelengkapan (*Completeness*)

Lengkap atau tidaknya data yang dikelola bersifat relatif baik terhadap kebutuhan pemakai maupun terhadap waktu. Dalam sebuah basis data, struktur dari basis data tersebut juga harus disimpan. Untuk mengakomodasi kebutuhan kelengkapan data yang semakin berkembang, maka tidak hanya menambah *record* data, tetapi juga melakukan penambahan struktur dalam basis data.

6. Keamanan (*Security*)

Sistem keamanan digunakan untuk dapat menentukan siapa saja yang boleh menggunakan basis data dan menentukan jenis operasi apa saja yang boleh dilakukan.

7. Kebersamaan pemakai

Pemakai basis data sering kali tidak terbatas hanya pada satu pemakaian saja atau oleh satu sistem aplikasi saja. Basis data yang dikelola oleh sistem (aplikasi) yang mendukung lingkungan *multiuser*, akan dapat memenuhi kebutuhan ini, tetapi dengan menjaga atau menghindari terhadap munculnya persoalan baru seperti inkonsistensi data (karena data yang sama diubah oleh banyak pemakai pada saat bersamaan).

2.1.7 Sistem Basis Data Terdistribusi

Sistem basis data terdistribusi adalah suatu sistem basis data yang tersebar pada tiap *site*. Sistem basis data ini bersifat otonom atau dapat bekerja sendiri-sendiri tanpa bergantung salah satu *site*. Penjelasan lain menjelaskan bahwa “Basis data terdistribusi (*distributed database*) adalah basis data yang terdiri dari

kumpulan *site (instalasi)*, dihubungkan satu dengan yang lainnya menggunakan jaringan komunikasi, dimana¹⁰ :

1. Setiap *site* adalah merupakan basis data sendiri atau masing-masing *site* merupakan betul-betul satu basis data yang memiliki sendiri DBMS maupun perangkat lunak manajemen transaksi (termasuk *locking local*, *logging*, dan prosedur *recovery*), serta perangkat lunak komunikasi data.
2. Semua *site* setuju bekerja sama (bila perlu), sehingga pengguna pada *site* yang mana saja dapat mengakses data di dalam jaringan, seolah data yang diakses tersebut berada pada lokasi pengguna itu sendiri. Tempat penyimpanan ini dapat berada di satu lokasi yang secara fisik berdekatan (misal: dalam satu bangunan) atau terpisah oleh jarak yang jauh dan terhubung melalui jaringan internet”.

Penggunaan basis data terdistribusi dapat dilakukan di *server* Internet, *intranet* atau *ekstranet* kantor, atau di jaringan perusahaan. Untuk menjaga agar basis data yang terdistribusi tetap aktual, ada dua proses untuk menjaganya, yakni replikasi dan duplikasi. Dalam replikasi, digunakan suatu perangkat lunak untuk mencari atau lebih tepatnya melacak perubahan yang terjadi di satu basis data.

Setelah perubahan dalam satu basis data teridentifikasi dan diketahui, baru kemudian dilakukan perubahan agar semua basis data sama satu dengan yang lainnya. Proses replikasi memakan waktu yang lama dan membebani komputer karena kompleksitas prosesnya. Sementara itu, proses duplikasi tidak sama dan tidak sekompleks replikasi. Dalam proses ini, satu basis data dijadikan master, kemudian diperbanyak menjadi sejumlah duplikat. Selama proses duplikasi

¹⁰ M subkti. Sistem Manajemen Basis Data, (Bogor, Ghalia Indonesia,2004) hlm 186

berlangsung, perubahan hanya boleh dilakukan pada basis data master agar data lokal tidak tertimpa.

2.1.7.1 Karakteristik Basis Data Terdistribusi

Karakteristik dari sistem basis data terdistribusi ¹¹:

1. Kumpulan data yang digunakan bersama secara *logic* tersebar pada sejumlah komputer yang berbeda.
2. Komputer yang dihubungkan menggunakan jaringan komunikasi.
3. Data pada masing-masing *site* dapat menangani aplikasi-aplikasi lokal secara otonom.
4. Data pada masing *site* dibawah kendali satu Sistem Manajemen Basis Data (SMBD).
5. Masing-masing Sistem Manajemen Basis Data (SMBD) berpartisipasi dalam sedikitnya satu aplikasi global.

2.1.7.2 Keuntungan Sistem Basis Data Terdistribusi

Keuntungan dari sistem basis data terdistribusi ¹²:

1. Pengawasan distribusi dan pengambilan data, Jika beberapa *site* yang berbeda dihubungkan, seorang pemakai yang berada pada satu *site* dapat mengakses data pada *site* lain.

Contoh : sistem distribusi pada sebuah bank memungkinkan seorang pemakai pada salah satu cabang dapat mengakses data cabang lain.
2. *Reliability* dan *Availability*, Sistem terdistribusi dapat terus menerus berfungsi dalam menghadapi kegagalan dari *site* sendiri atau mata rantai komunikasi antar *site*.

¹¹ Evy Poerbaningtyas, Manajemen Sistem Terdistribusi, (Jakarta, Graha Ilmu, 2009) hlm 58

¹² *Ibid*, hlm 61

3. Kecepatan pemrosesan *query* , Contoh : jika *site-site* gagal dalam sebuah sistem terdistribusi, *site* lainnya dapat melanjutkan operasi jika data telah direplikasi pada beberapa *site*.
4. Otonomi lokal Pendistribusian, Sistem mengizinkan sekelompok individu dalam sebuah perusahaan untuk melatih pengawasan lokal melalui data mereka sendiri. Dengan kemampuan ini dapat mengurangi ketergantungan pada pusat pemrosesan.
5. *Efisiensi* dan *fleksibel*, data dalam sistem distribusi dapat disimpan dekat dengan titik dimana data tersebut dipergunakan. Data dapat secara dinamik bergerak atau salinannya dapat dihapus.

2.1.8.3 Kekurangan Sistem Basis Data Terdistribusi

Kekurangan dari sistem basis data terdistribusi ¹³:

1. Harga *software* mahal, hal ini disebabkan sangat sulit untuk membuat sistem basis data distribusi.
2. Biaya pemrosesan tinggi, perubahan pesan dan penambahan perhitungan dibutuhkan untuk mencapai koordinasi antar *site*.
3. Perancangan basis data lebih kompleks, sebelumnya menjadi keuntungan. Tetapi karena distribusi menyebabkan masalah sinkronisasi dan koordinasi, kontrol terdistribusi menjadi kerugian atau kekurangan dimasalah ini.

¹³ Evy Poerbaningtyas, Manajemen Sistem Terdistribusi, (Jakarta, Graha Ilmu, 2009) hlm 59

2.1.8.4 Prinsip Dasar Basis Data Terdistribusi

Prinsip dasar dari sistem basis data terdistribusi ¹⁴:

1. Otonomi lokal, suatu *site* dalam sistem terdistribusi harus memiliki otonomi. Otonomi lokal mempunyai arti bahwa semua operasi yang dilaksanakan pada suatu *site* sepenuhnya dikendalikan oleh *site* tersebut, tidak ada *site* yang tergantung pada *site* lain, ataupun suatu *site* lokal tergantung pada *site* pusat. Sehingga bila suatu *site* mengalami gangguan (*down*), maka *site* yang lain masih beroperasi normal.
2. Tidak bergantung pada *site* pusat, otonomi lokal juga mempunyai implikasi bahwa semua *site* harus diperlakukan setingkat, tidak ada ketergantungan pada *site* induk pusat untuk mendapatkan berbagai bentuk dukungan, seperti pengolahan *query* terpusat atau manajemen transaksi terpusat sedemikian rupa, sehingga seluruh sistem tergantung pada *site* pusat.
3. Operasi yang berlanjut, pada sistem basis data terdistribusi, seperti halnya pada sistem yang *non*-distribusi, secara ideal tentunya sistem tidak pernah dimatikan. Yaitu sistem tidak harus dimatikan pada kondisi apapun, untuk misalnya saja melakukan fungsi tertentu, misalnya dalam rangka penambahan instalasi, peningkatan DBMS dengan *versi* yang baru, dan lain sebagainya.
4. Transparan lokasi (*independensi lokasi*), pengertian dasar dari transparan lokasi (atau juga disebut *inpedensi lokasi*) cukup sederhana. Pengguna tidak perlu mengetahui dimana data tersimpan secara fisik, sebaliknya

¹⁴ M subkti. Sistem Manajemen Basis Data, (Bogor, Ghalia Indonesia,2004) hlm 189

pengguna harus dapat berlaku secara *logical* seakan data berada di *site* sendiri, meskipun secara fisik berada dimana saja.

2.1.9 Metode Replikasi Basis Data Terdistribusi

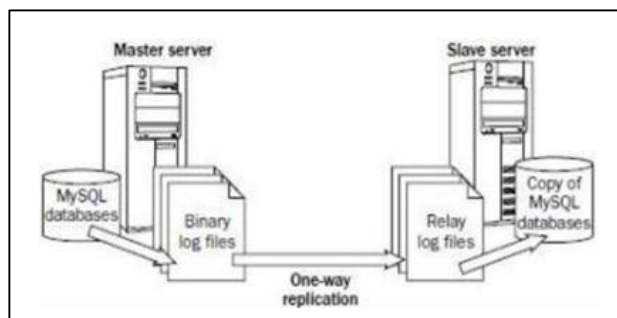
Replikasi adalah suatu teknik untuk melakukan *copy* serta pendistribusian data dan objek-objek basis data dari satu basis data ke basis data lain dan melaksanakan sinkronisasi antara basis data sehingga konsistensi data dapat terjamin. replikasi dapat dipahami sebagai teknik pengkopian basis data dan pengelolaan objek-objek basis data dalam suatu jaringan komputer yang dapat membentuk suatu sistem basis data terdistribusi untuk menjaga konsistensi data secara otomatis.

Jadi, dengan menggunakan teknik replikasi ini, data dapat didistribusikan ke lokasi yang berbeda melalui koneksi jaringan lokal maupun Internet. Replikasi juga memungkinkan untuk mendukung kinerja aplikasi, penyebaran data fisik sesuai dengan penggunaannya, seperti pemrosesan transaksi *online* dan DSS (*Decision Support System*) atau pemrosesan basis data terdistribusi melalui beberapa *server*. Dengan menggunakan teknik ini, kehandalan Basis Data akan lebih terjamin karena data dapat didistribusikan ke *server-server* yang lain.

Misalnya seperti transaksi yang terjadi pada *bank*, apa yang akan terjadi jika Bank hanya memiliki sebuah *database server*, apabila pada *database server* tersebut terjadi kerusakan atau padamnya arus listrik. Pasti transaksi perbankan akan berhenti total. Hal ini tidak demikian terjadi jika kita menggunakan basis data terdistribusi yang pada setiap transaksi juga direplikasi ke *server* yang lain.

2.1.9.1 Teknik *Single Master Replicated*

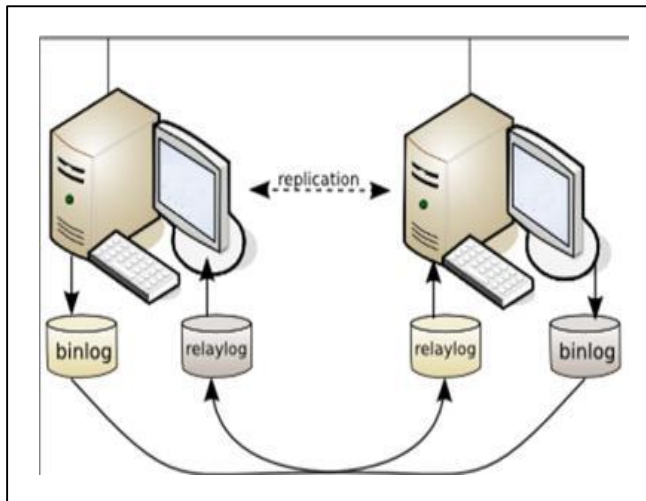
Dengan metode ini, salah satu komputer berfungsi sebagai *master* dan yang lainnya berfungsi sebagai *slave*. Pada prosesnya, komputer yang digunakan sebagai *server* akan dapat *read* dan *write* ke dalam *database*. Sedangkan komputer yang berfungsi sebagai *slave*, hanya akan *read* saja kedalam basis data tersebut. Apabila kita melakukan perubahan data pada *master*, maka otomatis data pada *slave* akan berubah. Tetapi jika kita melakukan perubahan data pada *slave*, basis data pada *master* tidak akan berubah. Untuk lebih jelasnya mengenai gambar teknik *single master replicated* dapat dilihat pada gambar 2.10.



Gambar 2.10 Teknik *Single Master Replication*

2.1.9.2 Teknik *Multi Master Replicated*

Dengan metode ini, salah satu komputer berfungsi sebagai *master server* dan yang lainnya berfungsi sebagai *master server* juga. Pada prosesnya, setiap komputer akan dapat *write* dan *read* data didalam *database*. Apabila kita melakukan perubahan data pada *master server* 1, maka otomatis data pada *master server* 2 akan berubah. Begitu juga jika kita melakukan perubahan data pada *master server* 2, basis data pada *master server* 1 akan berubah. Artinya setiap *master* 1 dan *master* 2 akan dapat mengubah dan menambah data pada basis data yang akan didistribusikan. Untuk lebih jelasnya mengenai gambar teknik *multi master replicated* dapat dilihat pada gambar 2.11



Gambar 2.11 Teknik *Multi Master Replication*

2.1.10 MySQL

MySQL (MY Structure Query Language) Adalah adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.¹⁵

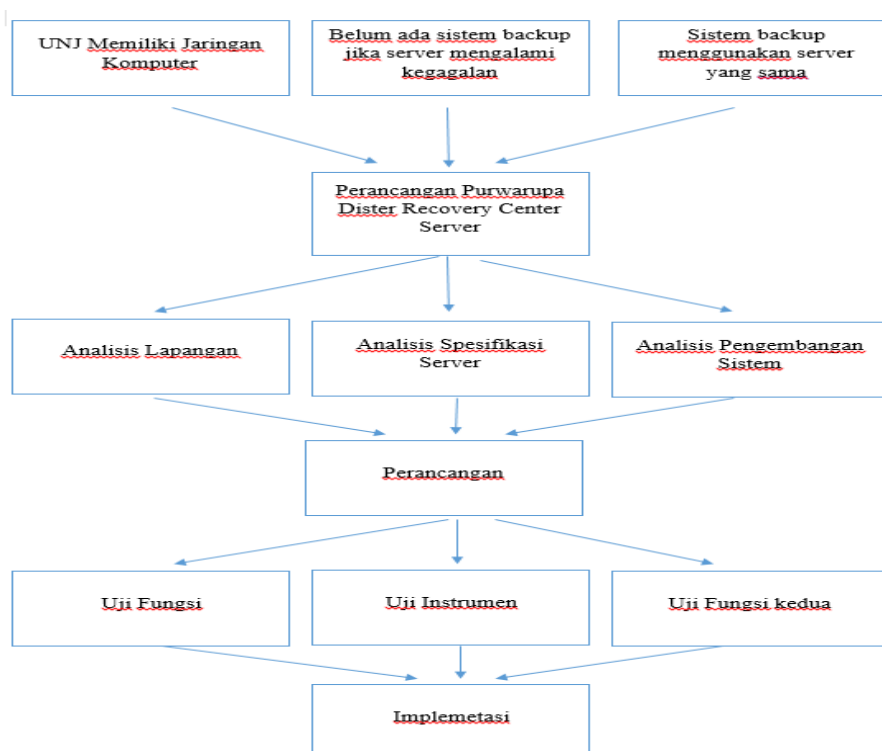
¹⁵ MySQL diakses, <https://id.wikipedia.org/wiki/MySQL> tanggal 12 november 2015 Pukul 14:23

2.1.11 Purwarupa Disaster Recovery Database server

Purwarupa *disaster recovery database server* di Universitas Negeri Jakarta adalah contoh awal dari sebuah pengembangan sistem pencegahan bencana terhadap sistem informasi di Universitas Negeri Jakarta yang menyelamatkan *database* yang tersimpan pada sebuah *server* dari kejadian yang tidak diinginkan seperti kerusakan atau kehilangan data yang dapat terjadi kapan saja. Serta dapat berfungsi untuk mengambil alih peranan fungsi suatu *database server master* jika terjadi gangguan serius dan *down* yang menimpa *server* tersebut.

2.2 Kerangka Berpikir

Dari uraian kerangka teori pada halaman sebelumnya dapat disusun suatu kesimpulan bahwa dalam melakukan pengembangan *recovery system* untuk *server* diperlukan adanya perencanaan yang baik. Adapun kerangka berpikir dari penelitian ini dapat divisualisasikan seperti pada gambar 2.12 :



Gambar 2.12 Kerangka Berpikir

Masalah utama yang dihadapi saat ini adalah belum adanya sistem *backup* yang diperlukan oleh *server* apabila suatu saat terjadi kesalahan pada proses operasi *server*. Karena itu diperlukan sebuah sistem *backup* yang bekerja dengan mensinkronkan data di dalam *server* sehingga bertindak sebagai *mirror* atau cermin.

Untuk mengetahui apa saja yang dibutuhkan maka pada tahap awal dilakukan analisa pada lapangan tentang kondisi *server* tersebut, spesifikasi *server*, dan pengembangan sistem itu sendiri. Setelah melalui tahap analisis maka selanjutnya adalah tahap perancangan yang bertujuan mengetahui apa saja fitur yang ada dalam sistem *recovery* ini. Langkah terakhir adalah pengujian yang terdiri dari 3 tahap, pengujian fungsi *backup* apakah berfungsi atau tidak, pengujian instrument yang digunakan untuk mengambil data pada sampel, dan pengujian fungsi pada populasi sampel.

2.3 Hipotesis Penelitian

Hipotesis sebagai suatu jawaban yang bersifat sementara terhadap masalah penelitian sampai terbukti melalui data yang akan terkumpul. Berdasarkan pendapat dan teori-teori yang telah disebutkan diatas, dapat ditarik sebuah hipotesis bahwa purwarupa *disaster recovery database server* dapat menjadi solusi untuk mencegah kehilangan data pada satu *database* dan bila terjadi masalah pada *server database master* maka *server database slave* akan otomatis menggantikan peranan *server database master* yang *down*.

BAB III

METODOLOGI PENELITIAN

3.1 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk merancang dan menganalisis purwarupa *disaster recovery database server* di Pusat Teknologi Informasi dan Komputer Universitas Negeri Jakarta.

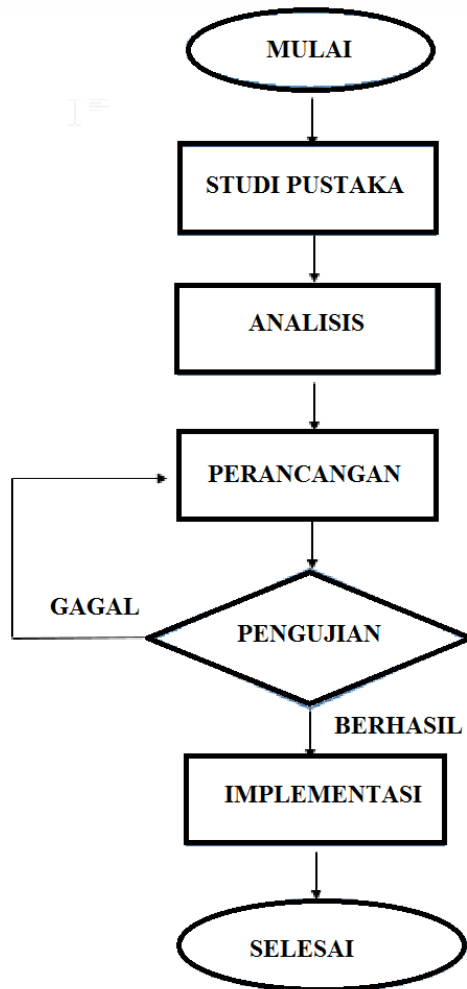
3.2 Tempat dan Waktu Penelitian

Penelitian dan pengembangan dilakukan di Pusat Teknologi Informasi dan Komunikasi Universitas Negeri Jakarta, adapun waktu penelitian dilaksanakan bulan April 2015 sampai dengan Desember 2015. Penulisan laporan skripsi ini dilaksanakan sejak September 2015.

3.3 Metode Penelitian

Metode penelitian yang digunakan pada penelitian ini adalah metode penelitian dan pengembangan (*Research & Development* atau *R&D*). Metode *Research & Development* adalah metode penelitian yang mengacu pada upaya yang diperlukan untuk menciptakan produk baru dan menguji keefektifan produk tersebut. tertentu dan menguji keefektifan produk tersebut¹. Metode *Research and Development* dapat dilihat diagram alir pada Gambar 3.1 .

¹ sugiono, metode penelitian kuantitatif, kualitatif dan R & D, (bandung, alfabeta, 2014) hlm 297



Gambar 3.1 Diagram Alir Metode Penelitian

Setelah disesuaikan dengan kebutuhan, maka tahapan-tahapan dari metode *R&D* yang diterapkan pada penelitian ini yaitu:

1. Mengidentifikasi masalah.
2. Mengumpulkan data atau studi literatur.
3. Mengembangkan purwarupa produk (*Design*).
4. Memvalidasi purwarupa produk.
5. Merivisi purwarupa produk.
6. Menguji purwarupa produk dalam kelompok kecil (*Testing*).
7. Merivisi purwarupa produk.

8. Menguji purwarupa produk dalam kelompok besar (*Testing*).
9. Merivisi purwarupa produk.
10. Memproduksi produk secara massal.

3.3.1 Metode Pengembangan

Metode pengembangan merupakan serangkaian tahapan yang dilakukan guna mengembangkan produk. Dalam penelitian pengembangan ini metode pengembangan yang digunakan adalah metode pengembangan purwarupa atau *prototyping*.

Purwarupa atau yang lebih dikenal dengan *prototyping* merupakan metode pengembangan lebih efektif dan efisien dalam memperbaiki dan mengoptimalkan produk yang dibuat melalui diskusi, eksplorasi, percobaan, dan perbaikan secara berulang-ulang sesuai dengan keinginan.

3.3.2 Studi Pustaka

Studi pustaka adalah teknik pengumpulan data dengan mengadakan studi penelaah terhadap buku – buku, literatur – literatur, catatan – catatan, dan laporan – laporan yang ada hubungannya dengan masalah yang dipecahkan. Dalam hal ini peneliti menggunakan referensi yang bersifat *online* maupun *offline* yang terkait dengan pembahasan *server*, replikasi *database*, *backup server* dan hal – hal terkait untuk pengembangan *server*. Oleh karena itu studi pustaka meliputi identifikasi teori secara sistematis, penemuan pustaka, dan analisis dokumen yang memuat informasi yang berkaitan dengan topik penelitian.

3.3.3 Analisis

Menurut Kamus Besar Bahasa Indonesia analisis merupakan penguraian suatu pokok atas berbagai bagiannya dan penelaahan bagian itu sendiri serta hubungan antar bagian untuk memperoleh pengertian yang tepat dan pemahaman arti keseluruhan. Analisis dalam penelitian ini berfokus pada hal – hal yang dibutuhkan dalam pembangunan sistem data *recovery server database*.

3.3.4 Perancangan

Perancangan adalah proses penerapan berbagai teknik dan prinsip yang bertujuan untuk mendefinisikan sebuah peralatan, satu proses atau satu sistem secara detail yang membolehkan dilakukan realisasi fisik. Kegiatan ini mengkonversikan hasil analisis kedalam suatu desain yang akan diimplementasikan, akan tetapi perlu pengujian untuk membuktikannya.

3.3.5 Pengujian

Pengujian merupakan proses membuktikan hasil perancangan apakah sudah berjalan sesuai dengan apa yang sudah dianalisis atau tidak, selain proses pembuktian juga dilakukan proses pencarian kesalahan. Melalui pengujian fungsi sistem replikasi *database* dapat diketahui fungsi – fungsi yang berjalan maupun tidak dan fungsi yang tidak berjalan dapat diperbaiki sesuai dengan tujuan awal fungsi tersebut.

3.3.6 Implementasi

Implementasi merupakan penerapan hasil rancangan setelah berhasil menyelesaikan pengujian, kemudian digunakan dalam fungsi sehari-hari.

3.4 Definisi Kebutuhan

Perangkat penelitian yang digunakan untuk membangun sebuah purwarupa *disaster recovery database server*. antara lain memerlukan perangkat keras dan perangkat lunak yang disarankan dalam menjalankan sistem ini adalah sebagai berikut :

1. Perangkat Keras terdiri dari :

1. Sebuah *server* Acer Altos G540 M2 Series sebagai *database server master* dengan spesifikasi sebagai berikut :

Processor : Intel® Xeon® CPU E5502 @ 1.87GHz

Memori : 16GB DDR3

Harddisk : 300GB

2. Sebuah *server* Acer Altos G540 M2 Series sebagai *database server slave* dengan spesifikasi sebagai berikut :

Processor : Intel® Xeon® CPU E5502 @ 1.87GHz

Memori : 16GB DDR3

Harddisk : 300GB

3. Sebuah *server* rakitan dengan *mainboard* Intel S2600CP sebagai *web server* dengan spesifikasi :

Processor : Intel® Xeon® E5 2690 v5

Memori : 64GB DDR3

Hardisk : 2 TB

4. *Switch* yang digunakan untuk menghubungkan *server* ke dalam jaringan.
5. Kabel UTP dan konektor RJ-45 sebagai alat penghubung antara *server* dan *switch*.

2. Perangkat Lunak

1. Sistem operasi linux CentOS 6.5 x86_64.
2. *Openssh-server-5.3p1-112.el6_7.x86_64* for Cantos *server*.
3. *MY SQL-5.1.73-5.el6_6.x86_64* for Cantos *server*.
4. *MY SQL-server-5.1.73-5.el6_6.x86_64* for Cantos *server*.
5. *PHP-5.3.3-46.el6_6.x86_64* for Cantos *server*.
6. *HTTPD-2.2.15-47.el6.centos.x86_64* for Cantos *server*.
7. Package phpMyAdmin-4.0.10.10-1.el6.noarch for Cantos *server*.
8. PuTTY Release 0.63 (ssh Client).
9. Google Chrome Version 46.0.2490.71 m.
10. Notepad ++ Versi 6.8.3.
11. Bitwise Ssh Client versi 6.4.

3.5 Instrumen Penelitian

Instrumen dalam penelitian ini terdiri dari 3 variabel bebas yaitu jumlah *user* yang meng-*input database*, banyak data saat peng-*input-an* dan jenis data yang dimasukan tersebut. Variabel bebas yang digunakan untuk jumlah *user* yang meng-*input* mulai dari satu orang, 5 orang dan 20 orang ,untuk banyak data saat peng-*input-an* mulai dari 5 data , 10 data 15 data 100 data hingga 5000 data secara bersamaan . untuk jenis data yang dimasukan antara lain *char*, dokumen dan tipe data lain nya.

Adapun variabel terikat dari penelitian ini yaitu kemampuan purwarupa *disaster recovery database server* dalam melakukan duplikasi *database* antara *server* utama dan *server* cadangan.

Metode yang digunakan dalam pengujian adalah metode *blackbox*. Jenis pengujian yang digunakan terdiri dari 3 jenis pengujian yaitu uji memasukan biodata secara bersamaan dengan jumlah 5, 10 dan 15. Uji mengunggah *file* dengan ukuran 1kb, 1mb, 5mb, 10mb dan 15mb. Uji *stress test* dengan jumlah data 10, 50, 100, 1000 dan < 5000.

3.6 Tahapan Penelitian

3.6.1 Analisis Requirement Singkat

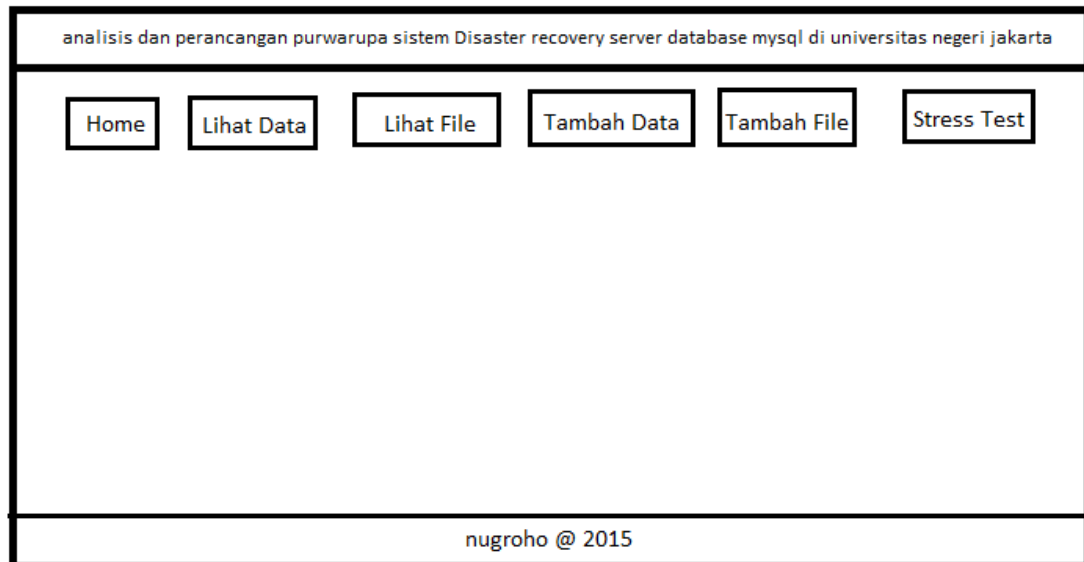
Tahap ini merupakan tahap awal untuk menentukan kebutuhan umum yang akan dibuat menjadi purwarupa dari sistem yang akan dibuat. Analisa dari kebutuhan dalam pengembangan purwarupa sistem yaitu :

1. Purwarupa *disaster recovery database server* dapat menyimpan data hasil masukan *user* dari *website* pengujian.
2. Purwarupa *disaster recovery database server* dapat melakukan replikasi *database* yang identik antara *master server* dengan *slave server*.
3. Purwarupa *disaster recovery database server* dapat secara langsung aktif otomatis ketika terjadinya gangguan pada *server database master*, menggunakan *database* yang ada dan identik pada *database server slave*.

3.6.2 Perancangan Web untuk Pengujian

Tahapan perancangan ini kita membuat sebuah *website* sederhana yang memiliki fungsi berupa pengisian biodata yang terhubung dengan *database server master*, nantinya *website* ini berfungsi untuk menambah, mengubah dan menghapus data oleh *user* untuk menguji replikasi *database* antara *server master* dengan *server slave*.

Adapun perancangan antarmuka pengguna dilakukan dengan membuat sketsa tampilan aplikasi. Adapun bentuk dari rancangan *user interface* secara umum ditunjukkan oleh Gambar 3.2 berikut :



Gambar 3.2 Antarmuka Web Pengujian

Pada *Website* untuk pengujian ini, isi atau konten dari *website* meliputi *headerweb*, *menubar*, *sidebar* berisi informasi dan navigasi setiap halaman dan konten utama. *Sub* halaman dari *Website* pemantauan jaringan meliputi:

- a. Halaman *Home* , merupakan tampilan awal dimana sebuah *website* pertama kali dibuka, menampilkan 6 tautan dari fungsi utama *website*, yaitu, Tambah data untuk menambah data ke *server database*, lihat data untuk melihat seluruh data yang ada di *database*, tambah *file* untuk menambahkan *file* ke *database*, lihat data untuk melihat *file* yang ada dalam *database* serta dapat pula berfungsi sebagai aman untuk *men-download file* yang ada dan *stress test* untuk memasukan data ke dalam *database* secara *random* dalam jumlah besar ke *server database*.

- b. Halaman lihat data, berisi tentang fungsi melihat data yang ada pada *server database* secara keseluruhan, apabila di buka terdapat pula *menu update* data dan *delete* data pada kolom kiri data yang akan di-*update* atau dihapus.
- c. Halaman Lihat *File*, berisi tentang fungsi melihat *File* yang ada pada *server database* secara keseluruhan. Dan juga memiliki fungsi *download* untuk *download File* yang ada pada *database*.
- d. Halaman Tambah data, berisi tentang fungsi menambah data untuk dimasukkan kedalam *database*.
- e. Halaman Tambah *file*, berisi tentang fungsi menambah *file* untuk dimasukkan kedalam *database*.
- f. Halaman *stress test*, berisi tentang fungsi menambah *data random* dengan jumlah tertentu yang akan di simpan ke *server database* hal ini digunakan untuk menguji replikasi *database* antara *server master* dan *slave server*.

3.6.3 Rancangan *Database Website*

Pada penelitian Purwarupa *disaster recovery database server*, dibutuhkan *file database* yang akan digunakan untuk menguji kehandalan replikasi *database*. Pada pengujian kali ini kita akan menggunakan *database* dengan nama *test* yang akan di rincikan *struktur* sebagai berikut :

1. *Database* pada *server master*

Nama *Database* : Test

Nama *Table* : *tbl_users*

Untuk struktur *Table* lebih lengkapnya dapat dilihat pada *Table 3.1*

Tabel 3.1 Struktur Data Tbl_Users pada Server Master

No	Field	Type data	Keterangan
1	Id	int(11)	auto_increment
2	nama	varchar(25)	
3	jenid	varchar(10)	
4	noid	varchar(20)	
5	warnar	varchar(30)	
6	jenkel	varchar(10)	
7	tanggallahir	Dae	
8	alamat	varchar(80)	
9	negara	varchar(15)	
10	prov	varchar(25)	
11	Kap	varchar(20)	
12	notelp	varchar(20)	
13	nohp	varchar(20)	
14	email	varchar(30)	
15	created	timestamp(6)	on update current_timestamp(6)

Nama Table : File

Untuk struktur Table lebih lengkapnya dapat dilihat pada Table 3.2

Tabel 3.2 Struktur Data Tabel File pada Server Master

No	Filed	The data	Keterangan
1	Id	int(10)	auto_increment
2	name	varchar(255)	
3	mime	varchar(50)	
4	Sie	bigint(20)	
5	Data	Longblob	
6	created	timestamp(6)	on update current_timestamp(6)

Nama Table : logdata

Untuk struktur Table lebih lengkapnya dapat dilihat pada Table 3.3

Tabel 3.3 Struktur Data Table Log Data pada Server Master

No	Filed	The data	Keterangan
1	Id	int(11)	auto_increment
2	nama	varchar(25)	
3	niod	varchar(50)	
4	Log	datetime(6)	

Nama *Table* : *logFile*

Untuk struktur *Table* lebih lengkapnya dapat dilihat pada *Table 3.4*

Tabel 3.4 Struktur Data Tabel Log File pada Server Master

No	Filed	The data	Keterangan
1	Id	int(10)	auto_increment
2	name	varchar(255)	
3	Sie	bigint(20)	
4	Log	datetime(6)	

2. Database pada *server slave*

Nama *database* : Test

Nama *Table* : *tbl_users*

Untuk struktur *Table* lebih lengkapnya dapat dilihat pada *Table 3.5*

Tabel 3.5 Struktur Data Tabel Tbl_Users pada Server Slave

No	Field	Type data	Keterangan
1	Id	int(11)	auto_increment
2	nama	varchar(25)	
3	jenid	varchar(10)	
4	Mok	varchar(20)	
5	warner	varchar(30)	
6	jenkel	varchar(10)	
7	tanggallahir	Dae	
8	alamat	varchar(80)	
9	negara	varchar(15)	
10	prov	varchar(25)	
11	Kap	varchar(20)	
12	notelp	varchar(20)	
13	nohp	varchar(20)	
14	email	varchar(30)	
15	created	timestamp(6)	on update current_timestamp(6)

Nama *Table* : *File*

Untuk struktur *Table* lebih lengkapnya dapat dilihat pada *Table 3.6*

Tabel 3.6 Struktur Data Tabel *File* pada *Server Slave*

No	<i>Filed</i>	The data	Keterangan
1	Id	int(10)	auto_increment
2	name	varchar(255)	
3	mime	varchar(50)	
4	Sie	bigint(20)	
5	Data	Longblob	
6	created	timestamp(6)	on update current_timestamp(6)

Nama *Table* : *logdata*

Untuk struktur *Table* lebih lengkapnya dapat dilihat pada *Table 3.7*

Tabel 3.7 Struktur Data Tabel *Logdata* Pada *Server Slave*

No	<i>Filed</i>	The data	Keterangan
1	Id	int(11)	auto_increment
2	Nama	varchar(25)	
3	niod	varchar(50)	
4	Log	datetime(6)	

Nama *Table* : *logFile*

Untuk struktur *Table* lebih lengkapnya dapat dilihat pada *Table 3.8*

Tabel 3.8 Struktur Data Tabel *LogFile* Pada *Server Slave*

No	<i>Filed</i>	The data	Keterangan
1	Id	int(10)	auto_increment
2	name	varchar(255)	
3	Sie	bigint(20)	
4	Log	datetime(6)	

Nama Trigger : tglogdata

Untuk struktur *triger* lebih lengkapnya dapat dilihat pada *Table 3.9*

Tabel 3.9 Struktur Data Trigger Tdlogdata Pada *server Slave*

Table	tbl_users
Time	AFTER
Even	Insert
Definition	INSERT INTO logdata (id, nama, noid, log) values (NEW.id, NEW.nama, NEW.noid, now(6))

Nama Trigger : tglogFile

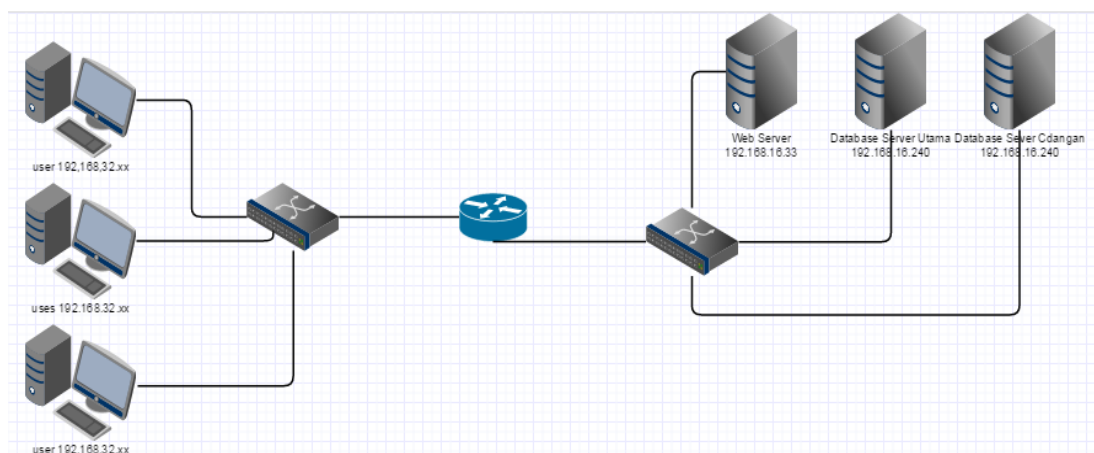
Untuk struktur *triger* lebih lengkapnya dapat dilihat pada *Table 3.10*

Tabel 3.10 Struktur Data Trigger TglogFile Pada *server Slave*

Table	File
Time	AFTER
Even	Insert
Definition	INSERT INTO logFile (id, name, size, log) values (NEW.id, NEW.name, NEW.size, now(6))

3.6.4 Topologi Pemasangan Perangkat Jaringan

Pada penelitian ini peneliti menggunakan topologi seperti gambar 3.3 dibawah ini.



Gambar 3.3 Topologi Pemasangan Perangkat Jaringan

Server database dan *web server* berada di dalam satu jaringan komputer yaitu pada *network* kelas C dengan *IP segment* 16 pada *oktet* ke 3 dalam pengalamatan IP, berikut adalah rincian konfigurasi pengalamatan IP pada masing – masing *server* :

1. *Web server*

IP Address : 192.168.16.33

Subnet Mark :255.255.255.0

Gateway : 192.168.16.1

2. *Database server master*

IP Address : 192.168.16.240

Subnet Mark :255.255.255.0

Gateway : 192.168.16.1

3. *Database server slave*

IP Address : 192.168.16.250

Subnet Mark :255.255.255.0

Gateway : 192.168.16.1

Ketiga *server* tersebut terhubung dengan *switch* yang sama, dan *switch* tersebut akan terhubung ke *router* yang akan menjembatani berkomunikasi dengan jaringan lainnya. pada saat pengujian *web server* akan di akses dari jaringan bersegment 32 pada *oktet* ke 3 dalam pengalamatan IP. dengan kata lain *web server* akan diakses dari jaringan luar yang bukan satu jaringan dengan *server* tersebut.

3.7 Pengujian Produk

Kriteria pengujian terhadap purwarupa *disaster recovery database server* didasarkan pada:

1. *Web* pengujian purwarupa *disaster recovery database server* dapat diakses melalui jaringan lokal UNJ.
2. *Web* pengujian purwarupa *disaster recovery database server* dapat melakukan manipulasi *database* baik menambah data, memperbaharui data, dan menghapus data secara bersamaan, dari beberapa *client* yang akan melakukan pengujian terhadap sistem replikasi.
3. Seluruh *database* baik *table*, *field* maupun *record* yang ada pada *master server* haruslah identik atau sama jumlah dan isi *database*, *table*, *field* maupun *record* pada *slave server*.
4. Pengujian pada purwarupa *disaster recovery database server* akan dilakukan dua kali pertama menggunakan 5 orang *user* dan kedua menggunakan 20 orang *user*.
5. Pada sub pengujian *recovery database*, asumsi *server* utama *down* dilakukan secara manual dengan cara mematikan *server* utama, setelah *server* utama mati baru di uji apakah sistem masih bisa bekerja secara baik diakses melalui *web server* atau tidak.

3.7.1 Tahapan Pengujian Fungsi Replikasi Database

3.7.1.1 Pengujian dengan Melakukan Pengisian Biodata

Untuk mengetahui apakah purwarupa *disaster recovery database server* dapat melakukan replikasi *database* secara *real time*, dapat diuji menggunakan *web* yang sengaja dibuat untuk menguji sistem tersebut dengan cara sebagai berikut:

1. Buka *web browser* yang ada di komputer.
2. Masukkan alamat lokal *server* yaitu : 192.168.16.33/skripsi.
3. Pilih *menu* tambah data pada *menubar* yang tersedia.
4. Masukkan dan lengkapi formulir biodata yang ada.
5. Khusus *field* no identitas di sini diberikan ketentuan khusus untuk peng-
input-an field sesuai *field* yang ditentukan.
6. Jika sudah tekan tombol simpan data.
7. Ulangi cara ke 4 dan ke 5 untuk meng-*input* data 5, 10 dan 15 data.
8. Lakukan jumlah peng-*input-an* sesuai dengan dengan yang diinstruksikan oleh peneliti saat pengambilan data berlangsung.
9. Setelah selesai, isi borang yang di sediakan oleh peneliti. Seperti *Table 3.12* yang ada di bawah. Pada dua karakter awal di isi dengan menggunakan nomor tempat duduk yang sudah di instruksikan oleh peneliti.

Tabel 3.11 Tabel Instrumen pada Tes Isi Biodata

Isi 5 Data Bersamaan	Isi 10 Data	Isi 15 Data

10. Saat *user* menggunakan ujicoba. Peneliti akan mencatat perubahan yang terjadi pada *server*.
11. Dan saat ujicoba selesai dilakukan peneliti akan menghitung ada tidaknya perbedaan antara jumlah data yang di-*input* pada saat pengujian dan data yang ada di *server* baik *server master* maupun *server Slave*.

3.7.1.2 Prosedur Pengujian dengan Melakukan Pengunggahan Dokumen

Pengujian selanjutnya dengan melakukan pengunggahan dari *web* pengujian yang diakses oleh *user* untuk mengunggah dokumen :

1. Buka *web browser* yang ada di komputer.
2. Masukkan alamat lokal *server* yaitu : 192.168.16.33/skripsi.
3. Pilih *menu* lihat *file*.
4. *Download* kelima data yang tersedia.
5. Duplikasi *file* tersebut dengan jumlah sesuai, pada kesempatan ini peneliti menentukan batas minimal pengunggahan yaitu 2 *file* dan maksimal pengunggahan yaitu 10 *file* untuk setiap masing masing *file*.
6. Beri nama *file* tersebut sesuai dengan arahan yang di berikan oleh peneliti, hal ini bertujuan untuk mempermudah proses pengambilan data.
7. Setelah itu pilih *menu* tambah *file* pada *menubar* yang tersedia .
8. Klik tombol *choose file*, maka akan terbuka kotak dialog yang berisi kumpulan *file* yang ada pada komputer kita.
9. Pilih *file* yang kita tadi sudah siapkan untuk di unggah, lalu *klik open*.
10. Setelah sudah *klik* tombol tambah *file* pada laman yang tersedia.
11. Tunggu beberapa saat sampai proses pengunggahan benar benar selesai.
12. Setelah selesai, isi borang yang di sediakan oleh peneliti. Seperti *table* 3.13 yang ada dibawah.

Tabel 3.12 Tabel Instrumen pada Tes Pengunggah File

Ukuran File	Jumlah*
1kb	
1mb	
5 mb	
10 mb	
15 mb	

12. Dan saat ujicoba selesai di lakukan peneliti akan menghitung ada tidaknya perbedaan antara jumlah data yang di-*input* pada saat pengujian dan data yang ada di *server* baik *server master* maupun *server slave*.

3.7.1.3 Prosedur pengujian dengan Melakukan *Stress Test*

Pengujian selanjutnya menggunakan *stress test*, hal ini sengaja dibuat agar *user* dapat memasukan jumlah data secara banyak dan *random* Untuk mengetahui apakah sistem replikasi *database* dapat melakukan *replikasi database secara real time*, dengan cara sebagai berikut:

1. Buka *web browser* yang ada di komputer.
2. Masukan alamat lokal *server* yaitu : 192.168.16.33/skripsi.
3. Pilih *menu stress test* pada *menubar* yang tersedia .
4. Masukan jumlah data yang akan di sesuai ketentuan yang dibagikan oleh peneliti.
5. Tekan mulai *input* untuk memulai *stress test* bersamaan sesuai arahan dari peneliti.
6. Tunggu berapa saat sampai program benar benar selesai membuat data acak baru.
7. Ulangi cara ke 2 sampai ke 6 jika ingin melakukan *stress test* lebih dari satu kali.
8. Isi borang yang di sediakan oleh peneliti seperti pada *table 3.14*.

Uji 3 *Stress Test*

Tabel 3.13 Tabel Instrumen pada Tes *Stress Test*

Uji Ke	Jumlah Data <i>Stress Test</i>
1	
2	
3	
4	
5	

9. Dan saat ujicoba selesai di lakukan peneliti akan menghitung ada tidak nya perbedaan antara jumlah data yang di-*input* pada saat pengujian dan data yang ada di *server* baik *server master* maupun *server slave*.

3.7.2 Tahapan Pengujian Fungsi *Recovery Database*

Selanjutnya adalah pengujian dengan sub penelitian *recovery database*, yang dimaksud dalam pengujian ini adalah kita akan mencoba melihat hasil dari replikasi *database* antara *server master* dan *slave*, apakah *server database* akan langsung menggantikan peran dan fungsi *database master* ketika *server database master* mangalami gangguan. Berikut dibawah ini dijelaskan prosedur pengujiannya :

1. Peneliti akan mencatat *record* terakhir yang ada pada *database master*, serta mencocokkan keidentikan kedua *database* antara *server master* dan *server save*
2. Selanjutnya peneliti akan mematikan *server database master* secara manual.
3. Pada kesempatan kali ini peneliti menggunakan tiga orang yang akan memasukan data ke *server*.
4. Selanjutnya buka *web browser* yang ada di komputer.
5. Masukan alamat lokal *server* yaitu : 192.168.16.33/skripsi.

6. Pastikan *web* pengujian *database* tetap berjalan dan cek fungsi tiap *menu*-nya
7. Pilih *menu* tambah data pada *menubar* yang tersedia .
8. Masukkan dan lengkapi formulir biodata yang ada .
9. Khusus *field* no identitas di sini diberikan ketentuan khusus untuk peng-*input-an field* sesuai *field* yang ditentukan.
10. Jika sudah tekan tombol simpan data.
11. Lakukan peng-*input-an* data sebanyak 5 kali. Dengan cara yang sama dengan mengulangi tahapan ke 7 sampai dengan 10.
12. Setelah selesai peneliti akan mencatat *record* yang tersimpan.
13. Setelah itu pilih *menu* tambah *file* pada *menubar* yang tersedia.
14. Klik tombol *choose file*, maka akan terbuka kotak dialog yang berisi kumpulan *file* yang ada pada komputer kita.
15. Pilih *file* yang kita tadi sudah siapkan untuk di unggah, laku *klik open*
16. Setelah sudah *klik* tombol tambah *file* pada laman yang tersedia
17. Tunggu beberapa saat sampai proses pengunggahan selesai.
18. Setelah selesai peneliti akan mencatat *record* yang tersimpan.
19. Pilih *menu stress test* pada *menubar* yang tersedia .
20. Masukkan jumlah data yang akan di sesuai ketentuan yang dibagikan oleh peneliti.
21. Tekan mulai *input* untuk memulai *stress test* bersamaan sesuai arahan dari peneliti.
22. Tunggu berapa saat sampai program benar benar selesai membuat data.

23. Lengkapi borang isian yang di sediakan oleh peneliti seperti *table 3.15* dan *Table 3.16* seperti yang ada dibawah ini.

Uji 1 Isi Biodata

Tabel 3.14 Tabel Isian Pada Tes Pengujian Fungsi *Recovery Server* Isi Biodata

No	Isian <i>Field No Id</i>
1	
2	
3	
4	
5	

Uji 2 *Upload File*

Tabel 3.16 Tabel Isian pada Tes Pengujian Fungsi *Recovery Server* pada Pengunggahan

no	nama <i>file</i>
1	
2	
3	
4	
5	

UJI 3 *Stress Test*

Banyak jumlah *stress test* yang di masukan :

24. Dan saat ujicoba selesai di lakukan peneliti akan menghitung data yang ada pada *server slave*.

3.8 Teknik Analisis Data

Dalam proses analisis data atau uji coba, purwarupa *disaster recovery database server* diuji dengan menggunakan metode *blackbox* dengan menggunakan jenis

pengujian fungsional dan *non-fungsional* dimana yang diuji hanya sebatas apakah sistem tersebut berfungsi dengan baik atau tidak sesuai *requirement* secara fungsional saja tanpa mengamati kinerja setiap kode program.

Adapun variabel dalam pengujian ini adalah jumlah *user* yang meng-*input* mulai dari satu orang, 5 orang dan 20 orang, untuk banyak data saat peng-*input*-an mulai dari 5 data , 10 data 15 data 100 data hingga 5000 data secara bersamaan . untuk jenis data yang dimasukan antara lain *char*, dokumen dan tipe data lain nya. Sedangkan jenis pengujian yang akan dilakukan yaitu uji kesamaan *record* antara *database server master* dan *slave* , uji perbandingan perbedaan waktu saat *record* tiba di *database*. Setiap pengujian dilakukan pada masing-masing variabel agar pengujian lebih *valid*, untuk borang pengujian lebih jelasnya dapat dilihat seperti pada lampiran 3.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1 Hasil Penelitian

4.1.1 Pengidentifikasian Masalah

Proses pengidentifikasian masalah diambil berdasarkan observasi secara langsung, adapun hasil dari proses observasi tersebut adalah:

1. Universitas Negeri Jakarta memiliki 23.000 mahasiswa S1, 7000 mahasiswa S2, 900 dosen dan 3000 karyawan mereka merupakan *client* yang harus dilayani oleh sebuah sistem komputer.
2. Data yang diolah dan dikelola oleh Universitas Negeri Jakarta menjadi hal yang sangat vital dikarenakan berisikan layanan akademik mahasiswa dan layanan keuangan universitas.
3. Dibutuhkan sebuah sistem pengolah data yang handal, untuk itu harus adanya sistem *realtime* yang otomatis dapat *membackup database* yang tersimpan dalam sebuah *server*.
4. Adanya *Peraturan Pemerintah Republik Indonesia Nomor 82 Tahun 2012 Tentang Penyelenggaraan Sistem Dan Transaksi Elektronik pada pasal 17 ayat (2) dan (3) yang mewajibkan Penyelenggara Sistem Elektronik Pelayanan Publik pemulihan bencana atau Data Recovery Center*, maka pelayanan yang dilayani oleh PUSTIKOM Universitas Negeri Jakarta tersebut wajib memiliki pusat pemulihan bencana atau *data recovery center* memulihkan kembali data atau informasi.

5. Belum adanya sistem *disaster recovery database server* untuk pencegahan kerusakan dan kehilangan data di Universitas Negeri Jakarta.
6. Data *center* di PUSTIKOM UNJ memiliki *server* yang dapat digunakan sebagai perangkat ujicoba dalam penelitian ini.

4.1.2 Perancangan

Dalam merancang purwarupa sistem *disaster recovery server database MySQL* untuk melakukan *backup server* maka diperlukan daftar fitur dan fungsi yang diperlukan untuk instalasi di dalam *server* dan *website* pengujian yang akan dibangun.

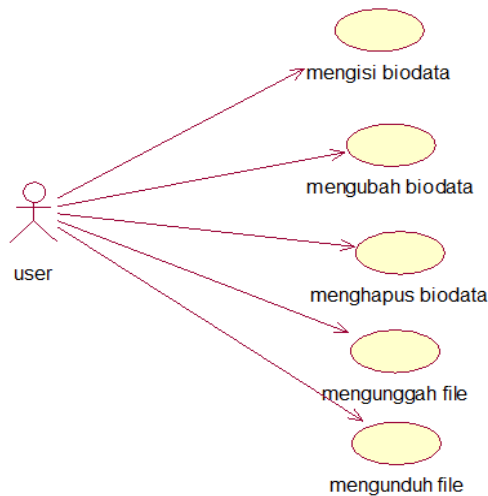
Berikut daftar fitur yang dibuat pada produk :

1. Pada perangkat *server*
 1. *Server web* merupakan gerbang utama *client* untuk melakukan *input, update* dan *delete* data.
 2. Data pada *server website* tersimpan ke *server database master*.
 3. *Server database master* dapat melakukan replikasi *database* ke *server slave*.
 4. Jika *server database master* tidak dapat diakses atau down maka *server database slave* dapat menggantikan *server* utama.
2. Pada *website* pengujian
 1. Halaman utama sebagai antarmuka awal saat *website* dibuka.
 2. Halaman lihat data untuk melihat data biodata yang tersimpan pada *server database master*.
 3. Halaman lihat *file* untuk melihat data *file* yang tersimpan pada *server database master*.
 4. Halaman tambah data untuk menambahkan biodata baru dan akan tersimpan pada *server database master*.

5. Halaman tambah *file* untuk menambahkan *file* baru dan akan tersimpan pada *server database master*.
6. Halaman *Stress test* untuk melakukan *stress test* untuk menguji *database*.

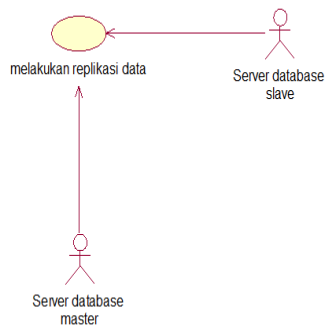
4.1.2.1 Use Case Diagram

Use case diagram dirancang untuk menggambarkan skenario penggunaan secara spesifik dalam bahasa yang jelas dari sudut pandang aktor yang telah didefinisikan sebelumnya. dari *use case* diagram di bawah dapat dilihat bahwa aktor yang dalam kasus ini adalah *user*, dapat melakukan mengisi biodata, mengubah biodata, menghapus biodata, mengunggah *file*, dan mengunduh *file*. bentuk rancangannya dapat dilihat pada gambar 4.1 berikut :



Gambar 4.1 Use Case Diagram Website Pengujian Database

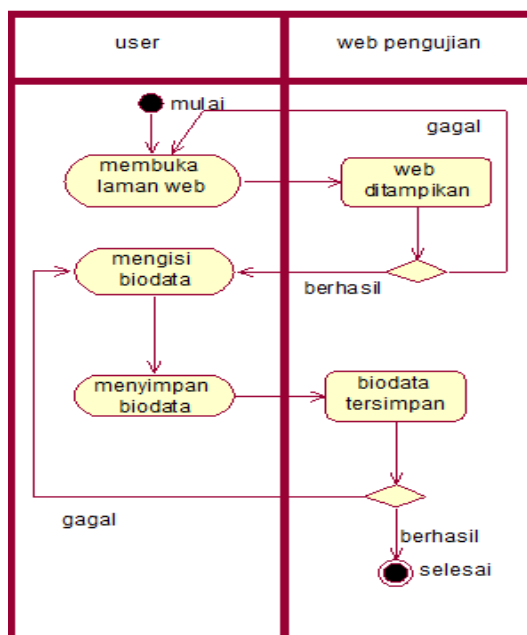
Sedangkan pada *use case* diagram dihalaman selanjutnya dapat dilihat bahwa *server database master* dan *server database slave* dapat melakukan replikasi. Bentuk rancangannya dapat dilihat pada gambar 4.2 berikut:



Gambar 4.2 Use Case Diagram Replikasi Database

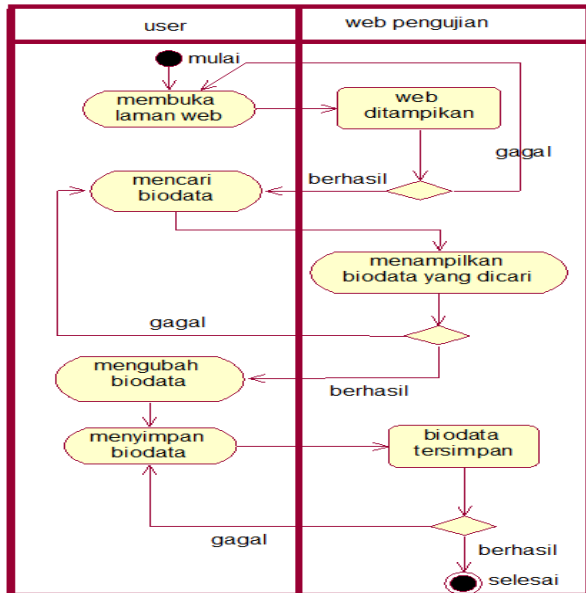
4.1.2.2 Activity Diagram

Activity diagram dirancang guna menjelaskan detail dari proses tiap *Use Case* diagram. Perancangan Activity diagram dibuat dengan sudut pandang masing – masing aktor pada setiap skenario. Adapun bentuk rancangan Activity diagram secara umum untuk Activity diagram mengisi biodata ditunjukkan oleh gambar 4.3 berikut.



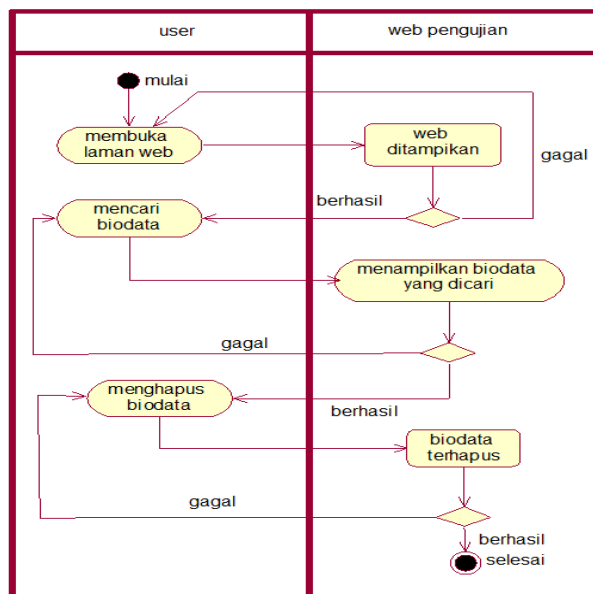
Gambar 4.3 Activity Diagram Mengisi Biodata

Adapun bentuk rancangan *activity* diagram mengubah biodata ditunjukkan oleh gambar 4.4 berikut :



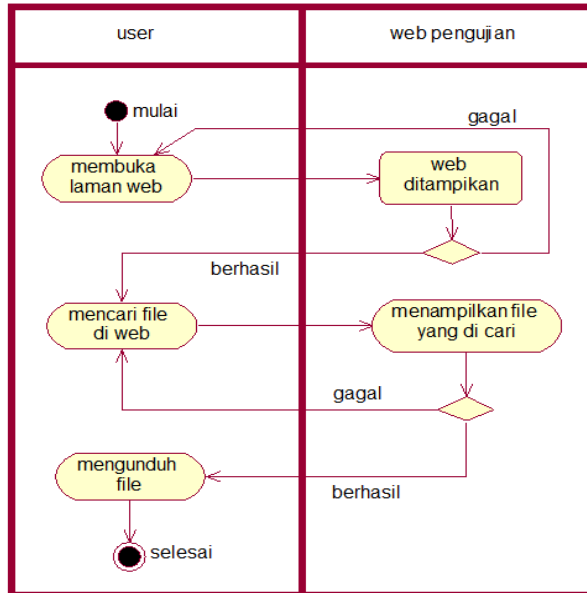
Gambar 4.4 Activity Diagram Mengubah Biodata

Adapun bentuk rancangan *activity* diagram menghapus biodata ditunjukkan oleh gambar 4.5 berikut :



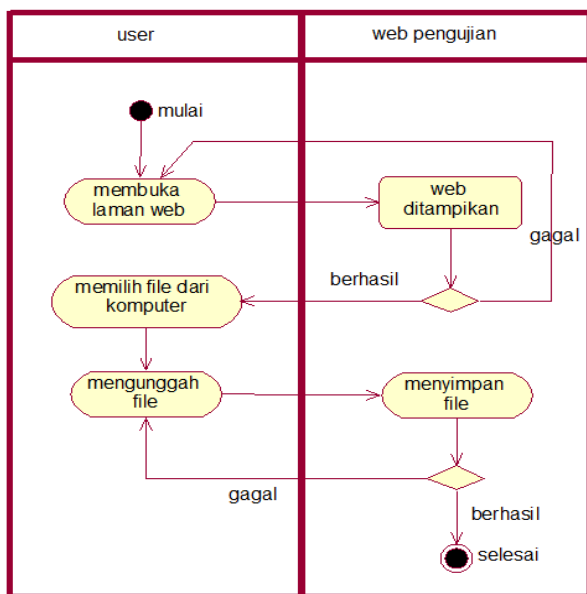
Gambar 4.5 Activity Diagram Menghapus Biodata

Adapun bentuk rancangan *activity* diagram mengunduh *file* ditunjukkan oleh gambar 4.6 berikut :



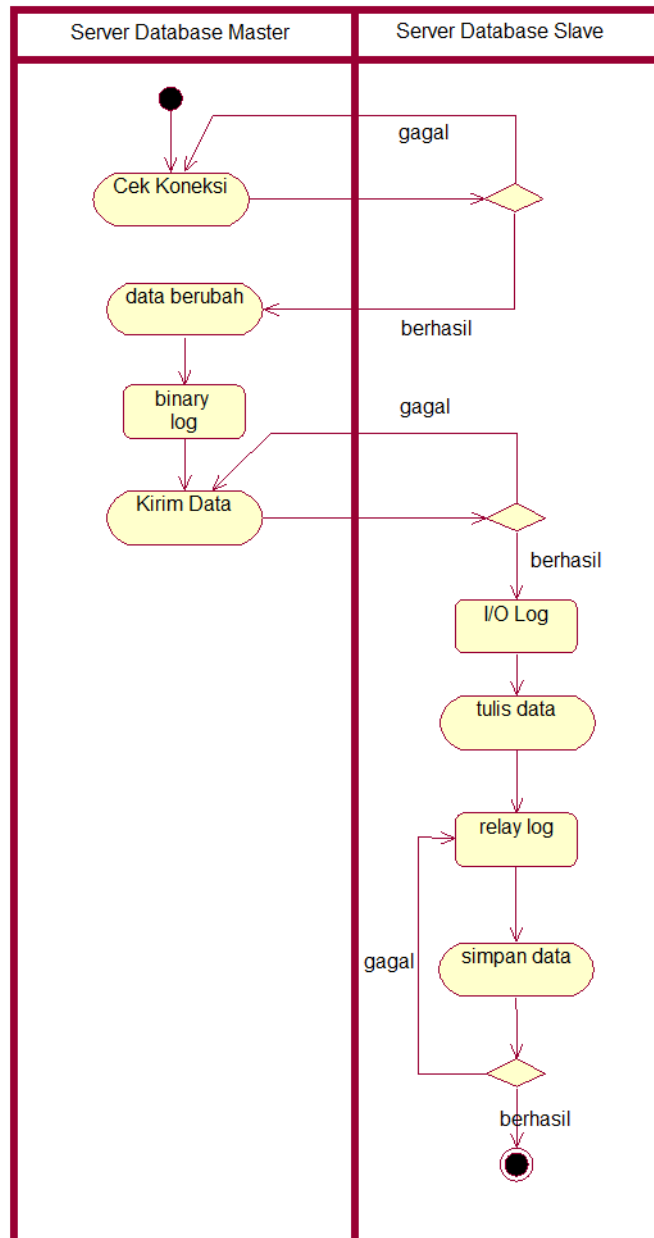
Gambar 4.6 Activity Diagram Mengunduh File

Adapun bentuk rancangan *activity* diagram mengunggah *file* ditunjukkan oleh gambar 4.6 berikut :



Gambar 4.7 Activity Diagram Mengunggah File

Dan bentuk rancangan *activity* diagram replikasi *database* ditunjukkan oleh gambar 4.8 berikut :



Gambar 4.8 Activity Diagram Replikasi Database

4.1.2.3 Instalasi *Server*

a. Sistem Operasi *Server*

Sistem *disaster recovery server database* MySQL yang dikembangkan memerlukan sistem operasi sebagai penghubung antara perangkat keras dan pengguna selain itu juga sebagai tempat dasar program pendukung di instalasi . pada penelitian ini peneliti mengembangkan penelitiannya menggunakan sistem operasi berbasis Linux *server* yaitu menggunakan sistem operasi Centos *versi* 6.5 x86_64, untuk langkah lebih jelas instalasi sistem operasi dapat dilihat pada lampiran 4.

b. Program

Selain sistem operasi *disaster recovery server database* MySQL yang dikembangkan memerlukan beberapa perangkat lunak untuk bisa berjalan dengan baik. program yang digunakan untuk mendukung fungsi sistem *disaster recovery server database* MySQL yang dikembangkan. ada beberapa perangkat lunak yang digunakan yaitu:

1. Ssh *server*
2. MySQL *server*
3. Phpmyadmin
4. Php
5. Apache

Untuk proses lebih jelas tentang pemasangan program tersebut dapat dilihat pada lampiran 5.

4.1.2.4 Pengaturan *Iptables*

Iptables merupakan suatu *tools* dalam sistem operasi Linux yang berfungsi sebagai alat untuk melakukan penyaringan terhadap *traffic* lalu lintas data, setelah kita menginstal sistem operasi pengaturan awal *iptables* adalah tertutup semua belum ada satu pun *Port* yang dibuka untuk melakukan komunikasi data. Maka dari itu kita harus mengkonfigurasi agar *Port - Port* pada *server* dapat terbuka dan digunakan untuk berkomunikasi

Pada penelitian ini menggunakan *Port* 22, 80, 3306 dan 50631 untuk melakukan replikasi *database* dan berkomunikasi antar *server*. Berikut adalah cara mengkonfigurasi *iptables* pada *server* :

1. Masuk pada terminal consol Linux,
2. Lakukan *Login* dengan memasukan *username* dan *password* root
3. Masukan kode perintah di bawah pada *command line* yang ada seperti pada

Gambar 4.9 dibawah ini :

```
iptables -I INPUT 1 -p tcp --dport 22 -j ACCEPT
iptables -I INPUT 1 -p tcp --dport 80 -j ACCEPT
iptables -I INPUT 1 -p tcp --dport 3306 -j ACCEPT
iptables -I INPUT 1 -p tcp --dport 50631 -j ACCEPT
service iptables save
service iptables restart
```

Gambar 4.9 Kode Perintah Konfigurasi *Iptables*

4. Untuk memastikan *iptables* kita sudah berjalan apa tidak, ketikkan perintah “*iptables -L -n*” pada *command line*, maka akan terlihat *iptables* yang sudah dimasukan seperti pada gambar 4.10 *iptables* untuk *server database master* dan gambar 4.11 *iptables* untuk *server database slave* dibawah.

```

Bitwise xterm - root@192.168.16.240:22 - root@localhost:~
[root@localhost ~]# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
ACCEPT    tcp  --  0.0.0.0/0             0.0.0.0/0           tcp dpt:50631
ACCEPT    tcp  --  0.0.0.0/0             0.0.0.0/0           tcp dpt:3306
ACCEPT    tcp  --  0.0.0.0/0             0.0.0.0/0           tcp dpt:80
ACCEPT    all  --  0.0.0.0/0             0.0.0.0/0           state RELATED,ESTABLISHED
ACCEPT    icmp --  0.0.0.0/0             0.0.0.0/0
ACCEPT    all  --  0.0.0.0/0             0.0.0.0/0
ACCEPT    tcp  --  0.0.0.0/0             0.0.0.0/0           state NEW tcp dpt:22
REJECT    all  --  0.0.0.0/0             0.0.0.0/0           reject-with icmp-host-prohi
bited
ACCEPT    tcp  --  0.0.0.0/0             0.0.0.0/0           tcp dpt:3306

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination
REJECT    all  --  0.0.0.0/0             0.0.0.0/0           reject-with icmp-host-prohi
bited

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
[root@localhost ~]#

```

Gambar 4.10 *Iptables pada Server Database Master*

```

Bitwise xterm - root@192.168.16.250:22 - root@localhost:~
[root@localhost ~]# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
ACCEPT    tcp  --  0.0.0.0/0             0.0.0.0/0           tcp dpt:50631
ACCEPT    tcp  --  0.0.0.0/0             0.0.0.0/0           tcp dpt:3306
ACCEPT    tcp  --  0.0.0.0/0             0.0.0.0/0           tcp dpt:80
ACCEPT    all  --  0.0.0.0/0             0.0.0.0/0           state RELATED,ESTABLISHED
ACCEPT    icmp --  0.0.0.0/0             0.0.0.0/0
ACCEPT    all  --  0.0.0.0/0             0.0.0.0/0
ACCEPT    tcp  --  0.0.0.0/0             0.0.0.0/0           state NEW tcp dpt:22
REJECT    all  --  0.0.0.0/0             0.0.0.0/0           reject-with icmp-host-prohibited

ACCEPT    tcp  --  0.0.0.0/0             0.0.0.0/0           tcp dpt:3306

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination
REJECT    all  --  0.0.0.0/0             0.0.0.0/0           reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
[root@localhost ~]#

```

Gambar 4.11 *Iptables pada Server Database Slave*

Penjelasan dari maksud kode program ini adalah

1. *Iptables -I INPUT 1 -p tcp --dPort 22 -j ACCEPT* : Pada kode ini dimaksudkan untuk membuka akses masuk *client* ke-Port 22 (SSH) agar dapat di akses dari jaringan luar yang masih terhubung dengan *server*, agar peneliti dapat melakukan *remote* dan mengkonfigurasi *server* dari jaringan luar.

2. *Iptables -I INPUT 1 -p tcp --dPort 80 -j ACCEPT* : pada kode ini dimaksudkan untuk membuka akses masuk *client* ke-Port 80 (*webserver*) agar dapat diakses dari jaringan luar yang masih terhubung dengan *server*, agar fitur *website* dapat dibuka oleh *client*.
3. *Iptables -I INPUT 1 -p tcp --dPort 3306 -j ACCEPT* : pada kode ini dimaksudkan untuk membuka akses masuk ke-Port 3306 *server database slave* atau *master* agar dapat berkomunikasi satu sama lain dan dapat melakukan replikasi *database*.
4. *Iptables -I INPUT 1 -p tcp --dPort 50631 -j ACCEPT* : pada kode ini dimaksudkan untuk membuka akses masuk ke-Port 50631 yang digunakan untuk mengirimkan pesan, *Log*, dan berhasil tidaknya replikasi.

4.1.2.5 Pengaturan MySQL

Pada penelitian kali ini kita menggunakan MySQL replikasi pada CentOS *server* sebagai alat untuk mereplikasi *database* antara *server database master* dan *server database slave*. Dengan replikasi MySQL kedua *database* yang ada pada *server* akan selalu sama dan melakukan sinkronisasi data. Agar MySQL tersebut dapat melakukan replikasi maka MySQL replikasi harus dikonfigurasi terlebih dahulu.

Pertama kita harus mengkonfigurasi *file my.conf* di masing-masing *server* dengan mengetikkan “*nano /etc/my.cnf*” pastikan konfigurasi nya seperti gambar 4.12 untuk *server database master* dan gambar 4.13 untuk *server database slave*.

```

[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
symbolic-links=0
sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES
port=3306
old_passwords=0
query_cache_size=256M
query_cache_type=1
query_cache_limit=8M
server-id=1
log_bin=/var/lib/mysql/mysql-bin.log
log_error=mysql-bin.err
long_query_time=1
slow_query_log=1
slow_query_log_file=/var/lib/mysql/mysql-slow.log
general_log=1
general_log_file=/var/lib/mysql/mysql.log
log_bin_index=/var/lib/mysql/mysql-bin.log.index
relay_log=/var/lib/mysql/mysql-relay-bin
relay_log_index=/var/lib/mysql/mysql-relay-bin.index
expire_logs_days=10
max_binlog_size=100M
log_slave_updates=1
bind-address=192.168.16.250
local-infile=0
max_allowed_packet=1024M
wait_timeout=6000
innodb_log_file_size=1024M
[mysqld_safe]
log=1

```

Gambar 4.12 Konfigurasi my.conf Server Database Master

Penjelasan dari maksud kode konfigurasi diatas adalah :

1. `datadir=/var/lib/mysql` : pada konfigurasi ini berartikan `datadir` (data *directory*) MySQL tersimpan pada *directory folder* `/var/lib/mysql` pada *server*.
2. `socket=/var/lib/mysql/mysql.sock` : pada konfigurasi ini berartikan *socket* konfigurasi MySQL tersimpan pada *directory folder* `/var/lib/mysql/mysql.sock` pada *server*.

3. `symbolic-links=0` : pada konfigurasi ini berartikan untuk menonaktifkan simbolik-link hal ini dianjurkan untuk mencegah risiko keamanan berbagai macam ancaman.
4. `sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES` : pada konfigurasi ini merupakan konfigurasi *default* dari pengaturan MySQL. Yang tidak memiliki pilihan engine yang spesifik.
5. `Port=3306` : konfigurasi ini berartikan kita menggunakan *Port* 3306 yang digunakan untuk melakukan komunikasi antar *server database* untuk melakukan replikasi.
6. `old_passwords=0` : pada konfigurasi ini dapat diartikan *old password* tidak diaktifkan karena bernilai 0 (*disable*) jika bernilai 1 (*enable*)
7. `query_cache_size=256M` : pada konfigurasi ini berartikan mengalokasikan ukuran *size* pada *query cache* sebesar 256 Mb. Untuk memampung *cache* replikasi.
8. `query_cache_type=1` : pada konfigurasi ini berartikan kita mengaktifkan *cache query* untuk menampung *query* sementara *database*, jika bernilai 0 (*disable*) jika bernilai 1 (*enable*) fitur ini.
9. `query_cache_limit=8M` : pada konfigurasi ini dapat diartikan ukuran *query cache* bernilai *limit* 8 Mb.
10. `server-id=1` : pada konfigurasi ini kita menginisialisasikan dengan angka *server* di dari *server* master, *server* id dari *server* master dan *Save* haruslah berbeda, *server* id ini nantinya akan berfungsi dalam pemanggilan data saat replikasi berlangsung.

11. *Log_bin=/var/lib/mysql/mysql-bin.Log* : pada konfigurasi ini dapat diartikan *Log binary* dari replikasi *database* disimpan pada *directory* */var/lib/mysql/mysql-bin.Log* pada *server*. *Log binary* menampung *history Log* dari setiap perubahan *database* baik *insert*, *update* dan *delete*.
12. *Log_error=mysql-bin.err* : pada konfigurasi ini dapat diartikan *Log error* dari replikasi *database* disimpan pada *file* *mysql-bin.err*. *Log error* berfungsi untuk menampung *history error* dari setiap perubahan *database* baik *insert*, *update* dan *delete*.
13. *long_query_time=1*: pada konfigurasi ini berartikan kita mengaktifkan *Log query time* untuk menampung *Log history* waktu perubahan *query database*, jika bernilai 0 (*disable*) jika bernilai 1 (*enable*) fitur ini.
14. *slow_query_Log=1* : pada konfigurasi ini berartikan kita mengaktifkan *slow query Log* untuk menampung *Log history query* sementara *database* secara lambat dan ditail, jika bernilai 0 (*disable*) jika bernilai 1 (*enable*) fitur ini.
15. *slow_query_Log_file=/var/lib/mysql/mysql-slow.Log* : pada konfigurasi ini dapat diartikan *slow query Log* dari *database* disimpan pada *directory* */var/lib/mysql/mysql-slow.Log* pada *server*. *Slow query Log* menampung *history Log* secara rinci dari setiap perubahan *database* baik *insert*, *update* dan *delete*.
16. *general_Log=1*: pada konfigurasi ini berartikan kita mengaktifkan *general Log* sebagai catatan umum tentang apa *mysqld* lakukan. *Server* menulis informasi ke *Log* ini ketika klien tersambung atau tidak, dan *Log* setiap pernyataan SQL yang diterima dari klien. *Log query* umum bisa sangat berguna ketika Anda mencurigai adanya kesalahan dalam klien dan ingin

tahu persis apa yang klien dikirim ke mysqld. Jika bernilai 0 (*disable*) jika bernilai 1 (*enable*) fitur ini.

17. *general _Log_file=/var/lib/mysqld.Log* : pada konfigurasi ini dapat diartikan *general Log* dari *database* disimpan pada *directory /var/lib/mysqld.Log* pada *server*.
18. *Log_bin_index=/var/lib/mysql/mysql-bin.Log.index* : pada konfigurasi ini dapat diartikan *Log binary index* dari *database* disimpan pada *directory /var/lib/mysql/mysql-bin.Log.index* pada *server*.
19. *relay_Log=/var/lib/mysql/mysql-relay-bin* : pada konfigurasi ini dapat diartikan *relay Log* dari *database* disimpan pada *directory /var/lib/mysql/mysql-relay-bin* pada *server*.
20. *relay_Log_index=/var/lib/mysql/mysql-relay-bin.index* : pada konfigurasi ini dapat diartikan *relay Log* dari *database* disimpan pada *directory /var/lib/mysql/mysql-relay-bin.index* pada *server*.
21. *expire_Logs_days=10* : pada konfigurasi ini dapat diartikan waktu maksimal penyimpanan dari *Log* histori *database* adalah selama 10 hari, ketika lebih dari 10 hari maka *Log* tersebut akan hilang dan terhapus.
22. *max_binLog_size=100M* : pada konfigurasi ini dapat diartikan bawa di alokasikan *binery Log* sebesar 100 Mb , untuk menampung *Log binary*.
23. *Log_slave_updates=1* : pada konfigurasi ini berartikan kita mengaktifkan *Log* ketika master melakukan replikasi update menuju *server slave* hal ini akan menyimpan seluruh *Log* dari replikasi databasa anantara slave dan master, jika bernilai 0 (*disable*) jika bernilai 1 (*enable*) fitur ini.

24. `bind-address=192.168.16.250` : pada konfigurasi ini berartikan alamat ini yang diikat atau dituju untuk melakukan replikasi (IP *server slave*) secara *default* IP ini bisa kita isikan 0.0.0.0, tetapi dari segi keamanan tidak dianjurkan karena dapat menyebabkan paket replikasi terdengar dan di sadap oleh host lain.
25. `local-infile=0` : pada konfigurasi ini berartikan menonaktifkan local *infile* demikeamanan maka sebaiknya local *infile* di nonaktifkan di karenakan adanya isu yang dapat melihat data yang ada didalam MySQL dan bisa diambil, jika bernilai 0 (*disable*) jika bernilai 1 (*enable*) fitur ini.
26. `max_allowed_packet=1024M` : pada konfigurasi dapat diartikan maksimum paket yang diijinkan adalah sebesar 1024 Mb.
27. `wait_timeout=6000` : pada konfigurasi dapat diartikan waktu tunggu dari paket yang akan masuk adalah sebesar 6000 detik
28. `innodb_Log_file_size=1024M` : : pada konfigurasi dapat diartikan *Log* dari *file size* buffer innodb adalah sebesar 1024 Mb
29. `[mysqld_safe]` : pada konfigurasi dapat diartikan tag program safe MySQL daemon
30. `Log=1` : pada konfigurasi ini berartikan mengaktifkan *Log* yang ada didalam MySQL, jika bernilai 0 (*disable*) jika bernilai 1 (*enable*) fitur ini.

```

[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
symbolic-links=0
sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES
port=3306
old_passwords=0
query_cache_size = 256M
query_cache_type=1
query_cache_limit = 8M
server_id = 2
log_bin=mysql-bin
log_error=mysql-bin.err
long_query_time = 1
slow_query_log = 1
slow_query_log_file = /var/lib/mysql/mysql-slow.log
general_log = 1
general_log_file = /var/lib/mysql/mysql.log
log_bin_index = /var/lib/mysql/mysql-bin.log.index
relay_log = /var/lib/mysql/mysql-relay-bin
relay_log_index = /var/lib/mysql/mysql-relay-bin.index
expire_logs_days = 10
max_binlog_size = 1000M
log_slave_updates = 1
bind-address = 192.168.16.240
local-infile = 0
max_allowed_packet=1024M
wait_timeout=6000
innodb_log_file_size = 1024M
log-error=/var/log/mysql.log
pid-file=/var/run/mysql/mysql.pid
master-host = 192.168.16.240
master-user = root
master-password = 123456
master-connect-retry = 60
[mysqld_safe]
log = 1

```

Gambar 4.13 Konfigurasi my.conf Server Database Slave

Penjelasan dari maksud kode konfigurasi diatas adalah :

1. *Datadir=/var/lib/mysql* : pada konfigurasi ini berartikan *datadir* (data *directory*) mysql tersimpan pada *directory folder /var/lib/mysql* pada *server*.
2. *Socket=/var/lib/mysql/mysql.sock* : pada konfigurasi ini berartikan *socket* konfigurasi mysql tersimpan pada *directory folder /var/lib/mysql/mysql.sock* pada *server*.

3. `Symbolic-links=0` : pada konfigurasi ini berartikan untuk menonaktifkan simbolik-link hal ini dianjurkan untuk mencegah risiko keamanan berbagai macam ancaman.
4. `Sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES` : pada konfigurasi ini merupakan konfigurasi *default* dari pengaturan mysql. Yang tidak memiliki pilihan engine yang spesifik.
5. `Port=3306` : konfigurasi ini berartikan kita menggunakan *Port* 3306 yang digunakan untuk melakukan komunikasi antar *server database* untuk melakukan replikasi.
6. `Old_passwords=0` : pada konfigurasi ini dapat diartikan *old password* tidak diaktifkan karena bernilai 0 (*disable*) jika bernilai 1 (*enable*)
7. `Query_cache_size=256M` : pada konfigurasi ini berartikan mengalokasikan ukuran *size* pada *query cache* sebesar 256 Mb. Untuk memampungkan *cache* replikasi.
8. `Query_cache_type=1` : pada konfigurasi ini berartikan kita mengaktifkan *cache query* untuk menampung *query* sementara *database*, jika bernilai 0 (*disable*) jika bernilai 1 (*enable*) fitur ini.
9. `Query_cache_limit=8M` : pada konfigurasi ini dapat diartikan ukuran *query cache* bernilai *limit* 8 Mb.
10. `Server-id=1` : pada konfigurasi ini kita menginisialisasikan dengan angka *server* di dari *server* master, *server* id dari *server* master dan *Save* haruslah berbeda, *server* id ini nantinya akan berfungsi dalam pemanggilan data saat replikasi berlangsung.

11. *Log_bin=/var/lib/mysql/mysql-bin.Log* : pada konfigurasi ini dapat diartikan *Log binary* dari replikasi *database* disimpan pada *directory /var/lib/mysql/mysql-bin.Log* pada *server*. *Log binary* menampung *history Log* dari setiap perubahan *database* baik insert, update dan delete.
12. *Log_error=mysql-bin.err* : pada konfigurasi ini dapat diartikan *Log error* dari replikasi *database* disimpan pada *file mysql-bin.err*. *Log error* berfungsi untuk menampung *history error* dari setiap perubahan *database* baik insert, update dan delete.
13. *Long_query_time=1*: pada konfigurasi ini berartikan kita mengaktifkan *Log query time* untuk menampung *Log history* waktu perubahan *query database*, jika bernilai 0 (*disable*) jika bernilai 1 (*enable*) fitur ini.
14. *Slow_query_Log=1* : pada konfigurasi ini berartikan kita mengaktifkan *slow query Log* untuk menampung *Log history query* sementara *database* secara lambat dan ditail, jika bernilai 0 (*disable*) jika bernilai 1 (*enable*) fitur ini.
15. *Slow_query_Log_file=/var/lib/mysql/mysql-slow.Log* : pada konfigurasi ini dapat diartikan *slow query Log* dari *database* disimpan pada *directory /var/lib/mysql/mysql-slow.Log* pada *server*. *Slow query Log* menampung *history Log* secara rinci dari setiap perubahan *database* baik insert, update dan delete.
16. *General _Log=1*: pada konfigurasi ini berartikan kita mengaktifkan *general Log* sebagai catatan umum tentang apa *mysqld* lakukan. *Server* menulis informasi ke *Log* ini ketika klien tersambung atau tidak, dan *Log* setiap pernyataan SQL yang diterima dari klien. *Log query* umum bisa sangat berguna ketika Anda mencurigai adanya kesalahan dalam klien dan ingin

tahu persis apa yang klien dikirim ke mysqld. Jika bernilai 0 (*disable*) jika bernilai 1 (*enable*) fitur ini.

17. *General _Log_file=/var/lib/mysqld.Log* : pada konfigurasi ini dapat diartikan *general Log* dari *database* disimpan pada *directory /var/lib/mysqld.Log* pada *server*.
18. *Log_bin_index=/var/lib/mysql/mysql-bin.Log.index* : pada konfigurasi ini dapat diartikan *Log binary index* dari *database* disimpan pada *directory /var/lib/mysql/mysql-bin.Log.index* pada *server*.
19. *Relay_Log=/var/lib/mysql/mysql-relay-bin* : pada konfigurasi ini dapat diartikan *relay Log* dari *database* disimpan pada *directory /var/lib/mysql/mysql-relay-bin* pada *server*.
20. *Relay_Log_index=/var/lib/mysql/mysql-relay-bin.index* : pada konfigurasi ini dapat diartikan *relay Log* dari *database* disimpan pada *directory /var/lib/mysql/mysql-relay-bin.index* pada *server*.
21. *Expire_Logs_days=10* : pada konfigurasi ini dapat diartikan waktu maksimal penyimpanan dari *Log* histori *database* adalah selama 10 hari, ketika lebih dari 10 hari maka *Log* tersebut akan hilang dan terhapus.
22. *Max_binLog_size=100M* : pada konfigurasi ini dapat diartikan bawa di alokasikan *binery Log* sebesar 100 Mb , untuk menampung *Log binary*.
23. *Log_slave_updates=1* : pada konfigurasi ini berartikan kita mengaktifkan *Log* ketika master melakukan replikasi update menuju *server slave* hal ini akan menyimpan seluruh *Log* dari replikasi *database* antara *slave* dan *master*, jika bernilai 0 (*disable*) jika bernilai 1 (*enable*) fitur ini.

24. `Bind-address=192.168.16.240` : pada konfigurasi ini berartikan alamat ini yang di ikat atau di tuju untuk melakukan replikasi (IP *server slave*) secara *default* IP ini bisa kita isikan 0.0.0.0, tetapi dari segi keamanan tidak di anjurkan karena dapat menyebabkan paket replikasi terdengar dan di sadap oleh host lain.
25. `Local-infile=0` : pada konfigurasi ini berartikan menonaktifkan local *infile* demikeamanan maka sebaiknya local *infile* di nonaktifkan di karenakan ada nya isu yang dapat melihat data yang ada didalam mysql dan bisa diambil, jika bernilai 0 (*disable*) jika bernilai 1 (*enable*) fitur ini.
26. `Max_allowed_packet=1024M` : pada konfigurasi dapat diartikan maksimum paket yang diijinkan adalah sebesar 1024 Mb.
27. `Wait_timeout=6000` : pada konfigurasi dapat diartikan waktu tunggu dari paket yang akan masuk adalah sebesar 6000 detik
28. `Innodb_Log_file_size=1024M` : : pada konfigurasi dapat diartikan *Log* dari *file size* buffer innodb adalah sebesar 1024 Mb
29. `Log-error=/var/Log/mysqlD.Log` : pada konfigurasi dapat diartikan *Log* error dari *server* tersimpan pada lokasi *file* `/var/Log/mysqlD.Log`
30. `Pid-file=/var/run/mysqlD/mysqlD.pid` : pada konfigurasi dapat diartikan *Pid file* tersimpan pada *folder* `var/run/mysqlD/mysqlD.pid`
31. `Master-host = 192.168.16.240` : pada konfigurasi dapat diartikan master host dari *server* beralamatkan IP 192.168.16.240
32. `Master-user = root` : pada konfigurasi ini dapat diartikan user dari *server* master adalah root

33. *Master-password* = 123456 : pada konfigurasi ini dapat diartikan *password* dari *server* master adalah 123456
34. *Master-connect-retry* = 60 : pada konfigurasi ini dapat diartikan waktu untuk mencoba konek kembali ke *server* utama adalah sebesar 60 detik
35. [*mysqld_safe*] : pada konfigurasi dapat diartikan tag program safe mysql daemon
36. *Log*=1 : pada konfigurasi ini berartikan mengaktifkan *Log* yang ada didalam mysql, jika bernilai 0 (*disable*) jika bernilai 1 (*enable*) fitur ini.

Untuk mengetahui replikasi kita sudah berjalan atau belum kita dapat melihatnya dengan cara mengetikkan perintah berikut pada *command line* di *server* “show master status;” untuk mengetahui status *server master*. Maka akan terlihat seperti gambar 4.14 dan untuk mengetahui replikasi di *slave database* kita dapat mengetikkan perintah berikut pada *command line* di *server* “show slave status \G” maka terlihat seperti gambar 4.15. pastikan *slave_IO_running* dan *slave_SQL_running* dalam keadaan yes. Dikarenakan jika no replikasi belum berjalan. Untuk lebih lengkapnya proses pemasangan dan pengaturan MySQL dapat dilihat pada lampiran 6.

```

+-----+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin.000032 | 1191490 |              |                   |                   |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

Gambar 4.14 Master Status Replikasi Server Database Master

Penjelasan dari maksud kode konfigurasi diatas adalah :

1. *File* yang menyimpan replikasi binary terdapat pada *file* mysql-bin.00032

2. Posisi *Log* dari perubahan data ada pada posisi 1191490 *Log* pada file

mysql-bin.00032

```

***** 1. row *****
Slave_IO_State           : Waiting for master to send event
Master_Host              : 192.168.16.240
Master_User              : replication
Master_Port              : 3306
Connect_Retry           : 60
Master_Log_File          : mysql-bin.000032
Read_Master_Log_Pos     : 1191490
Relay_Log_File           : mysql-relay-bin.000057
Relay_Log_Pos           : 1191653
Relay_Master_Log_File   : mysql-bin.000032
Slave_IO_Running        : Yes
Slave_SQL_Running       : Yes
Replicate_Do_DB         :
Replicate_Ignore_DB     :
Replicate_Do_Table      :
Replicate_Ignore_Table  :
Replicate_Wild_Do_Table :
Replicate_Wild_Ignore_Table :
Last_Errno              : 0
Last_Error              :
Skip_Counter            : 0
Exec_Master_Log_Pos    : 1191490
Relay_Log_Space         : 1191989
Until_Condition         : None
Until_Log_File         :
Until_Log_Pos          : 0
Master_SSL_Allowed     : No
Master_SSL_CA_File     :
Master_SSL_CA_Path     :
Master_SSL_Cert        :
Master_SSL_Cipher      :
Master_SSL_Key         :
Seconds_Behind_Master   : 0
Master_SSL_Verify_Server_Cert : No
Last_IO_Errno          : 0
Last_IO_Error          :
Last_SQL_Errno         : 0
Last_SQL_Error         :
Replicate_Ignore_Server_Ids :
Master_Server_Id       : 1
Master_UUID            : 16d5cf4c-848d-11e5-adde-000c2925e53f
Master_Info_File       : /var/lib/mysql/master.info
SQL_Delay              : 0
SQL_Remaining_Delay    : NULL
Slave_SQL_Running_State : Slave has read all relay log; waiting for the slave I/O
                        thread to update it
Master_Retry_Count     : 86400
Master_Bind            :
Last_IO_Error_Timestamp :
Last_SQL_Error_Timestamp :
Master_SSL_Crl         :
Master_SSL_Crlpath     :
Retrieved_Gtid_Set     :
Executed_Gtid_Set      :
Auto_Position          : 0
1 row in set (0,00 sec)

```

Gambar 4.15 *Slave Status Replikasi Server Database Slave*

Penjelasan dari maksud kode konfigurasi diatas adalah :

1. *Slave_IO_State* : *Waiting for master to send event* : pada konfigurasi ini dapat diartikan bawa *Slave IO* dari *server* sedang menunggu data yang di kirimkan dari *server* master
2. *Master_Host* : 192.168.16.240 : pada konfigurasi ini dapat diartikan master host beralamat kan IP 192.168.16.240
3. *Master_User* : *replication* : pada konfigurasi ini dapat diartikan master user adalah *replication* master user yang di isikan harus sesuai dengan di *server* master.
4. *Master_Port* : 3306 : pada konfigurasi ini dapat diartikan master *server* menggunakan *Port* 3306 untuk melakukan relikasi data.
5. *Connect_Retry* : 60 : pada konfigurasi ini dapat diartikan *server* akan mengulang 60 detik waktu apabila terjadi kegagalan koneksi (*disconnect*)
6. *Master_Log_File* : *mysql-bin.000032* : pada konfigurasi ini dapat diartikan *Log File* dari master *server* bernama kan *mysql-bin.000032*, nama ini harus sama di isikan di kedua *server* antara *server* master dan slave.
7. *Read_Master_Log_Pos* : 1191490 : pada konfigurasi ini dapat diartikan read master *Log* ada di urutan 1191490 urutan ini akan berubah jika *database* di lakukan perubahan *Log* post disini jumlahnya harus sama antara *server* master dan slave.
8. *Relay_Log_File* : *mysql-relay-bin.000057* : pada konfigurasi ini dapat diartikan reley *Log file* dari replikasi *database* ini tersimpan pada *file* *mysql-relay-bin.000057*

9. *Relay_Log_Pos* : 1191653 : pada konfigurasi ini dapat diartikan reley *Log* post ada di urutan 1191653 urutan ini akan berubah jika *database* di lakukan perubahan reley *Log* post disini jumlahnya harus sama antara *server* master dan slave.
10. *Relay_Master_Log_File* : mysql-bin.000032 : pada konfigurasi ini dapat diartikan reley master *Log file* dari replikasi *database* ini tersimpan pada *file* mysql-relay-bin.00032
11. *Slave_IO_Running* : Yes : pada konfigurasi ini dapat diartikan status untuk IO slave dalam posisi running dan siap untuk menerima replikasi dari *server* master.
12. *Slave_SQL_Running* : Yes : pada konfigurasi ini dapat diartikan status untuk SQL Slave dalam posisi running dan siap untuk menerima replikasi dari *server* master.
13. *Last_Errno* : 0 : pada konfigurasi ini dapat diartikan last error dari replikasi *database* ini adalah 0 jadi belum terdapat kesalahan error pada replikasi ini
14. *Skip_Counter* : 0 : pada konfigurasi ini dapat diartikan skip counter adalah 0 jadi belum ada counter data yang terlewat di replikasi *database* ini
15. *Exec_Master_Log_Pos* : 1191490 : pada konfigurasi ini dapat diartikan Exec master *Log* Posisi berada pada urutan 1191490.
16. *Relay_Log_Space* : 1191989 : pada konfigurasi ini dapat diartikan reley *Log* space yang dapat di gunakan adalah sebanyak 1191989.
17. *Until_Condition* : None : pada konfigurasi ini dapat diartikan tidak ada kondisi khusus yang di gunakan untuk replikasi ini

18. *Until_Log_Pos* : 0 : pada konfigurasi ini dapat diartikan *Log* posisi dari until adalah 0
19. *Master_SSL_Allowed* : No : pada konfigurasi ini dapat diartikan master sever tidak menggunakan kode SSL dalam melakukan replikasi jadi dapat di simpulkan kalau *file* yang di transfer tidak terenkripsi
20. *Seconds_Behind_Master* : 0 : pada konfigurasi ini dapat diartikan second behind master *server* berada pada urutan ke 0 atau belum ada yang mengisi.
21. *Master_SSL_Verify_Server_Cert* : No : pada konfigurasi ini dapat diartikan kita tidak menggunakan sertifikat verifikasi untuk kode SSL karena *file* pengirim tidak dienkripsi di awal nya.
22. *Last_IO_Errno* : 0 : pada konfigurasi ini dapat diartikan last error IO yang terjadi selama replikasi berlangsung adalah 0, atau tidak pernah terjadi error pada IO.
23. *Last_SQL_Errno* : 0 : pada konfigurasi ini dapat diartikan last error SQL yang terjadi selama replikasi berlangsung adalah 0, atau tidak pernah terjadi error pada SQL.
24. *Master_Server_Id* : 1 : pada konfigurasi ini dapat diartikan master *server* replikasi ber-id kan 1, id *server* harus lah sama karena id merupakan normor untuk mengkaitkan kedua nya.
25. *Master_UUID* : 16d5cf4c-848d-11e5-adde-000c2925e53f : pada konfigurasi ini dapat diartikan master *server* memiliki UUID dengan no 16d5cf4c-848d-11e5-adde-000c2925e53f, UUID adalah singkatan dari Universally Unique Identifier yang dijadikan sebagai standar identifier sama Open Software Foundation.

26. *Master_Info_File* : /var/lib/mysql/master.info : pada konfigurasi ini dapat diartikan master info *file* terdapat pada *file* /var/lib/mysql/master.
27. *SQL_Delay* : 0 : pada konfigurasi ini dapat diartikan SQL tidak memiliki delay.
28. *SQL_Remaining_Delay* : NULL : pada konfigurasi ini dapat diartikan SQL dari replikasi ini tidak memiliki delay (hambatan) dan sisa record yang belum terreplikasi
29. *Slave_SQL_Running_State* : Slave has read all relay *Log*; waiting for the slave I/O thread to update it : pada konfigurasi ini dapat diartikan slave SQL sedang berjalan dan menunggu thread update dari *server* master.
30. *Master_Retry_Count* : 86400 : pada konfigurasi ini dapat diartikan master retry count sebanyak 86400 kali dari replikasi tersebut di aktifkan
31. *Auto_Position* : 0 : pada konfigurasi ini dapat diartikan posisi *default* adalah 0

4.1.3 Uji Fungsional

Selanjutnya pengujian yang dilakukan oleh peneliti adalah pengujian fungsional dengan melihat apakah fitur yang terdapat pada sistem dapat berjalan sesuai dengan yang diharapkan. Data dari pengujian fungsional penelitian ini dapat dilihat pada table 4.1 seperti yang ada dibawah.

Tabel 4.1 Hasil Uji Fungsional

No.	Fitur	Berfungsi
1.	<i>Server database salve</i> dapat mereplikasi <i>database, table, dan record</i> yang baru ditambahkan pada <i>server database master</i>	√
2.	<i>Server database salve</i> dapat mereplikasi <i>database, table, dan record</i> yang baru diubah pada <i>server database master</i>	√
3.	<i>Server database salve</i> dapat mereplikasi <i>database, table, dan record</i> yang baru dihapus pada <i>server database master</i>	√
4.	<i>Website</i> pengujian dapat menambahkan , mengedit dan menghapus biodata yang terkoneksi dengan <i>database master</i>	√
5.	<i>Website</i> pengujian dapat mengunduh dan mengunggah <i>file</i> yang tersimpan di <i>database master</i>	√
6.	<i>Website</i> pengujian dapat melakukan <i>stress test database</i>	√
7.	<i>Website</i> pengujian dapat berfungsi sebagaimana mestinya saat <i>server database master</i> dimatikan	√
8.	<i>Server database slave</i> berperan menggantikan fungsi <i>server database master</i> saat <i>server database master</i> mati	√

Berdasarkan data pada tabel 4.1 maka dapat disimpulkan seluruh fitur berfungsi dengan baik.

4.1.4 Uji Replikasi

Pada tahap ini dilakukan pengujian sistem berupa uji coba replikasi yaitu dengan pengujian dalam kelompok kecil dan kelompok besar, dimana kita menguji semua unit – unit proses terkait, tentang replikasi *database* pada sub bab selanjutnya akan dipaparkan hasil pengambilan data yang sudah peneliti lakukan secara lengkap.

4.1.4.1 Uji Replikasi Kelompok Kecil

a. Pengisian Biodata

Berdasarkan hasil pengujian pengisian biodata kelompok kecil yang di lakukan oleh 5 orang *user* maka didapatkan data yang dapat dilihat pada Table 4.2 dibawah ini.

Tabel 4.2 Hasil Uji pengisian biodata kelompok kecil

Jenis Kegiatan	Jumlah Data Input*	Server Database Master		Server Database Slave		Kesamaan Jumlah Data		Jumlah perbedaan jika tidak sama
		Jumlah data sebelum input	Jumlah data setelah input	Jumlah data sebelum input	Jumlah data setelah input	Sama	Tidak	
Input 5 Biodata secara Bersamaan	25	0	25	0	25	✓		0
Input 10 Biodata secara Bersamaan	50	25	75	25	75	✓		0
Input 15 Biodata secara Bersamaan	75	75	150	75	150	✓		0

Terlihat pada tabel 4.2 pengisian biodata secara bersamaan, data pada *server database master* dan *slave* tidak ada perbedaan dan memiliki jumlah kesamaan *record* yang sama.

b. Pengunggahan Dokumen

Berdasarkan hasil pengujian pengunggahan dokumen kelompok kecil yang di lakukan oleh 5 orang *user* maka didapatkan data yang dapat dilihat pada table 4.3 dibawah ini.

Tabel 4.3 Hasil Uji Pengunggahan Dokumen Kelompok Kecil

Jenis Kegiatan	Jumlah Data Input		total	Server Database Master		Server Database Slave		Kesamaan Jumlah Data		Jumlah perbedaan jika tidak sama
	User	Banyak input		Jumlah data sebelum input	Jumlah data setelah input	Jumlah data sebelum input	Jumlah data setelah input	Sama	Tidak	
Unggah file 1kb	User 1	3	19	5	24	5	24	✓		0
	User 2	5								
	User 3	4								
	User 4	5								
	User 5	2								
Unggah file 1mb	User 1	3	18	24	42	24	42	✓		0
	User 2	5								
	User 3	3								
	User 4	5								
	User 5	2								
Unggah file 5mb	User 1	3	17	42	59	42	59	✓		0
	User 2	5								
	User 3	2								

	<i>User 4</i>	5								
	<i>User 5</i>	2								
Unggah file 10mb	<i>User 1</i>	3								
	<i>User 2</i>	5								
	<i>User 3</i>	2	17	59	76	59	76	✓	0	
	<i>User 4</i>	5								
	<i>User 5</i>	2								
Unggah file 15mb	<i>User 1</i>	3								
	<i>User 2</i>	0								
	<i>User 3</i>	2	10	76	86	76	86	✓	0	
	<i>User 4</i>	3								
	<i>User 5</i>	2								

Terlihat pada tabel 4.3 pengunggahan dokumen secara bersamaan, data pada *server database master* dan *slave* tidak ada perbedaan dan memiliki jumlah kesamaan *record* yang sama.

c. *Stress test*

Berdasarkan hasil pengujian *stress test* kelompok kecil yang di lakukan oleh 5 orang *user* maka didapatkan data yang dapat dilihat pada Table 4.4 dibawah ini.

Tabel 4.4 Hasil Uji *Stress test* kelompok kecil

Jenis Kegiatan	Jumlah Data Input			Server Database Master		Server Database Slave		Kesamaan Jumlah Data		Jumlah perbedaan jika tidak sama
	User	Banyak input	total	Jumlah data sebelum input	Jumlah data setelah input	Jumlah data sebelum input	Jumlah data setelah input	Sama	Tidak	
Stres Test 1	User 1	10	50	3171	3221	3171	3221	✓		0
	User 2	10								
	User 3	10								
	User 4	10								
	User 5	10								
Stres Test 2	User 1	50	250	3221	3471	3221	3471	✓		0
	User 2	50								
	User 3	50								
	User 4	50								

	<i>User 5</i>	50							
Stres Test 3	<i>User 1</i>	100							
	<i>User 2</i>	100							
	<i>User 3</i>	100	500	3471	3971	3471	3971	✓	0
	<i>User 4</i>	100							
	<i>User 5</i>	100							
Stres Test 4	<i>User 1</i>	1000							
	<i>User 2</i>	1000							
	<i>User 3</i>	1000	5000	3971	8971	3971	8971	✓	0
	<i>User 4</i>	1000							
	<i>User 5</i>	1000							
Stres Test 5	<i>User 1</i>	5294							
	<i>User 2</i>	5299							
	<i>User 3</i>	4542	24233	8971	33204	8971	33204	✓	0
	<i>User 4</i>	5372							
	<i>User 5</i>	3726							

Terlihat pada tabel 4.4 pengujian *stress test* secara bersamaan, data pada *server database master* dan *slave* tidak ada perbedaan dan memiliki jumlah kesamaan *record* yang sama.

4.1.4.2 Uji Replikasi Kelompok Besar

a. Pengisian Biodata

Berdasarkan hasil pengujian pengisian biodata kelompok besar yang di lakukan oleh 20 orang *user* maka didapatkan data yang dapat dilihat pada Table 4.5 dibawah ini.

Tabel 4.5 Hasil Uji Pengisian Biodata kelompok besar

Jenis Kegiatan	Jumlah Data Input*	Server Database Master		Server Database Slave		Kesamaan Jumlah Data		Jumlah perbedaan jika tidak sama
		Jumlah data sebelum input	Jumlah data setelah input	Jumlah data sebelum input	Jumlah data setelah input	Sama	Tidak	
Input 5 Biodata secara Bersamaan	100	1	101	1	101	✓		0
Input 10 Biodata secara Bersamaan	200	101	301	101	301	✓		0
Input 15 Biodata secara Bersamaan	300	601	601	601	601	✓		0

Terlihat pada tabel 4.5 pengisian biodata secara bersamaan, data pada *server database master* dan *slave* tidak ada perbedaan dan memiliki jumlah kesamaan *record* yang sama.

b. Pengunggahan Dokumen

Berdasarkan hasil pengujian pengunggahan dokumen kelompok besar yang di lakukan oleh 20 orang *user* maka didapatkan data yang dapat dilihat pada Table 4.6 dibawah ini.

Tabel 4.6 Hasil Uji pengunggah dokumen kelompok besar

Jenis Kegiatan	Jumlah Data			Server Database 1		Server Database 2		Kesamaan Jumlah Data		Jumlah perbedaan jika tidak sama
	User	Banyak Data Input	total	Jumlah data sebelum input	Jumlah data setelah input	Jumlah data sebelum input	Jumlah data setelah input	Sama	Tidak	
Unggah 1kb	User 1	2	66	72	138	72	210	✓		0
	User 2	3								
	User 3	5								
	User 4	2								
	User 5	3								

	<i>User 6</i>	2							
	<i>User 7</i>	4							
	<i>User 8</i>	2							
	<i>User 9</i>	7							
	<i>User 10</i>	5							
	<i>User 11</i>	5							
	<i>User 12</i>	3							
	<i>User 13</i>	5							
	<i>User 14</i>	2							
	<i>User 15</i>	2							
	<i>User 16</i>	3							
	<i>User 17</i>	5							
	<i>User 18</i>	2							
	<i>User 19</i>	2							
	<i>User 20</i>	2							
	<i>User 1</i>	2							
	<i>User 2</i>	2							
	<i>User 3</i>	4							
	<i>User 4</i>	2							
Unggah 1Mb	<i>User 5</i>	3	61	138	199	210	409	✓	0
	<i>User 6</i>	2							
	<i>User 7</i>	3							
	<i>User 8</i>	2							
	<i>User 9</i>	5							
	<i>User 10</i>	5							

	<i>User 11</i>	5							
	<i>User 12</i>	3							
	<i>User 13</i>	5							
	<i>User 14</i>	2							
	<i>User 15</i>	2							
	<i>User 16</i>	3							
	<i>User 17</i>	5							
	<i>User 18</i>	2							
	<i>User 19</i>	2							
	<i>User 20</i>	2							
	<i>User 1</i>	2							
	<i>User 2</i>	2							
	<i>User 3</i>	3							
	<i>User 4</i>	2							
	<i>User 5</i>	2							
	<i>User 6</i>	2							
	<i>User 7</i>	2							
Unggah 5Mb	<i>User 8</i>	2	54	199	253	409	662	✓	0
	<i>User 9</i>	3							
	<i>User 10</i>	3							
	<i>User 11</i>	5							
	<i>User 12</i>	3							
	<i>User 13</i>	5							
	<i>User 14</i>	2							
	<i>User 15</i>	2							

	<i>User 16</i>	3							
	<i>User 17</i>	5							
	<i>User 18</i>	2							
	<i>User 19</i>	2							
	<i>User 20</i>	2							
	<i>User 1</i>	2							
	<i>User 2</i>	2							
	<i>User 3</i>	2							
	<i>User 4</i>	2							
	<i>User 5</i>	2							
	<i>User 6</i>	2							
	<i>User 7</i>	2							
	<i>User 8</i>	2							
	<i>User 9</i>	3							
Unggah	<i>User 10</i>	2	50	253	303	662	965	✓	0
10Mb	<i>User 11</i>	5							
	<i>User 12</i>	3							
	<i>User 13</i>	5							
	<i>User 14</i>	2							
	<i>User 15</i>	2							
	<i>User 16</i>	3							
	<i>User 17</i>	3							
	<i>User 18</i>	2							
	<i>User 19</i>	2							
	<i>User 20</i>	2							

	<i>User 1</i>	2							
	<i>User 2</i>	2							
	<i>User 3</i>	2							
	<i>User 4</i>	2							
	<i>User 5</i>	2							
	<i>User 6</i>	2							
	<i>User 7</i>	2							
	<i>User 8</i>	2							
	<i>User 9</i>	2							
Unggah 15Mb	<i>User 10</i>	2	48	303	351	965	1316	✓	0
	<i>User 11</i>	5							
	<i>User 12</i>	3							
	<i>User 13</i>	5							
	<i>User 14</i>	2							
	<i>User 15</i>	2							
	<i>User 16</i>	3							
	<i>User 17</i>	2							
	<i>User 18</i>	2							
	<i>User 19</i>	2							
	<i>User 20</i>	2							

Terlihat pada tabel 4.6 pengunggahan dokumen secara bersamaan, data pada *server database master* dan *slave* tidak ada perbedaan dan memiliki jumlah kesamaan *record* yang sama.

c. *Stress test*

Berdasarkan hasil pengujian *stress test* kelompok besar yang di lakukan oleh 20 orang *user* maka didapatkan data yang dapat dilihat pada Table 4.7 dibawah ini.

Tabel 4.7 Hasil Uji *Stress test* kelompok besar

Jenis Kegiatan	Jumlah Data			<i>Server Database Master</i>		<i>Server Database Slave</i>		Kesamaan Jumlah Data		Jumlah perbedaan jika tidak sama
	<i>User</i>	Banyak Data <i>input</i>	total	Jumlah data sebelum <i>input</i>	Jumlah data setelah <i>input</i>	Jumlah data sebelum <i>input</i>	Jumlah data setelah <i>input</i>	Sama	Tidak	
Stres Test 1	<i>User 1</i>	10	200	601	801	601	801	✓		0
	<i>User 2</i>	10								
	<i>User 3</i>	10								
	<i>User 4</i>	10								
	<i>User 5</i>	10								
	<i>User 6</i>	10								
	<i>User 7</i>	10								
	<i>User 8</i>	10								
	<i>User 9</i>	10								

	<i>User 10</i>	10							
	<i>User 11</i>	10							
	<i>User 12</i>	10							
	<i>User 13</i>	10							
	<i>User 14</i>	10							
	<i>User 15</i>	10							
	<i>User 16</i>	10							
	<i>User 17</i>	10							
	<i>User 18</i>	10							
	<i>User 19</i>	10							
	<i>User 20</i>	10							
	<i>User 1</i>	50							
	<i>User 2</i>	50							
	<i>User 3</i>	50							
	<i>User 4</i>	50							
	<i>User 5</i>	50							
	<i>User 6</i>	50							
Stres	<i>User 7</i>	50							
Test 2	<i>User 8</i>	50	1000	801	1801	801	1801	✓	0
	<i>User 9</i>	50							
	<i>User 10</i>	50							
	<i>User 11</i>	50							
	<i>User 12</i>	50							
	<i>User 13</i>	50							
	<i>User 14</i>	50							

	<i>User 15</i>	50							
	<i>User 16</i>	50							
	<i>User 17</i>	50							
	<i>User 18</i>	50							
	<i>User 19</i>	50							
	<i>User 20</i>	50							
	<i>User 1</i>	100							
	<i>User 2</i>	100							
	<i>User 3</i>	100							
	<i>User 4</i>	100							
	<i>User 5</i>	100							
	<i>User 6</i>	100							
	<i>User 7</i>	100							
	<i>User 8</i>	100							
	<i>User 9</i>	100							
Stres Test 3	<i>User 10</i>	100	2000	1801	3801	1801	3801	✓	0
	<i>User 11</i>	100							
	<i>User 12</i>	100							
	<i>User 13</i>	100							
	<i>User 14</i>	100							
	<i>User 15</i>	100							
	<i>User 16</i>	100							
	<i>User 17</i>	100							
	<i>User 18</i>	100							
	<i>User 19</i>	100							

	<i>User 20</i>	100							
	<i>User 1</i>	1000							
	<i>User 2</i>	1000							
	<i>User 3</i>	1000							
	<i>User 4</i>	1000							
	<i>User 5</i>	1000							
	<i>User 6</i>	1000							
	<i>User 7</i>	1000							
	<i>User 8</i>	1000							
	<i>User 9</i>	1000							
Stres Test 4	<i>User 10</i>	1000	20000	3801	23081	3801	23081	✓	0
	<i>User 11</i>	1000							
	<i>User 12</i>	1000							
	<i>User 13</i>	1000							
	<i>User 14</i>	1000							
	<i>User 15</i>	1000							
	<i>User 16</i>	1000							
	<i>User 17</i>	1000							
	<i>User 18</i>	1000							
	<i>User 19</i>	1000							
	<i>User 20</i>	1000							
Stres Test 5	<i>User 1</i>	4646	68925	23801	92726	23801	92726	✓	0
	<i>User 2</i>	4646							

<i>User 3</i>	2392
<i>User 4</i>	2000
<i>User 5</i>	3000
<i>User 6</i>	4500
<i>User 7</i>	4646
<i>User 8</i>	1994
<i>User 9</i>	3500
<i>User 10</i>	4321
<i>User 11</i>	3333
<i>User 12</i>	3000
<i>User 13</i>	4999
<i>User 14</i>	1007
<i>User 15</i>	4449
<i>User 16</i>	1994
<i>User 17</i>	4999
<i>User 18</i>	2000
<i>User 19</i>	2500
<i>User 20</i>	4999

Terlihat pada tabel 4.7 pengujian *stress test* secara bersamaan, data pada *server database master* dan *slave* tidak ada perbedaan dan memiliki jumlah kesamaan *record* yang sama.

4.1.5 Uji waktu replikasi database

Pada tahap ini dilakukan pengujian sistem berupa uji coba pengukuran waktu replikasi *database* yaitu dengan pengujian ini peneliti akan mengukur perbedaan waktu kedatangan *database* untuk setiap *record*-nya berdasarkan general log yang ada pada *server database master* maupun *server database slave*.

Pada table 4.8 merupakan hasil pengujian waktu replikasi *database* saat melakukan pengisian biodata pada *database server master* dan *database server slave*. Ujicoba menggunakan 1 User dengan melakukan input sebanyak 5 data.

Tabel 4.8 Hasil Uji Waktu Replikasi Database saat Peng-input-an Biodata 5 dengan 1 User

NO	JENIS KEGIATAN	Data masuk		Waktu data tiba		Sellisih waktu	Keterangan
		Server master	Server Slave	Server master	Server Slave		
1	Input data 1	✓	✓	17:18:21	17:18:21	0	
2	Input data 2	✓	✓	17:18:37	17:18:37	0	
3	Input data 3	✓	✓	17:18:46	17:18:46	0	
4	Input data 4	✓	✓	17:18:57	17:18:57	0	
5	Input data 5	✓	✓	17:19:10	17:19:10	0	

Pada table 4.9 merupakan hasil pengujian waktu replikasi *database* saat melakukan pengubahan biodata pada *database server master* dan *database server slave*. Ujicoba menggunakan 1 User dengan melakukan *edit* biodata sebanyak 5 data.

Tabel 4.9 Hasil Uji Waktu Replikasi Database saat Peng-*edit*-an Biodata 5 dengan 1 User

NO	JENIS KEGIATAN	Data masuk		Waktu data tiba		Sellisih waktu	Keterangan
		Server master	Server Slave	Server master	Server Slave		
1	Edit data1	✓	✓	17:20:03	17:20:03	0	
2	Edit data2	✓	✓	17:20:25	17:20:25	0	
3	Edit data3	✓	✓	17:20:31	17:20:31	0	
4	Edit data4	✓	✓	17:20:42	17:20:42	0	
5	Edit data5	✓	✓	17:20:58	17:20:58	0	

Pada table 4.10 merupakan hasil pengujian waktu replikasi *database* saat melakukan penghapusan biodata pada *database server master* dan *database server slave*. Ujicoba menggunakan 1 User dengan melakukan *delete* biodata sebanyak 5 data.

Tabel 4.10 Hasil Uji Waktu Replikasi Database saat Peng-*delete*-an Biodata 5 dengan 1 User

NO	JENIS KEGIATAN	Data masuk		Waktu data tiba		Sellisih waktu	Keterangan
		Server master	Server Slave	Server master	Server Slave		
1	Delete data 1	✓	✓	17:22:05	17:22:05	0	
2	Delete data 2	✓	✓	17:22:07	17:22:07	0	
3	Delete data 3	✓	✓	17:22:09	17:22:09	0	
4	Delete data 4	✓	✓	17:22:10	17:22:10	0	
5	Delete data 5	✓	✓	17:22:13	17:22:13	0	

Pada table 4.11 merupakan hasil pengujian waktu replikasi *database* saat melakukan pengisian biodata pada *database server master* dan *database server slave*. Ujicoba menggunakan 5 User dengan melakukan input sebanyak 10 data.

Tabel 4.11 Hasil Uji Waktu Replikasi Database saat Peng-input-an 10 Biodata dengan 5 User

NO	JENIS KEGIATAN	Data masuk		Waktu data tiba		Sellisih waktu	Keterangan
		Server master	Server Slave	Server master	Server Slave		
1.	Input data 1	✓	✓	18:22:19	18:22:19	0	Waktu data masuk bersamaan
2.	Input data 2	✓	✓	18:22:19	18:22:19	0	
3.	Input data 3	✓	✓	18:22:19	18:22:19	0	
4.	Input data 4	✓	✓	18:22:19	18:22:19	0	
5.	Input data 5	✓	✓	18:23:12	18:23:12	0	Waktu data masuk bersamaan
6.	Input data 6	✓	✓	18:23:12	18:23:12	0	
7.	Input data 7	✓	✓	18:23:12	18:23:12	0	
8.	Input data 8	✓	✓	18:23:12	18:23:12	0	
9.	Input data 9	✓	✓	18:23:22	18:23:22	0	Waktu data masuk bersamaan
10.	Input data 10	✓	✓	18:23:22	18:23:22	0	
11.	Input data 11	✓	✓	18:23:22	18:23:22	0	
12.	Input data 12	✓	✓	18:23:22	18:23:22	0	
13.	Input data 13	✓	✓	18:23:30	18:23:30	0	Waktu data masuk bersamaan
14.	Input data 14	✓	✓	18:23:30	18:23:30	0	
15.	Input data 15	✓	✓	18:23:30	18:23:30	0	
16.	Input data 16	✓	✓	18:23:32	18:23:32	0	
17.	Input data 17	✓	✓	18:23:38	18:23:38	0	Waktu data masuk bersamaan
18.	Input data 18	✓	✓	18:23:38	18:23:38	0	
19.	Input data 19	✓	✓	18:23:38	18:23:38	0	
20.	Input data 20	✓	✓	18:23:39	18:23:39	0	
21.	Input data 21	✓	✓	18:23:45	18:23:45	0	Waktu data masuk bersamaan
22.	Input data 22	✓	✓	18:23:45	18:23:45	0	
23.	Input data 23	✓	✓	18:23:45	18:23:45	0	
24.	Input data 24	✓	✓	18:23:46	18:23:46	0	
25.	Input data 25	✓	✓	18:23:53	18:23:53	0	Waktu data masuk bersamaan
26.	Input data 26	✓	✓	18:23:53	18:23:53	0	
27.	Input data 27	✓	✓	18:23:54	18:23:54	0	
28.	Input data 28	✓	✓	18:23:57	18:23:57	0	
29.	Input data 29	✓	✓	18:23:59	18:23:59	0	Waktu data masuk bersamaan
30.	Input data 30	✓	✓	18:23:59	18:23:59	0	
31.	Input data 31	✓	✓	18:24:04	18:24:04	0	
32.	Input data 32	✓	✓	18:24:11	18:24:11	0	
33.	Input data 33	✓	✓	18:24:11	18:24:11	0	Waktu data masuk bersamaan
34.	Input data 34	✓	✓	18:24:11	18:24:11	0	
35.	Input data 35	✓	✓	18:24:12	18:24:12	0	
36.	Input data 36	✓	✓	18:24:20	18:24:20	0	
37.	Input data 37	✓	✓	18:24:20	18:24:20	0	Waktu data masuk bersamaan
38.	Input data 38	✓	✓	18:24:21	18:24:21	0	

39.	Input data 39	✓	✓	18:24:21	18:24:21	0	Waktu data masuk bersamaan
40.	Input data 40	✓	✓	18:24:26	18:24:26	0	
41.	Input data 41	✓	✓	18:24:30	18:24:30	0	Waktu data masuk bersamaan
42.	Input data 42	✓	✓	18:24:30	18:24:30	0	
43.	Input data 43	✓	✓	18:24:30	18:24:30	0	
44.	Input data 44	✓	✓	18:24:34	18:24:34	0	
45.	Input data 45	✓	✓	18:24:41	18:24:41	0	Waktu data masuk bersamaan
46.	Input data 46	✓	✓	18:24:41	18:24:41	0	
47.	Input data 47	✓	✓	18:24:41	18:24:41	0	
48.	Input data 48	✓	✓	18:24:44	18:24:44	0	Waktu data masuk bersamaan
49.	Input data 49	✓	✓	18:24:44	18:24:44	0	
50.	Input data 50	✓	✓	18:24:44	18:24:44	0	

Pada table 4.12 merupakan hasil pengujian waktu replikasi *database* saat melakukan perubahan biodata pada *database server master* dan *database server slave*. Ujicoba menggunakan 1 User dengan melakukan *edit* biodata sebanyak 5 data.

Tabel 4.12 Hasil Uji Waktu Replikasi Database saat Peng-edit-an Biodata 10 dengan 5 User

NO	JENIS KEGIATAN	Data masuk		Waktu data tiba		Sellisih waktu	Keterangan
		Server master	Server Slave	Server master	Server Slave		
1	Edit data 1	✓	✓	18:24:45	18:24:45	0	
2	Edit data 2	✓	✓	18:24:50	18:24:50	0	
3	Edit data 3	✓	✓	18:24:58	18:24:58	0	
4	Edit data 4	✓	✓	18:25:01	18:25:01	0	
5	Edit data 5	✓	✓	18:25:15	18:25:15	0	
6	Edit data 6	✓	✓	18:25:24	18:25:24	0	
7	Edit data 7	✓	✓	18:25:37	18:25:37	0	
8	Edit data 8	✓	✓	18:25:41	18:25:41	0	
9	Edit data 9	✓	✓	18:25:44	18:25:44	0	
10	Edit data 10	✓	✓	18:25:46	18:25:46	0	
11	Edit data 11	✓	✓	18:25:49	18:25:49	0	
12	Edit data 12	✓	✓	18:25:57	18:25:57	0	
13	Edit data 13	✓	✓	18:26:07	18:26:07	0	
14	Edit data 14	✓	✓	18:26:16	18:26:16	0	
15	Edit data 15	✓	✓	18:26:25	18:26:25	0	
16	Edit data 16	✓	✓	18:26:29	18:26:29	0	

17	Edit data 17	✓	✓	18:26:32	18:26:33	1	Delay 1 detik
18	Edit data 18	✓	✓	18:26:33	18:26:33	0	
19	Edit data 19	✓	✓	18:26:40	18:26:40	0	
20	Edit data 20	✓	✓	18:26:51	18:26:51	0	Waktu data
21	Edit data 21	✓	✓	18:26:51	18:26:51	0	masuk bersamaan
22	Edit data 22	✓	✓	18:26:54	18:26:54	0	
23	Edit data 23	✓	✓	18:27:09	18:27:09	0	Waktu data
24	Edit data 24	✓	✓	18:27:09	18:27:09	0	masuk bersamaan
25	Edit data 25	✓	✓	18:27:13	18:27:13	0	
26	Edit data 26	✓	✓	18:27:26	18:27:26	0	
27	Edit data 27	✓	✓	18:27:28	18:27:28	0	
28	Edit data 28	✓	✓	18:27:31	18:27:31	0	Waktu data
29	Edit data 29	✓	✓	18:27:31	18:27:31	0	masuk bersamaan
30	Edit data 30	✓	✓	18:27:38	18:27:38	0	
31	Edit data 31	✓	✓	18:27:39	18:27:39	0	
32	Edit data 32	✓	✓	18:27:47	18:27:47	0	Waktu data
33	Edit data 33	✓	✓	18:27:47	18:27:47	0	masuk bersamaan
34	Edit data 34	✓	✓	18:27:53	18:27:53	0	
35	Edit data 35	✓	✓	18:27:55	18:27:55	0	Waktu data
36	Edit data 36	✓	✓	18:27:55	18:27:55	0	masuk bersamaan
37	Edit data 37	✓	✓	18:27:59	18:27:59	0	
38	Edit data 38	✓	✓	18:28:13	18:28:13	0	
39	Edit data 39	✓	✓	18:28:14	18:28:14	0	Waktu data
40	Edit data 40	✓	✓	18:28:14	18:28:14	0	masuk bersamaan
41	Edit data 41	✓	✓	18:28:17	18:28:17	0	
42	Edit data 42	✓	✓	18:28:22	18:28:22	0	
43	Edit data 43	✓	✓	18:28:31	18:28:31	0	Waktu data
44	Edit data 44	✓	✓	18:28:31	18:28:31	0	masuk bersamaan
45	Edit data 45	✓	✓	18:28:31	18:28:31	0	
46	Edit data 46	✓	✓	18:28:35	18:28:35	0	
47	Edit data 47	✓	✓	18:28:37	18:28:37	0	Waktu data
48	Edit data 48	✓	✓	18:28:37	18:28:37	0	masuk bersamaan
49	Edit data 49	✓	✓	18:28:52	18:28:52	0	
50	Edit data 50	✓	✓	18:29:16	18:29:16	0	

Pada table 4.13 merupakan hasil pengujian waktu replikasi *database* saat melakukan penghapusan biodata pada *database server master* dan *database server slave*. Ujicoba menggunakan 1 user dengan melakukan *edit* biodata sebanyak 5 data.

Tabel 4.13 Hasil Uji Waktu Replikasi Database saat Pen-*delete*-an Biodata 10 dengan 5 User

NO	JENIS KEGIATAN	Data masuk		Waktu data tiba		Sellisih waktu	Keterangan
		Server master	Server Slave	Server master	Server Slave		
1	Delete data 1	✓	✓	18:29:28	18:29:28	0	
2	Delete data 2	✓	✓	18:29:38	18:29:38	0	Waktu data masuk bersamaan
3	Delete data 3	✓	✓	18:29:38	18:29:38	0	
4	Delete data 4	✓	✓	18:29:41	18:29:41	0	Waktu data masuk bersamaan
5	Delete data 5	✓	✓	18:29:41	18:29:41	0	
6	Delete data 6	✓	✓	18:29:43	18:29:43	0	
7	Delete data 7	✓	✓	18:29:49	18:29:49	0	
8	Delete data 8	✓	✓	18:29:50	18:29:50	0	
9	Delete data 9	✓	✓	18:29:51	18:29:51	0	
10	Delete data 10	✓	✓	18:29:58	18:29:58	0	Waktu data masuk bersamaan
11	Delete data 11	✓	✓	18:29:58	18:29:58	0	
12	Delete data 12	✓	✓	18:30:00	18:30:00	0	
13	Delete data 13	✓	✓	18:30:07	18:30:07	0	
14	Delete data 14	✓	✓	18:30:08	18:30:08	0	
15	Delete data 15	✓	✓	18:30:09	18:30:09	0	
16	Delete data 16	✓	✓	18:30:11	18:30:11	0	Waktu data masuk bersamaan
17	Delete data 17	✓	✓	18:30:11	18:30:11	0	
18	Delete data 18	✓	✓	18:30:13	18:30:13	0	
19	Delete data 19	✓	✓	18:30:15	18:30:15	0	
20	Delete data 20	✓	✓	18:30:18	18:30:18	0	Waktu data masuk bersamaan
21	Delete data 21	✓	✓	18:30:18	18:30:18	0	
22	Delete data 22	✓	✓	18:30:19	18:30:19	0	Waktu data masuk bersamaan
23	Delete data 23	✓	✓	18:30:19	18:30:19	0	
24	Delete data 24	✓	✓	18:30:21	18:30:21	0	
25	Delete data 25	✓	✓	18:30:23	18:30:23	0	
26	Delete data 26	✓	✓	18:30:24	18:30:24	0	
27	Delete data 27	✓	✓	18:30:27	18:30:27	0	
28	Delete data 28	✓	✓	18:30:28	18:30:28	0	
29	Delete data 29	✓	✓	18:30:29	18:30:29	0	
30	Delete data 30	✓	✓	18:30:33	18:30:33	0	
31	Delete data 31	✓	✓	18:30:36	18:30:36	0	
32	Delete data 32	✓	✓	18:30:38	18:30:38	0	
33	Delete data 33	✓	✓	18:30:40	18:30:41	1	<i>Delay 1 detik</i>
34	Delete data 34	✓	✓	18:30:43	18:30:43	0	Waktu data masuk bersamaan
35	Delete data 35	✓	✓	18:30:43	18:30:43	0	
36	Delete data 36	✓	✓	18:30:45	18:30:45	0	
37	Delete data 37	✓	✓	18:30:47	18:30:47	0	
38	Delete data 38	✓	✓	18:30:51	18:30:51	0	

39	Delete data 39	✓	✓	18:30:57	18:30:57	0	
40	Delete data 40	✓	✓	18:31:00	18:31:01	1	<i>Delay 1 detik</i>
41	Delete data 41	✓	✓	18:31:06	18:31:06	0	
42	Delete data 42	✓	✓	18:31:18	18:31:18	0	
43	Delete data 43	✓	✓	18:31:21	18:31:21	0	
44	Delete data 44	✓	✓	18:31:28	18:31:28	0	
45	Delete data 45	✓	✓	18:31:31	18:31:31	0	
46	Delete data 46	✓	✓	18:31:37	18:31:37	0	
47	Delete data 47	✓	✓	18:31:39	18:31:41	0	
48	Delete data 48	✓	✓	18:31:45	18:31:45	0	
49	Delete data 49	✓	✓	18:31:49	18:31:49	0	
50	Delete data 50	✓	✓	18:31:51	18:31:51	0	

Pada table 4.14 merupakan hasil pengujian waktu replikasi *database* saat melakukan pengunggahan file yang beragam dengan beberapa besar data yang berbeda Ujicoba menggunakan 1 user dengan melakukan pengunggahan sebanyak 20 *file* yang berbeda-beda ukurannya

Tabel 4.14 Hasil Uji Waktu Replikasi Database saat Pengunggahan File

NO	JENIS KEGIATAN	Data masuk		Waktu data tiba		Sellisih waktu	Keterangan
		Server master	Server Slave	Server master	Server Slave		
1	Upload 1kb	✓	✓	18:57:09	18:57:09	0	
2	Upload 1kb	✓	✓	18:57:09	18:57:09	0	
3	Upload 1mb	✓	✓	18:57:19	18:57:19	0	
4	Upload 5mb	✓	✓	18:57:34	18:57:35	1	<i>Delay 1 detik</i>
5	Upload 10mb	✓	✓	18:57:54	18:57:55	1	<i>Delay 1 detik</i>
6	Upload 15mb	✓	✓	18:58:40	18:58:43	3	<i>Delay 3 detik</i>
7	Upload 1kb	✓	✓	18:58:46	18:58:46	0	
8	Upload 15mb	✓	✓	18:58:51	18:58:54	3	<i>Delay 3 detik</i>
9	Upload 15mb	✓	✓	18:59:40	18:59:43	3	<i>Delay 3 detik</i>
10	Upload 1kb	✓	✓	18:59:47	18:59:47	0	
11	Upload 15mb	✓	✓	18:59:50	18:59:54	4	<i>Delay 4 detik</i>
12	Upload 15mb	✓	✓	18:59:57	19:00:00	3	<i>Delay 3 detik</i>
13	Upload 5mb	✓	✓	19:00:01	19:00:02	1	<i>Delay 1 detik</i>
14	Upload 1kb	✓	✓	19:00:21	19:00:21	0	
15	Upload 5mb	✓	✓	19:00:37	19:00:37	0	
16	Upload 10mb	✓	✓	19:00:38	19:00:40	2	<i>Delay 2 detik</i>
17	Upload 15mb	✓	✓	19:00:42	19:00:45	3	<i>Delay 3 detik</i>
18	Upload 20mb	✓	✓	19:00:51	19:00:55	4	<i>Delay 4 detik</i>
19	Upload 30mb	✓	✓	19:01:13	19:01:19	6	<i>Delay 6 detik</i>
20	Upload 50mb	✓	✓	19:01:25	19:01:35	10	<i>Delay 10 detik</i>

4.1.6 Uji Fungsi *Recovery Database*

Pada tahap ini dilakukan pengujian sistem berupa uji coba fungsi *recovery database* yaitu dengan pengujian berfungsi atau tidak sebuah website yang terkoneksi dengan *database master* jika diskenariokan *server database master* bermasalah atau mati. pada pengujian ini kita juga mengujikan peran *server database sale* dalam menggantikannya.

Pada table 4.15 merupakan hasil pengujian *recovery database* saat pengisian biodata dan *database server masternya* di matikan.

Tabel 4.15 Hasil Uji Fungsi *Recovery* saat Pengisian Biodata

Jenis Kegiatan	Jumlah Data		Server Database Master	Server Database Slave		data masuk	
	User	Banyak input	kondisi	Jumlah data sebelum input	Jumlah data setelah input	ya	tidak
menginput biodata	user 1	5	mati	159	164	✓	
	user 2	5	mati	164	169	✓	
	user3	5	mati	169	174	✓	

Dari table 4.15 diatas dapat kita lihat *web server* tetap dapat berfungsi menyimpan *database* meskipun kondisi databse *server master* dalam keadaan mati. Sedangkan pada table 4.16 merupakan hasil pengujian *recovery database* saat mengunggah *file* dan *database server masternya* di matikan.

Tabel 4.16 Hasil Uji Fungsi *Recovery* saat mengunggah *file*

Jenis Kegiatan	Jumlah Data		Server Database Master	Server Database Slave		data masuk	
	User	Banyak input	kondisi	Jumlah data sebelum input	Jumlah data setelah input	ya	tidak
Mengunggah file	user 1	2	mati	89	91	✓	
	user 2	2	mati	91	93	✓	
	user3	2	mati	93	95	✓	

Dari table 4.16 diatas dapat kita lihat *Web server* tetap dapat berfungsi menyimpan *database* meskipun kondisi databse *server master* dalam keadaan mati.

Dan pada table 4.17 merupakan hasil pengujian *recovery database* saat *stress test* dan *database server masternya* di matikan.

Tabel 4.17 Hasil Uji Fungsi *Recovery* saat *Stress test*

Jenis Kegiatan	Jumlah Data		Server Database Master	Server Database Slave		data masuk	
	User	Banyak input	kondisi	Jumlah data sebelum input	Jumlah data setelah input	ya	tidak
Stress test	user 1	1005	mati	174	1179	✓	
	user 2	888	mati	1179	2067	✓	
	user3	1094	mati	2067	3161	✓	

Dari table 4.17 diatas dapat kita lihat *Web server* tetap dapat berfungsi menyimpan *database* meskipun kondisi databse *server master* dalam keadaan mati.

4.2 Pembahasan

Pengembangan *disaster recovery database server* dimulai dengan menganalisa kebutuhan fitur dan fungsi yang akan dipasang pada sistem, hal ini dilakukan dengan analisa lapangan dan analisa teoretik. proses pengembangan dilakukan dengan metode *prototyping* dengan melakukan uji validitas replikasi pada *database* baik pada *server master database* maupun pada *server slave database*.

Setelah proses validasi replikasi produk selesai maka tahapan selanjutnya adalah membuat *website* untuk menguji kehandalan replikasi *database* kedua *server database*. Pada *website* pengujian tersedia fungsi fungsi dasar pengisian biodata, perubahan biodata dan penghapusan biodata. Selain itu ada juga fungsi untuk mengunggah dan mengunduh *file* hal ini berfungsi untuk menguji replikasi *database* menggunakan *file* yang berbeda. Setelah itu dilakukan uji fungsionalitas, yaitu menguji setiap fungsi dan fitur yang ada pada *website* pengujian untuk mengetahui ada atau tidaknya bug atau kesalahan pada proses pengembangan sistem. Hasil uji fungsionalitas menunjukkan adanya beberapa bug yang masih belum bekerja sesuai harapan, kemudian untuk memperbaikinya dilakukan tahapan revisi produk 2 dengan memperbaiki bug yang muncul pada *website* pengujian.

Tahapan lanjutan adalah menguji replikasi menjadi dua pengujian yaitu dengan kelompok kecil dan kelompok besar. Pada kelompok kecil peneliti menggunakan 5 orang untuk menguji replikasi. Seluruh penguji menguji melakukan manipulasi *database* menggunakan *website* pengujian. Dari hasil uji terbukti *database* yang ada pada *server database master* sama dan identik dengan yang ada pada *server database slave*.

Lanjut ke tahap berikutnya adalah pengujian menggunakan kelompok besar menggunakan 20 orang untuk menguji dengan skenario uji yang sama dengan kelompok kecil. Ternyata dari *table* di atas kita bisa melihat tidak terdapat perbedaan antara *server database master* dan *server database slave*.

Selanjutnya adalah pengujian waktu replikasi pada pengujian ini dilakukan penghitungan waktu antara kedatangan *record database* pada *server database master* dan kedatangan *record* pada *server database slave*. Pada pengujian ini dilakukan beberapa perlakuan baik pada tipe data, besar data dan jumlah *user* yang melakukan pengujian, dari tipe data dilakukan dua perlakuan yang berbeda antara lain data *char* dan data *file*. Dari dari besar data yang di-*input*-kan ada data 1kb, 1mb, 5mb, 15mb, 30mb, dan 50mb. Untuk user terdiri dari 1 *user* dan 5 *user* yang melakukan peng-*input*-an. Dari pengambilan data waktu replikasi dapat disimpulkan semakin besar data dan jumlah *user* yang melakukan manipulasi *database* maka akan mempengaruhi waktu tiba replikasi di *server slave*. *Delay* waktu hanya terjadi pada *file-file* ukuran besar *delay* pun tidak terlalu lama hanya dalam hitungan detik.

Tahapan lanjutan setelah menguji replikasi dengan dua adalah pengujian *recovery*, pengujian *recovery* masih menggunakan *website* pengujian yang sama tetapi akan ada skenario dimana *server database master* dimatikan sehingga *database* tidak bisa diakses ke *website*, akan tetapi dikarenakan kita memiliki *backup database* yang identik dengan *server master* maka *server database slave* dapat menggantikannya di saat *server database master down*.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis, hasil pengembangan produk, hasil pengujian dan revisi produk serta pembahasan yang telah dipaparkan sebelumnya, maka dapat ditarik kesimpulan sebagai berikut:

1. Hasil dari penelitian ini adalah sebuah produk, yaitu purwarupa *disaster recovery database server*.
2. Pada Penelitian ini dilakukan konfigurasi *my.cnf default* dengan modifikasi pada *server-id=1 bind-address=192.168.16.250* untuk *server* master dan modifikasi pada *server-id = 2 bind-address=192.168.16.240* untuk *server* slave.
3. Purwarupa *disaster recovery database server* telah diuji dan berdasarkan hasil pengujian dengan kondisi menggunakan 1 *user* yang melakukan peng-*input*-tan, peng-*update*-an dan pen-*delete*-an 5 buah biodata yang bertipe *text* tidak memiliki *delay* waktu dan memiliki tingkat keakuratan 100%. Dengan kondisi menggunakan 5 *user* yang melakukan peng-*input*-tan, peng-*update*-an dan pen-*delete*-an 10 buah biodata yang bertipe *text* database ada beberapa yang tiba bersamaan, ada satu *record* yang mengalami *delay* dan memiliki tingkat keakuratan 100%. Serta untuk melakukan peng-*upload*-an *file* 1kb tidak memiliki *delay* waktu, peng-*upload*-an *file* 1mb tidak memiliki *delay* waktu, peng-*upload*-an *file* 5mb memiliki *delay* waktu 1 detik, peng-*upload*-an *file* 10mb memiliki *delay*

waktu 2 detik, peng-*upload-an file* 15mb memiliki *delay* waktu 3 detik. peng-*upload-an file* 20mb memiliki *delay* waktu 4 detik, peng-*upload-an file* 30mb memiliki *delay* waktu 6 detik dan peng-*upload-an file* 50mb memiliki *delay* waktu 10 detik saat melakukan replikasi *database* dan memiliki tingkat keakuratan 100% yang sama antara *database master* dan *slave*.

4. *Delay* replikasi *database* terpengaruh pada besarnya data dan banyaknya data yang di replikasi. Serta banyaknya *user* yang melakukan peng-*input-tan*, peng-*update-an* dan pen-*delete-an* pada suatu *database*.
5. Purwarupa *disaster recovery database server* yang dikembangkan telah diuji dan berdasarkan hasil pengujian pada poin no 3. *server slave* dapat langsung menggantikan peran *server master* ketika *server master* mengalami masalah atau *down*.

5.2 Saran

Demi kelanjutan penelitian yang akan datang, saran yang dapat diajukan oleh peneliti setelah melakukan penelitian adalah:

1. Penelitian ini dapat dikembangkan lebih mendalam dengan melakukan penelitian pada modifikasi pada konfigurasi *my.cnf* dan melakukan perubahan jumlah *user* dan data, pada saat dilakukan pengambilan data.
2. Pengembangan fitur tambahan seperti *notifikasi* kepada *adminastrator* jaringan apa bila terjadi masalah baik pada *server* maupun pada replikasi *datasenya*.

3. Pengembangan *disaster recovery server* bukan hanya *membackup database* saja kedepan seharusnya ada suatu sistem yang dapat *membackup data* secara keseluruhan.
4. Kompatibilitas *Disaster Recovery server* yang didukung pada semua sistem operasi *server* lainnya.
5. Demi faktor keamanan sebaiknya *server slave* pada sistem *Disaster Recovery server* terdiri dari beberapa *server* dan berada di beberapa tempat yang berbeda.

DAFTAR PUSTAKA

- Andi Dkk. 2010. Linux System Administrator. Bandung
- Kadir, Abdul. 2003. Pengenalan Sistem Informasi. Yogyakarta: Andi Offset
- Poerbaningtyas, Evy. 2009. Manajemen Sistem Terdistribusi. Jakarta: Graha Ilmu
- Sofana, Iwan. 2009. Cisco Ccna Dan Jaringan Komputer. Bandung: Informatika
- Sofana, Iwan. 2013. Membangun Jaringan Komputer. Bandung: Informatika
- Subekti, M. 2004. Sistem Manajemen Basis Data. Bogor: Ghalia Indonesia
- Sugiono. 2014. Metode Penelitian Kuantitatif, Kualitatif Dan R & D. Bandung: Alfabeta
- Syafrizal, Melwin. 2009. Pengantar Jaringan Computer. Yogyakarta: Andi Yogyakarta
- Tim Penyusun. 2012. Buku Pedoman Skripsi/Komprehensif/Karya Inofatif. Jakarta. Universitas Negeri Jakarta
- Time Exelindo, 2014. Disaster Recovery Center. [Terhubung Berkala] <http://www.Te.Net.Id/V2/Produk/Disaster-Recovery-Center.aspx> [25 Oktober 2015 Pukul 12:15]
- Toigo, Jon William. 1989. Disaster Recovery Planning. Prentice Hall: Inc
- Wikipedia Terverifikasi. 2015 [Terhubung Berkala] <https://Id.Wikipedia.Org/Wiki/Mysql> [12 November 2015 Pukul 14:23]
- Zulhikam, Ahmad. 2011. Fungsi Dan Jenis – Jenis Komputer Server [Terhubung Berkala] <Http://Jaringankomputer.Org/Server-Fungsi-Pengertian-Jenis-Jenis-Komputer-Server/> [21 Oktober 2015 Pukul 15:25]

Lampiran 1 : Profil Universitas Negeri Jakarta



Universitas Negeri Jakarta (UNJ) menyelenggarakan program kependidikan dan non-kependidikan sebagaimana amanat yang diberikan pemerintah kepada UNJ melalui Keppres No. 093/1999 tanggal 4 Agustus 1999. Program kependidikan diselenggarakan untuk menyediakan tenaga kependidikan. Sebagai Lembaga Pendidikan Tenaga Kependidikan (LPTK) maka UNJ akan tetap mengembangkan program kependidikan, sebagaimana juga tercantum dalam visi dan misinya. Sedangkan penyelenggaraan program non-kependidikan akan terus ditingkatkan untuk menyediakan tenaga non-kependidikan dan mendukung penyelenggaraan program kependidikan.

Sistem pendidikan UNJ terdiri dari tiga strata, yakni: Sarjana atau Strata-1 (S1), Magister atau Strata-2 (S2), dan Doktor atau Strata-3 (S3). Selain menyelenggarakan ketiga strata tersebut, UNJ juga menyelenggarakan Pendidikan Vokasi dan Pendidikan Profesi. UNJ memiliki empat kampus dengan lokasi yang berbeda, lokasi kampus yaitu:

1. Kampus A (Pusat) Jl. Rawamangun Muka Jakarta Timur 13220
2. Kampus B (FMIPA & FIK) Jl. Pemuda 10, Rawamangun Jakarta Timur 13220
3. Kampus D (Psikologi) Jl. Halimun No.2 Jakarta Selatan
4. Kampus E (PGSD) Jl. Setiabudi No.1 Kelurahan Setiabudi Jakarta Selatan 12910

Lampiran 2 : Profil PUSTIKOM UNJ



Pusat Teknologi Informasi dan Komputer (PUSTIKOM) merupakan salah satu pusat yang mengelola pemanfaatan informasi dan komunikasi untuk mendukung kegiatan pendidikan, penelitian, pengabdian kepada masyarakat, penyimpanan dan pengolahan data serta dukungan bagi manajemen universitas. PUSTIKOM berlokasi di gedung A Universitas Negeri Jakarta (UNJ). PUSTIKOM UNJ dibentuk berdasarkan SK Rektor UNJ no. 166/SP/2013 tentang “Pengapusan UPT PPTI dan Pembentukan PUSTIKOM UNJ”. PUSTIKOM bersama dengan PUSKOM (yang dibentuk pada 15 Maret 2007 sesuai dengan KepMen no. 280/0/1999 tentang “Organisasi dan Tata Kerja Universitas Negeri Jakarta serta Statuta UNJ) berfungsi untuk mendukung pengembangan teknologi informasi dan komunikasi universitas, mendukung operasional dan manajemen universitas, mendukung penyelenggaraan kegiatan pembelajaran universitas, dan mendukung data dan informasi untuk pengambilan keputusan pimpinan dan strategi pengembangan universitas (diambil dari draft SOTK UNJ 2013).

Layanan PUSTIKOM

Sesuai dengan tugas dan fungsinya, PUSTIKOM menyediakan layanan data dan informasi melalui jaringan local (LAN dan wi-Fi) dan jaringan Internet. Jaringan local menghubungkan kampus A, kampus B, kampus E, dan kampus F, sedangkan jaringan Internet menjadi jembatan antara universitas dengan dunia luas kampus. Beberapa layanan yang disediakan adalah:

1. Layanan administrasi akademik berbasis web.

- a) Bagi Mahasiswa: pendaftaran calon mahasiswa baru, entry biodata, registrasi Mahasiswa baru dan Mahasiswa lama, aktifasi Nomor Induk Mahasiswa (NIM), pengisian Kartu Rencana Studi (KRS), serta akses berbagai data dan informasi yang berkaitan dengan akademik, perkuliahan, beasiswa, dan kemahasiswaan.
- b) Bagi Alumni: entry biodata alumni, akses berbagai data dan informasi alumni, serta bursa kerja.
- c) Unit kerja di lingkungan UNJ: entry dan akses berbagai data dan informasi sesuai dengan tupoksi setiap unit.
- d) Bagi mitra UNJ: akses berbagai data dan informasi terkait dengan hubungan public.

2. Layanan dukungan pembelajaran on line dan off line.

- a) Layanan email bagi seluruh mahasiswa, dosen, dan karyawan dengan kapasitas minimal 10 GB.
- b) Layanan milis bagi mahasiswa, dosen, dan karyawan.
- c) Layanan video conference antar institusi/lembaga pendidikan untuk proses pembelajaran atau diskusi dengan memanfaatkan teknologi informasi melalui
- d) jaringan Internet.

Lampiran 3 : Borang Pengujian Analisis Data

1. uji input biodata bersamaan

Jenis Kegiatan	Jumlah Data Input*	Server Database 1		Server Database 2		Kesamaan Jumlah Data		Jumlah perbedaan jika tidak sama
		Jumlah data sebelum input	Jumlah data setelah input	Jumlah data sebelum input	Jumlah data setelah input	Sama	Tidak	
Input 5 Biodata secara Bersamaan								
Input 10 Biodata secara Bersamaan								
Input 15 Biodata secara Bersamaan								

- = (banyak data input * jumlah korisponden)

Lampiran 4 : Instalasi Sistem Operasi

1. Nyalahkan *server* yang akan kita gunakan
2. Siapkan *Bootable* berupa CD atau Flashdisk yang akan digunakan
3. Arahkan *booting* utama menjadi *drive bootable*.
4. Pada jendela instalasi yang muncul pertama kali, pilih *Install or upgrade an existing system*, lalu pilih *Next*.
5. Pada pilihan pemeriksaan media, pilih *Skip* untuk melanjutkan ke tahap berikutnya, lalu *Next*.
6. Pada pilihan bahasa, pilih *English (English)*, lalu *Next*.
7. Pada pilihan standard keyboard yang digunakan sistem, pilih *U.S.English* lalu *Next*.
8. Pada pilihan *Device Type* yang akan diinstal, pilih *Basic Storage Device*, lalu *Next*.
9. Pada jendela *pop-up Storage Device Warning* yang muncul, centang *Apply my choice to all with undetected partitions or filesystems*, setelah itu pilih *Yes, discard any data*.
10. Pada jendela nama hostname, tetap dengan nama *localhost*, lalu *Next*.
11. Pada pilihan City & Timezone, pilih Asia/Jakarta, lalu *Next*.
12. Kemudian masukan Root Password dan konfirmasi ulang password yang dimasukan, lalu *Next*.
13. Pada pilihan format harddisk, pilih *Use All Space*, klik *Next*, kemudian pada
14. Kemudian pada pilihan instalasi software CentOS pilih *Web Server*, lalu *Next*.
15. Tunggu hingga proses instalasi selesai, hingga otomatis melakukan reboot.

Lampiran 5 : Pemasangan Program Pendukung

1. Pada server Cantos isikan Username pada isian “login as:”, setelah itu tekan tombol Enter
2. Isikan password sesuai pada password pada saat pertama kali menginstalasi OS.
3. Pastikan sudah masuk ke super user
root@localhost ~]#
4. Selanjutnya kita akan mengkonfigurasi jaringan komputer pada server dengan cara mengetik “Ifconfig –a” pada command line, hal ini untuk memastikan apakah ethernet Card sudah terdeteksi oleh sistem operasi.
5. Setelah itu kita akan mengedit konfigurasi ethernet server agar dapat berkomunikasi dengan jaringan komputer baik LAN maupun wan, dengan cara mengetik “nano /etc/sysconfig/network-scripts/ifcfg-eth0” pada command line.
6. Kali ini kita akan memberi alamat IP pada server dengan cara mengubah konfigurasi pada ifcfg-eth0 menjadi seperti berikut:

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.16.111
NETMASK=255.255.255.0
GATEWAY=192.168.16.1
DNS1=8.8.8.8
DNS2=8.8.4.4
```

7. Kemudian instal SSH pada Server, hal ini bertujuan agar konfigurasi Server lebih fleksibel karena bisa di kontrol dari jarak jauh dan dapat dibuka dari komputer mana saja, Untuk menginstalasi SSH pada server dengan cara ketik “yum -y install openssh-server openssh-server”, tunggu beberapa saat sampai program benar benar terunduh sempurna, setelah itu lanjutkan instalasi dengan mengikuti petunjuk, hingga selesai.
8. Setelah sudah terinstal SSH pada server, lalu aktifkan ssh pada server dengan cara mengetik perintah “chkconfig sshd on” pada command line

9. Selanjutnya restart SSH dengan perintah “/etc/init.d/sshd restart”, pada command line.
10. Setelah itu , update sistem operasi Cantos agar menjadi lebih mutakhir pertama harus mengubah lokasi webrepository update linux terlebih dahulu dengan mengetikan “rpm -Uvh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm”
11. Selanjutnya barulah mengupdate sistem operasi dengan cara perintah “yum update -y” pada command line.
12. tunggu beberapa saat dan ikuti proses update sampai dengan selesai, hal ini mungkin berjalan lama tergantung pada koneksi internet dan besaran update sistem yang tersedia.
13. Setelah sistem operasi selesai diupdate, selanjutnya adalah menginstal paket program antarlain (mysql, php, appace, dan phpmyadmin) di server cantos untuk membangun dasar sistem replikasi database, dengan cara mengetik code berikut “yum install mysql mysql-server php httpd phpmyadmin” pada command line
14. tunggu beberapa saat sampai penginstalan benar banar selesai dan seluruh program telah terpasang pada server.
15. setelah program terinstal sempurna selanjut nya kita akan mengkonfigurasi program tadi agar otomatis terbuka ketika server menyala, dengan cara mengetik kode chkconfig httpd on pada commandline untuk program apache
chkconfig mysqld on

service httpd start

service mysqld start
16. setelah itu program sudah bisa digunakan dan terpasang pada perangkat server

Lampiran 6 : Pemasangan dan pengaturan Mysql

1. Install replikasi pada masing masing server
2. Buatlah 3 user untuk keperluan replikasi ini: mysqlchkuser, mmm_monitor, mmm_agent, dan replication dengan mengikuti gambar dibawah

```

root@sql1:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1028
Server version: 5.5.38-0ubuntu0.12.04.1-log (Ubuntu)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All
rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or
its affiliates. Other names may be trademarks of their
respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.


mysql> create user 'mysqlchkuser'@'localhost' identified by
'rahasia';
Query OK, 0 rows affected (0.06 sec)
mysql> create user 'mmm_monitor'@'%' identified by 'rahasia';
Query OK, 0 rows affected (0.06 sec)
mysql> create user 'mmm_agent'@'%' identified by 'rahasia';
Query OK, 0 rows affected (0.00 sec)
mysql> create user 'replication'@'%' identified by 'rahasia';
Query OK, 0 rows affected (0.00 sec)
mysql> GRANT REPLICATION CLIENT ON *.* TO 'mmm_monitor'@'%'
IDENTIFIED BY 'rahasia';
Query OK, 0 rows affected (0.00 sec)
mysql> GRANT SUPER, REPLICATION CLIENT, PROCESS ON *.* TO
'mmm_agent'@'%' IDENTIFIED BY 'rahasia';
Query OK, 0 rows affected (0.00 sec)
mysql> GRANT REPLICATION SLAVE ON *.* TO 'replication'@'%'
IDENTIFIED BY 'rahasia';
Query OK, 0 rows affected (0.00 sec)
mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)
mysql> flush tables with read lock;
Query OK, 0 rows affected (0.00 sec)
mysql> show master status;

+-----+-----+-----+-----+
+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB
|
+-----+-----+-----+-----+
+
| mysql-bin.000001 |      1044 |              |
|
+-----+-----+-----+-----+
+
1 row in set (0.00 sec)

```


3. Jalankan slave pada master server 2 dan cek status slave-nya. Dengan mengetikan `start slave`; setelah itu cek slave status nya dengan cara mengetikan `show slave status \`
4. Cek status server master 2 dengan mengetik `show master status`;
5. Ubah master mysql dengan master host sesuai IP master server 2, sesuaikan port, user, password dan file log-nya. `mysql> CHANGE MASTER TO master_host='192.168.1.240', master_port=3306, master_user='repli`
6. Jalankan slave dan cek statusnya `show slave status\G`
7. Kali ini replikasi sudah dapat diuji dan berjalan

Lampiran 7 : Surat Izin Penelitian Penelitian Skripsi di Pustikom



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS NEGERI JAKARTA
 Kampus Universitas Negeri Jakarta, Jalan Rawamangun Muka, Jakarta 13220
 Telp./Fax. : Rektor : (021) 4893854, PR I : 4895130, PR II : 4893918, PR III : 4892926, PR IV : 4893982
 BAUK : 4750930, BAAK : 4759081, BAPSI : 4752180
 Bag. UHTP : Telp. 4893726, Bag. Keuangan : 4892414, Bag. Kepegawaian : 4890536, HUMAS : 4898486
 Laman : www.unj.ac.id

Nomor : 1429/UN39.12/KM/2015

30 Maret 2015

Lamp. : -

Hal : Permohonan Izin Mengadakan Penelitian
untuk Penulisan Skripsi

Yth. Kepala PUSTIKOM
Universitas Negeri Jakarta

Kami mohon kesediaan Saudara untuk dapat menerima Mahasiswa Universitas Negeri Jakarta :


Nama : Nugroho Saputra
 Nomor Registrasi : 5235117081
 Program Studi : Pendidikan Teknik Informatika dan Komputer
 Fakultas : Teknik Universitas Negeri Jakarta
 No. Telp/HP : 08588828825

Dengan ini kami mohon diberikan ijin mahasiswa tersebut, untuk dapat mengadakan penelitian guna mendapatkan data yang diperlukan dalam rangka Penulisan Skripsi. Skripsi tersebut dengan judul :

"Analisis Perancangan dan Implementasi Purwarupa Data Recovery Center Server Untuk Menjaga Ketersediaan Jaringan Pada Server di Universitas Negeri Jakarta"

Atas perhatian dan kerjasama Saudara, kami sampaikan terima kasih.

Kepala Biro Administrasi
Akademik dan Kemahasiswaan,



Dr. Syaifullah
NIP. 195702161984031001

Tembusan :

1. Dekan Fakultas Teknik
2. Kaprog / Jurusan Teknik Elektro

Lampiran 8 : Borang Peserta Pengujian kelompok kecil replikasi

FORM ISIAN UJI DATA SKIPSI

Nama : Nurmuhammad Septia Putra

IP : 192.168.32.171

UJI 1 ISI BIODATA

ISI 5 data bersamaan	ISI 10 Data		ISI 15 data		
0210100*	0220100	0220600	0230100	0230600	0231100
0210200	0220200	0220700	0230200	0230700	0231200
0210300	0220300	0220800	0230300	0230800	0231300
0210400	0220400	0220900	0230400	0230900	0231400
0210500	0220500	0221000	0230500	0231000	0231500

*02 di isi No urut dan di isikan di filed no identitas

UJI 2 UPLOAD FILE

UKURAN FILE	JUMLAH*
1kb	4
1MB	?
5 MB	2
10 MB	2
15 MB	2

*banyaknya file yang di input

UJI 3 STRESS TEST

UJI KE	JUMLAH data Stress Test
1	10 ✓
2	50 ✓
3	100 ✓
4	1000 ✓
5	4242 ✓ (<5000)*

*angka unik dan acak yang jumlahnya kurang dari 5000

JAKARTA, 5-11-2015

Nurmuhammad S. Putra
 Nurmuhammad S. Putra
 Nama jelas

FORM ISIAN UJI DATA SKIPSI

Nama : SHERLI Aulia UlfahIP : 192.168.32.185

UJI 1 ISI BIODATA

ISI 5 data bersamaan	ISI 10 Data		ISI 15 data		
<u>0510100*</u>	<u>0520100</u>	<u>0520600</u>	<u>0530100</u>	<u>0530600</u>	<u>0531100</u>
<u>0510200</u>	<u>0520200</u>	<u>0520700</u>	<u>0530200</u>	<u>0530700</u>	<u>0531200</u>
<u>0510300</u>	<u>0520300</u>	<u>0520800</u>	<u>0530300</u>	<u>0530800</u>	<u>0531300</u>
<u>0510400</u>	<u>0520400</u>	<u>0520900</u>	<u>0530400</u>	<u>0530900</u>	<u>0531400</u>
<u>0510500</u>	<u>0520500</u>	<u>0521000</u>	<u>0530500</u>	<u>0531000</u>	<u>0531500</u>

*__ di isi No urut dan di isikan di filed no identitas

UJI 2 UPLOAD FILE

UKURAN FILE	JUMLAH*
1kb	2
1MB	2
5 MB	2
10 MB	2
15 MB	2

*banyaknya file yang di input

UJI 3 STRESS TEST

UJI KE	JUMLAH data Stress Test
1	10 ✓
2	50 ✓
3	100 ✓
4	1000 ✓
5	3426 ✓ (<5000)*

*angka unik dan acak yang jumlahnya kurang dari 5000

JAKARTA, 05 Nov 2015


 Nama jelas

FORM ISIAN UJI DATA SKIPSI

Nama : Made Upadana

IP : 192.168.32.170

UJI 1 ISI BIODATA

ISI 5 data bersamaan	ISI 10 Data		ISI 15 data		
0410100*	0420100	0420600	0430100	0430600	0431100
0410200	0420200	0420700	0430200	0430700	0431200
0410300	0420300	0420800	0430300	0430800	0431300
0410400	0420400	0420900	0430400	0430900	0431400
0410500	0420500	0421000	0430500	0431000	0431500

* _ _ di isi No urut dan di isikan di filed no identitas

UJI 2 UPLOAD FILE

UKURAN FILE	JUMLAH*
1kb	5
1MB	5
5 MB	5
10 MB	5
15 MB	5
	3


*banyaknya file yang di input

UJI 3 STRESS TEST

UJI KE	JUMLAH data Stress Test
1	10
2	50
3	100
4	1000
5	4000 (<5000)*

*angka unik dan acak yang jumlahnya kurang dari 5000

JAKARTA, 5-11-2015


 Nama jelas Made Upadana

Lampiran 9 : Borang Peserta Pengujian kelompok besar replikasi

FORM ISIAN UJI DATA SKIPSI

Nama : RANDI SATRIA

IP : 192.168.32.134

UJI 1 ISI BIODATA

ISI 5 data bersamaan	ISI 10 Data		ISI 15 data		
0210100*	0220100	0220600	0230100	0230600	0231100
0210200	0220200	0220700	0230200	0230700	0231200
0210300	0220300	0220800	0230300	0230800	0231300
0210400	0220400	0220900	0230400	0230900	0231400
0210500	0220500	0221000	0230500	0231000	0231500

*__ di isi No urut dan di isikan di filed no identitas

UJI 2 UPLOAD FILE

UKURAN FILE	JUMLAH*
1kb	5
1MB	2
5 MB	1
10 MB	2
15 MB	2

*banyaknya file yang di input

UJI 3 STRESS TEST

UJI KE	JUMLAH data Stress Test
1	10
2	50
3	100
4	1000
5	4646 (<5000)*

*angka unik dan acak yang jumlahnya kurang dari 5000

JAKARTA, 10 - NOV - 2015


 RANDI SATRIA
 Nama jelas

FORM ISIAN UJI DATA SKIPSI

Nama : Nida Aulia

IP : 192.168.32.137

UJI 1 ISI BIODATA

ISI 5 data bersamaan	ISI 10 Data		ISI 15 data		
0310100*	0320100	0320600	0330100	0330600	0331100
0310200	0320200	0320700	0330200	0330700	0331200
0310300	0320300	0320800	0330300	0330800	0331300
0310400	0320400	0320900	0330400	0330900	0331400
0310500	0320500	0321000	0330500	0331000	0331500

*__ di isi No urut dan di isikan di filed no identitas

UJI 2 UPLOAD FILE

UKURAN FILE	JUMLAH*
1kb	5
1MB	4
5 MB	3
10 MB	2
15 MB	2

*banyaknya file yang di input

UJI 3 STRESS TEST

UJI KE	JUMLAH data Stress Test
1	10
2	50
3	100
4	1000
5	2332 (<5000)*

*angka unik dan acak yang jumlahnya kurang dari 5000

JAKARTA, 10 November 2015



Nama jelas
NIDA AULIA H.

FORM ISIAN UJI DATA SKIPSI

Nama : Aprianto Budiman

IP : 192.168.32.150

UJI 1 ISI BIODATA

ISI 5 data bersamaan	ISI 10 Data		ISI 15 data		
0110100*	0120100	0120600	0130100	0130600	0131100
0110200	0120200	0120700	0130200	0130700	0131200
0110300	0120300	0120800	0130300	0130800	0131300
0110400	0120400	0120900	0130400	0130900	0131400
0110500	0120500	0121000	0130500	0131000	0131500

*_ di isi No urut dan di isikan di filed no identitas

UJI 2 UPLOAD FILE

UKURAN FILE	JUMLAH*
1kb	4
1MB	3
5 MB	2
10 MB	2
15 MB	2

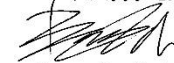
*banyaknya file yang di input

UJI 3 STRESS TEST

UJI KE	JUMLAH data Stress Test
1	10
2	50
3	100
4	1000
5	4646 (<5000)*

*angka unik dan acak yang jumlahnya kurang dari 5000

JAKARTA, 10 Nov 2015



Aprianto Budiman

Nama jelas

FORM ISIAN UJI DATA SKIPSI

Nama : Tony Risdiantoro

IP : 192.168.32.143

UJI 1 ISI BIODATA

ISI 5 data bersamaan	ISI 10 Data		ISI 15 data		
0010100*	0020100	0020600	0030100	0030600	0031100
0010200	0020200	0020700	0030200	0030700	0031200
0010300	0020300	0020800	0030300	0030800	0031300
0010400	0020400	0020900	0030400	0030900	0031400
0010500	0020500	0021000	0030500	0031000	0031500

* __ di isi No urut dan di isikan di filed no identitas

UJI 2 UPLOAD FILE

UKURAN FILE	JUMLAH*
1kb	2
1MB	2
5 MB	2
10 MB	2
15 MB	2

*banyaknya file yang di input

UJI 3 STRESS TEST

UJI KE	JUMLAH data Stress Test
1	10
2	50
3	100
4	1000
5	4500 (<5000)*

*angka unik dan acak yang jumlahnya kurang dari 5000

JAKARTA, 10 November 2015


Nama jelas
Tony Risdiantoro

FORM ISIAN UJI DATA SKIPSI

Nama : Siti Rosniah

IP : 102.168.32.104

UJI 1 ISI BIODATA

ISI 5 data bersamaan	ISI 10 Data		ISI 15 data		
0510100*	0520100	0520600	0530100	0530600	0531100
0510200	0520200	0520700	0530200	0530700	0531200
0510300	0520300	0520800	0530300	0530800	0531300
0510400	0520400	0520900	0530400	0530900	0531400
0510500	0520500	0521000	0530500	0531000	0531500

* __ di isi No urut dan di isikan di filed no identitas

UJI 2 UPLOAD FILE

UKURAN FILE	JUMLAH*
1kb	3
1MB	3
5 MB	2
10 MB	2
15 MB	2

*banyaknya file yang di input

UJI 3 STRESS TEST

UJI KE	JUMLAH data Stress Test
1	10
2	50
3	100
4	1000
5	3000 (<5000)*

*angka unik dan acak yang jumlahnya kurang dari 5000

JAKARTA, 10 Desember 2015


 Siti Rosniah
 Nama jelas

Lampiran 10 : Tampilan Website Pengujian

Analisis dan Perancangan Purwarupa Sistem Disaster Recovery Server Database Mysql Di Universitas Negeri Jakarta

[Home](#)
[Lihat Data](#)
[Lihat File](#)
[Tambah Data](#)
[Tambah File](#)
[Stress Test](#)

Selamat datang!
Web Pengujian Replikasi Database

SELALU ADA JALAN UNTUK ORANG YANG MENCOBAL!

Nugroho @2015

antar muka halaman utama *website* penguji

Analisis dan Perancangan Purwarupa Sistem Disaster Recovery Server Database Mysql Di Universitas Negeri Jakarta

[Home](#)
[Lihat Data](#)
[Lihat File](#)
[Tambah Data](#)
[Tambah File](#)
[Stress Test](#)

Cari Disini.

#	Nama Lengkap	Jenis Identitas	Nomor Identitas	Warga Negara	Jenis Kelamin	Tanggal Lahir	Alamat Tinggal	Negara	Provinsi	Kabupaten	Nomor Telpnon	Nomor HP	Alamat Email	waktu Input	Actions
1	Uta Moe Moe	KTP	02	Warga Negara Indonesia	Pria	1993-09-20	Tokyo	Naturalan	DKI	Tokyo Pusat	02142802539	+6289685953604	nummuhammad.sp@gmail.com	2015-11-05 15:41:57.727147	C D
2	Abdul Hafiz Adnin	SIM	0310100	Warga Negara Indonesia	Pria	1994-01-19	Dubai	Australia	Jawa Barat	IRAN	089899899899	089899899899	rajaempat94@gmail.com	2015-11-05 15:41:57.806231	C D
3	Sherli Aulia Uffa	KTP	0510100	Warga Negara Indonesia	Wanita	1994-01-28	Pondok Kopi	Indonesia	Jakarta	-	09995252	08929272373	sherli.aulia@yahoo.com	2015-11-05 15:41:57.961147	C D
4	Zaeri Khoiruddin	KTP	0110100	Warga Negara Indonesia	Pria	1994-08-18	Kranggan Pemai, Bekasi	Indonesia	Jawa Barat	Kota Bekasi	8440805	085790447974	zaeri@mail.com	2015-11-05 15:41:58.070185	C D
5	I Made Upadana	KTP	0410100	Warga Negara Indonesia	Pria	1993-09-18	Jl Kusuma Timur	Indonesia	Jawa Barat	Bekasi Timur	021881xxxxx	0857821xxxxxx	imxxxxxxxxxxx	2015-11-05 15:41:58.365837	C D
6	Uta Moe Moe	KTP	02	Warga Negara Indonesia	Pria	1993-09-20	Tokyo	Naturalan	DKI	Tokyo Pusat	02142802539	+6289685953604	nummuhammad.sp@gmail.com	2015-11-05 15:42:18.100398	C D
7	I Made Upadana	KTP	0410200	Warga Negara Indonesia	Pria	1993-09-18	Jl Kusuma Timur	Indonesia	Jawa Barat	Bekasi Timur	021881xxxxx	0857821xxxxxx	imxxxxxxxxxxx	2015-11-05 15:42:18.193702	C D
8	Sherli Aulia Uffa	KTP	0510200	Warga Negara Indonesia	Wanita	1994-01-28	Pondok Kopi	Indonesia	Jakarta	-	09995252	08929272373	sherli.aulia@yahoo.com	2015-11-05 15:42:18.365222	C D
9	Zaeri Khoiruddin	KTP	0110200	Warga Negara Indonesia	Pria	1994-08-18	Kranggan Pemai, Bekasi	Indonesia	Jawa Barat	Kota Bekasi	8440805	085790447974	zaeri@mail.com	2015-11-05 15:42:18.385011	C D
10	Abdul Hafiz Adnin	SIM	0310200	Warga Negara Indonesia	Pria	1994-01-19	Dubai	Australia	Jawa Barat	IRAN	089899899899	089899899899	rajaempat94@gmail.com	2015-11-05 15:42:18.396443	C D

Banyaknya data sekarang = 33204

SELALU ADA JALAN UNTUK ORANG YANG MENCOBAL!

Nugroho @2015

antar muka laman lihat data pada *website* penguji

Analisis dan Perancangan Purwarupa Sistem Disaster Recovery Server Database Mysql Di Universitas Negeri Jakarta

[Home](#)
[Lihat Data](#)
[Lihat File](#)
[+ Tambah Data](#)
[Tambah File](#)
[Stress Test](#)

Name	Type File file	Ukuran(bytes)	Waktu input	
file 1kb	application/octet-stream	452	2015-11-05 13:46:44.942696	Download
file 1mb.pdf	application/pdf	1236340	2015-11-05 13:46:51.482774	Download
file 5mb.exe	application/x-msdownload	5311104	2015-11-05 13:47:19.000977	Download
file 10mb.zip	application/octet-stream	10281741	2015-11-05 13:47:42.725878	Download
file 15mb.rar	application/octet-stream	17324437	2015-11-05 13:48:01.216693	Download
021	application/octet-stream	452	2015-11-05 16:52:31.230315	Download
0411kb	application/octet-stream	452	2015-11-05 16:52:31.541799	Download
0511kb	application/octet-stream	452	2015-11-05 16:52:33.117664	Download
0311kb	application/octet-stream	452	2015-11-05 16:52:33.414165	Download
0321kb	application/octet-stream	452	2015-11-05 16:52:37.017816	Download

[1](#)
[2](#)
[3](#)
[4](#)
[Next](#)
[Last](#)

Banyaknya data sekarang = 96

SELALU ADA JALAN UNTUK ORANG YANG MENCOBA!

Nugroho @2015

antar muka laman lihat *file* pada *website* penguji

Analisis dan Perancangan Purwarupa Sistem Disaster Recovery Server Database Mysql Di Universitas Negeri Jakarta

[Home](#)
[Lihat Data](#)
[Lihat File](#)
[+ Tambah Data](#)
[Tambah File](#)
[Stress Test](#)

Nama Lengkap	:	<input type="text"/>	Negara	:	<input type="text"/>
Jenis Identitas	:	<input type="text" value="KTP"/>	Provinsi	:	<input type="text"/>
Nomor Identitas	:	<input type="text"/>	Kabupaten	:	<input type="text"/>
Warga Negara	:	<input type="text" value="Warga Negara Indonesia"/>	Nomor Telpon	:	<input type="text"/>
Jenis Kelamin	:	<input checked="" type="radio"/> Pria <input type="radio"/> Wanita	Nomor HP	:	<input type="text"/>
Tanggal Lahir	:	<input type="text" value="mm/dd/yyyy"/>	Alamat Email	:	<input type="text"/>
Alamat Tinggal	:	<input type="text"/>			

[+ Tambah Data](#)

SELALU ADA JALAN UNTUK ORANG YANG MENCOBA!

Nugroho @2015

antar muka laman tambah data pada *website* penguji

Analisis dan Perancangan Purwarupa Sistem Disaster Recovery Server Database Mysql Di Universitas Negeri Jakarta

Home Lihat Data Lihat File + Tambah Data Tambah File Stress Test

Upload File

Choose File | No file chosen

+ Tambah Data

SELALU ADA JALAN UNTUK ORANG YANG MENCOBA!

Nugroho @2015

antar muka laman tambah *file* pada *website* penguji

Analisis dan Perancangan Purwarupa Sistem Disaster Recovery Server Database Mysql Di Universitas Negeri Jakarta

Home Lihat Data Lihat File + Tambah Data Tambah File Stress Test

GUI STRESS TEST

Masukan Banyak nya Data

MULAI!

Banyaknya data sekarang = 33204

SELALU ADA JALAN UNTUK ORANG YANG MENCOBA!

Nugroho @2015

antar muka laman *stress test* pada *website* penguji

Analisis dan Perancangan Purwarupa Sistem Disaster Recovery Server Database Mysql Di Universitas Negeri Jakarta

[Home](#)
[Lihat Data](#)
[Lihat File](#)
[+ Tambah Data](#)
[Tambah File](#)
[Stress Test](#)

Nama Lengkap : Negara :
 Jenis Identitas : Provinsi :
 Nomor Identitas : Kabupaten :
 Warga Negara : Nomor Telpn :
 Jenis Kelamin : Nomor HP :
 Tanggal Lahir : Alamat Email :
 Alamat Tinggal :

[UPDATE DATA](#)
[CANCEL](#)

SELALU ADA JALAN UNTUK ORANG YANG MENCOBA!

Nugroho @2015

antar muka laman ubah biodata pada *website* penguji

Analisis dan Perancangan Purwarupa Sistem Disaster Recovery Server Database Mysql Di Universitas Negeri Jakarta

[Home](#)
[Lihat Data](#)
[Lihat File](#)
[+ Tambah Data](#)
[Tambah File](#)
[Stress Test](#)

Sure ! to remove the following record ?

#	Nama Lengkap	Jenis Identitas	Nomor Identitas	Warga Negara	Jenis Kelamin	Tanggal Lahir	Alamat Tinggal	Negara	Provinsi	Kabupaten	Nomor Telpn	Nomor HP	Alamat Email
1	Uta Moe Moe	KTP	02	Warga Negara Indonesia	Pria	1993-09-20	Tokyo	Naturalian	DKI Tokyo	Tokyo Pusat	02142802539	+6289685953604	numhammad.sp@gmail.com

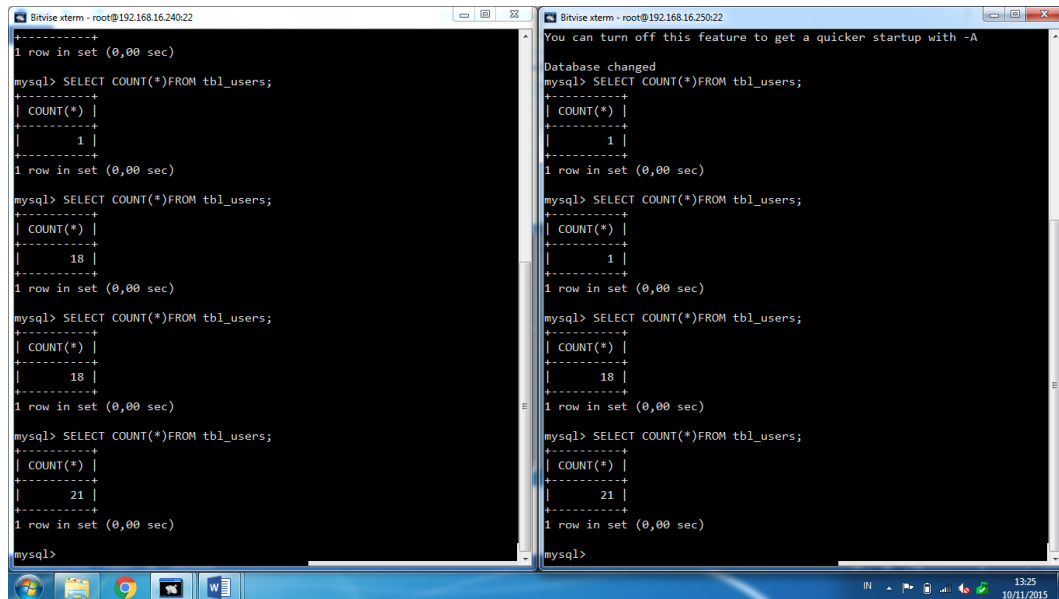
[YES](#)
[NO](#)

SELALU ADA JALAN UNTUK ORANG YANG MENCOBA!

Nugroho @2015

antar muka laman hapus data pada *website* penguji

Lampiran 11 : Pengecekan Kecocokan data Antara Server Master dan Slave



The image shows two terminal windows side-by-side, both running MySQL. The left window is connected to a server at 192.168.16.240:22, and the right window is connected to a server at 192.168.16.250:22. Both windows show the same sequence of queries and results for the table 'tbl_users'.

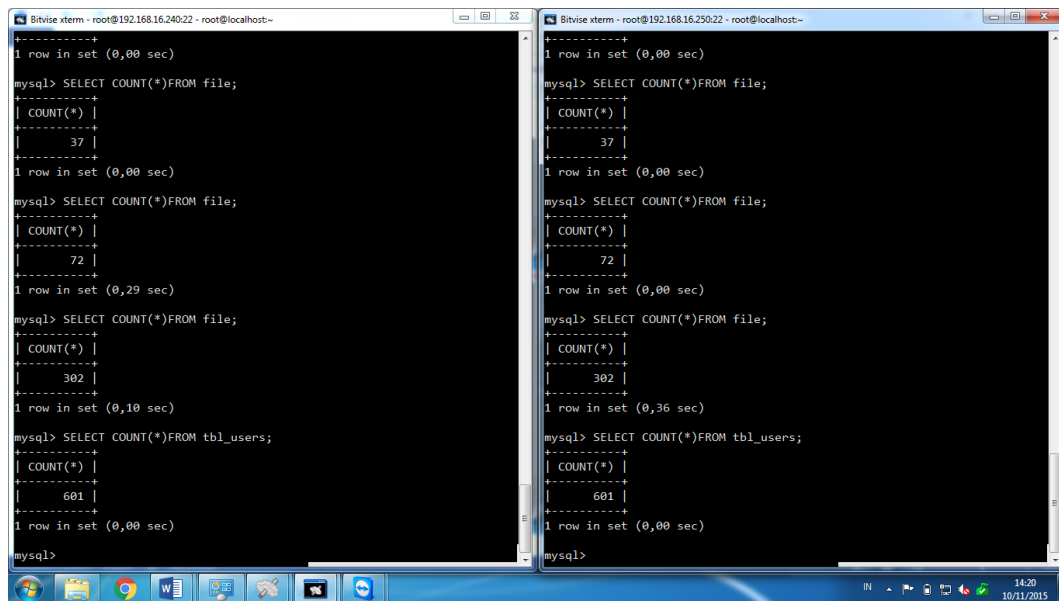
```
mysql> SELECT COUNT(*)FROM tbl_users;
+-----+
| COUNT(*) |
+-----+
|         1 |
+-----+
1 row in set (0,00 sec)

mysql> SELECT COUNT(*)FROM tbl_users;
+-----+
| COUNT(*) |
+-----+
|        18 |
+-----+
1 row in set (0,00 sec)

mysql> SELECT COUNT(*)FROM tbl_users;
+-----+
| COUNT(*) |
+-----+
|        18 |
+-----+
1 row in set (0,00 sec)

mysql> SELECT COUNT(*)FROM tbl_users;
+-----+
| COUNT(*) |
+-----+
|        21 |
+-----+
1 row in set (0,00 sec)

mysql>
```



The image shows two terminal windows side-by-side, both running MySQL. The left window is connected to a server at 192.168.16.240:22, and the right window is connected to a server at 192.168.16.250:22. Both windows show the same sequence of queries and results for the table 'file' and 'tbl_users'.

```
mysql> SELECT COUNT(*)FROM file;
+-----+
| COUNT(*) |
+-----+
|        37 |
+-----+
1 row in set (0,00 sec)

mysql> SELECT COUNT(*)FROM file;
+-----+
| COUNT(*) |
+-----+
|        72 |
+-----+
1 row in set (0,29 sec)

mysql> SELECT COUNT(*)FROM file;
+-----+
| COUNT(*) |
+-----+
|       302 |
+-----+
1 row in set (0,10 sec)

mysql> SELECT COUNT(*)FROM tbl_users;
+-----+
| COUNT(*) |
+-----+
|       601 |
+-----+
1 row in set (0,00 sec)

mysql>
```

```

Bitvise xterm - root@192.168.16.240:22 - root@localhost~
-----+-----+
1 row in set (0,29 sec)
mysql> SELECT COUNT(*)FROM file;
-----+-----+
| COUNT(*) |
-----+-----+
|      302 |
-----+-----+
1 row in set (0,10 sec)
mysql> SELECT COUNT(*)FROM tbl_users;
-----+-----+
| COUNT(*) |
-----+-----+
|      601 |
-----+-----+
1 row in set (0,00 sec)
mysql> SELECT COUNT(*)FROM tbl_users;
-----+-----+
| COUNT(*) |
-----+-----+
|      801 |
-----+-----+
1 row in set (0,03 sec)
mysql> SELECT COUNT(*)FROM tbl_users;
-----+-----+
| COUNT(*) |
-----+-----+
|     1801 |
-----+-----+
1 row in set (0,01 sec)
mysql>

Bitvise xterm - root@192.168.16.250:22 - root@localhost~
-----+-----+
1 row in set (0,00 sec)
mysql> SELECT COUNT(*)FROM file;
-----+-----+
| COUNT(*) |
-----+-----+
|      302 |
-----+-----+
1 row in set (0,36 sec)
mysql> SELECT COUNT(*)FROM tbl_users;
-----+-----+
| COUNT(*) |
-----+-----+
|      601 |
-----+-----+
1 row in set (0,00 sec)
mysql> SELECT COUNT(*)FROM tbl_users;
-----+-----+
| COUNT(*) |
-----+-----+
|      801 |
-----+-----+
1 row in set (0,05 sec)
mysql> SELECT COUNT(*)FROM tbl_users;
-----+-----+
| COUNT(*) |
-----+-----+
|     1801 |
-----+-----+
1 row in set (0,00 sec)
mysql>

```

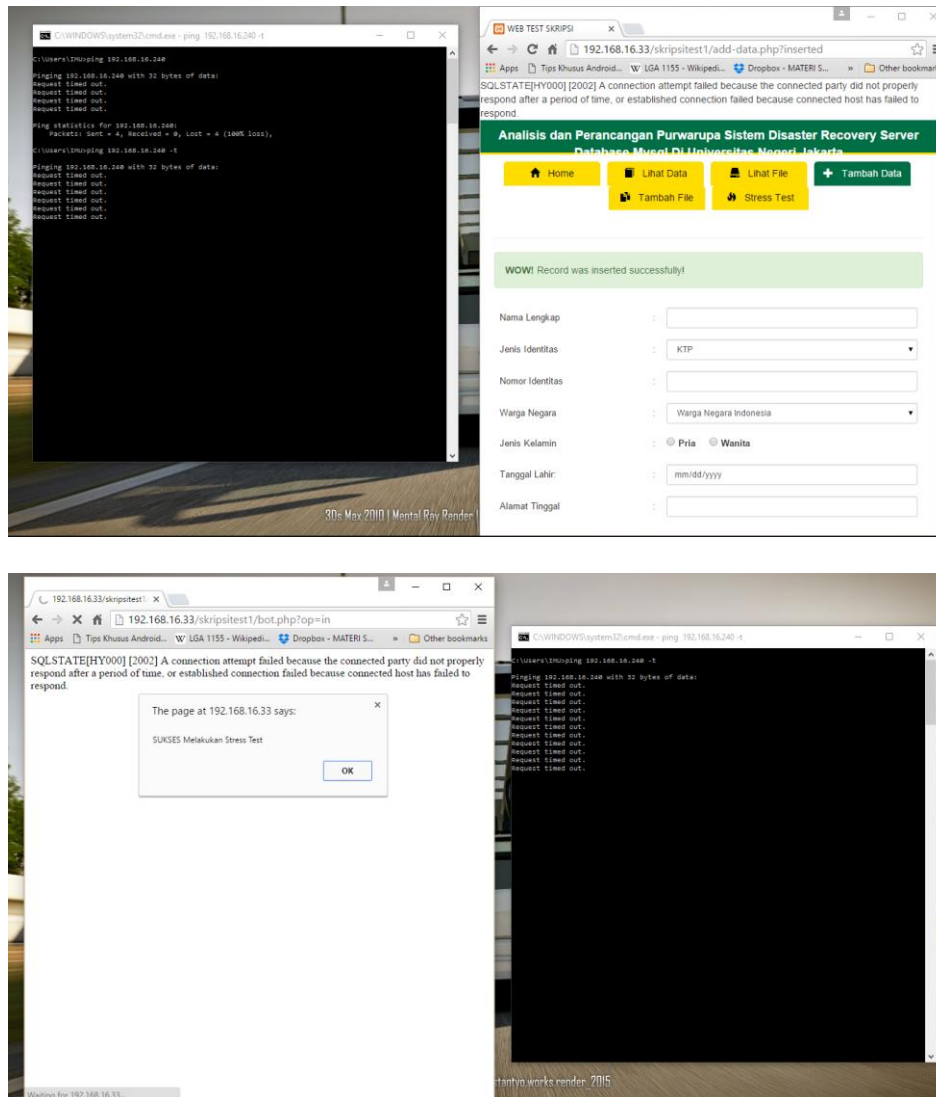
```

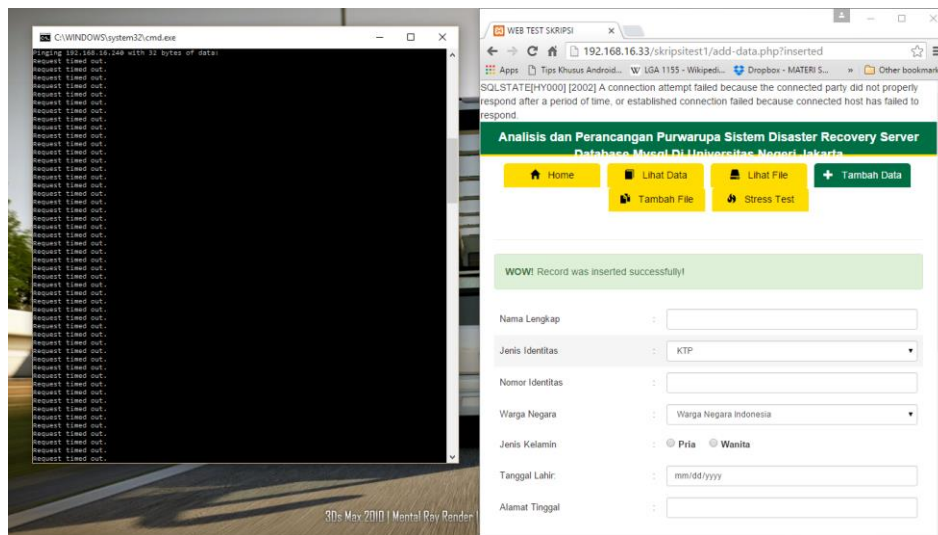
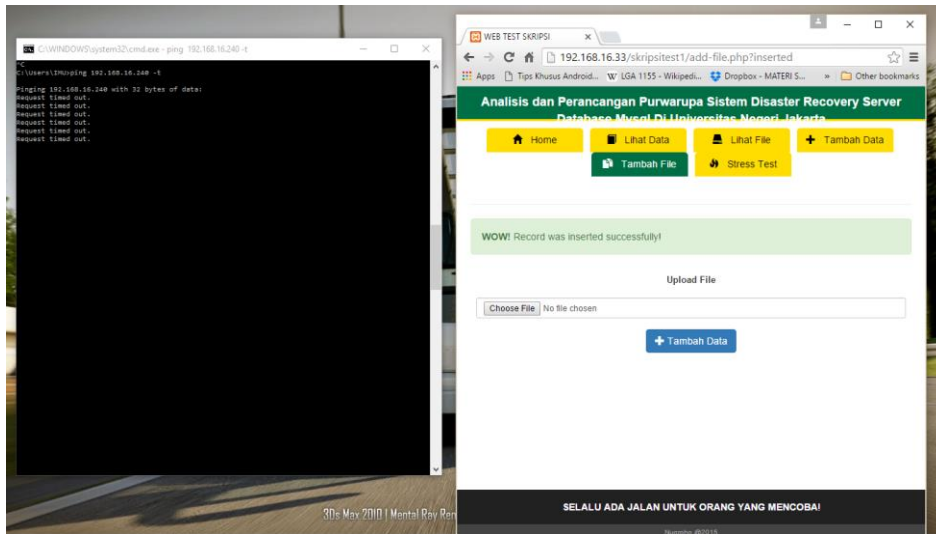
Bitvise xterm - root@192.168.16.240:22 - root@localhost~
-----+-----+
1 row in set (0,00 sec)
mysql> SELECT COUNT(*)FROM tbl_users;
-----+-----+
| COUNT(*) |
-----+-----+
|       41 |
-----+-----+
1 row in set (0,00 sec)
mysql> SELECT COUNT(*)FROM tbl_users;
-----+-----+
| COUNT(*) |
-----+-----+
|       61 |
-----+-----+
1 row in set (0,00 sec)
mysql> SELECT COUNT(*)FROM tbl_users;
-----+-----+
| COUNT(*) |
-----+-----+
|       81 |
-----+-----+
1 row in set (0,00 sec)
mysql> SELECT COUNT(*)FROM tbl_users;
-----+-----+
| COUNT(*) |
-----+-----+
|      101 |
-----+-----+
1 row in set (0,00 sec)
mysql>

Bitvise xterm - root@192.168.16.250:22 - root@localhost~
-----+-----+
1 row in set (0,00 sec)
mysql> SELECT COUNT(*)FROM tbl_users;
-----+-----+
| COUNT(*) |
-----+-----+
|       41 |
-----+-----+
1 row in set (0,00 sec)
mysql> SELECT COUNT(*)FROM tbl_users;
-----+-----+
| COUNT(*) |
-----+-----+
|       61 |
-----+-----+
1 row in set (0,00 sec)
mysql> SELECT COUNT(*)FROM tbl_users;
-----+-----+
| COUNT(*) |
-----+-----+
|       81 |
-----+-----+
1 row in set (0,00 sec)
mysql> SELECT COUNT(*)FROM tbl_users;
-----+-----+
| COUNT(*) |
-----+-----+
|      101 |
-----+-----+
1 row in set (0,00 sec)
mysql>

```


Lampiran 12 : Bukti Ping dan input data saat Pengujian Recovery server





TENTANG PENULIS



Nugroho Saputra Lahir pada tanggal 26 Mei 1993 di Jakarta. Riwayat pendidikan yang telah ditempuh oleh penulis adalah sebagai berikut: pada tahun 1998-2004 menempuh jenjang pendidikan sekolah dasar di SD Negeri Rawamangun 07 Pagi Jakarta Timur dan pada tahun 2004-2007 menempuh jenjang sekolah menengah pertama di SMP Negeri 74 Jakarta Timur serta tahun 2007-2010 menempuh jenjang sekolah menengah atas di SMA Negeri 30 Jakarta.

Saat masuk sekolah menengah atas penulis sudah teratik pada bidang Komputer. Perkenalan penulis dengan teman-teman pada lingkungan perkuliahan semakin membuat penulis untuk mendalami bidang *Network Engineering*. Untuk memenuhi sarat kelulusan penulis membuat skripsi dan melakukan penelitian dengan berjudul “Analisis, Perancangan Dan Pembuatan Purwarupa Disaster Recovery Database Server Di Universitas Negeri Jakarta” Penulis dapat dihubungi melalui e-mail dengan alamat nugrohosaputra26@gmail.com. Terima kasih kepada para pembaca yang telah bersedia membaca karya akhir ini. Semoga bermanfaat.