

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN  
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
PALCOMTECH PALEMBANG**

**SKRIPSI**

**DESAIN DAN OPTIMALISASI *WEB SERVER* BERBASIS  
*OPEN SOURCE* PADA KOPERTIS WILAYAH II  
PALEMBANG**



**Oleh :**

**FEBRIYANDA DWITA PUTRI GUMAI  
NPM.011090265**

**Untuk Memenuhi Sebagian dari Syarat-Syarat  
Guna Menyelesaikan Program Strata Satu**

**PALEMBANG  
2012**

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN  
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
PALCOMTECH PALEMBANG**

---

**HALAMAN PERSETUJUAN DOSEN PEMBIMBING**

**NAMA** : Febriyanda Dwita Putri Gumai  
**NOMOR POKOK** : 011090265  
**PROGRAM STUDI** : Teknik Informatika  
**JENJANG PENDIDIKAN** : Strata 1 (S1)  
**KONSENTRASI** : Jaringan  
**JUDUL LAPORAN** : *Desain dan Optimalisasi Web Server Berbasis  
Open Source pada Kopertis Wilayah II  
Palembang.*

**Palembang, September 2012**

**Mengetahui,  
Ketua STMIK,**

**Menyetujui,  
Pembimbing Skripsi,**

**Rudi Sutomo, S.Kom., M.Si  
NIP :028.PCT.08**

**Molavi Arman, S.Kom  
NIDN : 0205058003**

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN  
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
PALCOMTECHPALEMBANG**

---

**HALAMAN PENGESAHAN PENGUJI**

**NAMA** : Febriyanda Dwita Putri Gumai  
**NOMOR POKOK** : 011090265  
**PROGRAM STUDI** : Teknik Informatika  
**JENJANG PENDIDIKAN** : Strata 1 (S1)  
**KONSENTRASI** : Jaringan  
**JUDUL LAPORAN** : *Desain dan Optimalisasi Web Server Berbasis  
Open Source pada Kopertis Wilayah II  
Palembang.*

**Tanggal** : September 2012      **Tanggal** : September 2012  
**Penguji1** :                              **Penguji2** :

**Atin Triwahyuni, S.T., M.Eng**  
**NIDN. 0215028002**

**Rudi Sutomo, S.Kom., M.Si**  
**NIP. 028.PCT.08**

**Menyetujui,  
Ketua STMIK,**

**Rudi Sutomo, S.Kom., M.Si**  
**NIP. 028.PCT.08**

## ABSTRAK

Febriyanda Dwita Putri Gumai (011.090.265) : Desain dan Optimalisasi *Web Server* Berbasis *Open Source* pada Kopertis Wilayah II Palembang. Dibawah bimbingan Bapak Molavi Arman, S.Kom.

Perkembangan teknologi semakin cepat dan dibutuhkan, untuk bisa menangani permintaan atau *request* pada *website*, maka dibutuhkan *web server* yang handal, keberadaan *web server* sangat dibutuhkan karena *web server* akan menunggu permintaan dari *client* yang menggunakan *browser* seperti *Internet Explorer*, *Modzilla*, dan program *browser* lainnya. Jika ada permintaan dari *browser*, maka *web server* akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan, dibandingkan menggunakan jasa *hosting* pihak lain karena tidak optimal dalam menangani permintaan akses dari *client*.

Pada Penelitian ini dibangun sebuah teknologi *web server* pada jaringan komputer Kopertis Wilayah II Palembang yang disimulasikan pada sebuah *server* menggunakan sistem operasi *Debian Squeeze*, dengan sistem LAMPP (Linux Debian 6, *Apache*, *Mysql*, *PHP*), yang dikombinasikan dengan drupal, AWStats. Diharapkan agar teknologi ini dapat diterapkan pada instansi tersebut dan dapat membantu kelancaran permintaan atau *request* pada *website* tersebut.

Keyword: *web server*, *Debian Squeeze*, *Apache*, *Mysql*, *PHP*, drupal, AWStats.

## Motto

- *Mother, I love you for all that you do.*
- *Knowledge and skills are tools, the workman is character.*
- *Real power does not hit hard, but straight to the point.*
- *A healthy man has a hundred wishes, a sick man has only one.*
- *Smile is the shortest distance between two people.*
- *True love is love which only for two person, and no place for the third person.*
- *Man jadda wajada "Siapa yang bersungguh-sungguh, akan berhasil".*

## Kupersembahkan kepada:

- **Kedua orang tua tercinta atas do'a dan motivasinya**
- **Kakak dan adik tersayang, karena keluarga adalah segalanya.**
- **Teman-Teman seperjuangan yang telah banyak membantu dan memberi dukungan.**

## KATA PENGANTAR

Puji dan syukur Penulis panjatkan ke hadirat Allah SWT, atas segala berkat dan karunia-Nya sehingga Penulis dapat menyelesaikan laporan skripsi ini dengan baik. Dengan judul **“Desain dan Optimalisasi Web Server Berbasis Open Source pada Kopertis Wilayah II Palembang”**. Adapun tujuan penulisan Skripsi ini untuk memenuhi sebagian dari syarat-syarat guna menyelesaikan program strata satu.

Adapun selama penulisan dan penyusunan laporan Skripsi ini, Penulis mendapatkan banyak bimbingan, bantuan dan dukungan dari berbagai pihak. Oleh karena itu, sudah menjadi kewajiban bagi Penulis untuk mengucapkan terima kasih kepada berbagai pihak tersebut, yaitu kepada Ketua STMIK PalComTech, Bapak Rudi Sutomo, S.Kom., M.si, kepada Ketua Program Studi Teknik Informatika, Bapak D. Tri Octafian, S.kom.,M.Kom, kepada Dosen Pembimbing Skripsi, Bapak Molavi Arman, S.Kom, kepada staff Kopertis Wilayah II Palembang yang telah membantu, kepada kedua orang tua yang tercinta yang telah membesarkan, merawat dan mendidik serta senantiasa mendo'akan Penulis, kakak dan adik tersayang, kepada sahabat yang selalu ada disaat suka dan duka dan kepada teman – teman seperjuangan yang telah banyak membantu dan memberi dukungan.

Demikian kata pengantar dari Penulis, dengan harapan semoga laporan Skripsi ini dapat bermanfaat dan berguna bagi para pembaca, dengan kesadaran Penulis bahwa skripsi masih mempunyai banyak kekurangan dan kelemahan.

Sehingga membutuhkan banyak saran dan kritik yang membangun untuk menghasilkan sesuatu yang lebih baik. Terima kasih.

**Palembang, September 2012**

**Penulis**

## DAFTAR ISI

<b>HALAMAN JUDUL.....</b>	<b>i</b>
<b>HALAMAN PERSETUJUAN DOSEN PEMBIMBING.....</b>	<b>ii</b>
<b>HALAMAN PENGESAHAN PENGUJI .....</b>	<b>iii</b>
<b>ABSTRAK .....</b>	<b>iv</b>
<b>MOTTO DAN PERSEMBAHAN.....</b>	<b>v</b>
<b>KATA PENGANTAR.....</b>	<b>vi</b>
<b>DAFTAR ISI.....</b>	<b>vii</b>
<b>DAFTAR GAMBAR.....</b>	<b>xii</b>
<b>DAFTAR TABEL .....</b>	<b>xvii</b>
<b>BAB I PENDAHULUAN</b>	
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	5
1.3. Batasan Masalah.....	5
1.4. Tujuan .....	5
1.5. Manfaat .....	6
1.6. SistematikaPenulisan .....	7
<b>BAB II GAMBARAN UMUM KOPERTIS WILAYAH II PALEMBANG</b>	
2.1. <i>Profil</i> Kopertis.....	9
2.1.1. Sejarah Kopertis .....	9
2.1.2. Visi dan Misi Kopertis .....	11
2.2. Struktur Organisasi .....	12
2.2.1. Struktur Organisasi Kopertis Wilayah II Palembang.....	12
2.2.2. Tugas, Wewenang dan Tanggung Jawab .....	14
<b>BAB III TINJAUAN PUSTAKA</b>	
3.1. Teori Pendukung .....	20



3.1.1. Komunikasi Data .....	20
3.1.2. <i>Internet</i> .....	21
3.1.2.1 Sejarah <i>Internet</i> .....	22
3.1.3. Jaringan Komputer .....	23
3.1.4. Terminologi Jaringan .....	28
3.1.5. Topologi Jarkom .....	37
3.1.6. <i>Web Server</i> .....	43
3.1.6.1 Cara Kerja <i>Web Server</i> .....	43
3.1.7. <i>Content Management System</i> .....	44
3.1.7.1 Macam – macam <i>CMS</i> .....	45
3.1.7.2 Komponen Penunjang <i>Drupal</i> .....	48
3.1.8. <i>Domain Name System</i> .....	51
3.1.9. <i>Apache Benchmark</i> .....	54
3.1.10. <i>Apache Httpperf</i> .....	55
3.1.11. Sejarah <i>Linux</i> .....	56
3.1.11.1. <i>Debian Linux</i> .....	57
3.2. Hasil Penelitian Terdahulu .....	58

## **BAB IV METODE PENELITIAN**

4.1. Lokasi dan Waktu Penelitian .....	61
4.1.1. Lokasi penelitian .....	61
4.1.2. Waktu Penelitian .....	61
4.2. Jenis Data .....	62
4.2.1. Data Primer .....	62
4.2.2. Data Sekunder .....	63
4.3. Teknik Pengumpulan Data .....	63
4.4. Jenis Penelitian .....	64
4.5. Alur dan Teknik Pengembangan Sistem .....	65
4.5.1. Alur Proses .....	65
4.5.2. Teknik Pengembangan Sistem .....	67

## **BAB V HASIL DAN PEMBAHASAN**

5.1. Analisis .....	74
5.2. <i>Desain</i> .....	75
5.3. <i>Simulation Prototype</i> .....	77
5.4. <i>Implementation</i> .....	79
5.4.1. Konfigurasi DNS .....	80
5.4.2. Penginstallan <i>Apache Mysql</i> .....	85
5.4.3. Penginstallan <i>Drupal</i> .....	89
5.4.4. <i>Apache Benchmark</i> .....	93
5.4.5. <i>Apache Httpperf</i> .....	96
5.4.6. <i>Optimalisasi Web Server</i> .....	98
5.4.6.1. <i>PHP Xcache</i> .....	98
5.4.6.2. Hasil <i>AB</i> dan <i>Httpperf</i> Setelah <i>Optimalisasi</i> .....	99
5.4.8. <i>Konfigurasi Komputer Client</i> .....	105
5.5. <i>Monitoring</i> .....	106
5.6. <i>Manajemen</i> .....	109

## **BAB VI SIMPULAN DAN SARAN**

6.1. <i>Simpulan</i> .....	111
6.2. <i>Saran</i> .....	112

<b>DAFTAR PUSTAKA</b> .....	<b>xv</b>
-----------------------------	-----------

<b>HALAMAN LAMPIRAN</b> .....	<b>xvii</b>
-------------------------------	-------------

## DAFTAR GAMBAR

Gambar 1.1.	Grafik Mengakses <i>Website</i> Kopertis .....	4
Gambar 2.1.	Bagan Struktur Organisasi .....	13
Gambar 3.1.	Model Komunikasi Sederhana .....	20
Gambar 3.2.	<i>Local Area Network</i> .....	24
Gambar 3.3.	Model Jaringan <i>MAN</i> .....	25
Gambar 3.4.	Model Jaringan <i>WAN</i> .....	26
Gambar 3.5.	Lapisan OSI.....	30
Gambar 3.6.	<i>Point to point</i> .....	38
Gambar 3.7.	Topologi <i>BUS</i> .....	39
Gambar 3.8.	Topologi <i>Star</i> .....	40
Gambar 3.9.	Topologi <i>Ring</i> .....	41
Gambar 3.10.	Topologi <i>Tree</i> .....	42
Gambar 3.11.	Cara Kerja <i>Web Server</i> .....	44
Gambar 3.12.	Arsitektur Hirarki DNS .....	52
Gambar 4.1.	<i>Flowchart</i> Model Alur Proses .....	66
Gambar 4.2.	Tahapan Pada <i>NDLC</i> .....	67
Gambar 5.1.	Topologi Jaringan Kopertis Wilayah II Palembang .....	75
Gambar 5.2.	Topologi yang Dikembangkan .....	76
Gambar 5.3.	<i>Prosesor Intel Pentium 4</i> .....	77
Gambar 5.4.	Memori 512 Mb.....	78
Gambar 5.5.	<i>Hardisk Sata 80 Gb</i> .....	78

Gambar 5.6.	Kabel LAN .....	78
Gambar 5.7.	Switch D-link dan Modem Aztech.....	79
Gambar 5.8.	Mengisi <i>File Local</i> .....	81
Gambar 5.9.	Mengisi <i>File Zone Domain</i> .....	82
Gambar 5.10.	Mengisi <i>Reverse Address</i> .....	83
Gambar 5.11.	Mengisi <i>IP Address</i> .....	83
Gambar 5.12.	Mengedit <i>IP</i> pada <i>Name Server</i> .....	84
Gambar 5.13.	Me- <i>restart</i> Kartu Jaringan .....	84
Gambar 5.14.	<i>Test Resolver DNS</i> .....	84
Gambar 5.15.	<i>Virtual Hosting</i> .....	86
Gambar 5.16.	<i>Install Mysql</i> .....	86
Gambar 5.17.	<i>Install Mysql_Secure_Installation</i> .....	87
Gambar 5.18.	Membuat <i>Database</i> .....	87
Gambar 5.19.	Mengisi <i>Database</i> .....	87
Gambar 5.20.	Mengisi <i>PHP5-Suhosin</i> .....	88
Gambar 5.21.	Modifikasi <i>Konfigurasi</i> .....	88
Gambar 5.22.	Meng- <i>install</i> Paket <i>PHP5-Mysql</i> .....	89
Gambar 5.23.	Memilih Profil di <i>Drupal</i> .....	90
Gambar 5.24.	Memilih Bahasa yang Akan Digunakan.....	90
Gambar 5.25.	<i>Verify Requirements</i> .....	91
Gambar 5.26.	<i>Setup Database</i> .....	92
Gambar 5.27.	<i>Install Profile</i> .....	93
Gambar 5.28.	Persentase <i>Benchmark 50 Request</i> Sebelum <i>Optimalisasi</i> ....	94

Gambar 5.29.	Persentase <i>Benchmark 100 Request</i> Sebelum Optimalisasi ..	94
Gambar 5.30.	Persentase <i>Httpperf 50 Request</i> Sebelum Optimalisasi.....	96
Gambar 5.31.	Persentase <i>Httpperf 100 Request</i> Sebelum Optimalisasi.....	100
Gambar 5.32.	Persentase <i>Benchmark 50 Request</i> Setelah Optimalisasi .....	100
Gambar 5.33.	Persentase <i>Benchmark 100 Request</i> Setelah Optimalisasi ....	102
Gambar 5.34.	Persentase <i>Httpperf 50 Request</i> Setelah Optimalisasi .....	103
Gambar 5.35.	Persentase <i>Httpperf 100 Request</i> Setelah Optimalisasi .....	100
Gambar 5.36.	Grafik Sebelum dan Setelah Optimalisasi .....	104
Gambar 5.37.	<i>Setting IP Address</i> .....	105
Gambar 5.38.	Tampilah Drupal.....	107
Gambar 5.39.	Mengubah Akses <i>Awstat</i> .....	107
Gambar 5.40.	Tampilan <i>Awstat</i> .....	108
Gambar 5.41.	<i>Update Web Server</i> .....	109
Gambar 5.42.	<i>Upgrade Web Server</i> .....	110

## DAFTAR TABEL

Tabel 3.1. Hasil Penelitian Terdahulu Pertama.....	58
Tabel 3.2. Hasil Penelitian Terdahulu Kedua .....	59
Tabel 4.1. Waktu Penelitian .....	61
Tabel 5.1. <i>Benchmark 50 Request</i> dan <i>100 Request</i> Sebelum Optimalisasi ....	95
Tabel 5.2. <i>Httpperf 50 Request</i> dan <i>100 Request</i> Sebelum Optimalisasi.....	98
Tabel 5.3. <i>Benchmark 50 Request</i> dan <i>100 Request</i> Setelah Optimalisasi .....	101
Tabel 5.4. <i>Httpperf 50 Request</i> dan <i>100 Request</i> Setelah Optimalisasi .....	103

## DAFTAR PUSTAKA

- Kuncoro, Mudrajat. 2009. *Metode Riset Untuk Bisnis dan Ekonomi*. Jakarta : Erlangga.
- Sofana, Iwan. 2008. *Membangun Jaringan Komputer*. Bandung : Informatika.
- Sopandi, Dede. 2008. *Instalasi dan Konfigurasi Jaringan Komputer*. Bandung: Informatika.
- Simarmata, Janner. 2010. *Rekayasa Web*. Yogyakarta: Andi.
- Sugeng, Winarno. 2010. *Jaringan Komputer Dengan TCP/IP*. Bandung: Modula.
- Sutanta, Edhy. 2005. *Komunikasi Data dan Jaringan Komputer*. Yogyakarta: Graha Ilmu.
- Utomo, Eko Priyo. 2006. *Buku Pengantar Jaringan Komputer Bagi Pemula*. Bandung : Yrama Widya.
- Yani, Ahmad dan Fanny Hfiary. 2005. *Mudah Murah Membangun Jaringan Internet dengan Linux Router Project Leaf Bearing*. Jakarta : Elex Media Komputindo.
- Wahana Komputer. 2010. *MySQL Database Server*. Jakarta :Mediakita.
- Wahana Komputer. 2009. *PHP Programming*. Semarang : Andi.
- ([http:// apache.org/docs/2.0/programs](http://apache.org/docs/2.0/programs), diakses 27 Juni 2012).
- ([http:// eprints.unsri.ac.id](http://eprints.unsri.ac.id), diakses 20 Mei 2012).
- ([http:// estiko.info/php-accelerator/](http://estiko.info/php-accelerator/), diakses 20 Juni 2012).
- ([http:// hpl.hp.com/research /linux/httpperf](http://hpl.hp.com/research/linux/httpperf) diakses 20 Juni 2012).

(<http://ilmu-linux.com/category/PHP>, diakses 20 Juni 2012).

(<http://lup.lub.lu.se/luur>, diakses 20 Mei 2012).

(<http://opensource-indonesia.com>, diakses pada tanggal 20 Mei 2012).

(<http://www.kopertis2.or.id/v2>).



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Semakin berkembangnya ilmu pengetahuan dan teknologi diberbagai bidang serta informasi yang berkembang dengan sangat cepat dan luas, dimana akses terhadap informasi dapat diakses tanpa dibatasi ruang dan waktu, hal ini tentu saja tidak dapat lepas dari adanya *internet* sebagai sumber informasi.

Kebutuhan akan pengetahuan media *internet* tidak sebatas dalam dunia usaha saja, tetapi sudah merambah memasuki dunia pendidikan, baik itu sekolah maupun perguruan tinggi. Perguruan tinggi sebagai suatu lembaga pendidikan yang mencetak kader-kader pembangunan bangsa dituntut dapat menyesuaikan dengan perubahan-perubahan yang terjadi, inilah tantangan bagi perguruan tinggi untuk dapat menciptakan mahasiswa yang berdaya saing tinggi, berfikir kedepan serta mengenal dan mampu mengatasi ketinggalan akan ilmu pengetahuan dan teknologi.

Perkembangan pendidikan saat ini sangat besar, yang membutuhkan wadah untuk mendapatkan ilmu pengetahuan dan teknologi yang bermanfaat. Perguruan tinggi salah satu tempat untuk melanjutkan jenjang pendidikan yang lebih tinggi dan mencetak sumber daya manusia yang

handal, berdaya saing tinggi, lebih matang dalam ilmu pengetahuan dan teknologi. Perguruan tinggi yang baik dan terakreditasi harus diawasi, diperhatikan dan dibina.

Koordinasi Perguruan Tinggi Swasta (Kopertis) adalah instansi pemerintah di lingkungan Departemen Pendidikan Nasional. Kopertis merupakan perpanjangan tangan Direktorat Jenderal Pendidikan Tinggi Kementerian Pendidikan dan Kebudayaan di suatu wilayah tertentu guna melaksanakan pengawasan, pengendalian, dan pembinaan terhadap perguruan tinggi swasta (PTS) di wilayah tersebut. Perguruan Tinggi Swasta di wilayah bersangkutan, hingga saat ini ada 12 (dua belas) Kopertis Wilayah di seluruh Indonesia. Kopertis Wilayah II Palembang merupakan salah satu dari 12 Kopertis wilayah yang ada di Indonesia. Kopertis wilayah II membawahi semua PTS di 4 Provinsi, yaitu Sumatera Selatan, Bengkulu, Lampung, dan Bangka Belitung. Jumlah PTS yang ada di Sumatera selatan yaitu 103 PTS, sedangkan di Bengkulu terdapat 16 PTS, di Lampung terdapat 71 PTS dan di Bangka Belitung terdapat 14 PTS, jadi jumlah keseluruhan PTS di 4 Provinsi tersebut yaitu 204 PTS.

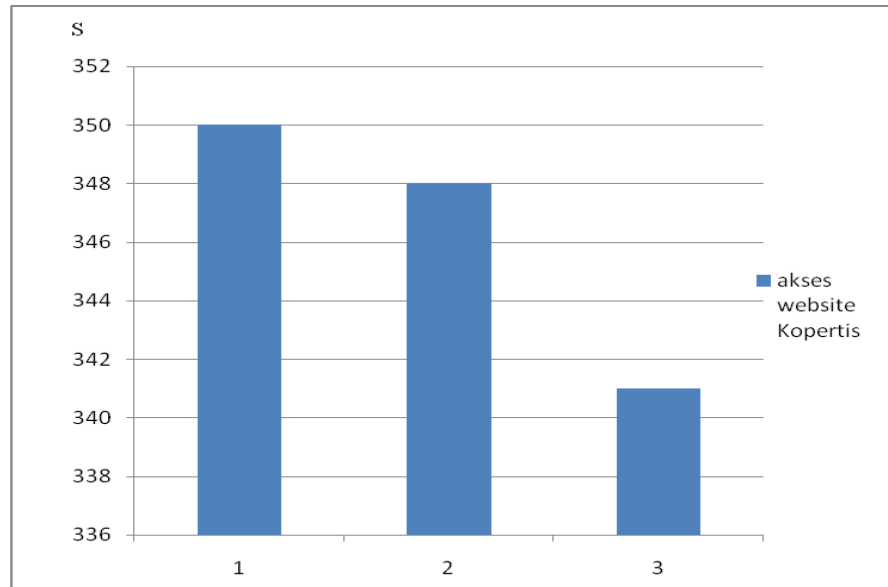
Pada Kopertis Wilayah II Palembang *Web server* sangat dibutuhkan karena *web server* adalah *software* yang menjadi tulang belakang dari *world wide web* (www), *web server* akan menunggu permintaan dari *client* yang menggunakan *browser* seperti *Internet Explorer*, *Modzilla*, dan program *browser* lainnya. Jika ada permintaan dari *browser*, maka *web server* akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data

yang diinginkan. Untuk menunjang kegiatan operasional berupa penyajian data yang terdapat di *website* Kopertis Wilayah II Palembang, maka diperlukan kajian kinerja *web server*, supaya informasi yang dibutuhkan *client* bisa berjalan dengan lancar, dapat sebagai media komunikasi dan *client* dapat mengakses *website* Kopertis untuk mengetahui profil dan sumber data yang terdapat di Kopertis.

Fungsi *web server* dari Kopertis Wilayah II Palembang sangat besar, untuk bisa menangani permintaan atau *request* pada *website* Kopertis Wilayah II Palembang, karena PTS yang dinaungi Kopertis Wilayah II Palembang berjumlah 204 PTS, maka dibutuhkan *web server* yang handal. Penulis telah melakukan penelitian pada Kopertis Wilayah II Palembang dengan fokus pada *web server*, di Kopertis Wilayah II Palembang belum terdapat *web server*, *website* yang terdapat di Kopertis Wilayah II Palembang menggunakan jasa *hosting* pihak lain dan tidak optimal dalam menangani permintaan akses dari *client*.

Masalah yang sering dihadapi adalah saat *client* melakukan akses secara bersamaan waktu *loading* yang diperlukan menjadi lebih lama sehingga *client* harus menunggu sampai beberapa menit untuk bisa melakukan akses ke halaman *website*, hal tersebut tentu saja membuat *client* yang sedang melakukan akses merasa tidak nyaman, karena akses *internet* yang lama, *client* yang dimaksud disini diantaranya mahasiswa, dosen, pegawai Kopertis.

Berikut grafik mengakses *website* Kopertis Wilayah II Palembang:



**Gambar 1.1** Grafik Mengakses *Website* Kopertis

Grafik diatas dihasilkan dari tiga *sample* data mengakses *website* Kopertis Wilayah II Palembang, dengan alamat [www.kopertis2.or.id](http://www.kopertis2.or.id) yang diambil pada tanggal 14 september 2012-16 september 2012 pada pukul 12.00 wib. Pada percobaan pertama dengan menggunakan *wifi* kecepatan akses 350 *second*, pada percobaan kedua dengan menggunakan modem kecepatan akses 348 *second*, dan pada percobaan ketiga dengan menggunakan *wifi* kecepatan akses 341 *second*, jadi lama akses *website* tersebut cukup lama.

Oleh karena itu untuk mengatasi masalah tersebut diperlukan sebuah *web server* dengan performa yang handal sehingga masalah tersebut dapat diminimalisasi, dan menggunakan sistem operasi linux yang berbasis *open source* guna menekan biaya pengeluaran yang sangat mahal.

Oleh sebab itu dalam penulisan Skripsi ini penulis tertarik untuk mengambil judul “**Desain dan Optimalisasi *Web Server* Berbasis *Open Source* pada Kopertis Wilayah II Palembang**”.

## **1.2 Rumusan Masalah**

Berdasarkan uraian latar belakang dan judul yang diambil penulis, maka penulis merumuskan masalah yang ada yaitu “Bagaimana desain dan mengoptimalkan performa *web server* berbasis *open source* pada Kopertis Wilayah II Palembang ?”.

## **1.3 Batasan Masalah**

Batasan masalah yang akan di bahas oleh penulis pada penulisan skripsi ini yaitu cara optimalisasi *web server* dengan menggunakan sistem operasi Debian 6.0 dengan *web server Apache* di Kopertis Wilayah II Palembang.

## **1.4 Tujuan Penelitian**

Tujuan yang ingin dicapai dalam penulisan skripsi ini berdasarkan latar belakang dan perumusan masalah di atas pada Kopertis Wilayah II Palembang adalah:

1. Untuk menerapkan ilmu yang didapat selama perkuliahan di Sekolah Tinggi Manajemen Informatika dan Komputer Palcomtech Palembang

2. Untuk mengoptimalkan performa atau kinerja dari *webserver* berbasis *open source* pada Kopertis Wilayah II Palembang.

### **1.5. Manfaat Penelitian**

Adapun manfaat yang ingin dicapai dalam penulisan skripsi ini berdasarkan latar belakang dan perumusan masalah di atas pada Kopertis Wilayah II Palembang adalah:

#### **1. Bagi Penulis**

1. Dapat menerapkan ilmu yang didapat selama perkuliahan di Sekolah Tinggi Manajemen Informatika dan Komputer Palcomtech Palembang.
2. Dapat melakukan optimalisasi *web server* berbasis *open source* pada Kopertis Wilayah II Palembang.

#### **2. Bagi Kopertis wilayah II Palembang**

1. Sebagai pilihan alternatif pada Kopertis Wilayah II Palembang dalam melakukan optimalisasi *web server* menggunakan *open source* agar berjalan dengan lebih baik.
2. Dapat meningkatkan pelayanan dan memberikan solusi dalam mengatasi kendala – kendala dalam pengaturan *webserver* terutama optimalisasi *webserver* di Kopertis Wilayah II Palembang.

#### **3. Bagi Akademik**

Sebagai bahan referensi dan perbandingan bagi penulisan yang akan datang agar dapat membuat penelitian dan pengembangan yang lebih

baik lagi mengenai permasalahan bagaimana konfigurasi *web server* berbasis *open source*.

## **1.6. Sistematika Penulisan**

Sistematika dalam penulisan skripsi ini terdiri dari 6 (enam) bab, pembuatan sistematika ini dimaksud untuk memberi gambaran isi laporan ini, sehingga antara bab yang satu dengan yang lainnya saling berkaitan. Adapun sistematika penulisan ini adalah sebagai berikut:

### **BAB I PENDAHULUAN**

Bab ini merupakan pendahuluan secara umum memberikan gambaran tentang hal-hal yang melatarbelakangi laporan ini dilaksanakan, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, sampai sistematika penulisan laporan.

### **BAB II GAMBARAN UMUM**

Bab II merupakan gambaran umum mengenai profil, visi dan misi dan struktur organisasi Kopertis Wilayah II Palembang.

### **BAB III TINJAUAN PUSTAKA**

Bab III merupakan teori – teori yang mendukung yang terkait dengan penelitian, hasil penelitian terdahulu, kerangka pemikiran.

#### **BAB IV METODE PENELITIAN**

Bab IV merupakan metode penelitian yang berisikan lokasi dan waktu penelitian, jenis data, teknik pengumpulan data, jenis penelitian, alat dan teknik pengembangan sistem, serta alat dan teknik pengujian.

#### **BAB V HASIL DAN PEMBAHASAN**

Bab V merupakan hasil dan pembahasan proses optimalisasi *web server* berbasis *open source* pada Kopertis Wilayah II Palembang.

#### **BAB VI SIMPULAN DAN SARAN**

Bab VI merupakan bab terakhir dari penulisan laporan yang berisi tentang simpulan dari uraian bab-bab sebelumnya dan saran yang sekiranya dapat bermanfaat.





## **BAB II**

### **GAMBARAN UMUM**

#### **2.1 Profil Kopertis**

##### **2.1.1 Sejarah Kopertis**

Kopertis akronim dari koordinasi perguruan tinggi swasta adalah instansi pemerintah dilingkungan departemen pendidikan nasional. Kopertis merupakan perpanjangan tangan Direktorat Jenderal Pendidikan Tinggi Kementrian Pendidikan dan Kebudayaan disuatu wilayah tertentu. Guna melaksanakan pengawasan, pengendalian, dan pembinaan terhadap perguruan tinggi swasta (PTS) di wilayah tersebut. Dengan memperhatikan luas wilayah dan jumlah PTS di wilayah bersangkutan hingga saat ini ada 12 (dua belas) kopertis wilayah di seluruh Indonesia. Kopertis wilayah II Palembang merupakan salah satu dari 12 kopertis wilayah di Indonesia.

Kopertis wilayah II Palembang yang berkantor di Jalan Srijaya no.883 Palembang, didirikan pada tahun 1982, pada saat itu wilayah kerja kopertis wilayah II Palembang meliputi 5 (lima) Provinsi, yaitu : Provinsi Sumatera selatan, Provinsi Lampung, Provinsi Bengkulu, Provinsi Jambi, dan Provinsi Kalimantan barat. Koordinator Kopertis Wilayah II pada saat itu adalah Drs. H. Syafran Syamsuddin (periode

1982-1986). Sebelumnya, Kopertis Wilayah II merupakan bagian Kopertis Wilayah Jakarta dengan Koordinator Girindro Pringgogidgo (periode 1976-1982).

Tahun 1986 Prof. Ir. H. Bochari Rachman, M.Sc. diangkat sebagai Koordinator Kopertis Wilayah II Palembang (periode 1986-1995). Pada masa beliau menjadi Koordinator terjadi pengurangan wilayah kerja. Pada tahun 1990 berdasarkan Surat Keputusan Menteri Pendidikan dan Kebudayaan No. 0135/O/1990 dan No. 19/O/1990 Kopertis Wilayah II Palembang yang semula meliputi 5 (lima) Provinsi dikurangi hanya meliputi 3 (tiga) provinsi yaitu Provinsi Sumatera Selatan, Provinsi Lampung dan Provinsi Bengkulu. Provinsi Jambi menjadi bagian dari Kopertis Wilayah X Padang, sedangkan Provinsi Kalimantan Barat menjadi bagian Kopertis Wilayah XI Banjarmasin.

Pejabat Koordinator berikutnya Dr. Ir. H. Saad Nasuhim (periode 1995-1998). Pada tahun 1998 Prof. Dr. Ir. H. Zainal Ridho Djafar diangkat sebagai Koordinator Kopertis Wilayah II (periode 1998-1999), Pada tahun 2001 terjadi pemekaran wilayah di mana Bangka Belitung yang sebelumnya berstatus Kabupaten berubah menjadi Provinsi Kepulauan Bangka Belitung. Dengan demikian wilayah kerja Kopertis Wilayah II Palembang meliputi 4 (empat) Provinsi, yaitu: Provinsi Sumatera Selatan, Provinsi Lampung, Provinsi Bengkulu dan Provinsi Kepulauan Bangka Belitung.

Kemudian jabatan Koordinator Kopertis Wilayah II selanjutnya Dr. Ir. H. Ali Yasmin Adam Wiralaga (periode 1999-2002), dan jabatan koordinator kopertis wilayah II Sejak bulan Mei 2002 sampai dengan Juli 2004 Koordinator Kopertis Wilayah II Palembang dijabat oleh Dr. H. Djamaah Sopah, M.Sc., Ed. Selanjutnya periode 2004-2008 Koordinator Kopertis Wilayah II dijabat oleh Prof. Chuzaimah Dahlan Diem, M.L.S., Ed.D. Sejak tanggal 7 November 2008 sampai dengan sekarang Koordinator Kopertis Wilayah II dijabat oleh Prof. Dr. Diah Natalisa, M.B.A.

## **2.1.2 Visi dan Misi Kopertis**

### **2.1.2.1 Visi Kopertis**

Menjadikan Kopertis Wilayah II sebagai lembaga pembina, pengendali dan pengawas yang handal serta profesional terhadap pengembangan perguruan tinggi swasta di wilayah kerjanya.

### **2.1.2.2 Misi Kopertis**

Peningkatan kualitas usaha pelayanan kepada masyarakat khususnya kepada masyarakat perguruan tinggi swasta, secara cepat, tepat dan akurat.

## 2.2. Struktur Organisasi

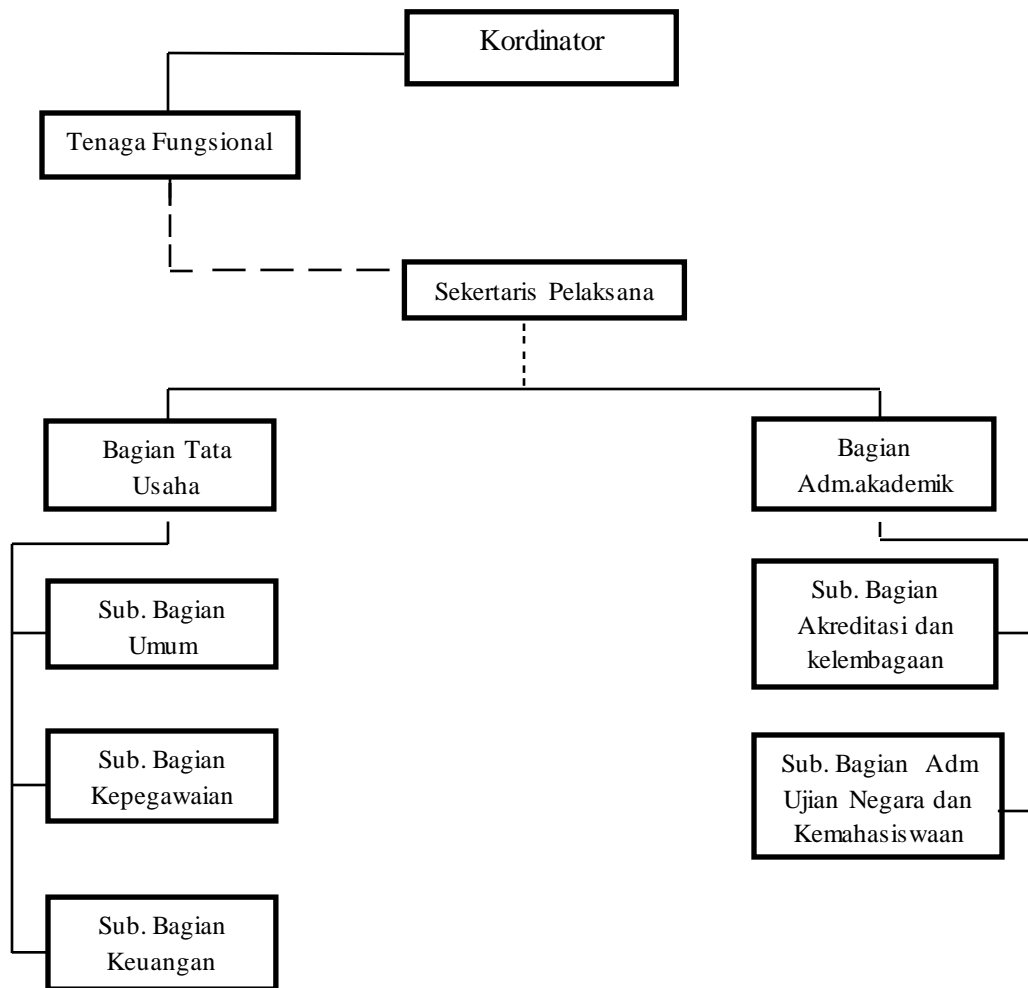
Organisasi akan dapat mencapai tujuannya apabila orang didalamnya melakukan kerjasama yang baik, dimana kerjasama hanya dapat terjadi apabila hubungan diantara orang yang satu dengan yang lainnya dapat berjalan sebagaimana mestinya.

Hubungan orang yang satu dengan yang lainnya akan semakin lancar ditentukan oleh struktur organisasi sekolah yang jelas, serta wewenang dan tanggung jawab masing-masing bagian. Setiap anggota organisasi akan mengetahui dari mana ia menerima perintah dan kepada siapa ia harus melapor. Dengan demikian berarti pula bahwa struktur organisasi merupakan bentuk komunikasi antara anggota-anggota organisasi itu.

Perkembangan semakin maju dan semakin besar suatu sekolah, maka persoalan mengenai organisasi semakin banyak dan rumit. Organisasi dibutuhkan karena adanya suatu kerjasama yang baik dalam melaksanakan aktivitas sekolah.

Suatu organisasi untuk mencapai target yang diinginkan perlu membentuk suatu badan organisasi yang disesuaikan dengan tugas dan tanggung jawab. Hal ini bertujuan agar tindakan yang dilakukan serta cara pengaturannya memang berdasarkan atas kewajiban karena tugas yang dibebankan oleh seorang pimpinan kepada bawahan. Struktur organisasi ini sangat erat hubungannya dengan tindakan yang dilakukan.

Berikut adalah struktur organisasi pada Kopertis Wilayah II Palembang.



Keterangan :

- Garis Komando
- - - - - Garis Pelayanan Administrasi
- - - - - Garis Tanggung Jawab Sehari – hari

**Gambar 2.1 Bagan Struktur Organisasi**

Sumber : Kopertis Wilayah II Palembang, 2012

### **2.3. Tugas Wewenang**

Berikut ini adalah pembagian tugas dari struktur organisasi pada Kopertis Wilayah II Palembang

#### **a. Koordinator**

Keputusan Menteri Pendidikan dan Kebudayaan Republik Indonesia Nomor 0135/0/1990 Pasal 4 tentang susunan organisasi:

1. Koordinator Kopertis dalam melaksanakan tugasnya dibantu oleh Sekertariat Pelaksana Kopertis, dan Kelompok Tenaga Fungsional.

#### **b. Sekertaris Pelaksana**

Keputusan Menteri Pendidikan dan Kebudayaan Republik Indonesia Nomor 0300/0/1992 Pasal 1 tentang perincian tugas Sekertariat Pelaksana Koordinasi Perguruan Tinggi Swasta :

1. Menyusun rencana dan program kerja tahunan Sekertariat Pelaksana, serta mempersiapkan penyusunan program kerja tahunan Koordinator Kopertis.
2. Mengumpulkan, mengolah dan mengadministrasikan dan informasi tentang evaluasi, akreditasi, kelembagaan, ujian negara, kemahasiswaan serta melaksanakan system informasi, publikasi, dan kerjasama.
3. Mengumpulkan, mengolah dan mengadministrasikan hasil kegiatan bimbingan penyelenggaraan Tridarma Perguruan Tinggi.

4. Mengumpulkan, mengolah, dan mengadministrasikan bahan pertimbangan dalam rangka pemberian bantuan.
5. Melaksanakan administrasi penyelenggaraan kegiatan Tridarma Perguruan Tinggi berdasarkan petunjuk dari Direktorat Jenderal Pendidikan Tinggi dan Koordinator Kopertis.
6. Menyiapkan konsep usulan pemberian bantuan kepada perguruan tinggi swasta di wilayah kerjanya.
7. Menyusun rencana dan mengatur penyelenggaraan ujian negara bagi mahasiswa perguruan tinggi swasta di wilayah kerjanya.
8. Melaksanakan pengumpulan data hasil akreditasi dan evaluasi terhadap perguruan tinggi swasta di wilayah kerjanya.
9. Mengumpulkan data kegiatan teknis pembinaan perguruan tinggi swasta di wilayah kerjanya.
10. Melaksanakan pembinaan penyelenggaraan urusan kepegawaian, keuangan, dan urusan umum di lingkungan kopertis.
11. Memberikan pelayanan teknis di bidang pembinaan dan pengembangan pengelolaan perguruan tinggi swasta di wilayah kerjanya.
12. Melaksanakan tugas lainnya sesuai dengan petunjuk pimpinan.



13. Menyusun laporan pelaksanaan program kerja Sekretariat Pelaksana dan mempersiapkan laporan pelaksanaan program kerja Koordinator Kopertis.

### **c. Bagian Tata Usaha**

Tugas Pokok berdasarkan Kepmendikbud No.0300/O/1992, antara lain:

1. Menyusun rencana dan program kerja tahunan Bagian serta mempersiapkan penyusunan program kerja tahunan Sekretariat Pelaksana.
2. Melaksanakan urusan kepegawaian yang meliputi perencanaan kebutuhan/formasi, pemutasian, pemensiunan, dan pengembangan pegawai di lingkungan Kopertis.
3. Melaksanakan urusan keuangan yang meliputi perencanaan, pemanfaatan/pembelanjaan, pembukuan pertanggungjawaban anggaran rutin dan pembangunan di lingkungan Kopertis.
4. Melaksanakan urusan surat menyurat yang meliputi surat masuk, surat keluar, ekspedisi, kearsipan, dan dokumentasi.
5. Melaksanakan urusan rumah tangga kantor yang meliputi kebersihan, keamanan, ketertiban, dan keindahan di lingkungan Kopertis.
6. Melaksanakan pengurusan alat tulis kantor di lingkungan kantor Kopertis.
7. Melaksanakan pengaturan perizinan tamu dan keprotokolan

8. Melaksanakan pengaturan dan penyediaan fasilitas rapat dinas dan upacara
9. Mempersiapkan pemberian penerangan.informasi yang menyangkut pelaksanaan tugas Kopertis
- 10.Melaksanakan pelayanan teknis ketatausahaan kepada bagian.
- 11.Melaksanakan pengurusan perpustakaan Kopertis.
- 12.Menyusun laporan pelaksanaan program kerja Bagian dan menyusun laporan pelaksanaan program kerja Sekretariat Pelaksana.

Bagian Tata Usaha Terdiri dari:

1. Sub Bagian Umum
2. Sub Bagian Kepegawaian
3. Sub Bagian Keuangan

#### **d. Bagian Akademik**

Tugas Pokok sejak diberlakukannya SK MENDIKNAS Nomor 184/U/2001 Tanggal 23 Nopember 2001.

1. Menghimpun serta mengolah Data Isian PTS dalam rangka Laporan Proses Belajar mengajar setiap akhir semester sesuai SK. Dikti Nomor 034/DIKTI/Kep/2002 tanggal 2 April 2002, untuk kemudian Data yang telah diolah tersebut diteruskan ke Dirjen Dikti.

2. Melaksanakan Evaluasi Akademik Penyelenggaraan Program studi bagi PTS yang mengajukan perpanjangan ijin yang status program studinya sudah kadaluwarsa dari Dirjen Dikti serta memproses Rekomendasinya ke Dirjen Dikti sesuai dengan SK Dikti Nomor 3492/D/T/2001 tanggal 9 November 2001.
3. Memproses rekomendasi bagi PTS yang mengajukan perpanjangan Ijin Penyelenggaraan berdasarkan SK. Dirjen Dikti No. 108/DIKTI/Kep/2001.
4. Menghimpun data PTS untuk pembuatan buku Direktori.
5. Memberikan peringatan kepada PTS terhadap laporan masyarakat tentang pelaksanaan kelas jauh, pelanggaran proses belajar mengajar, kalau diperlukan dilanjutkan dengan sidak untuk mengetahui kondisi dilapangan, untuk kemudian dilaporkan ke Dirjen Dikti.
6. Menyediakan buku pedoman pendirian PTS baru dan / atau pembukaan program studi baru.
7. Melayani permintaan data dan informasi dari unit kerja lintas sektoral menyangkut kondisi PTS (Jumlah PTS, jumlah program studi, jumlah dosen, jumlah mahasiswa, jumlah lulusan dan sebagainya).
8. Membantu masyarakat dalam hal memberikan informasi mengenai tata cara proses pendirian PTS baru dan/atau pembukaan program studi baru

sesuai dengan SK. Mendiknas No. 234/U/2000 dan SK. Dirjen Dikti No. 108/DIKTI/Kep/2001.

9. Melayani membuat surat keterangan tentang status PTS, bagi Lembaga/Instansi yang tidak/belum yakin dengan status sebuah PTS.
10. Membuat laporan LAKIP Sub bagian.

Bagian Administrasi Akreditasi dan Kelembagaan terdiri dari:

1. Sub Bagian Akreditasi dan Kelembagaan.
2. Sub Bagian Adm. Ujian Negara dan Kemahasiswaan.

## BAB III

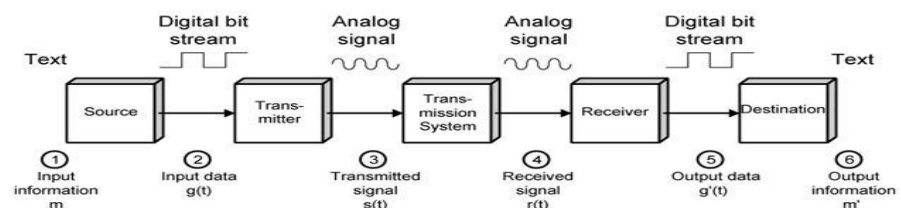
### TINJAUAN PUSTAKA

#### 3.1. Teori Pendukung

Teori pendukung terdiri dari teori – teori yang dikemukakan oleh para ahli maupun penjelasan yang berhubungan dengan penelitian ini, antara lain: komunikasi data, *internet*, jaringan komputer, terminologi jaringan, *web server*, *content management system*, komponen penunjang drupal, *domain name system*, linux dan Debian.

##### 3.1.1. Komunikasi data

Menurut Sutanta (2005:5) Komunikasi data merupakan transmisi data elektronik melalui sebuah media. Media tersebut dapat berupa kabel tembaga, *fiber optic*, radio *frequency* dan *micro wave* serta dibantu dengan alat-alat pengubah transmisi lainnya.



**Gambar 3.1** Model Komunikasi Data Sederhana  
(sumber : <http://ilmukomunikasidata.wordpress.com>)

##### a. Bagian-bagian model komunikasi sederhana:

Menurut Sutanta (2005:6) *Source* adalah sumber program yang berisi beberapa perintah yang akan dikerjakan oleh komputer.

*Transmitter* adalah pemancar atau perangkat I/O. Sedangkan Sistem *Transmisi* sistem yang berfungsi untuk konversi torsi dan kecepatan (putaran) dari mesin menjadi torsi dan kecepatan yang berbeda-beda untuk diteruskan ke penggerak akhir. Konversi ini mengubah kecepatan putar yang tinggi menjadi lebih rendah tetapi lebih bertenaga, atau sebaliknya. Kemudian *Receiver* adalah perangkat penerima informasi atau sinyal. Sedangkan *Destination* adalah berarti tujuan, tujuan dari suatu proses, misalnya kemana lalu lintas jaringan ditujukan oleh pengirim (*source*).

### **3.1.2. Internet**

*Internet* yaitu kelompok atau kumpulan dari jutaan komputer. Penggunaan *Internet* memungkinkan kita untuk mendapatkan informasi dari komputer yang ada di dalam kelompok tersebut dengan asumsi bahwa pemilik komputer memberikan izin akses. Untuk mendapatkan sebuah informasi, sekumpulan protokol harus digunakan, yaitu sekumpulan aturan yang menetapkan bagaimana informasi dapat dikirim dan diterima (Simarmata, 2010:47).

Salah satu unsur yang paling umum digunakan dari *Internet* selain *e-mail* adalah *World Wide Web*. WWW atau yang sering juga disebut sebagai "*Web*" saja merupakan aplikasi *Internet* yang paling populer. Karena *Web* begitu populer, banyak orang kemudian salah mengidentikkanya dengan *Internet* (Simarmata, 2006:47).

### 3.1.2.1. Sejarah *Internet*

*Internet* pertama kali digunakan sebagai proyek penelitian yang ditemukan oleh *Advanced Research Project Agency* (ARPA) *Department of Defense* (DOD) di Amerika Serikat. Pada dasarnya, *Internet* digunakan untuk menghubungkan komputer. Versi yang pertama disebut ARPANET. Pada tahun 1972, ARPA berubah menjadi DARPA dengan tetap mempromosikan proyek ARPANET.

Pengembangan *Internet* dengan jenis peralatan yang berbeda, namun bisa saling berhubungan satu sama lain merupakan tantangan yang besar pada saat itu. Pada tahun 1973-1974, peneliti merancang sebuah *Transmission Control Protocol/Internet Protocol* (TCP/IP). Pada awalnya, TCP/IP dimaksudkan untuk menyediakan dukungan untuk kebutuhan berikut:

1. interoperabilitas antar sistem heterogen,
2. komunikasi *end-to-end* berbagai jaringan yang berbeda, dan
3. operasi otomatis dan sempurna di dalam menghadapi terjadinya kegagalan hubungan data.

Pada awal tahun 1980-an, ARPANET dipecah menjadi dua bagian, yaitu MILNET dan ARPANET karena pertimbangan keamanan. Pihak militer berjalan terus dengan MILNET, sedangkan penelitian, pengembangan, dan sektor lain

tetap memakai ARPANET. Pada pertengahan tahun 1980-an, *National Science Foundation* (NSF) di Washington, D.C. mendistribusikan teknologi *Internet* kepada beberapa universitas (Berkeley, MIT, Stanford, dan UCLA). Selanjutnya, *Internet* pun mulai menyebar di dunia (Simarmata, 2010:51).

### **3.1.3. Jaringan Komputer**

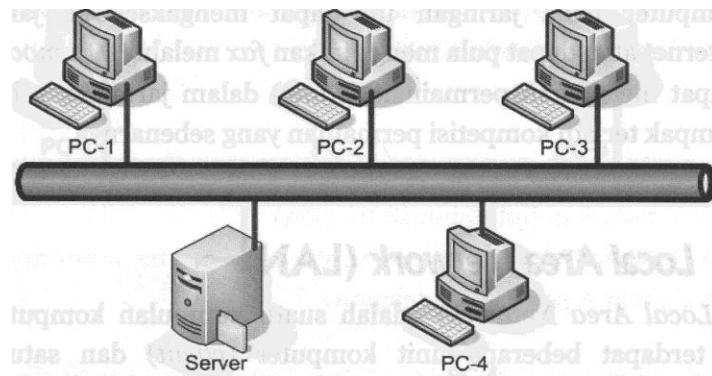
Jaringan komputer dibangun dalam bentuk dan ukuran yang berbeda, bergantung kondisi dan kebutuhan. Desain dari jaringan komputer disebut terminologi jaringan. Sebagai pengetahuan, saat ini *area network* yang ada adalah:

#### **a. Local Area Network (LAN)**

Menurut Sugeng (2010:19) LAN (*Local Area Network*) adalah suatu kumpulan komputer, dimana terdapat beberapa unit komputer (*client*) dan satu unit komputer untuk bank data (*server*).

Antara masing – masing *client* maupun antara *client* dan *server* dapat saling bertukar *file* maupun saling menggunakan printer yang terhubung pada unit-unit komputer yang terhubung pada jaringan LAN. Berdasarkan kabel yang digunakan, pada umumnya cara membuat jaringan LAN, yaitu dengan kabel BNC dan kabel UTP.





**Gambar 3.2.** *Local Area Network*

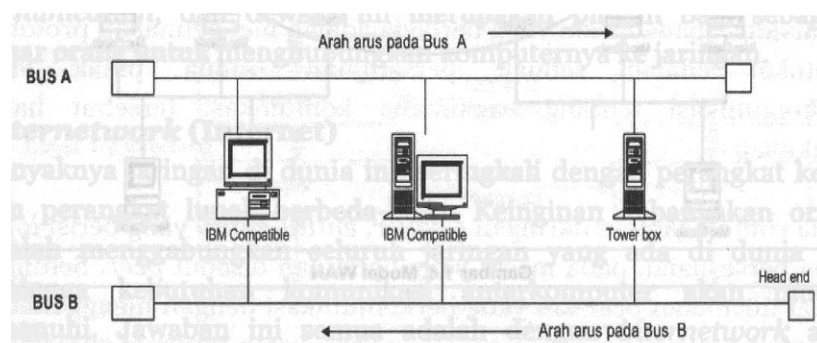
Sumber: Sugeng (2010:19)

Keuntungan dari Jaringan LAN:

1. Pertukaran file dapat dilakukan dengan mudah (*File sharing*), sehingga dapat dibentuk seolah-olah sistem mempunyai media penyimpanan (*storage*) yang besar, karena *user* dapat melakukan kerja *file* dari sembarang *storage* yang dibentuk dalam jaringan.
2. Pemakaian *printer* dapat dilakukan oleh semua unit komputer (*Printer Sharing*), sehingga seolah-olah setiap komputer mempunyai printer sendiri.
3. File-file data dapat disimpan pada *server*, sehingga data dapat diakses dari semua *client* menurut otorisasi sekuritas dari semua karyawan, yang dapat dibuat berdasarkan struktur organisasi perusahaan sehingga keamanan data terjamin.
4. File data yang keluar/masuk dari/ke *server* dapat dikontrol, atau dapat dibuat piranti lunak khusus untuk pantauan kondisi

- jaringan LAN. Dengan jaringan LAN dapat mudah dilakukan pantauan lalu lintas data dari mana sumbernya dan kemana data tersebut dikirim dan seberapa besar data tersebut dikirim.
5. Proses backup data menjadi lebih mudah dan cepat.
  6. Risiko kehilangan data oleh virus komputer menjadi sangat kecil sekali, karena memiliki data backup yang terpusat.
  7. Komunikasi antarkaryawan dapat dilakukan dengan menggunakan *E-Mail & Chat* secara multimedia. Pada jaringan LAN sistem lebih baik dibanding *Workgroup* karena dapat dibuat *Mail Server* dan *Chat Server*.
  8. Bila salah satu unit komputer terhubung dengan modem yang dihubungkan ke Internet, maka semua atau sebagian unit komputer pada jaringan ini dapat mengakses ke jaringan internet atau dapat pula mengirimkan *fax* melalui satu modem.

### b. Metropolitan Area Network (MAN)

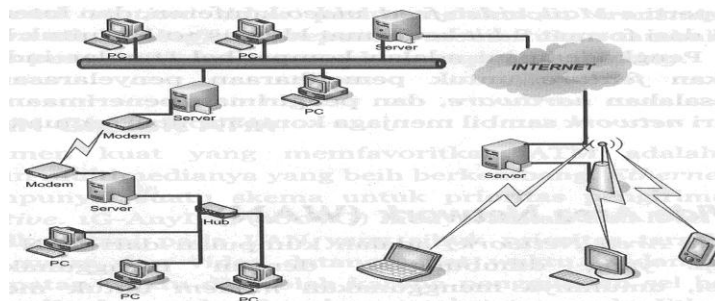


**Gambar 3.3.** Model Jaringan MAN

Sumber: Sugeng (2010:20)

Metropolitan Area Network atau disingkat dengan MAN adalah suatu jaringan dalam suatu kota dengan transfer data berkecepatan tinggi, yang menghubungkan berbagai lokasi seperti kampus, perkantoran, pemerintahan, dan sebagainya. Jaringan MAN adalah gabungan dari beberapa LAN. Jangkauan dari MAN ini antar 10 hingga 50 km, MAN ini merupakan jaringan yang tepat untuk membangun jaringan antar kantor-kantor dalam satu kota antara pabrik/instansi dan kantor pusat yang berada dalam jangkauannya.

### c. Wide Area Network (WAN)



**Gambar 3.4.** Model Jaringan WAN

Sumber: Sugeng (2010:21)

Wide Area Network (WAN) merupakan jaringan komputer yang mencakup area yang besar sebagai contoh yaitu jaringan komputer antar wilayah, kota atau bahkan negara, atau dapat didefinisikan juga sebagai jaringan komputer yang membutuhkan router dan saluran komunikasi publik.

#### **d. Internetwork (Internet)**

Terdapat banyak jaringan didunia ini, seringkali menggunakan perangkat keras dan perangkat lunak yang berbeda-beda. Orang yang terhubung ke jaringan sering berharap untuk bisa berkomunikasi dengan orang lain yang terhubung ke jaringan lainnya. Keinginan seperti ini memerlukan hubungan antar jaringan yang seringkali tidak kompatibel dan berbeda. Untuk melakukan hal ini diperlukan sebuah mesin yang disebut gateway guna melakukan hubungan dan melaksanakan terjemahan yang diperlukan, baik perangkat keras maupun perangkat lunaknya. Kumpulan jaringan yang terinterkoneksi inilah yang disebut dengan internet.

#### **e. Jaringan Tanpa Kabel (*Wireless Network*)**

Jaringan tanpa kabel merupakan suatu solusi terhadap komunikasi yang tidak bisa dilakukan dengan jaringan yang menggunakan kabel. Misalnya orang yang ingin mendapat informasi atau melakukan komunikasi walaupun sedang berada diatas mobil atau pesawat terbang, maka mutlak jaringan tanpa kabel diperlukan karena koneksi kabel tidaklah mungkin dibuat di dalam mobil atau pesawat. Saat ini jaringan tanpa kabel sudah marak digunakan dengan memanfaatkan jasa satelit dan mampu memberikan kecepatan akses yang lebih cepat dibandingkan

dengan jaringan yang menggunakan kabel Jaringan tanpa kabel dapat juga tidak hanya berbasis satelit juga berupa sambungan wireless.

### 3.1.4. Terminologi Jaringan Komputer

Komputer dalam jaringan dapat saling berhubungan melalui kabel, jaringan telepon, gelombang radio, satelit ataupun sinar infra merah. Ada tiga tipe jaringan dalam hubungannya dengan luas area yang dicakup yaitu:

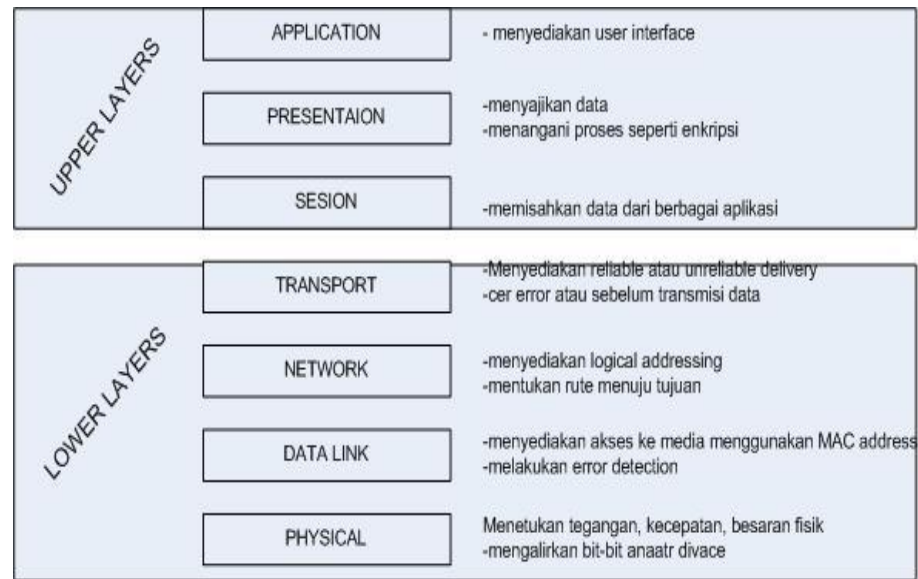
#### a. OSI Model

Suatu jaringan computer LAN dibangun dengan memperhatikan arsitektur standar yang dibuat standar industri dunia. Standar jaringan yang saat ini diakui dunia adalah *The Open System Connection* atau OSI yang dibuat oleh lembaga ISO (*The International Standard Organization*), Amerika Serikat. Seluruh fungsi kerja, jaringan komputer dan komunikasi antar terminal diatur dalam standar ini.

Menurut Sopandi (2008:53), OSI adalah suatu standar antar mesin yang terdiri atas 7 lapisan. Ketujuh lapisan tersebut mempunyai peran dan fungsi yang berbeda satu terhadap yang lain. Setiap *layer* bertanggung jawab secara khusus pada proses komunikasi data. Misalnya, satu *layer* bertanggung jawab untuk

membentuk koneksi antar perangkat, sementara *layer* lainnya bertanggung jawab untuk mengoreksi terjadinya “*error*” selama proses transfer data berlangsung. Model layer OSI dibagi dalam dua group : “*upper layer*” dan “*lower layer*”. “*Upper layer*” fokus pada aplikasi pengguna dan bagaimana *file* direpresentasikan di komputer. Untuk *Network Engineer*, bagian utama yang menjadi perhatian adalah pada “*lower layer*”. *Lower layer* adalah intisari komunikasi data melalui jaringan aktual.

Tujuan utama penggunaan model OSI adalah untuk membantu desainer jaringan memahami fungsi dari setiap *layer* yang berhubungan dengan aliran komunikasi data, termasuk jenis-jenis protokol jaringan dan metode transmisi. Model dibagi menjadi 7 *layer*, dengan karakteristik dan fungsinya. Tiap *layer* harus dapat berkomunikasi dengan *layer* di atasnya maupun di bawahnya melalui serentetan protokol dan standar (Sopandi, 2008:53).



**Gambar 3.5** Lapisan OSI

Sumber : (Diolah Sendiri)

Menurut Sopandi (2008:54) Penjelasan dari masing-masing layer tersebut adalah sebagai berikut:

#### 1. Layer 7: *Application Layer*

Lapisan paling tinggi ini mengatur interaksi pengguna komputer dengan program aplikasi yang dipakai. Lapisan ini juga mengatur bagaimana aplikasi dapat mengakses jaringan, dan kemudian membuat pesan-pesan kesalahan. Protokol yang berada dalam lapisan ini adalah HTTP, FTP, SMTP, dan NFS.s

#### 2. Layer 6 : *Presentation Layer*

Pada lapisan ini dilakukan konversi data agar data yang dikirim dapat dimengerti oleh penerima, kompresi teks dan penyediaan data.

Protokol yang berada dalam level ini adalah perangkat lunak *redirector (redirector software)*, seperti layanan *workstation* (dalam *Windows NT*) dan juga *Network shell* (semacam *Virtual Network Computing (VNC)* atau *Remote Desktop Protokol (RDP)*).

### 3. Layer 5 : *Session Layer*

Lapisan ini menyiapkan saluran komunikasi dan terminal dalam hubungan antar terminal, mengoordinasikan proses pengiriman, penerimaan serta mengatur pertukaran data.

### 4. Layer 4 : *Transport Layer*

Lapisan ini mengatur keutuhan data, menerima data dari lapisan *session* dan meneruskannya ke lapisan *network*. Lapisan ini berfungsi untuk memecah data ke dalam paket-paket data serta memberikan nomor urut ke paket-paket tersebut sehingga dapat disusun kembali pada sisi tujuan setelah diterima. Selain itu, pada level ini juga membuat sebuah tanda bahwa paket diterima dengan sukses (*acknowledgement*).

### 5. Layer 3: *Network Layer*

Lapisan ini menentukan rute pengiriman dan mengendalikan kemacetan (mendefinisikan alamat-alamat IP), membuat *header* untuk paket-paket, dan kemudian melakukan *routing* melalui *internet working* dengan menggunakan *router* dan *switch layer-3*. Agar data sampai ditempat tujuan dengan benar.



## 6. Layer 2 : *Data Link Layer*

Pada lapisan ini data diubah dalam bentuk paket, sinkronisasi paket yang dikirim maupun yang diterima menjadi format yang disebut *frame*. Selain itu, pada level ini terjadi koreksi kesalahan, *flow control*, pengalamatan perangkat keras (seperti halnya *Media Access Control Address (MAC Address)*), dan menentukan bagaimana perangkat-perangkat jaringan seperti *hub*, *Bridge*, *repeater*, dan *switch layer 2* beroperasi. Spesifikasi IEEE 802, membagi level ini menjadi dua level anak, yaitu lapisan *Logical Link Control (LLC)*, dan lapisan *Media Access Control (MAC)*.

## 7. Layer 1 : *Physical Layer*

Lapisan terendah ini mengatur sinkronisasi pengiriman dan penerimaan data, spesifikasi meknik, elektrik dan *interface* antar terminal, seperti besar tegangan, frekuensi, impedansi, koneksi *pin* dan jenis kabel. Layer ini juga untuk mendefinisikan media transmisi jaringan, arsitektur jaringan (seperti halnya *Ethernet* atau *Token Ring*), topologi jaringan dan pengabelan. Selain itu, level ini juga mendefinisikan bagaimana *Network Interface Card (NIC)* dapat berinteraksi dengan media kabel atau radio.

## **b. TCP/IP**

TCP/IP (singkatan dari *Transmission Control Protocol/Internet Protocol*) adalah standar komunikasi data yang digunakan oleh komunitas *internet* dalam proses tukar-menukar data

dari satu komputer ke komputer lain di dalam jaringan *internet*. Protokol ini tidaklah dapat berdiri sendiri, karena memang protokol ini berupa kumpulan protokol (*protocol suite*). Protokol ini juga merupakan protokol yang paling banyak digunakan saat ini. Data tersebut diimplementasikan dalam bentuk perangkat lunak (*software*) di sistem operasi. Istilah yang diberikan kepada perangkat lunak ini adalah TCP/IP *stack* (Sofana;2008;87).

Protokol TCP/IP dikembangkan pada akhir dekade 1970-an hingga awal 1980-an sebagai sebuah protokol standar untuk menghubungkan komputer-komputer dan jaringan untuk membentuk sebuah jaringan yang luas (WAN). TCP/IP merupakan sebuah standar jaringan terbuka yang bersifat independen terhadap mekanisme transport jaringan fisik yang digunakan, sehingga dapat digunakan di mana saja. Protokol ini menggunakan skema pengalamatan yang sederhana yang disebut sebagai alamat IP (*IP Address*) yang mengizinkan hingga beberapa ratus juta komputer untuk dapat saling berhubungan satu sama lainnya di Internet. Protokol ini juga bersifat *routable* yang berarti protokol ini cocok untuk menghubungkan sistem-sistem berbeda (seperti *Microsoft Windows* dan keluarga UNIX) untuk membentuk jaringan yang heterogen.

Protokol TCP/IP selalu berevolusi seiring dengan waktu, mengingat semakin banyaknya kebutuhan terhadap jaringan

komputer dan *internet*. Pengembangan ini dilakukan oleh beberapa badan, seperti halnya *Internet Society (ISOC)*, *Internet Architecture Board (IAB)*, dan *Internet Engineering Task Force (IETF)*. Macam-macam protokol yang berjalan di atas TCP/IP, skema pengalamatan, dan konsep TCP/IP didefinisikan dalam dokumen yang disebut sebagai *Request for Comments (RFC)* yang dikeluarkan oleh IETF (Sofana, 2008:88).

### c. IP Address

*Internet Protocol (IP) address* adalah alamat numerik yang ditetapkan untuk sebuah komputer yang berpartisipasi dalam jaringan komputer yang memanfaatkan *Internet Protocol* untuk komunikasi antara *node*-nya. Walaupun alamat IP disimpan sebagai angka biner, mereka biasanya ditampilkan agar memudahkan manusia menggunakan notasi, seperti 208.77.188.166 (untuk IPv4), dan 2001: db8: 0:1234:0:567:1:1 (untuk IPv6). Peran alamat IP adalah "Sebuah nama menunjukkan apa yang kita mencari. Sebuah alamat menunjukkan di mana ia berada. Sebuah route menunjukkan bagaimana menuju ke sana".

Perancang awal dari TCP/IP menetapkan sebuah alamat IP sebagai nomor 32-bit, dan sistem ini, yang kini bernama *Internet Protocol Version 4 (IPv4)*, masih digunakan hari ini. Namun, karena pertumbuhan yang besar dari *internet* dan penipisan yang terjadi

pada alamat IP, dikembangkan sistem baru (IPv6), menggunakan 128 bit untuk alamat, dikembangkan pada tahun 1995 dan terakhir oleh standar RFC 2460 pada tahun 1998.

*Internet Protocol* juga memiliki tugas *routing* paket data antara jaringan, alamat IP dan menentukan lokasi dari *node* sumber dan *node* tujuan dalam topologi dari sistem *routing*.

Dengan pengembangan jaringan pribadi, alamat IPv4 menjadi kekurangan, sekelompok alamat IP *private* dikhususkan oleh RFC 1918. Alamat IP *private* ini dapat digunakan oleh siapa saja di jaringan pribadi. Mereka sering digunakan dengan *Network Address Translation* untuk menyambung ke *internet* umum global, IP juga terbagi menjadi 5 kelas (Sofana, 2008:93).

### **Kelas A**

Oktet pertama dari 1-126, Alamat kelas A diberikan untuk jaringan skala besar. Nomor urut bit tertinggi di dalam alamat IP kelas A selalu *diset* dengan nilai 0 (nol). Tujuh bit berikutnya - untuk melengkapi oktet pertama - akan membuat sebuah *network identifier*. Sisanya 24 bit merepresentasikan *host identifier*. Ini mengizinkan kelas A memiliki hingga 126 jaringan, dan 16,777,214 host tiap jaringannya. Alamat dengan oktet awal 127 tidak diizinkan, karena digunakan untuk mekanisme *Interprocess Communication* (IPC) di dalam mesin yang bersangkutan (Sofana, 2008:106).

### **Kelas B**

Oktet pertama dari 128-191, Alamat-alamat *unicast* kelas B dikhususkan untuk jaringan skala menengah hingga skala besar. Dua bit pertama di dalam oktet pertama alamat IP kelas B selalu diset ke bilangan *biner* 10. 14 bit berikutnya (untuk melengkapi dua oktet pertama), akan membuat sebuah *network identifier*. 16 bit sisanya (dua oktet terakhir) merepresentasikan *host identifier*. Kelas B dapat memiliki 16,384 network, dan 65,534 host untuk setiap network-nya (Sofana, 2008:106).

### **Kelas C**

Oktet pertama dari 192-223, Alamat IP kelas C digunakan untuk jaringan berskala kecil. Tiga *bit* pertama di dalam oktet pertama alamat kelas C selalu diset ke nilai biner 110. 21 bit selanjutnya (untuk melengkapi tiga oktet pertama) akan membentuk sebuah *network identifier*. 8 bit sisanya (sebagai oktet terakhir) akan merepresentasikan *host identifier*. Ini memungkinkan pembuatan total 2,097,152 buah *network*, dan 254 *host* untuk setiap *network*-nya (Sofana, 2008:107).

### **Kelas D**

Oktet pertama dari 224-239, Alamat IP kelas D disediakan hanya untuk alamat-alamat *IP multicast*, sehingga berbeda dengan tiga kelas di atas. Empat *bit* pertama di dalam IP kelas D selalu diset ke bilangan biner 1110. 28 *bit* sisanya digunakan sebagai alamat

yang dapat digunakan untuk mengenali *host*. Untuk lebih jelas mengenal alamat ini, lihat pada bagian Alamat *Multicast IPv4* (Sofana, 2008:107).

### **Kelas E**

Oktet pertama dari 240-255 Oktet pertama dari 1, Alamat IP kelas E disediakan sebagai alamat yang bersifat "eksperimental" atau percobaan dan dicadangkan untuk digunakan pada masa depan. Empat bit pertama selalu diset kepada bilangan *biner* 1111. 28 bit sisanya digunakan sebagai alamat yang dapat digunakan untuk mengenali *host* (Sofana, 2008:108).

### **3.1.5. Topologi Jaringan Komputer**

Topologi Jaringan adalah gambaran dari struktur jaringan yang akan dibangun dan merupakan suatu aturan/*rules* bagaimana menghubungkan komputer (*node*) satu sama lain secara fisik dan pola hubungan antara komponen-komponen yang berkomunikasi melalui media/peralatan jaringan, seperti: *server*, *workstation*, *hub/switch*, dan pengabelannya (Utomo, 2006:21). Bentuk-bentuk topologi fisik yang ada dalam sistem jaringan dapat dikelompokkan dalam dua topologi fisik dasar yaitu:

#### **1. Point to Point**

Topologi fisik *point to point* adalah topologi yang menggambarkan bentuk hubungan antara dua komputer atau lebih tepatnya antara dua titik. Dua komputer dapat

dihubungkan dengan beberapa cara antara lain sebagai berikut  
Jika jarak antara dua komputer tersebut tidak jauh maka dapat dihubungkan langsung menggunakan media transmisi kabel *koaksial* atau *UTP*. Jarak terjauh dari hubungan ini tergantung pada jenis media transmisi yang digunakan. Jika jarak antara komputer jauh maka hubungan dilakukan menggunakan media transmisi kabel telepon, kabel serat optik atau gelombang elektromagnetik. Dalam hubungan ini biasanya digunakan modem yang tipenya disesuaikan dengan media transmisi yang digunakan. Berikut adalah gambar visualisasinya:



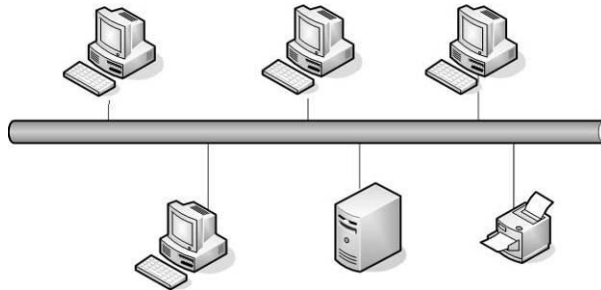
**Gambar 3.6** *Point to Point*

Sumber : (Diolah Sendiri)

## **2. Multi Point**

Topologi fisik *multi point* adalah topologi yang menggambarkan bagaimana beberapa komputer (lebih dari dua) terhubung menggunakan media transmisi. Ada beberapa tipe topologi multi point dalam jaringan yaitu topologi *bus*, topologi *star*, topologi *ring* dan topologi *tree*.

### a. Topologi *Bus*

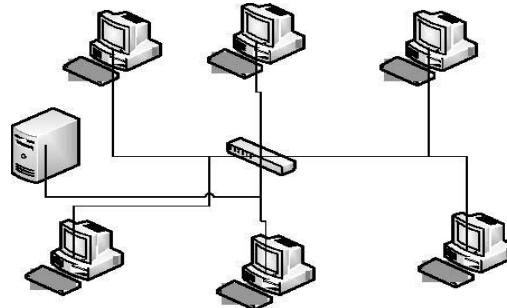


**Gambar 3.7** Topologi *Bus*  
Sumber : (Diolah Sendiri)

Topologi ini adalah topologi yang awal di gunakan untuk menghubungkan komputer. Dalam topologi ini masing masing komputer akan terhubung ke satu kabel panjang dengan beberapa terminal, dan pada akhir dari kabel harus di akhiri dengan satu terminator. Topologi ini memiliki beberapa kekurangan diantaranya kemungkinan terjadi nya tabrakan aliran data, jika salah satu perangkat putus atau terjadi kerusakan pada satu bagian komputer maka jaringan langsung tidak akan berfungsi sebelum kerusakan tersebut di atasi. Topologi ini awalnya menggunakan kabel *Coaxial* sebagai media pengantar data dan informasi. Tapi pada saat ini topologi ini di dalam membangun jaringan komputer dengan menggunakan kabal serat optik (*fiber optic*) akan tetapi digabungkan dengan topologi jaringan yang lain untuk memaksimalkan performanya (Utomo, 2006:21).



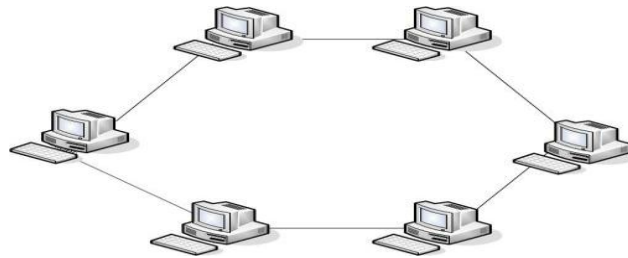
### b. Topologi *Star*



**Gambar 3.8** Topologi *Star*  
Sumber : (Diolah Sendiri)

Topologi bintang atau topologi *star*. Pada topologi ini kita sudah menggunakan bantuan alat lain untuk mengkoneksikan jaringan komputer. Contoh alat yang di pakai disini adalah *hub*, *switch*, dll. Pada gambar jelas terlihat satu hub berfungsi sebagai pusat penghubung komputer-komputer yang saling berhubungan. Keuntungan dari topologi ini sangat banyak sekali diantaranya memudahkan admin dalam mengelola jaringan, memudahkan dalam penambahan komputer atau terminal, kemudahan mendeteksi kerusakan dan kesalahan pada jaringan. Tetapi dengan banyaknya kelebihan bukan dengan artian topologi ini tanpa kekurangan. Kekurangannya diantaranya pemborosan terhadap kabel, kontrol yang terpusat pada hub terkadang jadi permasalahan kritis kalau seandainya terjadi kerusakan pada *hub* maka semua jaringan tidak akan bisa di gunakan (Utomo, 2006:23).

### c. Topologi *Ring*



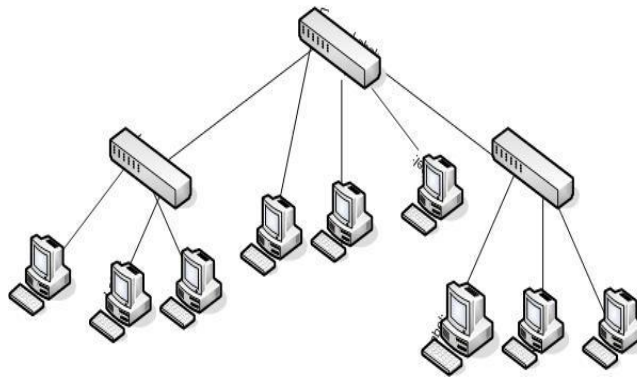
**Gambar 3.9** Topologi *Ring*

Sumber : (Diolah Sendiri)

Topologi cincin atau yang sering disebut dengan *ring* topologi adalah topologi jaringan dimana setiap komputer yang terhubung membuat lingkaran. Dengan artian setiap komputer yang terhubung kedalam satu jaringan saling terkoneksi ke dua komputer lainnya sehingga membentuk satu jaringan yang sama dengan bentuk cincin. Adapun kelebihan dari topologi ini adalah kabel yang digunakan bisa lebih dihemat. Tetapi kekurangan dari topologi ini adalah pengembangan jaringan akan menjadi susah karena setiap komputer akan saling terhubung (Utomo, 2006:22).

### d. Topologi *Tree*

Intinya bahwa sebuah jaringan bisa jadi kombinasi dari dua atau tiga topologi diatas. Misalnya saja ada yang menyebut *tree topology*, dimana sebenarnya topologi ini merupakan gabungan atau kombinasi dari ketiga topologi yang ada.



**Gambar 3.10** Topologi *Tree*  
Sumber : (Diolah Sendiri)

Topologi pohon atau di sebut juga topologi hirarki dan bisa juga disebut topologi bertingkat merupakan topologi yang bisa di gunakan pada jaringan di dalam ruangan kantor yang bertingkat. Pada gambar bisa kita lihat hubungan antar satu komputer dengan komputer lain merupakan percabangan dengan hirarki yang jelas. sentral pusat atau yang berada pada bagian paling atas merupakan sentral yang aktif sedangkan sentral yang ada di bawahnya adalah sentral yang pasif (Utomo, 2006:24).

Penulis menggunakan topologi *Star*, karena topologi *Star* dapat memudahkan admin dalam mengelola jaringan, memudahkan dalam penambahan komputer atau terminal, kemudahan mendeteksi kerusakan dan kesalahan pada jaringan

### 3.1.6. Web Server

*Web Server* adalah potongan perangkat lunak yang mendukung berbagai protokol *Web*, seperti HTTP, HTTPS, dan lain – lain untuk memproses permintaan *client* (Simarmata, 2010:88).

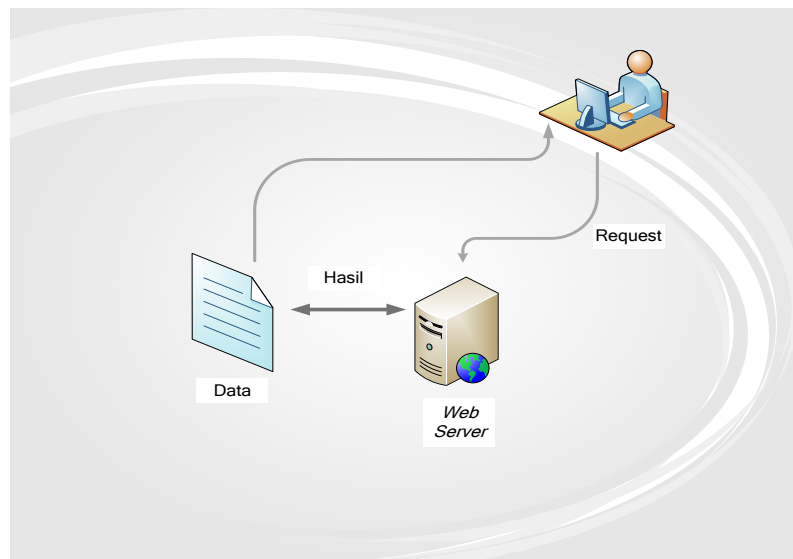
Ada dua komponen dasar di dalam arsitektur *web*, yaitu *browser web* dan *server web*. *browser web* menawarkan antarmuka grafis untuk pengguna dan bertanggung jawab untuk komunikasi dengan *server web*. Protokol komunikasi antara *browser* dan *server web* mengikuti protokol HTTP yang distandarisasi.

Antarmuka antar pengguna dan *browser* adalah bahasa HTML yang terstandarisasi. Sedangkan komunikasi antara *browser* dan *server* menggunakan protokol HTTP. HTTP disebut *client / server*, dengan arti bahwa *browser* adalah *client* dan *server web* adalah *server* (Simarmata, 2010:54).

#### 3.1.6.1. Cara Kerja Web Server

*Web server* menunggu permintaan dari *client* yang menggunakan *browser* seperti *Internet Explorer*, *Mozilla Firefox*, *Opera* dan program *browser* lainnya. Jika ada permintaan dari *browser*, maka *web server* akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke *browser*. Data ini mempunyai format yang standar, disebut dengan format SGML (*Standar*

*general markup language*). Data yang berupa format ini kemudian akan ditampilkan oleh *browser* sesuai dengan kemampuan *browser* tersebut (Sugeng, 2010:200).



**Gambar 3.11** Cara Kerja *Web Server*  
Sumber : (Diolah Sendiri)

### 3.1.7. *Content Management System*

*Content Management System* adalah sistem perangkat lunak yang memungkinkan seseorang untuk mengatur proses berjalannya suatu situs *Web*. Dengan program *CMS*, bisa menambahkan, mengubah, bahkan mengurangi isi dari situs *Web* (Simarmata, 2010:22).

### 3.1.7.1. Macam – Macam *Content Management System*

#### 1. Drupal

Drupal adalah sebuah perangkat lunak sistem manajemen konten yang bebas dan terbuka. Perangkat lunak ini didistribusikan di bawah lisensi GPL, pengembangan dan perawatannya dilakukan oleh ribuan komunitas pengguna dan pengembang di seluruh dunia. Drupal dapat diunduh secara bebas dan dapat digunakan secara bebas juga sehingga memungkinkan setiap orang baik secara individu maupun komunitas untuk mempublikasikan, mengatur, dan mengorganisasikan berbagai jenis isi/ konten pada situs *web* (Simarmata, 2010:24).

Drupal dapat digunakan untuk membangun

1. portal *Web* komunitas,
2. situs diskusi,
3. situs *Web* perusahaan,
4. aplikasi internal,
5. situs *Web* personal atau blog,
6. aplikasi komersial *e-commerce*,
7. kumpulan sumber informasi,
8. situs jaringan sosial,
9. situs *Web* multibahasa, dan lain – lain.

Drupal dapat langsung digunakan setelah mengunduh dan menginstalnya. Dengan berbagai fungsionalitas yang sudah dibangun dan dikombinasikan dengan ribuan paket modul Drupal dan tema tampilan bebas yang tersedia. Dengan bahasa PHP, Drupal dapat berjalan diberbagai sistem operasi, seperti *Microsoft Windows, Mac OS X, Linux, FreeBSD*. Mendukung *server Web Apache* atau IIS, dan membutuhkan program basis sata, seperti *MySQL* dan *PostgreSQL* untuk menyimpan semua datanya (Simarmata, 2010:25).

#### **a. Keunggulan Drupal**

1. Diklaim sebagai *CMS* terbaik untuk saat ini dan mengalahkan Joomla!
2. Digunakan pada situs *Web* yang sederhana hingga situs *Web* profesional.
3. Mudah dioperasikan.
4. Dapat dijalankan di komputer lokal.
5. Banyak *plugin* tambahan.
6. *Template* dapat diubah – ubah sekehendak hati dan mengijinkan pembuatan *template*.
7. Mendukung SEF (*Search Engine Friendly*) secara *default* sehingga lebih mudah ditemukan di mesin pencari.

## 2. Joomla

Joomla Adalah Sistem Manajemen Konten yang bebas dan terbuka (*free open source*), dan ditulis dengan PHP dan basis data MySQL untuk keperluan di *Intranet*. Joomla! Pertama kali dirilis pada versi 1.0.0. Fitur – fitur Joomla! Diantaranya adalah sistem *caching* untuk peningkatan performa, RSS, *blogs*, *polling*, dan lainnya.

Joomla Menggunakan lisensi GPL. Kata *Joomla* berasal dari kata Swahili yang mengandung arti kebersamaan. Joomla! terdiri dari 3 elemen dasar yaitu *server Web*, skrip PHP, dan basis data MySQL (Simarmata, 2010:28).

## 3. WordPress

*WordPress* adalah sebuah perangkat lunak blog yang ditulis dalam PHP dan mendukung sistem basis data MySQL. *WordPress* adalah penerus resmi dari *b2/cafelog* yang dikembangkan oleh Michael Valdrighi. Nama *WordPress* diusulkan oleh Cristine Selleck, teman dari ketua pengembang, Matt Mullenweg. Rilis terbaru *WordPress* adalah versi 2.5. *WordPress* didistribusikan dengan lisensi GNU *General Public License* (Simarmata, 2010:39).

Penulis menggunakan Drupal, karena Drupal dapat digunakan pada situs *Web* yang sederhana hingga situs *Web*



profesional mudah dioperasikan dan dapat dijalankan di komputer lokal.

### 3.1.7.2. Komponen Penunjang Drupal

#### 1. PHP (*Hypertext Preprocessor*)

PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman berbasis *web* yang memiliki kemampuan untuk memproses dan mengolah data secara dinamis. PHP dapat dikatakan sebagai sebuah *server-side embedded script language*, artinya semua sintaks dan perintah program yang ditulis akan sepenuhnya dijalankan oleh *server*, tetapi dapat disertakan pada halaman HTML biasa. Pada umumnya, semua aplikasi yang dibangun menggunakan PHP akan memberikan hasil pada *web browser*, tetapi prosesnya secara keseluruhan dijalankan pada *server*. Pada prinsipnya, *server* akan bekerja apabila ada permintaan dari *client* (Tim Wahana Komputer, 2009:3).

#### **Keuntungan PHP**

PHP (*Hypertext Processor*) mempunyai beberapa kemampuan yang merupakan salah satu keuntungan PHP, Kemampuan tersebut antara lain :

1. Cara koneksi dan *query database* yang sederhana.
2. Dapat bekerja pada sistem operasi berbasis *Windows, Linux, Max OS*, dan kebanyakan varian UNIX.
3. Biaya yang dibutuhkan untuk menggunakan PHP tidak mahal, atau bahkan gratis
4. Mudah digunakan karena memiliki fitur dan fungsi khusus untuk membuat *web* dinamis
5. *Security sistem* yang cukup tinggi (Tim Wahana Komputer, 2009:10).

## **2. Mysql**

*Mysql* adalah program *database* yang mampu mengirim dan menerima data dengan sangat cepat dan *multi user*. *Mysql* memiliki dua bentuk lisensi, yaitu *free software* dan *shareware*. (Tim Wahana Komputer, 2010:5).

### **Keuntungan Mysql**

1. Banyak ahli berpendapat *Mysql* merupakan *server* tercepat.
2. *Mysql* merupakan sistem manajemen *database* yang *open source*.
3. *Mysql* memiliki performa yang tinggi tapi simpel.

4. *Database Mysql* mengerti bahasa SQL (*Structured Query Language*).
5. *Mysql* dapat diakses melalui protocol ODBC (*Open Database Connection*) buatan *Microsoft*. Menyebabkan *Mysql* dapat diakses oleh banyak *software*.
6. Semua *client* dapat mengakses *server* dalam satu waktu tanpa harus menunggu yang lain untuk mengakses *database*.
7. *Mysql* dapat menyimpan data berkapasitas besar, sampai berukuran Gigabyte.
8. *Mysql* dapat berjalan diberbagai *operating system* seperti Linux, *Windows*, Solari, dan lain – lain (Tim Wahana Komputer, 2009:7).

### **3. Apache**

*Apache* merupakan *web server* yang paling banyak dipergunakan di *internet*. Program ini pertama kali didesain untuk sistem operasi lingkungan UNIX. Namun demikian, pada beberapa versi berikutnya *Apache* mengeluarkan yang dapat dijalankan dengan basis *Microsoft Windows NT*. *Apache* mempunyai program pendukung yang cukup

banyak. Hal ini, memberikan layanan yang cukup lengkap bagi penggunanya.

**Keuntungan *Web Server Apache* adalah sebagai berikut:**

1. *Apache* termasuk dalam katagori *freeware*.
2. Mampu beroperasi berbagai *platform* sistem operasi.
3. Kemudahan dalam mengatur konfigurasinya. *Apache* mempunyai hanya empat *file* konfigurasi.
4. Mudah dalam menambahkan *peripheral* lainnya ke dalam *platform web server* (Sugeng, 2010:201).

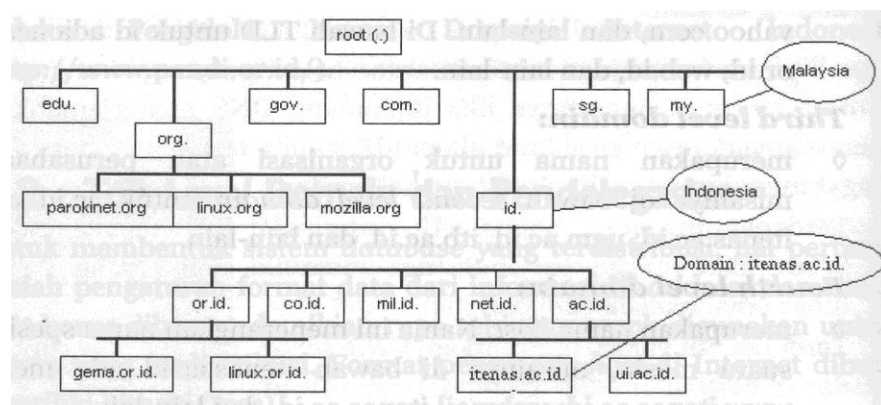
### **3.1.8. *Domain Name System (DNS)***

*Domain Name System* atau DNS adalah suatu teknik untuk mengingat *IP address* yang sulit diingat akibat terdiri dari sederetan angka. *Routing* paket *IP* yang berbasis TCP/IP sebenarnya tidak memerlukan teknik DNS tersebut, cukup dengan *IP address*. Teknik DNS diperlukan karena yang melakukan *routing* tidak lain adalah manusia, dan manusia pada umumnya lebih sulit menghafal deretan angka. Untuk itu, perlu cara lain agar manusia mudah menghafalnya atau mengingatnya. Manusia cenderung lebih mudah menghafal nama dibanding angka. Untuk itu agar *internet* lebih mudah digunakan, diperlukan suatu cara untuk memetakan dari *IP address* ke nama *host/komputer* dan sebaliknya, dan ini yang dinamakan DNS.

Jadi DNS dapat didefinisikan sebagai:

- a. Suatu sistem yang memungkinkan nama suatu *host* pada jaringan komputer atau *internet* ditranslasikan menjadi *IP address* atau sebaliknya.
- b. Merupakan *database* yang berisi daftar informasi *host*.
- c. Format penamaan *host* pada *internet* dibuat menurut hirarki berdasarkan level domain (Winarno 2010:143).

Pada umumnya manusia lebih sulit untuk menghafal deretan angka dibanding dengan nama, maka dibuatlah suatu cara untuk memetakan dari alamat IP. DNS diperlukan karena untuk memudahkan pengguna internet melakukan *routing*.



**Gambar 3.12.** Arsitektur hirarki DNS

Sumber: Sugeng (2010:143)

Menurut Sugeng (2010:143) DNS adalah suatu teknik untuk mengingat *IP address* yang sulit diingat akibat terdiri dari sederetan angka.

Format penamaan host pada internet dibuat menurut hirarki berdasarkan level domain, yang terdiri dari : *Root level domain, top level domain, second level domain, third level domain, fourth level domain.*

**a. Root Level Domain**

*Root level domain* merupakan tingkat teratas yang ditampilkan sebagai tanda titik (.).

**b. Top Level Domain**

TLD *Generic (Generic domain)* dibagi menjadi 7 jenis yang terdiri 3 huruf. TLD Negara (*Country domain*) untuk membedakan pemakai nama oleh suatu Negara dengan Negara lain digunakan tanda, misalnya untuk Indonesia adalah .id. TLD arpa untuk fasilitas *reverse mapping*, atau proses memetakan IP *address ke nama domain.*

**c. Second Level Domain**

Merupakan nama untuk organisasi atau perusahaan, misalnya di bawah TLD untuk com ada microsoft.com, yahoo.com, dan lain-lain. Di bawah TLD untuk id ada ac.id, or.id, web.id, dan lain-lain.

**d. Third Level Domain**

Merupakan nama untuk, organisasi atau perusahaan, misalnya dibawah *second level domain* untuk ac.id ada palcomtech.ac.id, itenas.ac.id, itb.ac.id dan lain-lain.

#### **e. Fourth Level Domain**

Merupakan nama *host*. Nama ini menerangkan nama spesifik suatu mesin, misalnya di bawah *kopertis2.or.id* ada mesin *www.kopertis2.or.id*, *webmail.kopertis2.or.id*, *ftp.kopertis2.or.id* dan lain-lain.

#### **3.1.9. Apache Benchmark**

*Apache Benchmark* adalah alat untuk mengukur kinerja *Apache server* terhadap *Hypertext Transfer Protocol (HTTP)*. Alat ini di desain untuk memberikan hasil tentang bagaimana instalasi *Apache* saat dilakukan dengan cara mensimulasikan kunjungan (*visitor*) dan melihat respon *Apache*.

#### **Keuntungan Apache Benchmark**

- Mengoptimalkan *hardware* yg dimiliki dengan menggunakan konfigurasi yg optimal.
- Mengetahui kelemahan *software* yg digunakan (penggunaan memori dan *cache system*)

(Sumber : <http://httpd.apache.org/docs/2.0/programs/ab.html> diakses 20 Juni 2012).

### 3.1.10. Apache Httperf

*Apache Httperf* adalah alat untuk mengukur kinerja *web server*, yang menyediakan fasilitas *fleksibel* untuk menghasilkan berbagai beban kerja HTTP untuk mengukur kinerja *server*. Fokus *httperf* tidak pada implementasi salah satu patokan tertentu, tetapi pada penyediaan, alat untuk mengukur performa yang tinggi, memfasilitasi pembangunan baik mikro dan makro tingkat *benchmark*. Tiga karakteristik yang membedakan dari *httperf* adalah ketahanan, yang mencakup kemampuan untuk menghasilkan dan mempertahankan *server overload*, dukungan untuk HTTP/1.1 dan protokol SSL, dan diperpanjang untuk generator beban kerja baru dan pengukuran kinerja.

#### **Keuntungan Apache Httperf**

- *Httperf* program untuk mengukur kinerja dari *Web server*
- *Httperf* menyediakan fitur yang fleksibel dalam pembuatan beban kerja pada *web/http server*
- *Httperf* dapat mensimulasikan berbagai macam beban kerja sesuai dengan parameter yang diberikan kepadanya
- Program ini diperlukan untuk mensimulasikan beban kerja pada skenario Aplikasi *Web*

(Sumber :<http://www.hpl.hp.com/research/linux/httperf> diakses 20 Juni 2012).



### 3.1.11. Sejarah Linux

Awalnya kernel linux ditulis sebagai hobi oleh pelajar universitas Finland Linus Torvalds yang belajar di Universitas Helsinki, untuk membuat kernel Minix yang bebas dan dapat sunting. (Minix adalah projek pelajaran menyerupai UNIX direka untuk mudah dan bukannya untuk kegunaan perniagaan.) Versi 0.01 dikeluarkan ke Internet pada September 1991, kemudian versi 0.02 pada tanggal 5 Oktober 1991.

Sejarah sistem pengoperasian berdasarkan linux berkaitan erat dengan projek GNU, projek program bebas terkenal diketuai oleh Richard Stallman. Projek GNU bermula pada 1983 untuk membuat sistem pengoperasian seperti Unix lengkap- kompilar, utiliti aplikasi, utiliti pembangunan dan seterusnya — dikarang sepenuhnya dengan Program Bebas. Pada 1991, jika versi pertama kerangka Linux ditulis, projek GNU project telah menghasilkan hampir kesemua komponen sistem ini — kecuali kernel. Torvalds dan pembangun kernel seperti linux menyesuaikan kernel mereka supaya dapat berfungsi dengan komponen GNU, dan seterusnya mengeluarkan Sistem Pengoperasian yang cukup berfungsi. Oleh karena itu, Linux melengkapi ruang terakhir dalam rancangan GNU (Sumber: <http://opensource-indonesia.com> diakses tanggal 20 Mei 2012).

### 3.1.11.1. Debian Linux

#### a. Pengertian Debian

Menurut Yani (2005:8) Debian adalah distribusi yang mengutamakan kestabilan dan keandalan meskipun mengorbankan aspek kemudahan dan kemutakhiran program. Debian menggunakan .deb dalam paket instalasi programnya

#### b. Kelebihan Debian

1. Debian adalah distribusi yang mengutamakan kestabilan dan keandalan.
2. Debian adalah sistem operasi yang *free*, legal dan dapat dikembangkan dengan paket-paket pendukungnya.
3. Debian dapat berjalan pada beberapa processor, sampai *release* terbaru, dapat berjalan pada 11 arsitektur processor.
4. Debian memiliki kernel independent yang dapat mendukung banyak kernel, sehingga jika suatu saat kernel linux tidak *free* lagi, maka debian dapat menggunakan kernel lain yang masih *free*.
5. Keunikan debian adalah aturan *release* distro debian, jika ada versi baru pada debian, versi lama tetap dapat diakses.

### 3.2. Hasil Penelitian Terdahulu

Hasil penelitian terdahulu adalah penelitian yang pernah dilakukan oleh penulis lain yang berhubungan dengan penelitian yang akan penulis lakukan atau disebut dengan jurnal, terdapat dua hasil penelitian terdahulu yang berhubungan dengan skripsi penulis yaitu jurnal pertama dan jurnal kedua.

**Tabel 3.1** Hasil Penelitian Terdahulu Pertama

Judul Penelitian	Analisis Perbandingan <i>Load Balancing Web Server</i> Tunggal dengan <i>Web Server Cluster</i> Menggunakan <i>Linux Virtual Server</i> (2010).
Penulis	Ahmad Fali Oklilas, Desy Lukitasari Universitas Sriwijaya, Palembang.
ABSTRAKSI	<i>Virtual server</i> adalah <i>server</i> yang mempunyai skalabilitas dan ketersediaan yang tinggi yang dibangun diatas sebuah <i>cluster</i> dari beberapa <i>real server</i> . <i>Real server</i> dan <i>load balancer</i> akan saling terkoneksi baik dalam jaringan local kecepatan tinggi atau yang terpisah secara geografis. <i>Load balancer</i> dapat mengirim permintaan – permintaan ke <i>server</i> yang berbeda dan membuat <i>parallel service</i> dari sebuah <i>cluster</i> pada sebuah alamat <i>IP</i> tunggal dan meminta pengiriman dapat menggunakan teknologi

	<p><i>load balancing</i>. <i>Linux Virtual Server</i> juga mampu pemeratakan beban <i>request</i> / paket dari <i>client</i> kepada beberapa <i>Real-Server</i> yang telah ter – <i>cluster</i> dan terintegrasi pada sebuah <i>Virtual – Service</i>.</p> <p><b>Kata Kunci :</b> <i>Load balancing</i>, <i>Linux Virtual Service</i>, <i>Web Service</i>, <i>Cluster Service</i>.</p>
--	--

**Tabel 3.2** Hasil Penelitian Terdahulu Kedua

Judul Penelitian	Kinerja Pemodelan Sebuah <i>Web Server Apache</i> dengan bursty (2003).
Penulis	Mikael Andersson, Jianhua Cao, Maria Kihl dan Kristen  Departemen Komunikasi Sistem, Institut Teknologi Lund.
ABSTRAKSI	Pemodelan Kinerja merupakan topik penting dalam kapasitas perencanaan dan pengendalian kelebihan untuk <i>server web</i> . Menyajikan suatu model dari sebuah <i>web server Apache</i> yang menggunakan bursty. Adanya permintaan HTTP adalah diasumsikan sebagai Markov Modulated Poisson Proses dan disiplin pelayanan <i>server</i> berbagi prosesor. Itu jumlah permintaan yang dapat diproses

	<p>pada satu waktu adalah terbatas pada K. Kemudian metrik <i>web server</i> kinerja seperti waktu respon rata-rata, <i>throughput</i> dan memblokir probabilitas dengan simulasi. Dibandingkan dengan model lain, model konseptual sederhana. Model telah divalidasi melalui pengukuran dan simulasi di lab. Kinerja metrik diprediksi oleh model sesuai dengan hasil</p> <p><b>Kata Kunci</b> : <i>Internet, World Wide Web, web server, kinerja model, MMPP.</i></p>
--	---



## BAB IV

### METODE PENELITIAN

#### 4.1. Lokasi dan Waktu Penelitian

Lokasi dan waktu penelitian yang dilakukan penulis pada penelitian di Kopertis Wilayah II Palembang.

##### 4.1.1. Lokasi Penelitian

Penelitian dilakukan pada Kopertis Wilayah II Palembang yang berada di jalan Srijaya No. 883 Palembang 30153, Telp (0711) 410722- 410423 Fax (0711) 419421.

##### 4.1.2. Waktu Penelitian

Waktu penelitian berjalan selama 4 (empat) bulan, mulai dari tanggal 30 Maret sampai dengan tanggal 30 Juli 2012.

**Tabel 4.1** Waktu Penelitian

No	Tahapan	Bulan ke 1				Bulan ke 2				Bulan ke 3				Bulan ke 4			
		Minggu ke				Minggu ke				Minggu ke				Minggu ke			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	Analisis	■	■														
2.	Desain			■													
3.	Simulasi <i>Prototype</i>				■	■	■	■									





#### 4.2.2. Data Sekunder

Menurut Kuncoro (2009:148), data Sekunder adalah data yang telah dikumpulkan oleh lembaga pengumpul data dan dipublikasikan kepada masyarakat pengguna data.

Data sekunder dalam penulisan skripsi ini didapat oleh penulis dari Kopertis Wilayah II Palembang berupa sejarah singkat, visi dan misi, serta struktur organisasi Kopertis Wilayah II Palembang.

#### 4.3. Teknik Pengumpulan Data

Adapun metode pengumpulan data yang digunakan dalam penyusunan tugas laporan ini, yaitu :

##### a. Metode Pengamatan (*Observasi*)

Menurut Made (2006:37), *observasi* adalah metode pengumpulan data dimana penulis laporan atau kolaboratornya mencatat informasi sebagaimana yang mereka saksikan selama disana.

Metode ini dilakukan penulis dengan cara mengamati bagaimana metode yang digunakan pada *web server* di Kopertis Wilayah II Palembang.

##### b. Metode Wawancara (*interview*)

Menurut Made (2006:37), wawancara adalah proses memperoleh keterangan untuk tujuan membuat laporan dengan cara tanya jawab dan tatap muka antara sipenanya atau pewawancara dengan sipenjawab atau *responder*.

Metode wawancara ini dilakukan penulis tanya jawab langsung dengan kepala bagian umum dan koordinator IT yaitu bapak Joko Purnomo, S.Kom. pada Kopertis Wilayah II Palembang, mengenai metode *web server* pada Kopertis Wilayah II Palembang .

### c. Studi Pustaka

Studi pustaka adalah pengumpulan data dengan cara membaca dan mengutip teori-teori yang berasal dari buku dan tulisan-tulisan lain yang relevan dengan laporan ini, serta beberapa artikel yang diperoleh dari situs-situs yang menjelaskan tentang *web server*.

## 4.4. Jenis Penelitian

Penulis menggunakan jenis penelitian Deskriptif, karena penelitian Deskriptif meliputi pengumpulan data untuk diuji hipotesis atau menjawab pertanyaan mengenai status terakhir dari subjek penelitian (Kuncoro, 2009:12).

Penulis mengumpulkan data mengenai metode *web server* yang ada di Kopertis, mendesain topologi jaringan di Kopertis Wilayah II Palembang, selanjutnya menginstall sistem operasi linux debian dan melakukan konfigurasi standar *web server* yang akan menjadi basis instalasi drupal, sampai tahap ini akan dilakukan pengukuran respon *web server* terhadap simulasi pengunjung. Langkah selanjutnya melakukan konfigurasi tambahan pada *web server* kemudin di *test* kembali dengan simulasi pengunjung. Hasil

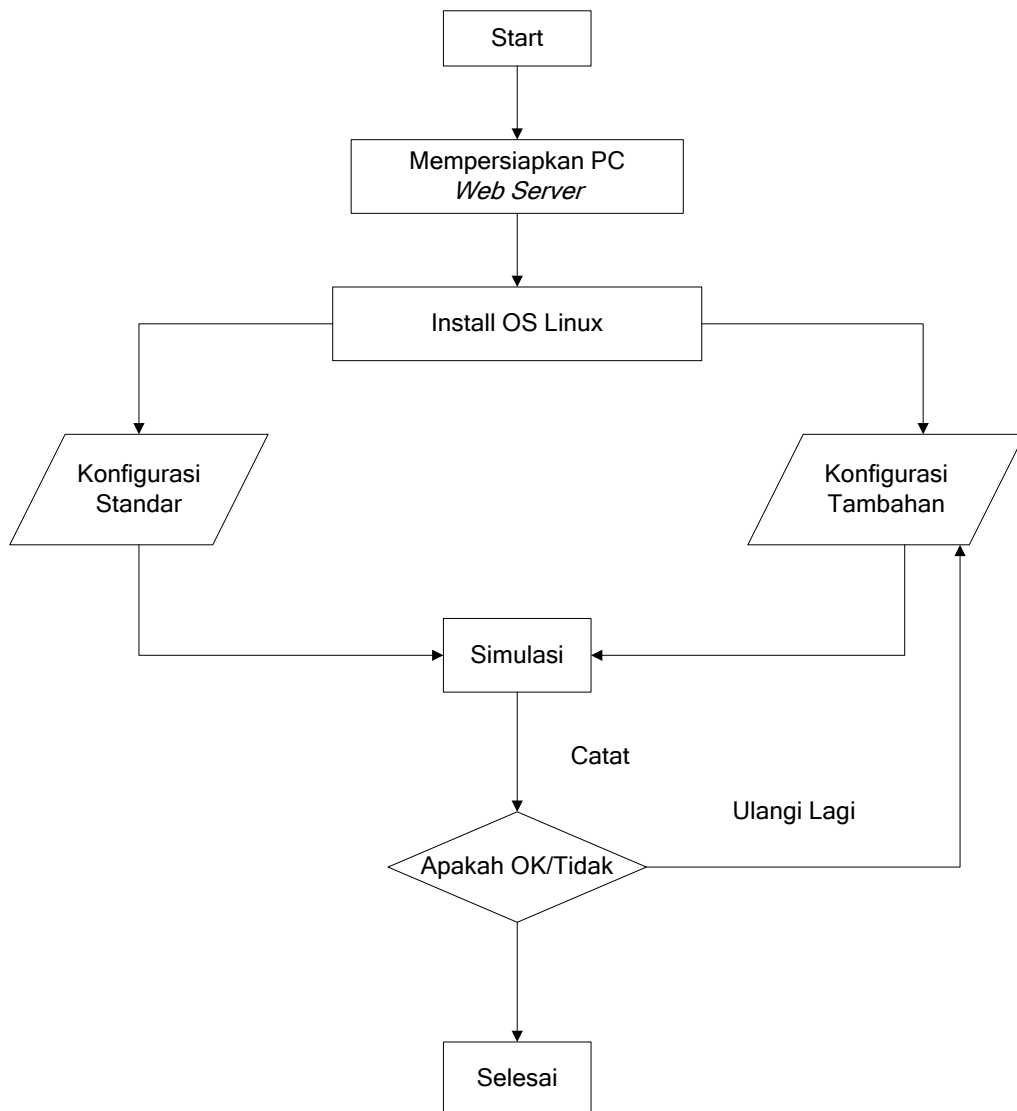
simulasi awal dan simulasi akhir akan dibandingkan dan disajikan dalam bentuk table dan grafik.

#### **4.5. Alur dan Teknik Pengembangan Sistem**

Alur dan teknik pengembangan sistem terdiri dari alur proses dan teknik pengembangan *web server*.

##### **4.5.1. Alur Proses**

Model Alur Proses menggunakan *Flowchart*, yang menggambarkan alur pengembangan *web server*. *Flowchart* menggambarkan secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. *Flowchart* membantu analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian



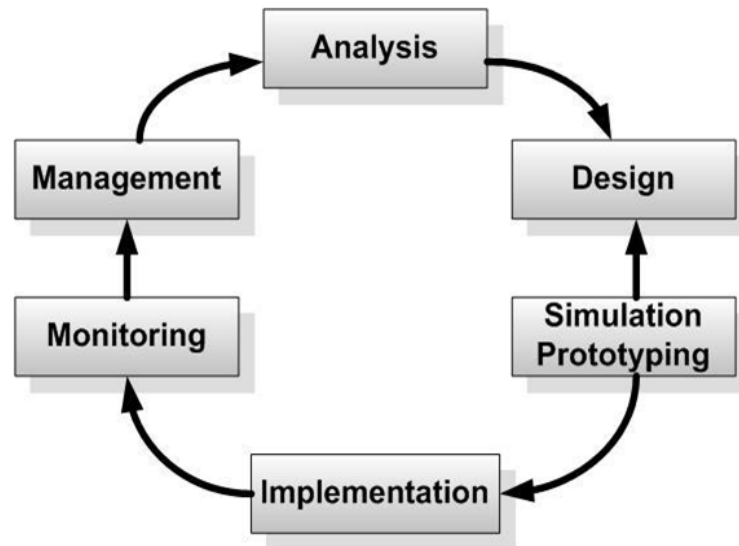
**Gambar 4.1.** *Flowchart* model alur proses

Sumber : (Diolah Sendiri)

Alur proses *flowchart* desain *web server* dimulai dari mempersiapkan *PC web server*, kemudian install OS Linux, lalu lakukan konfigurasi standar dan konfigurasi tambahan, kemudian lakukan simulasi, jika ok selesai, tetapi jika tidak ulangi lagi konfigurasi tambahan hingga selesai.

#### 4.5.2. Teknik Pengembangan Sistem

Teknik pengembangan sistem yang penulis gunakan yaitu NDLC



**Gambar 4.2.** Tahapan pada *Network Development Life Cycle*

Sumber : (Diolah Sendiri)

##### 1. Analisis

Tahap awal ini dilakukan desain dan mengoptimalkan performa *web server* berbasis *open source* pada kopertis wilayah II Palembang, analisa kebutuhan, analisa permasalahan yang muncul, analisa keinginan user, dan analisa topologi atau jaringan yang sudah ada saat ini. Metode yang bisa digunakan pada tahap ini diantaranya :

### **a. Wawancara**

Wawancara dilakukan dengan pihak terkait melibatkan dari struktur manajemen atas sampai ke level bawah atau operator agar mendapatkan data yang konkrit dan lengkap.

Tahap wawancara ini dilakukan penulis tanya jawab langsung dengan kepala bagian umum dan koordinator IT yaitu bapak Joko Purnomo, S.Kom. pada Kopertis Wilayah II Palembang, mengenai metode *web server* pada Kopertis Wilayah II Palembang.

### **b. Survey Langsung Kelapangan**

Pada tahap analisis juga biasanya dilakukan survey langsung kelapangan untuk mendapatkan hasil sesungguhnya dan gambaran seutuhnya sebelum masuk ke tahap desain, survey biasa dilengkapi dengan alat ukur atau metode yang sesuai kebutuhan untuk mengetahui detail yang dilakukan.

Tahap ini dilakukan penulis dengan cara mengamati secara langsung bagaimana metode yang digunakan pada *web server* dan mengambil data yang dibutuhkan di Kopertis Wilayah II Palembang.

### **c. Membaca Manual atau Blueprint Dokumentasi**

Analisis awal ini juga dilakukan dengan mencari informasi dari manual-manual atau blueprint dokumentasi yang mungkin pernah dibuat sebelumnya. Sudah menjadi keharusan dalam setiap pengembangan suatu sistem dokumentasi menjadi pendukung akhir dari pengembangan tersebut, begitu juga pada project *network*, dokumentasi menjadi syarat mutlak setelah sistem selesai dibangun.

Tahap ini dilakukan penulis dengan cara mengambil dokumentasi langsung peralatan yang digunakan, seperti *PC*, *CPU*, *server*, *switch* dan ruangan tempat alat diletakkan di Kopertis Wilayah II Palembang.

### **d. Menelaah Setiap Data yang Didapat dari Data-data Sebelumnya**

Maka perlu dilakukan analisa data tersebut untuk masuk ke tahap berikutnya. Adapun yang bisa menjadi pedoman dalam mencari data pada tahap analisis ini adalah ;

#### **1. *User***

Jumlah *user*, kegiatan yang sering dilakukan, level teknis *user*.

## 2. Media *Hardware* dan *Software*

Peralatan yang ada, status jaringan, ketersediaan data yang dapat diakses dari peralatan, aplikasi *software* yang digunakan

### a. Data :

Jumlah pelanggan atau pegawai , jumlah inventaris sistem, sistem keamanan yang sudah ada dalam mengamankan data, media *web server* yang digunakan.

### b. *Network* :

Konfigurasi jaringan, protocol, *monitoring network* yang ada saat ini, harapan dan rencana pengembangan kedepan.

### c. Perencanaan fisik :

Masalah tata letak, ruang khusus *hardware*, sistem keamanan yang ada, dan kemungkinan akan pengembangan kedepan.

## 2. Desain

Berdasarkan data-data yang didapatkan sebelumnya, tahap Desain ini akan membuat gambar desain topologi jaringan interkoneksi yang akan dibangun, diharapkan dengan gambar ini akan memberikan gambaran seutuhnya dari kebutuhan yang ada.



Desain ini dilakukan penulis dengan cara mendesain topologi yang digunakan, yaitu topologi *star*.

### **3. Simulasi *Prototype***

Tahap ini dilakukan penulis dengan cara mempersiapkan *Software* dan paket yang dibutuhkan. Sistem yang dipersiapkan adalah LAMPP (Linux Debian 6, *Apache*, *Mysql*, *PHP*), yang dikombinasikan dengan drupal, AWStats.

### **4. Implementasi**

Tahap ini dilakukan penulis dengan cara mengimplementasikan desain *webserver* berbasis *open source* pada Kopertis Wilayah II Palembang.

### **5. Monitoring**

Setelah implementasi tahapan *monitoring* merupakan tahapan yang penting, agar jaringan komputer dan komunikasi dapat berjalan sesuai dengan keinginan dan tujuan awal dari user pada tahap awal analisis, maka perlu dilakukan kegiatan *monitoring*. *Monitoring* bisa berupa melakukan pengamatan pada ;

- a. Infrastruktur *hardware* : dengan mengamati kondisi *reliability* atau kehandalan sistem yang telah dibangun.

- b. Memperhatikan jalannya paket data di jaringan ( *pewaktuan, latency, peektime,throughput*)

Tahap *monitoring* dilakukan penulis dengan menggunakan AWStats untuk *monitoring web server*. AWStats adalah singkatan dari *Advanced Web Statistics*. Keunggulan AWStats yaitu AWStats sangat *powerful* untuk *log analyzer* yang bisa digunakan untuk *memonitoring web, ftp, mail* dan *streaming server statistics*. Laporan disajikan dalam bentuk grafis dan statistik sehingga mudah untuk dibaca. AWStats dapat dijalankan melalui *browser web CGI (CommonGateway Interface)* atau langsung dari *command line*. AWStats ini dapat dengan cepat menganalisa log file yang besar. AWStats dapat menganalisa file log dari *Apache* dan sebagian besar *WebStar* web, proxy, wap dan *streaming media server* serta *server ftp* dan *mail log* .

## 6. Manajemen

Manajemen atau pengaturan, salah satu yang menjadi perhatian khusus adalah masalah *policy*, kebijakan perlu dibuat untuk membuat atau mengatur agar sistem yang telah dibangun berjalan dengan baik dapat berlangsung lama dan unsur *reliability* terjaga. *Policy* akan sangat tergantung dengan kebijakan level *management*. IT sebisa mungkin harus dapat mendukung atau

*alignment* dengan strategi jangka panjang Kopertis Wilayah II Palembang.

Tahap ini dilakukan penulis dengan cara mengatur atau manajemen desain *webservice* berbasis *open source* pada Kopertis Wilayah II Palembang.

## BAB V

### HASIL DAN PEMBAHASAN

#### 5.1. Analisis

Tahap awal analisis dengan metode NDLC, penulis melakukan analisa kebutuhan, analisa permasalahan yang muncul, dan analisa keinginan user.

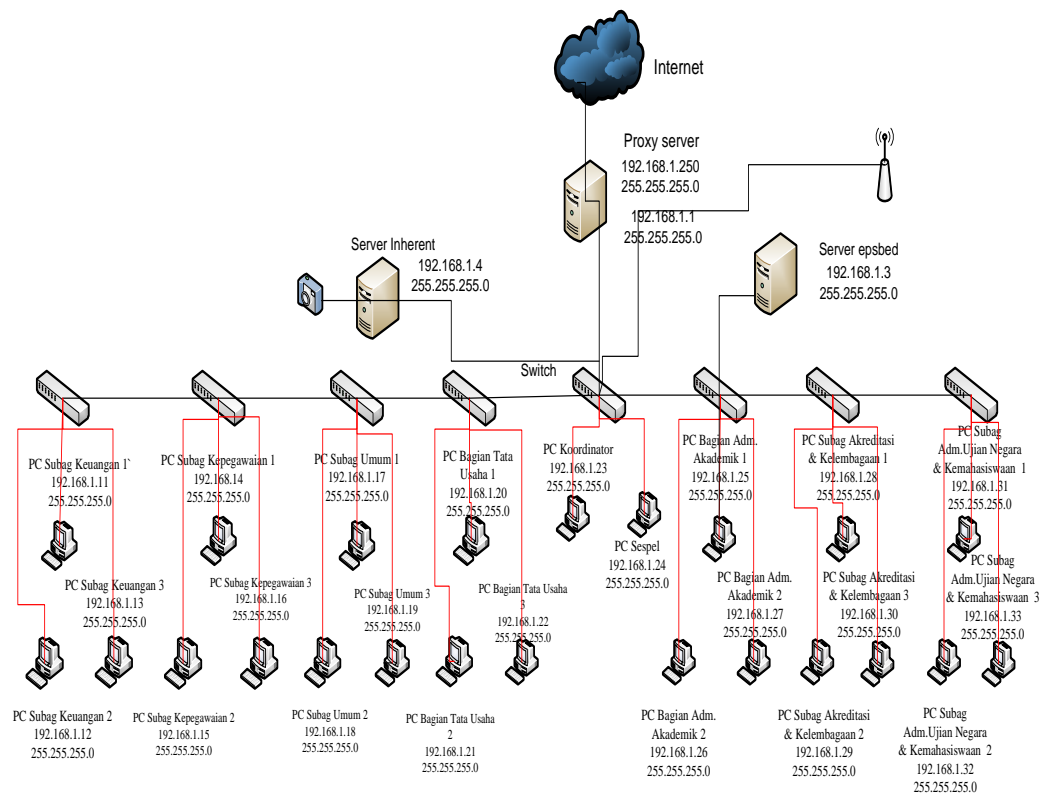
Langkah pertama yang penulis lakukan yaitu menganalisa kebutuhan akan sebuah *web server*. Fungsi dari *website* Kopertis Wilayah II Palembang sangat besar, untuk bisa menangani permintaan atau *request* pada *website* Kopertis Wilayah II Palembang, maka dibutuhkan *web server* yang handal. Penulis telah melakukan penelitian pada Kopertis Wilayah II Palembang dengan fokus pada *web server*, di Kopertis Wilayah II Palembang belum terdapat *web server*, *website* yang terdapat di Kopertis Wilayah II Palembang menggunakan jasa *hosting* pihak lain dan tidak optimal dalam menangani permintaan akses dari *client*. Disisi lain Kopertis Wilayah II Palembang memiliki infrastuktur jaringan dan perangkat keras *server* yang memadai, namun tidak digunakan dengan maksimal.

Masalah yang sering dihadapi adalah saat *client* melakukan akses secara bersamaan waktu *loading* yang diperlukan menjadi lebih lama sehingga *client* harus menunggu sampai beberapa menit untuk bisa melakukan akses ke halaman *website*, hal tersebut tentu saja membuat *client* yang sedang melakukan akses merasa tidak nyaman, karena akses *internet*

yang lama. Oleh karena itu untuk mengatasi masalah tersebut diperlukan sebuah *web server* dengan performa yang handal sehingga masalah tersebut dapat diminimalisasi.

## 5.2. Desain

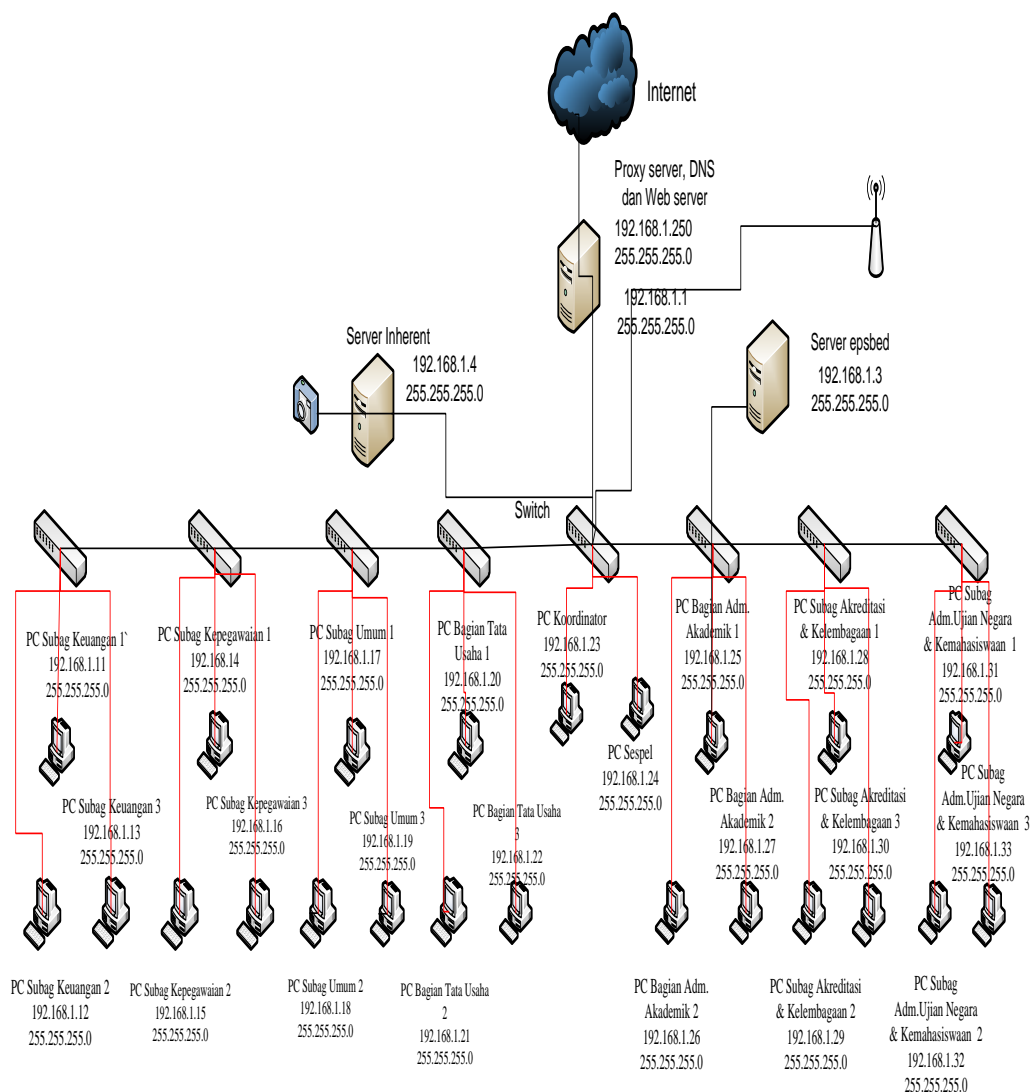
Tahap desain ini akan membuat gambar desain topologi jaringan yang sudah dimodifikasi berdasarkan pada topologi jaringan yang sudah ada. Dari hasil pengamatan yang telah dilakukan, maka topologi jaringan pada Kopertis Wilayah II Palembang dapat digambarkan secara sederhana seperti gambar dibawah ini.



**Gambar 5.1** Topologi Jaringan Kopertis Wilayah II Palembang

Sumber : Kopertis Wilayah II Palembang

Topologi yang berada di Kopertis Wilayah II Palembang yaitu topologi *Star*, yang menggunakan jaringan *internet* dengan modem ADSL, disana terdapat *Proxy server*, *server internet* dan *server epsbet*, menggunakan *switch* D-link 8 port dan diseluruh ruangan jumlah PC keseluruhan yaitu dua puluh tiga PC.



**Gambar 5.2** Topologi yang Dikembangkan

Sumber : Kopertis Wilayah II Palembang

Topologi yang saya gunakan yaitu topologi *Star*, ditambahkan *Proxy server*, *DNS* dan *Web Server*, terdapat *server internet* dan *server epsbet*, menggunakan *switch D-link 8 port* dan diseluruh ruangan jumlah *PC* keseluruhan yaitu dua puluh tiga *PC*.

### 5.3. *Simulation Prototype*

Tahap ini dilakukan penulis dengan cara mempersiapkan *Software* dan paket yang dibutuhkan. Sistem yang dipersiapkan adalah LAMPP (*Linux Debian 6, Apache, Mysql, PHP*), yang dikombinasikan dengan *drupal*, *AWStats*.

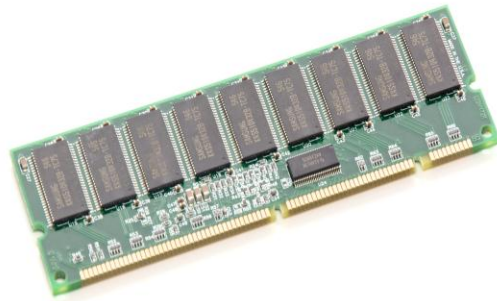
Teknologi jaringan yang digunakan pada desain dan optimalisasi *web server* pada *Kopertis Wilayah II Palembang* sederhana. Di optimalisasi dan disain *web server* tersebut terdapat konfigurasi *hardware* yang digunakan sebagai berikut:

1. Prosesor *Intell pentium 4*.



**Gambar 5.3** *Prosesor Intel Pentium 4*

2. Memori berkapasitas 512 Mb.



**Gambar 5.4** Memori 512 Mb

3. *Hardisk* Sata 80 Gb.



**Gambar 5.5** *Hardisk* Sata 80 Gb

4. Kabel *LAN*.



**Gambar 5.6** Kabel *LAN*



5. Switch D-link 8 port dan Modem Aztech DSL605E.



**Gambar 5.7** Switch D-link dan Modem Aztech

Kemudian konfigurasi *software* yang digunakan sebagai berikut:

1. Sistem Operasi Linux Debian 6 (squeeze).
2. *Apache 2*.
3. *PHP 5.3.3.-7 squeeze 13*.
4. *MySql 5.1.61-0 squeeze1*
5. *Drupal 7.14*
6. *PHP Xcache*
7. *PHP Memcached*
8. *Awstats*
9. *Apache Benchmark 2.3*
10. *Apache Httpperf*.

#### **5.4. Implementasi**

Pada tahapan implementasi ini, dilakukan pembuatan mesin *server* yang spesifikasinya telah di sebutkan pada bagian simulasi *prototype*.

Berdasarkan pengamatan yang didapat pada Kopertis Wilayah II Palembang. Penulis mengoptimalkan dan mendesain *web server* dengan menggunakan sistem operasi *linux Debian 6*, dengan cara simulasi kemudian menerapkan desain dan konfigurasi *web server* berbasis *open source* dan akan dipertimbangkan untuk digunakan pada Kopertis Wilayah II Palembang.

Hal pertama yang penulis lakukan adalah mengkonfigurasi DNS, kemudian *install MySQL*, *install drupal*, dan lakukan *benchmark*, *httpperf* dan selanjutnya melakukan *monitoring* dengan *Awstats*.

#### 5.4.1. Konfigurasi DNS

Untuk memulai instalasi DNS *server*, *file – file* yang akan digunakan agar DNS dapat bekerja dengan baik adalah sebagai berikut:

Langkah awal terlebih dahulu *install bind9*, dengan perintah :

```
# apt-get install bind9
```

*Change directory*, masuk ke direktori */etc/bind*, dengan mengetik perintah :

```
# cd /etc/bind
```

Mengedit *file* *named.conf*, yang berguna untuk mendefinisikan nama *domain* yang digunakan pada simulasi, dengan perintah :

```
# nano named.conf.local
```

Muncul tampilan di bawah ini, dan tambahkan *script* berikut:

```

GNU nano 2.2.4      File: named.conf.local      Modified
//
// Do any local configuration here
//
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "kopertis2.or.id" {
type master;
file "/etc/bind/db.febriyanda.net";
};

zone "1.168.192.in-addr.arpa. " {
type master;
file "/etc/bind/db.192.168.1";
};_
^G Get Help   ^O WriteOut  ^R Read File  ^V Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit       ^J Justify   ^W Where Is  ^N Next Page  ^U UnCut Text ^T To Spell

```

**Gambar 5.8** Mengisi *File Local*

*Copy file* db.local dan db.127 menjadi nama sendiri dan IP *class server*, dengan perintah :

cp db.local db.kopertis dan cp db.127 db.192.168.1

*Edit script* di *file zone domain*, yang mendefinisikan letak *file file* yang berisi *domain* yang digunakan, dengan melakukan perintah :

**# nano db.febriyanda.net**

```

GNU nano 2.2.4      File: db.febriyanda.net      Modified
;
; BIND data file for local loopback interface
;
$TTL 604800
kopertis2.or.id.      IN      SOA      www.kopertis2.or.id. root.kopertis2.or.$
                        2          ; serial
                        604800     ; Refresh
                        86400     ; Retry
                        2419200    ; Expire
                        604800 )   ; Negative Cache TTL
;
kopertis2.or.id.      IN      NS       www.kopertis2.or.id.
www.kopertis2.or.id. IN      A        192.168.1.250
mail.kopertis2.or.id. IN     A        192.168.1.250

```

<sup>^</sup>G Get Help    <sup>^</sup>O WriteOut    <sup>^</sup>R Read File    <sup>^</sup>Y Prev Page    <sup>^</sup>K Cut Text    <sup>^</sup>C Cur Pos  
<sup>^</sup>X Exit        <sup>^</sup>J Justify    <sup>^</sup>W Where Is    <sup>^</sup>V Next Page    <sup>^</sup>U UnCut Text <sup>^</sup>T To Spell

**Gambar 5.9** Mengisi *File Zone Domain*

Simpan hasil dengan menekan tombol kombinasi di *keyboard* Ctrl+O untuk menyimpan dan Ctrl+x untuk keluar dari halaman.

*Reverse address* untuk verifikasi apakah benar *IP* terasosiasi ke *domain* dan mengedit *file*, dengan perintah :

**# nano db.192.168.1**

```

GNU nano 2.2.4          File: db.192.168.1          Modified
;
; BIND reverse data file for local loopback interface
;
$TTL 604800
1.168.192.in-addr.arpa. IN      SOA      www.kopertis2.or.id. root.kopertis2.or.$
                        1          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 ) ; Negative Cache TTL
;
1.168.192.in-addr.arpa.      IN      NS       www.kopertis2.or.id.
250.1.168.192.in-addr.arpa. IN      PTR      www.kopertis2.or.id.

```

**Gambar 5.10** Mengisi *Reverse Address*

Penyetingan *IP address* menggunakan IP 192.168.1.250, dengan perintah :

**# nano /etc/network/interfaces**

```

GNU nano 2.2.4          File: /etc/network/interfaces          Mo
# The local network interface
auto eth0
iface eth0 inet static
    address 192.168.1.250
    netmask 255.255.255.0

auto eth1
iface eth1 inet static
    address 192.168.0.2
    netmask 255.255.255.0
    gateway 192.168.0.1

```

**Gambar 5.11** Mengisi *IP Address*

Test DNS , isi *name server* dengan IP 192.168.1.250

```
GNU nano 2.2.4 File: /etc/resolv.conf
search kopertis2.or.id
nameserver 192.168.1.250
```

**Gambar 5.12** Mengedit IP pada Name Server

Untuk keluar dari halaman, kemudian *restart ethernet* dengan perintah :

**#/etc/init.d/networking restart**

```
Running /etc/init.d/networking restart is deprecated because it may not enable a
gain some interfaces ... (warning).
Reconfiguring network interfaces...done.
```

**Gambar 5.13** Sebagian Hasil Me-restart Kartu Jaringan

Pengujian koneksi dengan perintah : `nslookup 192.168.1.250` dan `www.kopertis2.or.id`

```
root@kopertis2:/etc/bind# nslookup
-----
> 192.168.1.250
Server:         192.168.1.250
Address:        192.168.1.250#53

250.1.168.192.in-addr.arpa    name = www.kopertis2.or.id.
> www.kopertis2.or.id
Server:         192.168.1.250
Address:        192.168.1.250#53

Name:   www.kopertis2.or.id
Address: 192.168.1.250
>
```

**Gambar 5.14** Test Resolver DNS

### 5.4.2. *Install Apache mysql*

Sebelum melakukan install dan konfigurasi, periksa terlebih dahulu hostname, dengan cara :

```
# hostname
```

Memperbarui paket repositori dan program yang di-*install* yang *up to date* dengan mengeluarkan perintah berikut:

```
# apt-get update
```

```
# apt-get upgrade --show-upgraded
```

*Install web server apache*, dengan perintah :

```
# apt-get install apache2
```

Mengaktifkan modul *rewrite*, agar alamat yang dihasilkan *search engine friendly* dengan perintah:

```
# a2enmod rewrite
```

Untuk mengatur virtual *hosting*, agar dapat menampung banyak *domain* pada satu *IP address*, dengan perintah:

```
# nano /etc/apache2/site-available/kopertis2.or.id
```

```
GNU nano 2.2.4 File: /etc/apache2/sites-available/kopertis2.or.id
<VirtualHost *:80>
ServerAdmin webmaster@kopertis2.or.id
ServerName kopertis2.or.id
ServerAlias www.kopertis2.or.id
DocumentRoot /home/febriyanda/public_html
</VirtualHost>
```

**Gambar 5.15 Virtual Hosting**

Untuk menyimpan *file – file* Drupal maka folder *public\_html* diperlukan perintah untuk membuat folder tersebut sebagai berikut:

**# mkdir public\_html**

Perintah berikut ini berguna untuk menghubungkan *file host virtual* ke direktori *sites – available* sehingga dapat diakses dari *internet*, dengan menggunakan perintah:

**# a2ensite kopertis2.or.id**

Mengaktifkan, menonaktifkan, atau memodifikasi setiap bagian dari konfigurasi Apache, perlu memuat ulang konfigurasi Apache lagi dengan perintah:

**# /etc/init.d/apache2 reload**

Langkah selanjutnya *install* paket *mysql*, dengan perintah:

```
root@kopertis2:/etc/bind# apt-get install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

**Gambar 5.16 Install Mysql**



Lalu *install mysql\_secure\_installation*, yang berfungsi untuk membuat instalasi berjalan dengan lebih aman dengan cara menghapus *user* dan *database* yang tidak diperlukan, dengan perintah:

```
root@kopertis2:/etc/bind# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MySQL
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!
```

**Gambar 5.17** *Install Mysql\_secure\_installation*

Selanjutnya dengan menggunakan *debian 6* di *server*, untuk *Log in* ke *database* dan memberikan hak akses pengguna untuk menggunakan *database*, dengan perintah *login* ke MySQL:

```
root@kopertis2:/etc/bind# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 40
Server version: 5.1.61-0+squeeze1 (Debian)

Copyright (c) 2000, 2011, 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> _
```

**Gambar 5.18** Membuat database

```
mysql> create database kopertis2;
Query OK, 1 row affected (0.09 sec)

mysql> grant all on kopertis2.* to 'febriyanda' identified by '123456';
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye
```

**Gambar 5.19** Mengisi database

Agar Drupal bisa berjalan dengan baik maka diperlukan bahasa PHP yang diselipkan ke dalam *web server Apache* dalam bentuk modul. Berikut tampilan layar proses instalasi PHP pada Linux Debian 6, dengan perintah sebagai berikut:

```
# apt-get install php5 php-pear php5-suhosin
```

```
root@kopertis2:/home# apt-get install php5 php-pear php5-suhosin
Reading package lists... Done
Building dependency tree
Reading state information... Done
php5-suhosin is already the newest version.
php-pear is already the newest version.
php5 is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 40 not upgraded.
root@kopertis2:/home# _
```

**Gambar 5.20** Menginstall php5-suhosin

# **nano /etc/php5/apache2/php.ini** untuk mengubah *file PHP* dan sesuaikan isi didalam nya dengan format yang sudah ditentukan, dengan waktu maksimal 30 m/s, dengan limit *memory* 64M.

```
max_execution_time = 30
memory_limit = 64M
error_reporting = E_COMPILE_ERROR|E_RECOVERABLE_ERROR|E_ERROR|E_CORE_ERROR
display_errors = Off
log_errors = On
error_log = /var/log/php.log
register_globals = Off
```

**Gambar 5.21** Modifikasi konfigurasi

Menginstal paket `php5-mysql` untuk mendukung MySQL di PHP, lakukan perintah berikut:

```

root@kopertis2:/home# apt-get install php5-mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
php5-mysql is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 40 not upgraded.

```

**Gambar 5.22** Menginstal paket php5-mysql

Setelah melakukan perubahan pada file konfigurasi PHP lakukan *restart Apache*, dengan perintah:

```
# /etc/init.d/apache2 restart
```

### 5.4.3. *Install Drupal*

Langkah pertama yaitu *download* terlebih dahulu drupal 7.14, kemudian lakukan *extract* drupal pada *server* dengan perintah :

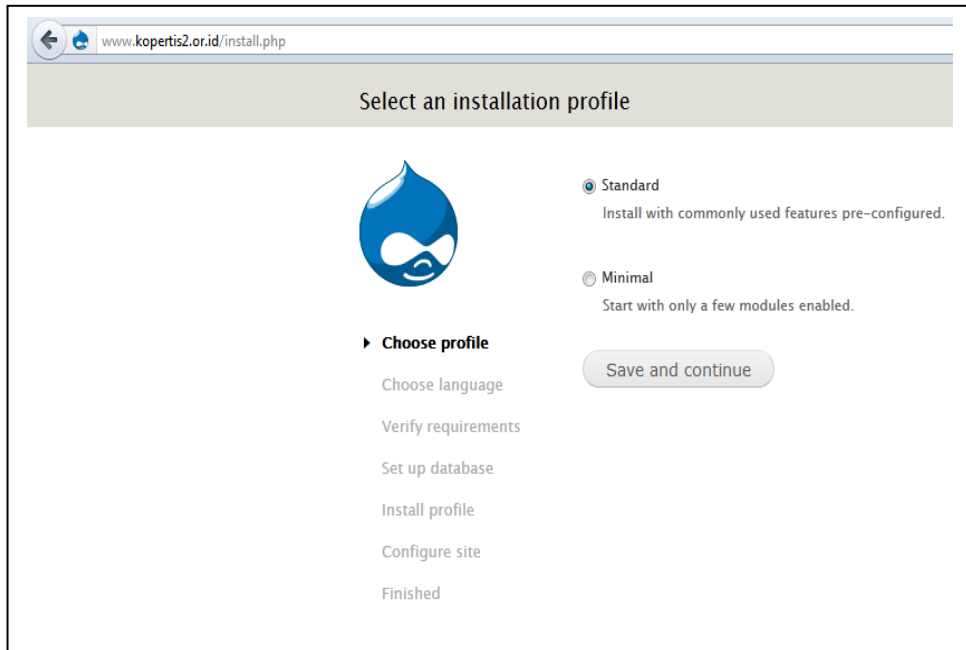
```
# tar xvzf drupal-7.14.tar.gz
```

Lalu *copy* setting.php, dengan perintah:

```
# cp default.settings.php settings.php
```

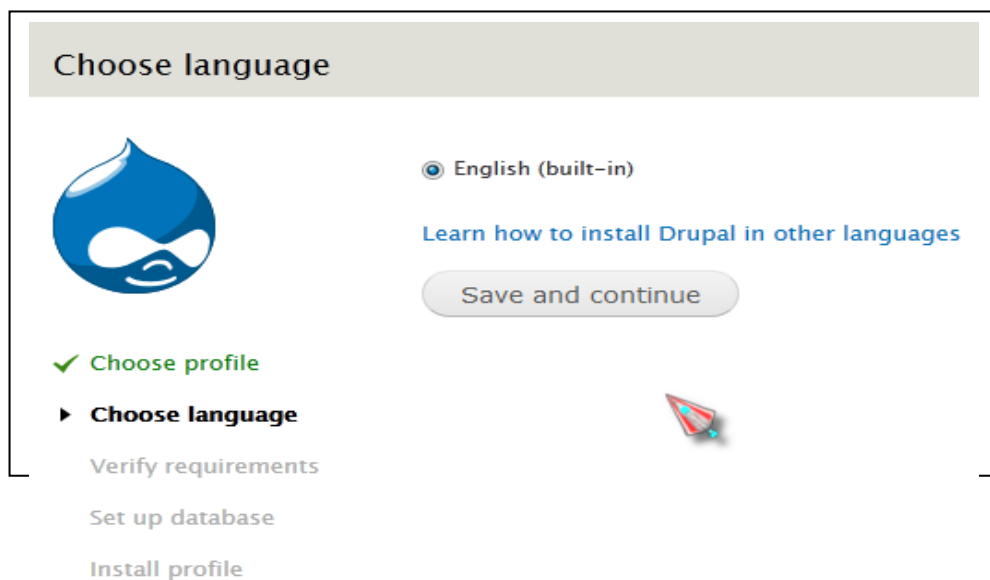
Setelah meng-*copy* perintah di atas, penulis memastikan *IP* *setting* 192.168.1.250

Langkah selanjutnya yaitu membuka halaman *browser* pada *client* dengan ketik : **www.kopertis2.or.id** kemudian pilih **save and continue**



**Gambar 5.23** Memilih Profil di Drupal


Langkah selanjutnya pilih bahasa yang ingin digunakan, pilih *english* lalu *save and continue*.



Kemudian lakukan *verify requirements* dengan cara memilih *database type*, lalu mengisi data di *database name* yaitu *kopertis2*, *database username* yaitu *febriyanda*, *database password* yaitu *1234567*, setelah selesai mengisi pilih *save and continue*.

**Gambar 5.25** *Verify Requirements*

Langkah selanjutnya lakukan *set up database* dengan mengisi *site name* yaitu *www.kopertis2.or.id*, lalu *site e-mail address* yaitu *gumay\_n@yahoo.com*, *user name* yaitu *febriyanda*, kemudian *password* *1234567*, *default country* pilih *Indonesia*, lalu *default time zone* pilih waktu saat men-*setting drupal* dan *update notifications* diaktifkan, kemudian pilih *save and continue*.



- ✓ Choose profile
- ✓ Choose language
- ✓ Verify requirements
- ✓ Set up database
- ✓ Install profile
- ▶ **Configure site**

Finished

#### SITE INFORMATION

**Site name \***

**Site e-mail address \***

Automated e-mails, such as registration information, will be sent from this address. Use an address ending in your site's domain to help prevent these e-mails from being flagged as spam.

#### SITE MAINTENANCE ACCOUNT

**Username \***

Spaces are allowed; punctuation is not allowed except for periods, hyphens, and underscores.

**E-mail address \***

**Password \***

**Confirm password \***

Password strength:

**Fair**

Passwords match:
yes

To make your password stronger:

- Add lowercase letters
- Add uppercase letters
- Add punctuation

#### SERVER SETTINGS

**Default country**

Select the default country for the site.

**Default time zone**

By default, dates in this site will be displayed in the chosen time zone

#### UPDATE NOTIFICATIONS

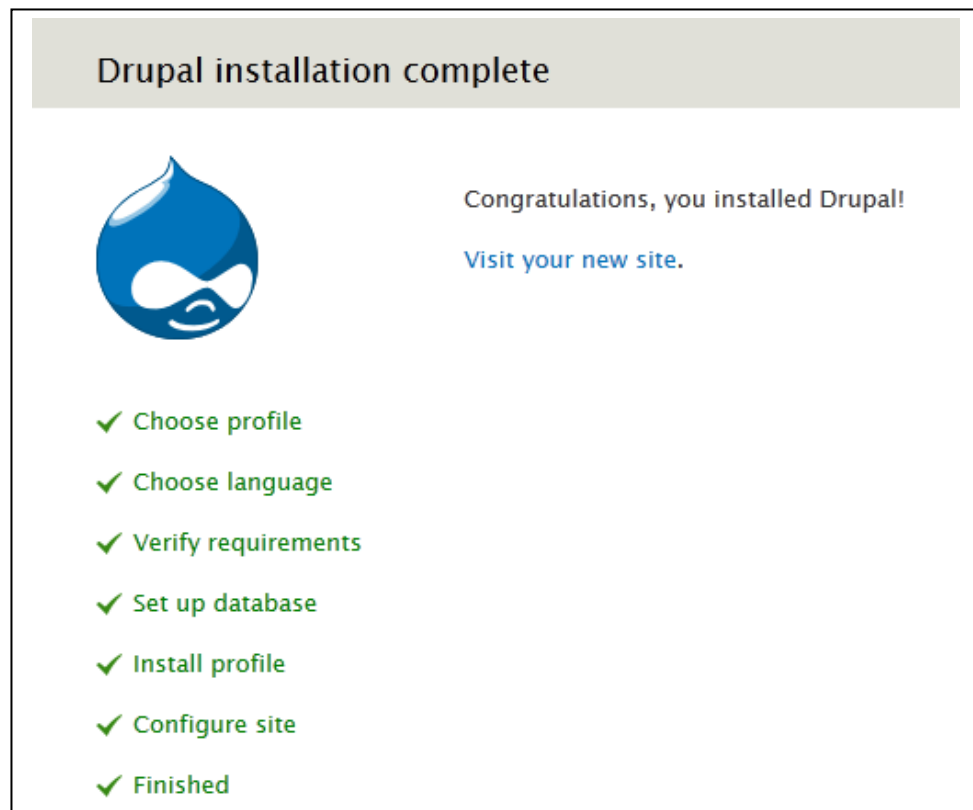
Check for updates automatically

Receive e-mail notifications

The system will notify you when updates and important security releases are available for installed components. Anonymous information about your site is *not* collected.

Save and continue

Setelah itu k



**Gambar 5.27** *Install Profile*

#### **5.4.4. Apache Benchmark**

Tahap ini yaitu melakukan simulasi *Apache Benchmark* sebelum dioptimalisasi, langkah pertama lihat *uptime* (sudah berapa lama *server* operasional) dengan perintah:

```
# uptime
```

Kemudian lakukan simulasi *benchmark* sebelum dioptimalisasi dengan simulasi 50 *request* 2 koneksi simultan dan 100 *request* 2 koneksi simultan

1. Simulasi dengan 50 request 2 koneksi *simultan*, dengan perintah:

```
# ab -n 50 -c 2 http://www.kopertis2.or.id/
```

```
Percentage of the requests served within a certain time (ms)
 50%    388
 66%    392
 75%    394
 80%    397
 90%    409
 95%    431
 98%    455
 99%    455
100%    455 (longest request)
```

**Gambar 5.28** Hasil Persentase *Benchmark* dengan 50 Request Sebelum Optimalisasi

2. Simulasi dengan 100 request 2 koneksi *simultan*, dengan perintah:

```
# ab -n 100 -c 2 http://www.kopertis2.or.id/
```

```
Percentage of the requests served within a certain time (ms)
 50%    389
 66%    392
 75%    393
 80%    394
 90%    396
 95%    400
 98%    408
 99%    509
100%    509 (longest request)
```

**Gambar 5.29** Hasil Persentase *Benchmark* dengan 100 Request Sebelum Optimalisasi



Berikut ini tabel perbandingan *konfigurasi* menggunakan *Apache Benchmark* sebelum dioptimalisasi, dengan simulasi 50 *request* dan 100 *request* dengan 2 koneksi simultan.

**Tabel 5.1** *Benchmark 50 Request dan 100 Request Sebelum Optimalisasi*

No.	<i>Benchmark Dengan 50 Request</i>		<i>Benchmark Dengan 100 Reque</i>	
	<i>Percent</i>	<i>Waktu (ms)</i>	<i>Percent</i>	<i>Waktu (ms)</i>
1.	50 %	388	50 %	389
2.	66 %	392	66 %	392
3.	75 %	394	75 %	393
4.	80 %	397	80 %	394
5.	90 %	409	90 %	396
6.	95 %	431	95 %	400
7.	98 %	455	98 %	408
8.	99 %	455	99 %	509
9.	100 %	455	100 %	509

Dari tabel di atas terdapat perbedaan hasil *Benchmark* dengan 50 *request* di 50% membutuhkan waktu 388 ms sedangkan 100 *request* di 50% membutuhkan waktu 389 ms, dan akhirnya hasil *Benchmark* dengan 50 *request* di 100% membutuhkan waktu 455 ms sedangkan 100 *request* di 100% membutuhkan waktu lebih lama yaitu 509 ms.

### 5.4.5. Apache Httpperf

Tahap ini yaitu lakukan simulasi *Apache Httpperf* sebelum dioptimalisasi, langkah pertama *install httpperf* dengan perintah:

```
# apt-get install httpperf
```

Kemudian lakukan simulasi *httpperf* sebelum dioptimalisasi dengan simulasi 50 *request* 2 koneksi simultan dan 100 *request* 2 koneksi simultan:

1. Simulasi dengan 50 *request*, 2 koneksi *simultan*, dengan perintah:

```
# httpperf -server www.kopertis2.or.id - -uri /index.php - -num-conn 50 - -num-cal 2 - -rate 2 timeout 5
```

```
Total: connections 50 requests 100 replies 50 test-duration 7.616 s
Connection rate: 6.6 conn/s (152.3 ms/conn, <=1 concurrent connections)
Connection time [ms]: min 142.7 avg 152.3 max 226.3 median 148.5 stddev 12.8
Connection time [ms]: connect 0.1
Connection length [replies/conn]: 1.000
Request rate: 13.1 req/s (76.2 ms/req)
Request size [B]: 81.0
Reply rate [replies/s]: min 6.4 avg 6.4 max 6.4 stddev 0.0 (1 samples)
Reply time [ms]: response 150.0 transfer 0.0
Reply size [B]: header 289.0 content 3778.0 footer 0.0 (total 4067.0)
Reply status: 1xx=0 2xx=0 3xx=0 4xx=0 5xx=50
CPU time [s]: user 0.70 system 3.00 (user 9.2% system 39.4% total 48.6%)
Net I/O: 27.1 KB/s (0.2*10^6 bps)
```

**Gambar 5.30** Hasil *Httpperf* dengan 50 *Request* Sebelum Optimalisasi

Jadi, setelah melakukan *httperf* dengan 50 *request*, waktu minimum yang dihasilkan yaitu 142,7ms dan nilai maximum 226,3ms dengan median 148,5.

**2. Simulasi dengan 100 request, 2 koneksi simultan,** dengan perintah:

```
# httperf -server www.kopertis2.or.id - -uri /index.php - -num-conn  
100 - -num-cal 2 - -rate 2 timeout 5
```

```
Total: connections 100 requests 200 replies 100 test-duration 15.071 s
Connection rate: 6.6 conn/s (150.7 ms/conn, <=1 concurrent connections)
Connection time [ms]: min 143.6 avg 150.7 max 297.9 median 148.5 stddev 15.9
Connection time [ms]: connect 0.1
Connection length [replies/conn]: 1.000

Request rate: 13.3 req/s (75.4 ms/req)
Request size [B]: 81.0

Reply rate [replies/s]: min 6.4 avg 6.6 max 6.8 stddev 0.2 (3 samples)
Reply time [ms]: response 149.2 transfer 0.0
Reply size [B]: header 289.0 content 3778.0 footer 0.0 (total 4067.0)
Reply status: 1xx=0 2xx=0 3xx=0 4xx=0 5xx=100

CPU time [s]: user 1.33 system 6.03 (user 8.8% system 40.0% total 48.8%)
Net I/O: 27.4 KB/s (0.2*10^6 bps)
```

**Gambar 5.31** Hasil *Httpperf* dengan 100 *Request* Sebelum Optimalisasi

Jadi, setelah melakukan *httperf* dengan 100 *request*, waktu minimum yang dihasilkan yaitu 143,6ms dan nilai maximum 297,9ms dengan median 148,5.

Berikut ini tabel perbandingan *konfigurasi* menggunakan *Httpperf* sebelum dioptimalisasi, dengan simulasi 50 *request* dan 100 *request*.

**Tabel 5.2** *Httpperf 50 Request dan 100 Request Sebelum Optimalisasi*

<i>Httpperf Dengan 50 Request</i>		<i>Httpperf Dengan 100 Request</i>	
Waktu Minimum	Waktu Maximum	Waktu Minimum	Waktu Maximum
142,7ms	226,3ms	143,6ms	297,9ms

#### 5.4.6. Optimalisasi *Web Server*

Tahap ini adalah tahap untuk mengoptimalkan *web server* dengan *PHP Xcache* dan *PHP Memcached*.

##### 5.4.6.1 *PHP XCache*

*PHP XCache* adalah *PHP opcode cacher* yang telah diuji oleh banyak *server* produksi di seluruh dunia yang membuat kinerja *script* / kode *php* yang di jalankan bisa lebih cepat, stabil dan mampu bekerja di bawah beban tinggi.

Langkah pertama yaitu *install PHP Xcache*, dengan perintah:

```
# apt-get install php5-xcache
```

Kemudian *restart php5-xcache*, dengan perintah:

```
# /etc/init.d/apache2 restart
```

#### 5.4.6.2 PHP Memcached

*PHP memcached* adalah *software* tambahan yang bertugas untuk menyimpan *cache* data dalam memori, memiliki kinerja tinggi, memori sistem objek terdistribusi *caching*, bersifat umum. Digunakan dalam mempercepat aplikasi *web* dinamis dengan meringankan beban *input* dan *output database*.

Langkah selanjutnya yaitu *install php5-memcached*, dengan perintah:

```
# apt-get install memcached php5-memcached
```

Kemudian *search memcache*, dengan perintah:

```
# apt-cache search memcache
```

Langkah selanjutnya setelah *install Xcache* dan *memcached* yaitu lakukan *Benchmark* dan *httperf*.

#### 5.4.7 Hasil Pengukuran Apache Benchmark dan Httperf Setelah Optimalisasi

Setelah di Optimalisasi menggunakan Xcache dan Memcached, kemudian langkah selanjutnya yaitu pengukuran kembali dengan *Benchmark* dan *Httperf* dengan simulasi 50 *request* 2 koneksi simultan dan 100 *request* 2 koneksi simultan.

1. **Benchmark Setelah Optimalisasi menggunakan Apache Benchmark (ab)**

Simulasi dengan 50 *request*, 2 koneksi *simultan*, dengan perintah:

```
# ab -n 50 -c 2 http://www.kopertis2.or.id/
```

```
Percentage of the requests served within a certain time (ms)
 50%    63
 66%    64
 75%    64
 80%    65
 90%    69
 95%    69
 98%    75
 99%    82
100%    82 (longest request)
```

**Gambar 5.32** Hasil Persentase *Benchmark* dengan 50 *Request* Setelah Optimalisasi

2. Simulasi dengan 100 *request*, 2 koneksi *simultan*, dengan perintah:

```
# ab -n 100 -c 2 http://www.kopertis2.or.id/
```

```
Percentage of the requests served within a certain time (ms)
 50%    64
 66%    65
 75%    66
 80%    66
 90%    67
 95%    72
 98%    89
 99%   116
100%   116 (longest request)
```

**Gambar 5.33** Hasil Persentase *Benchmark* dengan 100 *Request* Setelah Optimalisasi

Berikut ini tabel perbandingan *konfigurasi* menggunakan *Apache Benchmarks* setelah dioptimalisasi, dengan simulasi 50 *request* dan 100 *request*.

**Tabel 5.3** *Benchmark 50 Request dan 100 Request Setelah Optimalisasi*

No.	<i>Benchmark Dengan 50 Request</i>		<i>Benchmark Dengan 100 Request</i>	
	<i>Percent</i>	<i>Waktu (ms)</i>	<i>Percent</i>	<i>Waktu (ms)</i>
1.	50 %	63	50 %	64
2.	66 %	64	66 %	65
3.	75 %	64	75 %	66
4.	80 %	65	80 %	66
5.	90 %	69	90 %	67
6.	95 %	69	95 %	72
7.	98 %	75	98 %	89
8.	99 %	82	99 %	116
9.	100 %	82	100 %	116

Dari tabel di atas terdapat perbedaan hasil *Benchmark* dengan 50 *request* di 50% membutuhkan waktu 63 ms sedangkan 100 *request* di 50% membutuhkan waktu 64 ms, dan akhirnya hasil *Benchmark* dengan 50 *request* di 100% membutuhkan waktu 82 ms sedangkan 100 *request* di 100% membutuhkan waktu lebih lama yaitu 116 ms.

**a. Benchmark setelah optimasi menggunakan *Httpperf***

Tahap ini penulis melakukan simulasi dengan 50 *request* 2 koneksi simultan dan 100 *request* 2 koneksi simultan:

1. Simulasi dengan 50 *request*, 2 koneksi *simultan*, dengan perintah:

```
# httpperf -server www.kopertis2.or.id - -uri /index.php - -num-conn
50 - -num-cal 2 - -rate 2 timeout 5
```

```
Total: connections 50 requests 100 replies 50 test-duration 3.104 s
Connection rate: 16.1 conn/s (62.1 ms/conn, <=1 concurrent connections)
Connection time [ms]: min 53.2 avg 62.1 max 182.7 median 60.5 stddev 17.5
Connection time [ms]: connect 0.1
Connection length [replies/conn]: 1.000

Request rate: 32.2 req/s (31.0 ms/req)
Request size [B]: 81.0

Reply rate [replies/s]: min 0.0 avg 0.0 max 0.0 stddev 0.0 (0 samples)
Reply time [ms]: response 60.6 transfer 0.0
Reply size [B]: header 289.0 content 3778.0 footer 0.0 (total 4067.0)
Reply status: 1xx=0 2xx=0 3xx=0 4xx=0 5xx=50

CPU time [s]: user 0.26 system 1.22 (user 8.5% system 39.4% total 47.9%)
Net I/O: 66.5 KB/s (0.5*10^6 bps)
```

**Gambar 5.34** Hasil *Httpperf* dengan 50 *Request* Setelah Optimalisasi

Jadi, setelah melakukan *httpperf* dengan 50 *request*, waktu minimum yang dihasilkan yaitu 53,2ms dan nilai maximum 182,7ms dengan median 60,5.

2. Simulasi dengan 100 *request*, 2 koneksi *simultan*, dengan perintah:

```
# httpperf -server www.kopertis2.or.id - -uri /index.php - -num-conn
100 - -num-cal 2 - -rate 2 timeout 5
```



```

Total: connections 100 requests 200 replies 100 test-duration 6.106 s
Connection rate: 16.4 conn/s (61.1 ms/conn, <=1 concurrent connections)
Connection time [ms]: min 54.0 avg 61.1 max 186.8 median 58.5 stddev 16.0
Connection time [ms]: connect 0.1
Connection length [replies/conn]: 1.000

Request rate: 32.8 req/s (30.5 ms/req)
Request size [B]: 81.0

Reply rate [replies/s]: min 16.4 avg 16.4 max 16.4 stddev 0.0 (1 samples)
Reply time [ms]: response 58.3 transfer 0.0
Reply size [B]: header 289.0 content 3778.0 footer 0.0 (total 4067.0)
Reply status: 1xx=0 2xx=0 3xx=0 4xx=0 5xx=100

CPU time [s]: user 0.54 system 2.41 (user 8.8% system 39.5% total 48.3%)
Net I/O: 67.6 KB/s (0.6*10^6 bps)

```

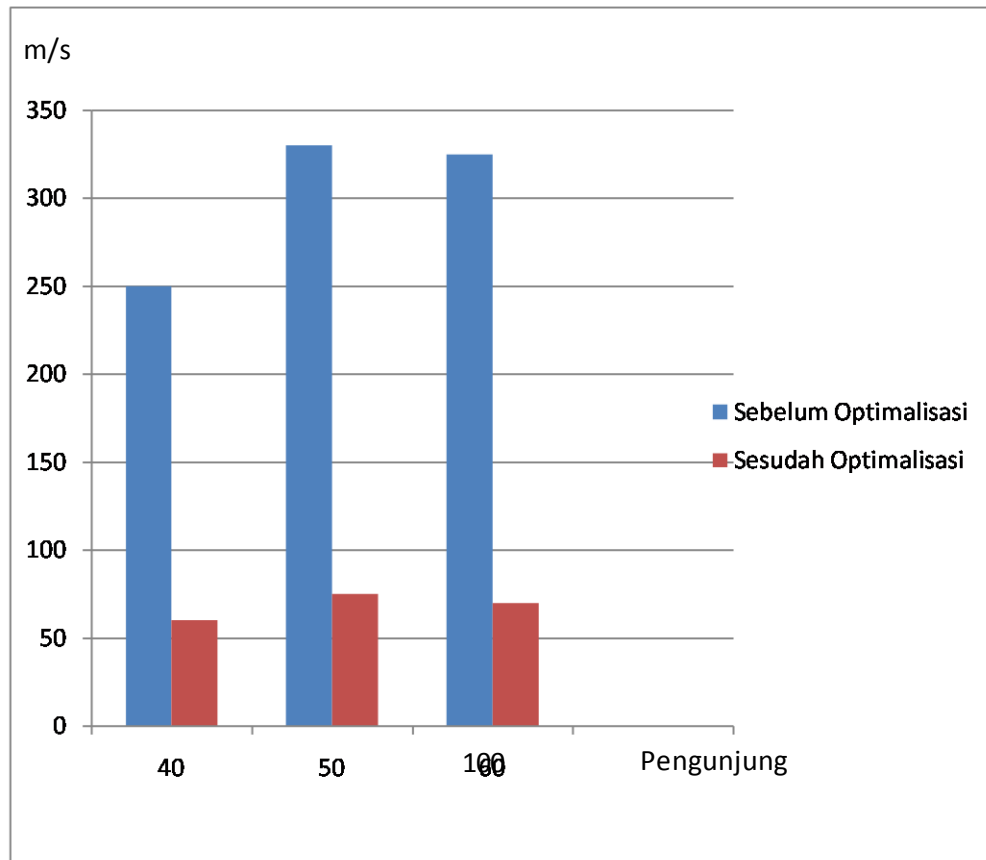
**Gambar 5.35** Hasil *Httpperf* dengan 100 Request Setelah Optimalisasi

Jadi, setelah melakukan *httpperf* dengan 100 request, waktu minimum 54,0ms dan nilai maximum 186,8ms dengan median 58,5

Berikut ini tabel perbandingan konfigurasi menggunakan *Httpperf* sebelum dioptimalisasi, dengan simulasi 50 request dan 100 request.

**Tabel 5.4** *Httpperf* 50 Request dan 100 Request Setelah Optimalisasi

<i>Httpperf</i> Dengan 50 Request		<i>Httpperf</i> Dengan 100 Request	
Waktu Minimum	Waktu Maximum	Waktu Minimum	Waktu Maximum
53,2ms	182,7ms	54,0ms	186,8ms



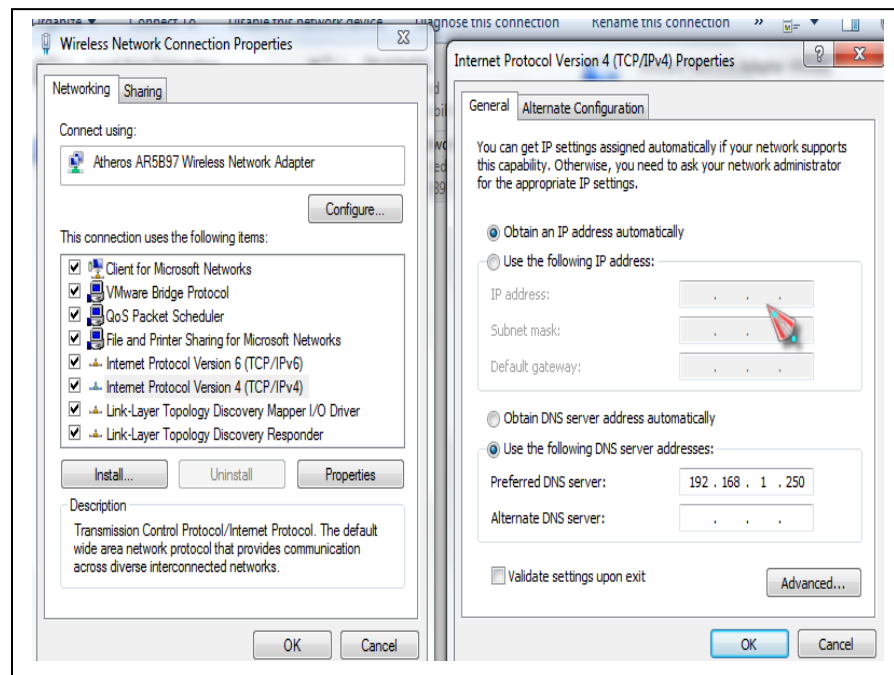
**Gambar 5.36** Grafik Sebelum dan Setelah Optimalisasi dengan 3 Kali Percobaan

Grafik diatas dihasilkan dari tiga *sample* data yang diambil pada tanggal 10 Juni 2012 – 12 Juni 2012 pada pukul 12.00 wib. Pada percobaan pertama sebelum optimalisasi kecepatan akses 250 m/s, kemudian dioptimalisasi menghasilkan kecepatan 60 m/s. Pada percobaan kedua sebelum optimalisasi kecepatan akses 330 m/s, kemudian dioptimalisasi menghasilkan kecepatan 75 m/s. Pada percobaan ketiga sebelum optimalisasi kecepatan akses 320 m/s, kemudian dioptimalisasi menghasilkan kecepatan 70 m/s.

#### 5.4.8. Konfigurasi Komputer Client

Konfigurasi komputer *client* menggunakan *windows seventh*:

Langkah pertama yaitu pilih *start*, lalu pilih *control panel*, pilih *network connection*. Setelah itu klik kanan *local area connection* pilih *properties* lalu *wireless network connection properties*. Masukkan *IP Address client* 192.168.1.250 pada *internet protocol version 4*, seperti gambar dibawah ini:



**Gambar 5.37** Setting IP Address

Setelah itu buka halaman *browser* ketik [www.kopertis2.or.id](http://www.kopertis2.or.id), maka muncul tampilan *drupal* berikut:



**Gambar 5.38** Tampilan *Drupal*

### 5.5. *Monitoring*

Setelah implementasi selanjutnya tahapan *monitoring* merupakan tahapan yang penting, agar jaringan komputer dan komunikasi dapat berjalan sesuai dengan keinginan dan tujuan awal dari *user* pada tahap awal analisis, maka perlu dilakukan kegiatan *monitoring*.

Tahap *monitoring* dilakukan penulis dengan menggunakan AWStats untuk *monitoring web server*.

Langkah pertama yaitu *install Awstats*, dengan perintah:

```
# apt-get install awstats
```

Kemudian ubah konfigurasi *Awstats* dan sesuaikan isinya, dengan perintah:

```
# nano /etc/awstats/awstats.conf
```

Lalu *setting* di *line* 122 ubah `LogFormat=1`, di *line* 153 ubah `SiteDomain=www.kopertis2.or.id`, di *line* 168 ubah `HostAliases="localhost 192.168.1.250 REGEX[server\..world$] REGEX[^10\0\0\0]"`, dan di *line* 222 ubah `DirIcons="/icon"`.

Langkah selanjutnya yaitu melihat letak akses *Awstats* dan *Simbolik link* atau membuat *shortcut*, gunakan perintah:

```
# mkdir /var/www/awstats
```

```
# ln -s /usr/share/awstats/icon /var/www/awstats/icons
```

Kemudian *setting* akses *Awstats* dari jaringan ubah dan tambahkan *settingan*, dengan perintah:

```
# nano /etc/apache2/sites-available/default
```

```
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
<Directory "/usr/lib/cgi-bin">
    AllowOverride None
    Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
    Order allow,deny
    Allow from 192.168.1.250/24
</Directory>

<Location /awstats>
    Order deny,allow
    Deny from all
    Allow from 192.168.1.250/24
</Location>
```

**Gambar 5.39** Mengubah Akses *Awstats*

Kemudian membuat laporan *Awstats* (*Generate report*), dengan perintah:

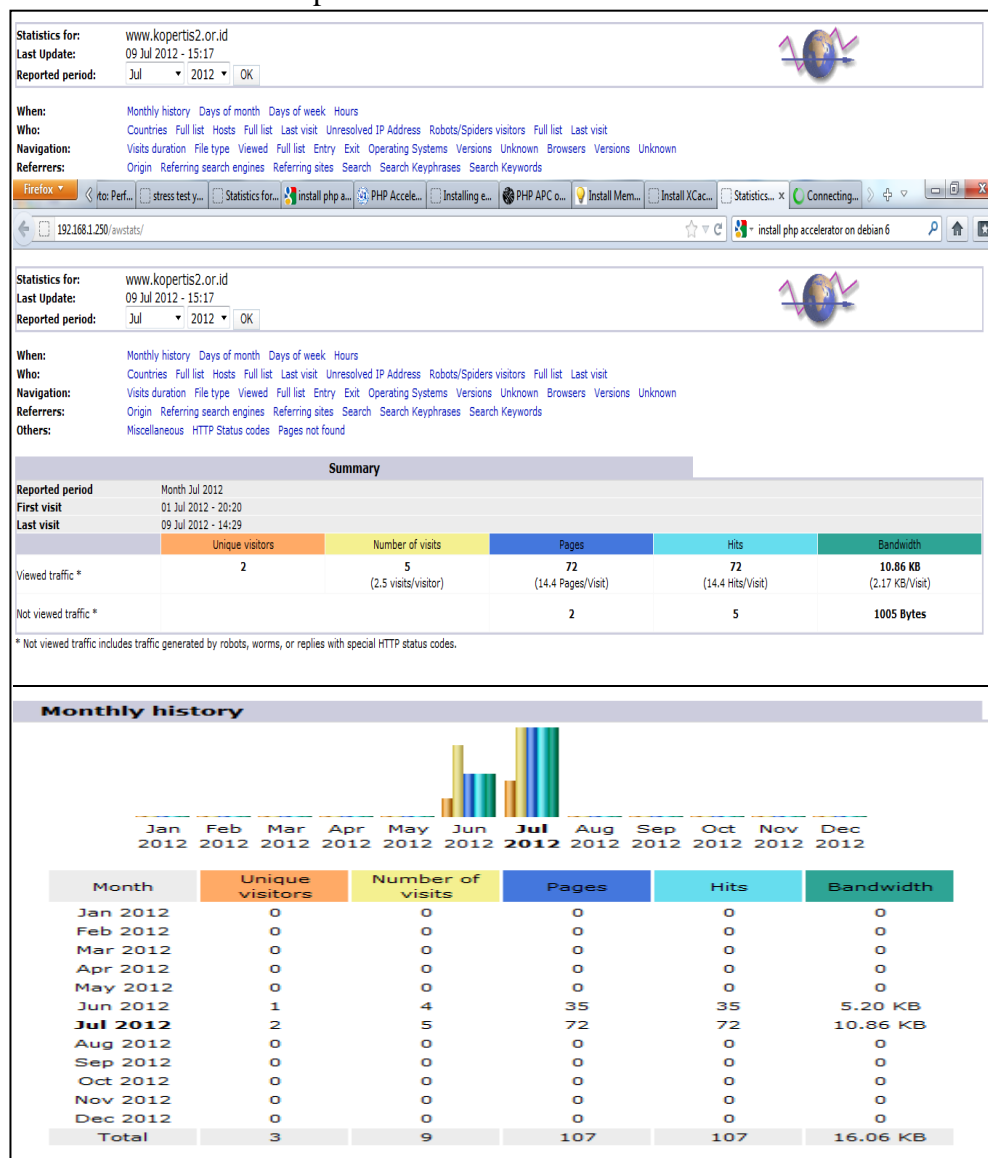
```
# /usr/lib/cgi-bin/awstats.pl --config=apache --update
```

Lalu langkah selanjutnya meng-*eksport* laporan agar dapat dilihat, gunakan perintah:

```
# /usr/lib/cgi-bin/awstats.pl --config=apache --output --staticlink >
/var/www/awstats/index.html
```

Kemudian buka halaman *browser* dengan ketik 192.168.1.250/awstats

Lalu akan muncul tampilan *awstats* berikut ini:



Gambar 5.40 Tampilan Awstats

Jadi, pada gambar diatas sudah terdapat 2 *unique visitor* yaitu *client* yang berkunjung ke situs [www.kopertis2.or.id](http://www.kopertis2.or.id), ada 5 *number of visits* yaitu jumlah berapa kali 2 *unique visitor* tersebut berkunjung, dengan 72 *pages* yaitu jumlah halaman yang diakses, lalu terdapat 72 *hits* yaitu jumlah seluruh kunjungan baik itu *client*, *robot*, atau *software* dan banyaknya data yang digunakan 10.86 kb.

## 5.6. Manajemen

Manajemen atau pengaturan adalah salah satu yang menjadi perhatian khusus masalah bagaimana mengatur kerja *web server* tersebut.

Tahap ini dilakukan penulis dengan cara mengatur atau memanajemen desain *webserver* berbasis *open source* pada Kopertis Wilayah II Palembang.

Sinkronisasi ulang berkas paket indeks Debian Linux termasuk pembaruan keamanan (akses *internet* yang diperlukan) , langkah ini akan mengupdate daftar *package* dari *repository* dengan cara:

**# apt-get update**

```

root@kopertis2:/home/febriyanda# apt-get update_
Ign cdrom://[Debian GNU/Linux 6.0.4 _Squeeze_ - Official i386 CD Binary-1 201201
28-12:53] squeeze Release.gpg
Ign cdrom://[Debian GNU/Linux 6.0.4 _Squeeze_ - Official i386 CD Binary-1 201201
28-12:53]/ squeeze/main Translation-en
Ign cdrom://[Debian GNU/Linux 6.0.4 _Squeeze_ - Official i386 CD Binary-1 201201
28-12:53]/ squeeze/main Translation-en_US
Ign cdrom://[Debian GNU/Linux 6.0.4 _Squeeze_ - Official i386 CD Binary-1 201201
28-12:53] squeeze Release
Ign cdrom://[Debian GNU/Linux 6.0.4 _Squeeze_ - Official i386 CD Binary-1 201201
28-12:53] squeeze/main i386 Packages/DiffIndex
19% [Connecting to kambing.ui.ac.id (152.118.24.30)] [Connecting to http.us.deb_

```

**Gambar 5.41** Update Web Server

*Update* yang berguna untuk memperbarui *database* paket-paket yang digunakan

Kemudian tahap selanjutnya yaitu upgrade *web server* yang sudah terinstall di debian 6 untuk memperbarui program – program ke versi baru hasil dari *update* tersebut, dengan cara:

**# apt-get upgrade**

```

root@kopertis2:/home/febriyanda# apt-get upgrade_
root@NANDA:/home/febriyanda# apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be upgraded:
  acpid at base-files bind9 bind9-doc bind9-host bind9utils dnsutils host
  initscripts libapr1 libbind9-60 libc-bin libc6 libc6-i686 libdns69 libisc62
  libisccc60 libiscfg62 liblwres60 libmysqlclient16 linux-base
  linux-image-2.6.32-5-686 locales lwresd mysql-client mysql-client-5.1
  mysql-common mysql-server mysql-server-5.1 mysql-server-core-5.1
  openssh-client openssh-server procps python python-minimal sysv-rc sysvinit
  sysvinit-utils tzdata
40 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 65.0 MB/65.8 MB of archives.
After this operation, 741 kB of additional disk space will be used.

```

**Gambar 5.42** *Upgrade Web Server*



## BAB VI

### SIMPULAN DAN SARAN

#### 6.1. Simpulan

Adapun kesimpulan yang dapat penulis ambil dari pembahasan bab-bab sebelumnya adalah sebagai berikut:

Kopertis Wilayah II Palembang belum memiliki *web server*, *website* yang terdapat di Kopertis Wilayah II Palembang menggunakan jasa *hosting* pihak lain dan tidak optimal dalam menangani permintaan akses dari *client*. Masalah yang sering dihadapi adalah saat *client* melakukan akses secara bersamaan waktu *loading* yang diperlukan menjadi lebih lama sehingga *client* harus menunggu sampai beberapa menit untuk bisa melakukan akses ke halaman *website*, hal tersebut tentu saja membuat *client* yang sedang melakukan akses merasa tidak nyaman, karena akses *internet* yang lama.

Penulis membuat sebuah *web server* menggunakan sistem operasi *open source*, dengan performa yang handal sehingga masalah tersebut dapat diminimalisasi.

Berdasarkan hasil desain dan optimalisasi *web server* diketahui bahwa desain dan optimalisasi *web server* berhasil. Telah dihasilkan sebuah konfigurasi sistem yang menunjang kinerja *web server Apache* sehingga mampu berkinerja lebih cepat dibandingkan konfigurasi awal.

Kecepatan rata – rata yang dihasilkan *Apache Benchmark* dengan 50 request 2 koneksi simultan sebelum dioptimalisasi yaitu 419,5ms, sedangkan setelah dioptimalisasi yaitu 70,33ms. Kemudian kecepatan rata – rata yang dihasilkan *Httpperf* dengan 50 request 2 koneksi simultan sebelum dioptimalisasi yaitu 184,5ms sedangkan setelah dioptimalisasi yaitu 117,95ms

Dengan *konfigurasi* ini Kopertis Wilayah 2 diharapkan akan mampu memberikan pelayanan dan informasi menggunakan jalur *on-line* dengan lebih baik lagi.

## 6.2. Saran

1. Karena penulis hanya membuat desain dan optimalisasi *webserver* menggunakan *Drupal* kedepannya diharapkan dapat menggunakan metode *web server* yang lain seperti *Lighttpd*, *Nginx* dan *Cherokee*.
2. Penulis berharap Kedepannya ditambah *hardware* baru sehingga dapat diterapkan mekanisme *load balancing*.
3. Untuk kedepannya penulis berharap sebaiknya *database* menggunakan *server* yang terpisah, sehingga kinerjanya dapat dievaluasi lebih optimal.
4. Kopertis wilayah 2 sebaiknya meng-*upgrade* koneksi *internet* yang dipakai sebagai *backbone* koneksi.

## DAFTAR PUSTAKA

- Andersson Mikael. 2005. *Performa Modeling of an Apache Web Server with Bursty Arrival Traffic*. *Internet Computing*, 5 (5) : 57-61.
- Kuncoro, Mudrajat. 2009. *Metode Riset Untuk Bisnis dan Ekonomi*. Jakarta : Erlangga.
- Lukita, Sari. 2010. *Analisis Perbandingan Load Balancing Web Server Tunggal dengan Web Server Cluster Menggunakan Linux Virtual Server*. *Generic*, 5 (2) : 31-33.
- Sofana, Iwan. 2008. *Membangun Jaringan Komputer*. Bandung : Informatika.
- Sopandi, Dede. 2008. *Instalasi dan Konfigurasi Jaringan Komputer*. Bandung: Informatika.
- Simarmata, Janner. 2010. *Rekayasa Web*. Yogyakarta: Andi.
- Sugeng, Winarno. 2010. *Jaringan Komputer Dengan TCP/IP*. Bandung: Modula.
- Sutanta, Edhy. 2005. *Komunikasi Data dan Jaringan Komputer*. Yogyakarta: Graha Ilmu.
- Utomo, Eko Priyo. 2006. *Buku Pengantar Jaringan Komputer Bagi Pemula*. Bandung : Yrama Widya.
- Yani, Ahmad dan Fanny Hfiary. 2005. *Mudah Murah Membangun Jaringan Internet dengan Linux Router Project Leaf Bearing*. Jakarta : Elex Media Komputindo.
- Wahana Komputer. 2010. *MySQL Database Server*. Jakarta :Mediakita.

Wahana Komputer. 2009. **PHP Programming**. Semarang : Andi.

([http:// apache.org/docs/2.0/programs](http://apache.org/docs/2.0/programs), diakses 27 Juni 2012).

([http:// eprints.unsri.ac.id](http://eprints.unsri.ac.id), diakses 20 Mei 2012).

([http:// estiko.info/php-accelerator/](http://estiko.info/php-accelerator/), diakses 20 Juni 2012).

([http:// hpl.hp.com/research /linux/httpperf](http://hpl.hp.com/research/linux/httpperf) diakses 20 Juni 2012).

([http:// ilmu-linux.com/category/PHP](http://ilmu-linux.com/category/PHP), diakses 20 Juni 2012).

([http:// lup.lub.lu.se/luur](http://lup.lub.lu.se/luur), diakses 20 Mei 2012).

([http :// opensource-indonesia.com](http://opensource-indonesia.com), diakses pada tanggal 20 Mei 2012).