

LAPORAN PENELITIAN



IMPLEMENTASI REPLIKASI BASIS DATA MELALUI JARINGAN VIRTUAL PRIVATE NETWORK (VPN) PADA PDAM KABUPATEN MALANG

Oleh:

IR. HERU NURWARSITO, M.KOM.
IR. HERI PRAYITNO
YUNAN NAUFAL

FAKULTAS TEKNIK
ITNIVERSITAS BRAWIJAYA
MALANG
2007

IDENTITAS DAN URAIAN UMUM PENELITIAN

1. Judul **Usulan:**

Implementasi replikasi Basis Data melalui jaringan Virtual Private Network (VPN) pada PDAM Kabupaten Malang

2. **Ketua** Peneliti

- a) Nama **lengkap** Ir. Heru Nurwarsito, M.Kom.
- b) Bidang keahlian : Sistem **Informasi**, Sistem komputer
- c) **Jabatan Struktural** : KKDK
- d) **Jabatan Fungsional/Golongan** : **Lektor Kepala/IV-a**
- e) Unit **kerja** : Fakultas Teknik
- f) **Alamat surat** : Jl MT. I-laryono 167, Malang
- g) **Telpon/Faks** : 0341-7098989
- h) E-mail : herunur@gmail.com

3. Anggota Tim Peneliti

No	Nama dan gelar akademik	Bidang Keahlian	Alokasi waktu
1	Ir. Heru Nurwasito, M.Kom	Sistem Informasi	6 jam/minggu
2	Ir. Heri Prayitno	Database	4 jam/minggu
3	Yunan Naufal	Programming	4 jam/minggu

4. **Objek** penelitian

Replikasi Basis Data

5. Masa **pelaksanaan** penelitian:

- **Mulai** : Nopember 2006
- **Berakhir** : April 2007

6. Anggaran yang **diusulkan:** Rp. 3.000.000,-7. Lokasi **penelitian:** Laboratorium Sistem **Informasi**

8. Hasil yang ditargetkan:

Implementasi replikasi **Basis** Data melalui jaringan Virtual Private Network (VPN) pada PDAM Kabupaten Malang

9. Institusi lain yang ferlibat: -

LAPORAN PENELITIAN



IMPLEMENTASI REPLIKASI BASIS DATA MELALUI JARINGAN VIRTUAL PRIVATE NETWORK (VPN) PADA PDAM KABUPATEN MALANG

Oleh:

**IR. HERU NURWARITO, M.KOM.
IR. HERI PRAYITNO
YUNAN NAUFAL**

**FAKULTAS TEKNIK
IJNIVERSITAS BRAWIJAYA
MALANG
2007**

HALAMAN PENGESAHAN

1. **Judul Penelitian** : **Implementasi replikasi Basis Data melalui jaringan Virtual Private Network (VPN) pada PDAM Kabupaten Malang**
2. **Ketua Peneliti**
 - a. **Nama lengkap** : **Ir. Heru Nurwarsito, M.Kom.**
 - b. **Jenis Kelamin** : **Laki-laki**
 - c. **NIP** : **131 879033**
 - d. **Jabatan Struktural** : **KKDK**
 - e. **Jabatan Fungsional/Golongan** : **Lektor Kepala/IV-a**
 - f. **Fakultas/Jurusan** : **Teknik/Teknik Elektro**
 - g. **Pusat Penelitian** : **BPP FT UB**
 - h. **Alamat** : **Jl. Mayjen Haryono 167 Malang**
 - i. **Telepon/Fax** : **(0341) 553286**
 - j. **E-mail** : **herunur@gmail.com**
3. **Anggota Penelitian**
 - a. **Nama Anggota Pelaksana** : **Ir. Heri Prayitno.**
 - ↳ **Mahasiswa Asisten Peneliti** : **Yunan Naufal**
4. **Lokasi Penelitian** : **Lab. Pemrograman Komputer, PDAM**
5. **Jangka Waktu Penelitian** : **6 bulan**
5. **Biaya Penelitian** : **Rp. 3.000.000,00 (Tiga juta Rupiah)**
6. **Sumber Dana** : **Mandiri**

Malang, 17 April 2007

Mengetahui,
Ketua BPP Fakultas Teknik
Universitas Brawijaya

Ketua Peneliti,



Dr.Ir.Arief Rahmansyah
NIP. 132 059 302

Ir. Heru Nurwarsito, M.Kom.
NIP. 131 879 033

Menyetujui,
Dekan Fakultas Teknik
Universitas Brawijaya



Ir. Imam Zaky, MT
NIP. 130 604 494

HALAMAN PENGESAHAN

1. Judul Penelitian : Implementasi replikasi Basis Data melalui jaringan Virtual Private Network (VPN) pada PDAM Kabupaten Malang
2. Ketua Peneliti
- a. Nama lengkap : Ir. Heru Nurwarsito, M.Kom.
 - b. Jenis Kelamin : Laki-laki
 - c. NIP : 131 879033
 - d. Jabatan Struktural : KKDK
 - e. Jabatan Fungsional/Golongan : Lektor Kepala/IV-a
 - f. Fakultas/Jurusan : Teknik/Teknik Elektro
 - g. Pusat Penelitian : BPP FT UB
 - h. Alamat : Jl. Mayjen Haryono 167 Malang
 - i. Telepon/Fax : (0341) 553286
 - j. E-mail : herunur@gmail.com
3. Anggota Penelitian
- a. Nama Anggota Pelaksana : Ir. Heri Prayitno.
 - b. Mahasiswa Asisten Peneliti : Yunan Na'fal
4. Lokasi Penelitian : Lab. Pemrograman Komputer, PDAM
5. Jangka Waktu Penelitian : 6 bulan
5. Biaya Penelitian : Rp. 3.000.000,00 (Tiga juta Rupiah)
6. Sumber Dana : Mandiri

Malang, 17 April 2007

Mengetahui,
Ketua BPP Fakultas Teknik
Universitas Brawijaya

Ketua Peneliti,



Dr.Ir.Arief Rahmansyah
NIP. 132 059 302

Ir. Heru Nurwarsito, M.Kom
NIP. 131 879 033

Menyetujui,
Dekan Fakultas Teknik
Universitas Brawijaya



Ir. Imam Zaky, MT
NIP. 130 604 494

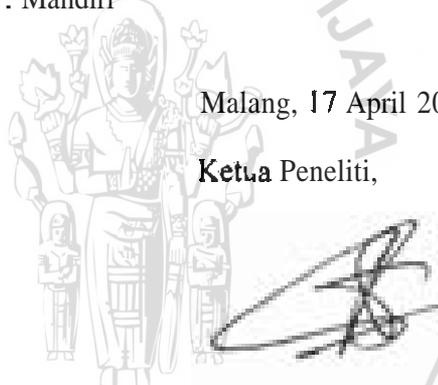
HALAMAN PENGESAHAN

1. Judul Penelitian : **Implementasi replikasi** Basis Data **melalui jaringan** Virtual Private Network (VPN) pada PDAM Kabupaten Malang
2. Ketua Peneliti
 - a. Nama lengkap : **Ir. Heru Nurwarsito, M.Kom.**
 - b. **Jenis** Kelamin : Laki-laki
 - c. NIP : **131 879033**
 - d. **Jabatan Struktural** : KKDK
 - e. **Jabatan Fungsional/Colongan** : Lektor **Kepala/IV-a**
 - f. **Fakultas/Jurusan** : **Teknik/Teknik** Elektro
 - g. **Pusat Penelitian** : BPP FT UB
 - h. Alamat : **Jl. Mayjen Haryono 167 Malang**
 - i. **Telepon/Fax** : (0341) 553286
 - j. E-mail : herunur@gmail.com
3. Anggota Penelitian
 - a. Nama Anggota Pelaksana : **Ir. Heri Prayitno.**
 - b. Mahasiswa **Asisten** Peneliti : Yunan Naufal
4. **Lokasi Penelitian** : Lab. **Pemrograman** Komputer, PDAM
5. **Jangka Waktu Penelitian** : **6** bulan
5. **Biaya Penelitian** : Rp. **3.000.000,00** (Tiga juta Rupiah)
6. **Sumber Dana** : Mandiri

Mengetahui,
Ketua BPP Fakultas **Teknik**
Universitas **Brawijaya**

Malang, 17 April 2007

Ketua Peneliti,



Dr.Ir.Arief Rahmansyah
NIP. **132 059 302**

Ir. Heru Nurwarsito, M.Kom.
NIP. **131 879 033**

Menyetujui,
Dekan Fakultas **Teknik**
Universitas **Brawijaya**

Ir. Imam Zaky, MT
NIP. **13G 504 494**



IDENTITAS DAN URAIAN UMUM PENELITIAN

1. Judul Usulan:

Implementasi replikasi Basis Data melalui **jaringan** Virtual Private Network (VPN) pada PDAM Kabupaten Malang

2. Ketua Peneliti

- a) Nama lengkap **Ir. Heru Nurwarsito, M.Kom.**
- b) Bidang keahlian : Sistem Informasi, Sistem **komputer**
- c) **Jabatan Struktural** : KKDK
- d) **Jabatan Fungsional/Golongan** : Lektor **Kepala/IV-a**
- e) Unit kerja : Fakultas Teknik
- f) **Alamat surat** : Jl **MT. Haryono** 167, Malang
- g) **Telpom/Faks** : 0341-7098989
- h) E-mail : herunur@grnail.com

3. Anggota Tim Peneliti

No	Nama dan gelar akademik	Bidang Keahlian	 Alokasi waktu
1	Ir. Heru Nurwasito, M.Kom	 Sistem Informasi	6 jam/minggu
 2	 Ir. Heri Prayitno	Database	4 jam/minggu
 3	Yunan Naufal	Programming	4 jam/minggu

4. Objek penelitian

Replikasi Basis Data

5. Masa pelaksanaan penelitian:

- **Mulai** : Nopember 2006
- **Berakhir** : April 2007

6. Anggaran yang **diusulkan**: Rp. 3.000.000,.

7. Lokasi penelitian: **Laboratorium Sistem Informasi**

8. Hasil yang **ditargetkan**:

Implementasi replikasi Basir Data **melalui jaringan** Virtual Private Network (VPN) **pada** PDAM Kabupaten Malang

9. Institusi lain yang terlibat: -

RINGKASAN

Implementasi Replikasi Basis Data Melalui Jaringan Virtual Private Network (VPN) pada PDAM Kabupaten Malang

Ir. Heru Nurwarsito, M.Kom.

Ir. Heri Prayitno

Yuuan Naufal.

Saat ini PDAM Malang sudah memiliki aplikasi basis data terpusat dan sudah digunakan lebih dari 10 tahun. Karena kebutuhan manajemen, pihak PDAM menginginkan mendistribusikan aplikasi basis datanya ke setiap cabang. Jika aplikasi basis data ini didistribusikan ke setiap cabang, maka akan muncul masalah saat mengintegrasikan data dari masing-masing cabang. Selain itu pihak PDAM menginginkan perubahan data di setiap cabang dikirim secara berkala agar dapat menekan biaya komunikasi. Salah satu alternatif untuk menyelesaikan permasalahan yang dihadapi oleh PDAM kabupaten Malang ini adalah menerapkan replikasi basis data terdistribusi menggunakan jaringan *Virtual Private Network* dengan proses *update* yang terjadwal.

Perancangan dan pengimplementasian Replikasi Basis Data Melalui Jaringan Virtual Private Network (VPN) pada PDAM Kabupaten Malang dilakukan dengan menggunakan Microsoft SQL server 2000 sebagai basis datanya dan OpenVPN 2.0.9 untuk program aplikasi koneksi VPN. Untuk bahan-bahannya diambil langsung dari PDAM yaitu Data pelanggan PDAM Kabupaten Malang dan Data transaksi pembayaran rekening air selama 1 tahun.

Pengujian implementasi replikasi basis data melalui jaringan *Virtual Private Network* (VPN) dilakukan apakah telah berjalan sesuai yang diharapkan oleh tujuan awal dari pengerjaan penelitian ini. Pengujian terdiri dari pengujian per blok dimana terdapat 3 pengujian yaitu; pengujian *restore database*, pengujian koneksi antar komputer, pengujian otomatisasi *Dial Up*, Pengujian otomatisasi koneksi VPN dan pengujian replikasi. Setelah pengujian per blok, dilakukan pengujian sistem secara keseluruhan, Pengujian *Koneksi Database* pada Aplikasi Microsoft Access dan Pengujian Keamanan Data. Pengujian-pengujian tersebut telah berjalan sesuai dengan jadwal yang telah ditentukan.

Kata Kunci : *replikasi, VPN, PDAM, database, jaringan.*

KATA PENGANTAR

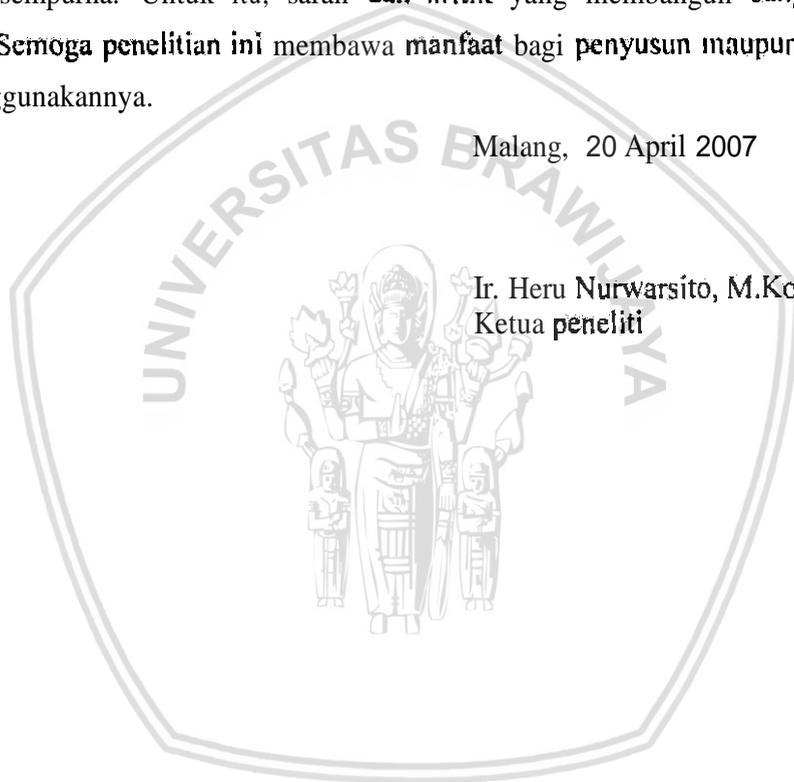
Alhamdulillah, Segala puji syukur ke hadirat Allah SWT atas rahmat, hidayah serta karunia-Nya, sehingga kami dapat menyelesaikan penelitian yang berjudul **“Implementasi** replikasi Basis Data **melalui** jaringan Virtual Private Network (VPN) **pada** PDAM Kabupaten **Malang”** dengan baik dan lancar.

Kegiatan ini merupakan salah satu perwujudan Tri Dharma Perguruan Tinggi yang harus dilakukan oleh setiap dosen.

Penulis menyadari bahwa penelitian ini masih banyak kekurangan dan masih jauh dari sempurna. Untuk itu, saran dan kritik yang membangun sangat penulis harapkan. Semoga penelitian ini membawa manfaat bagi penyusun maupun pihak lain yang menggunakannya.

Malang, 20 April 2007

Ir. Heru Nurwarsito, M.Kom.
Ketua peneliti



DAFTAR ISI

RINGKASAN.....	iii
KATA PENGANTAR.....	iv
DAFTAR ISI	v
DAFTAR GAMBAR.....	ix
DAFTAR TABEL	xiv
DAFTAR LAMPIRAN	xv
BAB I PENDAHULUAN	
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah	2
1.3. Ruang Lingkup	2
1.4. Tujuan	3
1.5. Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA	
2.1. PDAM (Perusahaan Daerah Air Minum).....	4
2.2. <i>Diagram Aliran</i> Data (DAD)	7
2.2.1. Kelebihan Pendekatan <i>Aliran</i> Data.....	8
2.2.2. Pengembangan <i>Diagram Aliran</i> Data	8
2.3. Basis Data (Database)	9
2.4. Basis Data Terdistribusi	11
2.4.1. Keuntungan dan Kerugian Basis Data Terdistribusi	12
2.4.2. Desain Basis Data Terdistribusi	13
2.5. Replikasi Basis Data	13
2.5.1. Model Replikasi	14
2.5.2. Jenis-jenis Replikasi	17
2.6. Windows 2000 Server	20
2.6.1. Fungsi Windows 2000 Server.....	21
2.6.1.1. <i>File Server</i>	21
2.6.1.2. <i>Application Server</i>	21
2.6.1.3. <i>Member Server</i>	22
2.6.1.4. <i>Domain Controller</i>	21
2.6.2. Fitur Baru Pada Windows 2000 Server	23
2.6.2.1. <i>Active Directory</i> Service	23

2.6.2.2.	<i>Group Policy</i>	23
2.6.2.3.	<i>Distributed File System</i>	24
2.6.2.4.	<i>Terminal Services</i>	24
2.7.	SQL Server	24
2.7.1.	Integrasi dengan Internet	25
2.7.2.	Skalabilitas dan Ketersediaan (<i>Scalability and Availability</i>).....	26
2.7.2.1.	Kemampuan <i>Database</i> Skala Besar	26
2.7.2.2.	<i>Query Optimizer</i>	27
2.7.2.3.	Dukungan Memori Berukuran Besar.....	28
2.7.3.	Fasilitas <i>Database Berskala Enterprise</i>	28
2.7.4.	Kernudahan Instalasi dan Penggunaan	28
2.7.5.	<i>Data Warehouse</i>	29
2.7.5.1.	<i>Data Warehousing Framework</i>	29
2.7.5.2.	<i>Data Transformation Services</i>	29
2.8.	TCP/IP (<i>Transmission Control Protocol/Internet Protocol</i>).....	30
2.8.1.	Dasar Arsitektur TCP/IP.....	31
2.8.2.	SLIP (<i>Serial Line Interface Protocol</i>) dan PPP (<i>Point to Point Protocol</i>)	34
2.8.2.1.	SLIP (<i>Serial Line Interface Protocol</i>)	34
2.8.2.2.	Dial-Up Networking dan PPP (<i>Point to Point Protocol</i>).....	33
2.9.	VPN (<i>Virtual Private Network</i>).....	36
2.9.1.	Komponen Jaringan VPN	36
2.9.2.	Jenis-jenis VPN	38
2.9.2.1.	VPN dengan <i>Point-to-Point Tunnelling Protocol</i> (PPTP)	38
2.9.2.2.	VPN dengan <i>Point-to-Point Tunnelling Protocol</i> (PPTP)	39
2.9.2.3.	VPN dengan <i>Secure Socket Layer Protocol/Transport Layer Security Protocol</i> (SSL/TLS)	41
2.9.3.	OpenVPN.....	41
2.9.2.	<i>Command</i> dan Konfigurasi pada OpenVPN.....	43
BAB III METODOLOGI		
3.1.	Studi Literatur.....	46
3.2.	Penentuan Spesifikasi Bahan dan Alat	46
3.3.	Perancangan Sistem.....	47
3.4.	Implementasi Sistem.....	47

3.5.	Metode Pengujian dan Analisis.....	47
3.5.1.	Pengujian Masing-masing Blok.....	47
3.5.2.	Pengujian Sistem Keseluruhan.....	47
3.5.3.	Pengujian Keamanan Data.....	48
3.6.	Pengambilan Kesimpulan.....	48
BAB IV PERANCANGAN		
4.1.	Spesifikasi Bahan dan Alat.....	49
4.1.1.	Perangkat keras (<i>Hardware</i>).....	49
4.1.2.	Perangkat lunak (<i>Software</i>).....	50
4.2.	Diagram Blok Sistem.....	50
4.3.	Cara Kerja Sistem.....	52
4.4.	Diagram Aliran Data (DAD) Sistem.....	53
4.5.	Perancangan dan Konfigurasi Sistem.....	55
4.5.1.	Perancangan Jaringan.....	55
4.5.1.1.	Topologi Jaringan.....	55
4.5.1.2.	Pengkoneksian VPN.....	57
4.5.2.	Perancangan <i>Database</i>	60
4.5.2.1.	Skema Tabel.....	60
4.5.2.2.	Penentuan <i>primary key</i>	62
BAB V IMPLEMENTASI		
5.1.	Konfigurasi <i>Dial Up Server (Server RAS)</i>	66
5.1.1.	Penjadwalan <i>Dial Up</i>	68
5.1.2.	Otomatisasi <i>Dial Up</i>	70
5.2.	Konfigurasi <i>Virtual Private Network (VPN)</i>	71
5.2.1.	Otomatisasi Koneksi VPN.....	76
5.3.	Konfigurasi Database.....	78
5.3.1.	<i>Restore Database</i>	78
5.3.2.	<i>Implementasi Replikasi</i>	79
5.3.2.1.	Skema Replikasi.....	79
5.3.2.2.	Konfigurasi Komputer Pusat.....	82
5.3.2.3.	Konfigurasi Komputer Cabang.....	84
5.3.2.4.	Registrasi <i>Remote SQL Server</i>	85
5.4.	Konfigurasi <i>Windows Authentication User</i>	87
5.5.	Prasyarat Parameter VPN.....	88

BAB VI PENGUJIAN DAN ANALISIS

6.1. Pengujian Per Blok 90

 6.1.1. Pengujian *Restore Database* 90

 6.1.2. Pengujian Koneksi Antar Komputer 92

 6.1.3. Pengujian Otomatisasi *Dial Up* 98

 6.1.4. Pengujian Otomatisasi Koneksi VPN 100

 6.1.5. Pengujian Replikasi 104

6.2. Pengujian Sistem secara Keseluruhan 107

6.3. Pengujian Koneksi *Database* pada Aplikasi Microsoft Access 111

6.4. Pengujian Keamanan Data 116

6.5. Pengujian analisis dari beberapa parameter dan keunggulannya 124

BAB VII KESIMPULAN DAN SARAN

7.1. Kesimpulan 132

7.2. Saran 132

DAFTAR PUSTAKA 133

LAMPIRAN :



DAFTAR GAMBAR

Gambar 2.1.	Empat simbol dasar aliran data.....	7
Gambar 2.2.a.	Lemari arsip di sebuah ruang.....	10
Gambar 2.2.b.	Basis data di sebuah harddisk.....	10
Gambar 2.3.	Model replikasi.....	14
Gambar 2.4.	Central publisher with separated distributor.....	15
Gambar 2.5.	Central subscriber with multiple publisher.....	16
Gambar 2.6.	Multiple publisher a d multiple subscriber.....	16
Gambar 2.7.	Ukuran halaman database pada SQL Sewer.....	27
Gambar 2.8.	Layer TCP/IP.....	32
Gambar 2.9.	Pergerakan data dalam layer TCP/IP.....	33
Gambar 2.10.	Point to Point Protocol.....	35
Gambar 2.11.	Blok Diagram VPN.....	3
Gambar 2.12.	Paket data pada PPP.....	39
Gambar 2.13.	paket data pada L2TP/IPSEC.....	40
Gambar 2.14.	Blok Diagram VPN menggunakan standar interface.....	42
Gambar 4.1.	Blok diagram contoh sistem dalam keadaan nyata (hanya diambil 5 cabang).....	50
Gambar 4.2.	Blok diagram sistem secara keseluruhan yang akan digunakan dalam riset.....	5 1
Gambar 4.3.	DFD level 0 proses replikasi melalui koneksi VPN.....	53
Gambar 4.4.	DFD level 1 dari proses Replikasi melalui Koneksi VPN.....	54
Gambar 4.5.	DFD level 2 dari proses Replikasi melalui Koneksi VPN.....	55
Gambar 4.6.	Topologi dan konfigurasi jaringan.....	56
Gambar 4.7.	Gambaran keadaan jaringan saat terkoneksi secara VPN.....	57
Gambar 4.8.	Komputer cabang melakukan Ping ke komputer pusat.....	58
Gambar 4.9.	Gambar setting konfigurasi dan key.....	59
Gambar 4.10.	Gambar start OpenVPN.....	59
Gambar 4.11.	Gambar status koneksi OpenVPN yang berhasil.....	60
Gambar 4.12.	Komputer cabang melakukan Ping IP VPN komputer pusat.....	60
Gambar 4.13.	Diagram relasi antara tabel di database PDAM.....	61
Gambar 4.14.	Normalisasi tabel pada database PDAM.....	61
Gambar 5.1.	Konfigurasi account pada komputer pusat.....	67



Gambar 5.2.	Koneksi untuk melayani <i>dial-up</i> pada komputer pusat	67
Gambar 5.3.	Urutan proses koneksi antara komputer cabang dengan pusat.....	68
Gambar 5.4.	Koneksi untuk melakukan <i>dial-up</i> pada komputer cabang	70
Gambar 5.5.	Penjadwalan proses pembuatan dan pemutusan koneksi <i>dial up</i>	71
Gambar 5.6.	Urutan proses koneksi antara komputer cabang dengan pusat	72
Gambar 5.7.	Gambar setting konfigurasi dan <i>key</i>	73
Gambar 5.8.	Gambar <i>start OpenVPN</i>	74
Gambar 5.9.	Gambar <i>window status</i> koneksi VPN pada komputer cabang.....	75
Gambar 5.10.	Gambar <i>window status</i> koneksi VPN pada komputer pusat.....	75
Gambar 5.11.	Penjadwalan proses pembuatan dan pemutusan koneksi VPN	77
Gambar 5.12.	Pengsetan waktu pemutusan koneksi VPN.....	78
Gambar 5.13.	<i>SQL Server Enterprise Manager</i>	78
Gambar 5.14.a.	Memilih <i>Restore Destination</i>	79
Gambar 5.14.a.	Lokasi <i>Backup File</i>	79
Gambar 5.15.	Proses replikasi antara komputer cabang.....	81
Gambar 5.16.	Konfirmasi penambahan kolom baru untuk replikasi	82
Gambar 5.17.	Konfigurasi pengaturan filter pada Tabel Pelanggan	83
Gambar 5.18.	Konfigurasi pengaturan filter <i>join</i> pada Tabel Pemakaian sebelum ada perubahan.....	83
Gambar 5.19.	Konfigurasi pengaturan filter <i>join</i> pada Tabel Pemakai yang berubah	84
Gambar 5.20.	<i>Initialize subscription</i> pada publikasi kedua	84
Gambar 5.21.	Konfigurasi komputer cabang sebagai <i>subscriber</i>	85
Gambar 5.22.	Konfigurasi <i>property subscriber</i>	85
Gambar 5.23.	Tampilan setelah registrasi SQL Server berhasil	86
Gambar 5.24.	Tampilan <i>window Computer Management</i>	87
Gambar 5.25.	Tampilan <i>window Administrators Properties</i>	87
Gambar 5.26.	Tampilan <i>window</i> anggota-anggota grup yang dapat dipilih.....	88
Gambar 5.28.	Capture packet data pada RAS saat Replikasi antara IP Komputer Database pusat dan komputer cabang.....	89
Gambar 5.29.	Replikasi yang berjalan antara komputer cabang dan komputer pusat	89
Gambar 6.1.	Macam-macam tabel pada database pada pusat.....	91
Gambar 6.2.	Data pada Tabel Pelanggan	92
Gambar 6.3.	Daftar koneksi komputer pusat sebelum melakukan koneksi dengan	



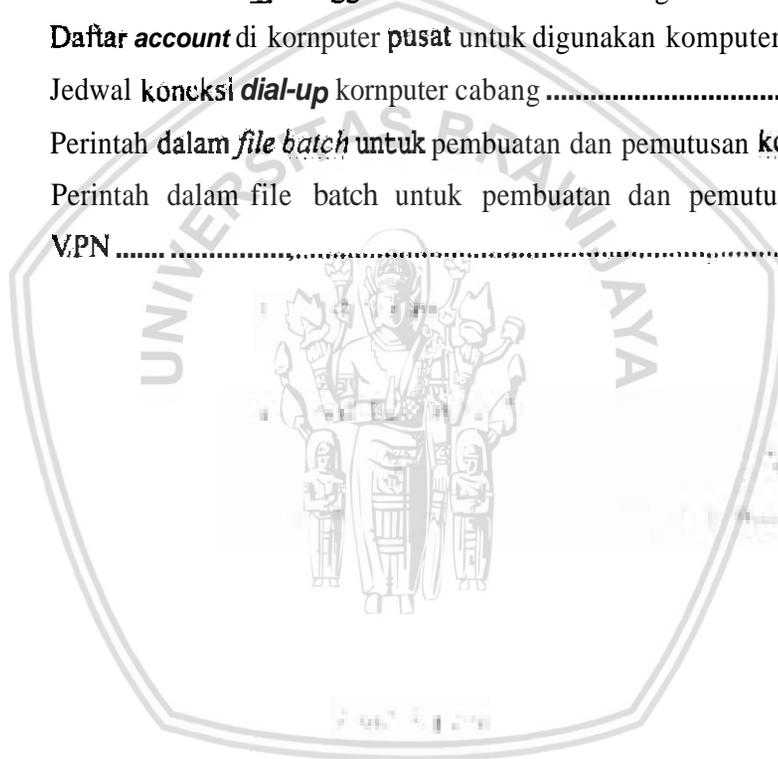
	<i>database SQL Server</i>	93
Gambar 6.4.	Daftar koneksi komputer cabang setelah melakukan koneksi dengan <i>database SQL Server</i>	95
Gambar 6.5.	Daftar koneksi komputer pusat setelah melakukan koneksi dengan <i>database SQL Server</i>	96
Gambar 6.6.	Tampilan <i>SQL Server</i> Enterprise Manager pada komputer pusat setelah koneksi.....	97
Gambar 6.7.	Proses koneksi terjadi secara otomatis	99
Gambar 6.8.	Komputer cabang sedang melakukan koneksi pada komputer server RAS	99
Gambar 6.9.	Proses pemutusan koneksi terjadi secara otomatis	100
Gambar 6.10.	Proses koneksi terjadi secara otomatis pada komputer pusat	102
Gambar 6.11.	Status koneksi VPN yang berhasil pada komputer pusat	103
Gambar 6.12.	Status koneksi VPN yang berhasil pada komputer cabang	103
Gambar 6.13.	Monitor replikasi pada <i>database sewer</i> komputer pusat	105
Gambar 6.14.	Tabel-tabel pada <i>database server</i> di komputer cabang	106
Gambar 6.15.	Data pada Tabel Pelanggan di <i>database sewer</i> komputer cabang	107
Gambar 6.16.	Proses koneksi terjadi secara otomatis di komputer cabang.....	109
Gambar 6.17.	proses replikasi berlangsung di komputer cabang 1.....	110
Gambar 6.18.	Proses replikasi berlangsung di komputer cabang 2.....	110
Gambar 6.19.	Properties file PDAM.adp.....	112
Gambar 6.20.	Koneksi berhasil	113
Gambar 6.21.	Tabel-tabel pada Microsoft Access	113
Gambar 6.22.	Tabel Bulan	114
Gambar 6.23.	Tampilan aplikasi Microsoft Access di PDAM pusat	115
Gambar 6.24.	Tampilan aplikasi Microsoft Access di kantor cabang 1	115
Gambar 6.25.	Tampilan aplikasi Microsoft Access di kantor cabang 2	116
Gambar 6.26.	Tampilan Konfigurasi seluruh koneksi pada komputer cabang	118
Gambar 6.27.	Tampilan hasil Ping komputer cabang ke komputer RAS	119
Gambar 6.28.	Tampilan hasil Piny komputer cabang ke komputer pusat	119
Gambar 6.29.	Tampilan hasil Ping komputer cabang ke IP VPN komputer pusat ...	119
Gambar 6.30.	Tampilan server <i>database</i> yang available untuk dikoneksikan.....	120
Gambar 6.31.	Tampilan <i>finishing</i> registrasi database	120
Gambar 6.32.	Tampilan pesan error setelah registrasi database	120

Gambar 6.33.	Tampilan hasil Ping komputer cabang ke IP VPN komputer pusat setelah terjadi VPN.....	121
Gambar 6.34.	Tampilan pesan <i>success</i> setelah registrasi <i>database</i>	121
Gambar 6.35.	Tampilan hasil <i>capture</i> ethereal pada komputer RAS saat replikasi <i>database</i>	122
Gambar 6.36.	Tampilan detail-detail hasil <i>capture</i> ethereal pada komputer RAS saat replikasi <i>database</i>	122
Gambar 6.37.	Tampilan hasil <i>capture</i> ethereal pada komputer cabang saat replikasi <i>database</i>	123
Gambar 6.38.	Tampilan detail-detail hasil <i>capture</i> ethereal pada komputer cabang saat replikasi <i>database</i>	1 2 3
Gambar 6.39.	Tampilan hasil pengukuran, menggunakan <i>tool</i> Iperf pada komputer cabang.....	126
Gambar 6.40.	Tampilan hasil pengukuran, menggunakan <i>tool</i> Iperf pada komputer pusat.....	126
Gambar 6.41.	Tampilan pengukuran waktu saat memulai replikasi	127
Gambar 6.42.	Tampilan pengukuran waktu saat akhir dari replikasi.....	127
Gambar 6.43.	Tampilan tabel-tabel yang direplikasikan.....	126
Gambar 6.44.	Tampilan hasil <i>capture</i> pada saat belum terjadi VPN	126
Gambar 6.45.	Tampilan hasil <i>capture</i> pada saat terjadi VPN	126



DAFTAR TABEL

Tabel 2.1.	Data golongan.....	5
Tabel 2.2.	Data unit	5
Tabel 2.3.	Data kelas unit	6
Tabel 2.4.	Contoh data pelanggan di kecamatan Pakis	6
Tabel 2.5.	Batas kenaikan tarif satuan meteran	6
Tabel 2.6.	Contoh data pemakaian di kecamatan Pakis.....	7
Tabel 2.7.	Parameter Command-command yang digunakan pada OpenVPN.....	43
Tabel 4.1.	Contoh data di kolom id_pelanggan pada Tabel Pelanggan.....	64
Tabel 4.2.	Data di kolom id_pelanggan setelah dikenakan fungsi substring.....	64
Tabel 5.1.	Daftar account di komputer pusat untuk digunakan komputer cabang..	66
Tabel 5.2.	Jadwal koneksi dial-up komputer cabang	70
Tabel 5.3.	Perintah dalam file batch untuk pembuatan dan pemutusan koneksi.....	71
Tabel 5.4.	Perintah dalam file batch untuk pembuatan dan pemutusan koneksi VPN	77



DAFTAR LAMPIRAN

Lampiran I KONEKSI SISTEM OPERASI.....LI-1
Lampiran II KONEKSI SQL SERVER LII-2
Lampiran III *SCEDULLING*..... LIII-3
Lampiran IV KONFIGURASI ADD SEKVER LIV-4
Lampiran V *PENAKTIFAN SERVER DAN RESTORE* BASIS DATA LV-5
Lampiran VI TES KONEKSI *DATABASE ACCESS*.....LVI-6
Lampiran VII KONFIGURASI DISTRIBUTOR..... LVII-7
Lampiran VIII PROSES REPLIKASI LVIII-8
Lampiran IX PROSES PUSH LIX-9
Lampiran X TRIGGER LX-10
Lampiran XI HASIL PENGUJIAN LXI-11



BAB I PENDAHULUAN

1.1. Latar Belakang

Sebagian besar aplikasi basis data menerapkan basis data terpusat. Basis data terpusat cocok diterapkan pada lingkungan yang terpusat dan jumlah client yang tidak terlalu banyak. Selain handal, basis data terpusat desainnya relatif sederhana. Untuk lingkungan yang tersebar dengan jumlah client yang besar, basis data terpusat mempunyai kelemahan, yaitu jika terjadi kegagalan pada basis data pusat maka seluruh sistem akan terganggu. Untuk mengatasi masalah ini dikembangkan basis data terdistribusi. Dengan basis data terdistribusi, kegagalan pada salah satu bagian tidak akan menyebabkan seluruh sistem terganggu. Selain itu, dengan didistribusikannya basis data ini, proses query menjadi lebih cepat karena beban dibagi kesejumlah basis data. Namun demikian, dalam penerapan basis data terdistribusi muncul sejumlah problem. Problem utama dalam basis data terdistribusi adalah peningkatan kompleksitas yang diperlukan untuk menjamin koordinasi yang baik antar simpul dalam sistem terdistribusi [FAT-041].

Dalam sejumlah kasus, beberapa korporasi yang memiliki banyak cabang menginginkan basis datanya pada setiap cabang dapat otonom dan secara berkala perubahan data di cabang dikirim ke kantor pusat melalui jaringan telepon dial-up dengan biaya komunikasi yang minimal. Perubahan basis data di cabang dikirim ke basis data pusat saat terkoneksi saja. Begitu juga sebaliknya saat di basis data pusat terjadi perubahan maka perubahan tersebut juga harus dikirim ke cabang saat terkoneksi. Dalam keadaan *offline* cabang harus tetap dapat melakukan transaksi. Kasus ini terjadi di PDAM Kabupaten Malang, Saat ini PDAM Malang sudah memiliki aplikasi basis data terpusat dan sudah digunakan lebih dari 10 tahun. Karena kebutuhan manajemen, pihak PDAM menginginkan mendistribusikan aplikasi basis datanya ke setiap cabang. Jika aplikasi basis data ini didistribusikan ke setiap cabang, maka akan muncul masalah saat mengintegrasikan data dari masing-masing cabang. Selain itu pihak PDAM menginginkan perubahan data di setiap cabang dikirim secara berkala agar dapat menekan biaya komunikasi.

Sebagai kelanjutan dari penelitian yang dilakukan oleh Husnul Khotimah Mahasiswa Teknik Elektrik Universitas Brawijaya Malang (0110630071) tentang "Implementasi Replikasi Basis Data Melalui Jaringan Telepon Pada PDAM Kabupaten

Malang" dan oleh Fika Hastarita **Rachman Mahasiswa Teknik Elektro Universitas Brawijaya Malang (0110630071)** tentang "**Pengelolaan Gudang Data untuk Sistem Replikasi melalui Jaringan Telepon pada PDAM Kabupaten Malang** ", maka timbul permasalahan baru yaitu **tentang pengelolaan data-data yang masuk pada kantor induk.**

Salah satu **alternatif** untuk **menyelesaikan permasalahan** yang **dihadapi** oleh PDAM kabupaten Malang ini adalah **menerapkan** replikasi basis data ttdistribusi menggunakan jaringan *Virtual Private Network* dengan proses update yang **terjadwal.**

1.2. Perumusan **Masalah**

Mengacu pada permasalahan yang telah diuraikan pada **latar** belakang, maka **rumusan** masalah dapat disusun sebagai **berikut:**

- Mengkonfigurasi OS (*Operating System*) agar dapat digunakan sebagai jaringan **Virtual Private Network (VPN).**
- Menentukan model, metode dan *type* replikasi yang sesuai untuk **kondisi** di PDAM kabupaten Malang.
- Memodifikasi basis data yang telah ada agar dapat mendukung proses **replikasi yang telah dipilih pada Microsoft SQL Server.**

1.3. Ruang Lingkup

Dalam **perencanaan** dan **pembuatan** sistem ini perlu **dilakukan** pembatasan masalah. Pembatasan masalah yang diajukan dalam penelitian ini **antara** lain:

- **Penelitian ini dilaksanakan** hanya pada **tataran uji laboratorium, karena** penelitian secara **langsung** dilapangan **membutuhkan biaya** yang besar dan **waktu yang lama.**
- Sentral telepon yang dipakai adalah Mini PABX (*Private Automatic Branch Exchange*) (**buatan Cina**) sebagai **pengganti jaringan** telepon TELKOM.
- Perangkat lunak *Database* menggunakan **Microsoft SQL Server 2000.**
- Perangkat lunak VPN menggunakan **OpenVPN 2.0.9.**
- **Sistem Operasi** yang digunakan adalah **Microsoft Windows Sewer 2000.**
- Mencakup **Cabang-cabang** di kabupaten Malang.



1.4. Tujuan

Tujuan penyusunan penelitian ini adalah mengimplementasikan replikasi basis data pada jaringan Virtual Private Network (VPN) dan mencoba mencari konfigurasi dan metode yang paling optimal untuk diterapkan di PDAM kabupaten Malang.

1.5. Sistematika Penulisan

Sistematika penulisan dalam penelitian ini sebagai berikut:

- BAB I** Pendahuluan
Memuat latar belakang, rumusan masalah, tujuan, batasan masalah, dan sistematika pembahasan.
- BAB II** Teori Penunjang
Membahas teori-teori yang mendukung dalam perencanaan dan pembuatan sistem.
- BAB III** Metode Penelitian
Berisi tentang metode penelitian dan perencanaan sistem serta pengujian.
- BAB IV** Perancangan dan Perealisasian Sistem
Perancangan dan perealisasian sistem replikasi basis data dengan menggunakan data PDAM Kabupaten Malang. Membahas tentang perencanaan dan pembuatan sistem.
- BAB V** Pengujian
Memuat hasil pengujian terhadap sistem yang telah direalisasikan.
- BAB VI** Kesimpulan dan Saran
Memuat kesimpulan dan saran-saran.

BAB II

TINJAUAN PUSTAKA

Dalam merencanakan dan merealisasikan replikasi basis data pada jaringan *Virtual Private Network* (VPN) dibutuhkan pemahaman tentang berbagai hal yang mendukung. Pemahaman ini akan bermanfaat untuk penerapan replikasi basis data dengan jaringan VPN pada PDAM kabupaten Malang. Pengetahuan yang mendukung perencanaan dan realisasi penelitian meliputi PDAM kabupaten Malang, teori dasar basis data, basis data terdistribusi, teknik replikasi basis data, Windows Server 2000 dan SQL Server 2000, TCP/IP.

2.1. PDAM (Perusahaan Daerah Air Minum)

Perusahaan Daerah Air Minum yang lebih dikenal dengan singkatan PDAM adalah perusahaan yang dikelola oleh daerah untuk menangani air minum untuk masyarakat. PDAM kabupaten Malang memiliki kurang lebih 66.000 pelanggan yang tersebar pada 22 unit. Unit-unit ini terletak dalam radius 56 km dari pusat kota Malang. (Data PDAM)

Untuk memperoleh data pemakaian air, pelanggan di unit-unit dilakukan pencatatan oleh petugas pencatat meter. Hasil data dari pencatatan meter tersebut akan diserahkan setiap tanggal 20 ke bagian Penerbitan Rekening di kantor pusat. Proses selanjutnya dilakukan pembuatan kwitansi pembayaran rekening air, dimana pembuatan kwitansi ini harus selesai pada tanggal 28 setiap bulannya, sehingga bagian Penerbitan Rekening mempunyai waktu sekitar 8 hari untuk menyelesaikan pekerjaan ini. Setelah semua kwitansi dicetak di kantor pusat, kwitansi tersebut didistribusikan kembali ke unit-unit untuk diserahkan ke pelanggan. Dari survey yang dilakukan di PDAM pusat saat ini, jumlah operator yang ada hanya 6 orang dengan jumlah workstation sebanyak 5 buah. Tentunya dengan kondisi ini, beban kerja dari masing-masing operator menjadi sangat berat.

Seringkali penerbitan kwitansi ini terlambat diakibatkan keterlambatan penyerahan hasil pencatatan meter dari unit-unit, mengingat letak yang tersebar dan jarak dari unit-unit ini ke kantor pusat relatif jauh untuk ukuran pengiriman data secara manual. Ada kalanya juga diakibatkan kerusakan pada sistem komputer.

Data-data pokok yang didapat dari PDAM pusat adalah data unit yang ditunjukkan dalam nama kecamatan, data pelanggan dan alamatnya yang didapat dari

data PDAM unit-unit, data macam-macam golongan untuk menentukan tarif dasar setiap bulan, data pemakaian untuk setiap bulan dalam satuan meteran. Data golongan sebagai bahan penentuan tarif dasar diperlihatkan pada Tabel 2.1.

Tabel 2.1. Data golongan

No	Kode	Nama Golongan	Kategori
1		Sosial Umum	Sosial
12		Rumah tangga	Non Niaga
13		Niaga Kecil	Niaga
14		Industri Kecil	Industri
21		Sosial Khusus	Sosial
22		Pemerintah	Non Niaga
23		Niaga Besar	Niaga
24		Industri Besar	Industri
25		ABRI	Non Niaga
33		Keseluruhan	

Sumber: Data PDAM

Data unit yang ditunjukkan dengan nama kecamatan diperlihatkan pada Tabel 2.2.

Tabel 2.2. Data unit

Kode	Nama Kecamatan	Kategori	Sub-kategori
A	NGAJUM	A1	Mai
B	KEPAKJEN	A1	Isih
C	LAWANG	A1	Hani
D	SINGOSARI	A1	Sulchah
E	TUREN	A1	Mai
F	DAMPIT	A1	Riyono
G	KARANGPLOSO	A2	Isih
H	DAU	A2	Mai
I	PAKSAJI	B1	Mai
J	TUMPANG	A2	Riyono
K	PLUON	B1	Isih
L	BULULAYANG	A2	Mai
M	GONDANGLEGI	A2	Riyono
N	PARIS	A2	Suparno
O	PONCOKUSUMU	A2	Riyono
P	TAJAMAN	A2	Suparno
Q	JABUNG	A2	Raspati
R	NGANTANG	B1	Raspati
S	AMPELGADING	B1	Sulchah
T	BANTUR	B2	Hani
U	DONDOMLYO	B2	Suparno
V	SBR. MANLING W	B2	Raspati

Sumber: Data PDA M

Tabel 2.3. Data kelas unit

A1	KLAS BESAR
A2	KLAS MENENGAH
B1	KLAS KECIL
B2	KLAS SEDANG

Sumber: Data PDAM

Tabel 2.3. menunjukkan data **kelas** unit sebagai acuan **kode_unit** untuk setiap kecamatann. **Tabel 2.4** menunjukkan **daftar** pelanggan dan alamat yang didapat dari masing-masing unit. Setiap unit akan memberikan data pelanggan yang **membayar** pad? unit tersebut pada setiap **bulan**, untuk **kemudian dikirim** ke **pusat** sebagai **bahan laporan** dan **perekapan** data pelanggan.

Tabel 2.4. Contoh data pelanggan di kecamatan Pakis

NO	NAMA	PAKIS JAJAR 1/1	NO	NO
1	HARI	PAKIS JAJAR 1/1	50	PA
2	TAKRIP	PJ 3/1	50	PA
3	KOASTARI	PAKIS JAJAR	50	PA
4	SLAMET	PJ 3/1	50	PA
5	SUMARIYAH	PJ 3/2	50	PA
6	SUTOMO	PJ 3/2	50	PA
7	SALI	PJ 3/3	50	PA
8	SATUKAN	PJ 1/2	50	PA
9	ARIFIN/PASAR PAKIS	PJ 4/2	50	PA
10	TAKRIP	PJ 3/1	50	PA
11	H. ZARNAL DUNNUJIN	PAKISJAJAR 3/1	50	PA
12	SARIP	PJ 3/1	50	PA

Sumber: Data PDAM

Data **tarif** setiap kenaikan **meteran** dapat **ditunjukkan** pada **Tabel 2.5.** sebagai **batas** kenaikan **tarif** setiap pemakaian per bulannya. Data **pemakaian** dapat **ditunjukkan** pada **Tabel 2.6.**

Tabel 2.5. Batas kenaikan tarif satuan meteran

NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
1	1500	10	20	30	1790.00	11.080.00	11.240.00	11.580.00	11.790.00	11.970.00
2	1500	10	W	30	1710.00	10970.00	11.100.00	11.500.00		

Sumber: Data PDAM



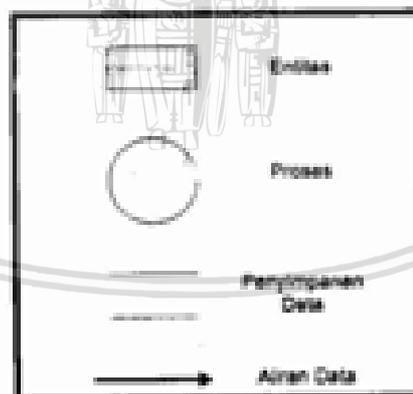
Tabel 2.6. Contoh data pemakaian di kecamatan Pakis

No. Pelanggan	Daerah	Tahun	Pembayaran (Rp)
1	1	2005	\$39.00
1	6	2004	\$23.00
1	7	2004	\$17.00
1	8	2004	\$20.00
1	9	2004	\$22.00
1	10	2004	\$36.00
1	11	2004	\$28.00
1	12	2004	\$34.00
2	1	2006	\$12.00
2	6	2004	\$14.00
2	7	2004	\$15.00
2	8	2004	\$16.00

Sumber: Data PDAM

2.2. Diagram Aliran Data (DAD)

Diagram Aliran Data atau dikenal juga sebagai *Data Flow Diagram (DFD)* adalah suatu teknik analisa data terstruktur dimana aliran data menekankan logika yang mendasari sistem. Dengan menggunakan kombinasi dari empat simbol dapat tercipta suatu gambaran proses-proses yang bisa menampilkan dokumentasi sistem yang solid. [KEN-03:263]



Gambar 2.1. Empat simbol dasar aliran data
Sumber: [KEN-03:265]

Sebagaimana ditunjukkan dalam Gambar 2.1. Suatu sistem secara keseluruhan dan beberapa subsistem bisa digambarkan secara grafis dengan kombinasi empat simbol dasar yang digunakan untuk memetakan gerakan diagram aliran data yaitu:

- kotak persegi panjang yang digunakan untuk menggambarkan suatu entitas eksternal (bagian lain, sebuah perusahaan, seseorang, atau sebuah mesin) yang dapat

mengirim data atau menerima data dari sistem. **Entitas eksternal**, atau hanya **entitas** disebut juga sumber atau tujuan data, dan dianggap **eksternal** terhadap sistem yang sedang **digambarkan**. Setiap **entitas** diberi label dengan sebuah nama yang sesuai. **Meskipun** berinteraksi dengan sistem, namun dianggap **diluar** batas-batas sistem. Entitas-entitas **tersebut** harus diberi nama dengan suatu kata benda. **Entitas** yang sama bisa digunakan lebih dari **sekali** atas suatu diagram aliran data **tertentu** untuk menghindari persilangan antara jalur-jalur aliran data.

- Tanda panah yang menunjukkan **perpindahan** data dari satu titik ke titik yang lain, dengan kepala **panah mengarah** ke tujuan **data**. Aliran data yang muncul secara simultan bisa **digambarkan** hanya dengan menggunakan **tanda panah paralel**. Karena sebuah tanda **panah** menunjukkan seseorang, **tempat**, atau **sesuatu** maka harus digambarkan **dalam** kata **benda**.
- Lingkaran dengan **sudut** membulat digunakan untuk menunjukkan adanya **proses transformasi**. Proses-proses **tersebut** selalu menunjukkan suatu perubahan data jadi aliran data yang meninggalkan suatu proses selalu diberi label yang berbeda **dari** aliran data yang masuk.
- Simbol terakhir yang digunakan **adalah** bujur sangkar dengan **ujung** terbuka (**tertutup** pada posisi sisi sebelah kiri dan terbuka pada sisi **sebelah** kanan) yang menunjukkan penyimpanan data-data yang memungkinkan penambahan dan perolehan data.

2.2.1. Kelebihan Pendekatan Aliran Data

Pendekatan aliran data memiliki empat kelebihan **utama** yaitu:

1. Kebebasan **dari** menjalankan **implementasi teknis** sistem yang terlalu dini.
2. Pemahaman lebih jauh mengenai **keterkaitan satu sama** lain **dalam** sistem dan subsistem.
3. **Mengkomunikasikan** pengetahuan sistem yang ada dengan **pengguna** melalui diagram aliran data.
4. **Menganalisis** sistem yang diajukan untuk menetapkan apakah data-data dan proses yang diperlukan sudah ditetapkan.

2.2.2. Pengembangan Diagram Aliran Data

Untuk memulai suatu diagram aliran data **diawali** dengan merangkum **narasi** sistem **organisasi** menjadi sebuah **daftar** dengan **empat** kategori yang terdiri dari **entitas**

eksternal, aliran data, proses, dan penyimpanan data. Daftar ini untuk membantu menentukan **batas-batas** sistem yang akan digambarkan. Begitu daftar unsur-unsur data dasar ini **tersusun**, maka penggambaran **aliran** data dapat dilakukan

- Menciptakan diagram **konteks**

Dengan pendekatan **atas-bawah** untuk membuat diagram **pengalihan data**, diagram **berganti** dari umum ke khusus. **Meskipun** diagram **pertama** membantu memahami **pengalihan** data, sifat umumnya membatasi **kegunaannya**. Diagram konteks awal harus **berupa** suatu pandangan yang mencakup masukan-masukan dasar, sistem umum dan keluaran. Diagram konteks adalah tingkatan **tertinggi** dalam diagram aliran data dan hanya **memuat** satu proses, menunjukkan sistem **secara keseluruhan**. Proses **tersebut** diberi nomor **nol (0)**. Semua **entitas** eksternal yang **ditunjukkan** pada diagram **konteks** berikut aliran **data-aliran**, data **utama** menuju **sistem** dan **dari** sistem.

- Menggambar diagram 0 (level berikutnya)

Diagram 0 adalah pengembangan **diagram** konteks dan bisa mencakup sampai sembilan proses. Setiap proses diberi **nomor** bilangan **bulat**, umumnya dimulai dari sudut sebelah kiri **atas** diagram dan **mengarah** ke sudut **sebelah kanan bawah**. Diagram 0, lebih mendetail **dibanding**, dengan diagram **konteks**, masukan dan keluaran yang ditetapkan **dalam** diagram yang **pertama** tetap konstan dalam semua diagram pengembangan **selanjutnya**.

- Menciptakan diagram anak (tingkat yang lebih mendetail)

Setiap proses **dalam** diagram 0 bisa dikembangkan untuk menciptakan diagram anak yang lebih mendetail. Semua **aliran data** yang **menuju** atau ke **luar dari proses induk** harus **ditunjukkan mengalir** ke dalam atau ke **luar** dari diagram anak.

2.3. Basis Data (*Database*)

Basis data (*database*) dapat **dibayangkan** sebagai sebuah **lemari** arsip. Jika sebuah lemari arsip akan dilakukan **pengaturan** dan **perawatan**, maka kemungkinan besar akan dapat dilakukan **hal-hal** seperti memberi **sampul/map** pada kumpulan arsip, memberi penomoran **dengan** pola **tertentu** yang nilainya unik pada setiap **sampul**, lalu **menempatkan** arsip-arsip **tersebut** dengan **cara/urutan** tertentu di dalam lemari.

Basis data terdiri **atas** 2 kata, yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai **markas** atau **gudang tempat** berkumpul. Sedangkan data adalah **representasi** fakta dunia nyata yang **mewakili** suatu objek seperti manusia (pegawai,

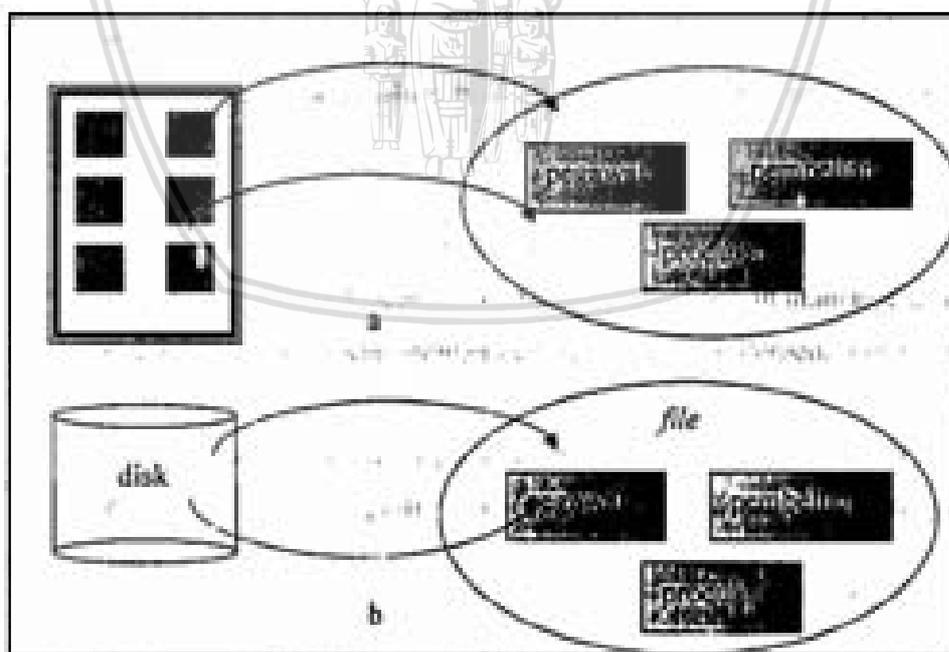
siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya. ,

Basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti:

- Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
- Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
- Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

[FAT-02:2]

Basis data dan lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengaturan data/arsip. Dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data/arsip. Perbedaannya hanya terletak pada media penyimpanan yang digunakan. Jika lemari arsip menggunakan lemari dari besi, atau kayu sebagai media penyimpanan, maka basis data menggunakan media penyimpanan elektronik seperti disk, (disket atau *harddisk*) seperti yang diperlihatkan dalam Gambar 2.2.



Gambar 2.2.a) Lemari arsip di sebuah ruang b) Basis data di sebuah *harddisk* [FAT-02:3]

Suatu hal yang juga harus diperhatikan, bahwa basis data bukan hanya sekedar penyimpanan data secara elektronik (dengan bantuan komputer). Artinya tidak semua bentuk penyimpanan data secara elektronik bisa disebut basis data. Dokumen berisi data dapat disimpan dalam file teks (dengan program pengolah kata), file *spread sheet*, dan lain-lain, tetapi tidak bisa disebut sebagai basis data. Karena di dalamnya tidak ada pemilahan dan pengelompokan data sesuai jenis/fungsi data, sehingga akan menyulitkan pencarian data kelak. Yang sangat ditonjolkan dalam basis data adalah pengaturan/pemilahan/pengelompokan/pengorganisasian data yang akan disimpan sesuai fungsi/jenisnya. Pemilahan/pengelompokan/pengorganisasian ini dapat berbentuk sejumlah jileltabel terpisah atau dalam bentuk pendefinisian kolom-kolom/*field-field* data dalam setiap file/tabel.

Operasi-operasi dasar yang dapat dilakukan berkenaan dengan basis data dapat meliputi:

- Pembuatan basis data baru (*create database*), yang identik dengan pembuatan lemari arsip yang baru.
 - Penghapusan basis data (*drop database*), yang identik perusakan lemari arsip (sekaligus beserta isinya, jika ada).
 - Pembuatan file/tabel baru ke suatu basis data (*create table*), yang identik dengan penambahan map arsip baru ke sebuah lemari arsip yang telah ada.
 - Penghapusan file/tabel dari suatu basis data (*drop table*), yang identik dengan perusakan map arsip lama yang ada di sebuah lemari arsip.
 - Penambahan/pengisian data baru ke sebuah file/tabel di sebuah basis data (*insert*) yang identik dengan penambahan dengan lembaran arsip ke sebuah map arsip.
- Pengambilan data dari sebuah jileltabel (*retrieve/search*) yang identik dengan pencarian lembaran arsip dari sebuah map arsip.
- Pengubahan data dari sebuah file/tabel (*update*), yang identik dengan perbaikan isi lembaran arsip yang ada di sebuah map arsip.
 - Penghapusan data dari sebuah file/tabel (*delete*) yang identik dengan penghapusan sebuah lembaran arsip yang ada di sebuah map arsip.

2.4. Basis Data Terdistribusi

Untuk pengimplementasian basis data yang berhubungan dengan jaringan ada beberapa cara yaitu basis data terpusat dan basis data terdistribusi. Pengertian dari

masing-masing sistem tersebut adalah jika pada sistem basis data **terpusat**, data ditempatkan di satu lokasi saja dan **semua** lokasi lain mengakses basis data di lokasi tersebut. **Sementara** pada sistem basis data terdistribusi, data ditempatkan di **banyak** (lebih dari satu) lokasi, tetapi **menerapkan** suatu mekanisme **tertentu** untuk membuatnya menjadi satu **kesatuan** basis data. Sistem terdistribusi berbeda dengan sistem **terpisah** (*isolated*). Dalam sistem terpisah, **basis** data ditempatkan di **banyak** lokasi tetapi **tidak** saling berhubungan sama **sekali**. Tentu saja, penerapan sistem basis data terdistribusi ini akan memunculkan keuntungan-keuntungan **sekaligus** problem-problem **baru** dalam pengoperasiannya.

2.4.1 Keuntungan dan Kerugian Basis Data Terdistribusi

Penerapan **sistem** basis data terdistribusi yang baik dan benar akan menghasilkan keuntungan-keuntungan **berikut** ini:

- Pembagian (pemakaian **bersama**) data dan kontrol yang tersebar.
Setiap **user** pada suatu lokasi (simpul) dapat mengakses data yang **berada** di lokasi lain, ya, sama halnya dengan **user-user** pada lokasi **tempat** data tersebut berada. Di sisi lain, **pengontrolan/pengelolaan** basis data di setiap lokasi **dilakukan secara** sendiri-sendiri (karena setiap lokasi/simpul memiliki **DBMS (Database Management Sistem)** sendiri) yang **tentu** saja akan **mengurangi jumlah/besar** data yang **dikelola** di tiap lokasi.
- Krhandalan dan ketersediaan
Jika ada **sebuah simpul mengalami kerusakan**, simpul/lokasi yang lain akan tetap dapat beroperasi. Apalagi jika **di dalam sebuah sistem terdistribusi digunakan** mekanisme **replikasi (penduplikasian data antar simpul)**, maka ketersediaan data akan **semakin** tinggi.
- Kecepatan **query**
Jika sebuah **query melibatkan data di sejumlah lokasi/simpul**, maka **query** tersebut dapat dipilah ke **sejumlah subquery** yang akan **dijalankan di simpul-simpul** (lokasi-lokasi) yang bersesuaian. Hal ini **berdampak** pada kecepatan dalam mendapatkan **hasil query**.

Sedangkan **kelemahan** utama sistem basis data terdistribusi terletak pada meningkatnya kompleksitas yang diperlukan untuk **menjamin** koordinasi yang baik di antara **simpul-simpul** (lokasi-lokasi) yang **terlibat**. Peningkatan kompleksitas ini berbentuk/berakibat:

- Biaya pembangunan perangkat lunak
Implementasi sistem basis data terdistribusi tentu akan lebih sukar, **sehingga** perlu biaya lebih besar.
- Potensi bug (sumber **kesalahan** program) yang lebih **besar/banyak**
Karena **simpul-simpul** (lokasi-lokasi) dalam sistem basis data terdistribusi beroperasi secara paralel, maka akan lebih sulit **menjamin** kebenaran algoritma/program.
- **Peningkatan waktu** proses (overhead)
Waktu untuk **pertukaran** data dan tambahan komputasi yang diperlukan untuk **mengupayakan** koordinasi **antar simpul** merupakan **beban** tambahan (*overhead*) yang tidak **dijumpai** dalam sistem **terpusat**.

2.4.2 Desain Basis Data **Terdistribusi**

Ada beberapa **pendekatan** yang **berkaitan** dengan **penyimpanan data/tabel** dalam sebuah sistem basis data **terdistribusi** yaitu:

- **Replikasi**
Sistem **memelihara** sejumlah **salinan/duplikat tabel/tabel** data. Setiap salinan **tersimpan** dalam **simpul/lokasi** yang berbeda, yang menghasilkan replikas data.
- **Fragmentasi**
Data dalam **tabel** dipilah dan disebar ke dalam sejumlah **fragmen**. **Tiap fragmen** disimpan di sejumlah **simpul/lokasi** yang berbeda-beda. **Fragmentasi data ini** dapat berbentuk **fragmentasi horizontal (pemilahan record** data) atau **fragmentasi vertikal (pemilahan field/atribut** data).
- **Replikasi dan fragmentasi**
Merupakan kombinasi **dari kedua hal sebelumnya**. **Data/tabel** dipilah dalam sejumlah **fragmen**. **Sistem lalu mengelola** sejumlah salinan dari masing-masing **fragmen tadi** di sejumlah **simpul/lokasi**.

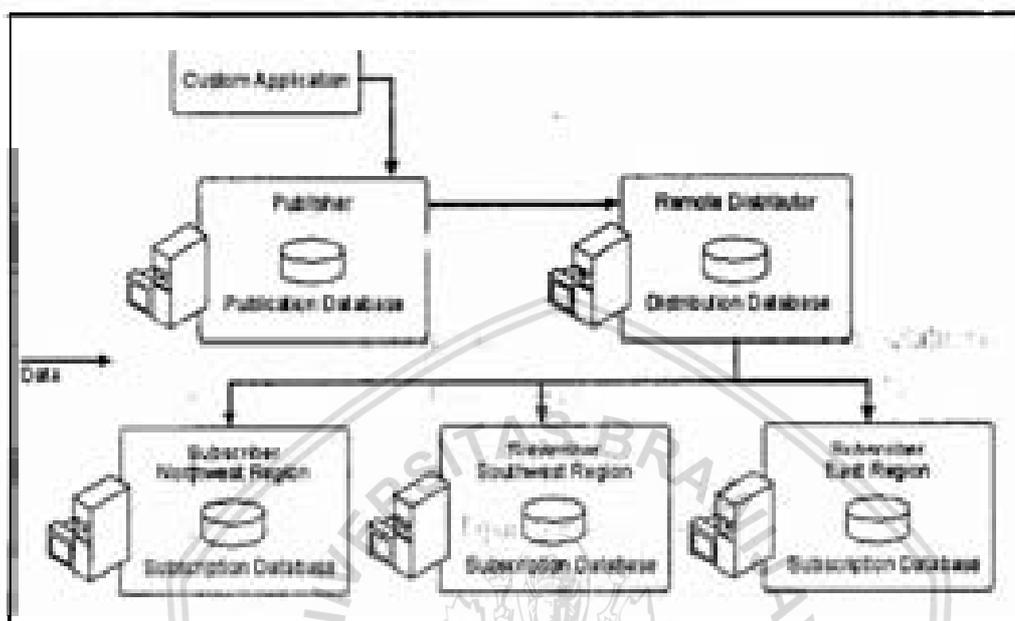
2.5. Replikasi Basis Data

Replikasi **adalah** teknologi yang **sangat penting** di dalam **lingkungan** perusahaan-perusahaan yang **umumnya** mempunyai **perusahaan** unit atau **perusahaan anak/cabang**. Replikasi memungkinkan basis data dan prosedur didistribusikan ke **seluruh** perusahaan. Data di dalam basis data bisa **digandakan** dan di-copy ke beberapa **tempat**

di dalam perusahaan. Basis data **terdistribusi** ini bisa disinkronkan agar **selalu memiliki nilai** yang sama, seperti misalnya **daftar harga** yang disinkronkan pada seluruh cabang.

2.5.1. Model Replikasi

Model **replikasi secara umum** dapat **dilihat dalam** Gambar 2.3.



Gambar 2.3. Model replikasi [ANO-00]

Komponen utama dalam replikasi adalah :

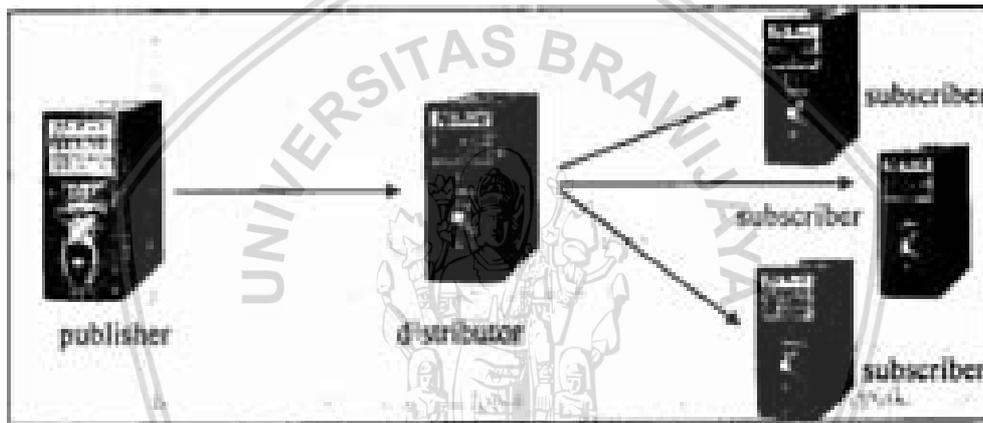
1. **Publisher** adalah *server yang menjamin* keberadaan data pada *server-server* lain dalam replikasi. Sebagai tambahan untuk **mengidentifikasi** data yang akan direplikasi, **publisher mengenali** data yang **telah diubah** dan **menjaga informasi tentang** seluruh publikasi dalam *server*. **Elemen** data apapun yang direplikasi memiliki **satu publisher**.
2. **Distributor** adalah *server* yang mendistribusikan **database**, menerima semua data yang direplikasi dari **publisher** untuk dipublikasikan ke **subscriber**.
3. **Pelanggan (subscriber)** adalah *server yang* menyimpan **replika-replika** dan menerima **perubahan-perubahan**.
4. **Publikasi** adalah suatu **koleksi dari** satu atau lebih artikel, dan suatu artikel adalah suatu **kelompok data yang akan direplika**, suatu artikel dapat **berupa** sebuah **tabel** dengnn hanya beberapa **kolom** atau hanya beberapa **baris**.
5. **Artikel** adalah objek-objek **yang disertakan** di dalam publikasi, seperti misalnya **tabel** atau **baris**.

6. *Subscription* adalah proses berlangganan antara *server* dan pelanggan (*subscriber*).

Replikasi mempunyai beberapa model yaitu:

1. *Publisher* terpusat dengan *Distributor* tersebar (*central publisher with separated distributor*)

Model replikasi ini hanya terdapat satu **pusat server publisher** dimana data utama disimpan dan beberapa *subscriber* yang menginginkan **salinan** data dan akan melakukan proses replikasi. **Pada** model ini replikasi bisa dilakukan dengan menggunakan tipe *transactional* maupun *snapshot*. Apabila tipe replikasi *merge* digunakan maka setiap **perubahan** data yang terjadi di *publisher* dan *subscriber* akan disimpan kemudian didistribusikan oleh SQL Server. Model replikasi ini diperlihatkan dalam Gambar 2.4.



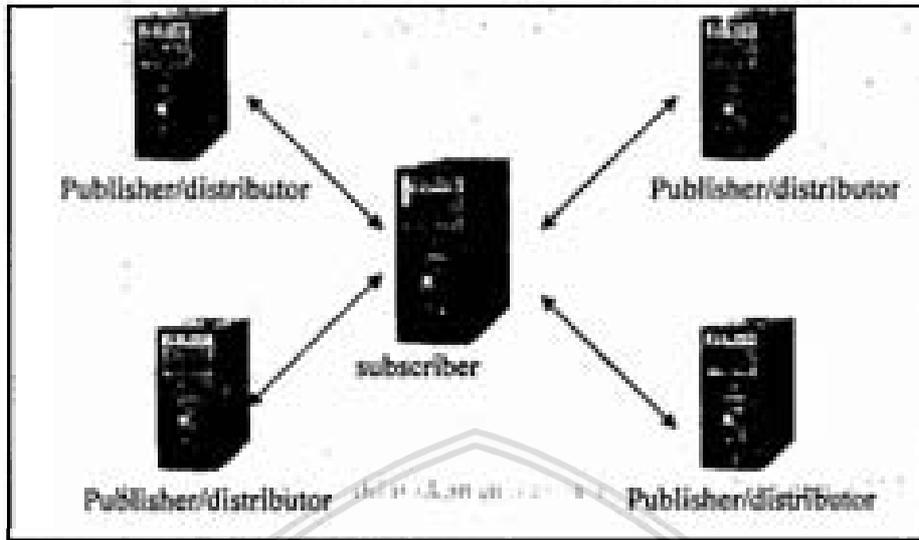
Gambar 2.4. Central publisher with separated distributor [PST-00,322]

2. *Subscriber* terpusat dengan banyak *publisher* (*central subscriber with multiple publisher*)

Model replikasi ini sesuai bila diterapkan dalam tipe replikasi *transactional* maupun tipe replikasi *merge*, karena kedua tipe replikasi ini hanya melakukan replikasi terhadap data yang **berubah**. **Selama** proses **sinkronisasi** hanya data yang **mengalami** perubahan yang akan **direplikasi sehingga** akan menghemat **bandwidth** keseluruhan yang digunakan dibandingkan tipe replikasi *snapshot*. Untuk tipe replikasi *transactional* masing-masing *publisher* sudah mempunyai data sendiri dan hanya bisa dikirim dari *publisher* ke *subscriber*, **tidak** bisa dilakukan **sebaliknya** jadi proses *update* data hanya bisa dilakukan satu **arah saja**. **Sedangkan** untuk tipe replikasi *merge*, *publisher* mempublikasikan data dan bisa *update* dari *subscriber*



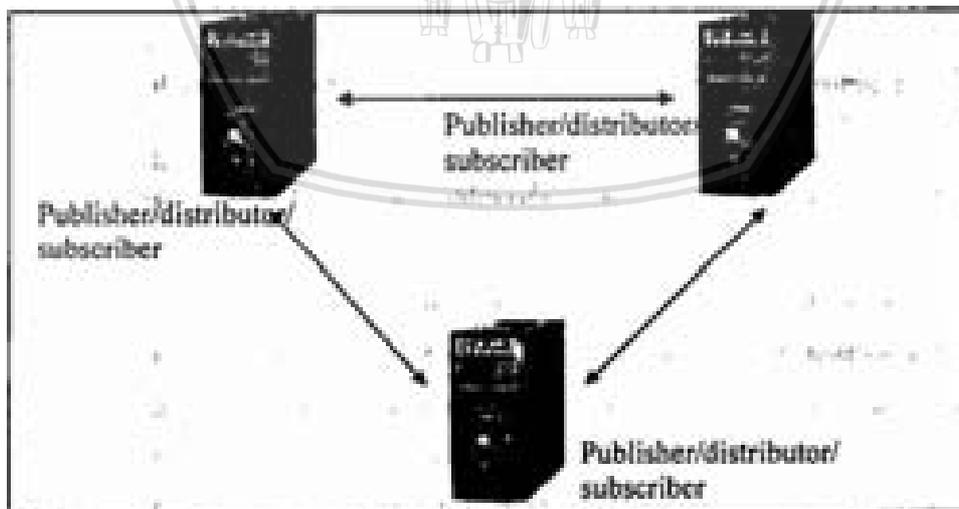
sehingga pengiriman data bisa dilakukan dua arah Gambar 2.5, menunjukkan model replikasi ini:



Gambar 2.5. *Central subscriber with multiple publisher* [LIN-01:401]

3. Banyak *publisher* dan banyak *subscriber* (*multiple publisher and multiple subscriber*)

Model replikasi ini sesuai dengan tipe replikasi *merge* karena proses publikasi dimodifikasi di masing-masing *publishing server* untuk selanjutnya data hasil modifikasi dapat dikirimkan ke komputer lain yang bertindak sebagai *subscriber* maupun *distributor*. Apabila dilakukan proses *update* data di komputer *subscriber* maupun *distributor* bisa dikirimkan ke komputer *publisher* sebagai proses *update* dua arah. Model ini dapat dilihat dalam Gambar 2.6.



Gambar 2.6. *Multiple publisher and multiple subscriber* [PET-00:517]



2.5.2. Jenis-jenis Replikasi

1. Snapshot

Replikasi snapshot membuat "foto" yaitu **tiruan persis dari** basis data yang direplikasi kepada para pelanggannya, yang akan menerima **salinan lengkap** dari data dan bukan hanya sekedar perubahan. Replikasi ini **sempurna** apabila basis data tidak diubah **terus-menerus**, dan bagi user yang tidak **ingin** meng-update table.

Metode paling **sederhana** untuk **diatur**, dan mungkin juga paling **mudah** untuk dimengerti, replikasi **snapshot** digunakan dengan **cara secara** periodik mengirim data dalam format tertentu. Anda akan menggunakan **metode** ini ketika server **tujuan** mendukung read-only environment dan ketika tidak **terjadi** update data.

Melakukan replikasi snapshot **tanpa** update data pada **periode waktu** tertentu disebut dengan 'latency'. Contohnya, pada **kasus** toko retail sepatu yang **digunakan** pada contoh sebelumnya, toko menggunakan replikasi snapshot **sebagai** bagian dari proses menjaga **keakuratan inventaris** sepatu di **gudang**. Inventaris bisa di **cek mingguan, atau** bulanan, toko **retail** ini dapat **mereplikasi** snapshot **tanpa** melakukan update pada server pusat selama beberapa **hari**. Skenario ini memiliki **tingkat** latency yang tinggi **serta** merupakan contoh yang **bagus** untuk **aplikasi replikasi snapshot**.

Salah satu **alasan** menggunakan tipe ini adalah akibat **rendahnya** koneksi bandwidth. Replikasi snapshot akan **memakan waktu** yang **cukup lama**, namun dapat **dijadwalkan** ketika jaringan sedang **tidak/jarang** digunakan atau **aktivitasnya** sedang rendah. Bila server **tujuan** bisa bertahan **tanpa** melakukan update, **metode** ini menyediakan **solusi** dengan biaya **rendah** dibandingkan dengan **metode** yang lain.

Replikasi snapshot **muncul** hampir di **setiap** skenario **replikasi**, karena SQL Server menggunakan replikasi **snapshot** untuk **membuat penggandaan pertama kali** pada server **tujuan**. Pada bab **selanjutnya**, kita akan belajar bagaimana SQL Server menginisiasi replikasi. Replikasi **snapshot** juga memiliki keuntungan lain, yaitu menjadi satu-satunya **metode** yang tidak **memerlukan tabel** hasil replikasi untuk memiliki primary key.

Replikasi snapshot **bekerja** dengan cara **membaca** database yang **di-publish** dan menciptakan file **dalam** direktori **kerja** pada distributor. File ini **disebut** dengan snapshot file **dan** mengandung data dari database yang telah **di-publish** sebelumnya **beserta** dengan informasi tambahan lainnya yang akan membantu pembuatan replikasi awal pada server **tujuan**. SQL Server **menyimpan konfigurasi** dan informasi status

dalam database distribusi, namun menyimpan **semua** data aktual dalam file snapshot.

[LIN-01]

2. Transactional

Replikasi transactional **mengijinkan** replikasi **tabel** dan **prosedur**. Replikasi ini **mengijinkan** penyaringan data yang akan dipublikasikan. Replikasi ini **menggunakan file log** untuk menyimpan perubahan yang dilakukan pada **artikel (misalnya tabel)** semenjak publikasi **terakhir, memantau perintah INSERT, UPDATE, dan DELETE**.

Dapat disebut **juga** sebagai **kebalikan** dari replikasi snapshot, replikasi transaksi bekerja dengan **mengirimkan** perubahan pada tujuan **secara** aktual. Seperti yang kita ketahui bersarna bahwa SQL Server memproses semua **kegiatan** yang terkait **dengan** database **menggunakan Transact-SQL statements**. Setiap statement disebut sebagai transaksi. Dalam replikasi transaksi, **setipa** transaksi direplikasi **setiap kali ia muncul**. Anda bisa **mengontrol** proses replikasinya sehingga proses dapat **mengakumulasikan** transaksi yang terjadi dan **mengirimkan** mereka **scara berkala** sesuai interval waktu tertentu, atau **mengirimkan** semua perubahan yang terjadi. Anda **menggunakan replikasi** tipe ini pada lingkungan yang **memiliki tingkat** latency yang **rendah** serta memiliki koneksi bandwidth yang **tinggi**. Replikasi ini **memerlukan** koneksi yang **reliable** dan **berkesinambungan**, sebab log transaksi akan **bertambah** dengan cepat bila server tidak mampu terhubung dengan replikan sehingga **besar kemungkinan akan terjadi kesimpang siuran**.

Replikasi transaksi dimulai dengan **snapshot** yang mengatur **replikasi pertama**. Replikasi pertama, tersebut kemudian akan **di-update** oleh transaksi yang telah digandakan.. Anda bisa memilih **seberapa sering meng-update** snapshot, atau memilih untuk tidak **meng-update snapshot** setelah replikasi pertama. Setelah **snapshot pertama** berhasil dilakukan, **replikasi transaksi menggunakan** agen pembaca **log** untuk **membaca** log transaksi dari database yang **telah dipublish** dan **menyimpan transaksi baru** pada database distribusi. Agen **distribusi kemudian mentransfer** transaksi-transaksi yang terjadi dari server **pusat menuju** server **tujuan [LIN-01]**.

3. Merge

Replikasi bertipe **merge** **mengendalikan** perubahan pada basis data **sumber** dan **mensinkronkan nilai-nilai** antara publisher (penerbit) dan subscriber. **Perubahan** yang dibuat basis data target baik pada **publisher** maupun subscriber akan **ikut mengubah**

basis data sumber dan **sebaliknya**. Untuk mempublikasikan sebagian data, perlu dibuat publikasi dengan **memilih tabel dan prosedur, serta ketersediaannya** kepada para *subscriber*. **Objek-objek** yang disertakan di dalam publikasi, seperti misalnya **tabel atau baris** disebut dengan istilah *artikel*. Prosedur dapat **dianggap** sebagai artikel dari replikasi **bertipe merge**.

Untuk menerima publikasi, **diperlukan proses** berlangganan kepada publisher, dan dilakukan penentuan basis data yang akan menerima publikasi ini. Ada dua **jenis publikasi pull dan push**. Langganan **bertipe push** dijalankan apabila subscriber **dikelola** secara terpusat. **Dalam hal ini**, salinan dari publikasi akan dikirim atau didorong dari publisher kepada para subscriber. **Keunggulan dari tipe push adalah** sistem keamanan yang tinggi. Langganan **bertipe pull** dijalankan apabila subscriber **dikelola** secara tidak terpusat. Tipe ini menarik salinan dari publikasi dari *editor*. **Setelah dikonfigurasi**, tabel publikasi dan **tabel target** harus disinkronkan agar proses **kontrol publikasi** bisa dijalankan dengan benar.

Karena metode ini **mengijinkan data bergerak** dua arah, agen penggabungan akan meng-kopi perubahan dari publisher dan mengaplikasikannya pada tujuannya, **setelah** perubahan dapat di akomodasi, agen penggabungan akan **melihat dan memecahkan konflik** yang mungkin terjadi. Setiap perubahan pada **setiap server** disimpan pada **databas e distribusi [LIN-0]**.

Ada **sejumlah keuntungan dan kerugian** yang bisa diperoleh dari, **penerapan metode replikasi data**:

- **Ketersediaan yang tinggi (*availability*)**
Jika karena suatu sebab sebuah simpul yang berisi **tabel r mengalami kerusakan**, maka **tabel yang sama** masih dapat diperoleh dari simpul yang lain. **Dengan begitu**, sistem tersebut masih dapat **melanjutkan proses query** yang melibatkan **tabel r** itu.
- **Peningkatan keparalelan (*increased parallelism*)**,
Pada kasus di mana **pengaksesan ke tabel r** pada umumnya hanya berupa proses pembacaan data, maka **pemrosesan query** pada **simpul-simpul (lokasi-lokasi)** yang melibatkan **tabel r** tersebut dapat **dieksekusi secara paralel (bersamaan)**.
- Peningkatan **beban** perubahan data (*increased overhead on update*)
Sistem harus dapat menjaga **konsistensi semua salinan** dari **tabel r** tersebut. Artinya jika **tabel r diubah**, maka perubahan tersebut harus **dijalarkan ke semua lokasi** yang

memiliki salinan **tabel r** tersebut. Akibatnya **beban** proses pengubahan data menjadi meningkat.

Secara umum, replikasi akan memperbaiki **performansi** dan operasi **query** (pembacaan data) dan meningkatkan ketersediaan data **khususnya untuk** transaksi-transaksi pembacaan **saja (read only)**. **Sebaliknya**, transaksi perubahan data akan berlangsung lebih lama dan sukar. Mengendalikan **persaingan** perubahan data oleh sejumlah **transaksi** ke data yang **tereplikasi** akan menjadi jauh lebih sukar dari pada menggunakan pendekatan **tersentralisasi**.

[FAT-04:224]

2.6. Windows 2000 Server

Windows 2000 Server merupakan **Network Operating Sistem (NOS)** untuk melakukan konfigurasi dan manajemen jaringan baik **skala kecil, menengah, maupun** besar. **Teknologi** sistem operasi Windows 2000 sebenarnya merupakan **kelanjutan** teknologi Windows NT yang **telah cukup lama digunakan** secara luas di **pasaran**. Keluarga Windows 2000 terdiri **dari 4 jenis sistem operasi**, 3 diantaranya merupakan sistem operasi untuk **server** yaitu Windows 2000 **Server**, Windows 2000 **Advance Server**, Windows 2000 **Data Center Server**, dan 1 untuk **workstation** yaitu Windows 2000 **Professional**. [AMR-03]

Windows 2000 Server ini merupakan **kelanjutan** teknologi Windows NT Server 4.0 dengan berbagai fasilitas **baru** yang semakin **memudahkan** **pengelolaan** jaringan. Keluarga **server** Windows 2000 terdiri dari 3 jenis yaitu versi standar (**server**), **Advance Server**, dan **Data Center Server**. Windows 2000 Server memiliki **semua** kemampuan yang ada pada **versi Professional** ditambah berbagai fasilitas inti, yang **dibutuhkan** sebagai **server jaringan**. **Versi ini dapat digunakan** sebagai **file dan print server, application server, web server, maupun communication server**. Fasilitas penting yang dimiliki versi ini **antara lain**:

- Dukungan untuk penggunaan **2 processor** bila diinstal dengan **mode clean install**, atau **4 processor** apabila instalasi dilakukan dengan **mengupgrade** Windows NT Server:
- Active Directory Service** untuk **memudahkan** **pengelolaan sumberdaya dan objek jaringan**.
- Sistem keamanan jaringan**, menggunakan **Kerberos dan public key infrastructure**

- *Internet* Connection Sharing.
- Web Server dengan menggunakan *Internet* Information Services versi 5.0.
- Windows Terminal Services untuk memudahkan **administrasi** jaringan dan **pemanfaatan** hardware komputer lama sehingga dapat digunakan untuk berbagai aplikasi **baru**.
- **Dukungan** penggunaan RAM (Random Access *Memory*) **hingga** 4 GB

2.6.1 Fungsi Windows 2000 Server

Sebuah server dapat menjalankan berbagai fungsi sesuai kebutuhan **bisnis**. Pada organisasi **skala** kecil fungsi-fungsi **tersebut** dapat digabungkan dalam **satu** server dan satu komputer. Untuk organisasi **besar**, **sebaiknya** setiap fungsi **dijalankan** pada **server** **terpisah** sesuai dengan **bebar** kerjanya.

2.6.1.1. File Server

Fungsi ini merupakan penggunaan paling umum dari sebuah server, dimana server digunakan sebagai **pusat penyimpanan file** dalam sebuah jaringan. Dengan sistem ini sistem **file** akan lebih terintegrasi sehingga memudahkan manajemen dan **pencarian file**. Sistem back up dan **penyimpanan file** juga dapat dilakukan dengan **lebih** baik.

Windows 2000 Server memiliki fasilitas distributed **file** sistem untuk memudahkan pengelolaan **file** dalam jaringan. Dengan sistem ini pengguna jaringan dapat dengan mudah menggunakan dan menyimpan **file** tanpa perlu mengetahui letak sebenarnya dari suatu **file**.

2.6.1.2. Application Server

Apabila server digunakan untuk menyimpan dan menjalankan suatu program aplikasi, maka server **tersebut** bertindak sebagai application server. Aplikasi diinstal di server dan **dijalankan** atau **diakses** oleh klien. Dengan demikian **aplikasi** tidak perlu diinstal di klien sehingga **memudahkan** proses implementasi dan perawatan sistem. Windows Terminal Services merupakan fasilitas untuk **memudahkan** penggunaan Windows 2000 Server sebagai **application** server. Dalam **layer** TCP/IP application server terletak pada **layer** aplikasi, yang **membedakan** adalah **protokol** yang digunakan dalam **layer** tersebut. Hal ini bisa dilihat dari contoh berikut:

Web Server

Web Server merupakan komputer yang digunakan sebagai *host* berbagai aplikasi web baik dalam lingkungan internet maupun intranet. *Internet information Service 5.0* merupakan komponen Windows 2000 Server untuk memudahkan konfigurasi dan manajemen *web site*. Protokol yang digunakan dalam layer aplikasi di TCP/IP untuk *web server* adalah HTTP (*Hypertext Transfer Protokol*) untuk melayani permintaan (*request*) halaman-halaman HTML (*Hypertext Markup Language*).

- *E-Mail Server*

Windows 2000 Server dapat juga digunakan sebagai *E-Mail server* dengan menggunakan berbagai *software* tambahan antara lain *Microsoft Exchange*, *Lotus Notes*, maupun *MDaemon*. Fungsi *E-Mail server* dapat dianalogikan dengan kantor pos dalam sistem surat menyurat konvensional. Untuk *E-Mail server*, protokol yang digunakan dalam layer aplikasi di TCP/IP adalah SMTP (*Simple Mail Transport Protokol*) untuk mengirim pesan/e-mail, sedangkan untuk mengaksesnya menggunakan IMAP (*Internet Message Access Protokol*) atau POP3 (*Post Office protokol*).

2.6.1.3. Member Server

Apabila Windows 2000 Server digunakan sebagai *member server* maka hanya dapat bertindak sebagai klien dalam jaringan dan tidak dapat menjalankan fungsi server untuk mengatur jaringan. Ketika Windows 2000 Server diinstal pertama kali, maka secara otomatis akan berfungsi sebagai *member server*. Untuk mengubahnya sebagai domain controller digunakan perintah *dcpromo* dari *commandprompt*.

2.6.1.4. Domain Controller

Domain Controller (DC) merupakan server yang berfungsi sebagai pengatur jaringan. Manajemen sumber daya dan obyek jaringan dilakukan dari DC, karena akses secara penuh terhadap *Active Directory* hanya dapat dilakukan dengan melakukan login ke DC. Di dalam jaringan berbasis Windows NT akan ditemui istilah *Primary Domain Controller (PDC)* dan *Backup Domain Controller (BDC)* namun dalam sistem jaringan Windows 2000 dua istilah tersebut sudah tidak dikenal lagi. Setiap DC dalam jaringan

adalah *peer* (setara) yang masing-masing dapat dikonfigurasi untuk melakukan replikasi objek *Active Directory*, sehingga apabila salah satu DC tidak berfungsi maka dapat segera digantikan oleh DC yang lain. Sangat disarankan dalam suatu organisasi untuk memiliki minimal 2 DC sehingga menjamin *fault tolerance*.

2.6.2. Fitur Baru Pada Windows 2000 Server

Untuk lebih memahami berbagai fasilitas dan kelebihan Windows 2000 Server dibandingkan sistem operasi terdahulu, berikut ini dipaparkan beberapa fitur baru yang penting pada Windows 2000 Server.

2.6.2.1. Active Directory Service

Directory Service dapat diumpamakan sebagai buku direktori telepon yang menyimpan berbagai informasi nama, alamat dan nomor telepon yang disusun berdasarkan abjad sehingga memudahkan proses pencarian. Peranan *Directory Service* dalam sebuah jaringan adalah sebagai *database* yang menyimpan berbagai informasi sumber daya dan objek jaringan secara terpadu sehingga dapat dikelola dan dikonfigurasi dengan mudah. Istilah *Active Directory Service* digunakan dalam lingkungan Windows 2000 untuk memberikan penekanan pada kemampuannya untuk melakukan berbagai fungsi manajemen secara dinamis dan otomatisasi dengan mudah dan cepat. Informasi yang disimpan dalam *Active Directory* antara lain meliputi *user* dan *group account*, printer, *file server*, serta berbagai *policy* menyangkut *user* dan *group*. *User* sebagai pengguna jaringan berkepentingan untuk dapat mengakses berbagai sumber daya dengan cepat dan mudah, sedangkan *administrator* berkepentingan untuk mengelola berbagai objek jaringan secara efisien. *Active Directory* memungkinkan pengelolaan jaringan menjadi lebih mudah karena berbagai sumber daya dan objek dapat disimpan secara terpusat untuk dikonfigurasi secara terpadu.

2.6.2.2. Group Policy

Group Policy merupakan media untuk mengatur profil *user* terutama yang berkaitan dengan *desktop setting*. Pengaturan yang dilakukan antara lain menentukan jenis aplikasi yang tersedia bagi *user*, konfigurasi *start menu*, serta akses terhadap berbagai *icon* seperti *Control Panel* dan *MyComputer*. Fasilitas ini sangat berguna untuk menyesuaikan lingkungan tampilan *desktop* dengan tingkat keahlian seorang *user*, serta memberikan tingkat keamanan sistem sehingga berbagai konfigurasi sensitif

tidak akan dapat diubah *user*. *Group Policy* dapat dikonfigurasi secara terpusat dengan menggunakan fasilitas *Active Directory*.

2.6.2.3. *Distributed File Sistem*

Ketika jaringan semakin besar dan jumlah user bertambah maka sering terjadi penyimpanan *file* menjadi tidak rapi lagi. *File-file* kerja dapat tersimpan di server maupun lokal di komputer masing-masing dengan memberikan hak *sharing* bagi pemakai lain. Proses pencarian *file* sering menjadi pekerjaan yang membingungkan karena peletakan *file* oleh *user* dilakukan dengan tidak konsisten. *Distributed File Sistem* (DFS) merupakan solusi masalah penyimpanan *file* dalam jaringan. *Administrator* menyediakan folder sesuai dengan kebutuhan, sedangkan folder pada DFS tersebut dihubungkan dengan letak *file* secara fisik. Dengan demikian seorang *user* dapat dengan mudah menyimpan dan mencari *file* pada folder yang telah disediakan tanpa perlu mengetahui di mana sebenarnya letak fisik suatu *file*. *File* pada DFS juga dapat disimpan secara *offline* di komputer lokal dan dilakukan proses sinkronisasi berkala dengan *file* di jaringan.

2.6.2.4. *Terminal Services*

Terminal Services merupakan fasilitas yang dapat digunakan untuk memanfaatkan komputer dengan *hardware* lama untuk dapat menjalankan berbagai aplikasi terbaru. *Terminal Services Server* diinstal pada komputer server dengan spesifikasi hardware yang mampu menjalankan *Windows 2000 Server*. Sedangkan *Terminal Services Client* diinstal pada komputer lama misalkan sekelas 486 atau *Pentium* klasik. Komputer klien mengakses berbagai aplikasi di server dengan menggunakan processing power komputer server. Fasilitas ini sangat berguna untuk memudahkan administrasi dan *maintenance* berbagai aplikasi secara terpusat karena instalasi aplikasi hanya dilakukan di server. Namun demikian berbagai aplikasi berat seperti *AutoCad* dan *Corel Draw* tidak akan berjalan maksimal dengan tools ini. Aplikasi yang cocok digunakan antara lain berbagai suite aplikasi office seperti *MS Office* dan *internet sharing*. *Terminal Services* juga dapat digunakan untuk melakukan remote administration terhadap suatu server.

27. *SQL Server*

Microsoft SQL Server adalah suatu *database relasional* yang dapat digunakan untuk melakukan analisa sistem *e-commerce*, jalur bisnis dan solusi data warehouse.

SQL Server 2000, pada versi terbarunya, memberikan dukungan terhadap XML (*Extensible Markup Language*) dan HTTP (*Hypertext Transfer Protocol*), ketersediaan dan unjuk kerja untuk **membagi beban** dan **menjamin uptime**, dan sebuah **fungsi** untuk manajemen dan *tuning* untuk melakukan otomatisasi **pekerjaan** yang **dilakukan secara rutin**.

Microsoft SQL Server 2000 merupakan sekumpulan komponen yang bekerja bersama untuk **menangani** penyimpanan data dan analisa dari sistem pemrosesan data skala *enterprise*. Berbagai **macam fitur** dimiliki oleh SQL Server, **berikut ini adalah** fitur-fitur utama yang dimiliki oleh SQL Server.

2.7.1. **Integrasi** dengan Internet

SQL Server 2000 telah **memiliki** dukungan terhadap XML pada komponen utama *database engine*. Skalabilitas, **ketersediaan** dan fitur **keamanan** juga merupakan fitur yang **tersedia** untuk dapat **beroperasi** sebagai **komponen penyimpanan data** pada *web site* berskala besar. SQL Server 2000 dapat bekerja dengan **fasilitas enkripsi** dari Windows 2000 Server untuk **menyimpan** data dengan **aman**. *Arsitektur security*-nya berdasarkan pembagian *user* dan *group* sebagaimana *security* pada Windows NT/2000, jadi keamanan di SQL Server bisa **terintegrasi** dengan keamanan Windows 2000/NT.

Model pemrograman pada SQL Server 2000 telah terintegrasi dengan *arsitektur* Windows DNA (*Windows Distributed InterNet Application Architecture*) untuk membangun **aplikasi berbasis web**, fitur seperti *English Query* dan *Microsoft Search Service* juga **didukung** untuk **memberikan** kemudahan dalam melakukan *query* dan kemampuan proses pencarian dalam **aplikasi berbasis web**.

2.7.2. Skalabilitas dan **Ketersediaan** (*Scalability and Availability*)

Microsoft SQL Server 2000 dapat **beroperasi** pada Microsoft windows 2000 Professional, Microsoft Windows 2000 Server, Microsoft Windows 2000 Advanced Server, Windows 98, Windows Millenium Edition dan Microsoft Windows XP. Selain itu juga dapat **beroperasi** pada semua edisi Microsoft Windows NT versi 4.0. Mesin *database* yang **digunakan** merupakan sebuah *server robust* yang dapat **menangani database** berukuran terabyte dan **diakses** oleh ribuan **pengguna**. Sebagai tambahan, apabila dijalankan dengan konfigurasi *setting default*, SQL Sewer 2000 memiliki fitur seperti *self-tuning* yang dapat berjalan efektif pada laptop dan desktop **tanpa** membutuhkan administrasi yang **rumit** dari **pengguna**. SQL Sewer 2000 Windows CE

Edition bahkan telah mengembangkan **kemampuan** model **pemrograman** pada SQL Server 2000 ke **piranti-piranti** mobile yang menggunakan Windows CE sehingga memberikan **kemudahan** untuk melakukan **integrasi** ke lingkungan SQL Server 2000.

SQL Server 2000 **dapat bekerja** dengan sistem **clustering failover** pada Windows NT dan Windows 2000 dengan demikian SQL Server 2000 **bisa dijalankan** pada **banyak** komputer. Apabila salah **satu** komputer **mengalami kerusakan** maka otomatis akan **ditangani** oleh sistem **komputer** lain yang masih aktif. Hal ini **digunakan** untuk menangani recovery **data** apabila **terjadi kerusakan**. Selain itu SQL Server 2000 juga memperkenalkan **fitur** log shipping yang **memungkinkan** untuk **mengelola** warm server standby pada lingkungan yang tidak **membutuhkan** ketersediaan tinggi.

Log shipping **merupakan** suatu **fasilitas** untuk melakukan backup data terhadap **semua** log transaksi yang **terjadi pada server** database utama ke **server database** sekunder. **Proses** ini dilakukan secara **periodik**. **Warm server** adalah sebuah komputer server yang **akan hidup** secara **periodik dalam waktu** yang **ditentukan** untuk menerima kiriman data log transaksi dari server utama.

2.7.2.1. Kemampuan Database Skala Besar

SQL Server 2000 memiliki **optimasi** berkecepatan tinggi yang mendukung lingkungan **database** skala sangat besar. SQL Server versi 6.5 dan sebelumnya dapat mendukung database dari ukuran 200 GB sampai 300 GB. SQL Server 2000 dan SQL Server versi 7.0 dapat **secara** efektif mendukung **database** sampai ukuran terabyte.

Database pada SQL Server 2000 pada **dasarnya** adalah **file sistem** biasa pada Windows, sehingga **mempermudah** pembuatan dan **administrasi** database. Ukuran tiap **halaman** dari database adalah 8KB dan setiap penambahan data akan **dilakukan** dalam langkah extents, tiap **extent** mempunyai ukuran **8 pages** dan tiap pages **memuat 8 page**. Apabila ada penambahan data maka akan dilakukan penambahan pages atau disebut **extent** sebesar 64KB yang berakibat pada peningkatan penggunaan I/O (*Input/Output*). Seperti **diperlihatkan** dalam **Gambar 2.7**,

(The content of Gambar 2.7 is not visible in the provided image.)

(The content of Gambar 2.7 is not visible in the provided image.)

(The content of Gambar 2.7 is not visible in the provided image.)

SQL Server

Data File		Data File	
File Group			
Heap/Index		Heap/Index	
Extent	Extent	Extent	Extent
Pages	Pages	Pages	Pages

Gambar 2.7. Ukuran halaman database pada SQL Server [ANO-05]

2.7.2.2. Query Optimizer

Query optimizer pada SQL Server 2000 memiliki metode-metode akses baru untuk meningkatkan kecepatan pemrosesan query. Metode-metode baru ini seringkali menggunakan peningkatan dan penyederhanaan pada struktur data di database:

- Query optimizer menggunakan metode akses serial, read-ahead I/O pada saat melakukan scanning table dan indeks untuk meningkatkan performa. Optimizer juga menggunakan algoritma hash dan merge untuk melakukan join. Hash dan merge merupakan algoritma yang bisa dilakukan untuk melakukan join, masing-masing memiliki kelebihan yang berbeda tergantung kondisinya. Apabila kedua tabel sumber yang akan di-join memiliki ukuran yang besar dan telah disortir pada kolom yang akan digabung, maka merge join akan digunakan karena operasi dapat dilakukan, dengan cepat. Akan tetapi apabila kedua tabel sumber memiliki perbedaan ukuran yang besar, maka hash join akan digunakan, karena operasinya akan jauh lebih cepat dibandingkan merge join.
- Query optimizer mendukung model prepare/execute dalam melakukan eksekusi pernyataan SQL. Pada saat sebuah aplikasi mengeksekusi pernyataan SQL, optimizer menerapkan sebuah algoritma efisien untuk menentukan apakah pernyataan yang sama telah dilakukan sebelumnya. Jika optimizer menemukan pernyataan yang sama maka execution plan untuk pernyataan itu akan digunakan sehingga akan menghemat waktu pemrosesan. Pada suatu sistem dimana banyak pengguna yang menggunakan aplikasi yang sama hal ini dapat mengurangi

penggunaan *resource* yang diperlukan untuk melakukan kompilasi terhadap pernyataan SQL untuk menghasilkan *execution plan*.

2.7.2.3. Dukungan Memori Berukuran Besar

SQL Server 2000 Enterprise Edition menggunakan Microsoft Windows 2000 Address Windowing Extensions API (Application Programming Interface) untuk mendukung penggunaan memori sampai berukuran 64 GB RAM. Hal ini memungkinkan SQL Server 2000 Enterprise Edition untuk melakukan proses *caching* terhadap sejumlah besar baris dalam memori, yang akan mengurangi *overhead* dan meningkatkan kemampuannya untuk memproses *query*.

2.7.3. Fasilitas Database Berskala Enterprise

Mesin *database* pada SQL Server 2000 mendukung fitur-fitur yang dibutuhkan pada lingkungan dimana kemampuan pemrosesan data berskala besar diperlukan. Mesin *database* akan melindungi integritas data dan secara bersamaan akan meminimalisir *overhead* yang dibutuhkan untuk menangani ribuan pengguna yang dalam waktu bersamaan melakukan modifikasi data pada *database*. SQL Server 2000 memiliki kemampuan *distributed query* yang memungkinkan untuk mengakses data dari berbagai macam sumber dengan kemudahan layaknya bagian dari database SQL Server 2000, sementara pada saat yang bersamaan dukungan terhadap transaksi terdistribusi akan melindungi integritas data dari setiap proses *update* yang terjadi pada data yang terdistribusi.

Kemampuan replikasi memberikan kemudahan untuk mengelola banyak salinan data, dimana masing-masing salinan data akan mengalami sinkronisasi. Satu set data dapat direplikasi ke banyak salinan data, *mobile*, pengguna yang tidak senantiasa terhubung dalam jaringan untuk kemudian dilakukan proses penggabungan data ke sumber.

2.7.4. Kemudahan Instalasi dan Penggunaan

SQL Server 2000 memiliki sejumlah *tool* administratif dan *development* yang akan meningkatkan kinerja saat melakukan instalasi, implementasi, pengelolaan dan penggunaan SQL Server pada berbagai macam tempat. SQL Server 2000 juga mendukung model pemrograman standar yang terintegrasi dengan Windows DNA, yang menyebabkan SQL Server *database* dan *data warehouse* sebagai bagian yang tak terlihat dari suatu sistem yang berskala besar. Fitur-fitur tersebut memungkinkan untuk

menghasilkan aplikasi SQL Server dengan produktif dan cepat, serta pengguna dapat melakukan implementasi dengan minimum instalasi dan proses administrasi lainnya.

Banyak database yang memiliki kemampuan berskala *enterprise* merupakan suatu sistem yang kompleks dan susah untuk dikonfigurasi. Microsoft SQL Server 2000 memiliki berbagai macam fitur dan *tool* yang akan menyederhanakan keseluruhan proses, SQL Server 2000 juga dapat dioperasikan pada sistem kecil sederhana yang tidak memiliki banyak pengguna, secara efisien dengan proses administrasi yang minimal.

2.7.5. Data Warehouse

Data warehouse adalah suatu sistem pusat pengolahan untuk keseluruhan data atau data-data yang vital yang dimiliki oleh suatu perusahaan. Data dari berbagai macam sumber diekstrak dan diorganisir secara selektif untuk dipergunakan lagi oleh aplikasi lain. Bagian dari *data warehouse* adalah *data mart*, dimana sekumpulan dari *data mart* akan membentuk suatu *data warehouse*. Masing-masing *data mart* menyimpan informasi yang khusus, spesifik menyangkut suatu departemen tertentu.

SQL Server 2000 memiliki berbagai macam *tool* untuk melakukan ekstraksi data dan melakukan analisa terhadap rangkuman data untuk digunakan dalam proses analisa online. SQL Server juga memiliki *tool* untuk melakukan desain database secara visual dan melakukan analisa data menggunakan pertanyaan-pertanyaan *English-based*.

2.7.5.1. Data Warehousing Framework

Data Warehousing Framework merupakan sekumpulan komponen dan API yang melakukan implementasi dari fitur-fitur data *warehouse* di SQL Server 2000. *Framework* ini menyediakan *interface* umum yang bisa digunakan oleh berbagai macam komponen lain untuk membangun sebuah *data warehouse* atau *data mart*.

2.7.5.2. Data Transformation Services

Data Transformation Services (DTS) menyediakan seperangkat layanan yang dapat digunakan untuk membangun sebuah *data warehouse* atau *data mart*. *DTS* memberikan dukungan untuk melakukan ekstraksi data dari sumber data heterogen dan melakukan rangkuman data untuk membentuk sebuah *data warehouse*.

2.8. TCP/IP (*Transmission Control Protocol/Internet Protocol*)

Dalam dunia komunikasi data komputer, protokol mengatur bagaimana sebuah komputer berkomunikasi dengan komputer lain. Dalam jaringan komputer, dapat digunakan banyak macam protokol tetapi agar dua buah komputer dapat berkomunikasi, keduanya perlu menggunakan protokol yang sama. Protokol berfungsi mirip dengan bahasa. Agar dapat berkomunikasi, orang-orang perlu berbicara dan mengerti bahasa yang sama.

TCP/IP (*Transmission Control Protocol/Internet Protocol*) adalah sekelompok protokol yang mengatur komunikasi data komputer di Internet [PUR-01]. Komputer-komputer yang terhubung ke Internet berkomunikasi dengan protokol ini. Karena menggunakan bahasa yang sama, yaitu protokol TCP/IP, perbedaan jenis komputer PC, dengan sistem operasi Windows dapat berkomunikasi dengan komputer Macintosh atau dengan Sun SPARC yang menjalankan Solaris. Jadi, jika sebuah komputer menggunakan protokol TCP/IP dan terhubung langsung ke Internet, maka komputer tersebut dapat berhubungan dengan komputer di belahan dunia manapun yang juga terhubung ke Internet.

Perkembangan TCP/IP yang diterima luas dan praktis menjadi standar *de-facto* jaringan komputer berkaitan dengan ciri-ciri yang terdapat pada protokol itu sendiri:

- Protokol TCP/IP dikembangkan menggunakan standar protokol yang terbuka sehingga siapapun bisa ikut mengembangkan standar ini.
- Standar protokol TCP/IP dalam bentuk *Request For Comment (RFC)* dapat diambil oleh siapapun tanpa biaya.
- TCP/IP dikembangkan dengan tidak tergantung pada sistem operasi atau perangkat keras tertentu.
- Pengembangan TCP/IP dilakukan dengan konsensus dan tidak tergantung pada vendor tertentu.

TCP/IP independen terhadap perangkat keras jaringan dan dapat dijalankan pada jaringan Ethernet, Token Ring, jalur telepon dial-up, jaringan X.25, dan praktis jenis media transmisi apapun.

Pengalaman TCP/IP bersifat unik dalam skala global. Dengan cara ini, komputer dapat saling terhubung walaupun jaringannya seluas Internet sekarang ini.

- TCP/IP memiliki fasilitas *routing* yang memungkinkan sehingga dapat diterapkan pada *inter-network*.

TCP/IP memiliki banyak jenis layanan...

2.8.1. Dasar Arsitektur TCP/IP

Pada dasarnya, komunikasi data merupakan proses mengirimkan data dari satu komputer ke komputer yang lain. Untuk dapat mengirimkan data, pada komputer harus ditambahkan alat khusus, yang dikenal sebagai *network interface* (jaringan antarmuka). Jenis antarmuka jaringan ini bermacam-macam, bergantung pada media fisik yang digunakan untuk mentransfer data tersebut. Dalam proses pengiriman data ini terdapat beberapa masalah yang harus dipecahkan.

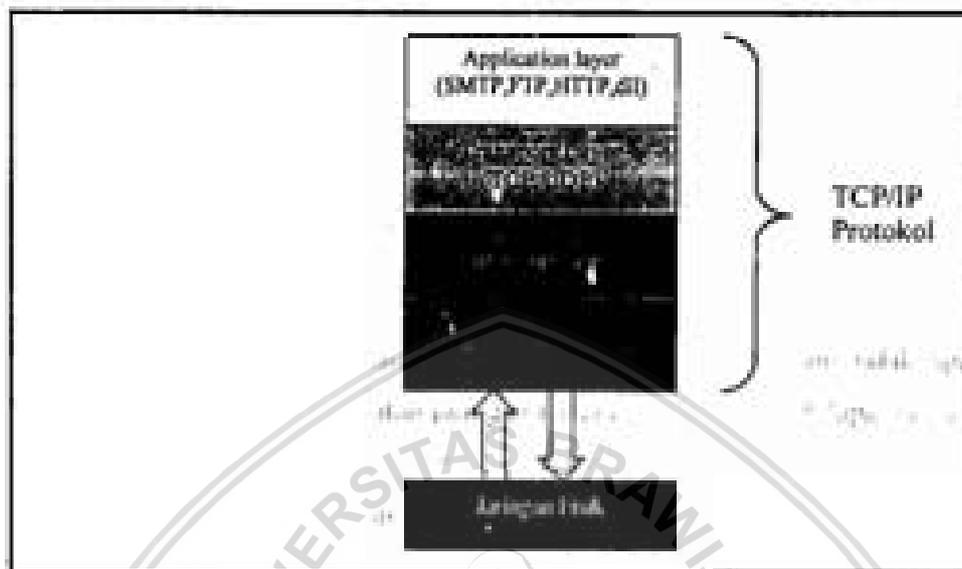
1. Data harus dapat dikirimkan ke komputer yang tepat sesuai tujuannya. Hal ini akan menjadi rumit jika komputer tujuan transfer data ini tidak berada pada jaringan lokal melainkan di tempat yang jauh. Jika lokasi komputer yang saling berkomunikasi jauh (secara jaringan) maka terdapat kemungkinan data rusak atau hilang. Oleh sebab itu, perlu mekanisme yang mencegah rusak atau hilangnya data ini.
2. Pada komputer tujuan transfer data mungkin terdapat lebih dari satu aplikasi yang menunggu datangnya data. Data yang dikirim harus sampai pada aplikasi yang tepat, pada komputer yang tepat.

Cara alamiah untuk menghadapi setiap masalah ialah memecahkan masalah tersebut menjadi bagian yang lebih kecil. Dalam memecahkan masalah transfer data di atas, para ahli jaringan komputer melakukan hal yang sama. Untuk setiap problem komunikasi data, diciptakan solusi khusus berupa aturan-aturan untuk menangani problem tersebut. Untuk menangani komunikasi data, keseluruhan aturan ini harus bekerja sama satu dengan lainnya. Sekumpulan aturan untuk mengatur pengiriman data ini disebut sebagai protokol komunikasi data. Protokol ini diimplementasikan dalam bentuk program komputer (*software*) yang terdapat dalam komputer dan peralatan komunikasi data lainnya.

TCP/IP adalah sekumpulan protokol yang didesain untuk melakukan fungsi-fungsi komunikasi data pada *Wide Area Network* (WAN). TCP/IP terdiri atas sekumpulan *Layer* yang masing-masing bertanggung jawab atas bagian-bagian tertentu dari komunikasi data. Setiap *layer* mempunyai tugas sendiri-sendiri sehingga setiap tugas masing-masing *layer* menjadi jelas dan sederhana. *Layer* yang satu tidak perlu mengetahui cara kerja *Layer* yang lain, sepanjang ia masih bisa mengirim dan menerima data.

Berkat penggunaan prinsip ini, TCP/IP menjadi protokol komunikasi data yang fleksibel. Protokol TCP/IP dapat diterapkan dengan mudah di setiap jenis komputer dan

interface jaringan, karena sebagian besar isi kumpulan protokol ini tidak spesifik terhadap satu komputer atau peralatan jaringan tertentu. Agar TCP/IP dapat berjalan di atas *interface* jaringan tertentu, hanya perlu dilakukan perubahan pada protokol yang berhubungan dengan *interface* jaringan saja.



Gambar 2.8. Layer TCP/IP [PUR-01:23]

Sekumpulan protokol TCP/IP ini dimodelkan dengan empat layer TCP/IP, sebagaimana terlihat dalam Gambar 2.8. TCP/IP terdiri atas empat lapis kumpulan protokol yang bertingkat. Keempat lapis/layer tersebut adalah:

- Network *Interface Layer*
- *Internet Layer*
- Transport Layer
- Application Layer

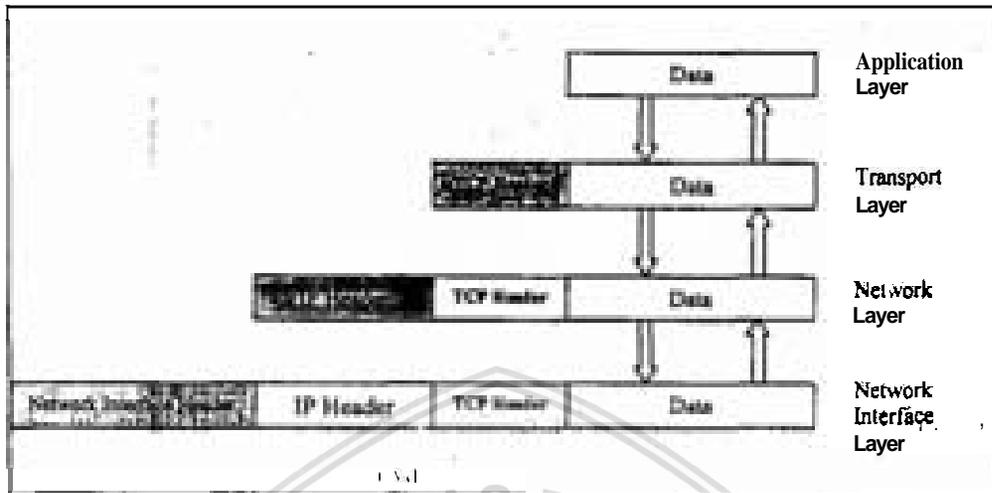
Dalam TCP/IP, terjadi penyampaian data dari protokol yang berada di satu layer ke protokol yang berada di layer yang lain seperti terlihat dalam Gambar 2.9. Setiap protokol memperlakukan semua informasi yang diterimanya dari protokol lain sebagai data.

Jika suatu protokol menerima data dari protokol lain di layer atasnya, ia akan menambahkan informasi tambahan miliknya ke data tersebut. Informasi ini memiliki fungsi yang sesuai dengan fungsi protokol tersebut. Setelah itu, data ini diteruskan lagi ke protokol pada layer di bawahnya.

Hal yang sebaliknya terjadi jika suatu protokol menerima data dari protokol lain yang berada pada layer di bawahnya. Jika data ini dianggap valid, protokol akan



melepas informasi **tambahan** tersebut, untuk kemudian **meneruskan data** itu ke **protokol** lain yang berada pada **layer** di atasnya.



Gambar 2.9. Pergerakan data dalam *layer* TCP/IP [STA-01:56]

Lapisan/*Layer* terbawah, yaitu Network *Interface* Layer, bertanggung jawab mengirim dan menerima data ke dan dari media fisik. Media fisiknya dapat berupa kabel, serat optik, atau gelombang radio. Karena tugasnya ini, protokol pada layer ini harus mampu menerjemahkan sinyal listrik menjadi data digital yang dimengerti komputer, yang berasal dari peralatan lain yang sejenis.

Lapisan/*Layer* protokol berikutnya ialah *Internet Layer*, protokol yang berada pada layer ini bertanggung jawab dalam proses pengiriman paket ke alamat yang tepat. Pada layer ini terdapat tiga macam protokol, yaitu IP, ARP, dan ICMP.

IP (*Internet Protocol*) berfungsi untuk menyampaikan paket data ke alamat yang tepat. ARP (*Address Resolution Protocol*) ialah protokol yang digunakan untuk menemukan alamat hardware dari host/komputer yang terletak pada network yang sama. Sedangkan ICMP (*Internet Control Message Protocol*) ialah protokol yang digunakan untuk mengirimkan pesan dan melaporkan kegagalan pengiriman data.

Layer berikutnya, yaitu *Transport Layer*, berisi protokol yang bertanggung jawab untuk mengadakan komunikasi antara dua host/komputer. Kedua protokol tersebut ialah TCP (*Transmission Control Protocol*) dan UDP (*User Datagram Protocol*).

Layer teratas, ialah *Application Layer*. Pada *layer* inilah terletak semua aplikasi yang menggunakan protokol TCP/IP ini.

2.8.2. SLIP (*Serial Line Interface Protocol*) dan PPP (*Point to Point Protocol*)

Layer terbawah dari TCP/IP adalah *Network Interface layer*. *Layer* ini bertanggung jawab mengirim data dan menerima data dari media fisik. Beberapa contohnya adalah SLIP (*Serial Line Interface Protocol*) dan PPP (*Point to Point Protocol*). *Interface* jaringan yang sangat banyak dipakai adalah modem telepon, yang dihubungkan ke komputer melalui *serial port*. Protokol yang banyak dipakai untuk menangani jalur *serial* ini adalah SLIP dan PPP.

2.8.2.1. SLIP (*Serial Line Interface Protocol*)

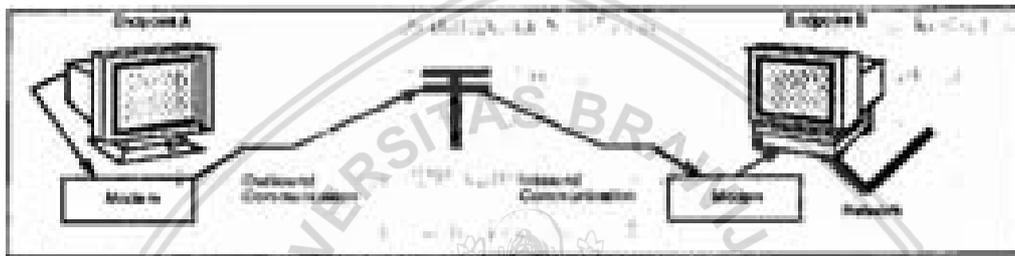
SLIP merupakan suatu protokol untuk melakukan transmisi datagram IP menggunakan jalur serial. Biasanya digunakan untuk komunikasi menggunakan *serial port* maupun modem. Dengan menggunakan SLIP maka berbagai macam *host* (komputer) dan *router* dalam suatu jaringan dapat saling berkomunikasi (*host-host, host-router, router-router*). SLIP melakukan modifikasi terhadap standar Internet datagram dengan cara menambahkan karakter khusus "SLIP END" pada awal dan akhir *frame* yang menyebabkan sekumpulan datagram dapat dipisahkan dan dibedakan. Penggunaan SLIP sendiri telah banyak digantikan oleh protokol PPP, yang menawarkan lebih banyak kemampuan.

2.8.2.2 Dial-Up Networking dan PPP (*Point to Point Protocol*)

Koneksi, *dial-up* adalah *point-to-point*. *Dial-up* menghubungkan sebuah IP tunggal pada salah satu ujung dengan IP tunggal lainnya pada ujung yang berbeda. *Ethernet* adalah media *broadcast* dan dapat memiliki banyak computer yang terhubung pada segmen tunggal. Pada hubungan, *point-to-point*, anda tidak dapat melakukan broadcast sehingga sembarang fungsi yang bergantung pada *broadcast* (ARP, query nama NetBIOS Windows, DHCP, dan sebagainya) tidak bekerja. PPP (*Point-to-Point Protocol*) adalah yang selalu digunakan untuk dial-up. Protokol ini serupa dengan *Ethernet* pada layer link pada stack protokol. Anda dapat menjalankan banyak protokol, tidak hanya IP pada PPP. Dalam koneksi dial-up ada 2 jenis koneksi server yaitu server dial-up privat dan penggunaan ISP (*Internet Service Provider*). NAN-021.

PPP lebih **banyak digunakan** daripada **SLIP** karena dapat menangani komunikasi sinkron maupun **asinkron**. PPP dapat melakukan pembagian **penggunaan satu jalur dengan banyak** pengguna dan memiliki **kemampuan** mendeteksi **kesalahan** yang tidak dimiliki oleh SLIP. Penggunaan PPP yang umum adalah untuk **menghubungkan** dua buah komputer dari satu titik **ke** titik lain. Jalur komunikasi **yang** dihasilkan akan bersifat **full duplex** dimana kedua belah pihak dapat **berkomunikasi dua arah** secara bersamaan dan **simultan**.

Contoh **konfigurasi** yang umum dilakukan terdiri **atas** dua **endpoint** yang terhubung **menggunakan** jalur komunikasi. **Endpoint** dapat berupa komputer yang memiliki lokasi **terpisah** atau terhubung secara fisik ke sebuah **jaringan**. Jalur **Point to Point** tersebut dapat digambarkan **seperti** dalam **Gambar 2.10**.



Gambar 2.10. Point to Point Protocol (PPP)

PPP terdiri **atas** beberapa protokol mini. **Protokol tersebut adalah sebagai berikut:**

- **LCP (Link Control Protocol)**. LCP ini berfungsi membentuk dan memelihara link.
- **Authentication Protocol**. Protokol ini digunakan untuk memeriksa boleh tidaknya **user** menggunakan **link** ini. Ada dua jenis autentikasi yang umum digunakan, yaitu **Password Authentication Protocol (PAP)** dan **Challenge Handshake Authentication Protocol (CHAP)**.
- **Network Control Protocol (NCP)**. NCP berfungsi mengkoordinasi operasi bermacam-macam protokol jaringan yang melalui **link** PPP ini. Beberapa hal yang dilakukan oleh protokol ini ialah menegosiasikan jenis protokol kompresi yang akan dipakai serta menanyakan **IP address** mitranya.

2.9 VPN (*Virtual Private Network*)

VPN merupakan pengembangan dan jaringan privat yang dapat user gunakan dengan memanfaatkan fasilitas jaringan yang sudah ada misalnya melalui jaringan Internet publik. Dengan menggunakan jaringan VPN, seolah-olah mempunyai jaringan pribadi yang dapat dibawa kemana-mana tanpa harus pergi ke tempat jaringan tersebut berada. Apabila menggunakan VPN maka user dapat mengirimkan data-data yang telah dienkripsi antara dua bunh komputer yang letaknya berjauhan.

Untuk melakukan pengamanan data maka digunakan suatu proses pengkapsulan, hal ini dikarenakan data melewati jaringan publik yang memungkinkan adanya pencurian data. Data yang telah dienkripsi tersebut tidak akan dapat dibaca oleh penerima jika mereka tidak mempunyai suatu kunci untuk membaca enkripsi.

Penelitian yang saya kerjakan ini merupakan kelanjutan dari penelitian yang dilakukan oleh Husnul Khotimah Mahasiswa Teknik Elektro Universitas Brawijaya Malang (0110630071) tentang "Implementasi Replikasi Basis Data Melalui Jaringan Telepon Pada PDAM Kabupaten Malang" dan oleh Fika Hastarita Rachman Mahasiswa Teknik Elektro, Universitas Brawijaya Malang (0110630071) tentang "Pengelolaan Gudang Data untuk Sistem Replikasi melalui Jaringan Telepon pada PDAM Kabupaten Malang".

Virtual Private Network mempunyai dua komponen penting yaitu VPN Server dan VPN Client. Untuk membangun jaringan VPN maka kedua komponen ini harus ada dan tidak boleh ditinggalkan. Untuk mengetahui prinsip kerja dari jaringan VPN terlebih dahulu harus memahami fungsi dari masing-masing VPN Server dan VPN Client. VPN juga menggunakan protocol *Point-to-Point Tunneling Protocol (PPTP)*, *Layer Two Tunneling Protocol (L2TP)* dan Protokol *Secure Socket Layer (SSL)* yang digunakan untuk mengamankan data.

VPN Server mempunyai prinsip yang hampir sama dengan sebuah gateway yang digunakan untuk jaringan client. VPN Server ini dapat dikonfigurasi untuk melaksanakan routing serta layanan *remote acces*. Sedangkan VPN Client melakukan hubungan ke suatu jaringan public seperti Internet. Hubungan antara client dan server VPN mirip dengan jaringan Point, to Point, Link, dalam sebuah jaringan pribadi. Sebelum VPN Client melakukan hubungan dengan VPN Server seperti juga dalam jaringan biasa harus melewati proses autentifikasi dan otorisasi.

2.9.1. Komponen jaringan W N

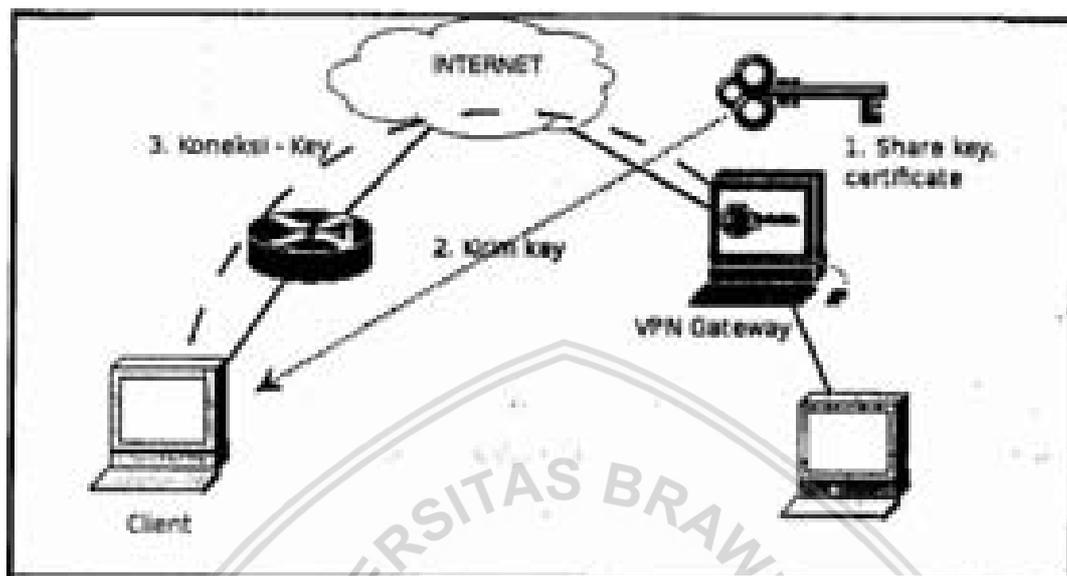
Untuk **membangun** sebuah jaringan VPN user membutuhkan beberapa komponen yang berkaitan dengan jaringan tersebut. Komponen yang dibutuhkan dalam membangun jaringan VPN antara lain :

- **VPN Server**
VPN **Server** merupakan komponen utama yang harus **ada** dalam jaringan VPN. VPN Server **ini** merupakan sebuah computer yang akan menerima koneksi dari VPN Client.
- **VPN Client**
VPN Client merupakan suatu device (komputer) yang akan melakukan koneksi dengan VPN Server. **Selain** dapat bempa computer VPN Client **juga** dapat bempa sebuah:outer.
- **Tunnel** ;
Tunnel merupakan **suatu cara di mana data** yang di transferkan (**dari VPN Server** ke VPN client **atau** sebaliknya) dibungkus dalam sebuah kapsul yang diamankan dengan pepgkapsulan, dan **kemudian** dilakukan **enkripsi**.
- **VPN Connection**
Merupakan sebuah koneksi **antara VPN Client** dan VPN Server **di mana data** yang **digunakan harus** dibungkus dan **dienkripsi** terlebih dahulu.

Seoara **umum** proses terjadinya **hubungan dalam VPN** dapat **dijelaskan** pada **gambar 2.11** dan **keterangannya berikut ini**.

1. VPN Client akan membuat **hubungan VPN** kepada remote acces **atau** VPN Sewer yang telah terhubung ke Internet. **Dalam hal ini** VPN Server juga **berfungsi mirip dengan** gateway.
2. **Setelah** VPN Server menerima **panggilan** dari VPN Client untuk melakukan suatu koneksi, **VPN Server** akan **menjawab panggilan tersebut melalui virtual call**.
3. Sebelum melakukan koneksi dengan VPN Client, VPN Sewer terlebih dahulu hams **melakukan autentifikasi dan otorisasi kepada** VPN Client.
4. **Setelah semua** proses **otentifikasi dan otorisasi selesai** dan VPN Client dianggap sah **sebagai** client-nya, maka antara VPN **Server** dan VPN Client **sudah** dapat melakukan koneksi yang biasanya digunakan untuk melakukan transfer data **antar VPN Client, dan** Server.

5. Apabila proses autentikasi dan otorisasi dianggap gagal oleh VPN Server maka VPN Client tidak diizinkan melakukan hubungan dengan VPN Server dan harus melakukan logon kembali seperti semula.



Gambar 2.11. Blok diagram VPN [ANO-07]

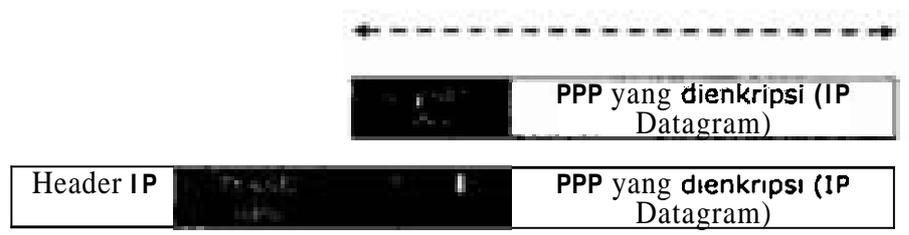
2.9.2. Jenis-jenis VPN

2.9.2.1. VPN dengan *Point-to-Point Tunneling Protocol* (PPTP)

Point-to-Point Tunneling Protocol (PPTP) merupakan salah satu protokol yang digunakan untuk enkripsi data. Protokol ini pertama kali didukung oleh sistem operasi Windows NT 4.0 dan Windows 98. PPTP merupakan protokol perluasan dari Point-to-Point Protocol (PPP) yang menyediakan fasilitas autentikasi, kompresi, dan juga tentunya enkripsi. Jika pada protokol PPP didukung oleh Windows NT 4.0 dan Windows 98 maka untuk PPTP ini didukung oleh Windows 2000 Server dan Windows Server 2003 untuk server dan Windows XP untuk sisi client.

Secara default protokol PPTP ini sudah terinstalasi di dalam protokol TCP/IP, dan protokol PPTP ini menggunakan port 128. Protokol PPTP dan MPPE menyediakan layanan pokok untuk proses penkapsulan dan enkripsi data pribadi. Proses penkapsulan merupakan salah satu cara yang digunakan untuk mengamankan data, prinsip dari proses penkapsulan adalah dengan membungkus sebuah frame PPP dengan header Generik Routing Encapsulation (GRE) dan header IP. Pada header IP akan tercantum sebuah alamat (IP Address) dari sumber dan tujuan yang merespon VPN Client dan VPN Server (Gambar 1).

Frame PPP



Gambar 2.12. paket data pada PPTP [HER-041]

Dalam jaringan VPN, frame PPP dienkripsi oleh MPPE dengan menggunakan key enkripsi yang dibangun MS-CHAP, MS-CHAP v2, atau EAP-TLS. Untuk dapat membaca data yang telah dienkripsi maka client dari VPN juga harus menggunakan MS-CHAP, MS-CHAP v2, atau EAP-TLS tersebut. Sebuah contoh sebuah VPN Server menggunakan enkripsi key jenis MS-CHAP maka untuk melakukan komunikasi dengan VPN Client, VPN Client juga harus menggunakan enkripsi key jenis MSCHAP yang sama pula

Protokol PPTP dan L2TP masing-masing menyediakan maksimal 1000 port yang bisa user gunakan. Dengan jumlah port 1000 ini maka user dapat membuat jaringan VPN dengan client sebanyak 1000 buah. Tetapi fasilitas ini hanya berlaku pada Windows Server 2003 versi Stuserit Edition saja, sedangkan untuk versi Web Edition walaupun mempunyai jumlah port yang sama tetapi hanya menyediakan untuk satu buah client VPN saja.

2.9.2.2. W N dengan Layer Two Tunneling Protocol (L2TP)

Protokol tunneling yang kedua dari jaringan VPN yang biasa digunakan adalah Layer Two Tunnelling Protocol. Protokol ini biasanya digunakan bersamaan dengan proses pengamanan dengan menggunakan IPSec. Protokol L2TP ini biasanya digunakan oleh sistem operasi Windows 2000 baik untuk server maupun untuk client dalam jaringan VPN. Dukungan dari protokol L2TP ini adalah IPSec dan untuk kombinasi keduanya biasa dinamakan dengan L2TP/IPSec. L2TP/IPSec sama, seperti halnya dengan protokol, PPTP juga menyediakan suatu fasilitas utama dan jaringan VPN yaitu penkapsulan data dan enkripsi data.

Jika user akan menggunakan protokol ini untuk membangun jaringan VPN maka ada beberapa hal yang perlu diperhatikan yaitu kedua pengguna jaringan VPN yaitu server dan client harus mendukung protokol ini. Untuk masalah sistem operasi protokol

L2TP ini juga mendukung client dari Windows XP dan server untuk Windows Server 2003 sama seperti pada PPTP. L2TP secara default telah terinstalasi di dalam protokol TCP/IP dan protokol ini menggunakan port 5 atau port 128.

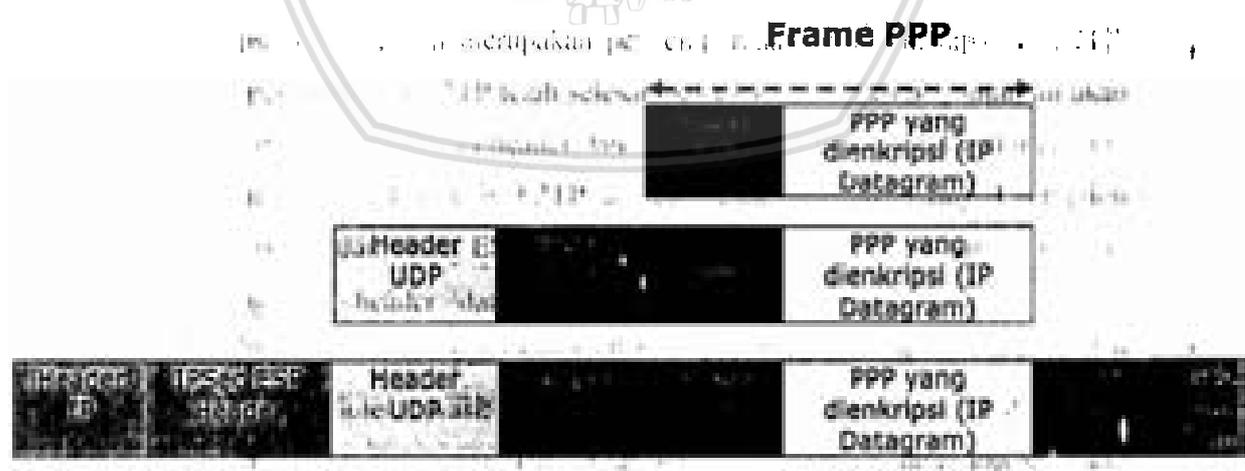
Sama seperti pada protokol PPTP pada protokol ini juga menyediakan fasilitas untuk enkripsi data dan pengkapsulan. Proses pengkapsulan terdiri atas dua lapisan, yang pertama adalah pengkapsulan L2TP dan yang kedua pengkapsulan IPsec.

□ **Penkapsulan L2TP**

Pada pengkapsulan L2TP, frame dari PPP atau IP datagram akan dibungkus oleh suatu header L2TP dan header UDP, setelah itu baru IP datagram akan dikirimkan.

□ **Pengkapsulan IPsec**

Pengkapsulan IPsec ini merupakan penyempurnaan dari pengkapsulan L2TP. Setelah pengkapsulan L2TP telah selesai maka hasil dari pengkapsulan ini akan kembali dibungkus dengan header dan trailer tertentu. Header dan trailer yang membungkus pengkapsulan L2TP adalah header Encapsulating Encryption Payload (ESP) dan trailer ESP. Selain header dan trailer ESP sebenarnya masih ada satu lagi header dan trailer yang digunakan pertama yaitu IPsec Authentication Trailer yang menyediakan sebuah pesan yang terintegrasi dan sebuah proses autentifikasi, dan yang kedua adalah header IP yang digunakan untuk menyimpan alamat (IP Address) dari sumber dan tujuan yang menggunakan jaringan VPN. Untuk lebih jelasnya lihat paket data pada protokol L2TP/IPsec pada gambar 2.



Gambar 2.13. paket data pada L2TP/IPSEC [HER-04]

repository.ub.ac.id

Sistem pengamanan yang kedua dari protokol ini adalah proses enkripsi. Pada L2TP proses enkripsi dilakukan oleh *Data Encryption Standard* (DES) atau *Triple DES* (3DES) dengan menggunakan sebuah *key* enkripsi yang dibangun dari *Internet Key Exchange* (IKE).

2.9.2.3. VPN dengan Secure Socket *Layer Protocol/Transport Layer Security Protocol (SSL/TLS)*

Sangatlah mungkin untuk melakukan tunnel VPN hanya pada lapisan aplikasi. Solusi dari Secure *Sockets Layer* (SSL) dan *Transport Layer Security* (TLS) yang dapat mengikuti pendekatan ini. Pemakai dapat mengakses jaringan VPN dari suatu perusahaan melalui suatu browser koneksi antar kliennya dan VPN server di perusahaan itu. Suatu koneksi sederhana dimulai dengan login ke dalam *HTTPS-SECURED website* dengan suatu browser. Sementara itu, ada beberapa peluang produk tersedia, seperti SSL-Explorer dari <http://3sp.com/showSslExplorer.do>, dan produk seperti penawaran ini fleksibilitas dikombinasikan dengan keamanan yang kuat dan mudah untuk digunakan. Menggunakan pengamanan koneksi yang ditawarkan oleh browser, para pemakai dapat menghubungkan jaringan dan mengakses jasa pada jaringan yang di-remote tersebut. Keamanan dapat dicapai dengan mengenkripsi lalu lintas menggunakan mekanisme SSL/TLS, yang sudah terbukti sangat dipercaya dan untuk selamanya ditingkatkan dan diuji.

2.9.3. OpenVPN

Pada saat ini, teknologi VPN selalu berhubungan erat dengan apa yang disebut dengan IPSEC yang merupakan kependekan dari *IP Security* dimana merupakan standar dalam membangun suatu komunikasi VPN yang terjadi pada Network Layer. IPSEC juga sering dipertimbangkan sebagai teknologi yang memiliki tingkat kesulitan cukup tinggi untuk dipahami bagi pemula dan mungkin juga tidaklah mudah untuk menerapkan dan mengelolanya dalam berbagai situasi seperti pada *filtered networks*, jaringan yang berhubungan dengan beberapa macam *Networks Address Translation* (NAT).

Karena hal tersebut, banyak vendor yang mulai mengimplementasikan sistem VPN yang berbasis SSL. OpenVPN adalah salah satu software opensource VPN yang berbasis SSL yang merupakan kependekan dari Secure Socket Layer yang memiliki pengertian yaitu sebuah protocol yang sering digunakan untuk mengamankan transaksi



dalam internet. Protokol ini sangatlah mudah sekali untuk dipahami dan dipelajari bagi pemula dan mudah untuk diimplementasikan dan diatur oleh administrator.

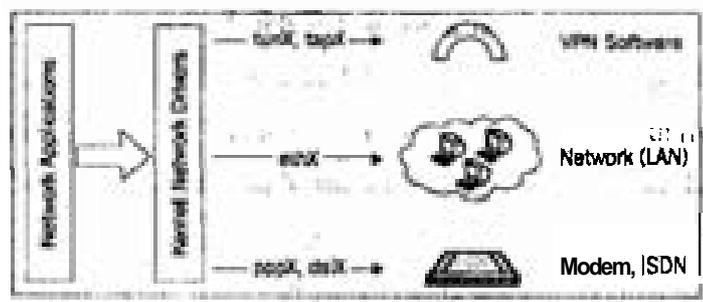
Struktur yang modular dari OpenVPN tidak hanya bisa ditemukan dalam model keamanannya, tetapi terdapat juga di dalam rencana jaringan. James Yonan (Pencipta OpenVPN) memilih driver Universal TUN/TAP untuk lapisan networking dari OpenVPN.

TUN/TAP driver adalah sebuah pmyek sumber terbuka yang tercakup di semua distribusi-distribusi Linux/UNIX yang modem seperti juga Windows dan Mac OSX. Seperti SSL/TLS yang digunakan di dalam banyak proyek, karena itu dapat memperbaiki dengan baik, dan fitur baru sedang ditambahkan. Dengan menggunakan alat-alat TUN/TAP, dapat mnyingkirkan banyak kompleksitas dari struktur OpenVPN. Svuktur sederhananya membawa keamanan yang lebih baik dibandingkan dengan solusi-solusi VPN yang lain. Kompleksitas adalah musuh utama dari keamanan. Sebagai contoh, IPSEC mempunyai suatu struktur kompleks dengan modifikasi-modifikasi yang kompleks di dalam kernel dan tumpukan protokol internet, dengan demikian mencipfakan banyak lubang kecil di dinding keamanan yang mungkin terjadi.

Universal Driver TUN/TAP dikembangkan untuk mendukung Linux kernel dalam membangun terowongan protokol inetemet lalu lintas. Hal ini merupakan suatu antar muka jaringan maya, yang kelihatan sebagai asli kepada semua aplikasi dan para pemakai, hanya nama tunX atau tapX mencirikan adanya alat-alat lainnya. Setiap aplikasi yang mampu menggunakan suatu antar muka jaringan dapat menggunakan antar muka terowongan. Setiap teknologi yang sedang anda jalankan di dalam jaringan, dapat berjalan di suatu TUN atau TAP.

Driver ini adalah salah satu faktor utama untuk memudahkan memahami pembuatan OpenVPN, mudah dalam pengaturan, dan keamanan pada waktu yang sama.

Gambar 2.14. berikut menjelaskan tentang OpenVPN menggunakan standar interface:



Gambar 2.14. Blok diagram VPN menggunakan standar interface [MAR-06]



Sebuah TUN device dapat digunakan sebagai suatu *Virtual Point-to-Point Interface*, seperti suatu modem atau DSL. TUN device disebut juga dengan routed mode, karena rute-rute disiapkan kepada mitra VPN.

Suatu TAP device, bagaimanapun, dapat digunakan seperti suatu adapter Ethernet yang maya. Hal ini memungkinkan mendengarkan daemon yang terhubung pada Frame ethernet, yang bukanlah mungkin dengan alat-alat TUN. Modus ini disebut modus penghubung karena jaringan itu dihubungkan seolah-olah di atas suatu jembatan perangkat keras.

Aplikasi-aplikasi dapat dibaca atau ditulis pada antar muka ini, perangkat lunak (tunnel driver) akan mengambil semua data dan menggunakan pustaka-pustaka yang cryptographic dari SSL/TLS ke encrypt mereka. Data itu dibungkus dan dikirim kepada yang lain yang merupakan akhir dari tunnel. Pengemasan ini dilakukan atas standardisasi UDP atau paket-paket TCP protokol kendali transmisi opsional. UDP merupakan pilihan pertama, tetapi TCP protokol kendali transmisi dapat sangat menolong dalam beberapa hal. Anda hampir dengan sepenuhnya bebas untuk memilih parameter-parameter konfigurasi, seperti angka-angka, protokol atau part, sepanjang kedua-duanya tujuan tunnel sepakat menggunakan gambar-gambar yang sama.

OpenVPN mendengarkan alat-alat TUN/TAP, mengambil traffic, mengenkripsinya, dan mengirimkan kepada mitra VPN yang lain, di mana proses OpenVPN yang lain menerima data, mengdeskripsikanya, dan menyampaikannya kepada *Virtual Network Device*, di mana aplikasi itu mungkin sedang menunggu data yang diproses.

2.9.4. Command dan Konfigurasi pada OpenVPN

Untuk mengaktifkan OpenVPN didahului dengan **command** `openvpn --config sample.ovpn`. Dimana konfigurasi OpenVPN yang berekstensi `*.ovpn` adalah file yang berisi konfigurasi yang akan digunakan pada OpenVPN dan berisi **command-command** OpenVPN yang sesuai dengan jenis koneksi VPN yang dibutuhkan. Berikut **command-command** pada OpenVPN :

Tabel 2.7: Parameter Command-command yang digunakan pada OpenVPN [MAR-06]

Parameter	Options	Function	Usage	Example
<code>--remote</code>	<code><hostname> <IP></code>	menunjuk ke ujung lain (remote) dari tunnel.	Command line and config file	<code>--remote vpn.dyndns.org</code>



dev	<device>	Perintah pemilihan device yg akan digunakan	Command line and config file	--dev tun --dev tap
ifconfig	For TUN device: <local IP> <remote IP> For TAP device: <local IP> <subnet mask>	Mengases IP virtual tunnel endpoint dan network pada tunnel	Command line and config file	--ifconfig 10.3.0.7 10.3.0.1 --ifconfig 10.3.0.2 10.3.33.255.0
secret	File containing the pre-shared key	Perintah pengalokasian dari pre-shared key	Command line and config file	--secret key.txt
comp-lzo	*yes* <no> *adaptive* (default)	openvpn menggunakan lzo library untuk memampatkan tunnel traffic	Command line and config file	--comp-lzo
port	*port number>	mempesifikasi port (baik lokal maupun remote) yang akan digunakan.	Command line and config file	--port 5001
proto	<udp> <tcp-client> <tcp-server>	Mengases protokol yang akan digunakan oleh OpenVPN. TCP client akan mencoba untuk memulai koneksi, sedangkan TCP server hanya menunggu client.	Command line and config file	--proto udp --proto tcp-client --proto tcp-server
tls-auth	<file path>	Mengases transmit unit secret maksimal	Command line and config file	--tls-auth 1200
	<interface name>	Mempesifikasi nama dari interface yang akan digunakan	Command line and config file	--dev-mode openvpn1
ping	<seconds>	Mengirimkan ping ke ujung tunnel partner yang lain melalui tunnel setelah beberapa detik tanpa traffic.	Command line and config file	--ping 10
ping-restart	<seconds>	Setelah beberapa detik tanpa menerima paket dari computer yang telah terkoneksi VPN, Tunnel akan melakukan restart.	Command line and config file	--ping-restart 60

	*	ping-restart berjalan hanya saat remote address diberikan	Command line and config file	--ping-alive-run
	*	Menjaga device tun/tap tetap up saat openvpn melakukan restart	Command line and config file	--persist-tun
	*	openvpn tidak akan membaca ulang keys pada saat restart	Command line and config file	--persist-key
	<seconds>	Mengontrol waktu dimana openvpn akan mencoba untuk memperbaiki hostname before sebelum menyuruh.	Command line and config file	--resolv-ecrty 6000
	<verbosity level>	mengontrol level dari verbosity, 0 adalah yang paling rendah, 11 adalah detail level maksimal.	Command line and config file	--verb 4
	<number of messages>	openvpn akan mencetak hanya 10 pesan consecutive dari kategori yang sama	Command line and config file	--mute 10
	<file>	Mendefinisikan file local machine's key.	Command line and config file	--key keys/VPN-Client.key
	*	Local machine berlagak seolah-olah sebagai TLS server	Command line and config file	--tlsv1-server
	*	Local machine berlagak seolah-olah sebagai TLS client	Command line and config file	--tlsv1-client

Perintah-perintah yang tercantum pada tabel diatas adalah beberapa perintah yang sering digunakan dalam OpenVPN.



BAB III METODOLOGI

Penyusunan penelitian ini didasarkan pada masalah yang bersifat **aplikatif**, yaitu perencanaan dan perealisasi sistem informasi agar dapat menampilkan unjuk kerja sesuai dengan yang direncanakan **dengan** mengacu pada **rumusan** masalah. Data dan spesifikasi bahan dan alat yang digunakan dalam perencanaan merupakan data sekunder yang diambil dari PDAM Kabupaten Malang. **Pemilihan** bahan dan alat berdasarkan perencanaan dan disesuaikan dengan **komponen** yang ada di pasaran.

Langkah-langkah yang **perlu dilakukan untuk** merealisasikan sistem informasi yang akan dibuat adalah sebagai berikut:

3.1. Studi Literatur

Studi **literatur** yang digunakan dalam perencanaan dan perealisasi sistem informasi ini mengumpulkan dan **mempelajari literatur** baik bersifat primer yaitu buku-buku yang berhubungan dengan **replikasi**, database, SQL Server 2000, Windows Server 2000, **OpenVPN** dan **TCP/IP** maupun yang bersifat sekunder yaitu data-data yang nantinya akan digunakan.

3.2. Penentuan **Spesifikasi** Bahan dan Alat

Sebelum melakukan **penelitian, maka ditentukan** spesifikasi bahan dan alat yang akan digunakan. Adapun **spesifikasi** bahan dan alat yang akan **direalisasikan** sebagai berikut:

Bahan :

- o Data **pelanggan** PDAM Kabupaten Malang.
- o Data transaksi **pembayaran rekening air selama 1 tahun.**

Alat dan *Software*:

- 3 unit PC dengan spesifikasi yang **cukup tinggi, karena** akan digunakan sebagai server basis data.
- o 2 unit modem 56 Kb untuk komunikasi **antar** komputer.
- **Sentral** telepon yang **dipakai** adalah Mini PABX (buatan **Cina**) sebagai **pengganti jaringan telepon TELKOM.**
- **Ethernet** Hub
- **Microsoft Access 2000.**

- Microsoft SQL Server 2000
- Windows Server 2000.
- OpenVPN 2.0.9

3.3. Perancangan Sistem

Setelah melakukan studi literatur dan menentukan spesifikasi alat, akan dilakukan perancangan dan perealisasi sistem. Hal yang harus dilakukan pertama kali adalah merencanakan blok diagram sistem secara keseluruhan kemudian menentukan dan menjelaskan fungsi dari masing-masing blok yang menyusun blok sistem keseluruhan. Berdasarkan hal tersebut, dilakukan dengan penentuan komponen-komponen pendukung yang diperlukan dalam perancangan.

Perangkat lunak atau *software* yang digunakan dalam perancangan dan perealisasi sistem ini menggunakan SQL Server 2000 dan OpenVPN 2.0.9.

3.4. Implementasi Sistem

Implementasi sistem merupakan langkah yang harus dilakukan setelah melakukan perancangan perangkat lunak yaitu dengan menentukan bagaimana sistem akan bekerja. Dengan menentukan hal tersebut, akan didapatkan suatu perangkat lunak atau *software* yang dapat mengerjakan fungsi-fungsi sistem yang diinginkan.

3.5. Metode Pengujian dan Analisis

Untuk mengetahui unjuk kerja sistem apakah sesuai dengan yang direncanakan maka dilakukan pengujian sistem. Pengujian dilakukan pada masing-masing blok dan secara keseluruhan.

3.5.1. Pengujian Masing-masing Blok

Pengujian masing-masing blok dilakukan meliputi pengujian koneksi menggunakan *Virtual Private Network (VPN)* dan pengujian proses replikasi. Hal ini bertujuan untuk mengetahui reliabilitas dan konsistensi koneksi yang dilakukan antara masing-masing komputer pada jaringan VPN dan juga untuk mengetahui integritas data di masing-masing komputer setelah proses replikasi.

3.5.2. Pengujian Sistem Keseluruhan

Pengujian ini merupakan tahap akhir dari keseluruhan rangkaian pengujian. Setelah melakukan pengujian terhadap koneksi menggunakan PABX, Ethernet Hub dan



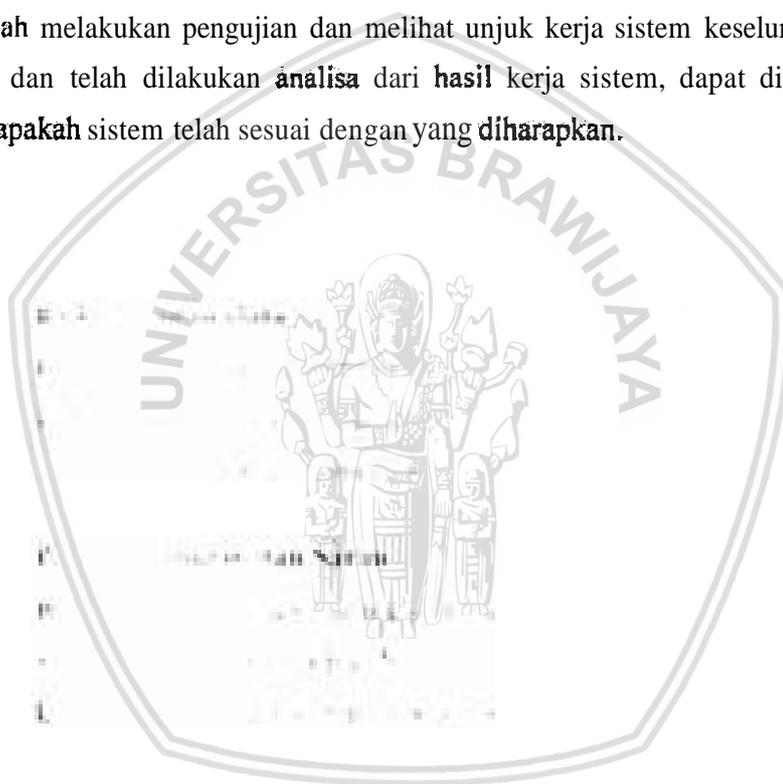
pengujian **proses** replikasi **maka** pengujian terhadap keseluruhan sistem dilakukan. Pengujian ini dilakukan untuk mengetahui kinerja sistem secara **keseluruhan**, yaitu integritas data pada *database* **setelah** terjadinya proses replikasi menggunakan koneksi dari PABX dan Ethernet Hub dengan **metode** *Virtual Private Network* (VPN).

3.5.3. Pengujian Keamanan Data

Pengujian ini merupakan pengujian **tambahan** untuk **membuktikan** bahwa dengan digunakannya VPN sebagai media koneksi **akan** menjadikan data yang lewat menjadi **aman** dari **pengambilan** oleh komputer yang lain

36. Pengambilan Kesimpulan dan Saran

Setelah melakukan pengujian dan melihat unjuk kerja sistem keseluruhan yang telah dibuat dan telah dilakukan **analisa** dari **hasil** kerja sistem, dapat ditarik suatu **kesimpulan apakah** sistem telah sesuai dengan yang **diharapkan**.



BAB IV PELANCANGAN

Perancangan dan **perealisasian** sistem dilakukan secara bertahap blok demi blok untuk memudahkan proses **analisa** sistem **per-blok** maupun sistem secara **keseluruhan**. Aspek utama yang akan dibahas dalam bab ini adalah spesifikasi sistem yang dirancang, blok diagram, **prinsip** kerja serta perancangan dan **konfigurasi** sistem.

4.1. Spesifikasi Bahan dan Alat

Dalam perancangan ini spesifikasi **bahan** dan **alat** yang akan **direalisasikan** adalah sebagai berikut:

Bahan :

- Data **pelanggan** PDAM Kabupaten Malang
- Data **transaksi** pembayaran rekening air **selama 1 tahun**

4.1.1. Perangkat keras (**Hardware**).

Perangkat kems yang digunakan **haruslah** memenuhi kebutuhan **minimum** untuk menjalankan proses replikasi maupun **software** lain yang digunakan. Dalam **Penelitian** digunakan 4 (**empat**) buah PC (Personal Computer) dengan spesifikasi yang **cukup** tinggi. PC yang **pertama** akan **digunakan** sebagai komputer **pusat** yang mempunyai data lengkap dari **masing-masing** unit. PC ke **dua** **digunakan** untuk komputer server RAS dan ke **tiga** dan **keempat** akan digunakan sebagai komputer client yaitu sebagai komputer cabang. Dalam **pelaksanaan** penelitian ini digunakan:

- Komputer **pertama**, kedua Dan ketiga sebagai client dan server **RAS** dengan spesifikasi:
 - PC dengan prosesor Intel Pentium IV (2,26 GHz), RAM 256 MB dan **Harddisk** 40 GB.
- Komputer ke tiga sebagai **server** (**komputer pusat**) dengan **spesifikasi**:
 - PC dengan prosesor Intel Pentium Core 2 duo (@ 1,61 GHz), RAM 512 MB dan **Harddisk** 80 GB.
- Untuk **proses** pengiriman data dari **komputer** unit ke komputer, **pusat** menggunakan **sistem replikasi**, dibutuhkan:
 - 3 unit modem 56 **Kbps** untuk **komunikasi** antar komputer.

- Sentral telepon yang dipakai adalah Mini PABX (buatan Cina) sebagai pengganti jaringan telepon TELKOM.
- PABX yang digunakan yaitu unit PABX merek Verophone TC-308.
- Ethernet Hub untuk komunikasi jaringan LAN

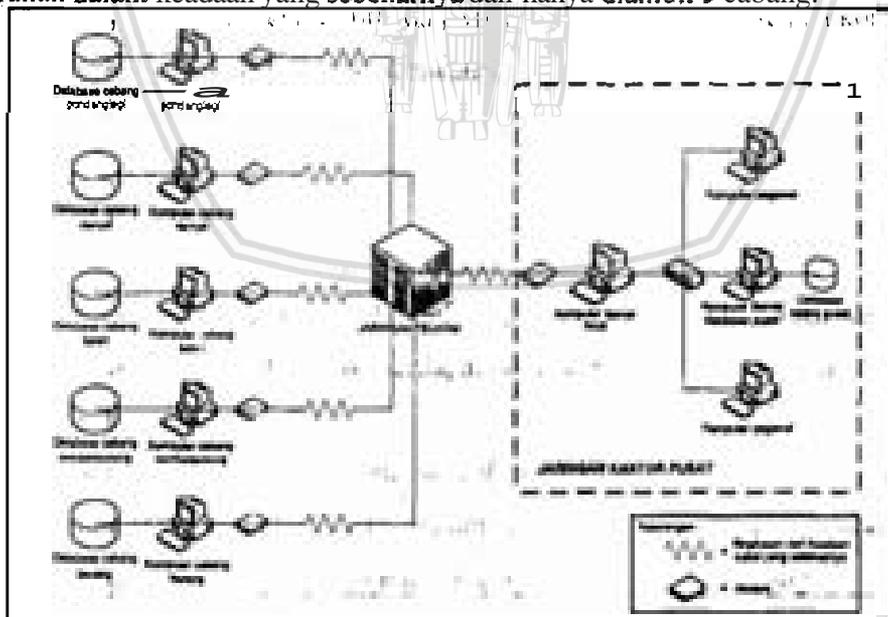
4.1.2. Perangkat lunak (*Software*)

Perangkat lunak atau *software* yang digunakan dalam penelitian ini adalah:

- Sistem Operasi : Windows 2000 Server komputer cabang dan pusat
Windows XP untuk Komputer RAS
- DBMS *software* : SQL Server 2000 dan Microsoft Access 2000 (semua komputer)
- VPN software : Open VPN 2.0.9, **Software** open source untuk menunjang VPN.
(komputer client cabang dan komputer server pusat)

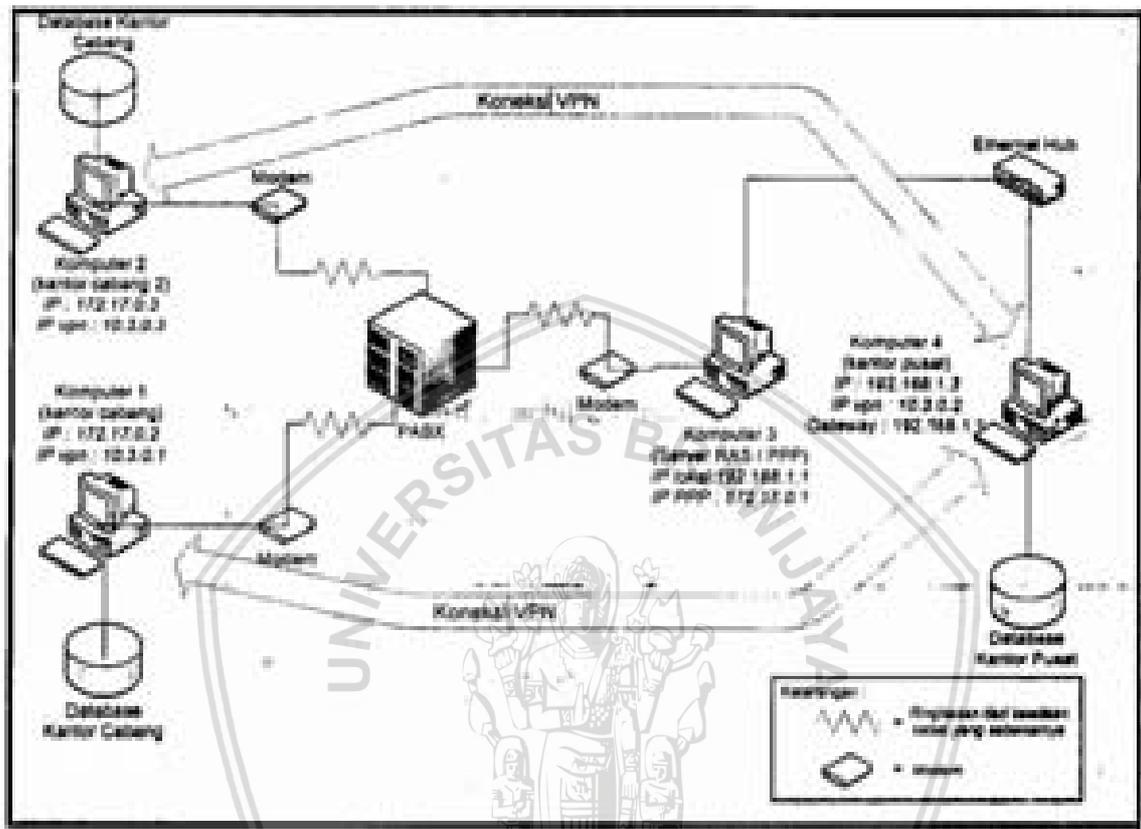
4.2. Diagram Blok Sistem

Agar perancangan dan **perealisasi sistem berjalan** secara sistematis maka perlu dirancang diagram blok yang **menjelaskan** sistem yang dirancang dibuat secara **garis besar**. Pada keadaan yang sebenarnya terdapat beberapa kantor cabang yang melakukan replikasi basis data dengan kantor pusat. Pada awalnya koneksi antara kantor pusat dan antar cabang masih menggunakan koneksi *dial up* untuk dapat saling berhubungan dan melakukan replikasi. Gambar 4.1. menunjukkan blok diagram sistem secara keseluruhan dalam keadaan yang sebenarnya dan hanya diambil 5 cabang.



Gambar 4.1. Blok diagram contoh sistem dalam keadaan nyata (hanya diambil 5 cabang).
Sumber: Perancangan

Dalam percobaan ini hanya diambil 2 cabang aja yang akan dikoneksikan ke komputer pusat Gambar 4.2. menunjukkan blok diagram sistem secara keseluruhan yang akan digunakan dalam riset.



Gambar 4.2. Blok diagram sistem secara keseluruhan yang akan digunakan dalam riset.
Sumber: Perancangan

Sistem secara garis besar terdiri atas komputer pusat (komputer 1), komputer cabang (komputer 3 dan computer 4) dan komputer yang berfungsi sebagai *server RAS* (computer 2). Masing-masing komputer memiliki spesifikasi sistem operasi dan *database server* yang sama, yaitu Windows 2000 Server dan SQL Server 2000. Kecuali pada komputer 2 yang berfungsi sebagai *sewer RAS* untuk dapat menghubungkan komputer cabang dengan komputer pusat .

Pada computer cabang dengan computer *server RAS* terhubung ke jaringan komputer dengan menggunakan modem, protokol komunikasi yang digunakan adalah TCP/IP dengan jenis koneksi *point to point*. PABX digunakan sebagai sentral telepon pengganti jaringan telepon TELKOM. Sedangkan dari *server RAS* dan komputer pusat

terkoneksi melalui jaringan lokal./ LAN dimana nanti antara *server RAS* dengan komputer pusat akan terhubung secara privat.

4.3. Cara Kerja Sistem

Cara kerja sistem secara garis besar adalah sebagai berikut:

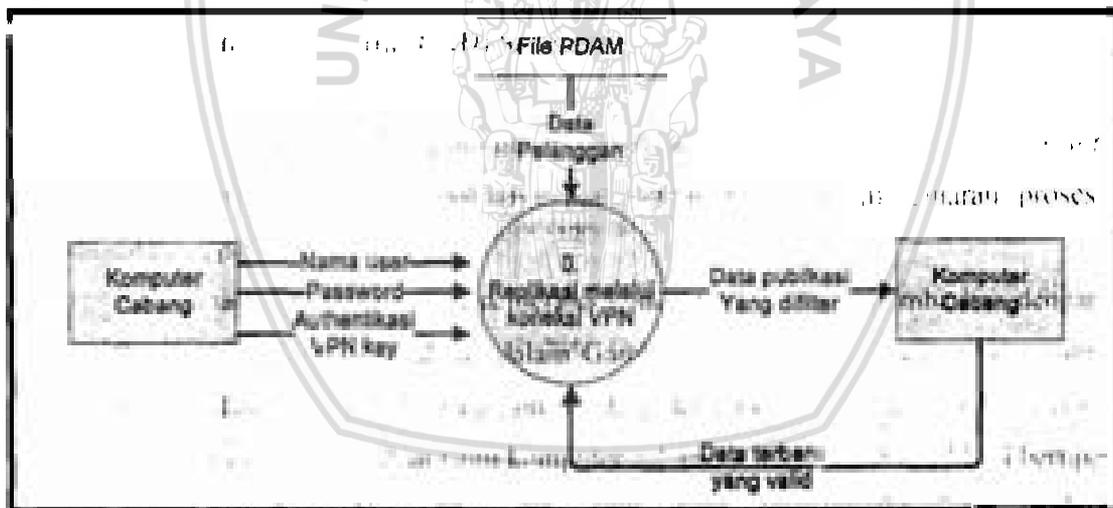
1. Aplikasi pada komputer-komputer unit melakukan transaksi (**pemrosesan** data) terhadap **database server** yang adapada Komputer cabang 1 (komputer 1) dan Komputer cabang 2 (komputer 2) yaitu dengan melakukan penyimpnnan dnn penghapusan data pada **database server**. Penyimpanan dan penghapusan data semuanya dikerjakan secara **offline**, tidak terhubung dengan jaringan.
2. Setelah transaksi **selesai** dan database **telah** terupdate, **secara** otomatis komputer-komputer cabang akan melakukan **dial up** ke jaringan komputer pusat menggunakan modem melalui jaringan telepon dimana koneksi dial akan diterima **oleh** komputer **server RAS** yang kemudian akan diteruskan menuju komputer pusat. Proses Dial Up dilakukan **bergantian sesuai** alokasi waktu.
3. Setelah komputer-komputer, cabang terhubung dengan jaringan komputer pusat, komputer cabang akan melakukan mengadakan hubungan privar (VPN) dengan komputer pusat menggunakan **software Open VPN**. Proses VPN dilakukan bergantian sesuai alokasi waktu **sesuai setelah** koneksi dial up berlangsung.
4. Setelah komputer-komputer cabang terhubung secara VPN dengan komputer pusat, proses replikasi dengan tipe **merge** dilakukan. Karena dengan tipe ini **update** data bisa dilakukan dua arah, perubahan data pada **database** di komputer pusat akan berpengaruh pada perubahan data pada **database** di komputer cabang, begitupun sebaliknya.
5. Selama proses replikasi, komputer cabang akan mengirimkan data-data terbaru yang mereka miliki ke kantor pusat, demikian juga sebaliknya.
6. Apabila terjadi konflik pada saat proses replikasi terjadi antara komputer di kantor pusat (Komputer 4) dan kantor cabang (Komputer 1 dan komputer 2), dimana keduanya melakukan perubahan pada data yang sama, misalkan komputer pusat dan komputer-komputer cabang sama-sama melakukan proses **update** pada data yang sama maka **Merge Agent** dari **SQL Server** yang ada pada komputer pusat akan menentukan data mana yang valid untuk digunakan.

7. Setelah replikasi dilakukan oleh komputer kantor pusat ke semua komputer cabang, maka data yang ada di *database* komputer kantor pusat merupakan data terbaru yang merupakan kombinasi dari keseluruhan cabang.
8. Proses replikasi selesai dan komunikasi diakhiri.
9. Masing-masing komputer kembali melakukan proses transaksi mereka sendiri.

4.4. Diagram Aliran Data (DAD) Sistem

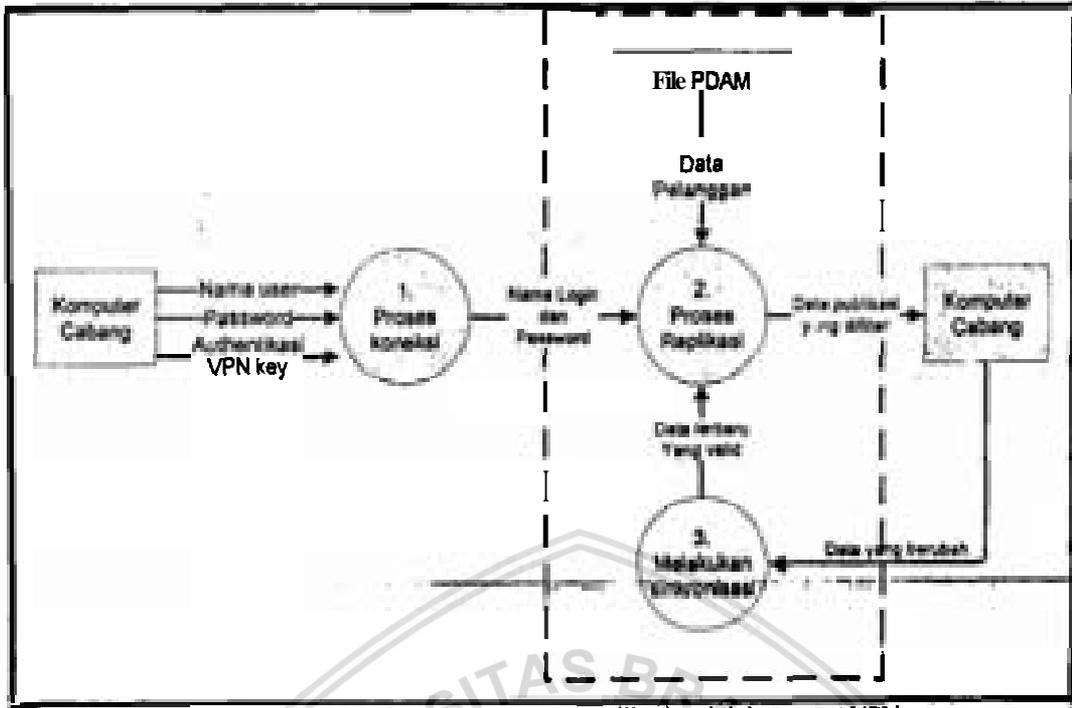
Pada perancangan sistem dirancang diagram alir yang dapat mempermudah pengembangan sistem dan dapat digunakan untuk analisa sistem dimana diagram alir digambarkan dengan simbol-simbol dasar baik entitas masukan dan keluaran, proses, aliran data dan penyimpanan.

Secara umum sistem replikasi melalui koneksi VPN bisa digambarkan dengan diagram konteks seperti ditunjukkan dalam Gambar 4.3, dimana proses secara umum ditunjukkan dalam satu proses yaitu proses Replikasi melalui koneksi VPN dalam komputer pusat dengan masukan yaitu komputer cabang, karena proses replikasi bertipe *merge* yaitu *update* secara dua arah maka komputer cabang merupakan keluaran dari proses dan akan dijadikan masukan untuk proses untuk *update* pada komputer pusat.



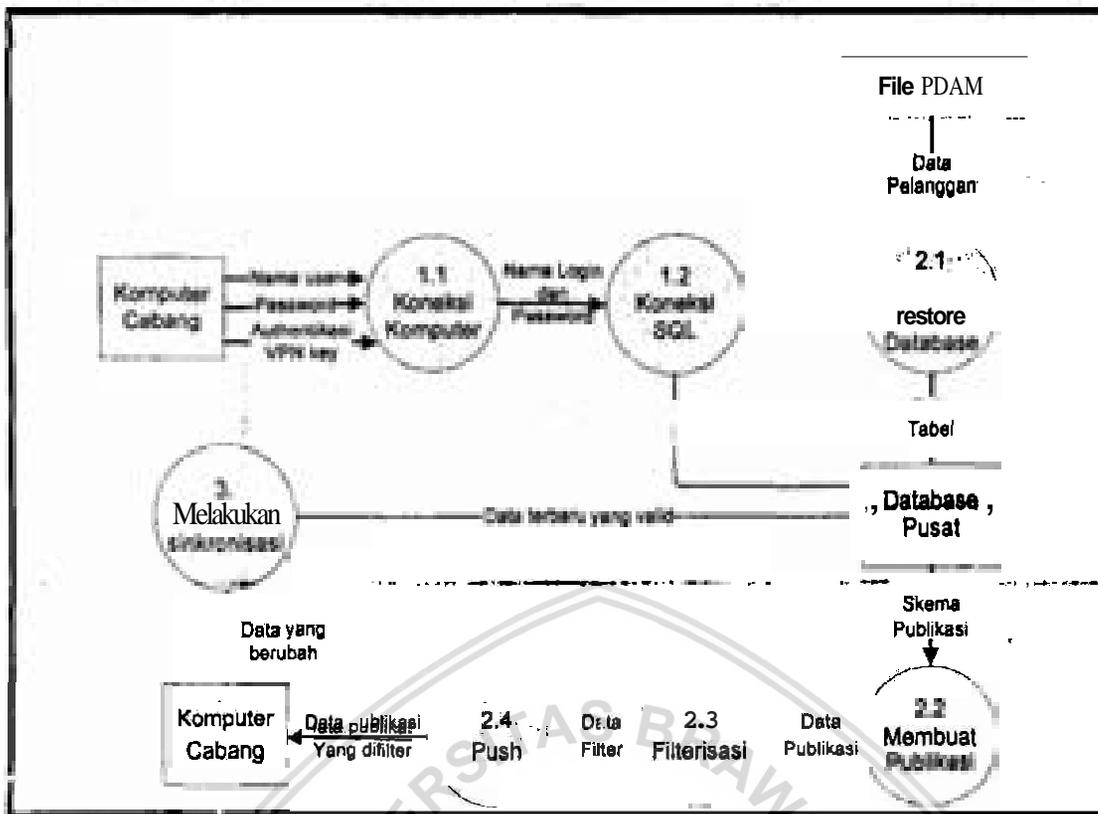
Gambar 4.3. DFD level 0 proses replikasi melalui koneksi VPN
Sumber: Perancangan

Dari DFD level 0 tersebut kemudian dikembangkan agar sistem bisa digambarkan lebih detail yaitu seperti ditunjukkan dalam Gambar 4.4, yaitu DFD level 1 dimana proses Replikasi melalui Koneksi VPN bisa dijelaskan menjadi tiga proses yaitu Proses Koneksi dan Proses Replikasi dan proses Melakukan Sinkronisasi.



Gambar 4.4. DFD level 1 dari proses Replikasi melalui Koneksi VPN
Sumber: Perancangan

Untuk sistem secara mendetail bisa dilihat dalam Gambar 4.5. yaitu DFD level 2 dimana proses Koneksi sendiri bisa dijelaskan sebagai Koneksi Komputer dan Koneksi SQL sedangkan proses Replikasi dijelaskan lebih detail menjadi tahapan-tahapan yaitu proses Melakukan *Restore Database*, Membuat Publikasi, proses Filterisasi dan proses *Push*. Gambar 4.5. menunjukkan tahapan membuat replikasi mulai awal/konfigurasi awal, sedangkan sistem secara keseluruhan, setiap melakukan koneksi antara komputer cabang dan komputer pusat adalah setelah proses koneksi maka akan dilanjutkan ke proses filterisasi, *push* dan sinkronisasi tanpa melewati pembuatan publikasi dan *restore database*.



Gambar 4.5 DFD level 2 dari proses Replikasi melalui Koneksi VPN

Sumber: Perancangan

4.5. Perancangan dan Konfigurasi Sistem

Perancangan dan konfigurasi meliputi perencanaan jaringan menggunakan koneksi PPP antara komputer di kantor cabang dengan komputer *server RAS* yang terhubung melalui jaringan lokal (LAN) dengan komputer di pusat yang kemudian akan diadakan hubungan privat (VPN) antara komputer cabang dengan komputer pusat konfigurasi *database* yang akan digunakan dalam replikasi dan konfigurasi dari proses replikasi itu sendiri.

4.5.1. Perancangan Jaringan

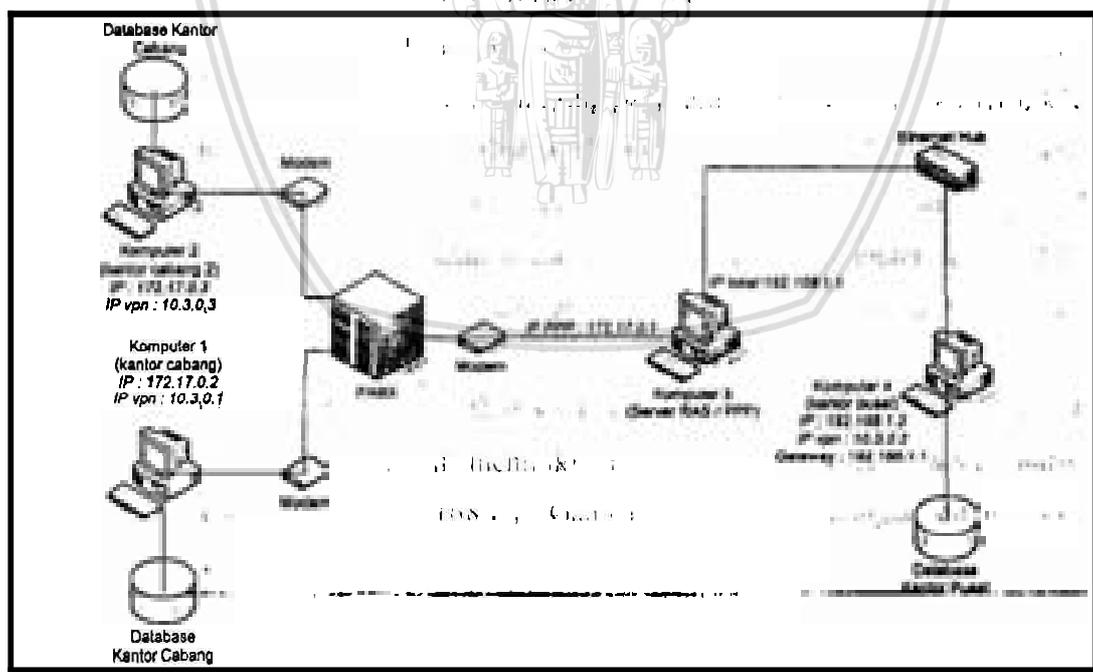
4.5.1.1. Topologi Jaringan

Dalam perancangan ini topologi jaringan yang digunakan secara fisik bentuknya seperti tipe straight, dimana PABX berfungsi sebagai penghubung antara komputer cabang dan komputer *server RAS*, dan komputer *server RAS* dengan komputer pusat terhubung melalui jaringan lokal / LAN dimana terhubung melalui *ethernet* hub. Untuk komputer cabang dengan komputer *server RAS* akan menggunakan protokol *point to point* dimana komputer cabang akan melakukan koneksi ke komputer *server RAS*. Karena antara komputer cabang telah terkoneksi secara *point to point* dengan



komputer *server RAS* maka komputer cabang telah dianggap menjadi satu jaringan dengan komputer RAS yang ada di jaringan kantor pusat tersebut karena telah terhubung melalui media modem dan koneksi dial up, sedangkan komputer RAS dengan komputer-komputer pada jaringan komputer pusat merupakan network sendiri. Untuk dapat terkoneksi secara khusus dengan komputer pusat yang memiliki database pusat diperlukan hubungan yang dilakukan secara privat atau yang biasa kita sebut dengan VPN. Penggunaan VPN ditujukan untuk menghubungkan dua komputer pada cabang dan pusat yang terdapat pada network yang berbeda.

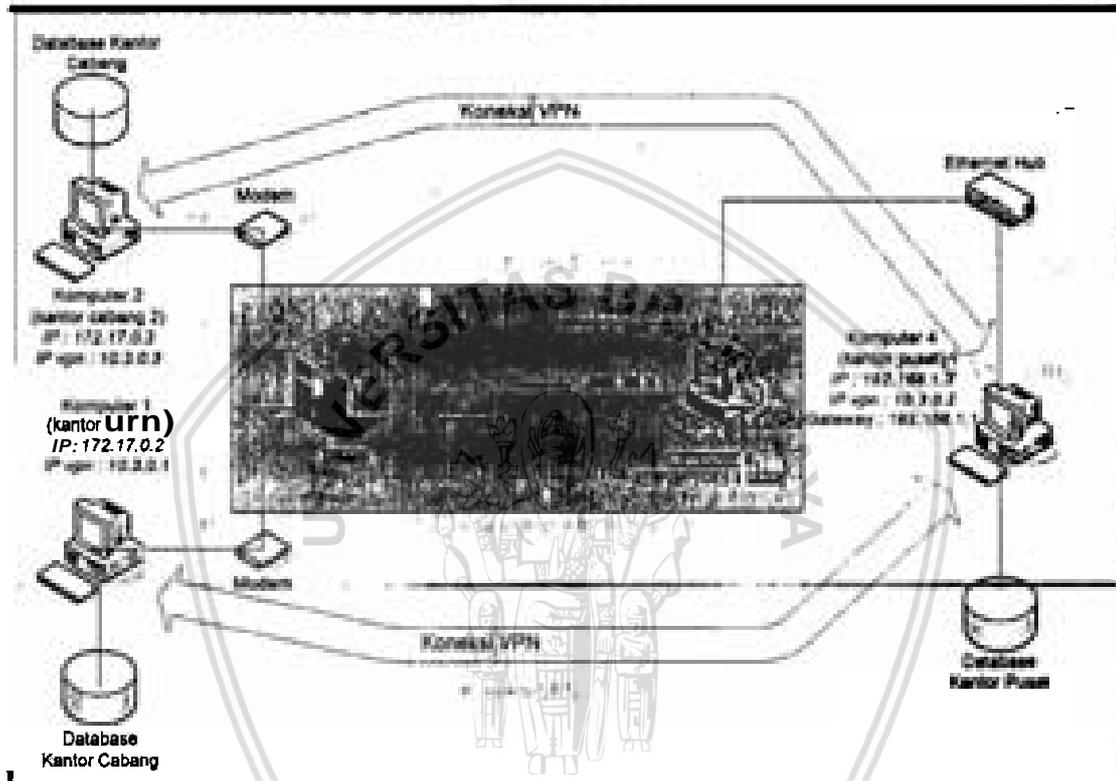
PABX yang digunakan dalam perancangan ini memiliki 8 buah port, dan yang digunakan hanya 3 port, dimana masing-masing port terhubung ke nomor ekstensi 82, 83 dan 84. Komputer cabang 1 akan terhubung ke nomor ekstensi 82, Komputer cabang 2 akan terhubung ke nomor ekstensi 83 dan komputer *server RAS* akan terhubung dengan nomor ekstensi 84. kemudian untuk hubungan antara komputer *server RAS* dengan komputer pusat terhubung melalui jaringan lokal LAN. Dimana 2 komputer tersebut memiliki IP masing-masing untuk komputer *server RAS* memiliki 2 IP yaitu IP PPP 172.168.0.1 dan IP lokal 192.168.1.1, dengan subnet mask 255.255.255.0 sedangkan untuk komputer pusat memiliki IP 192.168.1.3 dengan subnet mask 255.255.255.0 dan gateway 192.168.1.1. Gambar konfigurasi jaringan ditunjukkan dalam Gambar.4.6. berikut:



Gambar 4.6. Topologi dan konfigurasi jaringan.
Sumber: Perancangan

4.5.1.2. Pengkoneksian VPN

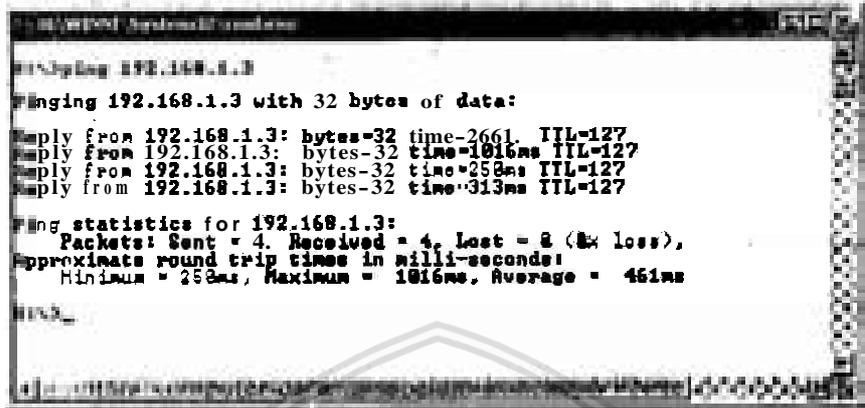
Koneksi VPN dapat dilakukan **setelah** komputer cabang telah terkoneksi dengan jaringan komputer lokal di kantor **pusat**. Jadi **setelah** koneksi point to **point** telah dilakukan oleh komputer cabang yang men-dial-up komputer server RAS yang **dianggap** sebagai server dial-up **maka** dapat dilakukan pengkoneksian komputer cabang dengan komputer **pusat** yang berada di jaringan komputer di kantor **pusat**. **Gambaran** keadaan jaringan saat terkoneksi VPN ditunjukkan dalam Gambar 4.7. berikut:



Gambar 4.7. Gambaran keadaan jaringan saat terkoneksi secara VPN.
Sumber: Perancangan

Dalam **melakukan** koneksi VPN ini digunakanlah software **OpenVPN 2.0.9**, dimana **aplikasi** tersebut dapat membuat koneksi point-to-point **tunnel** yang telah terenkripsi. **OpenVPN** menggunakan private keys, certificate, **atau** **username/password** untuk melakukan autentikasi dalam membangun koneksi, Dimana untuk **enkripsi** menggunakan **OpenSSL**. Pengkonfigurasi awal dilakukan di komputer cabang. Untuk **setting** konfigurasi pada OpenVPN ini dilakukan secara manual dengan **menggunakan notepad**. Konfigurasi yang **dilakukan** **terdahulu** adalah pemilihan jenis device VPN dan setting IP VPN. Untuk pemilihan jenis device **VPN** yang akan dipakai tergantung dari **kebutuhan**. Untuk device yang **digunakan** untuk **sistem** pada PDAM ini adalah device

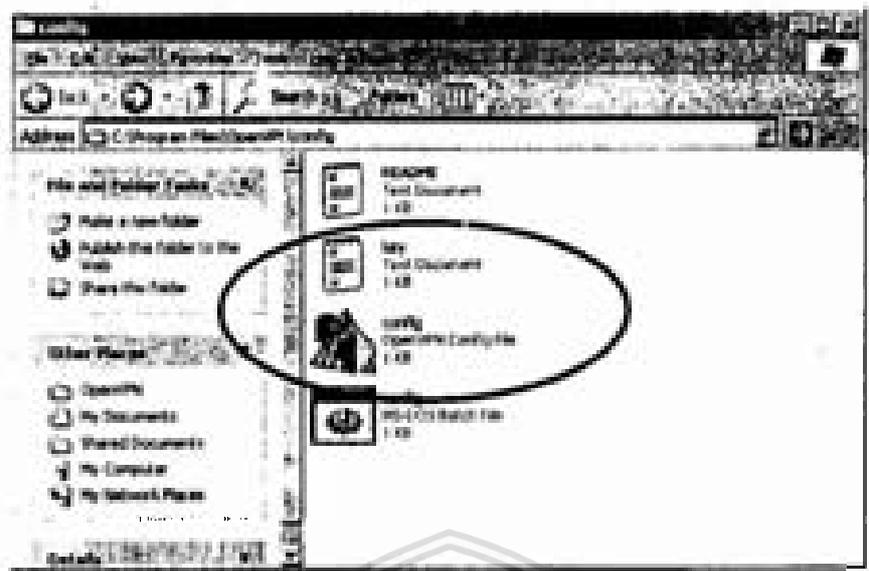
TAP, karena antara komputer cabang sebelum koneksi VPN telah melakukan koneksi Dial Up dengan komputer RAS yang terdapat pada komputer pusat. Jadi antara komputer cabang dan komputer pusat sudah mengetahui satu sama lain yang dapat dilihat menggunakan perintah ping seperti pada gambar 4.8 berikut



Gambar 4.8. Komputer cabang melakukan Ping ke komputer pusat. .
Sumber: Perancangan

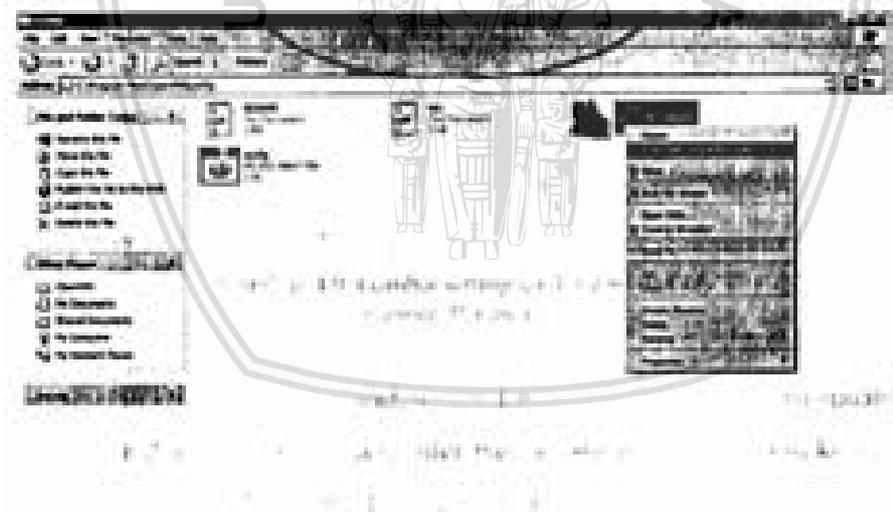
Pada setting IP VPN ini untuk komputer cabang 1 memiliki IP VPN 10.3.0.1 dengan subnet mask 255.255.255.0, komputer cabang 2 memiliki IP VPN 10.3.0.3 dengan subnet mask 255.255.255.0 dan komputer pusat untuk setting IP VPN memiliki IP 10.3.0.2 dengan subnet mask 255.255.255.0. Setelah dikonfigurasi dengan beberapa perintah yang dimiliki OpenVPN dilakukanlah generate OpenVPN static key, maksudnya adalah membuat private key yang dapat membuka jalan untuk dapat terkoneksi secara VPN. Jadi komputer cabang men-generate key terlebih dahulu yang kemudian diberi nama misalnya secret key.txt, kemudian key yang telah dibuat itu diberikan juga kepada komputer pusat supaya kedua komputer ini memiliki key yang sama karena kunci dari jalannya koneksi VPN ini adalah autentikasi key ini. Oleh karena itu kedua komputer ini harus memiliki key yang sama. Setting konfigurasi dan key terdapat pada folder yang sama seperti yang ditunjukkan pada gambar 4.9. berikut :





Gambar 4.9. Gambar setting konfigurasi dan key.
Sumber: Perancangan

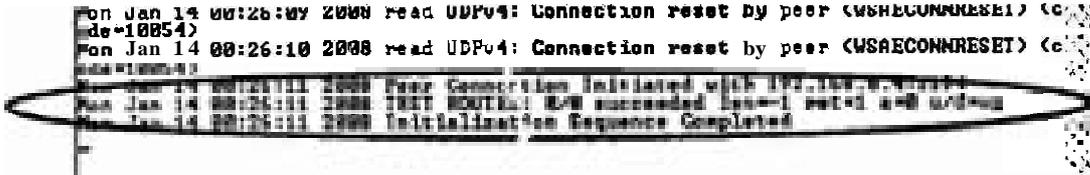
kemudian setelah kedua komputer sudah siap dan komputer cabang juga sudah men-dial komputer server KAS, maka tiap komputer baik komputer cabang atau komputer pusat melakukan *slur-OpenVPN* seperti ditunjukkan gambar 4.10. berikut :



Gambar 4.10. Gambar start OpenVPN.
Sumber: Perancangan

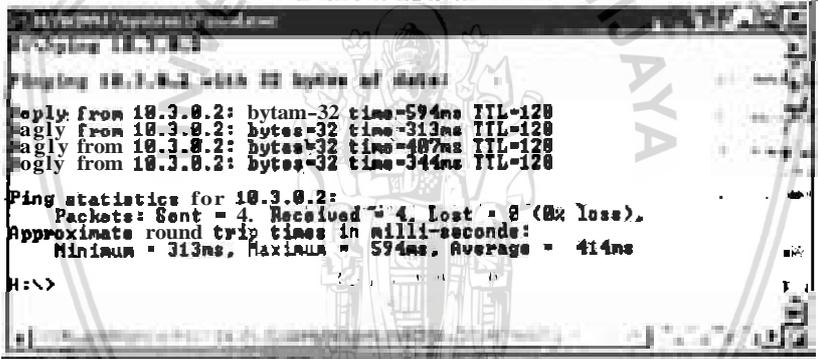


Kemudian setelah dilakukan *start* oleh kedua komputer tersebut baik komputer cabang dan komputer pusat. Akan keluar status koneksi apakah telah berhasil atau tidak. Jika berhasil maka akan keluar status yang ditunjukkan pada gambar 4.11. dimana koneksi VPN tersebut telah berhasil dan terkoneksi dengan benar.



Gambar 4.11. Gambar status koneksi OpenVPN yang berhasil.
Sumber: Perancangan

Untuk membuktikan sudah terhubung secara VPN selain dengan melihat status tersebut, yaitu dengan melakukan perintah ping langsung ke IP VPN tersebut dimana hanya bisa dilakukan oleh komputer cabang dan komputer pusat. Pada gambar 4.12, menunjukkan hasil *commandping* pada saat telah terkoneksi VPN.



Gambar 4.12. Komputer cabang melakukan Ping IP VPN komputer pusat.
Sumber: Perancangan

4.5.2. Perancangan Database

4.5.2.1. Skema Tabel

Tabel yang digunakan dalam sistem ini terdiri atas 11 tabel utama, dimana 6 tabel diantaranya saling berhubungan dan memiliki *referential integrity*. Tabel utama yang digunakan sebanyak 2 tabel, yaitu Tabel Pelanggan dan Tabel Pemakaian. Diagram relasi antara tabel ditunjukkan dalam Gambar 4.13. dan normalisasi tabel database PDAM ditunjukkan dalam gambar 4.14 berikut:

4.5.2.2. Penentuan primary *key*

Dalam perancangan tabel-tabel tersebut, hal yang perlu mendapatkan perhatian adalah penentuan penggunaan primary *key*, karena tabel di atas akan direplikasikan pada keseluruhan unit. Penentuan *sl.ema* primary *key* dilakukan untuk menghindari terjadinya data yang memiliki primary *key* yang sama antara unit-unit saat direplikasi.

Tabel yang paling utama adalah Tabel Pelanggan, dimana tabel ini menyimpan informasi unik mengenai pelanggan. Tabel ini terdiri atas 15 kolom dimana Id_pelanggan merupakan primary *key* dengan tipe data *nvarchar*.

Format yang digunakan untuk kolom Id_pelanggan adalah sebagai berikut:

contoh: A-135-61667, dimana kode kecamatan adalah 'A', kode desa '135' dan nomor urut '61667'.

Kode kecamatan memiliki range antara A sampai dengan V, kode desa dimulai dari 1, sedangkan nomor urut dimulai dari 1 sampai tak terhingga. Dengan format seperti ini maka masing-masing unit akan memiliki data yang dijamin berbeda dengan unit lainnya, sehingga tidak akan ada konflik data pada saat dilakukan replikasi.

Untuk menghasilkan primary *key* dengan format tersebut di atas dan secara otomatis dihasilkan saat dilakukan insert data maka digunakan sebuah trigger. Trigger ini bertanggung jawab untuk menghasilkan primary *key* sesuai dengan kode kecamatan yang diberikan. Statemen untuk membuat trigger tersebut adalah:

```
CREATE TRIGGER [Pelanggan_tri] ON [dbo].[Pelanggan]
FOR INSERT
AS
BEGIN
    declare @new_kode_kec varchar(2), @new_kode_desa int,
    @new_id_pelanggan int
    select @new_kode_kec=kode_kec from inserted
    select @new_kode_desa=kode_desa from inserted
    select
    @new_id_pelanggan=isnull(max(cast(substring(substring(id_pelanggan
    ,3,len(id_pelanggan)),charindex('-',
    ,substring(id_pelanggan,3,len(id_pelanggan)))+1,len(id_pelanggan)
    ) as int)),0)+1 from Pelanggan
```

```

update Pelanggan set id_pelanggan=@new_kode_kec + '-' +
cast(@new_kode_desa as varchar) + '-' + cast(@new_id_pelanggan as
varchar) where id_pelanggan='0'
END
GO

```

Trigger ini akan dijalankan setiap kali terjadi proses insert pada Tabel Pelanggan. Trigger ini merupakan jenis trigger AFTER, dimana trigger akan dijalankan sesaat setelah proses insert dilakukan. Terdapat lima baris pernyataan SQL di dalam badan trigger di atas, dengan penjelasan sebagai berikut:

- Baris 1

```

declare @new_kode_kec varchar(10),
        @new_kode_desa int,
        @new_id_pelanggan int

```

Pernyataan ini mendeklarasikan tiga buah variabel, yaitu @new_kode_kec, @new_kode_desa dan @new_id_pelanggan, dimana kode_kec bertipe varchar sedangkan kode_desa dan id_pelanggan bertipe int. Ketiga variabel ini akan digunakan untuk membentuk *primary key* yang merupakan gabungan dari huruf dan angka.

- Baris 2 dan 3

```

select @new_kode_kec=kode_kec from inserted
select @new_kode_desa=kode_desa from inserted

```

Tabel *inserted* merupakan tabel spesial yang secara otomatis akan dibuat oleh SQL Server saat dilakukan eksekusi trigger, tabel ini menyimpan salinan data dari baris yang akan di-insert ke tabel sebenarnya. Dengan menggunakan pernyataan di atas maka Kode_kec dan Kode_desa dari data yang akan disimpan ke tabel dapat diakses dan disimpan nilainya ke variabel @new_kode_kec dan @new_kode_desa.

- Baris 4

```

set rowcount 1
select @new_id_pelanggan=id_pelanggan from inserted
set rowcount 0

```

Pernyataan ini merupakan pernyataan yang paling kompleks yang fungsinya untuk menghasilkan angka yang berurutan nilainya dari Tabel Pelanggan itu sendiri dengan kenaikan sebesar 1. Sebagaimana dijelaskan di atas bahwa kolom Id_pelanggan memiliki format yang merupakan gabungan dari Kode_kec, kode_desa dan angka berurutan. Contoh data dari kolom Id_pelanggan dapat dilihat pada Tabel 4.1.

Tabel 4.1 Contoh data di kolom id_pelanggan pada Tabel Pelanggan.

A-135-61667
A-135-61668
A-135-61669
A-135-61670
A-135-61671
A-135-61672
A-135-61673
A-135-61674
A-135-61675
A-135-61676
A-135-61677
A-135-61678
A-135-61679

Untuk menghasilkan angka yang berurutan dari data seperti di atas maka perlu dilakukan pemotongan karakter terlebih dahulu, dimana karakter paling depan yang merupakan Kode_kec dan kode_desa harus dihilangkan terlebih dahulu. Untuk menghasilkan karakter yang terpotong tersebut digunakan fungsi substring. Sehingga dengan pernyataan

```

SUBSTRING (ID_PELANGGAN FROM 4 FOR 4)

```

akan dihasilkan data sebagaimana ditunjukkan pada Tabel 4.2.

Tabel 4.2 Data di kolom id_pelanggan setelah dikenakan fungsi substring.

61667
61668
61669
61670
61671
61672
61673
61674
61675
61676
61677
61678

Untuk memperoleh angka yang paling tinggi maka digunakan fungsi max() terhadap kolom Id_pelanggan dan sebelumnya kolom tersebut harus diubah ke tipe data



integer, dengan menggunakan fungsi `cast ()`. Dan untuk memperoleh angka yang baru maka angka tertinggi yang diperoleh **dijumlahkan** dengan 1. Sehingga pernyataan SQL akan menjadi:

```
select max(cast (kode_kecamatan as integer) + 1 as integer) as kode_kec,
       max(id_pelanggan) as id_pelanggan,
       len(id_pelanggan) as len_id_pelanggan
```

Pernyataan di atas akan menghasilkan nilai angka baru, dimana nilainya merupakan kenaikan **satu** angka dari angka tertinggi yang saat ini ada di tabel. Akan tetapi pada saat belum ada data sama sekali di kolom `Id_pelanggan`, pernyataan di atas akan menghasilkan nilai **NULL**. Sehingga perlu ditambahkan fungsi pengecekan dengan `isnull ()` yang akan **mengganti nilai NULL** dengan **nilai 0**.

Gabungan dari semua fungsi di atas akan menghasilkan pernyataan yang ada di baris 4 dari badan trigger di atas.

■ Baris 5

```
update Pelanggan set kode_kecamatan = (select kode_kecamatan from
(select max(cast (kode_kecamatan as integer) + 1 as integer) as kode_kec,
       max(id_pelanggan) as id_pelanggan,
       len(id_pelanggan) as len_id_pelanggan
```

Pernyataan ini merupakan pernyataan **akhir** yang akan **menggabungkan** karakter `kode_kec`, `kode_desa` dan nilai angka yang dihasilkan dari pernyataan di Baris 4, dimana antara ketiga kode tersebut dipisahkan dengan karakter pembatas `'-'`. Gabungan karakter tersebut akan di-update ke baris terbaru pada Pelanggan, dimana baris paling baru akan selalu memiliki `id_pelanggan` default '0' sebagaimana didefinisikan saat pembuatan Tabel Pelanggan. Pada saat keseluruhan trigger selesai dieksekusi maka tidak akan ada `id_pelanggan` yang memiliki nilai default '0', karena akan di-update oleh pernyataan di atas, sehingga menjadi kombinasi `kode_kec`, `kode_desa` dan angka berurutan.



**BAB V
IMPLEMENTASI**

Bab ini membahas mengenai implementasi Replikasi basis data melalui jaringan *Virtual Private Network (VPN)*. Implementasi dilakukan menggunakan *hardware* dan *software* yang telah disebutkan pada bab sebelumnya yaitu untuk *hardware* terdiri dari 4 buah PC, 3 buah modem, 1 buah PABX dan 1 buah ethernet hub. Sedangkan untuk *software* terdiri dari sistem operasi Microsoft Windows 2000, Microsoft SQL server 2000, Microsoft Access 2000 dan Open VPN 2.0.9.

5.1. Konfigurasi Dial Up Server (Server RAS)

Untuk dapat membentuk suatu jaringan antara komputer cabang dengan komputer pusat dengan menggunakan metode *dial up*, maka pada jaringan komputer pusat ada 1 komputer dikonstruksikan sebagai *server* untuk menerima hubungan *dial up* dari komputer cabang. Komputer di cabang melakukan *dial up* ke komputer pusat melalui PABX dengan menggunakan *account* yang ada di komputer *server RAS*.

Pada komputer *server RAS* dibuat *account* baru yang akan digunakan oleh komputer cabang. Konfigurasi *user* dan *password* dapat dilihat pada Tabel 4.1. berikut:

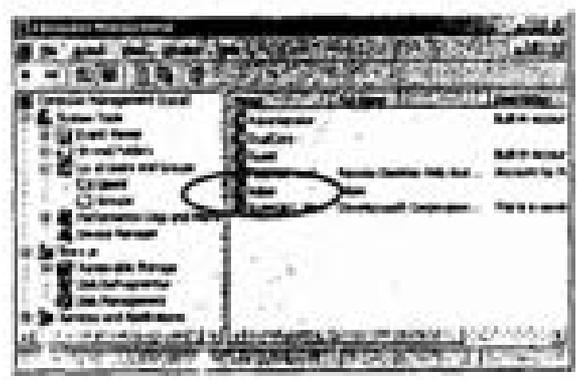
Tabel 5.1. Daftar *account* di komputer pusat untuk digunakan komputer cabang.

Komputer	User	Password	Group
Komputer 1	pdam1	pdam1	PDAM
Komputer 2	pdam2	pdam2	PDAM

Konfigurasi pada Windows Server 2000 dilakukan pada sub menu *Komputer Management* di menu *Administrative Tool* pada *Control Panel*, sebagaimana ditunjukkan dalam Gambar 5.1. berikut:

Gambar 5.1. Screenshot tampilan menu Administrative Tools

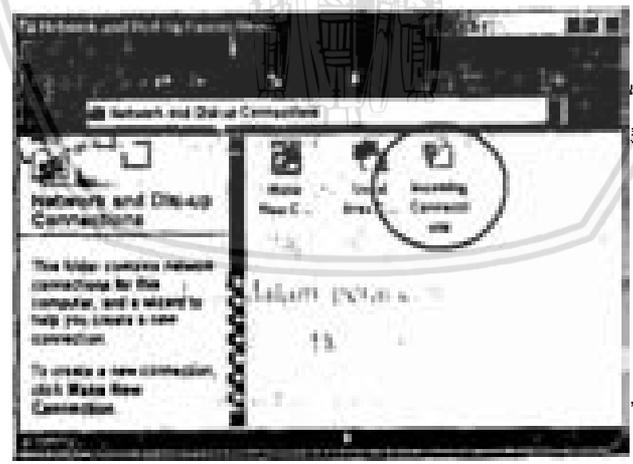




Gambar 5.1. Konfigurasi account pada komputer pusat.
Sumber: Implementasi

Untuk dapat melayani permintaan *dial up* dari komputer cabang, maka pada komputer *server* server RAS harus dikonfigurasi untuk menerima jenis koneksi ini. Komputer cabang yang akan melakukan *dial up* ke komputer *server* RAS menggunakan nomor ekstensi dan *account* sebagaimana yang telah ditunjukkan pada Tabel 5.1 di atas.

Pada Windows 2000 Server dapat dilakukan konfigurasi untuk menerima *Incoming Connection* pada modem, dalam perancangan ini modem digunakan untuk menangani semua permintaan koneksi. Dengan menggunakan wizard *Make New Connection* yang tersedia pada menu *Network and Dial-Up Connection* maka Komputer Pusat siap menangani semua permintaan koneksi. Koneksi yang telah dibentuk dapat dilihat dalam Gambar 5.2. yaitu di menu *Network and Dial-Up Connection* sebagai berikut:

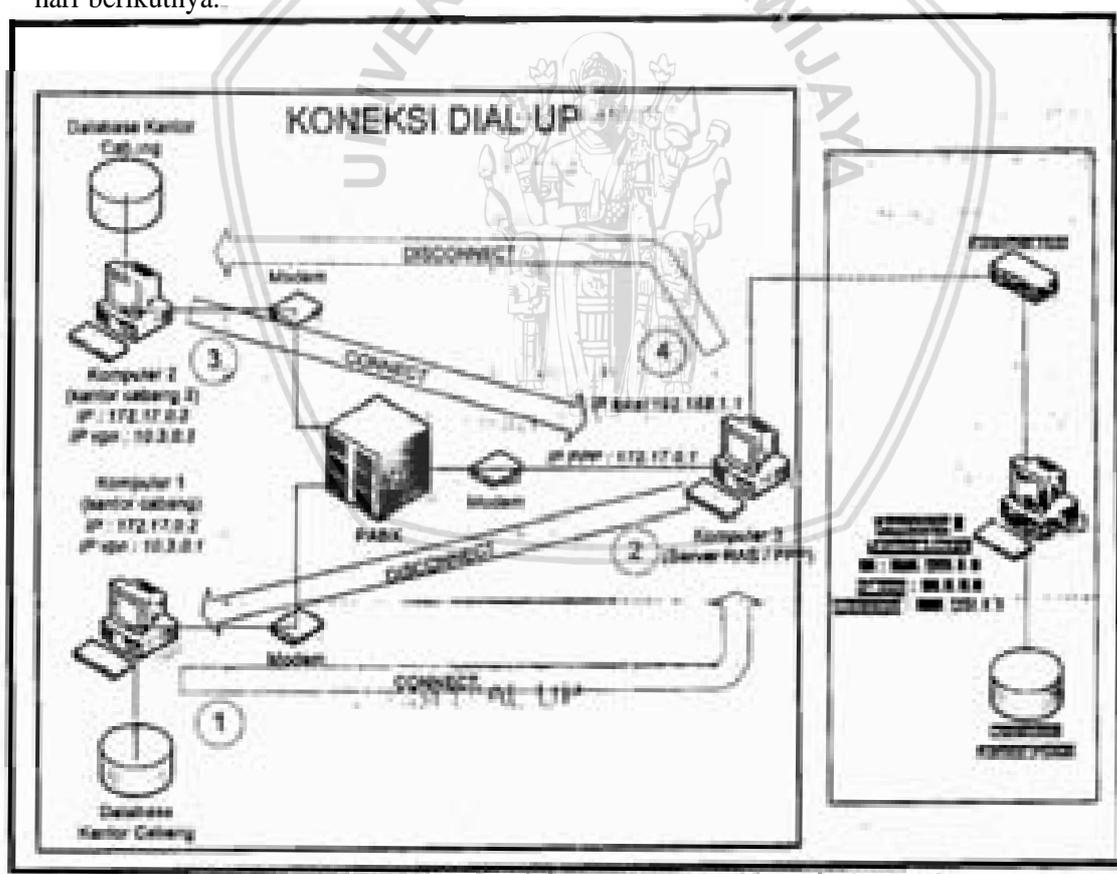


Gambar 5.2. Koneksi untuk melayani *dial-up* pada komputer pusat.
Sumber: Implementasi

5.1.1. Penjadwalan Dial Up

Proses koneksi **dirancang** dengan mekanisme penjadwalan koneksi **dial up**, dimana komputer cabang memiliki **alokasi** waktu tersendiri untuk **melakukan** koneksi ke komputer server RAS. Diagram **alir** penjadwalan koneksi dapat dilihat dalam Gambar 5.3.

Koneksi pertama kali dilakukan oleh Komputer 3, dengan melakukan **dial up** ke Komputer I-pusat (langkah 1) pada waktu yang telah ditentukan pada Tabel 5.2. Setelah terjadi koneksi maka proses replikasi akan segera dilakukan, dimana proses replikasi ini mendapatkan alokasi waktu yang telah ditentukan juga. Setelah alokasi waktu koneksi untuk Komputer 3 telah **habis** maka Komputer 3 akan **meminta** pemutusan koneksi dan Komputer 1 akan memutuskan koneksi dengan Komputer 2 (langkah 2), begitu juga dengan proses komputer 3 dan komputer 2. Apabila proses **update** data belum **selesai** pada saat replikasi **berlangsung** dengan alokasi waktu yang telah **ditentukan** maka update data akan dilanjutkan pada proses replikasi pada koneksi berikutnya yaitu pada hari berikutnya.



Gambar 5.3. Urutan proses koneksi antara komputer cabang dengan pusat.
Sumber: Implementasi



Dalam perancangan ini komputer cabang dijadwalkan untuk melakukan **satu** kali koneksi per hari. Pada malam hari **pukul 18.00 setelah transaksi** pada cabang **selesai dikerjakan**. Koneksi masing-masing dilakukan selama **30 menit**.

Pertimbangan waktu koneksi komputer **selama 30 menit adalah** dengan perhitungan berikut:

Modem yang digunakan dalam perancangan ini memiliki kecepatan **56 Kbps** (kilo *bits per second*), sehingga secara teori dapat melakukan transfer data sebanyak **56 kilo bit** untuk tiap detiknya. Akan tetapi kecepatan *Shi* diperoleh pada kenyataannya sangat tergantung pada berbagai macam faktor, yaitu kecepatan modem yang didukung oleh ISP (*Internet Service Provider*) penyedia jasa dial up, kecepatan tertinggi modem lain yang melakukan koneksi (modulation fallback) dan faktor noise dari jaringan telepon yang digunakan.

Apabila dua buah modem saling berkomunikasi dalam suatu koneksi, dimana keduanya memiliki kecepatan yang berbeda, maka kecepatan yang akan digunakan adalah kecepatan yang paling rendah, hal ini disebut juga modulation *fallback*. Dengan berbagai macam gangguan sebagaimana di atas, maka kecepatan sesungguhnya dari modem **56 Kbps** yang digunakan dalam perancangan diasumsikan menjadi **30 Kbps**, dimana kecepatan ini merupakan kecepatan rata-rata dari kebanyakan modem yang dipakai saat ini.

Ukuran data yang sudah difilter dan dilakukan replikasi dengan proses *update* dua arah antara komputer pusat dengan komputer cabang sekitar **5000 KB (kilobyte)** maka waktu yang dibutuhkan untuk melakukan transfer data tersebut adalah:

$$\frac{5000 \times 8}{30} = 1333,33 \text{ detik} = 22,22 \text{ menit.}$$

Sehingga waktu yang dialokasikan untuk proses replikasi sebesar ± 30 menit

Koneksi yang dilakukan bisa diatur untuk beberapa kali dial ulang (*redial*) apabila koneksi pertama mengalami kegagalan serta bisa juga dilakukan pengaturan penjadwalan selisih waktu *redial* dengan di sebelumnya.

Jadwal detail dari koneksi masing-masing komputer cabang dapat dilihat pada Tabel 5.2. Karena penelitian ini dilaksanakan hanya pada tatann uji laboratorium dengan 1 komputer cabang maka untuk implementasi di dunia nyata alokasi waktu di jadwal ini disesuaikan dengan jumlah cabang yang ada dan kondisi di lapangan.

Tabel 5.2 Jadwal koneksi *dial-up* komputer cabang.

Komputer	Waktu Koneksi	Hari	Alokasi Waktu
Komputer 1	18.00 – 18.30 WIB	Senin – Sabtu	30 menit
Komputer 2	18.30 – 19.00 WIB	Senin – Sabtu	30 menit

5.1.2. *Otomatisasi Dial Up*

Untuk melakukan *dial up* dari computer-komputer cabang maka dibuat satu jenis koneksi tipe *dial up* dengan menggunakan *Make New Connection Wizard*. Pada pembuatan koneksi ini dimasukkan nomor ekstensi komputer pusat (*84, sebagai nomor tujuan *dial* sehingga pada saat computer-komputer cabang melakukan koneksi ke komputer pusat akan melalui PABX dengan nomor ekstensi tujuan yaitu 84. Koneksi yang telah dibentuk dapat dilihat pada menu *Network and Dial-Up Connection* seperti diperlihatkan dalam Gambar 5.4.



Gambar 5.4. Koneksi untuk melakukan *dial-up* pada komputer cabang.
Sumber: Implementasi

Koneksi dapat dilakukan dengan melakukan perintah *rasdial* pada *command* prompt. Perintah ini dapat digunakan untuk melakukan koneksi *dial up*, dimana argumen yang dibutuhkan adalah nama koneksi, *username* dan *password*.

```
rasdial KoneksiKePusat "pdam1" "pdam1"
```

Contoh untuk melakukan koneksi di komputer cabang adalah:

```
C:\> rasdial KoneksiKePusat "pdam1" "pdam1"
```

Untuk memutuskan koneksi maka diberikan argumen *disconnect* pada perintah *rasdial*.

```
rasdial KoneksiKePusat /disconnect
```

Contoh untuk melakukan pemutusan koneksi di komputer cabang adalah:

```
C:\> rasdial KoneksiKePusat /disconnect
```



Dengan menggunakan kedua perintah di atas dalam suatu *batch file* dan memanfaatkan fasilitas *Scheduled Task* pada Windows 2000 maka proses pembentukan koneksi secara terjadwal dapat dilakukan.

Karena dibutuhkan dua rutinitas utama, yaitu pembentukan koneksi dan pemutusan koneksi maka dibuat dua buah *file batch*, yaitu *koneksiOn.bat* dan *koneksiOff.bat* yang masing-masing berisikan perintah sebagaimana ditunjukkan pada Tabel 5.3 berikut:

Tabel 5.3. Perintah dalam *file batch* untuk pembuatan dan pemutusan koneksi .

Nama File	Fungsi	Isi
<i>koneksiOn.bat</i>	Pembentukan koneksi dial up	cmd /c netsh interface dialup "pda1" "pda1" /add
<i>koneksiOff.bat</i>	Pemutusan koneksi dial up	cmd /c netsh interface dialup /disconnect

Otomatisasi eksekusi kedua *file batch* tersebut dapat dilakukan dengan menambahkan *Scheduled Tusk* untuk masing-masing *file* tersebut, dengan jadwal yang telah ditentukan sebelumnya. Penambahan *Scheduled Task* dapat dilakukan dengan Wizard yang telah disediakan pada menu *System Tool*. Apabila telah berhasil melakukan penambahan *task* maka akan terlihat sebagaimana Gambar 5.5. berikut:



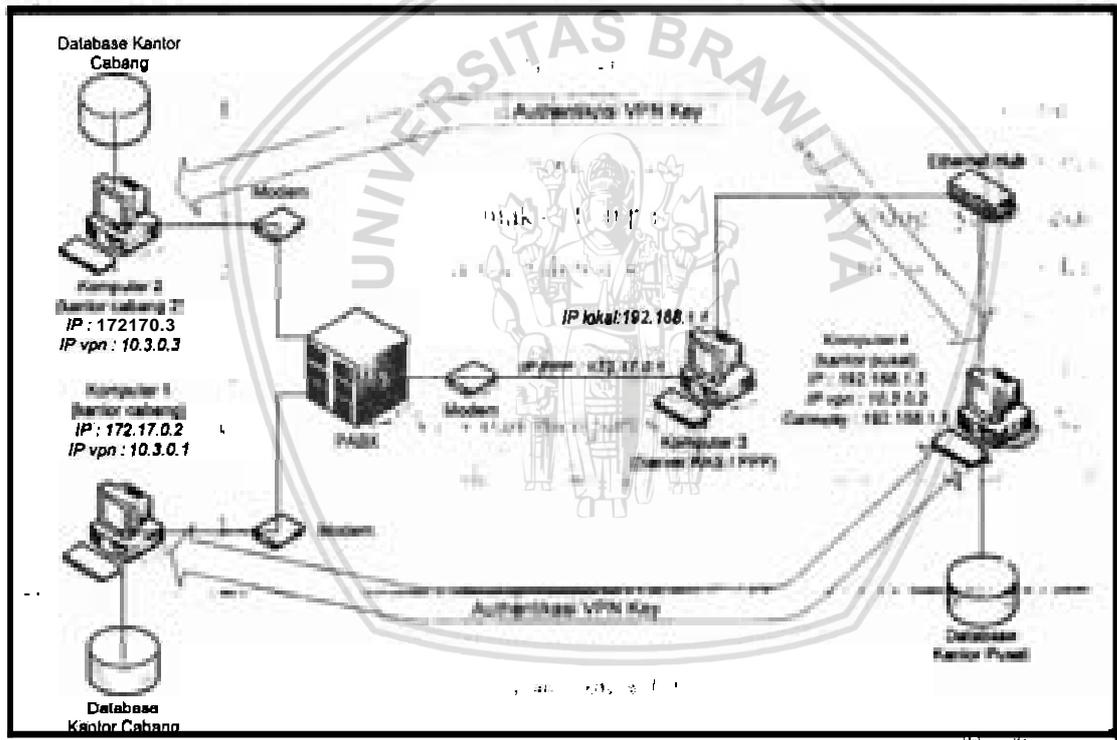
Gambar 5.5. Penjadwalan proses pembuatan dan pemutusan koneksi dial up.
Sumber: Implementasi

5.2. Konfigurasi *Virtual Private Network* (VPN)

Koneksi VPN dilakukan setelah komputer cabang telah terhubung dengan komputer server RAS dimana komputer server RAS berhubungan secara LAN dengan komputer pusat. Jadi setelah terhubung secara *dial up* komputer cabang dianggap telah menjadi anggota dalam jaringan komputer pusat dan memungkinkan untuk melakukan koneksi VPN antara komputer-komputer cabang dengan komputer pusat, seperti pada gambar 5.6, dimana sesuai dengan alokasi waktu yang sesuai dengan tabel 5.2., yaitu 30

menit waktu yang ada dipergunakan untuk pengkoneksian komputer cabang dengan kompufer server RAS yang terhubung secara dial up dan pengkoneksian antara komputer cabang dengan komputer pusat yang terkoneksi secara VPN.

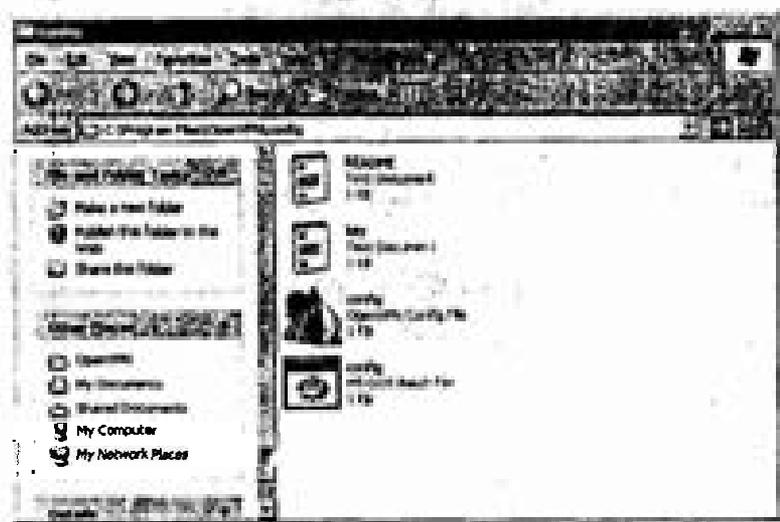
Sesuai dengan gambar 5.6, jadi setelah komputer-komputer cabang terhubung dengan komputer RAS yang disini berperan sebagai router dari komputer-komputer cabang dengan jaringan pusat, maka komputer-komputer cabang yang telah mendapatkan key (kunci kriptografi yang digunakan sebagai autentikasi identitas) dari komputer pusat dapat langsung melakukan koneksi VPN dengan komputer pusat secara bergantian (Langkah 1). karena koneksi ini merupakan koneksi 2 arah, maka komputer pusat juga melakukan koneksi VPN dan mengautentikasi key yang dimiliki, jika key yang dimiliki sama dan cocok dengan komputer cabang maka hubungan dapat dilakukan (langkah 2)



Gambar 5.6. Urutan pmses koneksi antara komputer cabang dengan pusat. Sumber: Implementasi

Cara pengkoneksian VPN dapat dilakukan dengan mensetting konfigurasi OpenVPN yang terdapat pada folder konfigurasi OpenVPN yang tampak seperti gambar 5.7. berikut :





Gambar 5.7. Gambar setting konfigurasi dan key.
Sumber: Implementasi

Pada bagian key dimana filenya berekstensi *.txt didalamnya terdapat kode-kode yang merupakan kunci yang saling dishare antara komputer pusat dengan komputer cabang supaya data yang melewati koneksi VPN ini terjaga dengan baik dan aman. Berikut adalah kode-kode yang terdapat dalam file key.txt tersebut.

```
#
# 2048 bit OpenVPN static key
#
-----BEGIN OpenVPN Static key V1-----
28b07926f7518b0f09d535c11c7e93e9
6f90fcd33290c80cafb4ed348aa20874
34ad1a3a1897f3f154oolff638f5e7ba
d0c901ace932aca067a6fa63b47f5273
07d1921e6defd70b30a8d2bf9c266ode
81bbabee400ce6ae3a8f176b1964fa44
6471b962f03f27e3e4703ab449e4f6cf
f809e5ad3a9effe0019d14c1ac09366d
c7631a4401ed41b6a889931441bf5434
42602631264d3dd6ea39a76d469a7bf
8a02b6c1923ade6f6732019293a4060f
12ef0ce42317e1275d5b3e25b70ca12f
efbb19a2c6306acc185f8b575f5e4c
d89e1f95b625fa31bbbedbe4ac82ccd67
1ae5d38393dcba6458b150303e6e5119
cbe52fbcd7af4a8691b231aba6a98b8f
-----END OpenVPN static key V1-----
```

Kode2 diatas adalah kode-kode acak yang dibuat oleh OpenVPN untuk memudahkan pengenalan komputer yang akan dikoneksikan secara VPN dimana kedua komputer yang akan dikoneksikan harus sudah memiliki kunci tersebut dan harus sama. Kemudian untuk setting konfigurasi pada file config.ovpn, cukup dengan mengklik dua kali icon config tersebut dan akan terbuka aplikasi notepad yang kemudian akan

diisikan beberapa perintah yang dapat menjalankan koneksi VPN tersebut. Pada komputer cabang 1 dan 2 dikonfigurasi pada notepadnya sebagai berikut :

```
remote 192.168.1.3
dev tap
ifconfig 10.3.0.2 255.255.255.0
secret key.txt
verb 3
mute 10
ping 10
```

dan pada komputer pusat pada field notepad akan diisikan konfigurasi sebagai berikut :

Konfigurasi untuk koneksi ke komputer cabang I

```
remote 172.17.0.2
dev tap
ifconfig 10.3.0.1 255.255.255.0
secret key.txt
verb 3
mute 10
ping 10
```

Konfigurasi untuk koneksi ke komputer cabang 2

```
remote 172.17.0.3
dev tap
ifconfig 10.3.0.3 255.255.255.0
secret key.txt
verb 3
mute 10
ping 10
```

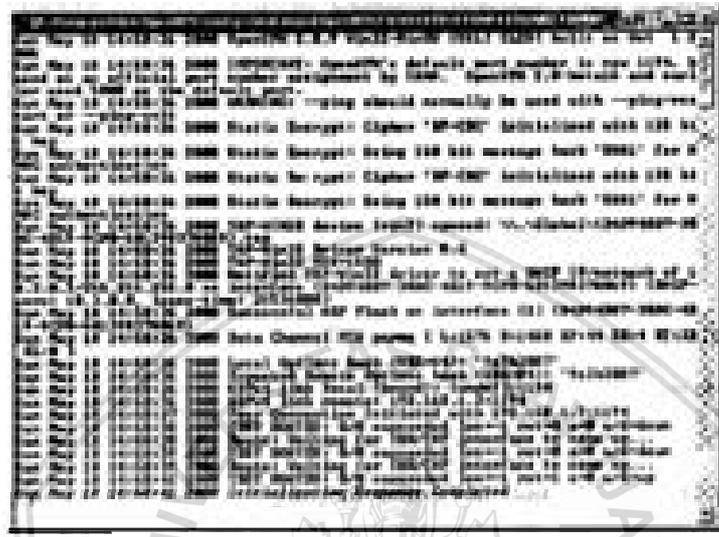
setelah pengetikan konfigurasi selesai maka dapat dilakukan start koneksi VPN oleh komputer cabang dan komputer pusat menggunakan konfigurasi yang telah dibuat tadi sesuai dengan gambar 5.8. berikut :



Gambar 5.8. Gambar start OpenVPN.
Sumber: Implementasi

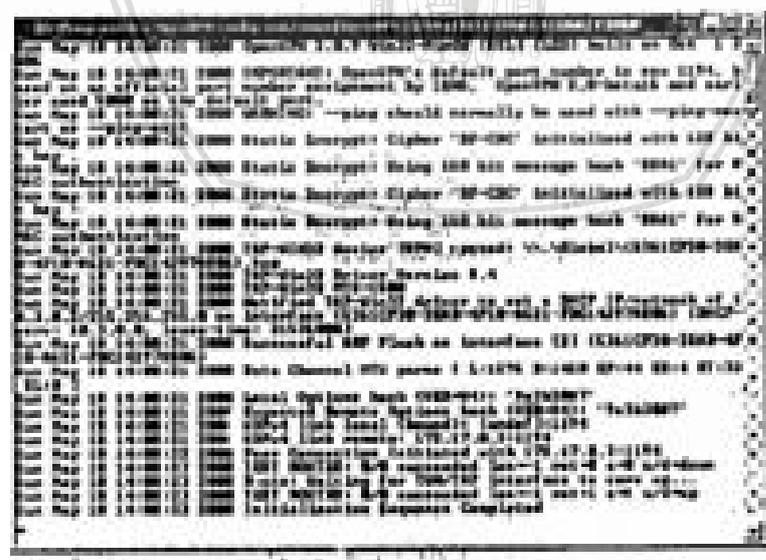


Setelah dilakukan komputer cabang melakukan *start* VPN maka secara otomatis akan keluar aplikasi *Command DOS prompt* yang menunjukkan *status window* dari koneksi VPN dan jika *window* tersebut ditutup maka otomatis koneksi VPN akan *terputus*. Pada gambar 5.9. berikut adalah window yang menunjukkan status koneksi yang berhasil pada komputer cabang :



Gambar 5.9. Gambar contoh *window status* koneksi VPN pada komputer cabang.
Sumber: Implementasi

Kemudian secara *bersamaan* pada komputer *pusat* juga melakukan *start* VPN dan akan keluar status window *sesuai* pada gambar 5.10. berikut :



Gambar 5.10. Gambar contoh *window status* koneksi VPN pa& komputer pusat.
Sumber: Implementasi



Pada masing-masing **status** window yang **ditunjukkan** pada gambar 5.9, dan gambar 5.10, pada akhir status **terdapat kalimat *Initializaion Sequence Completed*** yang menggambarkan bahwa koneksi **VPN telah berhasil dilakukan dan telah berjalan**. Masalah **alokasi** waktu yang diperlukan untuk mencapai **kata completed** tidak membutuhkan **waktu** yang lama. **Mungkin Cuma 1 menit. Maka** oleh **karena** itu tidak perlu mengeset **ulang** jadwal koneksi **dial up**.

5.2.1 Otomatisasi Koneksi VPN

Setelah pembahasan konfigurasi OpenVPN maka saat ini perlu adanya otomatisasi koneksi VPN **agar** replikasi berjalan secara otomatis **tanpa** perlu campur tangan operator. Koneksi dapat dilakukan dengan melakukan perintah **openvpn** pada **command prompt**. Perintah ini dapat digunakan untuk melakukan koneksi VPN, dimana argumen yang dibutuhkan adalah **perintah-perintah** yang dimiliki **OpenVPN**. Berikut adalah contoh perintah-perintah yang terdapat pada konfigurasi komputer **pusat**,

```
openvpn --remote 172.17.0.2 --dev tap --ifconfig 10.3.0.2
255.255.255.0 --secret key.txt --verb 3 --mute 10 --ping 10
```

Contoh untuk melakukan koneksi di komputer **pusat** adalah:

```
C:\> openvpn --remote 172.17.0.2 --dev tap --ifconfig 10.3.0.2
255.255.255.0 --secret key.txt --verb 3 --mute 10 --ping 10
```

Contoh untuk melakukan koneksi di komputer cabang adalah:

```
C:\> openvpn --remote 192.168.1.3 --dev tap --ifconfig 10.3.0.1
255.255.255.0 --secret key.txt --verb 3 --mute 10 --ping 10
```

Dengan **menggunakan** kedua perintah di **atas** dalam suatu **batch file** dan **memanfaatkan** fasilitas **Scheduled Task** pada Windows 2000 maka proses pembentukan koneksi **secara** terjadwal dapat **dilakukan**.

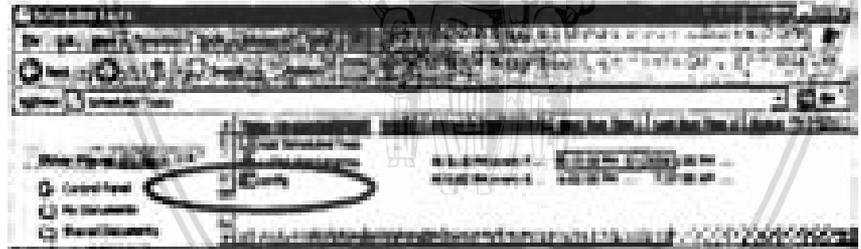
Karena dibutuhkan dua **rutinitas utama** pada komputer cabang dan komputer **pusat**, yaitu pembentukan koneksi dan **pemutusan** koneksi maka dibuat **satu** buah **file batch** pada masing-masing komputer, yaitu **config.bat** yang **berisikan** perintah sebagaimana **ditunjukkan** pada Maka dibuatlah **file batch** sesuai dengan **tabel 5.4** berikut :



Tabel 5.4. Perintah dalam file batch untuk pembuatan dan pemutusan koneksi VPN.

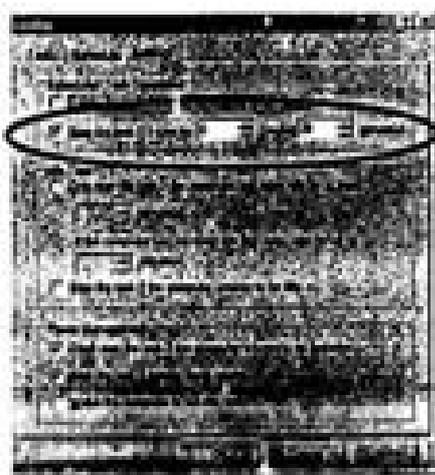
Nama File	Fungsi	Perintah
unit1.bat	Pembentukan koneksi VPN pada komputer pusat	<pre> openvpn --remote 172.17.0.2 --dev tap --ifconfig 10.3.0.2 255.255.255.0 -- secret key.txt --verb 3 --mute 10 -- ping 10 </pre>
config.bat	Pembentukan koneksi VPN pada komputer cabang	<pre> openvpn --remote 192.168.1.3 --dev tap --ifconfig 10.3.0.1 255.255.255.0 -- secret key.txt --verb 3 --mute 10 -- ping 10 </pre>

Otomatisasi eksekusi file batch tersebut dapat dilakukan dengan menambahkan Scheduled Task untuk file tersebut pada komputer pusat dan komputer cabang, dengan jadwal yang telah ditentukan dimana koneksi akan dimulai 2 menit setelah komputer cabang telah terkoneksi secara dial up dengan komputer server RAS. Penambahan Scheduled Task dapat dilakukan dengan Wizard yang telah disediakan pada menu System Tool. Apabila telah berhasil melakukan penambahan task maka akan terlihat sebagaimana Gambar 5.11, berikut:



Gambar 5.11. Penjadwalan proses pembuatan dan pemutusan koneksi VPN.
Sumber: Implementasi

Untuk pemutusan koneksi VPN, pada file .properties config.bat diset, untuk dinonaktifkan setelah 28 menit sesuai dengan jumlah alokasi waktu dari koneksi dial up. Berikut pengesetan waktu untuk menonaktifkan file batch config.bat yang ditunjukkan pada gambar 5, 2;

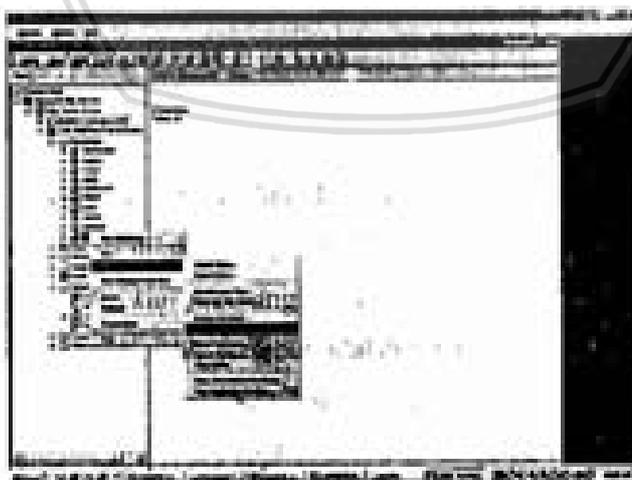


Gambar 5.12. Pengesetan waktu pemutusan koneksi VPN.
Sumber: Implementasi

5.3. Konfigurasi Database

5.3.1. Restore Database

Karena proses replikasi basis data dilakukan melalui fitur di SQL Server 2000, maka terlebih dahulu harus dilakukan *restore database*. Proses *restore* dilakukan untuk menghubungkan antara *backup file* yang sudah ada di PDAM dengan *database* yang dibuat di SQL Server. *Backupfile* tersebut adalah *backup* dari data-data *transaksi* yang sudah disimpan di PDAM Pusat. Sedangkan *database* yang dibuat di SQL Server tersebut nantinya dilakukan proses replikasi. Tahapan proses *Restore* dilakukan pada SQL Server 2000 pada *SQL Server Enterprise Manager* dengan memilih *all Tasks* | Restore Database seperti terlihat dalam tampilan di dalam Gambar 5.13.



Gambar 5.13. SQL Server Enterprise Manager
Sumber: Implementasi

Selanjutnya dilakukan pembuatan database pada SQL Server dengan nama pdam pusat dan pemilihan backup device atau file name yaitu file cadangan dimana file tersebut adalah backup file dari semua data di basis data yang ada pada PDAM kabupaten Malang. Seperti yang diperlihatkan dalam Gambar 5.14.



a) Memilih Restore Destination b) Lokasi Backup File
 Sumber: Implementasi

Setelah dilakukan *restore database* maka *database* pdam pusat akan terisi dengan data-data transaksi yang ada pada backup file untuk selanjutnya dilakukan proses replikasi pada *database* tersebut. Setelah *database* pdam pusat sudah terisi dengan data lengkap, transaksi maupun pembuatan kwitansi dan laporan-laporan pada PDAM pusat maupun cabang-cabang bisa dilakukan melalui tampilan aplikasi Microsoft Access. Di PDAM sudah mempunyai file Access berupa file .adp (Access Data Project) dimana untuk bisa membuka file tersebut dilakukan koneksi terlebih dahulu antara file tersebut dengan database pdam pusat di SQL Server. Koneksi dilakukan dengan menjalankan file tersebut dilanjutkan dengan memilih File | connection, kemudian memasukkan nama server dan nama database yang datanya akan dibuka dengan menggunakan tampilan di Microsoft Access tersebut.

5.3.2. Implementasi Replikasi

5.3.2.1. Skema Replikasi

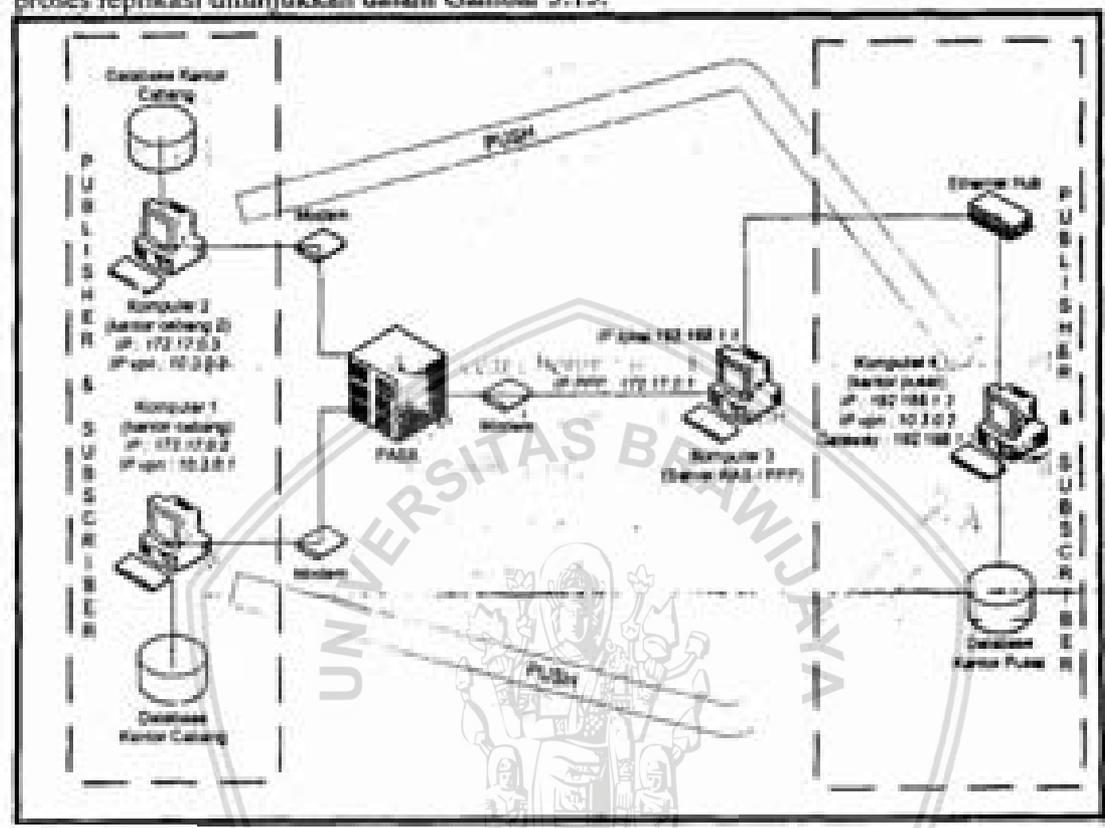
Jenis replikasi yang digunakan pada perealisasi sistem ini adalah tipe replikasi merge yang memungkinkan update data secara dua arah, baik dari kantor pusat ke cabang-cabang maupun dari cabang-cabang ke kantor pusat. Kantor pusat dibuat

publikasi untuk selanjutnya publikasi-publikasi tersebut di-push ke cabang, langkah pertama dalam pengiriman data adalah publikasi-publikasi tersebut dibuat filter yaitu kode-kec sesuai dengan kecamatan masing-masing yaitu cabang adalah dengan kode-kec sama dengan 'A'. Setelah proses push pertama selesai maka dilakukan penghapusan push untuk selanjutnya filter pada publikasi yaitu pada Tabel Pemakaian dirubah menjadi gabungan kode-kec, bulan dan tahun yang sedang berjalan sehingga nantinya data pemakaian yang dikirim adalah data yang sesuai dengan bulan dan tahun yang sedang berjalan. Perintah filter yang digunakan adalah Kode_kec='A' untuk Tabel Pelanggan sedangkan untuk Tabel Pemakaian pada publikasi sebelum dilakukan perubahan adalah Pelanggan.id_pelanggan=Pemakaian.id_pelanggan. Selanjutnya untuk Tabel Pemakaian dilakukan perubahan filter yaitu dengan penambahan parameter filter dengan perintah (Pelanggan.id_pelanggan=Pemakaian.id_pelanggan) and bulan-month(getdate()) and tahun=year(getdate()). Setelah dilakukan perubahan maka publikasi-publikasi tersebut di-push kembali ke masing-masing cabang sesuai dengan kode-kec.

Proses replikasi ini diset waktu berlangganan/proses publikasi pada saat komputer di pusat sudah terhubung (connect) dengan komputer di cabang. Sesuai dengan scheduling dial up dan VPN yang sudah ditentukan sebelumnya komputer kantor cabang terhubung dengan komputer kantor pusat dan pada waktu itu juga proses update data dijadwalkan. Setiap cabang akan mempunyai waktu connect yang berbeda sehingga bentrokan replikasi antar komputer cabang tidak akan terjadi.

Komputer di kantor cabang dijadikan sebagai subscriber untuk menerima kiriman data terbaru dari pusat, di komputer cabang data bisa dilakukan update dan secara otomatis akan berpengaruh pada komputer pusat karena proses replikasi bertipe merge. Sedangkan di kantor pusat proses replikasi dilakukan dengan membuat publication untuk cabang-cabang, dimana publication tersebut akan dihubungkan dengan database yang ada di cabang-cabang. Setiap publication yang terhubung dengan komputer pusat berjenis push dan berisi data masing-masing cabang, cabang akan dibuatkan replika dari database pusat yang berisi data pelanggan pada cabang dalam hal ini dibuat data dengan Kode_kec=A, yaitu kecamatan Ngajum. Dengan proses replikasi maka data pada kantor cabang akan berbeda tiap cabangnya, sedangkan kantor pusat akan berisi data lengkap penggabungan dari data-data yang dikirimkan dari cabang.

Proses replikasi **dikerjakan secara berurutan** sebagaimana proses koneksi VPN juga dilakukan **secara berurutan**. Akan tetapi antara waktu koneksi VPN dengan waktu replikasi diberi selisih waktu 5 menit untuk memberikan waktu yang cukup untuk melakukan koneksi ulang apabila proses koneksi pertama mengalami kegagalan. Urutan proses replikasi ditunjukkan dalam Gambar 5.15.



Gambar 5.15. Proses replikasi antara komputer cabang
Sumber: Implementasi

Replikasi dilakukan oleh **Komputer cabang pada waktu yang telah** ditentukan pada Tabel 5.5. Komputer pusat akan melakukan push terhadap publikasi yang dimilikinya, dalam hal ini data di pusat ke komputer cabang. Alokasi waktu ini dapat diubah-ubah sesuai besarnya data yang ada di komputer cabang. Seiring dengan penambahan waktu maka data akan semakin besar dan waktu yang dibutuhkan untuk melakukan replikasi akan semakin lama juga.

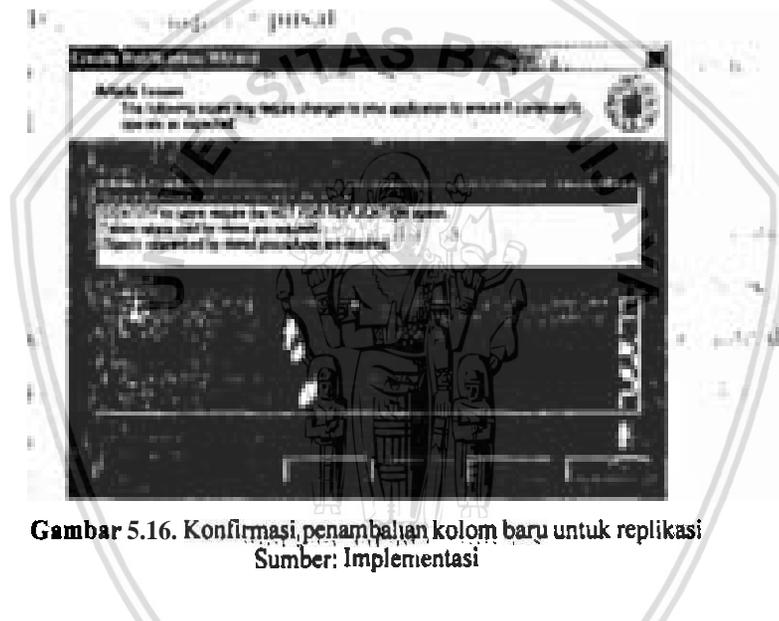
Tabel 5.5. Jadwal push replikasi komputer cabang.

Komputer	Waktu Replikasi	Hari	Alokasi Waktu
Komputer 1	18.05 – 18.25 WIB	Senin – Sabtu	20 menit
Komputer 2	18.35 – 18.55 WIB	Senin – Sabtu	20 menit



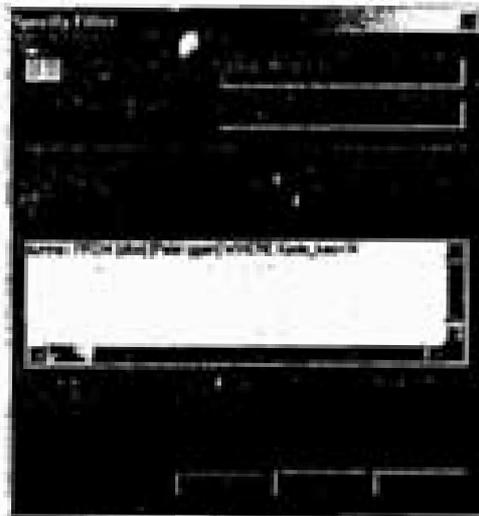
5.3.2.2. Konfigurasi komputer pusat

Komputer pusat dikonfigurasi sebagai *distributor* dan publisher untuk mengirimkan data publikasi ke cabang-cabang, karena replikasi yang digunakan merupakan jenis merge dan *di-push* ke komputer cabang. Sebagaimana ditunjukkan dalam Gambar 4.8, Tabel Pelanggan saling terhubung dengan tabel yang lainnya dengan suatu *foreign key*. Masing-masing tabel juga telah memiliki primary key tersendiri, akan tetapi pada Tabel-tabel tersebut tidak terdapat kolom *uniqueidentifier* yang diperlukan dalam proses replikasi. Oleh karena itu keseluruhan struktur tabel tersebut akan berubah selama proses replikasi. SQL Server akan menambahkan secara otomatis sebuah kolom baru yang merupakan *uniqueidentifier*. Dalam konfigurasi komputer cabang keputusan untuk penambahan kolom ini ditunjukkan dalam Gambar 5.16.



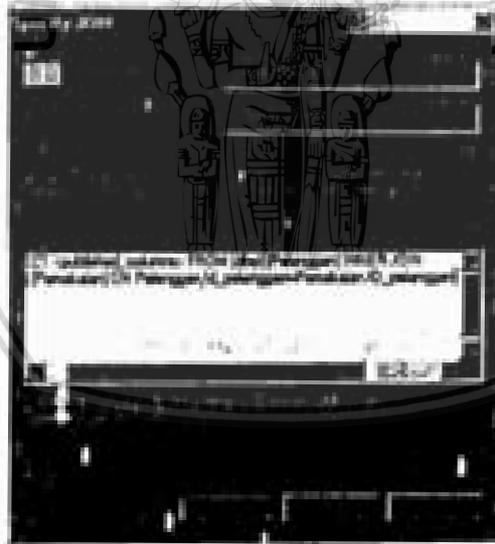
Gambar 5.16. Konfirmasi penambahan kolom baru untuk replikasi
Sumber: Implementasi

Konfigurasi selanjutnya adalah penentuan filter pada Tabel Pelanggan dan Tabel Pemakaian, dimana kedua tabel tersebut merupakan tabel utama dan saling berhubungan. Masing-masing cabang hanya bertanggung jawab terhadap data untuk cabangnya sendiri. Hal ini dapat dilakukan dengan melakukan filter per baris pada Tabel Pelanggan dengan berdasarkan kolom *Kode_kec* (kode kecamatan). Penentuan filter pada Tabel Pelanggan dapat dilihat dalam Gambar 5.17.

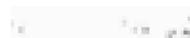


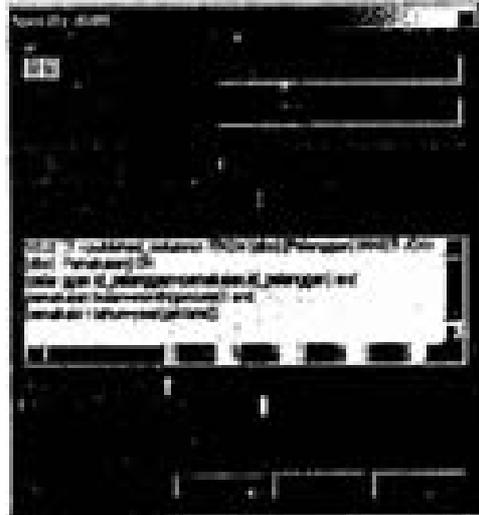
Gambar 5.17. Konfigurasi pengaturan filter pada Tabel Pelanggan
 Sumber: Implementasi

Karena Tabel Pelanggan dan Tabel Pemakaian saling berhubungan maka untuk melakukan filter terhadap Tabel Pemakaian dilakukan dengan menggunakan *join* antara kedua tabel tersebut, dimana kolom yang digunakan untuk *join* adalah *Id_pelanggan* sebagaimana ditunjukkan dalam Gambar 5.18. dan Gambar 5.19.



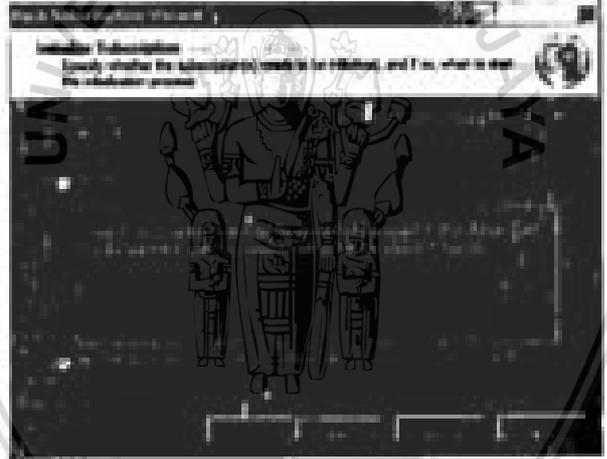
Gambar 5.18. Konfigurasi pengaturan filter *join* pada Tabel Pemakaian sebelum ada perubahan
 Sumber: Implementasi





Gambar 5.19. Konfigurasi pengaturan filter *join* pada Tabel Pemakaian yang berubah
 Sumber: Implementasi

Untuk proses pengiriman publikasi setelah dilakukan perubahan filter pada Tabel Pemakaian berbeda dengan proses pengiriman publikasi yang pertama karena hanya data saja yang dikirim seperti diperlihatkan dalam Gambar 5.20.

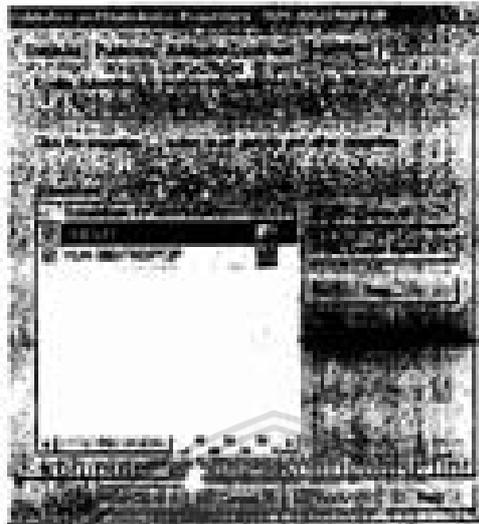


Gambar.5.20 *Initialize subscription* pada publikasi kedua
 Sumber: Implementasi

5.3.2.3 Konfigurasi Komputer Cabang

Pada komputer cabang dikonfigurasi sebagai subscriber untuk menerima publikasi dari komputer pusat. Sebelumnya harus dibuat terlebih dahulu database di SQL Server sebagai penerima publikasi dari komputer pusat, hal ini perlu dibuat sebelum dilakukan konfigurasi replikasi karena pada waktu konfigurasi push publikasi pada wizard akan melalui tahapan memilih database yang akan menerima publikasi. Pada database ini setelah proses replikasi berlangsung akan terisi oleh data yang sudah difilter, yang di-push dari komputer pusat dan pada database ini bisa dilakukan update

data pada masing-masing komputer cabang. Konfigurasi untuk menjadikan komputer cabang sebagai *subscriber* dapat dilihat dalam Gambar 5.21. dan Gambar 5.22.



Gambar 5.21. Konfigurasi komputer cabang sebagai *subscriber*

Sumber: Implementasi



Gambar 5.22. Konfigurasi *property subscriber*

Sumber: Implementasi

5.3.2.4 Registrasi *Remote SQL Server*

Untuk dapat melakukan proses replikasi antara *database SQL Server* yang berada dalam komputer lain, maka perlu dilakukan registrasi *database server* yang dituju di *database server* lokal. Kedua *database* di komputer cabang dan komputer pusat harus dikonfigurasi untuk dapat saling berhubungan dengan menggunakan nomor *port* yang sama.

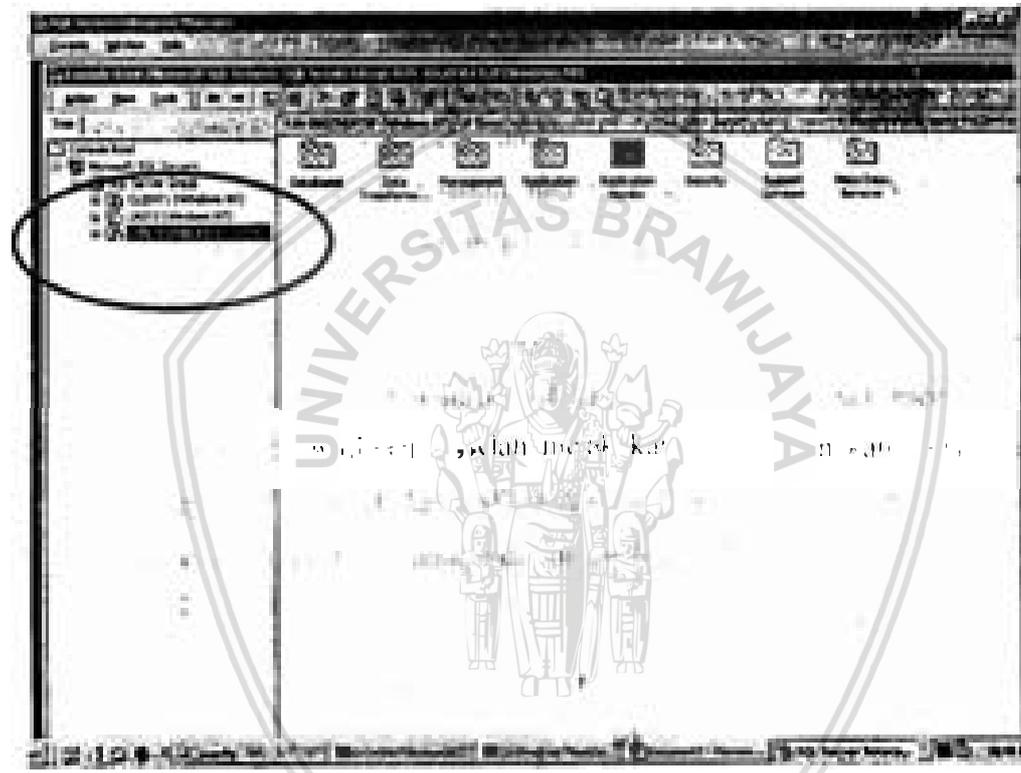
Sumber: Implementasi

Sumber: Implementasi

Sumber: Implementasi

Dalam perancangan ini koneksi antara *database* SQL Server dilakukan oleh OpenVPN dimana pengesetan port sudah merupakan default dari OpenVPN tersebut, yaitu port 139 dan IP yang digunakan pada konfigurasi command VPN yang telah dibahas diatas

Setelah alias dibuat maka langkah berikutnya adalah menjalankan *wizard* untuk registrasi SQL Server yang diinginkan. *Wizard* dilakukan dengan menggunakan informasi alias yang telah dibuat.. Setelah melakukan langkah-langkah pada *wizard* maka apabila registrasi berhasil akan terlihat ikon SQL Server yang baru di *tree* SQL Sewer Group, sebagaimana ditunjukkan dalam gambar 5.23.



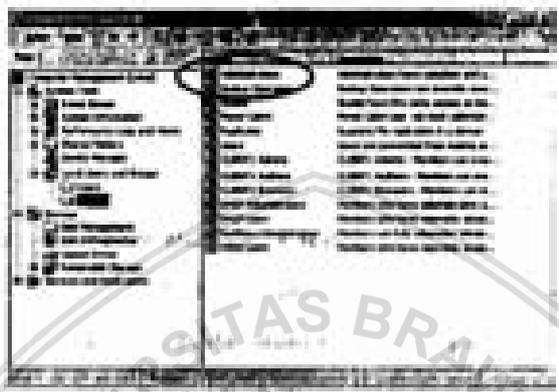
Gambar 5.23. Tampilan setelah registrasi SQL Server berhasil
Sumber: Implementasi

Dengan cara yang sama komputer cabang **dikonfigurasi** agar terhubung dengan *sewer* di komputer **pusat** dengan **cara registrasi *database sewer*** yang langkah-langkahnya sama dengan **konfigurasi** di komputer **pusat**. Akan tetapi di komputer cabang pada Enterprise Manager tidak **perlu** dibuat registrasi baru untuk memunculkan *server* di *tree* SQL Server Group sehingga *sewer* dan data pada komputer **pusat** tidak terlihat di komputer cabang. Hal ini dilakukan dengan **alasan keamanan** agar data pada komputer **pusat** tidak dirubah **melaui** komputer cabang.



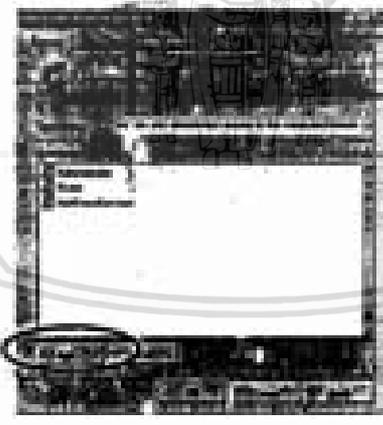
5.4. Konfigurasi Windows Authentication User

Pada saat replikasi basis data dilakukan **pasti** ada kalanya **berhasil** atau **gagal**. Dan biasanya jika terdapat konflik pada waktu replikasi **perlu adanya penyesetan** pada *windows security*. Penyesetan itu dilakukan pada computer cabang dan pusat yang secara langsung terkoneksi secara VPN. Penyesetan dilakukan pada perangkat *Microsoft windows 2000 advance server* yaitu *computer management* yang digambarkan pada gambar 5.24.



Gambar 5.24. Tampilan window *Computer Management*
Sumber: Implementasi

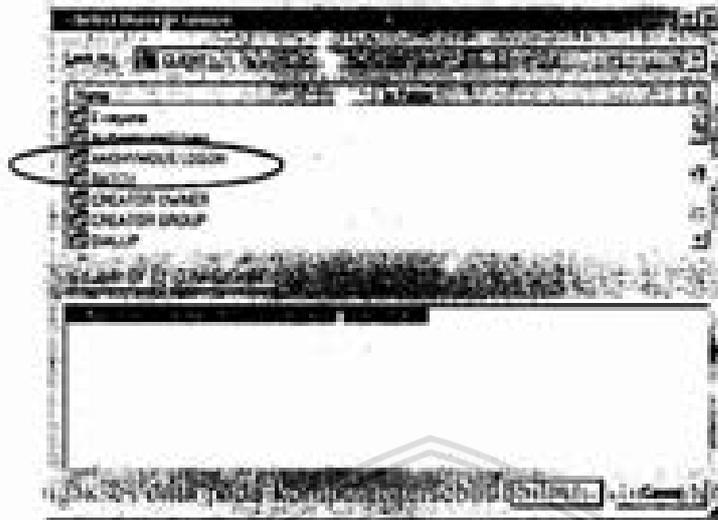
Penyesetan dilakukan pada *folder groups* yang terdapat pada System Tools | Local Users and Groups. Kemudian masuk pada bagian Administrators dan akan keluar tampilan window yang tergambar pada gambar 5.25



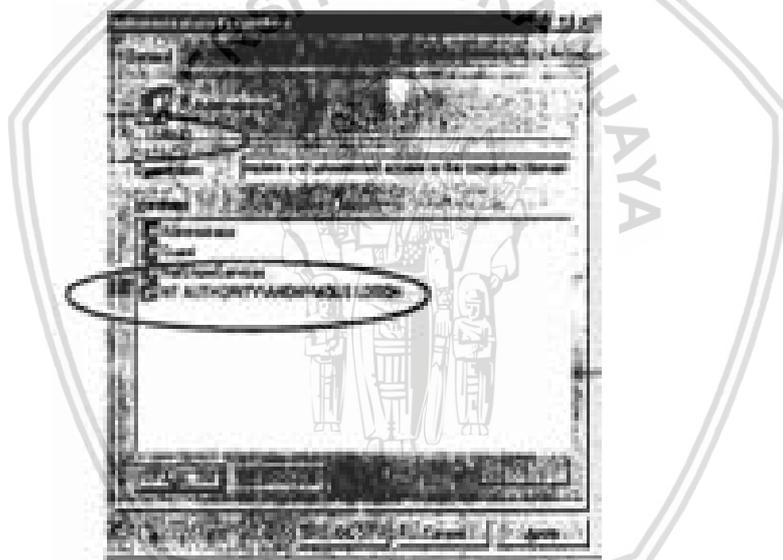
Gambar 5.25. Tampilan window *Administrators Properties*
Sumber: Implementasi

Setelah masuk ke dalam *Administrators Properties*, dilakukan penambahan anggota dalam grup *Administrators* dengan meng-klik tombol *add*. Setelah masuk didalam window penambahan anggota *Administrators*, maka pilihlah *ANONYMOUS LOGON*, dimana dalam hal ini *Administrators* mengizinkan segala macam user yang tidak dikenal

untuk dapat mengakses data pada komputer tersebut baik itu komputer cabang ataupun komputer pusat, sebagaimana ditunjukkan pada gambar 5.26 dan 5.27.



Cambar5.26. Tampilan window anggota-anggota grup yang dapat dipilih
Sumber: Implementasi

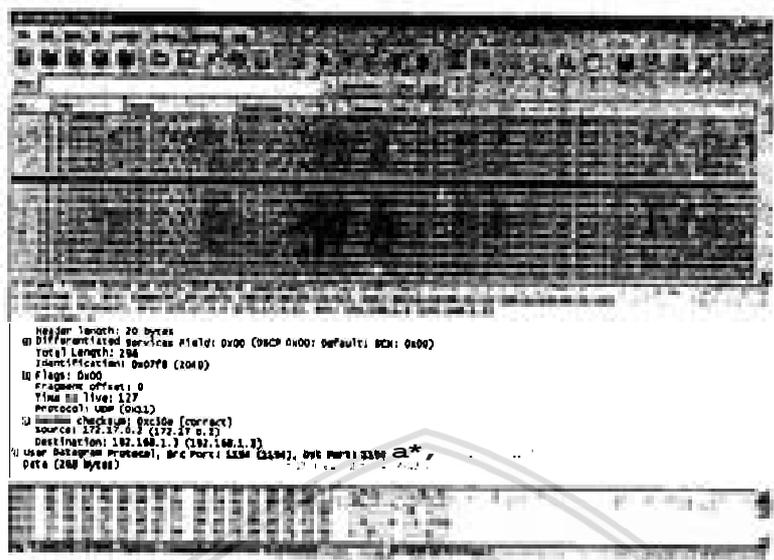


Gambar 5.27. Tampilan window bahwa anggota baru telah dipilih
Sumber: Implementasi

5.5 Prasyarat Parameter Keberhasilan VPN

Dalam percobaan ini parameter penggunaan VPN adalah bagaimana data yang dikirim dari cabang menuju pusat dapat dengan aman (secure) tertransfer. Maka oleh karena itu digunakanlah VPN sebagai media pengkoneksi antara komputer cabang dan komputer pusat. Parameter secure dalam percobaan ini adalah paket-paket data yang melalui VPN tidak dapat diketahui oleh komputer - komputer yang dilalui oleh koneksi VPN antara kedua komputer tersebut. Berikut capture data yang dilihat dari komputer

RAS yang dilewati koneksi VPN komputer cabang dan komputer pusat ditunjukkan oleh gambar 5.28.



Gambar 5.28. Capture packet data pada RAS saat Replikasi antara IP Komputer Database pusat oleh komputer cabang
Sumber: Implementasi

Parameter lain yang digunakan dalam percobaan ini adalah kedua komputer antara komputer cabang dan komputer pusat dapat terkoneksi databasanya dan melakukan replikasi database dengan sempurna. Replikasi yang berhasil dapat dilihat pada gambar 5.29 berikut,



Gambar 5.29. Replikasi yang berjalan antara komputer cabang dan komputer pusat.
Sumber: Perancangan



BAB VI

PENGUJIAN DAN ANALISIS

Dalam bab ini dibahas **pengujian dan analisis** sistem pada **masing-masing blok** yang sudah **dirancang** untuk **mempermudah dalam menganalisa hasil perancangan dan** pengujian yang **dilakukan, bab ini juga dibahas** pengujian sistem **secara keseluruhan**. Setelah dilakukan pengujian sistem **secara keseluruhan** juga dilakukan pengujian koneksi aplikasi **Microsoft Access** setelah terjadi proses **replikasi** baik di **komputer pusat** maupun di komputer **cabang** untuk mengetahui apakah aplikasi **tersebut** bisa digunakan untuk melakukan transaksi dan pembuatan **kwitansi** maupun **laporan**. Hasil pengujian ini kemudian dianalisa dengan membandingkannya terhadap **perancangan**. Pengujian dilakukan terhadap **blok-blok sistem** yang meliputi:

6.1. Pengujian Per Blok

6.1.1. Pengujian **Restore Database**

Pengujian ini dilakukan **dengan** menghubungkan data transaksi yang ada pada PDAM yang **berupa backup file dengan database** di SQL Server. Backup file **disini** tidak disebutkan prosesnya karena, **file backup ini langsung didapat berupa softcopy file .adp (Access Data Project)** dari PDAM. Pengujian ini dilakukan untuk **mengetahui** apakah **database** pada **SQL Server** dapat terisi dengan data yang **terdapat** pada file tersebut.

a. Tujuan

- Mengetahui apakah **database** pada **SQL Server** akan terisi dengan data yang **terdapat** pada **backup file**.

b. Spesifikasi dan Konfigurasi Komputer

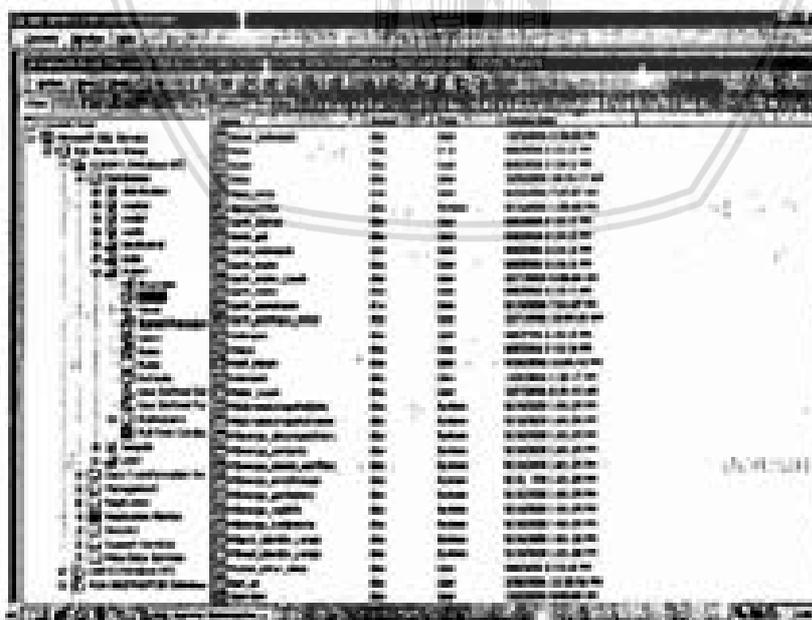
- Empat buah komputer, komputer pertama dan kedua dijadikan sebagai komputer cabang dengan IP **172.17.0.2** dan **172.17.0.3** . komputer ketiga dijadikan komputer **server RAS** dengan IP PPP **172.17.0.1** dan IP Lokal **192.168.1.1** sedangkan komputer keempat dijadikan komputer **pusat** dengan IP **192.168.1.4**, dengan **gateway 192.168.1.1**.
- Komputer **pusat: Prosesor Intel Pentium 4 • 2.26 GHz**, memori **512 MB**.
- Komputer server **RAS : Processor Intel Centrino – 1.8 GHz**, memori **512 MB**.

MB



- Komputer cabang: **Prosesor Intel Pentium Dual Core - @1.6 GHz, memori 512 MB.**
 - **Sistem Operasi Microsoft Windows 2000 *Server* dan Windows XP.**
- c. **Software Aplikasi**
- *Server database SQL Server 2000.*
- d. **Prosedur Pengujian**
- Mengaktifkan *server* pada *SQL Server*.
 - Membuat *database* pdam pusat pada *SQL Server* yang akan dilakukan *restore database*.
 - Membuka *database* pdam pusat tersebut yang terlihat pada *console root* di *SQL Server Enterprise Manager*.
 - Apabila *restore* berhasil maka *database* akan menampilkan **tabel-tabel**, **view-view** maupun *procedure*, dan pada **tabel-tabel** terdapat **data**.
- e. **Hasil Pengujian .**

Diperlihatkan dalam Gambar 6.1. bahwa *database* pdam pusat sudah terisi dengan beberapa tabel, yang merupakan hasil dari proses *restore database* dimana proses *restore database* berhasil menghubungkan antara *SQL Server* dengan *backup file* yang berisi data PDAM.



Gambar 6.1. Macam-macam tabel pada *database* pdam pusat
Sumber: Pengujian

Gambar 6.2. menunjukkan data pada Tabel Pelanggan yang ada pada *backup file* dan bisa dilihat dengan menggunakan *SQL Server*. Begitu juga apabila tabel-tabel hasil *restore* dibuka maka terlihat data PDAM yang sudah diisikan. Baik Tabel Bulan, Tabel Desa, Tabel Golongan, Tabel Kecamatan, dan tabel-tabel lain hasil *restore*.

id_pelanggan	nama_pelanggan	alamat	no_telp	jenis_kelamin	status	tanggal_lahir	tanggal_daftar	tanggal_bayar	jumlah_bayar	jumlah_piutang	jumlah_sisa	jumlah_denda	jumlah_sisa_denda	jumlah_sisa_piutang	jumlah_sisa_denda_piutang
1
2

Gambar 6.2. Data pada Tabel Pelanggan
Sumber: Pengujian

f. **Kesimpulan**

Proses *restore database* yang dilakukan pada *SQL Server* akan didapatkan koneksi antara *database* yang dibuat di *SQL Server* terisikan data yang ada pada *backup file*. Sehingga data-data tersebut yang nantinya akan dilakukan *replikasi* pada *SQL Server*.

6.1.2. **Pengujian Koneksi Antar Komputer**

Pengujian ini dilakukan untuk mengetahui koneksi antar komputer seluruhnya, baik itu koneksi *Dial Up* dari komputer cabang menuju komputer server **RAS** dan koneksi *VPN* dari komputer cabang menuju komputer pusat yang sebelumnya dilakukan *setting* terlebih dahulu di Sistem Operasi untuk memungkinkannya adanya koneksi antar komputer dengan menggunakan *PABX* dan *Ethernet Hub* sebelum dan sesudah dilakukan koneksi *server database* antar komputer.



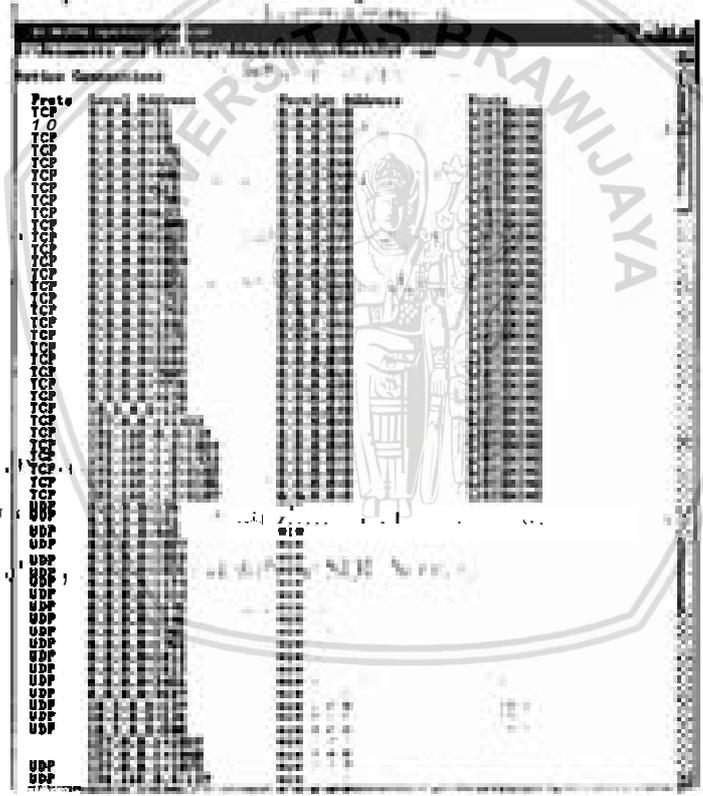
- a. Tujuan
- Mengetahui proses koneksi antar komputer dengan pengaturan koneksi di Sistem Operasi Windows 2000 Server dan SQL Server 2000.
- b. Spesifikasi dan Konfigurasi Komputer
- Empat buah komputer, komputer pertama dan kedua dijadikan sebagai komputer cabang dengan IP 172.17.0.2 dan 172.17.0.3 . komputer ketiga dijadikan komputer server RAS dengan IP PPP 172.17.0.1 dan IP Lokal 192.168.1.1 sedangkan komputer keempat dijadikan komputer pusat dengan IP 192.168.1.4 dengan gateway 192.168.1.1.
 - Komputer pusat: Prosesor Intel Pentium 4 - 2,26 GHz, memori 512 MB.
 - Komputer server RAS : Processor Intel Centrino - 1.8 GHz, memori 512 MB
 - Komputer cabang: Prosesor Intel Pentium Dual Core - @1.6 GHz, memori 512 MB.
 - Sistem Operasi Microsoft Windows 2000 Server dan Windows XP.
- c. Software Aplikasi
- Server database SQL Server 2000.
 - Open VPN 2.0.9.
 - Control Panel.
- d. Prosedur Pengujian
- Menjalankan Command Prompt dari Start | Run... | Open:cmd.exe |.
 - Menampilkan koneksi yang sedang aktif pada komputer cabang sebelum melakukan koneksi pada server database SQL Server dengan memberikan perintah:

```
C:\>netstat -an
```
 - Melakukan setting untuk koneksi pada server database SQL Server.
 - Melakukan koneksi Dial Up dari komputer cabang ke komputer Server RAS melalui modem dan PABX.
 - Menjalankan command-command seperti C:\>ping [IP] dan C:\>ipconfig /all untuk melihat sudah berjalan atau tidak koneksi yang telah dibangun setelah Dial Up dan sebelum koneksi VPN.

- Melakukan koneksi privat / VPN menggunakan program Open VPN 2.0.9 dari komputer cabang ke komputer pusat.
- Menjalankan command-command seperti C:\>ping (IP) dan C:\>ipconfig /all untuk melihat sudah berjalan atau tidak koneksi yang telah dibangun setelah koneksi VPN.
- Menampilkan koneksi yang sedang aktif pada komputer setelah melakukan koneksi pada database SQL Server dengan memberikan perintah:
C:\> netstat -an

e. Hasil Pengujian

Gambar 6.3. memperlihatkan koneksi yang aktif pada komputer cabang sebelum melakukan koneksi pada server database SQL Server.



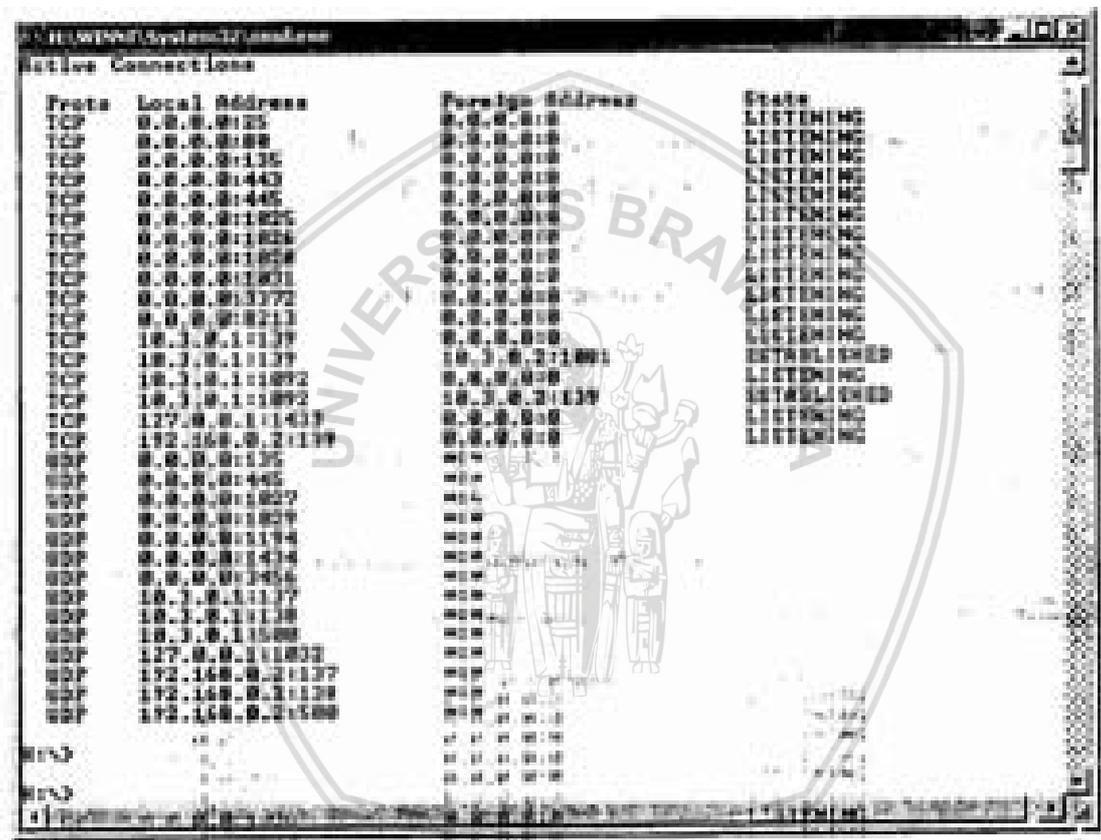
Gambar 6.3. Daftar koneksi komputer pusat sebelum melakukan koneksi dengan database SQL Server
Sumber: Pengujian

Gambar 6.3. menunjukkan hasil dari penggunaan perintah 'netstat -an', perintah tersebut digunakan untuk menampilkan semua jenis koneksi yang terdapat pada komputer pusat dan port yang sedang melakukan listening dengan memperlihatkan alamat IP komputer dan port yang digunakan dalam bentuk numerik. Setelah dilakukan



koneksi **tanpa** menyeting IP tujuan dari **SQL** server terlebih dahulu, karena telah terkoneksi VPN dan kedua IP VPN tersebut'seakan-akan telah **dihubungkan** oleh **sebuah** tunnel dan **secara** otomatis kedua IP VPN tersebut sudah mengetahui **satu sama lainnya**. User yang digunakan **adalah pdam dengan** password **pdam**. Koneksi antara komputer pusat dan komputer cabang dengan melakukan proses koneksi **server database** SQL *Server* yang telah aktif diperlihatkan **dalam** Gambar 6.1.

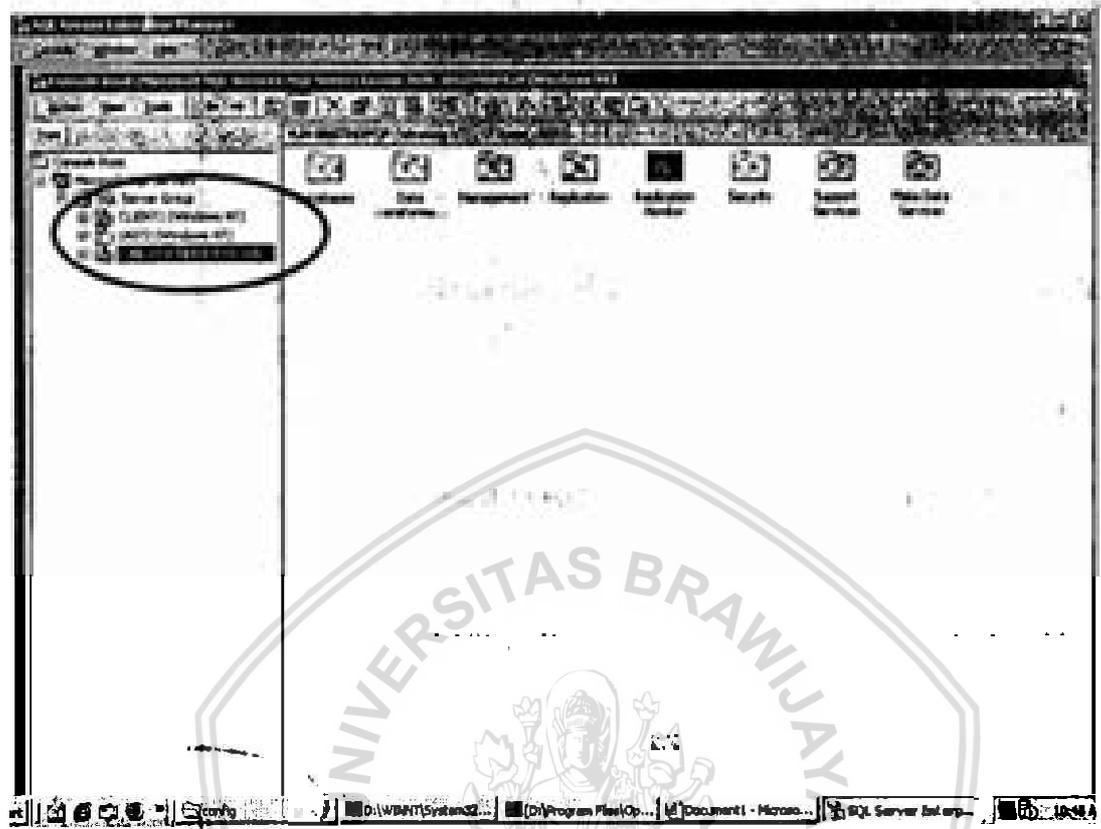
Gambar 6.4. memperlihatkan koneksi yang terjadi antara komputer cabang dengan alamat 10.3.0.1 dan komputer **pusat** yang mempunyai alamat 10.3.0.2 pada **port** 139 dengan melakukan koneksi **server database** SQL *Server*.



Gambar 6.4. Daftar koneksi komputer cabang setelah melakukan koneksi dengan *database* SQL Server

Sumber: Pengujian

melibatkan dua *database server* yaitu data pada *server* di komputer cabang maupun di komputer pusat.



Gambar 6.6. Tampilan SQL Server Enterprise Manager pada komputer pusat setelah koneksi.

Sumber: Pengujian

Karena pada Enterprise Manager di komputer cabang tidak dibuat **registrasi** baru untuk memunculkan *server* di *tree SQL Server Group* seperti dijelaskan pada Sub Bab 5.2.2.4 tentang **Registrasi Remote SQL Sewer** maka *sewer* dan data pada komputer pusat tidak terlihat di komputer cabang.

f. Kesimpulan

Komputer cabang dapat melakukan koneksi dengan *database SQL Server* yang ada pada komputer pusat dengan **pengaturan** koneksi pada Sistem Operasi dan **SQL Sewer**.

6.1.3. Pengujian Otomatisasi *Dial Up*

Pada pengujian ini akan dilakukan otomatisasi *Dial Up* dalam hubungannya dengan koneksi antar komputer.

a. Tujuan

- Mengetahui apakah koneksi antar komputer bisa dilakukan secara otomatis dengan *schedule* yang sudah ditentukan sebelumnya tanpa harus melakukan *Dial Up* secara manual.

b. Spesifikasi dan Konfigurasi Komputer

- Empat buah komputer, komputer pertama dan kedua dijadikan sebagai komputer cabang dengan IP 172.17.0.2 dan 172.17.0.3. komputer ketiga dijadikan komputer server RAS dengan IP PPP 172.17.0.1 dan IP Lokal 192.168.1.1 sedangkan komputer keempat dijadikan komputer pusat dengan IP 192.168.1.4, dengan gateway 192.168.1.1.
- Komputer pusat: Prosesor Intel Pentium 4 - 2,26 GHz, memori 512 MB.
- Komputer server RAS : Processor Intel Centrino - 1.8 GHz, memori 512 MB
- Komputer cabang: Prosesor Intel Pentium Dual Core @1.6 GHz, memori 512 MB.
- Sistem Operasi Microsoft Windows 2000 Server dan Windows XP.

c. Software Aplikasi

- Server database SQL Server 2000.
- Control Panel.

d. Prosedur Pengujian

- Melakukan penjadwalan pada *Control Panel* dengan membuat *batch file* untuk mengaktifkan koneksi, dimana isi dari file tersebut dapat dilihat pada Sub Bab 5.1.2 tentang *Otomatisasi Dial Up*.
- Melakukan *setting schedule* pada Control Panel | *Scheduled Task*.
- Menunggu sampai pada saat yang telah dijadwalkan apakah akan terjadi koneksi secara otomatis.

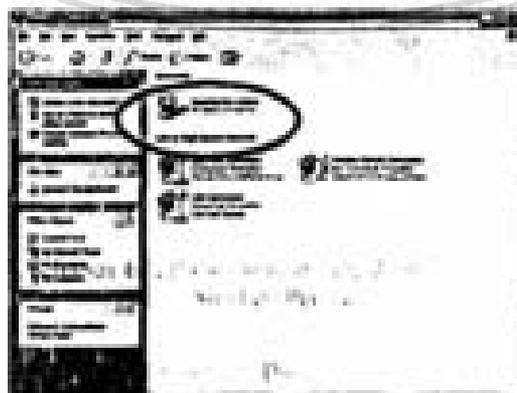
e. Hasil Pengujian

Pada waktu yang telah dijadwalkan proses koneksi berlangsung secara otomatis terlihat dalam Gambar 6.7. Proses koneksi dari komputer cabang terjadi ke komputer server RAS dengan proses *setting* koneksi sama seperti proses koneksi antar komputer secara manual, tetapi dengan ditambah proses *shedulling* pada komputer cabang akan membuat komputer cabang secara otomatis melakukan proses *Dial* Up ke komputer server RAS. Dengan menggunakan *batch file* berupa perintah untuk melakukan koneksi maka koneksi komputer akan dilakukan secara otomatis, apabila proses koneksi yang dilakukan mengalami kegagalan maka dengan perintah pada *batch file* tersebut memungkinkan pengaturan adanya *dial* ulang. Sehingga koneksi akan dilakukan pada penjadwalan koneksi hari berikutnya.



Gambar 6.7. Proses koneksi terjadi secara otomatis
Sumber: Pengujian

Gambar 6.8. memperlihatkan bahwa pada saat koneksi berhasil dilakukan dengan cara penjadwalan yang telah ditentukan dan secara otomatis terlihat pada komputer pusat sudah berhasil mendeksi komputer cabang sedang melakukan koneksi pada waktu tersebut.



Gambar 6.8. Komputer cabang sedang melakukan koneksi pada komputer server RAS
Sumber: Pengujian

Selain proses koneksi dilakukan secara otomatis dengan waktu yang telah ditentukan proses pemutusan koneksi **juga** bisa dilakukan dengan cam yang **sama** dengan proses koneksi secara otomatis hanya dengan program yang berbeda. Dalam Gambar 6.9. **ditunjukkan** proses **pemutusan koneksi dengan** waktu yang **sesuai dengan waktu** yang telah di-set **sebelumnya**.



Gambar 6.9. Proses pemutusan koneksi terjadi secara otomatis
Sumber: Pengujian

f. **Kesimpulan**

Proses koneksi dan **pemutusan** koneksi antar komputer bisa dilakukan secara otomatis dengan penjadwalan yang **sudah ditentukan** dan di-set, **jadi pada saat** yang telah **ditentukan proses koneksi dan pemutusan koneksi** akan **berlangsung** sendiri **tanpa** harus dilakukan proses **dial** secara manual untuk **selanjutnya** dilakukan proses **replikasi** dalam hubungannya **update** data **antara** server RAS dan komputer cabang.

6.1.4. Pengujian **Otomatisasi** koneksi VPN

Pada pengujian ini akan dilakukan otomatisasi **koneksi** VPN dalam hubungannya dengan koneksi **antar** komputer.

a. **Tujuan**

- Mengetahui apakah koneksi **antar** komputer bisa dilakukan secara **otomatis** dengan **schedule** yang **sudah ditentukan sebelumnya** tanpa harus melakukan **start VPN connection** secara manual.



b. Spesifikasi dan Konfigurasi Komputer

- Empat buah komputer, komputer pertama dan kedua dijadikan sebagai komputer cabang dengan IP 172.17.0.2 dan 172.17.0.3 . komputer ketiga dijadikan komputer *server* RAS dengan IP PPP 172.17.0.1 dan IP Lokal 192.168.1.1 sedangkan komputer keempat dijadikan komputer pusat dengan IP 192.168.1.4. dengangateway192.168.1.1.
- Komputer pusat: Prosesor Intel Pentium 4 - 2,26 GHz, memori 512 MB.
- Komputer server RAS : Processor Intel Centrino - 1.8 GHz, memori 512 MB
- Komputer cabang: Pmsesor Intel Pentium Dual Core - @1.6 GHz, memori 512 MB.
- Sistem Operasi Microsoft Windows 2000 Server dan Windows XP,

g. Software Aplikasi

- *Server* database SQLServer 2000.
- Open VPN 2.0.9.
- Control Panel.

c. Proscdur Pengujian

- Melakukan penjadwalan pada *Control* Panel dengan membuat batch file untuk mengaktifkan koneksi, dimana isi dari *file* tersebut dapat dilihat pada Sub Bab 5.2.1. tentang *Otomatisasi Koneksi VPN*.
- Melakukan setting schedule pada Control Panel | Scheduled Task.
- Menunggu sampai pada saat yang telah dijadwalkan apakah akan terjadi koneksi secara otomatis,

d. Hasil Pengujian

Pada waktu yang telah dijadwalkan proses koneksi berlangsung secara otomatis. Proses koneksi dari komputer cabang terjadi ke komputer pusat dengan proses setting koneksi sama seperti proses koneksi antar komputer secara manual, tetapi dengan ditambah proses shedulling pada komputer cabang akan membuat komputer cabang secara otomatis melakukan proses *Diul Up* ke komputer pusat. Dengan menggunakan batch file berupa perintah untuk melakukan koneksi maka koneksi komputer akan dilakukan secara otomatis, apabila proses koneksi yang dilakukan mengalami kegagalan



maka dengan perintah pada *batch file* tersebut memungkinkan pengaturan koneksi VPN ulang. Sehingga koneksi akan dilakukan pada penjadwalan koneksi hari berikutnya. Pengaturan batch file dan *scheduling* tidak hanya dilakukan pada komputer pusat saja, tetapi dilakukan pada komputer cabang juga karena koneksi VPN ini memerlukan autentikasi yang dilakukan masing-masing komputer. Untuk kasus *scheduling* supaya dapat terjadi secara bersamaan perlu adanya pengesetan setting *Date and Time Properties* yang sama. Pada gambar 6.10. adalah contoh proses koneksi VPN yang telah berjalan secara otomatis yang terjadi pada komputer pusat.

```

Sun May 18 13:03:10 2008 OpenVPN [cmd] [5361CF30-3EAB] [2] built on Oct 1 2007
Sun May 18 13:03:10 2008 IMPORTANT: OpenVPN's default port number is now 1194. b
used on an official port number assignment by IANA. OpenVPN 2.0-beta16 and ear
Sun May 18 13:03:10 2008 used 5000 as the default port.
Sun May 18 13:03:10 2008 WARNING: --ping should normally be used with --ping-res
part or --ping-exit
Sun May 18 13:03:10 2008 Static Encrypt: Cipher 'BF-CBC' initialized with 128 bi
Sun May 18 13:03:10 2008 Static Encrypt: Using 160 bit message hash 'SHA1' for H
Sun May 18 13:03:10 2008 Static Decrypt: Cipher 'BF-CBC' initialized with 128 bi
Sun May 18 13:03:10 2008 Static Decrypt: Using 160 bit message hash 'SHA1' for H
Sun May 18 13:03:10 2008 TAP-WIN32 device (UPN) opened: \\.\Global\{5361CF30-3EAB
Sun May 18 13:03:10 2008 TAP-Win32 Driver Version 8.4
Sun May 18 13:03:10 2008 TAP-Win32 MTU 1500
Sun May 18 13:03:10 2008 Notified IRP-Win32 driver to set a DHCP IP/netmask of 1
Sun May 18 13:03:10 2008 Successful ARP Flush on interface [2] (5361CF30-3EAB-4F
Sun May 18 13:03:10 2008 Data Channel MTU pa [ L:1576 D:1450 EP:44 EB:4 ET:32
Sun May 18 13:03:10 2008 Local Options hash (VER=U4): '9e3b3087'
Sun May 18 13:03:10 2008 Expected Remote Options hash (VER=U4): '9e3b3087'
Sun May 18 13:03:10 2008 UDPv4 link local (bound): [undef]:1194
Sun May 18 13:03:10 2008 UDPv4 link remote: 172.17.0.2:1194

```

Gambar 6.10. Proses koneksi terjadi secara otomatis pada komputer pusat

Gambar 6.11. dan 6.12. memperlihatkan bahwa pada saat koneksi berhasil dilakukan dengan cara penjadwalan yang telah ditentukan dan secara otomatis terlihat pada komputer pusat dan komputer cabang sudah berhasil melakukan koneksi VPN satu sama lain.



Selain proses koneksi dilakukan secara otomatis dengan waktu yang telah ditentukan, proses pemutusan koneksi dilakukan dengan cara mengeset timer pada scheduled task dimana pada waktu yang telah diset proses koneksi VPN akan mati (menutup *window command prompt*) dengan sendirinya.

e. **Kesimpulan**

Proses koneksi dan pemutusan koneksi antar komputer bisa dilakukan secara otomatis dengan penjadwalan yang sudah ditentukan dan di-set, jadi pada saat yang telah ditentukan proses koneksi dan pemutusan koneksi akan berlangsung sendiri tanpa harus dilakukan proses koneksi VPN secara manual untuk selanjutnya dilakukan proses replikasi dalam hubungannya update data antara komputer pusat dan komputer cabang.

6.15. **Pengujian Replikasi**

Proses replikasi yang dilakukan akan berlangsung untuk meng-update data terbaru baik dari pusat maupun dari cabang.

a. Tujuan

- Mengetahui proses replikasi akan berlangsung pada saat koneksi antar komputer dimulai.

b. **Spesifikasi dan Konfigurasi Komputer**

- Empat buah komputer, komputer pertama dan kedua dijadikan sebagai komputer cabang dengan IP 172.17.0.2 dan 172.17.0.3 . komputer ketiga dijadikan komputer server RAS dengan IP PPP 172.17.0.1 dan IP Lokal 192.168.1.1 sedangkan komputer keempat dijadikan komputer pusat dengan IP 192.168.1.4. dengan gateway 192.168.1.1.
- Komputer pusat: Prosesor Intel Pentium 4 - 2.26 GHz, memori 512 MB.
- Komputersewer RAS : Processor Intel Centrino - 1.8 GHz, memori 512 MB
- Komputer cabang: Prosesor Intel Pentium Dual Core - @1.6 GHz, memori 512 MB.
- Sistem Operasi Microsoft Windows 2000 Server dan Windows XP.



c. Software Aplikasi

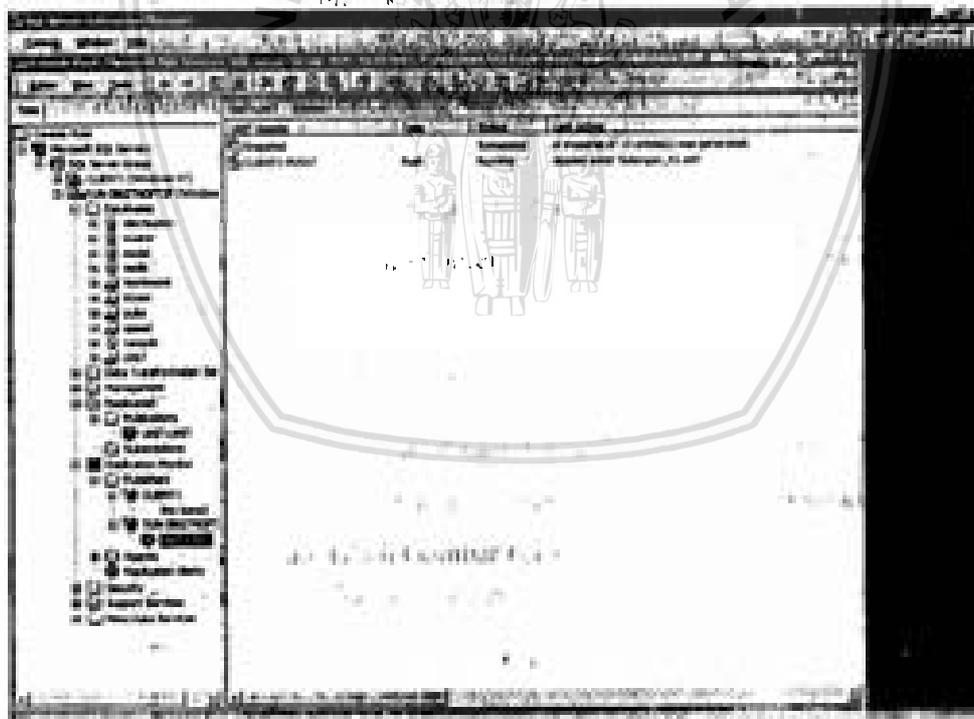
- *Server database SQL Server 2000.*
- Open VPN 2.0.9.

d. Prosedur Pengujian

- Melakukan konfigurasi komputer pusat sebagai publisher.
- Membuat publikasi pada komputer pusat untuk kemudian di-push ke *database* di komputer cabang.
- Pada saat komputer cabang melakukan koneksi ke komputer pusat maka proses *push* dan update data akan dilakukan.
- Mengamati data pada komputer cabang dan mengamati monitor replikasi pada komputer pusat.

e. Hasil Pengujian

Dari hasil replikasi maka didapat monitor replikasi pada database SQL Server di komputer pusat yang menunjukkan bahwa proses *push* publikasi dan update data sedang berjalan seperti diperlihatkan dalam Gambar 6.13.



Gambar 6.13. Monitor replikasi pada *database server* komputer cabang.
Sumber: Pengujian

Proses replikasi berhasil bila terlihat data yang dipublikasikan dari *database server* pada komputer pusat bisa sampai ke *database server* di komputer cabang sesuai dengan filter yang telah ditentukan, tentunya proses replikasi berlangsung pada saat komputer pusat dan komputer cabang sedang melakukan koneksi atau komputer-komputer tersebut terhubung.

Seperti yang diperlihatkan dalam Gambar 6.14. bahwa *database server* di komputer cabang akan terisi dengan data yang sudah difilter, baik tabel-tabel, *views* dan *stored procedures* maupun elemen lain yang terdapat di *database server* di komputer pusat yang dianggap perlu dipublikasikan ke komputer cabang agar bisa nantinya data-data tersebut dilakukan transaksi dalam *database server* di komputer cabang-cabang.

Gambar 6.15. memperlihatkan bahwa Tabel Pelanggan pada *database server* di komputer cabang sudah berisi data yang hanya dengan Kode_kec sama dengan 'A'. Karena proses filter pada saat pembuatan publikasi maka nantinya masing-masing komputer cabang akan berbeda data dan bisa melakukan transaksi pada komputer masing-masing, Dengan proses filter tentunya proses *update* data akan lebih cepat karena hanya data dengan kode tertentu saja yang dilakukan publikasi pada saat komputer terhubung.



Gambar 6.14. Tabel-tabel pada *database server* di komputer cabang.
 Sumber: Pengujian



ID Pelanggan	Nama	Alamat	No. Telp
1	A. B. C.	Jl. Sudirman No. 1	021-1234567
2	D. E. F.	Jl. Diponegoro No. 2	021-7654321
3	G. H. I.	Jl. Veteran No. 3	021-9876543
4	J. K. L.	Jl. Soekarno No. 4	021-2109876
5	M. N. O.	Jl. Sisinga No. 5	021-5432109
6	P. Q. R.	Jl. H. S. Ronggolawe No. 6	021-8765432
7	S. T. U.	Jl. Pahlawan No. 7	021-1098765
8	V. W. X.	Jl. Garuda No. 8	021-4321098
9	Y. Z. A.	Jl. Bhayangkara No. 9	021-7654321
10	B. C. D.	Jl. Satria No. 10	021-0987654

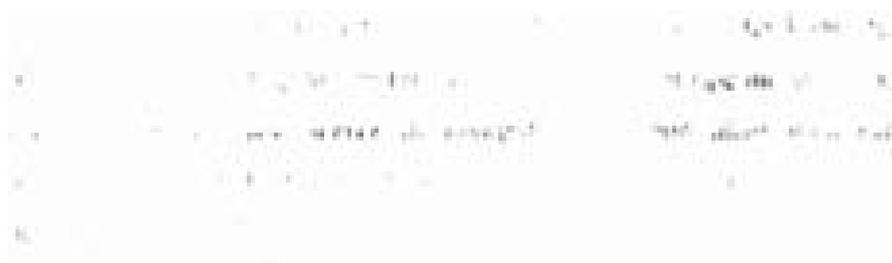
Gambar 6.15. Data pada Tabel Pelanggan di database server komputer cabang
Sumber: Pengujian

f. Kesimpulan

Berjalannya proses replikasi maka data yang akan dipublikasikan ke database server di komputer cabang dilakukan filter agar hanya data tertentu saja yang diterima oleh komputer cabang untuk selanjutnya transaksi dilakukan di komputer-komputer cabang dengan data yang berbeda tiap komputer cabang. Penggunaan tipe merge setiap ada perubahan data pada database server di komputer manapun akan membuat perubahan juga pada komputer lain yang terhubung baik dalam jaringan maupun terhubung dalam replikasi.

6.2. Pengujian Sistem secara Keseluruhan

Pengujian sistem secara keseluruhan adalah gabungan dari pengujian masing-masing blok dimana semua sistem sudah dihubungkan dan proses koneksi akan dilakukan secara otomatis dan waktu dua komputer terhubung proses replikasi juga berlangsung.



- a. Tujuan
- **Mengetahui** proses replikasi berlangsung dengan koneksi yang terjadwal dan update data yang dilakukan **akan berpengaruh** pada komputer lain yang terhubung.
- b. **Spesifikasi** dan **Konfigurasi** Komputer
- Empat buah komputer, komputer **pertama dan** kedua dijadikan sebagai komputer cabang dengan IP 172.17.0.2 dan 172.17.0.3 . komputer **ketiga** dijadikan komputer server **RAS** dengan IP PPP 172.17.0.1 dan IP Lokal 192.168.1.1 sedangkan komputer keempat dijadikan komputer **pusat** dengan IP **192.168.1.4**. dengan **gateway 192.168.1.1**.
 - Komputer **pusat**: Prosesor Intel Pentium 4 - **2,26** GHz, memori 512 MB.
 - Komputer server RAS : **Processor** Intel Centrino - 1.8 GHz, memori **512 MB**
 - Komputer cabang: Prosesor Intel Pentium Dual **Core** - @1.6 GHz, memori 512 MB.
 - **Sistem Operasi Microsoft Windows 2000** Server dan Windows XP.
- c. Software **Aplikasi**
- **Server** database **SQL Server 2000**.
 - **Microsoft** Access 2000.
 - Windows Explorer.
 - Open VPN **2.0.9**.
 - **Control Panel**.
- d. Prosedur **Pengujian**
- **Menghubungkan** semua komputer, modem. PABX dan Ethernet hub **sesuai** perancangan.
 - Mengkonfigurasi **masing-masing** komputer **untuk mendukung** proses **koneksi secara** terjadwal.
 - Mengkonfigurasi masing-masing komputer untuk proses replikasi.
 - Mengarnati proses koneksi dan proses replikasi masing-masing komputer **cabang** yang berlangsung bergantian.
 - Mengubah data dengan **menambah atau** menghapus data dari satu komputer.

- Mengamati perubahan pada komputer lain yang sedang terhubung jaringan.

e. Hasil Pengujian

Pada jadwal yang sudah diatur untuk komputer cabang melakukan koneksi ke komputer server RAS, setelah komputer cabang dan komputer server RAS terhubung sesuai dengan jadwal yang ditentukan akan terjadi proses koneksi VPN secara otomatis yang dilakukan oleh komputer cabang dan komputer pusat. Dan segera setelah autentikasi dan koneksi VPN telah terjadi, maka proses replikasi akan berjalan sesuai dengan jadwal publikasi yang di-push sudah diatur sesuai dengan jadwal koneksi antar komputer.

Sesuai dengan jadwal yang diatur dalam Scheduled Task, komputer cabang berhasil melakukan proses *Dial Up*. Hal ini dapat terlihat dengan dijalankannya *batch file* yang berfungsi untuk melakukan koneksi pada komputer cabang. Proses pembuatan koneksi yang terjadi pada masing-masing komputer cabang dapat dilihat dalam Gambar 6.16.



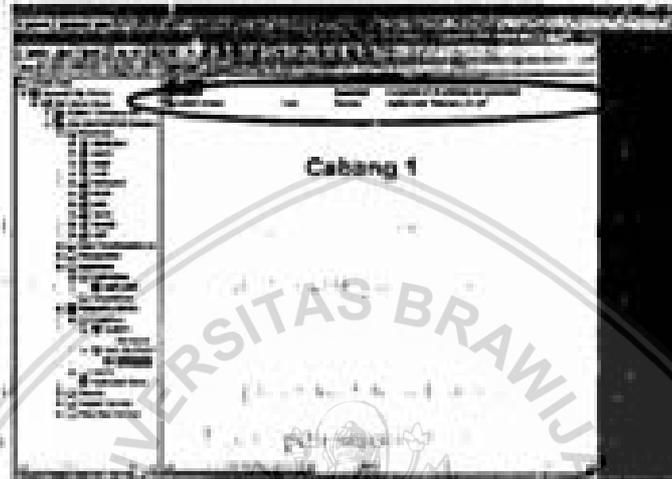
Gambar 6.16. Proses koneksi terjadi secara otomatis di komputer cabang
Sumber: Pengujian

Demikian juga dengan proses pemutusan koneksi, dapat dilakukan secara otomatis pada masing-masing komputer cabang. Proses yang terjadi dapat dilihat sebagaimana ditunjukkan dalam Gambar 6.9. di atas.

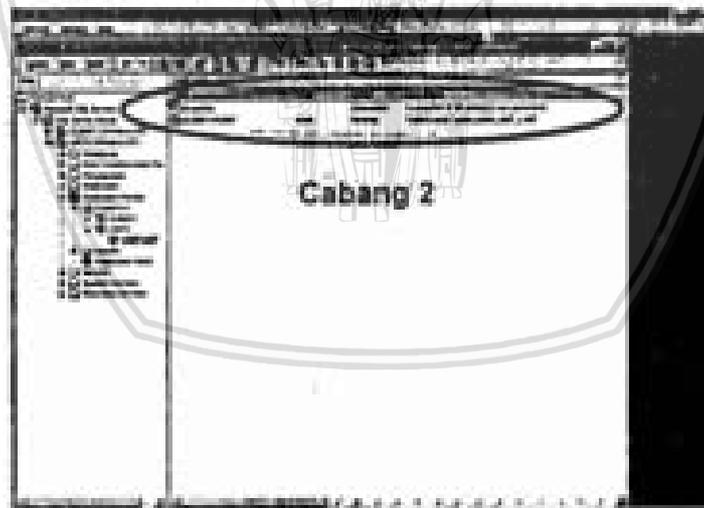
Segera setelah proses *Dial Up* dari komputer cabang menuju komputer server RAS, maka sesuai *scheduling* yang dilakukan di Scheduled *Task* akan terjadi koneksi VPN antara komputer cabang dengan komputer pusat. Koneksi tersebut dianggap

berhasil jika akan keluar window yang menggambarkan status koneksi tersebut telah berhasil dan berjalan dengan benar seperti yang ditunjukkan pada gambar 6.11, dan gambar 6.12.

Replikasi antara komputer pusat dan komputer cabang juga dapat berlangsung sebagaimana halnya dengan hasil pengujian yang telah dilakukan sebelumnya. Berlangsungnya proses replikasi antara komputer cabang dengan komputer pusat dapat dilihat dalam Gambar 6.17 dan Gambar 6.18..



Gambar 6.17. Proses replikasi berlangsung di komputer cabang' I
Sumber: Pengujian



Gambar 6.18. Proses replikasi berlangsung di komputer cabang 2
Sumber: Pengujian

f. Kesimpulan

Replikasi melalui jaringan VPN bisa dilakukan dengan proses setting yang dilakukan baik dalam Sistem Operasi maupun dalam SQL Server. Koneksi yang

repository.ub.ac.id

dilakukan adalah dengan penjadwalan yang ditentukan dan akan terlaksana secara otomatis. *Update* data terjadi secara dua arah baik dari *database server* di komputer cabang ke *database server* di komputer pusat, maupun dari *database server* di komputer pusat ke *database server* di komputer cabang.

6.3. Pengujian Koneksi *Database* pada Aplikasi Microsoft Access

Mengacu pada latar belakang penelitian ini agar kwitansi bisa dicetak langsung di cabang-cabang dengan memanfaatkan proses replikasi maka pengujian ini dilakukan untuk mengetahui apakah aplikasi Microsoft Access yang ada pada PDAM bisa dijalankan dan terisinkan dengan data-data yang sama pada *database SQL Server* yang sudah dilakukan proses *restore database* sebelumnya.

Hal ini dilakukan setelah pengujian sistem secara keseluruhan dan setelah proses replikasi dijalankan sehingga bisa diketahui apakah proses transaksi dan pembuatan kwitansi maupun laporan bisa dilakukan melalui aplikasi yang sudah ada di PDAM pada Microsoft Access baik di komputer pusat maupun di komputer cabang pada saat komputer tidak sedang terhubung (*connect*).

a. Tujuan

- Mengetahui koneksi antara *database SQL Server* dengan aplikasi Microsoft Access yang ada pada PDAM dan mengetahui aplikasi Microsoft Access yang sudah ada di PDAM bisa digunakan untuk proses transaksi dan pembuatan kwitansi maupun laporan baik yang dilakukan di komputer pusat maupun di cabang-cabang setelah komputer tidak terhubung (*connect*).

b. Spesifikasi dan Konfigurasi Komputer

- Empat buah komputer, komputer pertama dan kedua dijadikan sebagai komputer cabang dengan IP 172.17.0.2 dan 172.17.0.3. komputer ketiga dijadikan komputer server RAS dengan IP PPP 172.17.0.1 dan IP Lokal 192.168.1.1 sedangkan komputer keempat dijadikan komputer pusat dengan IP 192.168.1.4. dengan gateway 192.168.1.1.
- Komputer pusat: Prosesor Intel Pentium 4 - 2,26 GHz, memori 512 ME.
- Komputer server RAS : Prosesor Intel Centrino - 1.8 GHz, memori 512

ME



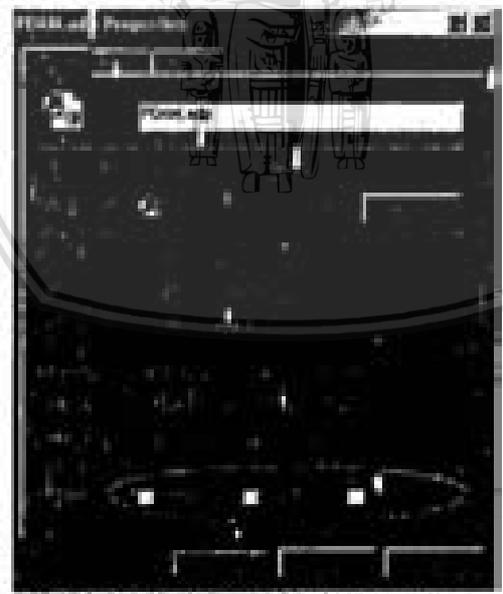
- Komputer cabang: **Prosesor Intel Pentium Dual Core - @1.6 GHz, memori 512 MB.**
- Sistem Operasi Microsoft Windows 2000 **Server** dan Windows XP.

c. **Software Aplikasi**

- Server **database** SQL Server 2000.
- **Microsoft Access 2000.**
- windows Explorer

d. **Prosedur Pengujian**

- **Setelah dilakukan replikasi komputer pusat dan komputer cabang diputuskan koneksinya (disconnect).**
- **Meng-copy file PDAM.adp (Access Data Project) yang ada di komputer pusat ke komputer cabang.**
- **Menjalankan windows Explorer dan membuka properties file PDAM.adp baik di komputer pusat maupun di komputer cabang.**
- **Tidak mengaktifkan semua attributes pada properties file seperti diperlihatkan dalam Gambar 6.19.**



Gambar 6.19. Properties file PDAM.adp
Sumber: pengujian

- Menjalankan *file* PDAM.adp tersebut dengan diikuti menekan tombol Shift pada *keyboard* agar terlihat tampilan design dan tidak masuk ke tampilan aplikasi.
- Menjalankan proses koneksi dan mengetes dengan cara File | Connection | Test connection

e. Hasil Pengujian

Dari hasil pengujian maka bisa dilihat secara langsung koneksi berhasil atau tidak dengan ditunjukkan tampilan seperti yang ditunjukkan dalam Gambar 6.20.



Gambar 6.20. Koneksi berhasil
Sumber: Pengujian

Apabila koneksi sudah berhasil tahap pengujian selanjutnya bisa dilihat dengan munculnya atribut-atribut antara lain tabel-tabel, *view-view* dan *procedure-procedur* yang ada pada Microsoft Access seperti yang ditunjukkan dalam Gambar 6.21. terdapat beberapa tabel yang muncul setelah dilakukan *restore database* dan koneksi *SQL Server*, begitu juga dengan tabel-tabel tersebut bisa dilihat sudah terisi dengan data seperti ditunjukkan dalam Gambar 6.22. terlihat Tabel Bulan sudah terisi dengan data bulan Januari sampai Desember.



Gambar 6.21. Tabel-tabel pada Microsoft Access
Sumber: Pengujian



Gambar 6.23. Tabel Dalam
Sumber: Pengujian

f. **Kesimpulan**

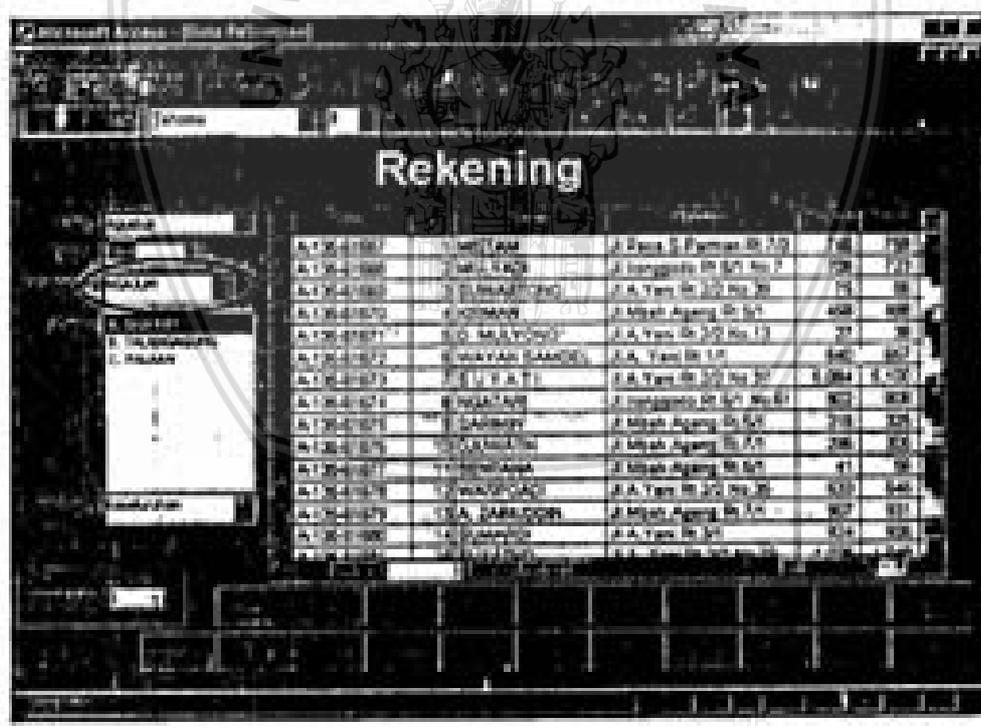
Setelah dilakukan restore *database* dan koneksi antara SQL Server dan aplikasi Microsoft Access maka didapatkan bahwa PDAM bisa melakukan transaksi dengan menggunakan *interface* aplikasi Microsoft Access dalam pembuatan laporan maupun kwitansi seperti yang ditunjukkan dalam Gambar 6.22.

Setelah proses replikasi maka *database* di komputer cabang 1 dan komputer cabang 2 akan terisi data masing-masing hanya pelanggan yang mempunyai kode_kec='A' untuk cabang 1 dan kode_kec='B' untuk cabang 2 sehingga dengan prosedur pengujian koneksi *database* pada aplikasi Microsoft Access dilakukan sama dengan prosedur pengujian yang sudah dijelaskan diatas akan membuat transaksi dan pembuatan kwitansi maupun laporan bisa dilakukan melalui aplikasi Microsoft Access yang ada di kantor cabang dengan data hanya pelanggan di cabang - cabang tersebut. Tampilan aplikasi Microsoft Access yang ada di cabang dengan data hanya pelanggan yang mempunyai kode_kec='A', yaitu kecamatan Ngajum pada cabang 1 dan kode_kec='B' yaitu kecamatan Kepanjen pada cabang 2 dapat dilihat dalam Gambar 6.24 dan Gambar 6.25.





Gambar 6.23. Tampilan aplikasi Microsoft Access di PDAM pusat
Sumber: Pengujian



Gambar 6.24. Tampilan aplikasi Microsoft Access di kantor cabang 1
Sumber: Pengujian





Gambar 6.15. Tampilan aplikasi Microsoft Access di cabang 2

Sumber: Pengujian

6.4. Pengujian Keamanan Data

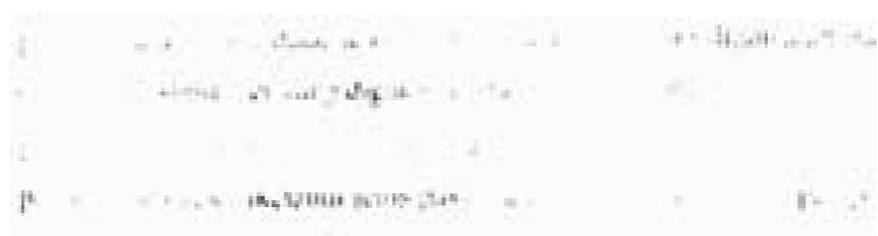
Pengujian ini dilakukan untuk mengetahui keamanan data dan paket-paket data baik yang dikirim atau yang diterima, karena keunggulan dari penggunaan VPN ini adalah paket data yang tersampaikan tidak dapat terbaca oleh komputer-komputer yang ditewati.

a. Tujuan

- Mengetahui paket-paket data, jenis protokol dan port yang digunakan pada masing-masing komputer yang digunakan dalam percobaan.
- Mengetahui apakah data yang tersampaikan dari komputer cabang menuju komputer pusat tidak diketahui jenis paket data, jenis protokol, maupun port yang digunakan oleh komputer yang dilewati.

b. Spesifikasi dan Konfigurasi Komputer

- Empat buah komputer, komputer pertama dan kedua dijadikan sebagai komputer cabang dengan IP 172.17.0.2 dan 172.17.0.3 . komputer ketiga dijadikan komputer server RAS dengan IP PPP 172.17.0.1 dan IP Lokal



192.168.1.1 sedangkan komputer keempat dijadikan komputer pusat dengan IP 192.168.1.4. dengan gateway 192.168.1.1.

- Komputer pusat: Prosesor Intel Pentium 4 " 2,26 GHz, memori 512 MB.
- Komputer server RAS : Processor Intel Centrino – 1.8 GHz, memori 512 MB.
- Komputer cabang: Prosesor Intel Pentium Dual Core " @1.6 GHz, memori 512 MB.
- Sistem Operasi Microsoft Windows 2000 Server dan Windows XP.

c. Software Aplikasi

- *Server* database SQL Server 2000.
- OpenVPN2.0.9.
- *Control* Panel.
- *Software network* analysis Ethereal.

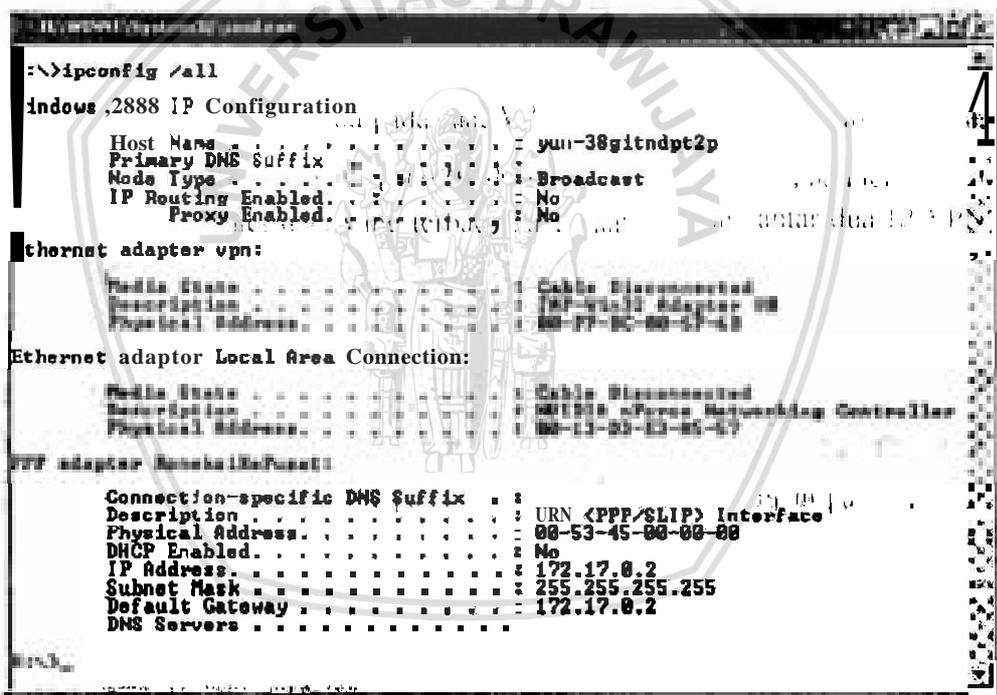
d. Prosedur Pengujian

- Melakukan koneksi Dial Up dari komputer cabang ke komputer Server RAS melalui modem dan PABX.
- Menjalankan command Prompt dari Start | Run... | Open:cmd.exe |.
- Menjalankan Command "ipconfig /all", untuk melihat koneksi apa saja yang available pada komputer cabang.
- Melakukan ping ke komputer IP PPP server RAS (172.17.0.1) dan diambil screenshot-nya.
- Melakukan ping ke IP VPN komputer pusat (10.3.0.2) dan diambil screenshot-nya.
- Melakukan ping ke IP Lokal komputer pusat (192.168.1.3) dan diambil screenshot-nya.
- Melakukan registrasi database pada enterprise manager apakah bias melakukan registrasi atau tidak dan diambil screenshot-nya.
- Melakukan koneksi privat /VPN inenggunakan program Open VPN 2.0.9. dari komputer cabang ke komputer pusat.
- Melakukan ping ke IP VPN komputer pusat (10.3.0.2) dan diambil screenshot-nya.

- Melakukan registrasi database pada enterprise manage? apakah bias melakukan registrasi atau tidak dan diambil *screenshot*-nya.
- Bandingkan screen shot antara sebelum VPN dijalankan dan Setelah VPN dijalankan.
- *Capture* paket-paket data pada saat VPN berjalan dan melakukan replikasi. Dalam hal ini dilakukan capture pada komputer: RAS untuk mengetahui apakah ada paket data yang terbaca pada saat replikasi antar dua IP VPN berjalan.

e. Hasil Pengujian

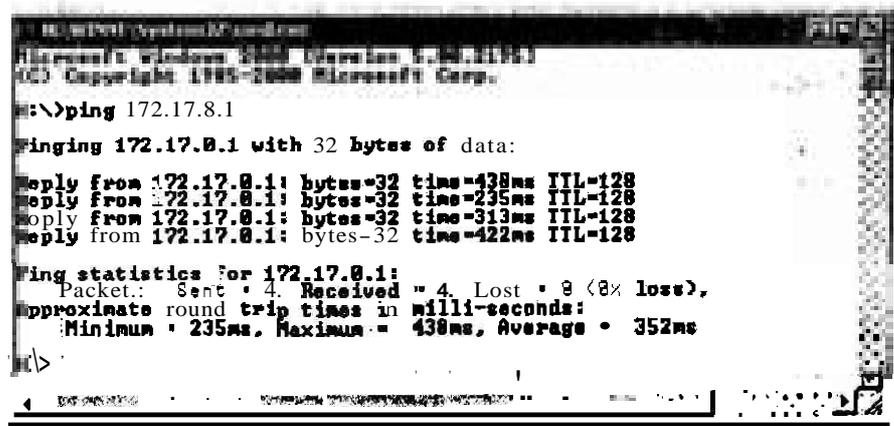
Dari hasil pengujian dapat dilihat bahwa setelah koneksi *Dial* Up dilakukan berikut adalah koneksi yang available pada komputer cabang ditunjukkan pada gambar 6.26.:



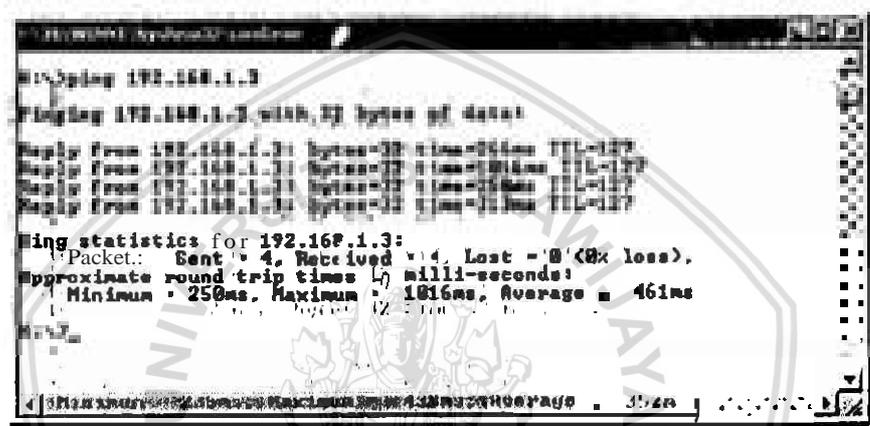
Gambar 6.26. Tampilan Konfigurasi seluruh koneksi pada komputer cabang
Sumber: Pengujian

Setelah mengerti koneksi apa saja yang *available* pada komputer cabang maka dilakukan pengujian berupa ping ke tiga IP yang disebutkan pada prosedur pengujian tadi dan hasilnya ditunjukkan pada Gambar 6.27., Gambar 6.28., Gambar 6.29. berikut :

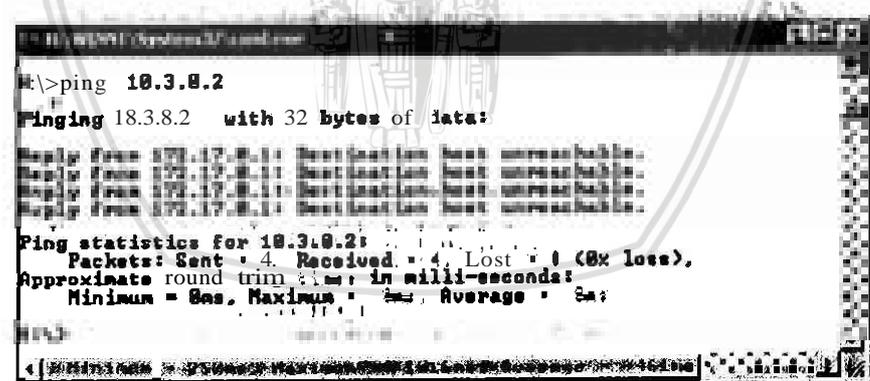




Gambar 6.27. Tampilan hasil Ping komputer cabang ke komputer RAS
 Sumber: Pengujian

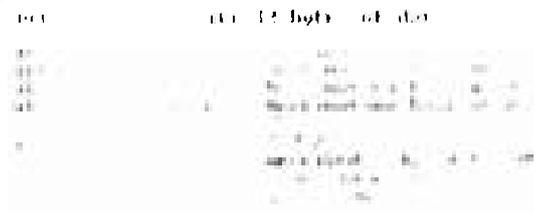


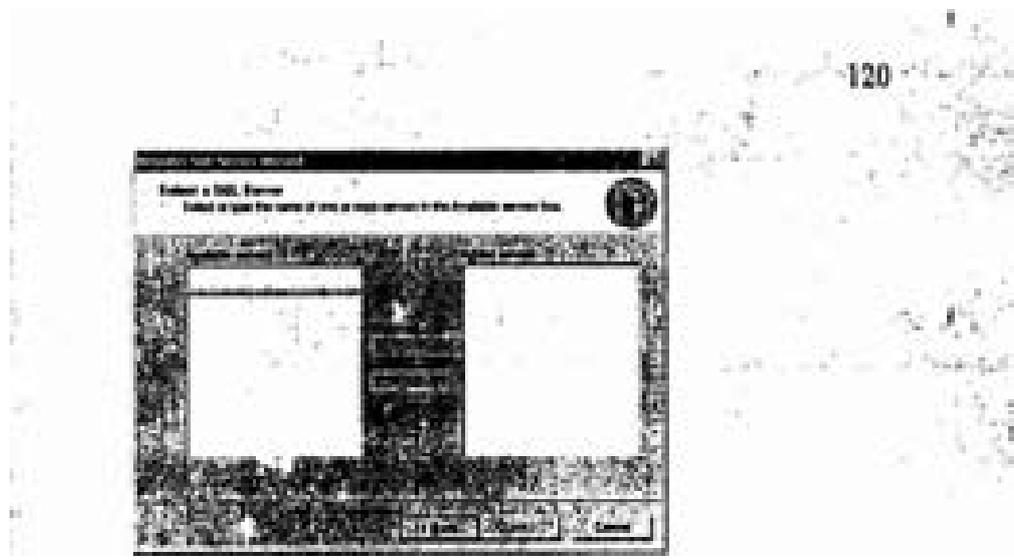
Gambar 5.28. Tampilan hasil Ping komputer cabang ke komputer pusat
 Sumber: Pengujian



Gambar 6.29. Tampilan hasil Ping komputer cabang ke IP VPN komputer pusat
 Sumber: Pengujian

Setelah memperoleh hasil pengujian ping ke tiga IP tadi, kita lakukan registrasi database apakah bisa dilakukan registrasi jika kita belum menjalankan VPN yaitu dengan cara langsung mengetikkan alias dari database yang akan diregistrasi dan hasilnya ditunjukkan pada Gambar 6.30, Gambar 6.31 dan Gambar 6.32 berikut :

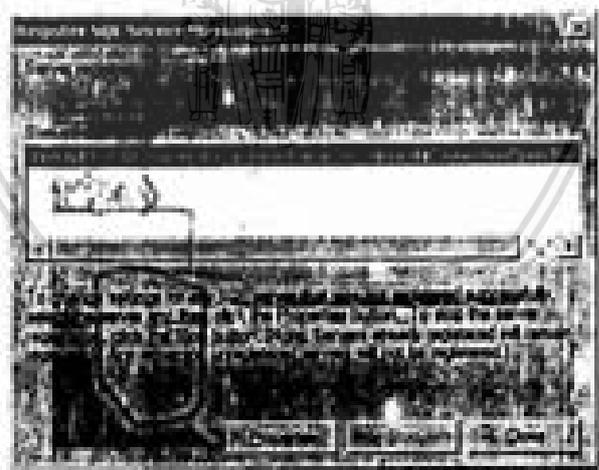




Gambar 6.30. Tampilan server *database* yang available untuk dikoneksikan
 Sumber: Pengujian



Gambar 6.31. Tampilan *finishing* registrasi *database*
 Sumber: Pengujian



Gambar 6.32. Tampilan pesan error setelah registrasi *database*
 Sumber: Pengujian

Setelah itu kita jalankan VPN oleh komputer cabang dan komputer pusat. Kemudian kita lakukan *ping* IP VPN komputer pusat oleh komputer cabang dan dilakukan



registrasi *database* kembali. Hasilnya ditunjukkan pada Gambar 6.33 dan Gambar 6.34 berikut :

```

C:\>ping 10.3.0.2

Pinging 10.3.0.2 with 32 bytes of data:

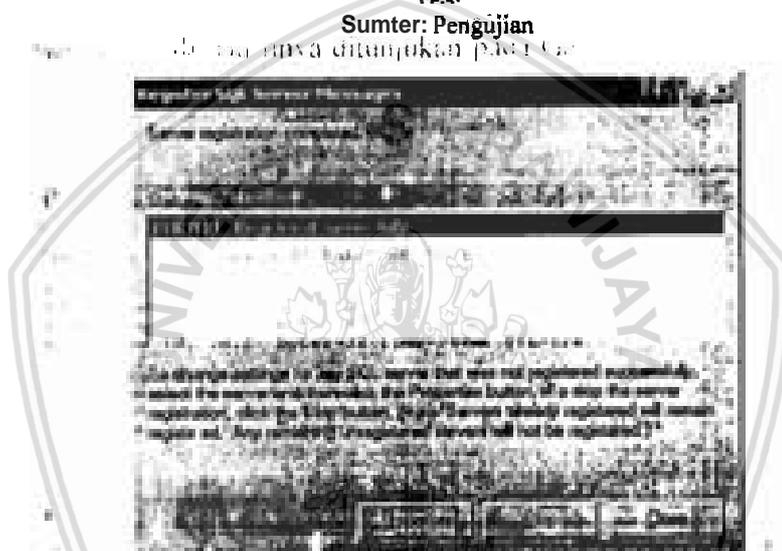
Reply from 10.3.0.2: bytes=12 time=694ms TTL=128
Reply from 10.3.0.2: bytes=32 time=313ms TTL=128
Reply from 10.3.0.2: bytes=32 time=487ms TTL=128
Reply from 10.3.0.2: bytes=32 time=344ms TTL=128

Ping statistics for 10.3.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 313ms, Maximum = 694ms, Average = 414ms

C:\>

```

Gambar 6.33. Tampilan hasil Ping komputer cabang ke IP VPN komputer pusat setelah terjadi VPN

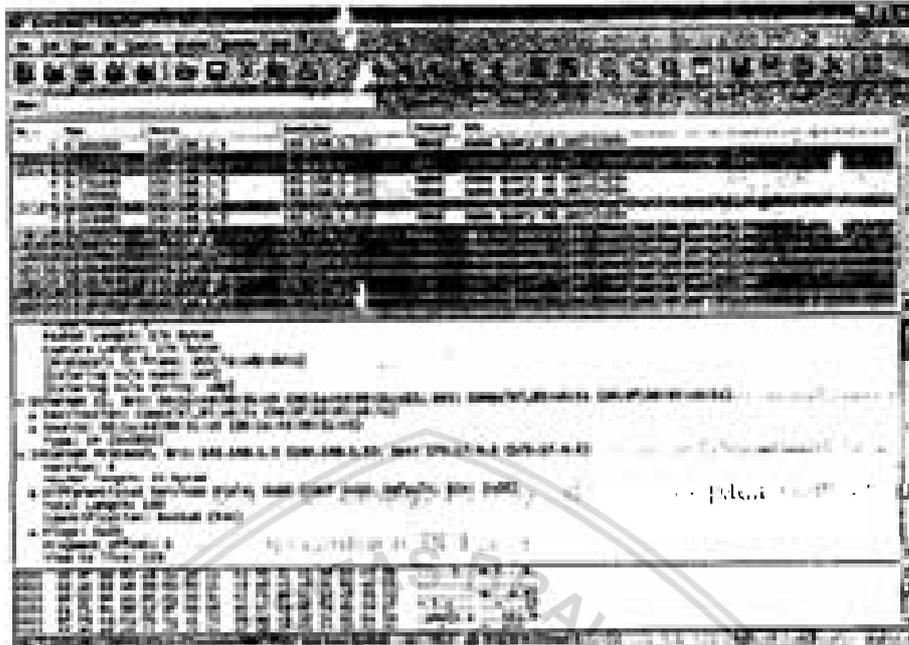


Gambar 6.34. Tampilan pesan rancor setelah registrasi *database*

Sumber: Pengujian

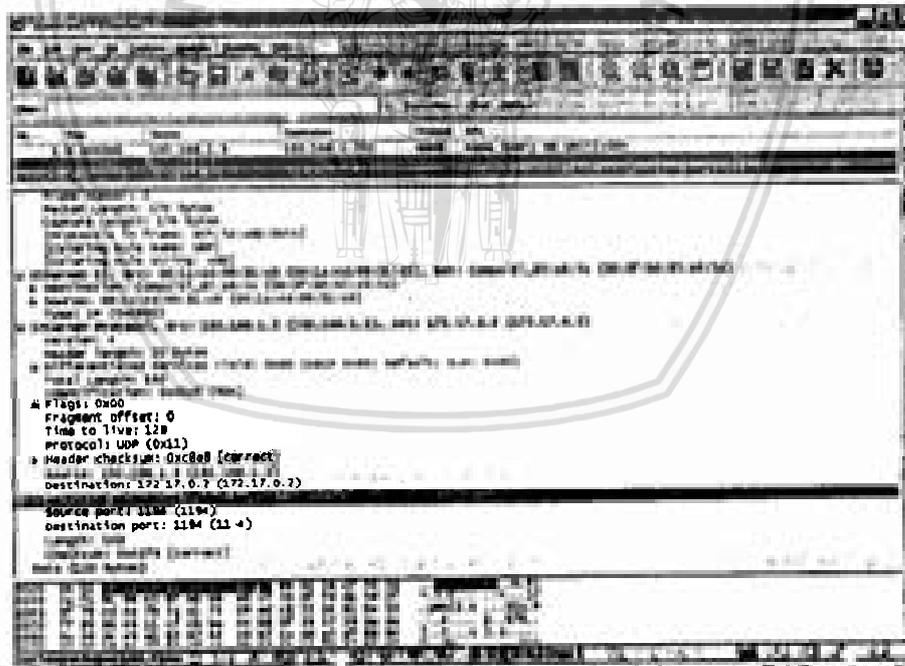
Dari gambar-gambar yang ditunjukkan tadi dapat dibandingkan bahwa pada saat terjadi koneksi Dial Up dari komputer cabang ke komputer RAS, kita tidak dapat melakukan ping IP VPN dan registrasi *database* gagal. Kemudian setelah terjadi koneksi VPN Ping ke IP VPN dapat dilakukan dan registrasi *database* berhasil. Setelah membandingkan kita lakukan replikasi *database* dan kita akan melakukan *capture* paket-paket data, jenis protokol yang digunakan dan port yang digunakan oleh komputer RAS dimana kita dapat menilai jalur VPN yang kita gunakan benar-benar aman dari komputer lain dengan parameter komputer lain tersebut tidak dapat mendeteksi paket-paket data, IP yang digunakan untuk replikasi, jenis protokol yang digunakan dan port yang digunakan. Berikut hasil *capture* ethereal yang digunakan untuk menganalisa paket data

pada komputer RAS dan komputer cabang. Hasilnya ditunjukkan pada Gambar 6.35, Gambar 6.36., Gambar 6.37, dan Gambar 6.38. berikut :



Gambar 6.35. Tampilan hasil capture etersal pada komputer RAS saat replikasi database

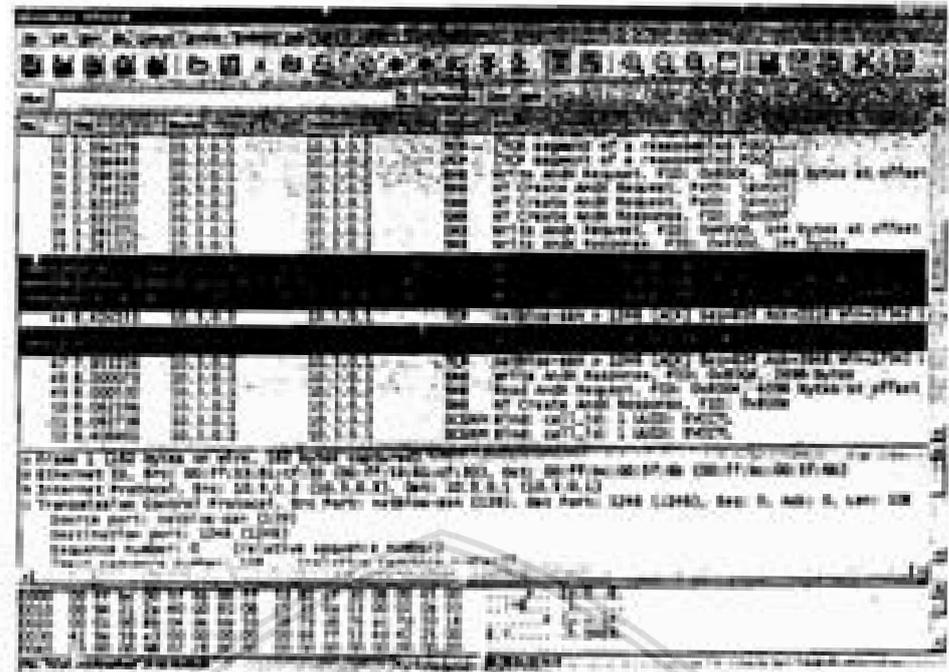
Sumter: Pengujian



Gambar 6.36. Tampilan detail-detail hasil capture etersal pada komputer RAS saat replikasi database

Sumter: Pengujian





Gambar 6.37. Tampilan hasil capture ethereal pada komputer cabang saat replikasi database
Sumber: Pengujian



Gambar 6.38. Tampilan detail-detail hasil capture ethereal pada komputer cabang saat replikasi database
Sumber: Pengujian

Dari keempat gambar diatas sudah dapat menunjukan bahwa paket-paket data yang dikirimkan oleh komputer cabang menuju komputer pusat melalui VPN, tidak dapat terbaca oleh komputer RAS baik dari sisi IP maupun port yang digunakan. Pada

Gambar 6.35. dan Gambar 6.36. dapat dilihat pada saat terjadi replikasi basis data, *capture ethereal* pada komputer RAS banyak terjadi koneksi antara komputer cabang (172.17.0.2) dan komputer pusat (192.168.1.3) dimana terjadi hubungan menggunakan protokol UDP pada port 1194, Kemudian pada Gambar 6.37. dan Gambar 6.38. pada waktu yang sama pada komputer cabang terjadi koneksi menggunakan protokol TCP yang berarti telah terjadi komunikasi transfer data antara IP VPN komputer pusat (172.17.0.2) dan IP VPN komputer cabang' (172.17.0.1). Dan kedua IP VPN tidak terlihat dalam hasil *capture* paket data pada komputer RAS yang merupakan komputer, yang pasti dilewati paket data replikasi. Jadi kesimpulanya data yang ditransfer dari komputer cabang dan komputer pusat menurut analisis paket data bisa dianggap aman.

f. Kesimpulan

Dari hasil pengujian diatas dapat disimpulkan bahwa dengan menggunakan koneksi VPN, untuk melakukan registrasi database tidak perlu dilakukan pengesetan IP dan port yang akan digunakan untuk melakukan koneksi database-nya. Jadi jika kita melakukan koneksi database, VPN secara otomatis telah menyediakan port khusus yang digunakan untuk melakukan koneksi dan replikasi database. Kemudian VPN juga menjamin keamanan dalam penyampaian data. Hal tersebut dapat dilihat dari hasil *capture* ethereal diatas, dimana program ethereal berfungsi sebagai *network* analisis. Dari hasil pengujian analisa paket data pada komputer RAS tadi memperlihatkan bahwa, paket-paket data yang melalui jaringan VPN tidak dapat terdeteksi oleh ethereal pada komputer RAS.

6.5. Pengujian Analisis Dari Beberapa Parameter dan Keunggulanya

Pengujian ini dilakukan untuk mengetahui parameter-parameter apa saja yang menjadi tolak ukur dalam percobaan replikasi basis data menggunakan VPN ini.

a. Tujuan

- Mengetahui besar *bandwidth*, besar paket data, enkripsi dan besar basis data yang digunakan pada masing-masing komputer yang digunakan dalam percobaan.
- Mengetahui keunggulan penggunaan VPN dengan membandingkan hasil analisis pembacaan *sniffing* ethereal dan *Iperf* sebagai tool untuk mengetahui besar *bandwidth* yang digunakan untuk melakukan koneksi database.

b. Spesifikasi dan Konfigurasi Komputer

- Empat buah komputer, komputer pertama dan kedua dijadikan sebagai komputer cabang dengan IP 172.17.0.2 dan 172.17.0.3 . komputer ketiga dijadikan komputer server RAS dengan IP PPP 172.17.0.1 dan IP Lokal 192.168.1.1 sedangkan komputer keempat dijadikan komputer pusat dengan IP 192.168.1.4. dengan gateway 192.168.1.1.
- Komputer pusat: Prosesor Intel Pentium 4 - 2,26 GHz, memori 512 MB.
- Komputer server RAS : Processor Intel Centrino - 1.8 GHz, memori 512 MB
- Komputer cabang: Prosesor Intel Pentium Dual Core - @1.6 GHz, memori 512 MB.
- Sistem Operasi Microsoft Windows 2000 Server dan Windows XP.

c. Software Aplikasi

- Server database SQL Server 2000.
- Open VPN 2.0.9.
- Software network analysis Ethereal.
- Software pengukur performansi jaringan Iperf.

d. Prosedur Pengujian

- Menjalankan perintah “iperf -c 10.3.0.2” pada komputer cabang dan “iperf -s” pada komputer pusat untuk uji performansi VPN.
- Menjalankan perintah “iperf -c 192.168.1.3” pada komputer cabang dan “iperf -s” pada komputer pusat untuk uji performansi jaringan lokal.

e. Hasil Pengujian

Hasil pengukuran performansi bandwidth pada koneksi antara IP Lokal dan IP VPN dapat dilihat pada gambar 6.39 dan 6.40 berikut :

```

C:\WINDOWS\system32\cmd.exe
C:\>ipconfig

ipconfig ~ 10.3.0.1
Client connecting to 10.3.0.1, TCP port 5001
TCP window size: 8.00 KByte (default)

[140] local 10.3.0.1 port 5001 connected with 10.3.0.1 port 1045
[140] interval: Transfer: Bandwidth
[144] 0.0-79.9 sec 24.0 KBytes 2.51 Kbits/sec

ipconfig ~ 192.168.1.3
Client connecting to 192.168.1.3, TCP port 5001
TCP window size: 8.00 KByte (default)

[140] local 192.17.8.2 port 5001 connected with 192.168.1.3 port 5001
[140] interval: Transfer: Bandwidth
[144] 0.0-10.0 sec 168 KBytes 132 Kbits/sec
    
```

Gambar 6.39. Tampilan hasil pengukuran menggunakan *lool/iperf* pada komputer cabang
 Sumber: Pengujian

```

C:\WINDOWS\system32\cmd.exe
F:\MASTER>iperf -s

Server listening on TCP port 5001
TCP window size: 8.00 KByte (default)

[176] local 10.3.0.1 port 5001 connected with 10.3.0.1 port 1045
[176] interval: Transfer: Bandwidth
[176] 0.0-77.9 sec 24.0 KBytes 2.51 Kbits/sec

F:\MASTER>iperf -s

Server listening on TCP port 5001
TCP window size: 8.00 KByte (default)

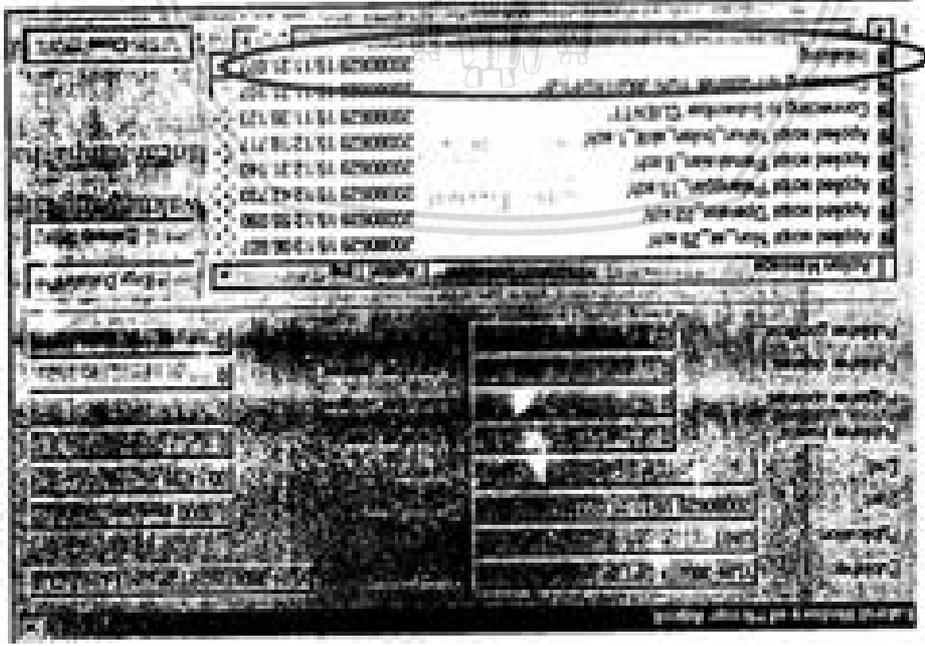
[176] local 192.168.1.3 port 5001 connected with 172.17.8.2 port 1046
[176] interval: Transfer: Bandwidth
[176] 0.0-10.4 sec 168 KBytes 132 Kbits/sec
    
```

Gambar 6.40. Tampilan hasil pengukuran menggunakan *tool Iperf* pada komputer pusat
 Sumber: Pengujian

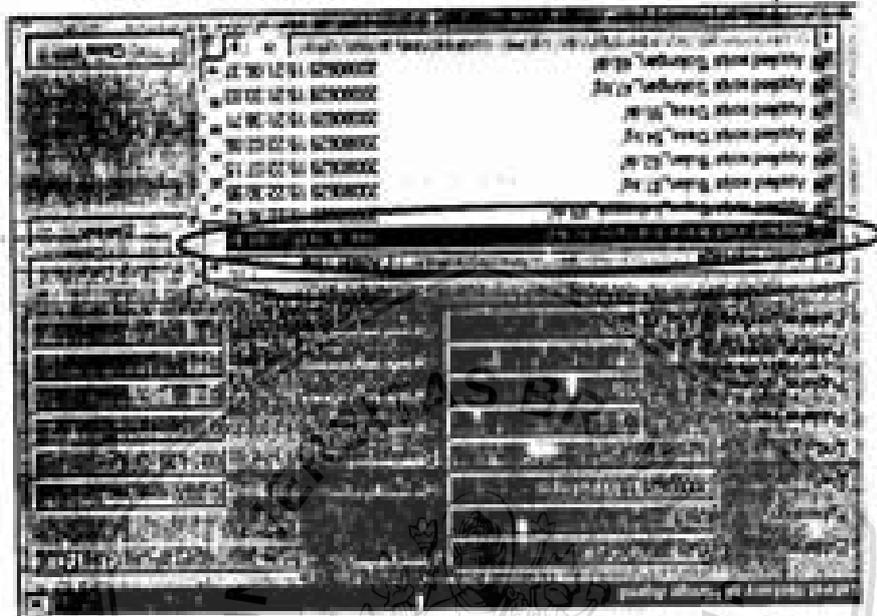
dilihat dari gambar 6.39 dan 6.40 dapat dilihat **performansi** koneksi pada IP lokal dan IP VPN pada saat koneksi VPN terjadi. Bandwidth pada IP lokal terlihat tinggi karena terjadi koneksi VPN pada masing-masing komputer yang **bersifat point to point**. Jadi **bandwidth** yang digunakan merupakan gabungan antara koneksi VPN dan jaringan lokal. Sedangkan IP VPN terlihat rendah karena belum terjadi **replikasi** antara kedua komputer tersebut. Kemudian untuk **transfer** data juga tinggi pada IP lokal, karena penggunaan VPN tersebut maka pada IP lokal **terhitung gabungan** antara 2 koneksi.

Dilihat dari **paramater ukuran database** yang digunakan pada percobaan replikasi pada komputer pusat memiliki **kapasitas lebih besar** karena menampung semua data pelanggan dari berbagai cabang. **Besarnya** basis data dilihat dari jumlah pelanggan dan jumlah transaksi yang terjadi pada cabang-cabang. Untuk pengukuran kecepatan pengiriman data telah dilakukan dengan *tool* pada MS SQL Server 2000 yaitu dengan

melihat *history* dari sesi-sesi replikasi. Berikut adalah snapshot waktu yang digunakan untuk melakukan replikasi pada saat traffic berjalan dengan lancar tanpa hambatan dapat dilihat pada gambar 6.41 dan 6.42 :

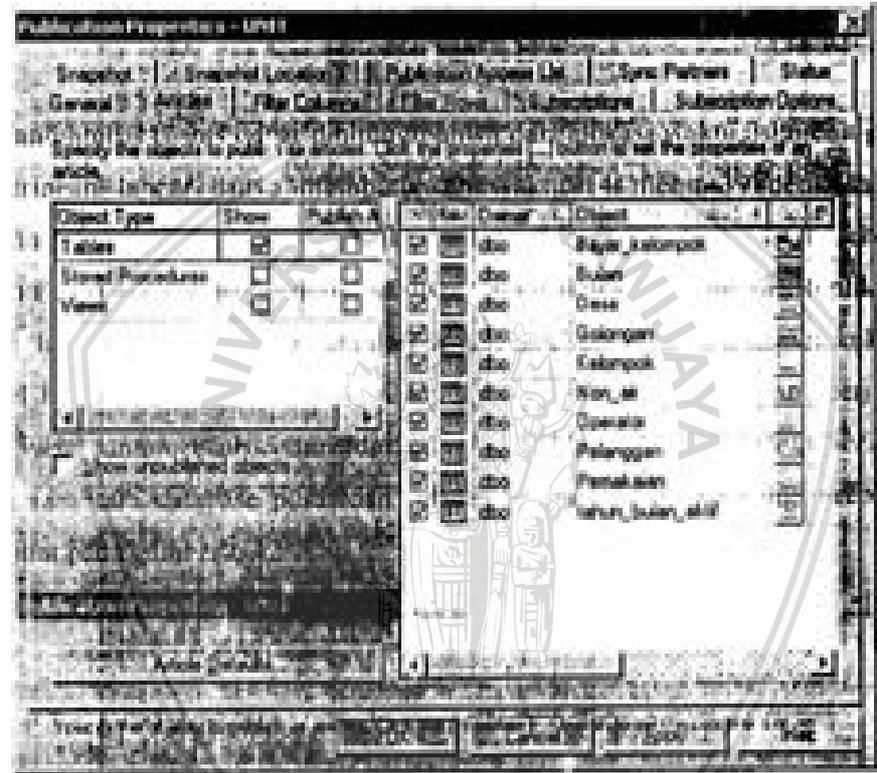


Gambar 6.41. Tampilan pengisian waktu saat memulainya replikasi



Gambar 6.42. Tampilan pengisian waktu saat akhir dari replikasi

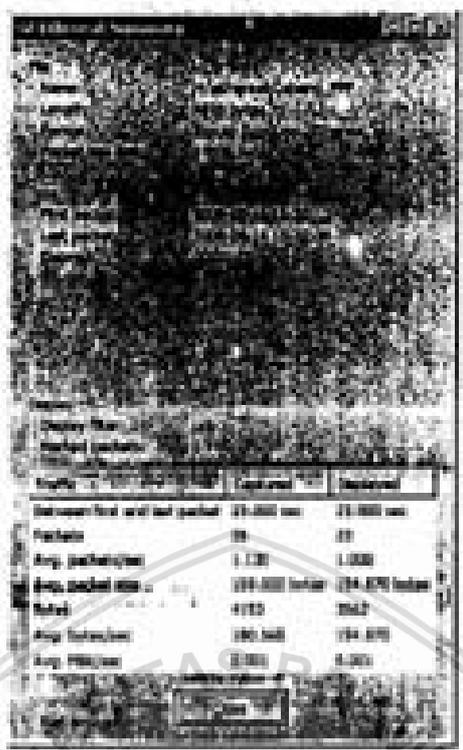
Dari kedua gambar diatas dapat dihitung bahwa dari alokasi waktu 30 menit pada saat traffic jaringan lancar hanya membutuhkan waktu 14 menit 39 detik saja untuk melakukan replikasi. Jika terjadi konflik pada saat replikasi maka CQL server akan melakukan inisialiasi terus menerus sampai terjadi replikasi kembali. Replikasi yang dilakukan hanyalah sebatas data-data dari semua tabel dan jikalau kita melakukan suatu join tabel maka akan termasuk dalam replikasi ini. Jadi replikasi yang terjadi tidak meliputi view dan store procedure dimana Query dari view dan store procedure-nya sudah sama antara database pusat dan cabang. Berikut data-data yang direplikasikan dapat dilihat pada gambar 6.42



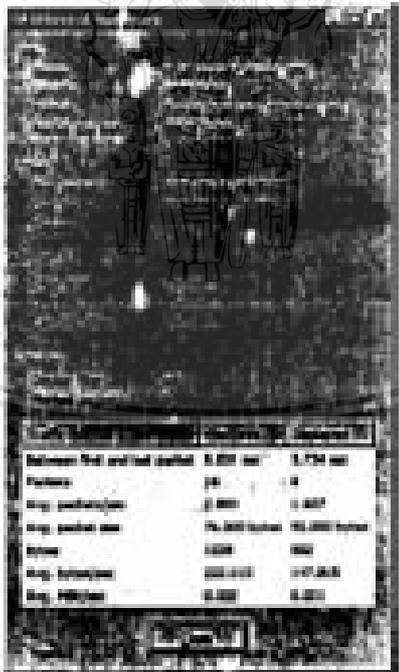
Gambar 6.43. Tampilan tabel-tabel yang direplikasikan .
 Sumber: Pengujian

Dari gambar 6.42 menunjukkan ada 10 tabel yang akan direplikasikan antara cabang dan pusat.

Dalam pembacaan ethereal dapat dibandingkan menjadi beberapa parameter keunggulan dari penggunaan VPN. Hasil dari capture dan summary dapat dilihat pada gambar 6.43 dan 6.44 berikut :



Gambar 6.44. Tampilan hasil *capture* pada saat belum terjadi VPN
Sumber: Pengujian



Gambnr 6.45. Tampilan hasil *capture* pada saat rerjadi VPN
Sumber: Pengujian

Dari gambar 6.43 dan 6.44 dilakukan perbandingan pada protokol UDP-nya, karena pada saat belum terjadi VPN tidal, ada penggunaan protokol TCP pada IP lokalnya. Jadi



dilakukan perbandingan pada sisi protokol UDP-nya. Berikut adalah hasil analisis dari pengujian dilihat dari beberapa parameter :

1) Parameter 1 adalah **Enkripsi**. Perbedaan antara output pada saat tanpa VPN dengan VPN adalah terletak pada ada tidaknya *username* dan *password* suatu user pemakai aplikasi VPN untuk dilihat dengan pihak lain. Pada hasil capture paket data pada waktu percobaan belum bisa ditemukan analisis yang menunjukkan adanya enkripsi. Tetapi penggunaan password digunakan pada sisi VPN-nya sekaligus pada aplikasi SQL servernya, hanya analisis pencarian yang belum ditemukan.

2) Parameter 2 adalah **ukuran total file**. Ukuran rata-rata total paket pada saat VPN yang melalui filter adalah 76 KB, sedangkan ukuran rata-rata total paket pada saat tanpa VPN yang melalui filter adalah 159 KB.

3) Parameter 3 adalah **jumlah file**. Jumlah file rata-rata yang melalui filter pada saat VPN adalah 16 buah paket, sedangkan jumlah file rata-rata yang melalui filter tanpa VPN adalah 26 buah paket.

4) Parameter 4, **tanpa algoritma kompresi dilihat** saat terkoneksi VPN. Pada percobaan ini tidak digunakan kompresi LZOP pada VPN-nya, jadi tidak bisa dilakukan perbandingan pada saat terjadi kompresi data. Kompresi yang dilakukan yaitu pada aplikasi MS SQL server saja.

g. Kesimpulan

Dari hasil pengujian diatas dapat disimpulkan bahwa dengan menggunakan koneksi VPN ada beberapa **keunggulan**, untuk parameter penggunaan bandwidth jika kita menggunakan VPN akan memerlukan **bandwidth** besar, karena adanya **enkapsulasi** paket data yang berlapis menambah kerja tiap layer OSI untuk **mendekripsikan** paket data yang **terenkapsulasi** tersebut. Seperti penggunaan **bandwidth** yang tinggi, Pada parameter transfer paket data juga menjadi **sangat** besar untuk **pengiriman** paket datanya. Karena dengan enkapsulasi yang berlapis akan **menambah** besar paket data yang akan dikirim. Untuk **menanggulangi** itu ada **pengkompresian** data pada Microsoft

SQL server dan penggunaan algoritma kompresi LZO (*on abbreviation for Lempel-Ziv-Oberhumer*) pada OpenVPN.

Pada parameter ukuran basis data yang digunakan, besarnya tergantung jumlah pelanggan dan transaksi yang terjadi pada suatu cabang. Penghitungan performansi basis data dilakukan dengan menghitung waktu yang digunakan untuk melakukan semua operasi replikasi dari alokasi waktu yang disediakan.

Jadi keunggulan penggunaan VPN pada percobaan ini adalah :

- 1) Karena koneksi VPN yang bersifat *Point-to-point*, registrasi basis data akan berlangsung secara otomatis tanpa perlu melakukan pengesetan IP pada masing-masing komputer.
- 2) Data yang terkirim akan aman dengan menggunakan VPN, karena adanya enkripsi dan deskripsi paket data.
- 3) Penggunaan OpenVPN yang memiliki algoritma kompresi LZO dapat meminimalisasi penggunaan bandwidth dan besarnya paket data yang akan dikirim.
- 4) Pada alokasi waktu 30 menit hanya membutuhkan waktu rata-rata 15 menit untuk melakukan replikasi pada saat lalu lintas jaringan lancar.



BAB VII PENUTUP

6.1. KESIMPULAN

Hasil penelitian ini berdasarkan perancangan, implementasi dan pengujian yang telah dilakukan maka dapat disimpulkan sebagai berikut:

1. Sesuai dengan perumusan masalah, otomatisasi koneksi VPN yang dilakukan untuk melakukan replikasi terjadwal setelah transaksi selesai, merupakan salah satu cara untuk mengoptimalisasi waktu dan biaya supaya pendistribusiar. basis data antara cabang dan pusat dapat terlaksana dengan baik.
2. Penggunaan VPN sebagai solusi koneksi, merupakan salah satu metode alternatif dalam pengkoneksian database. Karena dengan VPN, paket-paket data yang tersinkronisasi antara pusat dan cabang menjadi aman karena adanya proses enkripsi dan deskripsi. Dan juga dengan penggunaan VPN ini dalam proses registrasi database dapat dilakukan langsung tanpa harus ada pengesetan IP database yang dituju karena prosesnya yang *point to point*.
3. Penggunaan OpenVPN memungkinkan adanya pengkompresian data yang akan dikirim dan dengan adanya fasilitas kompresi data ini akan mengurangi penggunaan bandwidth dan juga didukung pula dengan fasilitas pada Microsoft SQL Server yang dapat meminimalisasi besar data yang akan dipublikasikan.

6.2. SARAN

1. Replikasi basis data melalui jaringan VPN dapat dikembangkan dengan adanya suatu mekanisme untuk menentukan lama koneksi yang dilakukan masing-masing unit berdasarkan besar data yang ditransfer, bukan berdasarkan rentang waktu tertentu.
2. Apabila tidak ada perubahan data pada komputer unit, maka proses koneksi VPN dan replikasi yang telah dijadwalkan tidak perlu dilakukan lagi.
3. Replikasi basis data melalui jaringan VPN dapat ditambahkan mekanisme pengamanan data terhadap program Microsoft Access



sehingga data maupun desain dari aplikasi tidak dapat dirubah oleh semua user.



DAFTAR PUSTAKA

- [BER-99] Bernstein, P. A., 1999, "Replication, Transaction Processing Concepts and Techniques": Western Institute for Computer Science at Stanford Univ.
- [CER-85] Ceri & G. Pelagatti, 1985, "Distributed Databases Principles and System", McGraw-Hill, Singapore.
- [ELM-00] Elmsri & Navathe, 2000, "Fundamentals of Databases System," Addison-Wesley, USA
- [FAT-041] Fathansyah, 2004, "Sistem Basis Data Lanjutan Buku Basis data", Penerbit Informatika, Bandung.
- [GUN-00] Gunderloy, M & Jorden, J.L, 2000, "Mastering SQL Server 2000", Sybex, San Francisco.
- [HEN-01] Hendandar, 2001, "Implementasi Replikasi Basis Data Mahasiswa Universitas Gajah Mada", Skripsi Teknik Elektro Universitas Gajah Mada, Yogyakarta
- [LIN-01] Linsenbart, M & Stigler, S., 2001, "SQL SERVER 2000 Administration", Osborne/McGraw Hill, California, USA.
- [MED-98] Medi, 1998, "Rancangan WEB Akademik dengan Basis Data Terdistribusi", Thesis Ilmu Komputer Universitas Gajah Mada, Yogyakarta.
- [SUK-04] Sukmawan, 2004, "Optimasi Replikasi Data Sistem Informasi Terdistribusi (Studi Kasus di Lembaga Pendidikan Primagama)", Thesis Ilmu Komputer Universitas Gajah Mada, Yogyakarta.
- [OZS-99] Ozsu, M. T. & P. Valduriez, 1999, "Principles of Distributed Database System", Second Edition, Prentice Hall, Boston.
- [PET-00] Petkovic, D., 2000, "SQL SERVER 2000 A Beginner's Guide", Osborne /McGraw Hill, California, USA.,
- [RAM-011] Ramalho, J., 2001, "SQL Server, 7.0", Elex Media Komputindo, Jakarta
- Ramakrishnan, R. & J. Gehrke, 2000, "Database Management System" Second Edition, Mc Graw-Hill, Boston.
- [SIL-97] Silberschatz, H.F.Korth, S.Sudarshan, 1997, "Databases System Concepts", Fourth Edition, McGraw-Hill Companies, New York
- [MAN-021] Mansfield, N., 2002, "Practical Tcp/Ip, Designing, Using And Troubleshooting Tcp/Ip On Linux® And Windows Networks®", Pearson



Education, Inc

- [KHO-05] Khotimah, H. 2005, "Implementasi Replikasi Basis Data Melalui Jaringan Telepon Pada PDAM Kabupaten Malang", Malang
- [KEN-03] Kendall, Kenneth, E & Kendall, Julie, E. 2003. *Analisis dan Perancangan Sistem*. PT Prenhallindo. Jakarta.
- [FAT-02] Fathansyah. 2002. *Basis Data. Informatika. Bandung*.
- [ANO-00] Anonymous, 2000. *Replication*.
- [PUR-01] Purbo, Onno W. 2001. *TCP/IP*. Elex Media Komputindo. Jakarta.
- [AMR-01] Amri, M. Choirul. 2003. *Cepat Mahir Windows 2000 Server*. IlmuKomputer.Com
Akses dari: <http://ilmukomputer.com/berseri/choirul-win2000server/index.php>
Tanggal akses: 18 Mei 200
- [STA-01] Stallings, William. 2001. *Dasar-dasar Komunikasi Data Salemba Teknik*.
- [ANO-07] Anonymous. *Praktikum Jaringan Komputer 2 : Open VPN*
Akses dari: <http://lecturer.eepis-its.edu/~dhotokuliah/prak-jarkom2/jarkom2%20-%205.%20OpenVPN.pdf>
Tanggal akses : 22 November 2007
- [ANA-07] Anonymous. *OpenVPN™ 2.0 HOWTO*.
Akses dari : <http://openvpn.net/howto.html>
Tanggal akses : 22 November 2007
- [ANB-07] Anonymous. *OpenVPN™ 2.0 on Windows notes*.
Akses dari : <http://openvpn.net/INSTALL-win32.html>
Tanggal akses : 22 November 2007
- [ANC-07] Anonymous. *OpenVPN™ and the SSL VPN Revolution*.
Akses dari : http://www.sans.org/reading_room/whitepapers/vpns/1459.php
Tanggal akses : 22 November 2007
- [MAR-06] Feilner, Markus. 2006. *OpenVPN, Building and Integrating Virtual Private Networks*. Birmingham. PACKT Publishing
- [HER-04] Prayitno, Heri. 2004. *Tugas Jaringan Komputer*.

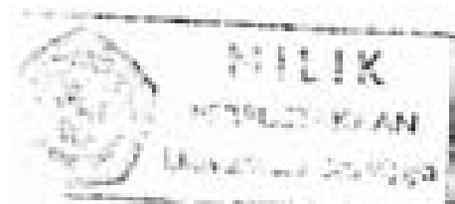
DAFTAR RIWAYAT HIDUP

1. Nama Lengkap dan Gelar : Ir. Heru Nunvarsito, M.Kom
2. Jenis kelamin : Laki-laki
3. Fakultas/Jurusan : Teknik/Teknik Elektro
4. Pekerjaan/Jabatan sekarang : Dosen/Lektor Kepala
5. NIP : 131 879033
6. Pangkat/Golongan : Pembina/IV-a
7. Bidang keahlian : Sistem Informasi, Komputer
8. Pengalaman Penelitian:
- Rekayasa antamuka Transmisi infra merah penghubung komputer dan printer, 1998
 - Rekayasa Pengontrol telepon dengan IBM PC, 1999
 - Perancangan Audiometer Sebagai Peralatan Survei Pendengar Radio, 1999
 - Simulasi Dan Analisis Rangkaian Elektronika Dengan Metode Nodal, 2000
 - Rekayasa Pencatat Telepon Lokal Dengan Mikrokontroler Intel 8031, 2000
 - Pencatat Pulsa Pemakaian Telepon Pada Rumah Pondokan, 2000
 - Rekayasa Emulator Mikrokontroler Intel 8752, 2001
 - Analisis performansi proxy server dengan simulasi, 2005

Malang, 20 April 2007

Ketua Peneliti,

Ir. Heru Nunvarsito, M.Kom



DAFTAR RIWAYAT HIDUP

1100169

1. Nama Lengkap : Ir. Heri **Prayitno**
2. **Tempat** dan tgl. Lahir : Bondowoso, 10 September 1968
3. **Alamat** : Jl. **Tondano** Dalam II A2/C22, Sawojajar Malang
Telp. 0341-716385, HP. 081 555 679310
4. **Jenis kelamin** : **Laki-laki**
5. Bidang **keahlian** : Programmer Komputer.
6. Pengalaman Penelitian
 - Rekayasa **Instrumen** Pembangkit Listrik Tenaga **Mikrohidro**, 1997.
 - Pengolahan Sistem database PDAM Malang, 1995
 - Pemrograman **Simulasi** Dan **Analisis** Sistem **Elektrik** Dengan **Metode** Neural Network, 2000
7. Pengalaman Pekerjaan :
 - Kepala Lab. Sistem Informasi, 2002-2005
 - Staf Ahli bidang Komputer, 1992-sekarang, pada bidang:
 - Programming
 - Jaringan Komputer
 - Tim **Pembuatan** dan Pengembangan **Software** Billing System dan Akuntansi RS **Syuhada** Haji
- 10: **Pendidikan**:
 - **S1** Teknik **Elektro** Universitas **Brawijaya**, 1991
 - **S2** Teknik **Elektro/Informatika** UGM

Malang, 20 April 2007

Peneliti,

(Ir. Heri Prayitno)

