23.159/H/05

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

PERANCANGAN DAN PEMBUATAN SISTEM MONITORING RUMAH MENGGUNAKAN WEBCAM DENGAN AKTIFASI DIAL-UP INTERNET MELALUI SMS

TUGAS AKHIR

RSIf
004.678
Bis
P-1
2005

PERPUSTAKAAN
I T S
Tgl. Terima | 5-8-2005
Terima dari | H
No. Agenda Prp. | 222899

Disusun Oleh :

JANUAR BISAPTANTO
NRP. 5100 100 064

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2005

DEPARTEMEN PENDIDIKAN NASIONAL
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
FAKULTAS TEKNOLOGI INFORMASI
JURUSAN
TEKNIK INFORMATIKA

PERANCANGAN DAN PEMBUATAN SISTEM MONITORING RUMAH MENGGUNAKAN WEBCAM DENGAN AKTIFASI DIAL-UP INTERNET MELALUI SMS

TUGAS AKHIR

Diajukan untuk Memenuhi Sebagian Persyaratan

Memperoleh Gelar Sarjana Komputer

Pada

Jurusan Teknik Informatika

Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember

Surabaya

Mengetahui / Menyetujui

Dosen Pembimbing I	Dosen Pembimbing II
Muchammad Husni, M.Kom	Royyana Muslim I, S.Kom.
NIP. 131411100	

SURABAYA

Juli 2005

ABSTRAK

Internet merupakan produk teknologi yang berkembang pesat dan menjadi dasar terhadap munculnya berbagai macam teknologi dan peralatan multimedia yang ada sekarang ini. Salah satunya adalah SMS gateway dan Webcam. SMS gateway, produk yag didukung oleh teknologi mobile phone ini adalah untuk berkomunikasi antara komputer dengan telepon selular. Sedangkan Webcam adalah salah satu produk teknologi multimedia untuk pengambilan gambar video atau gambar.

Tugas akhir ini mencoba mengimplementasikan sebuah sistem untuk memonitor rumah dari jauh dengan mengintegerasikan teknologi SMS Gateway, Webcam dan Dial Up Networking untuk menunjang fitur sistem. Teknis secara global sistem adalah sebagai berikut: Sistem diaktifkan menggunakan SMS. Aktifasi yang dimaksud adalah dial up komputer server ke internet. Output aktifasi adalah alamat IP yang dikirimkan kembali ke user. Informasi alamat IP inilah yang digunakan komputer client untuk memulai viewing gambar dari webcam.

Sistem yang dibuat dalam tugas akhir ini dapat melakukan viewing kondisi rumah dari tempat lain melalui internet. Koneksi ke internet komputer server dapat dikontrol menggunakan SMS sehingga mengefisienkan penggunaan internet pada sistem.

KATA PENGANTAR

Segala Puji Syukur bagi Allah SWT, karena atas limpahan karunia dan perlindungan-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul:

PERANCANGAN DAN PEMBUATAN SISTEM MONITORING RUMAH MENGGUNAKAN WEBCAM DENGAN AKTIFASI DIAL-UP INTERNET MELALUI SMS

Tugas Akhir ini dibuat guna memenuhi persyaratan akademik dalam rangka ujian akhir bagi mahasiswa Strata 1 (S1) Jurusan Teknik Informatika , Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Dalam Tugas Akhir ini penulis mengembangkan sebuah sistem untuk memonitor keadaan rumah dengan menggunakan *webcam* dengan aktifasi *dial-up* internet pada komputer server melalui SMS.

Pada kesempatan ini penulis menyampaikan penghargaan dan rasa terima kasih yang sebesar – besarnya atas bimbingan, doa, serta bantuan yang diberikan kepada penulis sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan baik, khususnya kepada :

1. Ibunda tercinta, Hendar Yulies A. yang senantiasa memberi semangat, dukungan, dan doa kepada penulis.
2. Ayahanda Djoko Walujo (Alm) yang selalu menjadi inspirasi penulis untuk belajar.
3. Saudara-saudaraku tercinta Febi Wineko, Wulaneka Dwisaptantri dan Dimas Hendarin AH atas segala dukungan dan doa yang diberikan.

4. Bapak Muchamad Husni, M.Kom, selaku dosen pembimbing I dan kepala Lab. AJK, yang telah memberikan banyak bimbingan selama pembuatan Tugas Akhir serta pengayoman selama kuliah.
5. Bapak Royyana Muslim I, S.Kom., selaku dosen pembimbing II, yang telah memberikan banyak bimbingan selama pembuatan Tugas Akhir.
6. Bapak Ir. Suhadi Lili, S.Kom, selaku Dosen Wali penulis yang telah memberikan dukungan selama pembuatan Tugas Akhir.
7. Bapak Nunut Priyo Jatmiko, selaku Koordinator Lab AJK yang telah banyak memberikan bantuan dan pengayoman selama kuliah.
8. Mas Ahmad Suaidi yang membantu penulis menemukan jalan keluar setiap masalah dalam pengerjaan Tugas Akhir ini.
9. Saudara seperjuanganku AJK, Dwi Arie, Prevan, Roy, Putu Arie, Alifah, Dian, Surya TW, Delis, Tyarso, Victor, Adit, Yoyok, Ronie, Sokam, Pras, Deka, Willy, Rudy, Galih, Wafa, Jian, Husin, Nando, serta seluruh kru AJK yang tidak dapat disebutkan satu persatu, terima kasih telah menemani penulis dalam pengerjaan Tugas Akhir ini.
10. Kakak – kakak seniorku AJK, Guruh, Paimo, Andi KI, Putu, Roni, Mang Idi, Arydewo, Gusmul, Purna, Tiyok, Damen, terima kasih atas semangat dan bantuan yang diberikan kepada penulis.
11. Semua rekan – rekan C10 dan seluruh mahasiswa Teknik Informatika ITS Surabaya.Terima kasih atas tahun-tahun indah yang telah kita lalui bersama.
12. Para dosen yang telah banyak memberi bimbingan kepada penulis selama berada di ITS.

13. Para karyawan yang telah membantu penulis selama berada di ITS.
14. Rekan-rekan tim FC Informatika yang menemani penulis menghilangkan rasa penat dengan berolahraga.
15. Semua yang menemani voli tiap sore hari. Mari kita berharap lapangan tercinta kita tetap dipertahankan.
16. Semua pihak yang tidak dapat disebutkan satu per satu yang telah mendukung penulis dalam menyelesaikam Tugas Akhir ini.

Akhir kata penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan., semoga Tugas Akhir ini dapat memberikan manfaat bagi kita semua.

Surabaya, Juli 2005

Januar Bisaptanto

DAFTAR ISI

ABSTRAK ii
KATA PENGANTAR iii
DAFTAR ISI vi
DAFTAR GAMBAR ix
DAFTAR TABEL xi
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Tujuan dan Manfaat 2
1.3 Perumusan Masalah 2
1.4 Pembatasan Masalah 3
1.5 Metodologi Pembuatan Tugas Akhir 3
1.6 Sistematika Pembahasan 4
BAB II DASAR TEORI 6
2.1 Teknologi Internet 6
2.1.1 TCP/IP 6
2.1.2 Socket 8
2.1.3 Dial-up Networking 16
2.1.3.1 Dial up Internet Service Provider (DISP) 18
2.1.3.2 Windows Internet Service (WinInet) 18
2.2 Teknologi Webcam 21
2.2.1 Metode Viewing pada Webcam 21
2.2.2 Image Capturing 22
2.2.2.1 Avicap32 Library 22
2.2.2.2 User32 Library 25
2.2.2.3 Python Image Library 27
2.2.3 Kompresi JPEG 27
2.3 Teknologi SMS (Short Message Service) 30
2.3.1 Format PDU 30
2.3.1.1 Menerima sebuah pesan pada mode PDU 31
2.3.1.2 Menginterpretasikan 8-bit oktet menjadi pesan 7-bit 34
2.3.1.3 Mengirim sebuah pesan pada mode PDU 34

2.3.2 AT COMMAND 37
BAB III PERANCANGAN PERANGKAT LUNAK 42
3.1 Deskripsi Umum 42
3.1.1 Arsitektur Sistem 43
3.1.2 Fitur yang dimiliki Aplikasi 44
3.2 Perancangan Proses 44
3.2.1 Proses pada Server 45
3.2.1.1 Proses Pengambilan Gambar dari Webcam 45
3.2.1.2 Proses Penerimaan SMS 46
3.2.1.3 Proses Validasi Data dari SMS 46
3.2.1.4 Proses Dial-up Internet 47
3.2.1.5 Proses Pengiriman SMS 48
3.2.1.6 Proses Pengiriman Gambar 49
3.2.2 Proses pada Client 50
3.2.2.1 Proses Konfigurasi 50
3.2.2.2 Proses Penerimaan Gambar 51
3.2.3 Proses Komunikasi Client Server 52
3.3 Perancangan Antar Muka 53
3.3.2 Perancangan Antarmuka Aplikasi Server 53
3.3.2.2 Form Utama Server 53
3.3.2.3 Form Service Socket 54
3.3.2.4 Form Service Kamera 54
3.3.2.5 Form Service SMS Gateway 55
3.3.3 Perancangan Antarmuka Aplikasi Client 56
3.3.3.2 Form Utama Client 56
3.3.3.3 Form Layar Gambar 56
3.3.3.4 Form Konfigurasi 57
BAB IV IMPLEMENTASI PERANGKAT LUNAK 58
4.1. Implementasi Aplikasi Server 58
4.1.1 Implementasi Proses Memulai Socket Server 59
4.1.2 Implementasi Proses Pengambilan Gambar dari Webcam 59
4.1.3 Implementasi Proses Penerimaan SMS 60
4.1.4 Implementasi Proses Validasi Data SMS 62
4.1.5 Implementasi Proses Dial up Internet 62
4.1.6 Implementasi Proses Pengiriman SMS 63
4.1.7 Implementasi Proses Parsing Parameter Client-Server 64
4.1.8 Implementasi Proses Pengiriman Gambar 65
4.2. Implementasi Aplikasi Client 66

4.2.1 Implementasi Proses Koneksi ke Server 67
4.2.2 Implementasi Proses Konfigurasi 67
4.2.3 Implementasi Proses Penerimaan Gambar dari Server 68
BAB V UJI COBA DAN EVALUASI 70
5.1 Deskripsi Hardware dan Software Uji Coba 70
5.2 Skenario Uji Coba 71
5.2.1 Uji Coba Skenario Penerimaan SMS 71
5.2.2 Uji Coba Skenario Dial Up Internet 72
5.2.3 Uji Coba Skenario Pengambilan Gambar dari Webcam 76
5.2.4 Uji Coba Skenario Pengiriman Gambar 78
5.2.4.1 Ujicoba pada LAN 79
5.2.4.2 Ujicoba pada internet 82
BAB VI PENUTUP 87
6.1. Kesimpulan 87
6.2. Saran 88
DAFTAR PUSTAKA 89

DAFTAR GAMBAR

Gambar 2.1 Lapisan pada TCP/IP 6
Gambar 2.2 Pipa komunikasi client server 8
Gambar 2.3 *Port number* 9
Gambar 2.4 Socket di dalam TCP/IP 10
Gambar 2.5 Skenario komunikasi menggunakan socket 11
Gambar 2.6 Skema pengiriman gambar pada Winsock 16
Gambar 2.7 *Dial up* ke internet menggunakan modem 17
Gambar 2.8 Protocol PPP untuk modem-to-modem link 17
Gambar 2.9 TCP/IP untuk setelah Dial up ke Internet 18
Gambar 2.10 JPEG Encoder Block Diagram 28
Gambar 2.11 JPEG Decoder Block Diagram 29
Gambar 2.12 Skema pengambilan *snapshot* gambar 29
Gambar 3.1 Arsitektur Sistem 43
Gambar 3.4 Flowchart Proses Pengambilan Gambar dari Webcam 45
Gambar 3.4 Flowchart Proses Penerimaan SMS 46
Gambar 3.5 Flowchart Proses Validasi Data dari SMS 47
Gambar 3.6 Flowchart Proses Dial-up Internet 48
Gambar 3.7 Flowchart Proses Mengirim SMS 49
Gambar 3.8 Flowchart Proses Pengiriman Gambar 50
Gambar 3.9 Flowchart Proses pada client 51
Gambar 3.10 Flowchart proses komunikasi sisi server 52
Gambar 3.11 Flowchart proses komunikasi sisi client 53
Gambar 3.12 Form utama server 54
Gambar 3.13 Form service socket 54
Gambar 3.14 Form service kamera 55
Gambar 3.15 Form service SMS Gateway 55
Gambar 4.1 Skema pemaketan file gambar 65
Gambar 5.1 Hasil decode SMS masuk di grid 72
Gambar 5.2 Form dialog pada proses dialing 73
Gambar 5.3 Status koneksi berhasil 74
Gambar 5.4 IP dinamik dari DISP pada komputer 74
Gambar 5.5 Status koneksi gagal 74

Gambar 5.6 IP hasil dial up dikirim melalui SMS **75**
Gambar 5.6 Pesan dial up gagal **75**
Gambar 5.7 File *tmp.jpg* pada 0.5 detik pertama **76**
Gambar 5.8 File *tmp.jpg* pada 0.5 detik kedua **77**
Gambar 5.9 File *tmp.jpg* pada 0.5 detik ketiga **77**
Gambar 5.10 File *tmp.jpg* pada 0.5 detik keempat **77**
Gambar 5.12 Form pencatat frame terkirim **79**
Gambar 5.13 Form pencatat frame datang **79**

DAFTAR TABEL

Tabel 2.1 Tabel Property Kontrol Winsock 12
Tabel 2.2 Tabel Method Kontrol Winsock 13
Tabel 2.3 Tabel Even Kontrol Winsock 14
Tabel 2.4 Tabel Produk DUN di Indonesia 18
Tabel 2.5 Tabel string PDU yang diterima 32
Tabel 2.6 Tabel string PDU yang terkirim 34
Tabel 2.7 Tabel Hayes Standard Command 37
Tabel 2.8 Tabel Acknowledge untuk komunikasi data normal 38
Tabel 2.9 Tabel AT Command untuk memeriksa SMS 39
Tabel 2.10 Tabel AT Command untuk mengirim SMS 40
Tabel 2.11 Tabel AT Command untuk menghapus SMS 41
Tabel 5.1 Tabel hasil pengiriman gambar dalam LAN 79
Tabel 5.2 Tabel hasil penerimaan gambar dalam LAN 80
Tabel 5.3 Tabel perhitungan bandwith pada LAN 81
Tabel 5.4 Tabel hasil pengiriman gambar melalui internet 1 83
Tabel 5.5 Tabel ujicoba penerimaan gambar melalui internet 1 83
Tabel 5.6 Tabel perhitungan bandwith pada internet 1 84
Tabel 5.7 Tabel hasil pengiriman gambar melalui internet 2 84
Tabel 5.8 Tabel ujicoba penerimaan gambar melalui internet 2 85
Tabel 5.9 Tabel perhitungan bandwith pada internet 2 85

BAB I

PENDAHULUAN

PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

BAB I

PENDAHULUAN

Dalam bab ini akan akan dijelaskan mengenai latar belakang, permasalahan, serta tujuan dan manfaat dari pembuatan Tugas Akhir ini. Selain itu bab ini juga menjelaskan batasan masalah, metodologi pembuatan Tugas Akhir dan sistematika pembahasan keseluruhan Tugas Akhir.

1.1 Latar Belakang

Teknologi informasi saat ini sudah semakin memiliki peran yang penting dalam banyak bidang. Paradigma lama publik bahwa teknologi informasi hanya memiliki dunianya sendiri berangsur-angsur terkikis seiring dengan banyakanya pengembangan-pengembangan dari teknologi informasi untuk mengatasi permasalahan yang lebih luas.

Salah satu keunggulan teknologi informasi adalah teknologi yang dapat menjangkau seluruh dunia melalui jaringan internet. Hal ini dapat dimanfaatkan untuk menyelesaiakan permasalahan-permasalahan manusia. Salah satu permasalahan yang muncul adalah bagaimana seseorang bisa memantau keadaan rumah ketika dia berada di tempat lain. Dimana informasi yang didapatkan adalah berupa gambar/video dari rumah tersebut pada saat itu.

Dalam menyelesaikan permasalahan tersebut dapat menggunakan keunggulan teknologi informasi. Beberapa fitur teknologi informasi yang

digunakan untuk meyelesaikan ini adalah *webcam*, teknologi internet, dan *SMS Gateway*.

Webcam adalah salah satu produk teknologi multimedia untuk pengambilan gambar video. Teknologi internet penggunaannya sangat luas karena jaringan internet menjangkau seluruh tempat di dunia, penggunaan teknologi internet adalah untuk mendapatkan informasi yang realtime dimana saja. *SMS Gateway*, produk yang didukung oleh teknologi *mobile phone* ini adalah untuk berkomunikasi antara komputer dengan telepon selular. *SMS Gateway* lebih memungkinkan pertukaran informasi dapat lebih personal dan *real time* karena didukung *mobile phone*.

1.2 Tujuan dan Manfaat

Tujuan pembuatan tugas akhir ini adalah untuk membuat sistem untuk memonitor keadaan rumah dengan menggunakan *webcam* dengan aktifasi *dial-up* internet pada komputer server melalui SMS.

1.3 Perumusan Masalah

Permasalahan yang dihadapi dalam pengembangan tugas akhir ini adalah bagaimana membuat sebuah sistem monitoring ini dapat memberikan informasi berupa gambar hasil bidikan *webcam* yang dipasang dirumah. Informasi gambar tersebut harus dapat diakses dimana saja. Permasalahan berikutnya adalah untuk mengurangi biaya dari sistem akibat koneksi terus-menerus ke internet.

1.4 Pembatasan Masalah

Sejumlah permasalahan yang dibahas di dalam Tugas Akhir ini akan dibatasi ruang lingkup pembahasannya, antara lain:

1. Sistem keamanan jaringan seperti firewall dalam sistem ini untuk sementara diabaikan.
2. *Webcam* yang digunakan adalah jenis yang dapat mengambil snapshot gambar dari objek.
3. Handphone yang digunakan harus mendukung teknologi SMS.
4. Protokol yang digunakan adalah TCP/IP.
5. Komputer server dan komputer client memiliki koneksi secara langsung ke internet. Koneksi ke internet menggunakan *Dial-Up Networking*.

1.5 Metodologi Pembuatan Tugas Akhir

Pembuatan tugas akhir ini dilakukan dengan mengikuti metodologi sebagai berikut:

1. Studi literatur dan software yang ada

   Pada tahap ini akan dipelajari sejumlah literatur mengenai konsep dan teknologi yang akan digunakan. Informasi yang dibutuhkan dapat diperoleh dari literatur yang berhubungan dengan teknologi *webcam*, internet dan *SMS (Short Message Service)*. Informasi ini dapat meliputi buku referensi, majalah dan dokumentasi internet.

2. Perancangan Sistem dan Aplikasi

- Pemodelan dan perancangan sistem

  Pada tahap ini akan dibuat rancangan sistem serta elemen-elemen yang ada di dalam sistem dan alur kerja dari elemen-elemen tersebut.

- Perancangan Antarmuka (*Interface*)

  Pada tahap ini akan dilakukan perancangan antar muka yang meliputi perancangan menu, dan perancangan *form-form* dialog berdasarkan studi literatur yang ada.

3. Pembuatan Perangkat Lunak

   Pada tahap ini, model dan rancangan sistem yang telah dibuat akan diimplementasikan. Server dan client dari sistem dibangun menggunakan Visual Basic 6.

4. Uji Coba dan Evaluasi

   Pada tahap ini, dilakukan uji coba dengan beberapa data untuk kemudian dilakukan evaluasi terhadap hasil uji coba tersebut.

5. Penyusunan Buku Tugas Akhir

   Tahap ini merupakan tahap terakhir dari proses pengerjaan Tugas Akhir ini. Buku dokumentasi akan disusun sebagai laporan dari seluruh proses pengerjaan Tugas Akhir ini.

1.6 SISTEMATIKA PEMBAHASAN

Sistematika penulisan tugas akhir ini dapat dijelaskan sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, tujuan dan manfaat, perumusan masalah, batasan permasalahan, metodologi yang digunakan, dan sistematika penyusunan tugas akhir.

BAB II DASAR TEORI

Bab ini membahas konsep dasar dari teknologi-teknologi yang menunjang sistem ini. Teknologi-teknologi tersebut yaitu teknologi *webcam*, teknologi pemrograman *socket* beserta penggunaannya, teknologi SMS beserta penggunaannya dan teknologi internet.

BAB III PERANCANGAN PERANGKAT LUNAK

Bab ini membahas desain dari perangkat lunak yang akan dibuat meliputi : desain antarmuka,desain data, dan desain proses.

BAB IV IMPLEMENTASI PERANGKAT LUNAK

Bab ini membahas implementasi dari desain-desain sistem yang dilakukan pada tahap desain termasuk didalamnya disertakan *source code* yang penting dalam aplikasi.

BAB V UJI COBA DAN EVALUASI

Pada bab ini dibahas mengenai uji coba dari aplikasi yang dibuat dengan melihat output yang dihasilkan oleh aplikasi. Dari output tersebut dapat dilakukan evaluasi untuk mengetahui kemampuan dari aplikasi yang dibuat.

BAB VI KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan aplikasi selanjutnya.

BAB II

DASAR TEORI

BAB II

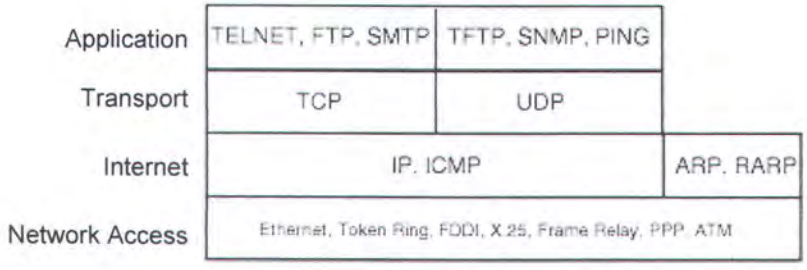
DASAR TEORI

Pada bab ini akan dibahas mengenai teori-teori penunjang dalam perancangan dan pembuatan aplikasi.

2.1 TEKNOLOGI INTERNET

2.1.1 TCP/IP

TCP/IP adalah salah satu model protokol jaringan internet yang sangat popular. Model ini dibentuk dari dua protokol utamanya, TCP (*Transmission Control Protocol*) dan IP (*Internet Protocol*). Pada model ini terdapat empat buah *layer* seperti gambar di bawah ini:



Gambar 2.1 Lapisan pada TCP/IP

Pada *layer* yang paling bawah adalah Network Access. *Layer* ini berfungsi untuk menangani *mapping* alamat IP ke alamat fisik jaringan dan enkapsulasi

paket data dari *layer* sebelumnya yaitu *IP datagram* menjadi data yang dimengerti oleh jaringan.

Layer berikutnya adalah Internet, merupakan jantung dari TCP/IP dimana IP (*Internet Protocol*) bekerja. IP menyediakan framework untuk pengiriman data dari titik A ke titik B. IP adalah protokol yang *connectionless*, yang berarti IP tidak memerlukan konfirmasi paket data telah tiba di tujuan atau belum.

Diatas *layer* Internet adalah Transport. Pada *layer* ini terdapat dua macam protokol yang digunakan yaitu TCP dan UDP. TCP menyediakan aturan pengiriman data yang reliabel dengan *error correction* pada pergerakan data dari sumber ke tujuan. UDP menyediakan aturan pengiriman datagram yang *connectienless* tanpa *error correction.*

Layer paling atas adalah Application, yaitu tempat protokol aplikasi berada. Protokol aplikasi yang populer diantaranya adalah HTTP (Hyper Text Transport Protocol), yang digunakan untuk halaman web; FTP (File Transfer Protocol), cara paling umum untuk mengakses file melalui internet; SMTP (Simple Mail Transfer Protocol), untuk pengiriman pesan email.

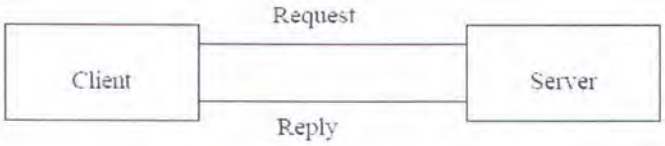
Pengalamatan IP

IP menggunakan skema pengalamatan khusus untuk mengidentifikasi sebuah koneksi kemputer di dalam jaringan yang disebut dangan *IP Address*. *IP Address* terdiri dari empat nomor 8-bit (0-255) yang dipisahkan oleh titik. *IP Address* dalam internet dunia diatur oleh oleh *Internet Network Information Center* (InterNIC)

Di dalam jaringan internet, sebuah *IP Address* mengidentifikasikan sebuah komputer atau gateway dari jaringan.

2.1.2 Socket

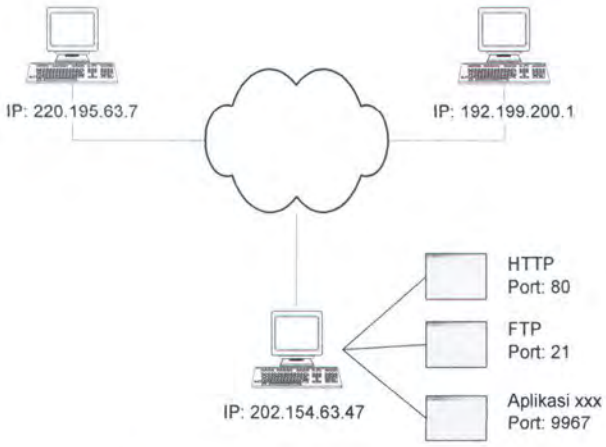
Socket secara sederhana diartikan sebagai *end-point* untuk sebuah komunikasi antar aplikasi pada dua komputer. Pada gambar 2.3 digambarkan sebuah komputer *server* dan sebuah komputer *client* dihubungkan dengan sebuah pipa sebagai tempat berlangsungnya aliran data, dan tiap ujung pipa inilah yang digambarkan sebagai *end-point*. Socket menyediakan logical *end-point* dari dua komputer tersebut. Ada dua macam socket, yaitu *Stream socket* untuk aplikasi berbasis TCP/IP dan *Datagram socket* untuk aplikasi berbasis UDP.



Gambar 2.2 Pipa komunikasi client server

Port Number

Port number atau nomor port mengidentifikasikan sebuah aplikasi yang berjalan di dalam komputer yang berada dalam jaringan internet. Alokasi nomor port ini memiliki limit atau jumlah maksimal yang tidak terbatas. Gambar 2.3 menunjukkan posisi *port number* dalam jaringan internet.



Gambar 2.3 *Port number*

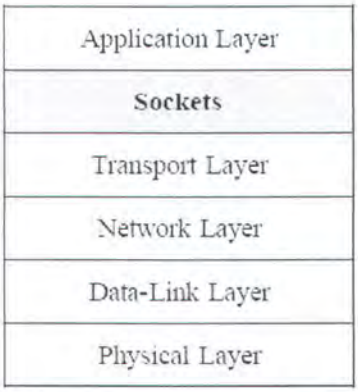
Ada beberapa nomor yang khusus untuk aplikasi yang sudah umum digunakan dalam dunia internet.

HTTP(WWW)	80
FTP	20 dan 21
Gopher	70
SMTP	25
POP3(email)	110
Telnet	23

Winsock (Windows Socket)

Gambar 2.4 menggambarkan posisi socket bekerja didalam model TCP/IP. Socket digambarkan bekerja dantara level bawah (atau di *layer* transport) dan *layer* aplikasi untuk menyediakan interface kepada user di *layer* aplikasi ke *layer* bawah.

PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER



Gambar 2.4 Socket di dalam TCP/IP

Socket merupakan sekumpulan prosedur dan fungsi dalam *API (Application Programming Interface)* yang di dalam UNIX dikenal dengan nama *Barkeley Sockets*. *Barkeley Socket* ini digunakan oleh programmer UNIX untuk melakukan komunikasi internet.

Windows Socket (Winsock) adalah sekumpulan prosedur dan fungsi API pada Windows. Kontrol Winsock di Visual Basic 6 digunakan untuk membuat sebuah aplikasi memiliki akses ke *low-level functions* dari *Transmission Control Protocol/Internet Protocol* (TCP/IP). Winsock adalah standar yang digunakan oleh Microsoft. Standar ini pada dasarnya adalah sekumpulan prosedur yang menggambarkan komunikasi ke dan dari *stack* TCP/IP. Prosedur-prosedur ini bertempat di dalam sebuah *dynamic link library*(dll) yang berjalan di sistem operasi Windows. DLL Windows ini behubungan secara langsung dengan TCP/IP. Dari sinilah ia berinteraksi di dalam internet.

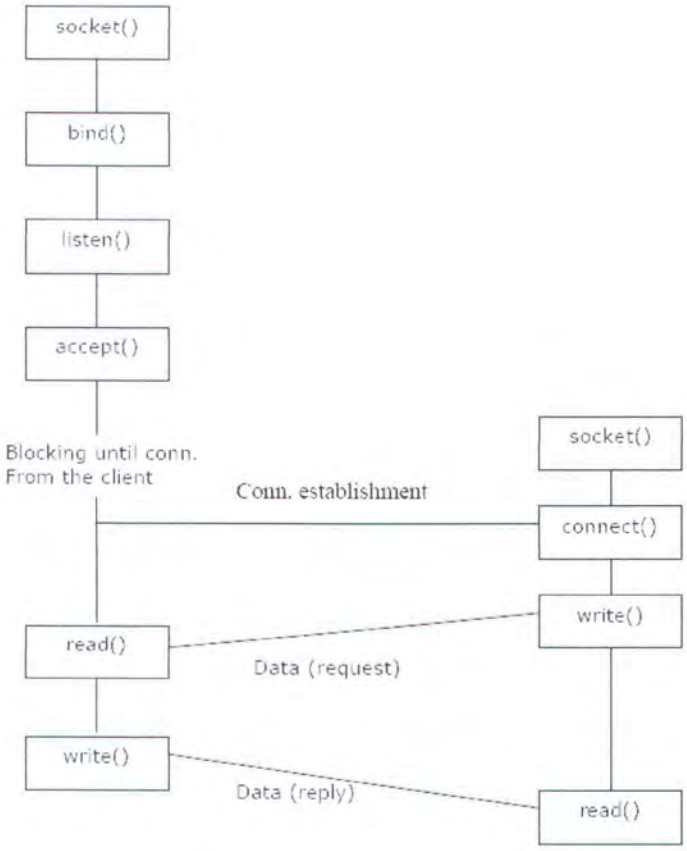
Mode Operasi pada Winsock

Inti protokol dari Transport *layer* adalah TCP dan Unit Datagram Protokol (UDP). Kontrol Winsock mendukung dua mode operasi di bawah ini:

- sckTCPProtocol

- sckUDPProtocol

Berikut adalah model operasi pada Winsock:



Gambar 2.5 Skenario komunikasi menggunakan socket

Berikut adalah property, method dan even dari kontrol dari Winsock yang digunakan untuk proses komunikiasi client server:

Tabel 2.1 Tabel Property Kontrol Winsock

Property	Type	Keterangan
BytesReceived	Long	Total jumlah byte yang diterima dalam satu *session.*
LocalHostName	String	Nama host yang terdaftar si DNS. Bekerja di LAN.
LocalIP	String	IP Address dari host, atau 127.0.0.1.
LocalPort	Long	Nomor port dimana Winsock bekerja, baik untuk listening maupun koneksi.
Protocol	ProtocolConstant	Protokol yang digunakan. Dapat berupa sckTCPProtocol atau sckUDPProtocol. 0 – sckTCPProtocol (Default) 1 – sckUDPProtocol
RemoteHost	String	Komputer tujuan dimana Winsock akan dihubungkan. Digunakan untuk komputer client.
RemoteHostIP	String	Alamat IP komputer dimana Winsok dihubungkan. Hanya bisa diakses ketika Winsok telah dihubungkan.
RemotePort	Long	Nomor port komputer tujuan Winsock akan dihubungkan. Digunakan untuk komputer client.

State	StateConstants	Angka yang merepresentasikan kondisi koneksi dari Winsock. 0 - Closed (sckClosed) 1 - Open (sckOpen) 2 - Listening (sckListening) 3 - Connection pending 4 - Resolving Host (sckResolvingHost) 5 - Host Resolved (sckHostResolved) 6 - Connecting (sckConnecting) 7 - Connected (sckConnected) 8 - Closing (sckClosing) 9 - Error (sckError)

Tabel 2.2 Tabel Method Kontrol Winsock

Method	Protokol	Keterangan
Accept (requestID As Long)	TCP	Menerima request yang datang untuk koneksi secara penuh.
Close	TCP	Menutup semua koneksi socket saat ini. Menutup alur transfer data dan me-*reset* porperti seperti RemoteHostIP.
GetData (data, [type,] [maxLen])	TCP/UDP	Digunakan di even DataArrival untuk mendapatkan data dari internal buffer dari Winsock. Data di buffer dibersihkan

		setelah penerimaan dab disimpan ke blok variant. Dan dapat di set sebagai tipe data string
Listen	TCP	Membuka sebuah socket untuk menunggu koneksi ke host.
PeekData (data, [type,] [maxLen])	TCP/UDP	Digunakan di even DataArrival untuk mendapatkan data dari internal buffer dari Winsock. Data di buffer tidak dibersihkan setelah penerimaan.
SendData (data)	TCP/UDP	Digunakan untuk mengirimkan data dalam koneksi

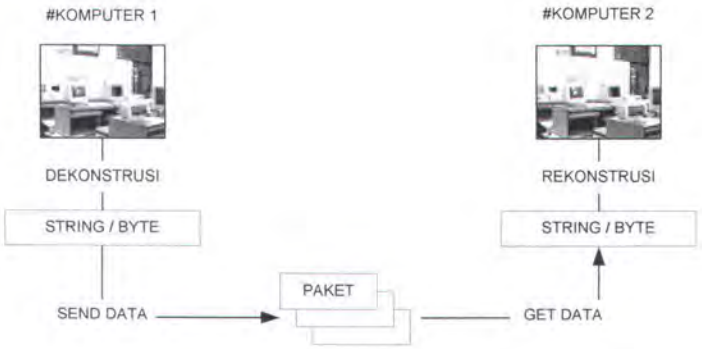
Tabel 2.3 Tabel Even Kontrol Winsock

Event	**Keterangan**
Close	Terjadi ketika koneksi kontrol di control Winsock terputus.
Connect	Terjadi ketika control Winsock membuka koneksi baru keluar.
ConnectionRequest (requestID As Long)	Terjadi ketika sebuah request konesi datang. Di even inilah sebuah socket server menerima mombuka koneksi baru.
DataArrival (bytesTotal As Long)	Terjadi ketika data telah diterima ke internal buffer dari Winsock. Di even inilah method GetData atau PeekData

	digunakan.
Error (number As Integer, Description As String, Scode As Long, Source As String, HelpFile as String, HelpContext As Long, CancelDisplay As Boolean)	Terjadi ketika terjadi error pada Winsock untuk beberapa alasan, contohnya gagal untuk mengirim dan menerima secara sempurna.
SendComplete	Terjadi ketika sebuah data telah terkirim semuanya.
SendProgress (bytesSent As Long, bytesRemaining As Long)	Terjadi ketika proses pengiriman data, memberikan detail dari status transmisi.

Pengiriman Gambar Melalui Winsock

Pengiriman data oleh Winsock dapat menggunakan protocol TCP/IP ataupun UDP. Data yang dapat dikirimkan oleh winsok adalah data string dan data byte. Gambar 2.6 menunjukkan skema pengiriman gambar menggunakan Winsock:



Gambar 2.6 Skema pengiriman gambar pada Winsock

Komputer #1

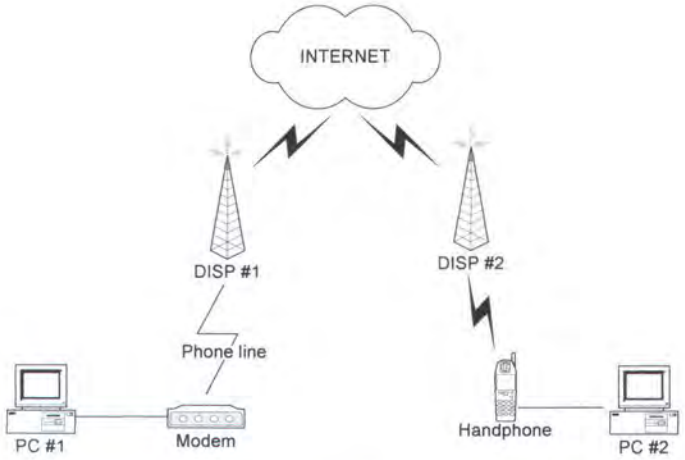
1. Dekonstruksi file gambar ke string atau byte.
2. String / byte hasil dekonstruksi dikirim oleh Winsock kedalam paket-paket.

Komputer #2

1. Winsock menerima paket-paket data dalam tiap even *data_arival()*
2. Konstruksi string/byte ke file gambar.

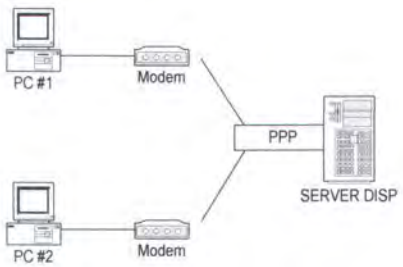
2.1.3 Dial-up Networking

Dial-up Networking (DUN) adalah sebuah konsep jaringan internet dimana koneksi internet pada komputer dengan dial up menggunakan modem dan saluran telepon, yang diperlihatkan pada gambar 2.6.

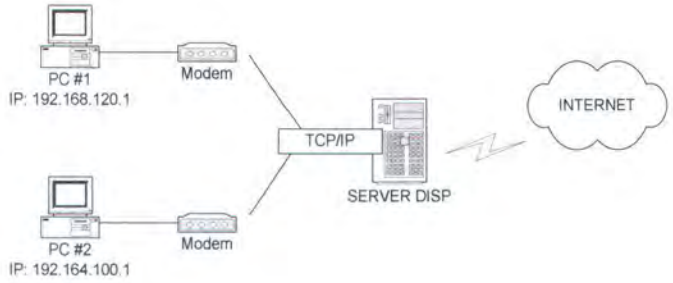


Gambar 2.7 *Dial up* ke internet menggunakan modem

Pengertian *Dial up* berarti sebuah PC melakukan koneksi ke internet menggunakan modem *Dial up Service Provider (DISP)*, yaitu sebuah server yang menggunakan *Point-to-Point Protocol (PPP)* dan TCP/IP untuk membangun *modem-to-modem link*, dan kemudian melakukan *routing* ke internet. PPP digunakan untuk mengidentifikasi alamat komputer ketika PC melakukan *dialing* ke server *DISP*. Setelah *dialing* PC mendapatkan IP yang dapat diakses melalui internet.



Gambar 2.8 Protocol PPP untuk modem-to-modem link



Gambar 2.9 TCP/IP untuk setelah Dial up ke Internet

2.1.3.1 Dial up Internet Service Provider (DISP)

Dial up Internet Service Provider (DISP) adalah adalah usaha bisnis atau organisasi yang menyediakan layanan *dial-up* akses ke internet bagi komputer rumah. Beberapa *DISP* yang populer di Indonesia diantaranya TelkomNet, Indosat, dan Radnet.

Tabel 2.4 Tabel Produk DUN di Indonesia

Provider	Produk
Telkomnet	Telkomnet @instan
	Telkomnet @flexi
	Telkomnet @speedy
Indosat	IM2
Radnet	Radnet
	Cerm@t

2.1.3.2 Windows Internet Service (WinInet)

WinInet adalah library Windows yang menyediakan fungsi-fungsi dasar pemrograman internet di Windows. Di dalam WinInet API terdapat fungsi-fungsi

yang mensupport penggunaan untuk FTP Client, Autodial, Passport, Internet Explorer 6.0, dan MSXML Parser.

Diantara fungsi-fungsi dalam Wininet yang digunakan untuk keperluan dial up ke internet adalah fungsi *InternetDial( )* dan *InternetHangUp( )*.

1. InternetDial()

Untuk dial up ke internet. Juga menampilkan dialog box seperti yang terlihat ketika melakukan dial up ke internet.

Parameter-parameter fungsi

- *hwndParent*

  [input] Handel ke *parent window*.

- *lpszConnectoid*

  [input] Pointer ke nama *Dial-up Connection* yang akan digunakan.

- *dwFlags*

  [input] Option. Parameter ini dapat diisi salah satu nilai berikut:

  INTERNET_AUTODIAL_FORCE_ONLINE

  Pemaksaan pengaplikasian online connection.

  INTERNET_AUTODIAL_FORCE_UNATTENDED

  Pemaksaan pengaplikasian sebuah Internet *dial-up* yang tidak jelas statusnya. Bila intervensi dari user diperlukan, maka fungsi akan gagal diaplikasikan.

  INTERNET_DIAL_FORCE_PROMPT

  Mengacuhkan seting "dial automatically" dan memaksa interface dari dialing user untuk ditampilkan.

INTERNET_DIAL_UNATTENDED

Menghubungkan pada internet melalui modem, tanpa menampilkan sebuah user interface, bila mungkin. Sebaliknya, fungsi akan menunggu inputan dari user.

INTERNET_DIAL_SHOW_OFFLINE

Menampilkan tombol Work Offline dan Cancel pada interface dialing user.

- *pdwConnection*

[output] Pointer yang merujuk pada sebuah variabel yang menspesifikasikan.

- *dwReserved*

[input] Cadangan. Harus nol.

Return Values

Mengembalikan ERROR_SUCCESS bila berhasil, atai error value jika gagal. Kode error yang ada dapat berupa salahsatu dari value dibawah ini.

ERROR_INVALID_PARAMETER

Satu atau lebih paramater yang ada tidak benar.

ERROR_NO_CONNECTION

Ada masalah pada koneksi *dial-up*.

ERROR_USER_DISCONNECTION

User menekan baik tobol Work Offline atau Cancel pada dialog box koneksi Internet

.

2. *InternetHangUp()*

Fungsi untuk memutuskan koneksi internet.

Parameter-parameter fungsi

- *dwConnection*

[input] Nomor koneksi yang akan diputus.

- *dwReserved*

[input] Reserved. Harus bernilai 0.

Return Values

Mengembalikan ERROR_SUCCESS jika sukses, atau nomer error jika sebaliknya.

2.2 TEKNOLOGI WEBCAM

Webcam adalah sebuah *PC Camera* yang dikoneksikan ke internet. *Webcam* secara luas mulai digunakan sekitar satu dekade yang lalu oleh banyak aplikasi dan dunia web.

2.2.1 Metode Viewing pada Webcam

Ada dua macam metode *viewing* pada *webcam* yaitu *video streamming* dan *snapshot*.

Metode *video streamming* adalah jenis *webcam* dimana *view* yang ditampilkan seperti siaran langsung dalam TV. *Video streamming* berarti menampilkan gambar secara kontiniu seperti TV. Model ini memerlukan banyak *bandwith* untuk pengiriman *video streamming*-nya.

Snapshot webcam adalah jenis *webcam* yang mana gambar di-*refresh* selama xx detik, tanpa ada *continuous motion* seperti layaknya TV. Kelebihan dari *webcam* ini adalah bandwith yang dibutuhkan tidak terlalu besar. Transfer rate dari *webcam* ini tergantung dari ukuran gambar itu sendiri. Semakin besar ukuran gambar maka transfer rate-nya semakin rendah. Selain itu juga dipengaruhi oleh infrastruktur yang digunakan, apakah jaringan LAN ataupun telepon.

2.2.2 Image Capturing

PC Camera adalah salah satu device yang digunakan untuk proses *image capturing* atau pengambilan gambar dan menyimpannya ka file. Untuk keperluan *image capturing* ke file, sistem operasi Windows memiliki *library* yang mendefinisikan fungsi-fungsi yang berhubungan *image capturing* dalam *library User32* dan *Avicap32* yang digunakan oleh *compiler* macam Visual Basic atau Visual C++. Disamping itu ada juga interpreter yang mendefinisikan library tersendiri untuk mengurusi *image capturing* dari *webcam*, contohnya *Python Video Capture* dan *Python Image Library* untuk interpreter Python.

2.2.2.1 Avicap32 Library

Avicap32 adalah library Windows yang khusus digunakan komputer ketika melakukan pengambilan gambar video dari *PC Camera*. Berikut adalah fungsi-fungsi dari *Avicap32* yag digunakan untuk *image capturing*.

1. capCreateCaptureWindow ()

Proses Video capturing dimulai dengan fungsi ini. Fungsi ini untuk mengambil gambar video dari *pc camera* dan menempelkannya ke dalam window yang disebut *capture window*.

Parameter-parameter fungsi

- *lpszWindowName*

  [input] Nama driver PC Camera yang digunakan untuk menghasilkan *capture window*.

- *dwStyle*

  [input] Jenis dari window yang digunakan untuk capture window..

- *x*

  [input] Koordinat sumbu x dari ujung kiri atas *capture window*.

- *y*

  [input] Koordinat sumbu y dari ujung kiri atas *capture window*.

- *nWidth*

  [input] Lebar dari *capture window*.

- *nHeight*

  [input] Tinggi dari *capture window*.

- *hWnd*

  [input] *Handle* ke *parent window*.

- *nID*

  [input] Pengindentifikasi window.

Return Values

Mengembalikan *handle* dari *capture window* jika sukses atau NULL jika gagal.

2. ***capGetDriverDescriptionA ()***

Fungsi ini digunakan untuk membaca deskripsi driver *PC Camera* yang digunakan.

Parameter-parameter fungsi

- *wDriverIndex*

  [input] Index dari driver, antara 0-9. *Plug-and-Play drivers* diprioritaskan, diikuti dengan driver yang terdaftar di register, kemudian driver yang terdaftar di SYSTEM.INI.

- *lpszName*

  [input] Pointer ke buffer yang terdiri dari string yang berkorespondensi ke nama driver dari *pc camera.*

- *cbName*

  [input] Panjang, dalam *bytes*, berada di buffer dan ditunjukkan oleh lpszName.

- *lpszVer*

  [input] Pointer ke buffer yang terdiri dari string yang berkorespondensi ke deskripsi driver dari *pc camera.*

- *cbVer*

  [input] Panjang, dalam *bytes*, berada di buffer dan ditunjukkan oleh lpszVer.

Return Values

Mengembalikan TRUE jika sukses atau FALSE jika sebaliknya.

2.2.2.2 User32 Library

Library ini biasa disebut dengan *Windows User API Client Library*, diperlukan oleh Wondows dan terdiri dari fungsi-fungsi yang membentuk komponen utama GUI. Berikut adalah fungsi-fungsi dalam *User32* yang digunakan untuk keperluan *image capturing*.

*1. **SendMessage ( )***

Fungsi ini digunakan untuk mengirimkan pesan yang sebagai penunjuk prosedur yang dilakukan untuk spesifikasi tertentu pada windows.

Parameter-parameter fungsi

- *hwnd*

  [input] *Handle* ke window dimana akan dilakukan spesifikasi oleh prosedur windows sesuai dengan pesan yang dikirimkan.

- *Msg*

  [input] Pesan yang dikirimkan. Berupa bilangan yang menandai prosedur tertentu dalam *Windows procedures* yang akan dijalankan. Biasanya digunakan flag sebagai penanda bilangan-bilangan yang melakukan spesifikasi tertentu. Berikut adalah flag-flag yang digunakan untuk keperluan *image capturing*:

  WM_CAP_DRIVER_CONNECT

  Pesan untuk melakukan koneksi ke driver PC Camera.

WM_CAP_DRIVER_DISCONNECT

Pesan untuk memutuskan koneksi ke driver

WM_CAP_SET_PREVIEWRATE

Pesan untuk menampilkan *preview rate* dari *webcam.*

WM_CAP_SET_SCALE

Pesan untuk mengijinkan perubahan skala dari *hwnd.*

WM_GET_FRAME

Pesan untuk mendapatkan frame dari *hwnd.*

WM_COPY

Pesan untuk meng-copy frame yang didapatkan ke *clipboard.*

- *wParam*

[input] Konstanta untuk nilai spesifikasi bedasarkan pesan yang dikirimkan.

- *lParam*

[input] Pesan informasi pesan spesifikasi tambahan.

Return Values

Hasil dari proses terhadap pesan, tergantung dari pesan yang dikirimkan.

2. ***DestroyWindow ()***

Fungsi ini digunakan untuk membersihkan window dari layar.

Parameter-parameter fungsi

- *hwnd*

[input] *Handle* ke window yang akan dibersihkan.

PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

Return Values

Jika fungsi sukses maka menghasilkan nilai non zero.

2.2.2.3 Python Image Library

Untuk akses ke ke *webcam* dapat menggunakan ekstensi Python untuk WIN32 Video Capture. Video Capture terdiri modul-modul *low level* (vidcap.pyd) dan modul *high level* (VideoCapture.py). Juga mempunyai tool-tool yang dapat digunakan untuk secara periodik meng-upload gambar *webcam* ke web server, memonitor dan menyimpan sebuah gambar ke harddisk. Berikut adalah perintah mengambil snapshot dari *webcam* menggunakan Python:

```
from VideoCapture import Device
cam = Device()
cam.saveSnapshot('image.jpg')
```

Sedangkan untuk keperluan pengolahan *image* atau gambar yang dihasilkan digunakan Python Image Library [6].

2.2.3 Kompresi JPEG

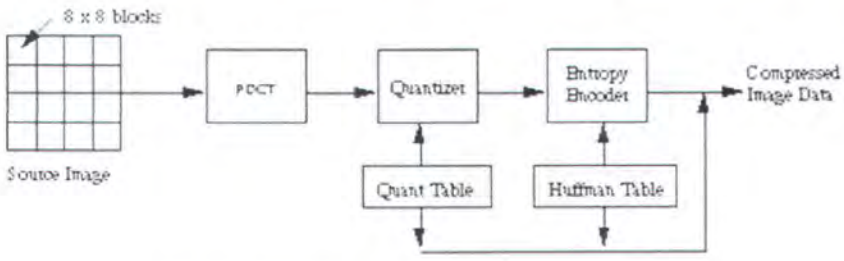
Dalam pengolahan data, kompresi adalah sebuah cara untuk memperkecil ukuran data untuk mempersingkat waktu pengiriman maupun untuk menghemat penggunaan media penyimpanan. Kompresi data ada dua macam yaitu kompresi *lossy* dan kompresi non-*lossy*. Kompresi non-*lossy* berarti data dimampatkan dengan tidak menghilangkan sedikitpun informasi yang terkandung didalamnya. Contoh metode kompresi ini adalah ZIP. Kompresi *lossy* berarti menghilangkan sebagian informasi dalam file untuk memperbesar rasio kompresi. Kompresi lossy biasanya digunakan pada file-file multimedia berukuran besar, misalnya metode

JPEG (Joint Photographic Expert Group) yang digunakan untuk mengkompresi file grafis.

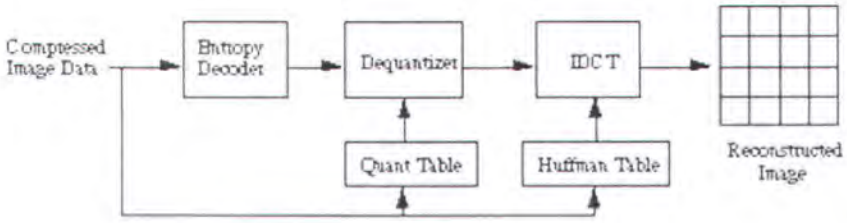
Proses kompresi JPEG menggunakan *Discrete Cosine Transform (DCT)* [8]. Berikut adalah langkah-langkah kompresi JPEG secara umum:

1. *Image* dipecah menjadi blok pixel 8 x 8.
2. Dari kiri ke kanan, atas ke bawah, dimana DCT akan dioperasikan di setiap blok
3. Tiap blok dioperasikan dengan persamaan DCT
4. Array dari blok yang dikompresi yang membentuk gambar disimpan kedalam drive space.
5. Untuk rekonstruksi gambar dengan proses dekompresi menggunakan *Inverse Discrete Cosine Transform (IDCT).*

Entropy Coding (EC) menghasilkan kompresi tambahan dengan meng-*encode* koefisien DCT menjadi lebih *compact* berdasarkan karakteristik statistiknya[9]. Kode Huffman digunakan untuk menghasilkan *Variable Length Coding (VLC)* [9]. Gambar 2.10 menggambarkan proses encoding atau gambar ke JPEG dan gambar 2.11 menggambarkan proses decoding gambar JPEG ke asal.



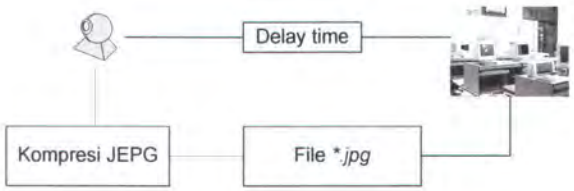
Gambar 2.10 JPEG Encoder Block Diagram



Gambar 2.11 JPEG Decoder Block Diagram

Kompresi JPEG digunakan secara luas dalam dunia internet dan saat ini banyak *library-library* yang mendukung kompresi gambar JPEG, salah satunya dengan menggunakan *Python Image Library*.

Untuk pengambilan gambar dari *webcam* metode snapshot yaitu mengambil *static image* secara simultan dan menyimpan ke dalam file **.jpg,* proses menampilkan gambar dilakukan setiap file tersebut ter-*update*.



Gambar 2.12 Skema pengambilan *snapshot* gambar

Gambar 2.12 menunjukkan setiap *n delay time* gambar **.jpg* di update sehingga menimbulkan kesan gambar bergerak. Metode *snapshot* ini cocok digunakan jika untuk keperluan menampilkan gambar saja (bukan *video recording*) karena *drive space* yang digunakan hanya seukuran satu frame. Jika pengambilan gambar dan menampilkan gambar dilakukan di dua komputer yang berbeda maka metode ini fleksibel digunakan untuk infrastruktur jaringan dengan bandwith tinggi ataupun rendah. Fleksibilitas ini terdapat pada *n delay time* dimana semakin tinggi bandwith jaringan maka *n delay time* dapat semkin rendah begitu juga sebaliknya.

2.3 TEKNOLOGI SMS (SHORT MESSAGE SERVICE)

Short messaging service (SMS), sebagaimana didefiniskan oleh organisasi Etsi (Dokumen GSM 03.40 dan GSM 03.38), dapat memuat sebanyak 160 karakter, dimana setiap karakter terdiri dari 7 bit sesuai dengan aturan *7-bit default alphabet*. Pesan yang memuat karakter ber-bit 8 (satu pesan SMS hanya dapat memuat 140 karakter jenis ini) biasanya tidak dimunculkan dalam bentuk teks, melainkan dalam bentuk citra, nada bunyi (ring tones) ataupun *setting* WAP yang disediakan OTA. Sedangkan pesan yang memuat 16 bit per karakter (satu pesan SMS hanya dapat memuat 70 karakter jenis ini) digunakan untuk pesan teks Unicode (UCS2), yang dapat dilihat pada kebanyakan telepon. Sebuah pesan 16-bit yang terdiri dari class 0, pada beberapa telepon akan nampak seperti sms flash—seperti: sms yang berkedip-kedip pada desktop telepon, peringatan adanya sms.

2.3.1 Format PDU

Terdapat 2 cara untuk mengirim dan menerima pesan SMS, yaitu dengan menggunakan mode teks dan mode PDU (protocol description unit). Mode teks (kadang tidak tersedia pada beberapa telepon) adalah sebuah kode dari sekelompok bit yang direpresentasikann oleh mode PDU. Huruf-huruf dapat nampak berbeda dan terdapat beberapa macam alternatif pengkodean ketika sebuah SMS ditampilkan. Pengkodean yang biasanya digunakan adalah "PCCP437", "PCDN", "8859-1", "IRA", dan "GSM". Semua jenis pengkodean tersebut ditentukan dengan menggunaka *at-command* AT-CSCS ketika sebuah

SMS ditampilkan untuk dibaca dari sebuah aplikasi pada PC. Jadi sebuah telepon akan memilih pengkodean yang sesuai untuk menampilkan setiap pesan SMS yang diterimanya. Sebuah aplikasi untuk membaca SMS dapat menggunakan salah satu dari mode teks dan mode PDU. Pada aplikasi berbasis mode teks, hanya beberapa SMS tertentu yang dapat dibaca, tergantung berapa pengkodean yang didefinisikan sebelumnya. Sementara pada aplikasi berbasis PDU, semua format pengkodean dapat diimplementasikan.

2.3.1.1 Menerima sebuah pesan pada mode PDU

String PDU tidak hanya terdiri dari sebuah pesan, melainkan juga beberapa informasi-meta tentang pengirim pesan, *SMS service centre*, waktu, dan lain-lain. Semua informasi tersebut berbentuk heksadesimal oktet, ataupun decimal semi-oktet. String berikut ini diterima melalui telepon selular berjenis nokia 6110 ketika ponsel yang bersangkutan mengirim pesan yang berisi pesan ,"hellohello".

07 917283010010F5 040BC87238880900F10000993092516195800AE8329BFD4697D9EC37

Sekuensial oktet tersebut terdiri dari 3 bagian, sebuah inisialisasi oktet yang menunjukkan panjang dari SMS ("07"), informasi SMSC ("917283010010F5"), dan bagian SMS_DELIVER(didefinisikan oleh ETSI pada GSM 03.40)

Tabel 2.5 Tabel string PDU yang diterima

Oktet	Penjelasan
07	Panjang dari informasi SMSC (pada kasus ini 7 oktet)
91	Tipe alamat dari SMSC. (91 berarti format internasional dari sebuah nomor ponsel)
72 83 01 00 10 F5	Nomor pusat layanan (dalam decimal semi-oktet). Panjang dari nomor ponsel adalah ganjil(11), sehingga sebuah tambahan F ditambahkan untuk membentuk susunan oktet yang benar. Nomor ponsel dari pusat layanan ini adalah "+27381000015". Lihat dibawah ini.
04	Oktet pertama dari pesan SMS_DELIVER
0B	Panjang Alamat. Panjang dari nomor ponsel (0B hex = 11 dec)
C8	Tipe alamat dari nomor pengirim.
72 38 88 09 00 F1	Nomor pengirim (desimal semi-oktet), dengan tambahan F
00	TP-PID. Identifikasi protokol

00	TP-DCS bentuk pengkodean data.
99 30 92 51 61 95 80	TP-SCTS. Waktu (semi-oktet)
0A	TP-UDL. Panjang data pengguna, panjang dari pesan. TP-DCS mengindikasikan 7-bit data, jadi panjang disini adalah nomor dalam septet(10). Jika TP-DCS di-set untuk mengindikasikan 8-bit data atau Unicode, maka panjang data akan berbentuk oktet(9).
E8329BFD4697D9EC37	TP-UD. Pesan "hellohello" , 8-bit oktet mewakili 7-bit data.

Semua bilangan berbasis 8(oktet) di atas merupakan heksadesimal 8-bit oktet, kecuali nomor layanan pusat, nomor pengirim, dan waktu yang berbentuk desimal semi-oktet. Bagian pesan yang berada pada akhir dari string PDU terdiri dari heksadesimal 8-bit oktet, tapi oktet tersebut mepresentasikan 7-bit data.

Semi-oktet merupakan bilangan desimal, sebagai contoh nomor pengirim didapatkan melalui pembentukan swap internal dengan semi-oktet dari "72 38 88 09 00 F1"+"27 83 88 90 00 1F". Panjang dari nomor ponsel merupakan bilangan ganjil, jadi untuk membentuk sebuah sekuensial oktet yang sesuai, maka F ditambahkan. 'Waktu'(timestamp) yang ditunjukkan oleh "99 03 29 15 16 59 08", dimana 6 karakter pertama menunjukkan tanggal, 6 karakter berikutnya

menunjukkan waktu, dan 2 karakter terakkhir menunjukkan zona waktu menurut GMT.

2.3.1.2 Menginterpretasikan 8-bit oktet menjadi pesan 7-bit

Transformasi ini dijelaskan secara detail pada GSM 03.38, dan sebuah contoh dari transformasi "hellohello" ditunjukkan disini. Transformasi didasarkan pada aturan 7 bit default alphabet, tapi aplikasi yang dibangun berdasarkan mode PDU dapat menggunakan sembarang pengkodean karakter.

2.3.1.3 Mengirim sebuah pesan pada mode PDU

Contoh berikut menunjukkan bagaimana mengirimkan pesan "hellohello" pada mode PDU menggunakan nokia 6110

```
AT+CMGF=0     //mengeset mode PDU
AT+CSMS=0     //mengecek apakah modem mendukung perintah-perintah
              SMS
AT+CMGS=23    //mengirim pesan, 23 oktet (tidak termasuk 2
              inisialisasi //nol )
>0011000B916407281553F80000AA0AE8329BFD4697D9EC37<ctrl-z>
```

Terdapat 23 oktet pada pesan ini (46 'karakter'). Oktet pertama ("00") tidak dihitung karena hanya merupakan indikator dari panjang informasi SMSC yang ada (0). String PDU terdiri dari bagian-bagian sebagai berikut:

Tabel 2.6 Tabel string PDU yang terkirim

Oktet	Penjelasan
00	Panjang dari informasi SMSC. Pada

	kasus ini panjangnya 0 yang menandakan bahwa SMSC yang disimpan pada ponsel harus digunakan. *Catatan: oktet ini opsional sifatnya. Pada beberapa ponsel, octet ini bisa tidak nampak.*
11	Oktet pertama dari pesan SMS-SUBMIT
00	TP-Message-Reference. Nilai "00" disini menyerahkan pengesetan referensi pesan dilakukan oleh ponsel yang bersangkutan.
0B	Panjang alamat. Panjang dari nomor (11)
91	Tipe alamat, (91 berarti format internasional dari sebuah nomor ponsel)
6407281553F8	Nomor ponsel pada semi-oktet (46708251358).. Panjang dari nomor ponsel merupakan bilangan ganjil, sehingga sebuah F ditambahkan, sehingga nomor ponsel berubah menjadi "46708251358F" Penggunaan format yang tidak dikenal

	seperti tipe alamat 81 misalnya akan memunculkan nomor ponsel berbasis oktet 7080523185 (0708251358). Sebagai catatan, nomor tersebut memiliki panjang A(10) yang merupakan bilangan genap.
00	TP-PID. identifikasi protokol
00	TP-DCS. Bentuk pengkodean data . Pesan ini dikodekan menurut aturan 7-bit default alphabet. Penggunaan "04"(tidak "00") menunjukkan bahwa bagian TP-User-Data field dari pesan ini harus diinterpretasikan sebagai karakter 8-bit, bukan 7-bit (seperti bentuk citra, nada bunyi (ring tones) ataupun *setting* WAP yang disediakan OTA).
AA	TP-Validity-Period. Periode validitas SMS, "AA" berarti 4 hari. *Catatan: octet ini opsional.*
0A	TP-User-Data-Length. Panjang dari pesan. Bagian TP-DCS menunjukkan 7-bit data, sehingga panjangnya

	merupakan bentuk bilangan berbasis 7, septet(10) Apabila bagian TP-DCS diset menjadi 8-bit data atau Unicode, maka panjangnya akan merupakan bilangan berbasis 8 (oktet)
E8329BFD4697D9EC37	TP-User-Data. Oktet ini merepresentasikan pesan "hellohello".

2.3.2 AT COMMAND

AT COMMAND merupakan perintah untuk melakukan control terhadap mobile telephone GSM melalui interface serial (baik melalui kabel data atau koneksi infrared).

Perintah harus dimulai dengan "AT" dan diakhiri dengan "<CR>" (=0x0D). Input dari perintah akan mendapatkan respon balik (*acknowledge*) dengan "OK" atau "ERROR".

Hayes Standard Command

Hayes-standard Command merupakan command dari AT Hayes yang compatible. Hayes standard command dapat dilihat pada tabel berikut ini:

Tabel 2.7 Tabel Hayes Standard Command

Command	Function
A/	Mengulangi perintah yang terakhir
AT...	Awalan Untuk Semua Perintah yang Lain

ATD<str>;	Dial untuk *dialing string* <str> dengan menggunakan utilitas suara (*voice*) *Valid dial*: "T" (*Tone dialing*) dan jika gagal "P" (Pulse dialing) *ignored.*
ATD><n>;	Memanggil (*Dial*) nomor telepon dari lokasi buku telepon saat ini <n> dan buku telepon diseleksi dengan perintah at+cpbs atau (at^spbs).
ATE0	Menonaktifkan perintah echo
ATE1	Mengaktifkan perintah echo
ATZ	Set Ke konfigurasi default

Acknowledge untuk Komunikasi Data Normal

Tabel 2.8 Tabel Acknowledge untuk komunikasi data normal

Response	Numeric	Meaning
OK	0	Perintah dieksekusi tidak ada Error
RING	2	Ring terdeteksi
NO CARRIER	3	Link Tidak ada(Disconnect)
ERROR	4	Perintah Salah
NO DIAL TONE	6	Tidak ada dial (mode salah)

BUSY	7	*Remote Station* sedang sibuk (*Busy*)

AT Command dan Response

Berikut ini merupakan beberapa cara untuk mengeksekusi AT Command dalam berbagai bentuk yaitu:

Test command	AT+CXXX=?	Telepon akan merespond dengan mengirimkan list parameter dan range nilai
Read command	AT+CXXX?	Perintah ini akan menginformasikan pada setting nilai sekarang dari parameter (s).
Write command	AT+CXXX=<...>	Perintah ini digunakan untuk menge-set parameter yang bias di-set.
Execute command	AT+CXXX	Perintah execute membaca parameter yang tidak di-set tetapi berpengaruh pada proses internal dalam telepon.

Beberapa Perintah AT Cellular Pada GSM

- Perintah untuk memeriksa SMS pada handphone

Tabel 2.9 Tabel AT Command untuk memeriksa SMS

AT+CMGL	LIST SMS
Test Command AT+CMGL=?	Response +CMGL: (List of supported <stat>s) parameter

	<stat> 0 "REC UNREAD" ,menerima semua pesan yang tidak dibaca. 1 "REC READ", menerima semua pesan yang dibaca. 2 "STO UNSENT", menyimpan pesan yang tidak dikirim. 3 "STO SENT", menyimpan pesan yang dikirim. 4 "ALL", Untuk semua pesan.

Perintah untuk mengirim SMS

Tabel 2.10 Tabel AT Command untuk mengirim SMS

AT+CMGS	***Send an SMS***
Test command AT+CMGS=?	*Response* OK

Perintah untuk menghapus SMS

Tabel 2.11 Tabel AT Command untuk menghapus SMS

AT+CMGD	***Delete an SMS in SMS memory***
Test command AT+CMGD=?	*Response* OK

BAB III

PERANCANGAN PERANGKAT LUNAK

BAB III

PERANCANGAN PERANGKAT LUNAK

Bab ini menjelaskan tentang tahapan proses perancangan perangkat lunak mulai dari deskripsi secara umum sampai perancangan data, perancangan aplikasi dan perancangan antar muka.

3.1 DESKRIPSI UMUM

Sistem yang dibuat pada tugas akhir berbasis client-server dan terdiri dari beberapa perangkat lunak yang manunjang.

Perangkat Lunak yang dibuat pada tugas akhir ini terdiri dari dua bagian utama, yaitu perangkat lunak server dan client. Server terdiri dari empat modul program yang dijalankan yaitu:

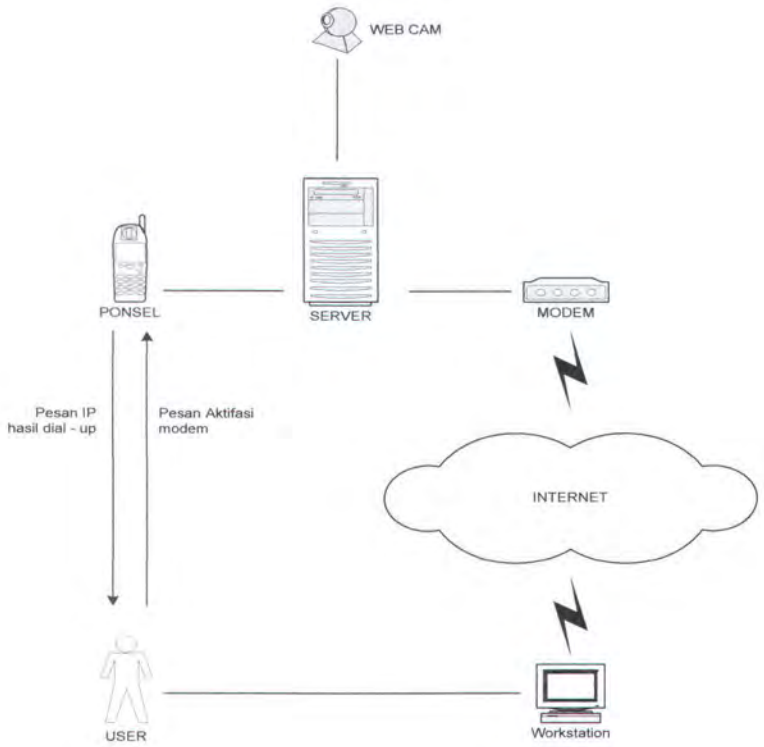
1. Image capture
2. *SMS Gateway*
3. Dialer
4. Socket server

Sedangkan pada sisi client ada dua modul program yaitu :

1. Viewer
2. Socket client

3.1.1 Arsitektur Sistem

Arsitekstur sistem terbagi menjadi dua bagian utama yaitu komputer server dan client. Arsitektur sistem aplikasi ini dapat dilihat pada gambar 3.1 berikut ini.



Gambar 3.1 Arsitektur Sistem

Komputer Server

Komputer server adalah tempat berjalannya aplikasi server dan menempelnya *device-device* penunjang. Pada komputer server terdapat:

1. *Webcam* untuk mengambil gambar ruangan/rumah
2. Modem untuk koneksi ke internet
3. GSM modem untuk mengirim dan menerima pesan aktifasi
4. Aplikasi server

PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

3.1.2 Fitur yang dimiliki Aplikasi

Aplikasi ini terbagi menjadi dua bagian, yaitu aplikasi server (gateway) dan aplikasi client. Fitur pada aplikasi server adalah:

1. Mengambil frame gambar dari PC Camera dan disimpan kedalam sebuah file gambar (**.jpg*).
2. Menerima SMS dari user yang melakukan permintaan *dial-up* dan langsung melakukan proses validasi atas isi SMS.
3. Melakukan *dial-up* ke internet setelah menerima SMS user yang telah divalidasi.
4. Mengirimkan pesan SMS ke user yaitu IP publik yang didapatkan setelah *dialing* ke internet..
5. Mengirimkan gambar hasil *capturing* dari *webcam* ke komputer client secara periodik.

Sedangkan pada aplikasi client memiliki fitur-fitur sebagai berikut:

1. Melakukan koneksi ke server. Default koneksi adalah IP publik dari server yang didapatkan dari provider. Mengolah frame gambar yang dikirim oleh server untuk ditampilkan ke antar muka.

3.2 Perancangan Proses

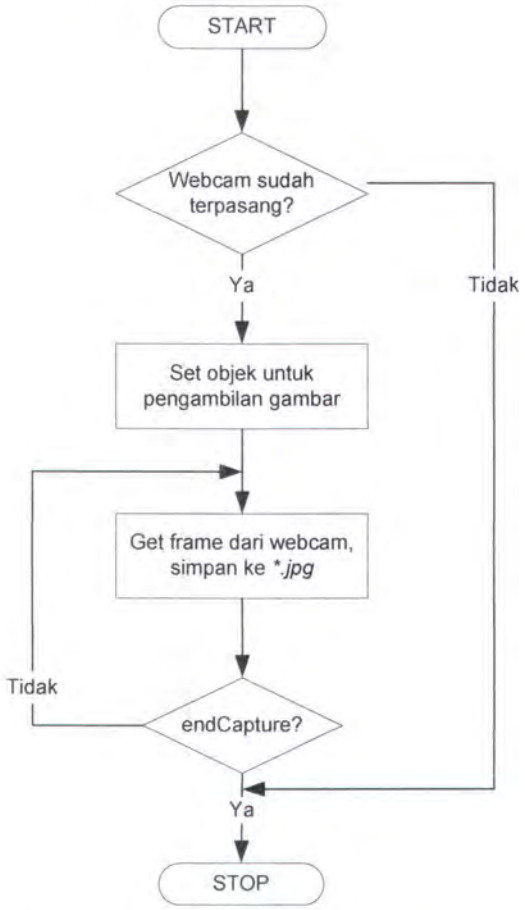
Secara keseluruhan proses-proses yang terjadi di dalam aplikasi ini dapat dibagi menjadi dua, yaitu proses pada server dan proses pada client. Berikut akan dijelaskan lebih lanjut mengenai perancangan proses aplikasi.

3.2.1 Proses pada Server

Didalam server terdapat proses-proses yang dilakukan untuk memenuhi fitur-fitur yang terdapat di aplikasi server. Proses-proses tersebut dijelaskan sebagai berikut:

3.2.1.1 Proses Pengambilan Gambar dari Webcam

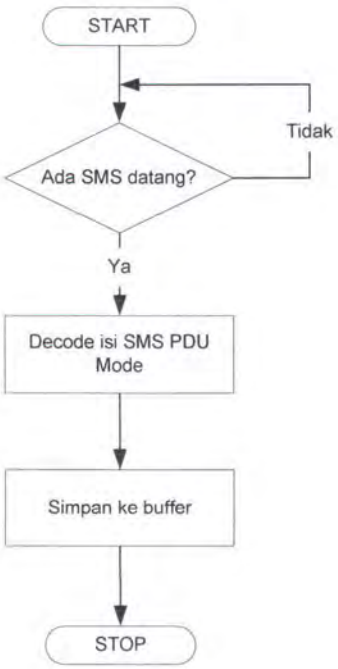
Proses pengambilan gambar dilakukan setelah ada request dari client. Proses ini adalah mengambil gambar dari *webcam* dan menyimpan ke dalam file **.jpg* yang siap dikirim ke client.



Gambar 3.4 Flowchart Proses Pengambilan Gambar dari *Webcam*

3.2.1.2 Proses Penerimaan SMS

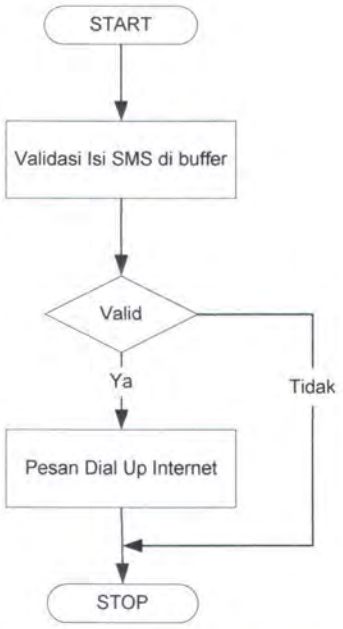
Dalam proses ini terdapat proses decoding SMS karena pada waktu gateway menerima pesan SMS, data yang masuk berupa kode PDU 7 bit sehingga data yang masuk masih belum bisa diproses. Proses decoding akan mengubah kode PDU tersebut menjadi text ASCII normal.



Gambar 3.4 Flowchart Proses Penerimaan SMS

3.2.1.3 Proses Validasi Data dari SMS

Proses ini dilakukan setelah data (dari SMS) diterima dan disimpan di buffer. Data tersebut kemudian divalidasi untuk melakukan *dial-up* ke internet.



Gambar 3.5 Flowchart Proses Validasi Data dari SMS

3.2.1.4 Proses Dial-up Internet

Proses ini digunakan untuk melakukan *dial-up* ke internet melalui provider yang tersedia di sistem server. Provider yang dimaksud adalah penyedia layanan internet menggunakan saluran telepon seperti Telkomnet, Rad Net, IM2. Jika pada sistem belum terdaftar satupun provider maka proses berhenti. Jika terdaftar, proses memeriksa apakah sistem sudah *dial-up* ke jaringan internet melaui modem. Jika belum maka sistem akan melakukan dialing ke provider yang tersedia di sistem.

Gambar 3.6 Flowchart Proses *Dial-up* Internet

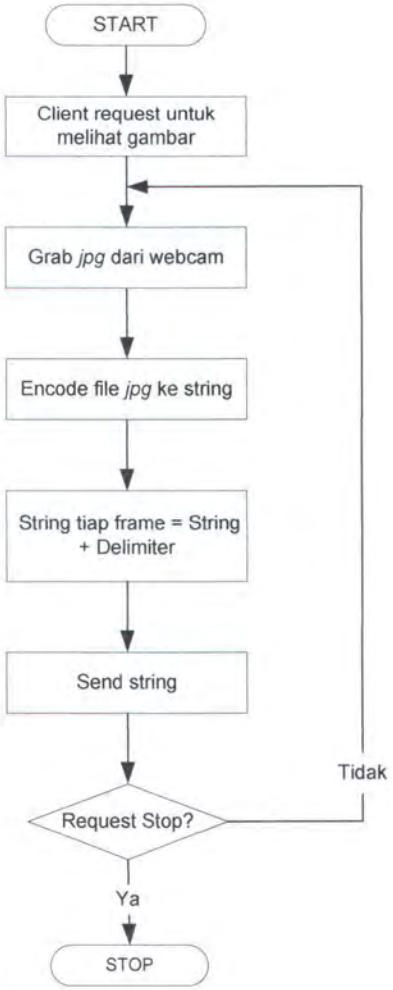
3.2.1.5 Proses Pengiriman SMS

Proses ini adalah untuk mengirimkan kepada user IP publik computer yang didapatkan setelah melakukan *dial-up*. Dalam proses ini terdapat proses encoding SMS sebelum SMS dikirim ke nomer tujuan. Karena pengiriman SMS menggunakan format PDU 7 bit sedangkan teks yang umum masih berupa ASCII. Oleh sebab itu sebelum pengiriman terlebih dahulu harus adanya encoding data pesan.

Gambar 3.7 Flowchart Proses Mengirim SMS

3.2.1.6 Proses Pengiriman Gambar

Sebelum gambar dikirim, gambar dari *webcam* yang disimpan di buffer dikompresi ke dalam format yang lebih kecil yaitu menggunakan kompresi JPEG. Dalam proses pengiriman dilakukan *looping* yang dimulai dari pengambilan gambar, encode data gambar ke string, kemudian pengiriman data string. Karena pengiriman dilakukan secara simultan maka diperlukan pemisah antar file (delimiter) yang berupa konstata string yang selalu disertakan dalam pengiriman tiap frame. Alasan penggunaan kompresi JPEG adalah agar proses pengiriman



Gambar 3.8 Flowchart Proses Pengiriman Gambar

3.2.2 Proses pada Client

Didalam server terdapat proses-proses yang dilakukan untuk memenuhi fitur-fitur yang terdapat di aplikasi server. Proses-proses tersebut dijelaskan sebagai berikut:

3.2.2.1 Proses Konfigurasi

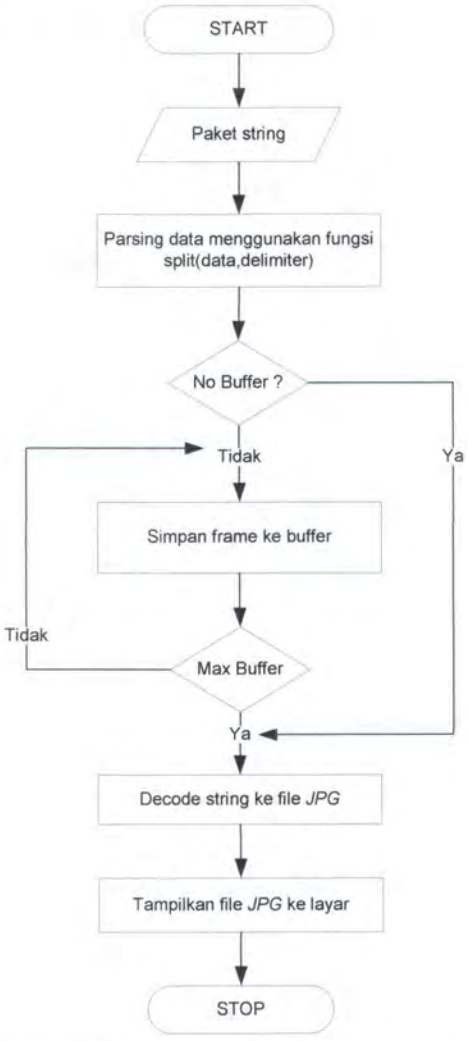
Proses ini menset variabel-variabel sistem yaitu:

1. Buffer size: untuk menentukan ukuran buffer frame

2. Frame rate: untuk menentukan interval waktu *viewing* antar frame

3.2.2.2 Proses Penerimaan Gambar

Pada client terdapat proses penerimaan string berisi data gambar dari server dan merekonstruksi kembali menjadi sebuah file gambar dan ditampilkan ke window. Prosesnya adalah sebagai berikut:



Gambar 3.9 Flowchart Proses pada client

3.2.3 Proses Komunikasi Client Server

Proses ini terjadi pada client dan server. Yang ada di dalam proses ini adalah komunikasi antara client dan server menggunakan socket. Berikut adalah flowchart proses komunikasi dari sisi client dan sisi server:

Gambar 3.10 Flowchart proses komunikasi sisi server

Gambar 3.11 Flowchart proses komunikasi sisi client

3.3 PERANCANGAN ANTAR MUKA

Pada perancangan antar muka ini, aplikasi akan dirancang sehingga dapat diakses dengan mudah oleh user. Karena dalam sistem ini terdapat dua aplikasi yaitu aplikasi server dan client, maka akan dibahas perancangan antar muka dari dua aplikasi tersebut.

3.3.2 Perancangan Antarmuka Aplikasi Server

3.3.2.2 Form Utama Server

Form ini adalah start up dari server. Berfungsi sebagai kontrol dari servis-servis server. Menu utama *form* ini adalah menyalakan dan mematikan servis server yaitu *socket* dan *SMS Gateway*.

Gambar 3.12 *Form* utama server

3.3.2.3 Form Service Socket

Form ini untuk menjalankan servis socket, yaitu membuka socket sebagai gateway untuk lalu lintas data dengan client

Gambar 3.13 *Form* service socket

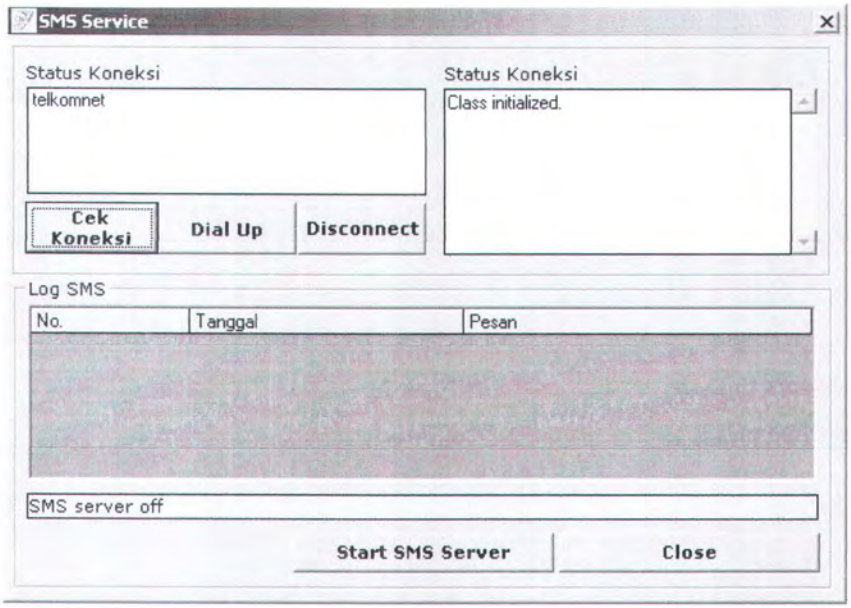
3.3.2.4 Form Service Kamera

Form ini untuk menjalankan servis pengambilan gambar dari *webcam*. Untuk viewing gara bisa menempatkan posisi *webcam* sesuai yang diinginkan.

Gambar 3.14 *Form* service kamera

3.3.2.5 Form Service SMS Gateway

Form untuk mengatur menu-menu *SMS Gateway* dan dial up koneksi ke internet.



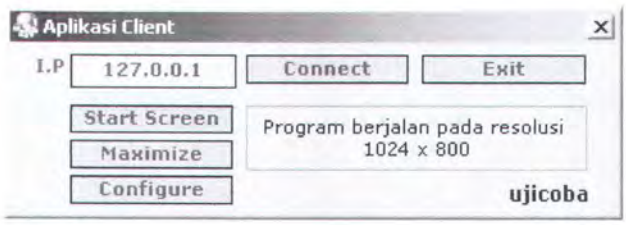
Gambar 3.15 *Form* service *SMS Gateway*

3.3.3 Perancangan Antarmuka Aplikasi Client

3.3.3.2 Form Utama Client

Form untuk mengatur menu-menu utama pada aplikasi client, yaitu:

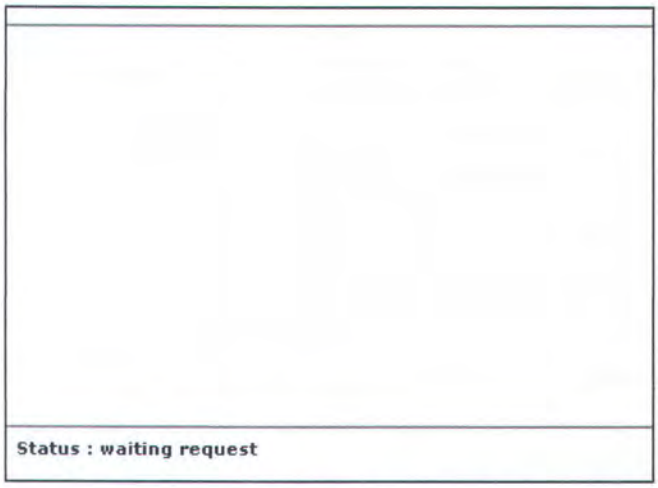
1. *Connect* : untuk koneksi ke server
2. *Start Screen* : untuk memulai pengambilan gambar
3. *Maximize* : untuk menampilkan gambar *full screen*
4. *Configure* : untuk masuk ke window konfigurasi
5. *Exit* : untuk keluar dari program



Gambar 3.16 *Form* menu utama client

3.3.3.3 Form Layar Gambar

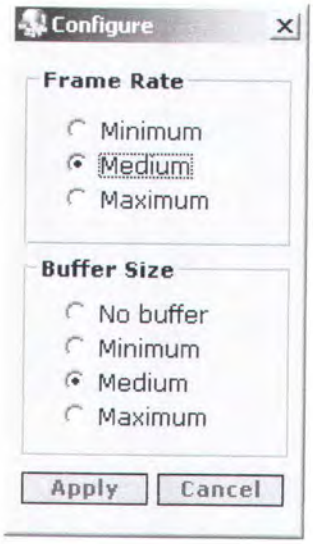
Form untuk menampilkan gambar yang datang pada aplikasi client.



Gambar 3.17 *Form* layar gambar (gambar negatif)

3.3.3.4 Form Konfigurasi

Form untuk mengatur konfigurasi ukuran buffer dan frame rate.



Gambar 3.17 *Form* konfigurasi

BAB IV

IMPLEMENTASI PERANGKAT LUNAK

BAB IV

IMPLEMENTASI PERANGKAT LUNAK

Berdasarkan perancangan aplikasi telah dijelaskan pada bab 3, maka pada bab ini akan dilakukan pembuatan aplikasi dengan mengacu pada perancangan sistem yang sudah ada. Pembuatan aplikasi terbagi menjadi 2 yaitu aplikasi server dan aplikasi client.

4.1. IMPLEMENTASI APLIKASI SERVER

Di dalam aplikasi server terdapat beberapa modul proses yang perlu diperhatikan secara detail. Modul – modul tersebut antara lain:

1. Proses memulai Socket server
2. Proses pengambilan gambar dari *webcam*
3. Proses penerimaan SMS
4. Proses validasi data SMS.
5. Proses dial up internet
6. Proses pengiriman SMS
7. Proses parsing parameter client-server
8. Proses pengiriman gambar

Berikut ini akan dijelaskan tentang proses – proses yang telah disebutkan sebelumnya:

4.1.1 Implementasi Proses Memulai Socket Server

Proses untuk memalai socket server menggunkan kontrol Winsock dengan prosedur sebagai berikut.

```
Public Sub Start_Socket()
    Winsock1.LocalPort = 6777
    Winsock1.Listen
    tcpClient.LocalPort = 6778
    tcpClient.Listen
    Timer1.Enabled = False
    Clipboard.Clear
End Sub
```

Spesifikasi penggnaan port adalah sebagai berikut:

1. Port 6777 : untuk pengiriman string gambar
2. Port 6778 : untuk parsing parameter client server

4.1.2 Implementasi Proses Pengambilan Gambar dari Webcam

Proses ini adalah melakukan pengambilan frame dari webcam, menyimpannya ke buffer, kemudian menyimpannya kedalam file gambar tipe *JPEG* didalam sebuah direktori. Untuk kebutuhan pengambilan gambar dari kamera menggunakan package *video capture* dari Python. Fungsi untuk pengambilan gambar disimpan dalam register sebagai kelas. Berikut adalah *script* dalam file *cap6.py* dan disimpan dalam kelas *Cap6*:

```
import pythoncom
import base64
class cap6:
        def __init__(self):
                from VideoCapture import Device
                self.c=Device()
        _public_methods_ = [ 'contoh','take_img' ]
        _reg_progid_ = "cap6.gambar"
        _reg_clsid_ = pythoncom.CreateGuid()
        def take_img(self,foutput):
                img=self.c.getImage()
                img.resize((320,240))
                img.save(foutput)
                f=open(foutput,"rb")
                f.flush()
                hasil=f.read()
                return hasil
if __name__ == '__main__':

        print "Registering COM server ..."
        import win32com.server.register
        win32com.server.register.UseCommandLine(cap6)
```

Pada waktu start up program kelas *Cap6.gambar* harus dipanggil untuk pengenalan device webcam. Kemudian setiap melakukan pengambilan gambar pada aplikasi server dilakukan pemanggilan fungsi `uloJPG.take_img` dengan return value berupa string JPG hasil snap shot dari webcam.

```
Set uloJPG = CreateObject("cap4.gambar")

Do While not x <> ""
    DoEvents
   x = uloJPG.take_img("c:\\tmp_sent.jpg")
Loop
```

Setiap kali melakukan pemanggilan fungsi *cap4*, maka dihasilkan file "*tmp_sent.jpg*" yang baru.

4.1.3 Implementasi Proses Penerimaan SMS

Proses ini adalah menunggu pesan masuk ke handphone atau berfungsi sebagai *SMS Gateway*. Pada proses ini dimulai dengan membuka koneksi ke COMM tempat kabel data *handphone* berada.

PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

```
Public Function SetPort(Port As Byte) As Boolean
      SetPort = True
       MSComm1.CommPort = Port
       If MSComm1.PortOpen = False Then
            MSComm1.PortOpen = True
            ModemPortStatus = True
            If Err Then
                MSComm1.PortOpen = Not (MSComm1.PortOpen)
                SetPort = False
                Exit Function
            End If
        End If
End Function
```

Kemudian proses dilanjutkan dengan mendownload inbox dari handphone. Perintah yang digunakan adalah perintah "AT+CMGL=0", yaitu perintah melihat semua surat tak terbaca, dalam variabel strOut. Fungsi yang digunakan adalah sebagai berikut

```
Public Function Download(strOut As Variant, Baudrate As Variant) As Variant
         Dim SendString$, RecieveString$, Dummy As Variant
         SendString$ = LTrim$(strOut) + vbCr

         MSComm1.Settings = Baudrate & ",N,8,1"
         MSComm1.RTSEnable = True
         MSComm1.RThreshold = Len(SendString$)
         MSComm1.InBufferCount = 0
         MSComm1.Output = SendString$

         Do
           Dummy = DoEvents()
           If MSComm1.InBufferCount > 0 Then
               RecieveString$ = RecieveString$ + MSComm1.Input
               Download = RecieveString$
           End If
         Loop  Until  InStr(RecieveString$,  "OK")  Or  InStr(RecieveString$,
"ERROR")

End Function
```

Keluaran dari fungsi diatas adalah string mode PDU. Kemudian string PDU ini di decode oleh algoritma dan disimpan ke dalam *buffer* berupa grid.

```
RecordToGrid (frmSMS.Download("AT+CMGL=0", 19200))
```

Hasil dari fungsi ini adalah grid sudah terisi oleh komponen-komponen SMS yaitu tanggal, no pengirim, dan isi pesan. Untuk melakukan pengecekan secara terus menerus, maka fungsi ini diulang terus menerus menggunakan timer.

```
Do While selesai
   RecordToGrid (frmSMS.Download("AT+CMGL=0", 19200))
   i = 1
   If GridSMS.Rows > 1 Then
      If Validasi_Isi (GridSMS.TextMatrix(i, 0)) then
            koneksi_modem Trim(GridSMS.TextMatrix(i, 0))
      end if
   End If
Loop
```

4.1.4 Implementasi Proses Validasi Data SMS

Selama pengulangan proses fungsi *RecordToDb* program melakukan proses validasi terhadap isi SMS dengan memanggil fungsi *Validasi_Isi*. Fungsi ini menggunakan *keyword* = "1111" untuk kode aktivasi modem dari user.

```
Public Function Validasi_Isi(strInput as String) as Boolean
   If trim(strInput) = "1111" then
      Validasi_Isi = True
   Else
      Validasi_Isi = False
   Endif
End Function
```

4.1.5 Implementasi Proses Dial up Internet

Setelah melakukan validasi kemudian dilakukan proses dialing ke internet Tahap pertama dari proses ini adalah memeriksa koneksi ke internet dan provider yang terdaftar di sistem. Tahap ini dilakukan oleh fungsi `koneksi_modem` seperti dibawah ini:

```
Sub koneksi_modem(noTujuan As String)
   Dim CurrIP As String
   If fInet.Connected Then
      CurrIP = Winsock1.LocalIP
      Call frmSMS.Kirim(noTujuan, "62855000000", CurrIP)
   Else
      If lstConnections.ListCount = 0 Then
         Pesan = "Sistem belum memiliki provider yang terdaftar"
         Call frmSMS.Kirim(noTujuan, "62855000000", Pesan)
         Exit Sub
      End If
      lstConnections.ListIndex = 0
      Call dialing (noTujuan)
   End If
End Sub
```

Terminologi dari fungsi `koneksi_modem` adalah

3. Jika komputer sudah terkoneksi ke internet maka panggil fungsi `frmSMS.Kirim` untuk mengirim pesan IP komputer.
4. Jika tidak ada daftar koneksi ke provider tertentu pada komputer maka panggil fungsi `frmSMS.Kirim` untuk mengirimkan pesan koneksi gagal
5. Jika komputer belum dial up dan ada daftar koneksi maka panggil fungsi `dialing`

```
Private Sub dialing(noTujuan As String)
   Dim plResult As Long
   plResult = fInet.StartDUN(Me.hWnd,lstKoneksi.List(lstKoneksi.ListIndex))

   If plResult = 0& Then
      Call frmSMS.Kirim(noTujuan, "62855000000", Winsock1.LocalIP)
   Else
      If plResult = -1 Then
         Call AddToStatus("Already connected.")
      Else
         Pesan = "dial up internet gagal"
         Call frmSMS.Kirim(noTujuan, "62855000000", pesan)
      End If
   End If
End Sub
```

Jika proses dialing berhasil maka komputer akan memanggil fungsi `frmSMS.Kirim` untuk mengirimkan IP hasil dialing ke user. Jika gagal akan mengirimkan pesan error.

4.1.6 Implementasi Proses Pengiriman SMS

Proses pengiriman SMS digunakan untuk merespon SMS dari user yang berisi pesan aktivasi modem. Jika kode aktivasi benar sistem megirimkan alamat IP yang didapatkan setelah *dial up*. Fungsi untuk pengiriman SMS ini memanggil parameter berupa no tujuan, sms center, isi pesan.

```
Public Function Kirim(strKpd As String, smsc As String, strIsi As String) As
Integer
```

Parameter-parameter fungsi:

1. *strKpd* : adalah nomor pnsel user yang melakukan request
2. *smsc* : nomor call center dari SIM Card modem yang digunakan
3. *strIsi* : isi pesan yang dikirim.

Didalam fungsi tersebut terdapat algoritma untuk encoding parameter-paramater diatas kedalam *PDU mode*. Pengiriman SMS menggunakan perintah "AT+CMGS=".

4.1.7 Implementasi Proses Parsing Parameter Client-Server

Setelah user mendapatkan alamat IP server malalui SMS, kemudian user melakukan request koneksi ke server dan diterima server.

```
Private Sub Winsock1_ConnectionRequest(ByVal requestID As Long)
    Winsock1.Accept requested
```

Setelah koneksi berhasil dibangun, client dapat mengirimkan perintah-perintah yang didefinisikan sebagai berikut:

1. *screen* : untuk memulai pengiriman gambar
2. *scrend* : untuk mengakhiri pengiriman gambar

```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
    Dim data As String
        Winsock1.GetData data
        If data = "SCREEN" Then
        EndScreen = False
        cmdSend_Click
    End If
        If data = "SCREND" Then
        EndScreen = True
    End If
End Sub
```

Sedangkan respon dari server terhadap perintah-perintah client adalah sebagai berikut:

1. *exit* : Untuk konfirmasi bahwa client sudah bisa mengakhiri program
2. *putus* : Untuk konfirmasi bahwa client sudah bisa memutus koneksi
3. *ScreenError* : Pemberitahuan terjadi kegagalan dalam pengambilan gambar

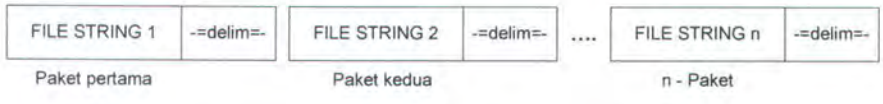
Port yang digunakan untuk parsing parameter ini menggunakan kontrol kontrol Winsock1 dengan port 6777.

4.1.8 Implementasi Proses Pengiriman Gambar

Proses pengiriman gambar dari server diawali dengan pengiriman parameter dari client untuk mengirimkan gambar yaitu *"screen"*. Setelah itu dilakukan pemanggilan fungsi `uloJPG.take_img` yang merupakan fungsi dari objek cap6 untuk mendapatkan string gambar dari webcam yang disimpan dalam fungsi `StrGambar`.

```
Function StrGambar() as String
    StrGambar = uloJPG.take_img("c:\\tmp_sent.jpg")
end sub
```

Kemudian dilakukan proses pemaketan data. Sebuah frame gambar terdiri dari string file dan string delimiter sebagaimana pada gambar 4.1.



Gambar 4.1 Skema pemaketan file gambar

- File String : Hasil encode file *JPG* ke string

- delimiter : -=delim=-

Pengiriman frame dilakukan secara simultan secara terus menerus dengan interval waktu 0.5 detik. Berikut adalah penjelasan mengenai proses pengiriman frame.

Pengiriman data dilakukan oleh prosedur `kirim_gambar` yang melakukan pengiriman data string satu frame yaitu: `StrGambar & "-=EndF=-"`

```
Sub kirim_gambar()
    sockByte.SendData StrGambar & "-=EndF=-"
End Sub
```

Prosedur kirim_gambar dipanggil didalam prosedur `Timer1_Timer` yang merupakan prosedur berulang setiap interval waktu 0.5 detik.

```
Private Sub Timer1_Timer()
    kirim_gambar
    If EndScreen = True Then
        Winsock1.SendData "SCREND"
    End If
End Sub
```

4.2. IMPLEMENTASI APLIKASI CLIENT

Di dalam aplikasi client terdapat beberapa proses yang perlu diperhatikan secara detail. Modul – modul tersebut antara lain:

1. Proses koneksi ke server
2. Proses konfigurasi
3. Proses penerimaan gambar dari server

Berikut ini akan dijelaskan tentang proses – proses yang telah disebutkan sebelumnya:

4.2.1 Implementasi Proses Koneksi ke Server

Proses koneksi ke server Koneksi terdiri dari permintaan koneksi dari user ke server dan memutuskan koneksi ke server.

- Proses koneksi ke server

```
Winsock1.RemoteHost = ip.Text
Winsock1.RemotePort = 6777
Winsock1.connect
tcpServer.RemoteHost = ip.Text
tcpServer.RemotePort = 6778
tcpServer.connect
```

- Proses memutuskan koneksi dari server

```
Winsock1.Close
sockFile.Close
```

4.2.2 Implementasi Proses Konfigurasi

Konfigurasi yang diatur adalah buffer size dan frame rate yang disimpan dalam variabel-variabel sebagai berikut:

```
Type config
    cRate As Long
    cBuff As Integer
End Type
NO_BUFFER
```

Input nilai variabel-variabel adalah sebagai berikut

1. cRate
   - Maksimum : 500 ms
   - Medium : 1000 ms
   - Minimum : 5000 ms
2. cBuff
   - Maksimum : 5 frame
   - Medium : 10 frame
   - Minimum : 20 frame
3. NO_BUFFER

   Bernilai benar jika tidak menggunakan buffer

4.2.3 Implementasi Proses Penerimaan Gambar dari Server

Data gambar yang diterima oleh client adalah data string yang ditangkap prosedur even `DataArrival` pada Winsock. String yang ditangkap pada even ini bukanlah string yang utuh sebagai sebuah frame gambar, melainkan paket-paket string yang besarnya tidak lebih besar dari 8192 byte sehingga perlu proses rekonstruksi atas paket-paket string tersebut menjadi frame-frame yang berurutan. Ada dua algoritma dalam rekonstruksi file ini, yaitu penggabungan paket-paket string menjadi sebuah string frame dan pemisahan antar frame. Algoritma ini diletakkan didalam prosedur even `DataArrival`.

- Penggabungan paket-paket menjadi satu frame gambar

```
sockFile.GetData sdata, vbString

    TotalData = TotalData & sdata
    i = InStr(1, TotalData, "-=EndF=-")
    If i <> 0 Then
        On Error GoTo localerr
        EndOF = True
        varobj = Split(TotalData, "-=EndF=-")
        a1 = varobj(0)
        a2 = varobj(1)
        GoTo nextStat
    End If
localerr:
    a2 = ""
    GoTo nextStat
nextStat:
       -- isi buffer jika EndOF ---
```

Penggabungan dilakukan ketika parameter `"-=EndF=-"` pada fungsi `split(sData,"-=EndF=-")` tidak ditemukan. Dilakukan ketika terbaca pada paket terdapat header file yang dikenali dari string delimiter.

- Isi buffer dan tampilkan frame

Ada dua pilihan cara menampilkan frame gambar yang diterima, yaitu menampilkan per frame gambar secara langsung dan menyimpan frame ke buffer terlebih dahulu baru ditampilkan setelah buffer penuh.

```
If EndOF Then
    If Not NO_BUFFER Then
'================ menggunakan Buffer ================
    Tmp_Str = pyUtil.gen_arr(a1)
    frmViewer.EndScreen = False
    TotalFrame = TotalFrame + 1
'================ menggunakan Buffer ================
    Else
'================ tanpa Buffer ======================
        frmViewer.status.Caption = "Status: Now playing.... "
        Dim s() As Byte
        ReDim s(Len(a1))
        s = StrConv(a1, vbFromUnicode)
        Open "tmp_recv.jpg" For Binary As #1
           Put #1, 1, s
        Close #1
        Set frmViewer.picture1.Picture = LoadPicture("tmp_recv.jpg")
'================ tanpa buffer ======================
    End If
        EndOF = True
        varobj = Split(TotalData, "-=EndF=-")
        a1 = varobj(0)
        a2 = varobj(1)
        GoTo nextStat
End If
```

Untuk menampilkan menggunakan buffer dilakukan di dalam even prosedur Timer yang dengan tiap cRate

```
Private Sub Timer1_Timer()
    Timer1.Interval = konf.cRate
    DoEvents
    If Not EndScreen Then
        Dim s() As Byte
        On Error Resume Next
        If UBound(Tmp_Str) < konf.cBuff Then
            status.Caption = "Status: Collecting...."
        Else
            s = StrConv(Tmp_Str(0), vbFromUnicode)
            Open "tmp_recv.jpg" For Binary As #1
                Put #1, 1, s
            Close #1
            frmViewer.status.Caption = "Status: Now playing.... "
            Set picture1.Picture = LoadPicture("tmp_recv.jpg")
            Tmp_Str = pyUtil.pop_arr(0)
            Frmmainw.TotalFrame = Frmmainw.TotalFrame - 1
        End If
    End If
End Sub
```

BAB V

UJI COBA DAN EVALUASI

BAB V

UJI COBA DAN EVALUASI

Sedangkan pada bab ini akan dibahas mengenai uji coba terhadap sistem yang telah dibuat.

5.1 DESKRIPSI *HARDWARE* DAN *SOFTWARE* UJI COBA

Uji coba terhadap sistem yang telah dibuat dilakukan dengan menggunakan lingkungan dengan spesifikasi sebagai berikut:

Spesifikasi sistem:

- Microsoft Windows XP Profesional
- Default dial up koneksi internet menggunakan Telkomnet@instan

Spesifikasi hardware:

- Pentium III 800 Mhz
- RAM 512 MB
- 40 GB Hardisk
- FaxModem 3Com US Robotics
- Siemens C45
- Orite I-Cam.Speed
- Kabel data serial
- Nokia 2280i CDMA

5.2 SKENARIO UJI COBA

Pada uji coba yang dilakukan ini akan diberikan beberapa skenario uji coba untuk mengetahui fungsionalitas-fungsionalitas per bagian dari sistem.

Uji coba dilakukan untuk menguji jalannya program mulai dari pemasukan input sampai output dihasilkan. Selain itu uji coba juga digunakan untuk menguji kesesuaian perangkat lunak dengan fitur ingin dibuat.

Berikut ini 4 skenario yang akan diujikan yaitu:

1. Uji Coba Skenario Penerimaan SMS
2. Uji Coba Skenario Dial Up Internet
3. Uji Coba Skenario Pengambilan Gambar dari Webcam
4. Uji Coba Skenario Pengiriman Gambar

5.2.1 Uji Coba Skenario Penerimaan SMS

Pada ujicoba yang kedua dilakukan skenario pengiriman SMS ke sistem. Setelah menerima SMS, sistem akan mengolah SMS yang diterima menyimpannya ke dalam grid. Parameter keberhasilan ujicoba adalah tanggal, no pengirim, dan isi pesan dari sms yang datang tersimpan di grid. Tahapan-tahapan dalam ujicoba ini adalah:

- Instalasi hanphone ke komputer server.

Pemasangan handphone Siemens C45 ke komputer menggunakan kabel data serial dan dipasang di port COM1 pada PC.

- Penerimaan SMS oleh program

Untuk tahapan ini dijalankan sub program service SMS. Service ini adalah untuk memulai *SMS Gateway*. Setelah dijalankan program akan secara simultan

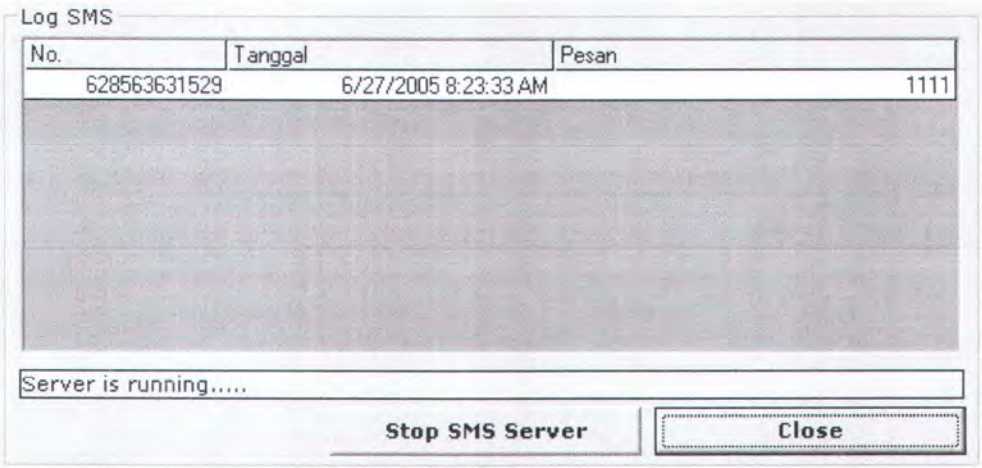
membaca inbox dari SMS. Inisialisasi *incoming message* adalah munculnya surat tak terbaca pada inbox, yang langsung didecode program menjadi string dengan format sebagai berikut:

[tanggal] : *mm/dd/yyyy hh:mm:ss PM/AM*

[no pengirim] : 62xxxxxxxx

[pesan] : string

Masing-masing string disimpan ke dalam grid pada *form Service SMS.*



Gambar 5.1 Hasil decode SMS masuk di grid

Evaluasi Terhadap Skenario Penerimaan SMS

Gambar 5.6 menunjukan bahwa SMS masuk disimpan sementara didalam grid. Dengan demikian ujicoba penerimaan SMS berjalan dengan baik.

5.2.2 Uji Coba Skenario Dial Up Internet

Pada ujicoba yang ketiga dilakukan skenario dial up ke internet menggunakan modem setelah melakukan validasi isi SMS. Tahapan-tahapan dalam ujicoba ini adalah:

1. Instalasi modem ke komputer server.

   Pemasangan modem ke komputer pada port COM2 pada PC.

2. Menyediakan koneksi ke jaringan *Dial Up Networking*.

   Dalam ujicoba ini yang digunakan adalah DISP yang digunakan adalah Telkomnet.

3. Mengirimkan kode aktivasi ke sistem

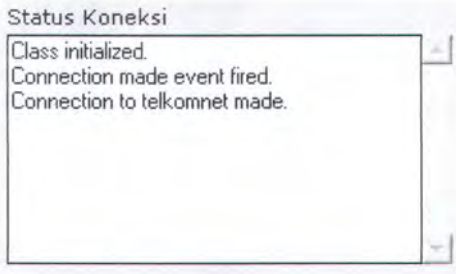
   Pengiriman kode aktivasi dilakukan oleh user melalui pesan SMS ke nomer handphone di komputer server.

4. Dial Up Internet oleh program

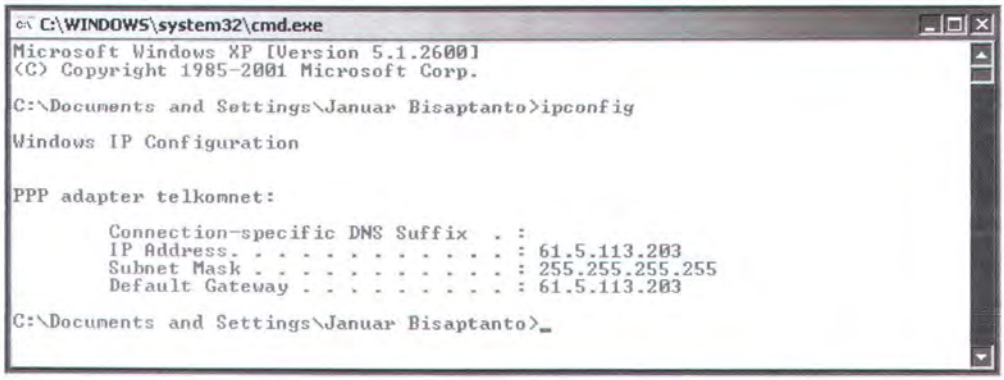
   Untuk tahapan ini dijalankan di *form* service SMS. Proses *dial up* dilakukan setelah proses validasi terhadap *keyword* di Grid SMS berhasil. Pada saat proses dialing, akan muncul default dialog *Dial-up Connection* seperti yang ditunjukkan pada gambar 5.7

Gambar 5.2 *Form* dialog pada proses dialing

Jika proses dialing berhasil maka di listbox status koneksi akan muncul pesan seperti pada gambar 5.8 dan komputer akan mendapatkan IP dinamik dari Telkomnet yang ditunjukkan pada gambar 5.9.

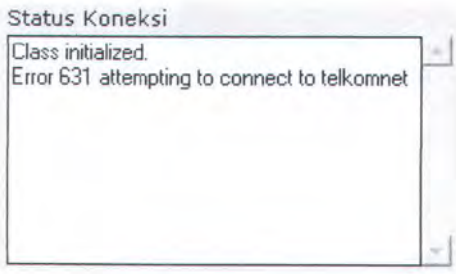


Gambar 5.3 Status koneksi berhasil



Gambar 5.4 IP dinamik dari DISP pada komputer

Jika koneksi gagal maka di listbox status koneksi akan muncul pesan seperti pada gambar 5.5



Gambar 5.5 Status koneksi gagal

5. Pengiriman IP hasil *dial-up* oleh server

Setelah proses dial up berhasil, system kemudian mengirimkan hasil *dial-up* ke user melalui SMS. Ada tiga maca, pesan yang dikirimkan ke user yaitu:

Pesan berupa nomor IP computer server dari provider, yang didapatkan melalui proses dial up atau atau jika computer sudah dalam kondisi terkoneksi ke ke DISP.

Gambar 5.6 IP hasil dial up dikirim melalui SMS

Pesan error, yaitu ketika proses dialing gagal

Gambar 5.6 Pesan dial up gagal

Evaluasi Terhadap Skenario Dial Up Internet

Hasil ujicoba menunjukan bahwa proses dialing berhasil dilakukan dan dengan parameter kebeehasilan adalah nomor IP dari DISP yang diterima oleh user melalui SMS.

5.2.3 Uji Coba Skenario Pengambilan Gambar dari Webcam

Pada uji coba ini yang pertama kali ini akan dilakukan pengambilan gambar dari webcam. Hasil dari skenario ini adalah file gambar *jpg* yang didapatkan dari proses *capturing* secara simultan dari webcam. Tahapan-tahapan dari ujicoba ini adalah:

4. Instalasi PC Camera beserta drivernya.

   Hal yang dilakukan pertama kali agar ujicoba ini berhasil adalah memastikan hardware yang dibutuhkan sudah terinstall ke Windows. Instalasi ini dapat menggunkan driver yang disediakan oleh produk *Orite I-Cam Speed* itu sendiri

5. Pengambilan gambar oleh program

   Untuk tahapan ini dijalankan sub program service kamera. Service ini adalah untuk menampilkan image video pada window. Ujicoba pengambilan gambar dilakukan selama 5 kali dalam interval waktu 500 ms dan secara simultan disimpan ke dalam file *tmp.jpg*.

Gambar 5.7 File *tmp.jpg* pada 0.5 detik pertama

Gambar 5.8 File *tmp.jpg* pada 0.5 detik kedua

Gambar 5.9 File *tmp.jpg* pada 0.5 detik ketiga

Gambar 5.10 File *tmp.jpg* pada 0.5 detik keempat

Gambar 5.11 File *tmp.jpg* pada 0.5 detik kelima

Evaluasi Terhadap Skenario Pengambilan gambar

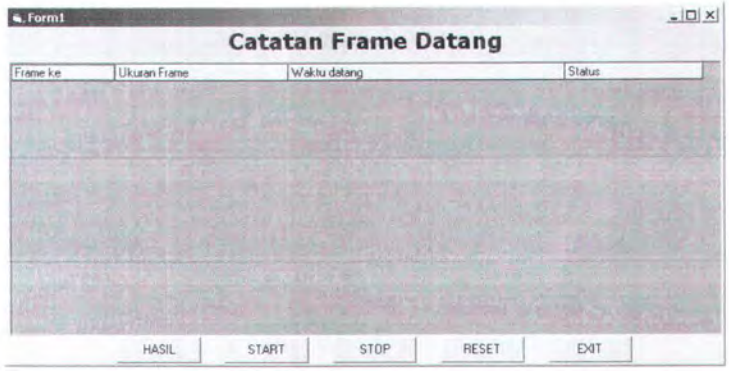
Dari hasil uji coba ini memperlihatkan pengambilan gambar dari kamera berjalan dengan baik.

5.2.4 Uji Coba Skenario Pengiriman Gambar

Pada ujicoba yang keempat dilakukan skenario pengiriman gambar dari komputer server ke komputer client atau user. Tahapan dari ujicoba ini adalah yang pertama adalah mejalankan sistem dalam lingkungan yang ideal yaitu komputer client dan server berada dalam satu lingkup Local Area Network. Kemudian hasil ujicoba di dalam LAN akan dijadikan tolak ukur pengukuran performa sistem saat melakukan ujicoba berikutnya yaitu komputer client dan server berada di lingkup di internet.

Dalam pelaksanaan ujicoba digunakan software pendukung yaitu *form* ujicoba pada program server dan client yang berisi grid untuk mencatat tiap frame yang dikirim dan diterima. Data pada grid ditransfer ke MS Excel untuk direkap sebagai data hasil ujicoba.

Gambar 5.12 *Form* pencatat frame terkirim



Gambar 5.13 *Form* pencatat frame datang

PERPUSTAKAAN INSTITUT TEKNOLOGI SEPULUH - NOPEMBER

5.2.4.1 Ujicoba pada LAN

Ujicoba dilakukan dalam domain AJK-LAB dengan menggunakan dua komputer sebagai client dan server.

Pelaksanaan Ujicoba

1. Data 20 frame yang dikirimkan oleh program server

Tabel 5.1 Tabel hasil pengiriman gambar dalam LAN

Frame ke	Ukuran Frame	Waktu kirim	Status
1	18787	5:56:03 AM	Completed
2	18660	5:56:04 AM	Completed
3	18682	5:56:05 AM	Completed
4	18960	5:56:05 AM	Completed

5	19220	5:56:06 AM	Completed
6	19127	5:56:06 AM	Completed
7	19328	5:56:07 AM	Completed
8	19899	5:56:08 AM	Completed
9	20029	5:56:08 AM	Completed
10	20555	5:56:09 AM	Completed
11	20876	5:56:09 AM	Completed
12	21661	5:56:10 AM	Completed
13	19737	5:56:10 AM	Completed
14	20023	5:56:11 AM	Completed
15	20045	5:56:11 AM	Completed
16	20269	5:56:12 AM	Completed
17	20039	5:56:13 AM	Completed
18	21469	5:56:13 AM	Completed
19	20733	5:56:14 AM	Completed
20	20261	5:56:14 AM	Completed

2. Data 20 frame yang diterima oleh program client

Tabel 5.2 Tabel hasil penerimaan gambar dalam LAN

Frame ke	Ukuran Frame	Waktu kirim	Status
1	18787	5: 56:05 PM	Completed
2	18660	5: 56:06 PM	Completed
3	18620	5: 56:06 PM	Completed
4	18960	5: 56:07 PM	Completed
5	19220	5: 56:09 PM	Completed
6	19127	5: 56:10 PM	Completed
7	19127	5: 56:10 PM	Completed
8	19751	5: 56:11 PM	Completed
9	20029	5: 56:11 PM	Completed
10	20224	5: 56:12 PM	Completed
11	20876	5: 56:12 PM	Completed
12	20876	5: 56:13 PM	Completed
13	21151	5: 56:13 PM	Completed
14	19737	5: 56:14 PM	Completed

15	20023	5: 56:14 PM	Completed
16	20053	5: 56:15 PM	Completed
17	20269	5: 56:15 PM	Completed
18	20269	5: 56:16 PM	Completed
19	21284	5: 56:16 PM	Completed
20	21649	5: 56:17 PM	Completed

3. Bandwith yang diperlukan

Tabel 5.3 Tabel perhitungan bandwith pada LAN

Frame ke	Ukuran Frame	Kb/s
1	18787	9292.5
2	18660	9330
3	18682	18682
4	18960	9480
5	19220	6406.7
6	19127	4781.8
7	19328	6442.7
8	19899	6633
9	20029	6676.3
10	20555	6851.7
11	20876	6958.7
12	21661	7220.3
13	19737	6579
14	20023	6674.3
15	20045	6681.7
16	20269	6756.3
17	20039	10019.5
18	21469	7156.3
19	20733	10366.5
20	20261	6753.7

Perkiraan bandwith yang diperlukan sistem untuk berjalan ideal adalah adalah: 79871.5 Bps atau 80 Kbps

5.2.4.2 Ujicoba pada internet

Ujicoba dilakukan dalam dengan menggunakan dua komputer sebagai client dan server yang terhubung dalam jaringan internet. Ujicoba dilakukan dalam dua rentang waktu yang berbeda, yaitu:

a. Pukul 11.00 – 13.00 WIB

b. Pukul 00.00 – 01.00 WIB

Ujicoba ini juga memperhitungkan biaya yang diperlukan sistem jika dijalankan di kedua waktu diatas. Dan efisiensi dari segi biaya dalam hal pemilihan produk untuk koneksi ke internet baik baik dari sisi client maupun server. Produk dial up koneksi internet yang digunakan dalam ujicoba adalah Telkomnet Instan dan Telkomnet@Flexi. Data awal dari produk diatas adalah (didapat dari http://www.telkom.net/):

1. Telkomnet Instan

   Tarif Rp. 165,00 per menit

   Bandwith yang disediakan 56 Kbps

2. Telkomnet@Flexi

   Tarif Rp 5,00 per Kb

   Bandwith yang disediakan 30 hingga 70 Kbps.

Ujicoba ini menggunakan koneksi internet Telkomnet@Instan di komputer server dengan FaxModem 3Com US Robotics dan dan Telkomnet@Flexi dengan ponsel Nokia 2280i CDMA di komputer client.

1. Pukul 11.00 – 13.00 WIB

   - Pengiriman gambar pada komputer server

Tabel 5.4 Tabel hasil pengiriman gambar melalui internet 1

Frame ke	Ukuran Frame	Waktu kirim	Status
1	62857	12:10:40 PM	Completed
2	62668	12:10:40 PM	Completed
3	62414	12:10:41 PM	Completed
4	62478	12:10:41 PM	Completed
5	62416	12:10:42 PM	Completed
6	62677	12:10:42 PM	Completed
7	62754	12:10:43 PM	Completed
8	62665	12:10:43 PM	Completed
9	62478	12:10:44 PM	Completed
10	62325	12:10:44 PM	Completed

   - Penerimaan gambar pada komputer client

Tabel 5.5 Tabel ujicoba penerimaan gambar melalui internet 1

Frame ke	Ukuran Frame	Waktu terima	Status
1	62857	12:10:55 PM	Completed
2	62668	12:11:27 PM	Completed
3	62414	12:12:04 PM	Completed
4	62478	12:12:35 PM	Completed
5	62416	12:12:59 PM	Completed
6	62677	12:13:24 PM	Completed
7	62754	12:13:49 PM	Completed
8	62665	12:14:15 PM	Completed
9	62478	12:14:42 PM	Completed
10	62325	12:15:17 PM	Completed

➢ Transfer rate tiap frame

Tabel 5.6 Tabel perhitungan bandwith pada internet 1

Frame ke	Ukuran Frame	Kb/s
1	62857	4190.467
2	62668	935.3433
3	62414	751.9759
4	62478	548.0526
5	62416	455.5912
6	62677	386.8951
7	62754	337.3871
8	62665	295.5896
9	62478	262.5126
10	62325	228.2967

Total paket yang diterima adalah : 625.732 b atau 625,732 Kb

Rata-rata kecepatan pengiriman adalah : 8.392,111 bps atau 8.4 Kbps

Rasio dari kondisi ideal : 8.4 / 80 x 100%= 10.5 %

Total biaya penerimaan menggunakan Telkom Flexi : 625,732 x Rp. 5, = Rp. 3.128,66

2. Pukul 00.00 – 01.00 WIB

➢ Pengiriman gambar pada komputer server

Tabel 5.7 Tabel hasil pengiriman gambar melalui internet 2

Frame ke	Ukuran Frame	Waktu kirim	Status
1	48533	00:16:22 PM	Completed
2	49682	00:16:22 PM	Completed
3	51051	00:16:23 PM	Completed
4	50621	00:16:23 PM	Completed
5	49769	00:16:24 PM	Completed
6	50961	00:16:24 PM	Completed

7	50392	00:16:25 PM	Completed
8	50914	00:16:25 PM	Completed
9	50043	00:16:26 PM	Completed
10	50014	00:16:26 PM	Completed

- Penerimaan gambar pada komputer client

Tabel 5.8 Tabel ujicoba penerimaan gambar melalui internet 2

Frame ke	Ukuran Frame	Waktu terima	Status
1	48533	00:16:33 PM	Completed
2	49682	00:16:54 PM	Completed
3	51051	00:17:10 PM	Completed
4	50621	00:17:31 PM	Completed
5	49769	00:17:52 PM	Completed
6	50961	00:18:17 PM	Completed
7	50392	00:18:34 PM	Completed
8	50914	00:18:51 PM	Completed
9	50043	00:19:11 PM	Completed
10	50014	00:19:30 PM	Completed

- Transfer rate tiap frame

Tabel 5.9 Tabel perhitungan bandwith pada internet 2

Frame ke	Ukuran Frame	Kb/s
48533	48533	4412.091
49682	49682	1552.563
51051	51051	1086.191
50621	50621	744.4265
49769	49769	565.5568
50961	50961	450.9823
50392	50392	390.6357
50914	50914	348.726
50043	50043	303.2909
50014	50014	271.8152

Total paket yang diterima adalah : 501.980 b atau 501,980 Kb

Rata-rata kecepatan pengiriman adalah : 10.126,28 bps atau 10,1 Kbps

Rasio dari kondisi ideal : 10,1 / 80 x 100 % = 12,6%

Total biaya penerimaan menggunakan Telkom Flexi : 501,980 x Rp. 5,00 = Rp. 2.509,90

Evaluasi Terhadap Skenario Pengiriman gambar

Berikut adalah perbandingan kecepatan dari ketiga ujicoba diatas

- LAN : 80 Kb/s
- Internet
  - Pukul 13.00 – 13.00 WIB : 8.4 Kb/s

    Rasio dari kondisi ideal : 10.5 %
  - Pukul 00.00 – 01.00 WIB : 10,1 Kbps

    Rasio dari kondisi ideal : 12,6%

Hasil ujicoba memperlihatkan bahwa transfer gambar pada jaringan internet menggunakan provider Telkomnet dan Flexi tidak berjalan ideal. Karena hal-hal sebagai berikut

a. Bandwith yang disediakan oleh DISP yang menyediakan koneksi internet pada sistem tidak mencapai bandwith yang dibutuhkan sistem secara ideal.
b. Waktu. Sistem berjalan optimal jika dijalankan pada waktu malam hari, yaitu dimana jalur internet tidak begitu padat.

BAB VI

PENUTUP

BAB VI

PENUTUP

Bab ini menyimpulkan hasil uji coba yang telah dilakukan. Selanjutnya diberikan beberapa saran yang mungkin dapat dijadikan pertimbangan untuk mengembangkan hasil yang diperoleh pada tugas akhir ini.

6.1. KESIMPULAN

Setelah dilakukan serangkaian uji coba dan analisa terhadap perangkat lunak yang dibuat, maka dapat diambil kesimpulan sebagai berikut:

1. Berdasarkan uji coba yang telah dilakukan, aplikasi yang telah dibuat dapat digunakan untuk mengirimkan streaming gambar sehingga dapat melihat kondisi rumah.
2. Pengiriman gambar pada aplikasi memiliki performa yang berbeda-beda bergantung pada hal-hal sebagai berikut:
    a. Bandwith yang disediakan oleh DISP yang menyediakan koneksi internet pada sistem
    b. Waktu. Sistem berjalan optimal jika dijalankan pada waktu malam hari, yaitu dimana jalur internet tidak begitu padat.

6.2. SARAN

Berdasarkan hasil evaluasi yang dilakukan terhadap sistem, saran yang perlu dipertimbangkan dalam pengembangan aplikasi ini, yaitu peningkatan kompresi tiap frame gambar yang dikirim perlu dilakukan untuk mengurangi beban di internet sehingga pengiriman tiap frame lebih cepat.

DAFTAR PUSTAKA

DAFTAR PUSTAKA

[1] Franklin, Carl, *Visual Basic Internet Programming*, Canada : John Wiley & Sons Inc, 1996.

[2] Will Barden, *Winscok Complete Reference*, http://www.winsockvb.com/, 2002.

[3] Gritsch, Markus, *A Win32 Python Extension for Accessing Video Devices*, http://videocapture.sourceforge.net/

[4] Fernando, Thushan, *Controlling Dial Up Networking using the WinInet API*, http://www.developerfusion.co.uk/show/1920/

[5] *Using the Communications Control*, Microsoft Corporation, http://msdn.microsoft.com/library/, 2005.

[6] Lundh, Fredrik and Matthew Ellis, *Python Imaging Overview*, www.pythonware.com/products/pil/pil-handbook.pdf, 2002

[7] Marcello Scatà, *SMS PDU-Mode*, http://www.gsmworld.it/, 1997-2002

[8] Kabeen, Ken, *Image Compression and Discrate Cosine Transform*, http://online.redwoods.cc.ca.us/instruct/ darnold/laproj/Fall98/PKen/dct.pdf

[9] *Image Compression Comparison & Image Transform Coding for JPEG Image Compression Algorithm*, http://www.dspexperts.com/dsp/projects/dctcomp/