

AUTOMATICS SOFTWARE ANALISYS

Arief Setyanto

Dosen STMIK AMIKOM Yogyakarta

Abstract

Software engineering is one of engineering area. As engineering area software engineering has special differences with other. Software engineering result is an abstract product. It's very difficult to measure the volume because there was no raw material used. The only measurement is the usage for users. Sometime a short software is very usefull for user so the value is very high on the other hand a software with 1000 times longer in code is not usefull enough for the same users so it doesn't have any value. As engineering process it has to be measured. This paper try to study how to measure the metrics of software and then propose automation tools for software analist.

Keywords: *Software engineering, software metrics, software analisys, automation software analisys.*

Pendahuluan

Rekayasa perangkat lunak merupakan salah satu disiplin rekayasa yang memberikan hasil abstrak. Kondisi ini berbeda dengan disiplin rekayasa yang seperti elektronika mesin maupun sipil dimana hasil proses rekayasa memberikan bentuk nyata. Pengukuran volume sebuah hasil rekayasa yang abstrak menjadi tidak semudah pengukuran volume hasil rekayasa yang memberikan hasil konkret. Pengukuran volume pekerjaan rekayasa perangkat lunak yang abstrak hanya didasarkan kepada kegunaan dari hasil rekayasa. Volume dalam ukuran byte panjangnya kode program terkadang tidak linier dengan kegunaan dari hasil rekayasa perangkat lunak.

Bahan baku yang relatif tidak diperlukan pada rekayasa perangkat lunak juga turut berperan dalam mempersulit pengukuran volume pekerjaan. Sementara rekayasa di bidang lain salah satu ukuran yang dapat dijadikan pedoman adalah jumlah bahan baku yang digunakan.

Sifat sifat perangkat lunak diatas menyebabkan pengukuran volume pekerjaan menjadi lebih sulit dilakukan. Akibatnya adalah perkiraan biaya pengembangan perangkat lunak juga akan menjadi tidak mudah dilakukan. Makalah ini akan mencoba memberikan satu alternatif analisis biaya pengembangan perangkat lunak yang akan dicoba di automasi. Dokumen awal yang dijadikan pedoman adalah dokumen *View Point Oriented Document* (VORD). Dokumen VORD merupakan hasil kerja dari seorang analis dalam mendefinisikan kebutuhan fungsional dari stake holder yang terdiri dari para calon pemakai perangkat lunak.

Pembahasan

1. Permasalahan

a. Analisis Kebutuhan

Pekerjaan yang memiliki posisi yang cukup menentukan dalam urutan pekerjaan pembangunan perangkat lunak adalah proses analisis kebutuhan pengguna. Rekayasa kebutuhan (*requirement engineering*) merupakan tahapan pengumpulan kebutuhan layanan dari software yang akan dibangun oleh masing masing pengguna yang terlibat dalam sebuah sistem.

b. View Point Oriented Document (VORD)

Salah satu bentuk dokumen yang dapat digunakan untuk mencatat kebutuhan pengguna adalah VORD. Pembuatan dokumen ini diawali dengan mengoleksi user user yang akan berperan dalam menjalankan sistem perangkat lunak yang akan di bangun. Setelah mengumpulkan daftar pengguna yang selanjutnya akan disebut *view point*, kepadanya akan didokumentasikan service apa saja yang akan

diberikan oleh software terhadap view point tersebut. Selanjutnya semua service dan semua view point akan diberikan definisi detailnya.

Penggunaan dokumen ini dapat membantu analisis untuk mengerjakan tugas rekayasa kebutuhan pengguna dengan lengkap. Walaupun pada prakteknya proses ini tidak selalu berjalan dengan baik karena ada saja beberapa service yang tercecer tidak ter-amati oleh analis. Kejadian ini merupakan sesuatu yang biasa terjadi dalam rekayasa perangkat lunak seperti disebutkan dalam beberapa buku teori. Adapun set dokumen VORD contohnya adalah sebagai berikut :

Tabel 1 Daftar View Point

Pelanggan
Petugas Pencatat Meter Air
BANK
Petugas Bank

Pelanggan	Petugas Catat Meter air	Admin BANK	Kasir Bank
<ul style="list-style-type: none"> • Mendapat info tagihan • Mendapat peringatan ketika terlambat • Membayar tagihan 	<ul style="list-style-type: none"> • Mencatat meter air • Melaporkan catatn meter air • Mndapat jadwal kunjungan • Mendapat Honor 	<ul style="list-style-type: none"> • Menerima daftar tagihan • Membuat daftar tagihan terbayar • Mengirim daftar tagihan dibayar 	<ul style="list-style-type: none"> • Menerima pembayaran konsumen • Mendapat informasi tagihan pelanggan • Mencetak Slip pembayaran tagihan

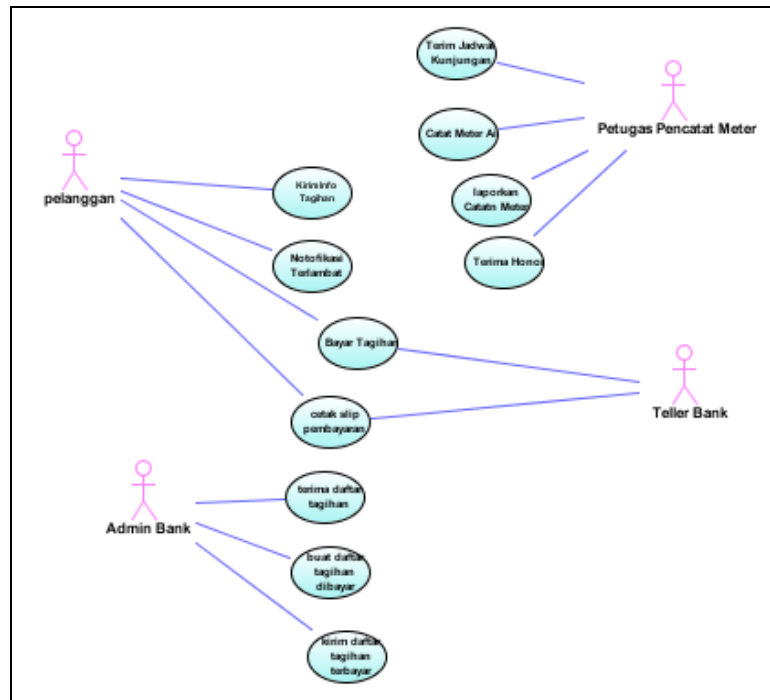
Gambar 1. Daftar Services masing masing View Point

2. Solusi

a. Vord untuk Bahan Membuat Use Case

Salah satu dokumen dalam satu set dokumen UML yang dibuat pada perancangan perangkat lunak berorientasi obyek – Object oriented analysis (OOA) adalah Use case. Dokumen ini mewakili gambaran yang akan diberikan kepada pengguna (AKTOR) tentang apa saja yang dapat dilakukan ketika perangkat lunak yang direncanakan telah dibuat.

Dengan modal dokumen VORD dapat dibuatkan sebuah proses automasi untuk menggenerate secara otomatis dokumen Use Case ini. Cara yang dapat dilakukan adalah dengan mengkonversi view point dalam VORD menjadi aktor. Serta services menjadi use case. Dengan mengubah dokumen VORD menjadi use case maka Service yang melibatkan lebih dari 1 view point hanya akan tergambar menjadi 1 use case saja. Dibawah ini akan disajikan contoh sederhana yang dapat menjadi gambaran proses konversi ini.



Gambar 2. Use Case berdasarkan dokumen VORD

b. Pembobotan Use Case

Untuk keperluan pengukuran volume pekerjaan setiap use case yang muncul diberi bobot. Pembobotan ini menjadi tugas analisis yang diberikan kepada setiap use case. Sebagai angka perkiraan tentunya angka yang disajikan tidaklah sepenuhnya tepat dengan kondisi pada pelaksanaan rekayasa. Sebagai pedoman untuk memudahkan pemberian angka bobot dapat di equivalenkan dengan jam orang kerja (JOK). Misalnya saja terdapat use case menghitung IPK, untuk merealisasikan use case ini diperlukan 10 JOK dari programmer untuk menuliskan program maka use case ini akan diberi bobot 10.

Berdasarkan use case yang ada dalam UML yang telah di otomasi tersebut dapat diperoleh jumlah total JOK. Jumlah jam orang kerja yang di butuhkan inilah yang akan dijadikan ukuran dari pekerjaan (volume pekerjaan). Perlu di ingat bahwa rekayasa perangkat lunak sebagai sebuah pekerjaan dan produk, biaya yang dibutuhkan selama proses rekayasa sepenuhnya akan di jadikan biaya untuk membayar orang dan operasional selama proses itu berlangsung. Tidak ada biaya (jika ada tidak signifikan) yang diperlukan untuk membeli bahan baku.

Hasil pembobotan ini dapat berguna untuk memperkirakan jumlah sumber daya manusia yang dibutuhkan terlibat dalam tim pengembangan perangkat lunak. Pembobotan juga dapat digunakan untuk menghitung biaya produksi yang akan dihabiskan selama proses pembuatan program. Kedua faktor diatas selanjutnya akan menentukan berapa lama waktu yang dibutuhkan untuk membangun perangkat lunak tersebut. Penghitungan kasar ini akan menghasilkan waktu pengembangan dengan mengabaikan faktor urutan pekerjaan dalam jadwal pengembangan perangkat lunak. Contoh perhitungan itemitem pekerjaan yang dicontohkan pada requirement pada gambar 1 adalah sebagai berikut.

Tabel 2. Simulasi Penghitungan Bobot

NO	Service	Bobot
1	Mendapat info tagihan	40
2	Mendapat peringatan ketika terlambat	30
3	Membayar tagihan	50
4	Mencatat meter air	80
5	Melaporkan catatn meter air	50
6	Mendapat jadwal kunjungan	30
7	Mendapat Honor	30
8	Menerima daftar tagihan	90

NO	Service	Bobot
9	Membuat daftar tagihan terbayar	40
10	Mengirim daftar tagihan dibayar	30
11	Menerima pembayaran konsumen	70
12	Mendapat informasi tagihan pelanggan	50
13	Mencetak Slip pembayaran tagihan	50
TOTAL BOBOT PEKERJAAN		640

Angka angka yang tercantum diatas bukanlah harga mati. Pada implementasi metode ini seorang analis harus secara tepat memperkirakan bobot pekerjaan dari masing masing service yang akan di buat dalam sebuah perangkat lunak. Jika angka total bobot pekerjaan telah dihitung, ini dapat dijadikan acuan untuk menghitung secara kasar berapa biaya yang diperlukan untuk membangun perangkat lunak yang sedang dirancang. Angka yang diwakili oleh 1 point pekerjaan akan bervariasi antara 1 perusahaan dengan perusahaan lain yang tergantung dari berapa besar gaji yang diberikan kepada semua pihak yang akan terlibat secara operasional menjalankan proyek tersebut. Misal 1 point pada pekerjaan diatas akan di hargai Rp. 25.000,- maka untuk proyek diatas akan memakan biaya developmen sebesar : 16.000.000,- (enam belas juta rupiah).

c. Otomasi

Proses proses penyusunan dokumen VORD, konversi menjadi use case maupun pembobotan biasanya dilakukan secara manual oleh analis. Permasalahan menjadi cukup rumit ketika terjadi penambahan services, perubahan services maupun penambahan beberapa view point. Kejadian tersebut sangat mungkin terjadi karena teori – always incomplete- memang biasa terjadi di lapangan. Pada kondisi tersebut terpaksa analis akan menghitung dari awal lagi baik kebutuhan SDM, jadwal maupun estimasi biaya. Oleh karena aturan yang sudah cukup jelas, bentuk dokumen juga cukup jelas maka penulis mengusulkan pengembangan tolls untuk melakukan otomasi proses ini.

Melihat susunan dokumen VORD maupun use case maka pada level pencatatan penulis mengusulkan di buat dokumen XML untuk menyimpan dokumen VORD maupun use case. Rencana dokumen XML untuk keperluan tersebut adalah sebagai berikut :

```
<XML
  <VP nama viewpoint
    <Service nama layanan /Service> <bobot  bobot layanan
/bobot>
    <Service nama layanan /Service> <bobot  bobot layanan /bobot>
    <Service nama layanan /Service> <bobot  bobot layanan /bobot>
    .
  /VP>
  <VP nama viewpoint
    <Service nama layanan /Service> <bobot  bobot layanan
/bobot>
    <Service nama layanan /Service> <bobot  bobot layanan /bobot>
    <Service nama layanan /Service> <bobot  bobot layanan /bobot>
    .
  /VP>
/XML >
```

Contoh potongan dokumen XML untuk kasus diatas :

```
<XML
  <VP Pelanggan
    <Service dapat info tagihan /Service> <bobot 40 /bobot>
    <Service Mendapat peringatan ketika terlambat /Service> <bobot
30 /bobot>
    <Service Membayar tagihan /Service> <bobot 30 /bobot>
  /VP>
  <VP petugas meter
    <Service Mencatat meter air/Service> <bobot 80 /bobot>
    <Service Melaporkan catatn meter air /Service> <bobot 50
/bobot>
  /VP>
/XML >
```


User interface di rencanakan menggunakan notasi grafis agar pengguna dapat melakukan aktivitas perancangan seperti ketika menggunakan kertas dan pensil. Representasi output juga diusulkan dalam bentuk grafis.

Penutup

Makalah ini dibuat untuk menyelesaikan salah satu permasalahan permulaan dari proses rekayasa perangkat lunak. Otomasi proses pembuatan dokumen rekayasa ini diharapkan dapat menutup salah satu kelemahan software engineer yang di beberapa tempat sering teledor dalam mendokumentasikan hasil kerjanya. Keteledoran inilah yang kemudian menjadi penyebab mundurnya jadwal penyelesaian pekerjaan, membengkaknya biaya pengembangan dan kesulitan untuk melakukan penelusuran kembali proses rekayasa itu sendiri. Pengembangan ke berikutnya dapat dilakukan untuk melakukan otomasi proses selanjutnya seperti penyediaan dokumen diagram kelas, diagram kolaborasi, dokumen rencana testing dan lain sebagainya. Pengembangan ini diharapkan akan menciptakan tools yang cukup lengkap untuk membantu proses pengembangan perangkat lunak.

Daftar Pustaka

- Sommerville, Ian, Software Engineering practical approach, Prentice hall 2000
- Pressman Roger, Software Engineering, prentice hall
- Shach, Stephen R., Classical and Object Oriented Programming
Software engineering, Mc. Graww Hill, New Jersey, 1999
- Sholiq, Permodelan Sistem Informasi Berorientasi Obyek dengan
UML, Graha Ilmu, 2006

MEMBANGUN APLIKASI CHATING BERBASIS MULTIUSER

Roni Setiawan¹ dan Edhy Sutanta²

^{1,2} Jurusan Teknik Informatika, Fakultas Teknologi Industri,
IST AKPRIND Yogyakarta
Email: edhy_sst@yahoo.com

Abstract

The rapid development of information technology especially of computer technology has many benefits in many areas of human life. Variety and computer software capabilities developed rapidly and increase the value of the benefits of computer technology.

Currently has available several applications software for communication between users in a network. To support the success of the utilization of a local computer network needed an application that can be used for communication between users in the form of text in which the network can be shared.

This research has developed a chat application that is different from what is available so far, for communications services between users who are in a local network. Applications built using Visual Basic 6.0 programming language. Chat feature can use by everyone and private, also facilities to send files that can be operated in a multi-user.

Keywords: *aplikasi chatting, multiuser, Visual Basic 6.0*

Pendahuluan

Teknologi informasi yang tak pernah mundur dalam perjalanannya selalu menghadirkan kemudahan baru bagi umat manusia, termasuk bidang jaringan komputer. Jaringan komputer bukanlah sesuatu yang baru saat ini, hampir di setiap instansi terdapat jaringan komputer untuk memperlancar arus informasi di dalam instansi tersebut. Jaringan komputer adalah sebuah kumpulan

komputer yang saling terhubung sehingga memungkinkan pengguna dapat saling bertukar data dan informasi, juga untuk bertukar sumber daya. Di dalam jaringan terdapat beberapa aplikasi pendukung yang mempermudah para penggunanya. Beberapa contoh aplikasi yang dimaksud di antaranya aplikasi *chatting* yang berguna untuk berkomunikasi, *telnet* sebagai fasilitas *remot login*, *FTP* untuk transfer data dan masih yang lainnya.

Dalam prakteknya pengguna jaringan komputer sering dihadapkan pada masalah komunikasi antar pengguna, misalkan seorang pengguna ingin membagikan informasi kepada pengguna lain yang berada di tempat berbeda maka dibutuhkan sebuah aplikasi komunikasi atau *chat*. Keberadaan aplikasi tersebut dapat menunjang suksesnya penggunaan sebuah jaringan komputer di dalam sebuah instansi. Aplikasi *chatting* digunakan sebagai media komunikasi antar sesama pengguna komputer yang terhubung dalam suatu jaringan, baik itu berupa melalui teks, *image*, ataupun suara. Dalam kenyataannya, aplikasi *chatting* yang tersedia biasanya hanya dapat digunakan oleh dua pengguna saja, dan apabila ada penyampaian informasi antar pengguna, maka harus dilakukan berulang pada pengguna yang lain, oleh sebab itu dibutuhkan sebuah aplikasi *chatting* yang bersifat *multiuser*.

Penelitian ini akan mengembangkan aplikasi *chatting* yang dapat dipakai secara bersama-sama dalam satu *list room* pada suatu jaringan komputer. Adapun tujuan yang ingin dicapai dalam penelitian ini adalah memberikan solusi alternatif penggunaan aplikasi *chatting* yang mempermudah komunikasi secara bersama-sama antar pengguna jaringan. Harapannya, aplikasi yang dibuat akan memberikan manfaat antara lain memudahkan penyampaian informasi kepada pengguna lain dalam sebuah jaringan yang dapat bekerja secara praktis, karena tidak perlu repot dengan penyampaian informasi yang berulang-ulang.

Pembuatan program aplikasi *chatting multiuser* dalam penelitian ini menggunakan seperangkat komputer dengan spesifikasi *motherboard* Asrock Intel Series 3.3e, *processor* Intel pentium 4

2.24GHZ, *memory* Mc Pro 256MB, *harddisk* Seagate 40MB, VGA Geforce 4000.

Pengumpulan data dilakukan dengan menggunakan metode observasi, studi kepustakaan, dan eksperimen. Pembuatan program aplikasi dilakukan melalui tiga tahapan utama yaitu analisis, desain, dan implementasi sistem. Berdasarkan hasil analisis teridentifikasi masalah bahwa proses komunikasi antar pengguna komputer dalam jaringan LAN dirasa kurang praktis karena aplikasi *chatting* yang tersedia bersifat *user to user* dan belum berbasis *multiuser*. Alternatif solusi yang diusulkan adalah dengan mengembangkan aplikasi *chatting* berbasis *multiuser*. Pengembangan program aplikasi dilakukan menggunakan OS Microsoft windows XP Sp2, dan *programming language* Visual Basic 6.0. Tahap desain dilakukan untuk menggambarkan proses dalam aplikasi yaitu:

Langkah 0 : Mulai

Langkah 1 : Login pengguna

Langkah 2 : Pengguna masuk *Chat rom* atau *Everyone chat*

Langkah 3: Cek mode *private chat*, jika benar pengguna masuk *mode private chat* dan jika tidak dengan tetap berada di menu *chat room* pengguna dapat langsung melakukan *chat* dengan semua pengguna (*EveryOne*)

Langkah 4: Masuk *private chat*, pengguna dapat melakukan *chat* hanya dengan seorang pengguna dan kembali ke menu *chat room*

Langkah 5: Pada *private chat* pengguna dapat melakukan transfer data melalui menu *Send file*

Langkah 6: *Browse file*

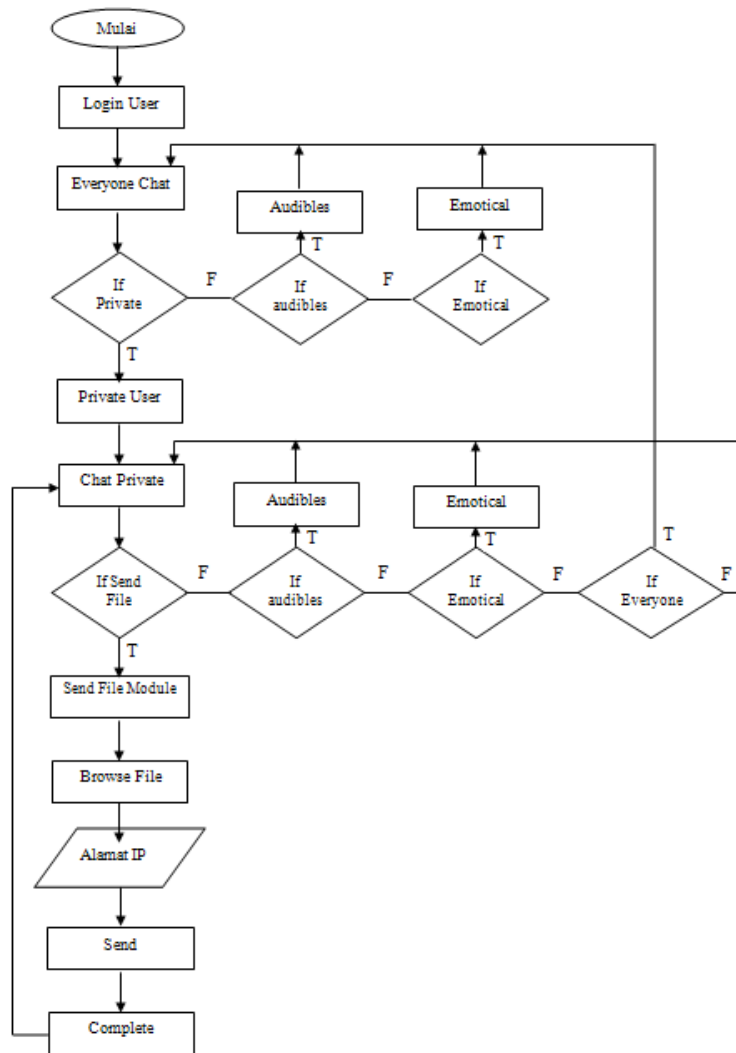
Langkah 7: Memasukan alamat IP

Langkah 8: *Send file*

Langkah 9 : *Complete*

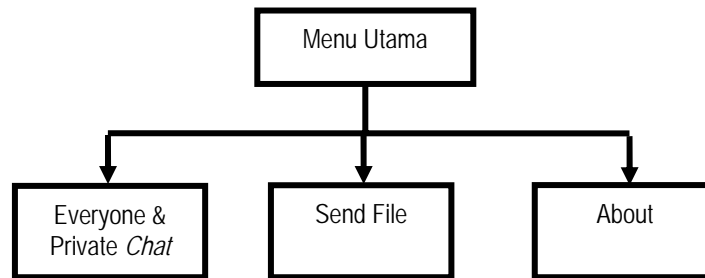
Langkah10: Kembali ke menu *private chat*

Diagram alir (*flow chart*) aplikasi tampak pada Gambar 1.



Gambar 1: Desain struktur menu aplikasi

Desain struktur menu program aplikasi tampak pada Gambar 2.



Gambar 2: Desain struktur menu aplikasi

1. LAN (Local Area Network)

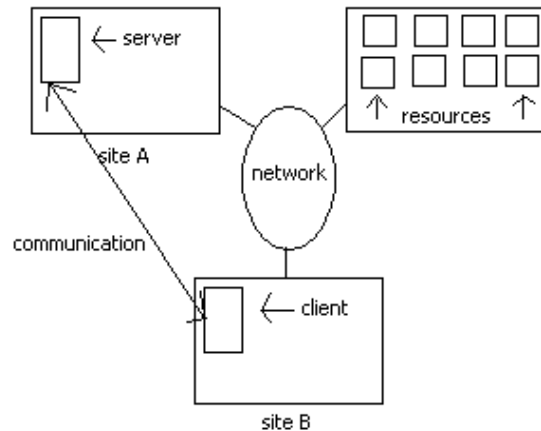
Istilah *Local Area Network* (LAN) muncul pada awal tahun 1970-an sebagai pengganti dari sistem komputer *mainframe*. LAN adalah sebuah jaringan lokal yang digunakan untuk menghubungkan komputer *server* dengan komputer *client* dalam suatu wilayah jaringan dan mempunyai pengalamatan terhadap jaringan tersebut. LAN didesain untuk jangkauan area geografis yang kecil, misalnya, LAN digunakan untuk jaringan dalam sebuah bangunan atau beberapa bangunan yang berdekatan. Pengalamatan pada LAN disebut dengan IP (*Internet Protocol*). Dalam IP *networking*, sebuah LAN umumnya terbentuk dalam sebuah IP *subnet*. Contoh IP *address* pada sebuah LAN adalah sebagai berikut:

IP address = 192.168.0.10 (menunjukkan jenis IP *class C*)

Network Mask = 255.255.255.0

Subnet = 255.255.255.224 (menunjukkan *subnet*)

Ilustrasi LAN tampak seperti Gambar 3.



Gambar 3: Ilustrasi *Local Area Network*

Umumnya, jarak antara unit terhubung dalam LAN saling berdekatan. Oleh karena itu, kecepatan komunikasinya lebih tinggi dan peluang terjadi kesalahan (*error rate*) lebih rendah. LAN membutuhkan *high quality cable* agar memiliki kecepatan yang tinggi dan reliabilitas tercapai. Jenis kabel yang biasanya dipakai adalah *twisted-pair* dan *fiber-optic*. LAN juga memiliki corak spesifik, biasanya dikendalikan, dimiliki, dan dikelola oleh sebuah organisasi atau *operator* tunggal. LAN juga menggunakan konektivitas spesifik, yaitu *ethernet* dan *token ring*.

2. Visual Basic

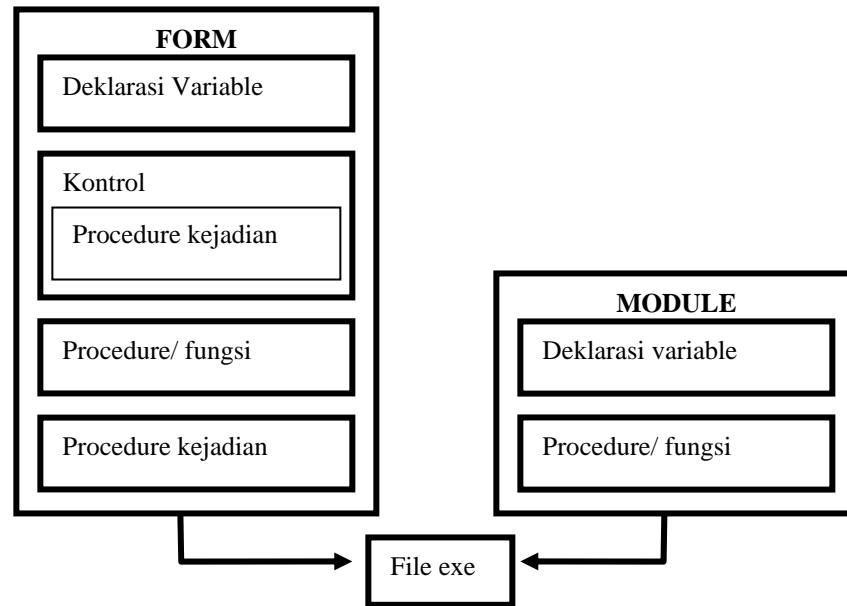
Basic merupakan bahasa pemrograman yang banyak digunakan oleh *programmer* pemula. Bahasa ini mudah digunakan dan tidak banyak ketentuan yang mengikat, dibandingkan bahasa prosedural seperti Bahasa C atau Pascal. Pada Visual Basic perancangan aplikasi dimulai dari mendefinisikan tujuan program, merancang keluaran sebagai media komunikasi dengan pengguna, dan menuliskan kode programnya (Putra, 2004).

Pemrograman dengan Visual Basic banyak menggunakan istilah obyek. Obyek-obyek digunakan pada *layer* untuk melakukan pengaturan properti terhadap obyek yang digambarkan. Pada saat program dijalankan, dituliskan metode-metode terhadap obyek tersebut sesuai dengan tujuan program. Untuk membuat sebuah program aplikasi dengan Visual Basic, dimulai dengan membuat *form* terlebih dahulu, kemudian dibuat file dan modul lain. Setelah komponen dipadukan dan kode selesai ditulis, dilanjutkan dengan membuat proyek menjadi file yang dapat dieksekusi (Nalwan, 2004).

Beberapa kelebihan Visual Basic 6.0 dibandingkan dari versi-versi sebelumnya adalah (Nalwan, 2004):

1. Merupakan bahasa pemrograman *event-driven* yang berasal dari bahasa Basic. Makna istilah *event-driven* adalah program menunggu sampai adanya respon dari pengguna berupa kejadian tertentu, misalnya tombol diklik, atau menu dipilih ketika *event* terdeteksi, *event* yang berhubungan akan melakukan aksi sesuai kode yang diberikan.
2. Disertai berbagai sarana untuk membuat aplikasi *database* dan sarana visual yang menjadikannya yang terbaik untuk mengembangkan aplikasi *client* dan *server*.
3. Memiliki tambahan sarana *wizard*, yaitu sarana yang mempermudah dalam pembuatan aplikasi dengan mengotomatisasi tugas-tugas tersebut.
4. Memiliki *compiler* handal yang dapat menghasilkan *executable file* yang lebih cepat dan efisien dari yang sebelumnya.
5. Kecepatan pengembangan aplikasi.

Pembuatan program aplikasi dengan Visual Basic menggunakan dua tipe kode sumber, yaitu *form* dan *module*. Suatu program aplikasi dapat disusun dari beberapa *form* dan *module*. Struktur kode pembentuk program aplikasi dengan Visual Basic tampak pada Gambar 4.



Gambar 4: Struktur program dengan Visual Basic

Visual Basic menyediakan beberapa kontrol standar yang digunakan dalam pembuatan program aplikasi, antara lain *command button*, *label*, *text box*, *frame*, *combo box*, *listbox*, *checkbox*, *option button*, *picture box*, *common dialog*. Deklarasi variabel dan array dalam Visual Basic dapat dilakukan menggunakan *dim*, *private statement*, *public statement* (Pamungkas, 2001).

Pembahasan

1. Menu Utama

Aplikasi yang dikembangkan dalam penelitian ini memiliki sebuah menu utama yang di dalamnya terdapat beberapa sub menu yang diharapkan mempermudah pengguna melakukan komunikasi melalui *chatting*. Pada saat aplikasi dijalankan, sebelum masuk ke

menu utama, aplikasi menampilkan *form* konfirmasi untuk mengisi sebuah *nick name* yang digunakan untuk menampilkan status pengguna siapa saja yang sedang *online*. Tampilan konfirmasi *nick name* tampak pada Gambar 5.



Gambar 5: Tampilan konfirmasi *nick name*

Setelah pengguna mengisi *nick name* dengan benar, baru kemudian muncul tampilan menu utama, dan pengguna dapat langsung menggunakan fasilitas *chat* yang di dalamnya menyediakan menu *private* atau *everyone chat*, *emotical chat*, *send file*, dll. Tampilan menu utama aplikasi tampak seperti Gambar 6.



Gambar 6: Tampilan menu utama

Untuk mengirimkan pesan *text* digunakan *code editor* yang menggunakan fungsi *if....then....else* dan *while*. Berikut potongan *code editor* untuk mengirimkan pesan text:

```
Private Sub sck1_DataArrival(ByVal bytesTotal As Long)
On Error Resume Next
Dim medata As String
Dim j As String
'ambil data
sck1.GetData medata
```

```
merlin.Speak medata
'tampilkan suara
WindowsMediaPlayer1.URL = "Whistleu.wav"
If Left$(medata, 2) = "@@" Then
ListNick.AddItem Replace(medata, "@@", "")
ListIp.AddItem sck1.RemoteHostIP

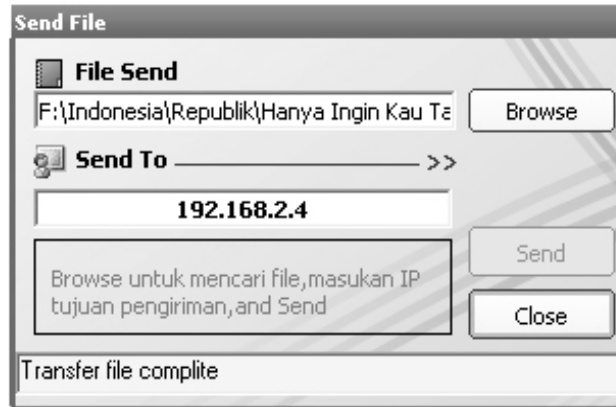
j = 0
Do While j < ListIp.ListCount
    ListIp.Text = ListIp.List ( j )
    If ListIp.ListIndex <> j Then
        ListIp.RemoveItem j
        ListNick.RemoveItem j
    Else
        j = j + 1
    End If
Loop
medata = Replace(medata, "@@", "")
'cek data jika pengguna online
medata = medata & " Logged on." & vbCrLf
'tampilkan font
rt.SelStart = Len(rt.Text)
rt.SelColor = vbRed
rt.SelFontName = "tahoma"
rt.SelItalic = False
rt.SelBold = True
rt.SelText = medata
'tampilkan nama
sck1.SendData "^*" & nick
'everyone nama and ip
ElseIf Left$(medata, 2) = "^*" Then
    If medata = "^*" & nick Then
        Exit Sub
    Else
```

```
ListNick.AddItem Replace(metadata, "^*", "")
ListIp.AddItem sck1.RemoteHostIP
'bersihkan lagi tampilan listboxes
j = 0
Do While j < ListIp.ListCount
    ListIp.Text = ListIp.List(j)
    If ListIp.ListIndex <> j Then
        ListIp.RemoveItem j
        ListNick.RemoveItem j
    Else
        j = j + 1
    End If
Loop
End If
'pemberitahuan ket jika pengguna log Off
'hapus nama dan IPaddress
ElseIf Left$(metadata, 2) = "!%" Then
    Dim X As Long
    For X = 0 To ListIp.ListCount - 1
        If sck1.RemoteHostIP = ListIp.List(X) Then Exit For
    Next X
    If X = ListIp.ListCount Then Exit Sub
    ListIp.RemoveItem X
    ListNick.RemoveItem X
'menampilkan keterangan
'bila seseorang akan keluar (off)
'tampilkan keterangan keluar pada txtdialog
metadata = Replace(metadata, "!%", "")
metadata = metadata & " Logged off." & vbCrLf
rt.SelStart = Len(rt.Text)
rt.SelColor = vbRed
rt.SelFontName = "tahoma"
rt.SelItalic = False
rt.SelBold = True
```

```
rt.SelText = medata
Else
'tampilan text jika pengguna online!
'warna text dalam txtDialog.
rt.SelStart = Len(rt.Text)
rt.SelColor = vbBlue
rt.SelFontName = "tahoma"
rt.SelItalic = False
rt.SelText = medata
End If
Sub cmdKirim()
merlin.Show
'menampilkan warna text
rt.SelStart = Len(rt.Text)
rt.SelBold = True
rt.SelFontName = "Tahoma"
rt.SelColor = vbBlack
rt.SelText = nick & ": " & txtSend.Text & vbCrLf
merlin.Speak txtSend.Text & vbCrLf
On Error Resume Next
    ListIp.Selected(ListNick.ListIndex) = True
    sck1.RemoteHost = ListIp.Text
'mengirimkan text
If Left$(txtSend.Text, 2) = "#$" Then
    txtSend.Text = Replace(txtSend.Text, "#$", " ")
    sck1.SendData "$Message from: " & nick & vbCrLf &
vbCrLf & txtSend.Text
Else
    sck1.SendData nick & ": " & txtSend.Text & vbCrLf
End If
DoEvents
'menghapus text
txtSend.Text = " "
End Sub
```

2. Menu *Send File*

Menu *Send File* digunakan untuk mengirimkan file atau data dari seorang pengguna ke pengguna lainnya. Cara pengirimannya adalah dengan melakukan *browse* pada *form Send file*, kemudian mengisi alamat IP pada *textbox Send to*, setelah itu klik tombol *Send*. Saat file atau data telah berhasil dikirim maka akan muncul *notification* pada *textbox* paling bawah seperti Gambar 7.



Gambar 7: Tampilan menu *Send File*

Menu *Send File* dibuat dengan menggunakan fungsi *if....then....else*. Berikut potongan *code editor* dalam *form* menu *Send File*:

```
'Menentukan tempat atau lokasi file yang ditransfer akan diletakkan  
Private Sub wsTCP_DataArrival(Index As Integer, ByVal bytesTotal As Long)  
    Dim temp As String  
    Dim bOK As Boolean
```

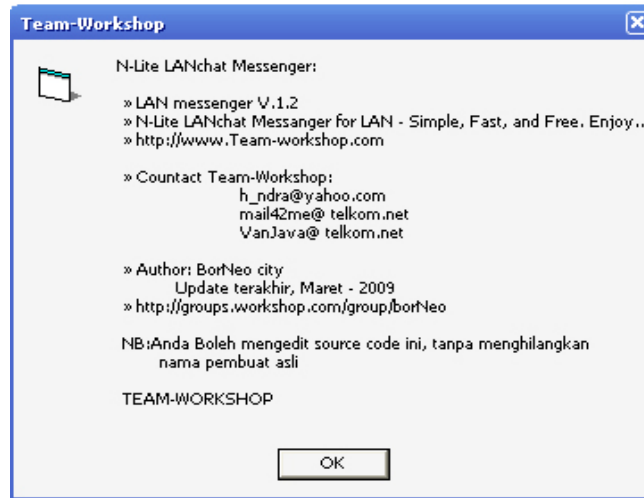
```
Dim Fname As String
If Not bOK Then
    wsTCP(1).GetData Fname
If InStr(Fname, vbCrLf) <> 0 Then Fname = Left(Fname,
InStr(Fname, vbCrLf) - 1)
    bOK = True
    If Dir(Dir1.Path & "\" & Fname) <> "" Then Kill
Dir1.Path & "\" & Fname
    Open TempPath & "transfer\" & Fname For Binary As
1
    l Pos = 1
    wsTCP(1).SendData "OK" & vbCrLf
Else

Dim buffer( ) As Byte
    wsTCP(1).GetData buffer
    Put #1, lPos, buffer
    lPos = lPos + UBound(buffer) + 1
End If
End Sub
'mengirimkan file
Private Sub cmdsendfile_Click()
Dim buffer( ) As Byte
    cmdsendfile.Enabled = False
    l Bytes = 0
ReDim buffer(FileLen(dlg.FileName) - 1)
    Open dlg.FileName For Binary As 1
    Get #1, 1, buffer
    Close #1
    Load wsTCP(1)
    wsTCP(1).RemoteHost = txtIP.Text
    wsTCP(1).RemotePort = 1111
    wsTCP(1).Connect
    lblStatus = "Connecting..."
```


End Sub

3. Menu About

Menu *About* berisi penjelasan singkat tentang aplikasi *chatting* yang dikembangkan, mulai dari spesifikasi program hingga alamat *email* yang bisa dihubungi. Tampilan menu *About* tampak pada Gambar 8.



Gambar 8: Tampilan menu *About*

Penelitian ini berhasil membangun sebuah program aplikasi *chatting* dapat menjadi solusi komunikasi antar pengguna dalam sebuah jaringan secara *multiuser*. Aplikasi yang dikembangkan juga dilengkapi dengan menu *Send File* untuk mengirimkan *text* yang mempermudah komunikasi antar pengguna yang berdaa dalam sebuah jaringan.

Kekurangan program aplikasi *chatting* yang dikembangkan adalah tidak ada pemisahan *form* antara *private chat* dengan *everyone*

chat. Model pengiriman *text* melalui menu *Send File* masih harus memasukan alamat IP tujuan pengguna sehingga dinilai masih kurang praktis. Dan, program aplikasi hanya dapat berjalan pada sistem operasi Ms Windows.

Penutup

Berdasarkan hasil penelitian ini, dapat diambil kesimpulan sebagai berikut:

1. Program aplikasi *chatting* yang dikembangkan dapat menjadi alternatif solusi media komunikasi bagi para pengguna yang berada dalam sebuah jaringan lokal.
2. Program aplikasi yang bersifat *multiuser* mempermudah seorang pengguna melakukan komunikasi dengan banyak pengguna.
3. Fasilitas *Send File* dalam program aplikasi dapat membuat komunikasi antar pengguna menjadi lebih efektif dan praktis, dan dapat mendukung pengguna dalam menyelesaikan pekerjaanya.

Perbaikan atas program aplikasi yang dikembangkan dapat dilakukan dengan menambahkan menu *history* agar pengguna dapat melihat *history* aktifitas *chatting*, dan mekanisme yang mempermudah penentuan tujuan pengiriman file.

Daftar Pustaka

- Cisco System, Inc., 2005, *IP Addressing and Subnetting for New User*, Update: 25 September 2005, ID 13788.
- Nalwan, A., 2004, *Membuat Program Profesional Secara Cepat dengan VB*, Gramedia, Jakarta.
- Putra, I., 2004, *Membangun Aplikasi Nyata Dengan Visual Basic 6.0*, Penerbit Andi, Yogyakarta.
- Rafiudin, R., 2006, *Sistem Komunikasi Data Muthakhir*, Penerbit Andi, Yogyakarta.
- Silberschatz, A., Galvin, P., dan Gagne, G., 2005, *Operating System Concepts*, 7th Edition, John Wiley & Sons.
- Wardana, 2006, *Pembuatan Control ActiveX di Visual Basic*, buku kedua, Elex Media Komputindo, Jakarta.

Sumber dari *Internet*:

<http://id.wikipedia.org/wiki/Chatting>, diakses tanggal 20 April 2009.

http://id.wikipedia.org/wiki/Jaringan_komputer, diakses tanggal 20 April 2009.

<http://id.wikipedia.org/wiki/Multi-user>, diakses tanggal 20 April 2009.