

**PERANCANGAN DAN PEMBUATAN USER INTERFACE
REMOTE MONITORING TATA SUARA PANGGUNG MENGGUNAKAN
DELPHI 7.0**

SKRIPSI



**Disusun Oleh:
IMAM BASHORI
NIM : 09.12.505**

**INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER
2013**

LEMBAR PERSETUJUAN
PERANCANGAN DAN PEMBUATAN USER INTERFACE
REMOTE MONITORING TATA SUARA PANGGUNG MENGGUNAKAN
DELPHI 7.0

*Disusun dan diajukan untuk melengkapi dan memenuhi persyaratan guna mencapai
gelar Sarjana Teknik Komputer Strata satu (S-1)*

Disusun Oleh :

IMAM BASHORI

09.12.505

Mengetahui,

Ketua Program Studi Teknik Elektro S-1



M. Ibrahim Ashari, ST, MT

NIP.P. 1030100358

Diperiksa dan Disetujui oleh :

Dosen Pembimbing I

A blue ink signature of M. Ibrahim Ashari, ST, MT.

M. Ibrahim Ashari, ST, MT
NIP.P.1030100358

Dosen Pembimbing II

A blue ink signature of Michael Ardita, ST, MT.

Michael Ardita, ST, MT
NIP.P.1031000434

PROGRAM STUDI TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG

2013

PERANCANGAN DAN PEMBUATAN USER INTERFACE REMOTE MONITORING TATA SUARA PANGGUNG MENGUNAKAN DELPHI 7.0

IMAM BASHORI
(09.12.505)

Dosen Pembimbing:
M. Ibrahim Ashari, ST, MT
Michael Ardita, ST, MT

Konsentrasi Teknik Komputer, Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Nasional Malang
Jln. Raya Karanglo Km 2 Malang
Email: el_fazim@yahoo.co.id

ABSTRAK

Delphi secara umum dapat diartikan sebagai sebuah IDE Compiler untuk bahasa pemrograman Pascal dan lingkungan pengembangan perangkat lunak yang digunakan untuk merancang suatu aplikasi program. Delphi juga dapat diartikan sebagai suatu software yang digunakan untuk membuat aplikasi yang berbasis visual.

Aplikasi ini membuat sistem remote monitoring yang dapat memantau speaker yang berada diatas panggung yang bertolak belakang dengan pengontrol dan kalah besar suaranya dengan sound system FOH (Front Of House) yang saling berhadapan dengan pengontrol. Aplikasi ini juga di sertai kemampuan untuk mengatur tampilan tata letak speaker yang berada di atas panggung, sehingga hasil monitoring lebih mudah dan cepat. Oleh karena itu dalam rancangan aplikasi remote monitoring tata suara panggung dibuat dengan tujuan untuk mempermudah pengontrol tata suara panggung dalam mengetahui permasalahan-permasalahan yang terjadi secara langsung dan visual, sehingga mampumeminimalkan terjadinya gangguan pada tata suara panggung. Implementasi aplikasi remote monitoring tata suara panggung ini menggunakan bahasa pemograman Delphi dan Database Access.

Dengan memnfaatkan aplikasi ini dapat menemukan gangguan pada speaker dan aktif tidaknya speaker lebih mudah, yang kondisinya akan ditampilkan berupa level – level speaker sesuai tegangan dari speaker dan menggunakan komunikasi wireless yang memiliki jarak terbatas karena menggunakan modul TLP dan RLP yang maksimal jarak nya 20 meter jika tidak ada penghalang.

Kata Kunci: teknologi komputer, aplikasi monitoring tata suara panggung

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

Nama : Imam Bashori
NIM : 09.12.505
Program Studi : Teknik Elektro S-1
Konsentrasi : Teknik Komputer

Dengan ini menyatakan bahwa Skripsi yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam Skripsi ini tidak memuat karya orang lain, kecuali dicantumkan sumbernya sesuai dengan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat, dan apabila di kemudian hari ada pelanggaran atas surat pernyataan ini, saya bersedia menerima sanksinya.

Malang, 5 Agustus 2013

Yang membuat Pernyataan,



Imam Bashori

NIM : 09.12.505

KATA PENGANTAR

Puji Syukur Kehadirat Allah SWT atas berkat dan rahmatnya, sehingga kami selaku penyusun dapat menyelesaikan Laporan Skripsi ini yang berjudul “**PERANCANGAN DAN PEMBUATAN USER INTERFACE REMOTE MONITORING TATA SUARA PANGGUNG MENGGUNAKAN DELPHI 7.0**” dapat terselesaikan.

Adapun maksud dan tujuan dari penulisan laporan ini merupakan salah satu syarat untuk dapat menyelesaikan studi dan mendapatkan gelar Sarjana Jurusan Teknik Elektro S-1, Konsentrasi Teknik Komputer ITN Malang.

Sebagai penulis sangat menyadari tanpa adanya kemauan dan usaha serta bantuan dari berbagai pihak, maka laporan ini tidak dapat diselesaikan dengan baik. Oleh sebab itu, penyusun mengucapkan terima kasih kepada yang terhormat:

1. Ir. Soeparno Djiwo, MT selaku Rektor Institut Teknologi Nasional Malang.
2. Ir. Anang Subardi, MT selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. M.Ibrahim Ashari,ST, MT selaku Ketua Jurusan Teknik Elektro S-1 Institut Teknologi Nasional Malang, dan selaku Dosen Pembimbing satu Tugas Skripsi
4. Michael Ardhita, ST, MT. selaku Dosen Pembimbing dua Tugas Skripsi.
5. Sahabat-sahabat dan rekan-rekan yang tidak kami sebutkan satu-persatu, kami ucapkan banyak terima kasih atas bantuannya dalam proses pembuatan Skripsi yang telah saya kerjakan, begitu juga dengan penyelesaian laporan ini.

Penulis menyadari bahwa penelitian ini masih jauh dari sempurna, untuk itu kritik dan saran dari semua pihak sangat kami harapkan untuk perbaikan penelitian ini

Malang, 5 Agustus 2013

Penyusun

DAFTAR ISI

LEMBAR PERSETUJUAN	ii
ABSTRAK	iii
SURAT PERNYATAAN	iv
KATA PENGANTAR	v
DAFTAR ISI	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metode Penelitian	3
1.6 Sistematika Penulisan	4
BAB II LANDASAN TEORI	
2.1 Pendahuluan	5
2.2 Borland Delphi 7.0	5
2.2.1 IDE Delphi	6
2.2.2 Menu Delphi	8
2.2.3 Komponen Standart Delphi 7.0	10
2.2.4 Tipe Data	11
2.2.5 Operator	13
2.3 Komunikasi Serial	15
2.3.1 RS-232	15
2.3.2 Vacomm	16

2.4 Distorsi	17
2.5 Tata Suara	17
2.6 Wireless	17
2.7 Mikrokontroler ATmega 8535	19
BAB III PERANCANGAN DAN DESAIN APLIKASI	
3.1 Deskripsi Umum	20
3.2 Analisa Kebutuhan Sistem	20
3.2.1 Analisa Kebutuhan Perangkat Lunak (Software)	21
3.2.2 Analisa Kebutuhan Perangkat Keras (Hardware)	22
3.3 Desain Sistem	22
3.3.1 Gambaran Umum Aplikasi	22
3.3.2 Pemodelan Aplikasi	24
3.3.3 Flowchart	24
3.3.3.1 Flowchat Program Aplikasi	25
3.3.3.2 Flowchat Proses Aplikasi	25
3.4 Perancangan dan pembuatan Desain Aplikasi	26
3.5 Perancangan Sistem Komunikasi	29
3.6 Perancangan Tampilan Program	30
3.6.1 Perancangan Form Setting Komunikasi	30
3.6.2 Perancangan Form Monitoring	31
3.6.3 Perancangan Database	34
3.7 Perhitungan Untuk Mendapatkan Outputan Volt dan dB pada Aplikasi	37
3.8 Perancangan Drag Pada Aplikasi	38

BAB IV IMPLEMENTASI DAN PENGUJIAN APLIKASI

4.1 Spesifikasi Program	40
4.2 Petunjuk Operasional dan Uji Coba	40
4.3 Implementasi	41
4.4 Pengujian hasil aplikasi monitoring	42
4.4.1 Pengujian Komunikasi	42
4.4.2 Pengujian Tampilan Aplikasi Monitoring	45
4.4.3 Pengujian Perhitungan Tegangan (Volt) dan Decibel (dB)	49
4.4.4. Pengujian Aplikasi dengan Speaker di Panggung	50
4.4.5 Pengujian Jarak Pada Hardware	51
4.4.6 Pengujian Drag Speaker	52
4.5 Tabel Hasil Pengujian	53

BAB V PENUTUP

5.1 Kesimpulan	55
5.2 Saran	55

DAFTAR GAMBAR

Gambar 2.1 Jendela Utama Delphi 7.0	6
Gambar 2.2 Object Treeview	6
Gambar 2.3 Object Inspector	7
Gambar 2.4 Form Designer	8
Gambar 2.5 Komponen Vacomm	16
Gambar 3.1 Gambaran Umum Secara Keseluruhan	23
Gambar 3.2 Gambaran Umum Aplikasi Yang Akan Dirancang	24
Gambar 3.3 Flowchat Aplikasi Monitoring	25
Gambar 3.4 Flowchat Proses Aplikasi	26
Gambar 3.5 Setting Komunikasi Hyper Terminal	29
Gambar 3.6 Form Perancangan Komunikasi	30
Gambar 3.7 Rancangan Form Monitoring	31
Gambar 3.8 Menu Managing	33
Gambar 3.9 Menu Aplikasi	33
Gambar 3.10 Menu Available Skin	34
Gambar 3.11 Hasil Perancangan Database dengan Ms. Access	36
Gambar 3.12 Tampilan Database Pada Delphi	37
Gambar 3.13 Tampilan Bar Speaker	38
Gambar 4.1 Pengujian Hper Terminal Tanpa Hardware	42
Gambar 4.2 Komunikasi dengan Hyper Terminal Yang Sudah Terhubung Dengan Hardware	43
Gambar 4.3 Setting Komunikasi Pemilihan COMPORT	44
Gambar 4.4 Setting Komunikasi Penyimpanan COMPORT	44
Gambar 4.5 Tampilan Monitoring Tanpa AlphaSkin	45

Gambar 4.6 Tampilan Monitoring Setelah Menggunakan AplhaSkin	46
Gambar 4.7 Tampilan Jika Tidak ada Komunikasi	47
Gambar 4.8 Tampilan Jika Ada Komunikasi	47
Gambar 4.9 Tampilan Jika Belum Ada Inputan Dari Speaker	48
Gambar 4.10 Tampilan Jika ada Inputan Dari Hardware	48
Gambar 4.11 Tampilan Bar Speaker	49
Gambar 4.12 Pengujian Dengan 4 Speaker	50
Gambar 4.13 Tampilan Monitoring Normal	52
Gambar 4.14 Tampilan Setelah Proses Drag	53

DAFTAR TABEL

Tabel 2.1 Komponen Standart Delphi	10
Tabel 2.2 Tipe Data Text String.....	11
Tabel 2.3 Tipe Data Bilangan Integer	11
Table 2.4 Tipe Data Bilangan Real	12
Table 2.5 Tipe Data Bilangan Boolean	13
Tabel 2.6 Operator Aritmatik	13
Table 2.7 Operator Perbandingan	14
Tabel 3.1 Databese Aplikasi Monitoring	35
Tabel 4.1 Hasil Pengujian Jarak	52
Tabel 4.2 Hasil Pengujian	53

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Suara merupakan kompresi mekanikal atau gelombang yang merambat melalui medium dan bagi yang mempunyai pendengaran normal hampir setiap saat kehidupannya selalu diwarnai oleh berbagai macam suara. Sebagai contoh mendengarkan suara musik melalui *sound system* yang dimiliki.

Saat ini banyak penataan panggung yang tidak memperhitungkan tingkat kejelasan suara dalam sebuah penataan *sound system* yang berada di atas panggung dan yang sering di perhatikan hanya kejelasan *sound Front Of House* (FOH) yang ke audient, maka pengesetan atau pengontrolan dilakukan dengan baik supaya menghasilkan suara yang baik bagi audient dan yang di atas panggung. Dalam acara-acara besar seperti konser banyak hal yang tidak terduga, semakin banyak penonton maka semakin berkurang *Sinyal to Noise Ratio* (SNR) dari suara yang ditimbulkan. Apalagi *sound* yang berada di atas panggung sulit untuk mengetahui gangguan-gangguan yang terjadi karena arahnya saling berlawanan dengan pengontrol dan kalah dengan suara yang ditimbulkan *Front Of House* (FOH). Untuk memastikan kualitas suara yang terjadi pada tata suara panggung, diperlukan ketelitian dalam mencari permasalahan tata suara panggung seperti kualitas suara, kenyaringan suara dan kerusakan suara akibat distorsi.

Dari latar belakang tersebut maka dibuat suatu remote monitoring tata suara panggung. Sistem ini menggunakan bahasa pemrograman Borland Delphi untuk mempermudah pengontrol *sound* yang berada di atas panggung untuk mengetahui permasalahan-permasalahan yang terjadi secara langsung dan visual.

1.2 Rumusan Masalah

1. Bagaimana untuk memonitor kondisi suara di panggung karena kalah kuat dengan *Front Of House* (FOH) yang berhadapan dengan operator.
2. Bagaimana untuk mengamati multikanal secara simultan

1.3 Tujuan dan Manfaat Penelitian

1.3.1 Tujuan

Melakukan monitoring suara yang ada di panggung yang tidak terdengar dari tempat pengontrol

1.3.2 Manfaat

Pembuatan remote monitoring tata suara panggung dapat di manfaatkan untuk :

1. Membantu pengontrol tata suara untuk mencari permasalahan-permasalahan yang ada dengan mudah.
2. Dapat memonitoring kondisi yang tidak terdengar secara langsung karena jarak dan arah yang berlawanan dengan pengontrol.

1.4 Batasan Masalah

- 1 Pada penelitian ini hanya dilakukan pembahasan bagaimana perancangan software remote monitoring tata suara panggung, tidak dibahas tentang pembuatan hardware.
- 2 Hanya memonitor sound yang berada di panggung dan sekitarnya.

1.5 Metodologi Penelitian

Berdasarkan dengan judul karya ilmiah, metodologi penelitiannya adalah sebagai berikut:

1. Ide dan Solusi Umum

Setelah melakukan beberapa tahap pemikiran dalam menciptakan ide karya ilmiah, akhirnya timbul solusi umum, yaitu dengan mencari referensi dari berbagai sumber serta mengumpulkan semua data khususnya dari institusi, yang berkaitan dengan karya ilmiah.

2. Eliminasi Data Kebutuhan

Berdasarkan data yang diperoleh dari berbagai sumber, dilakukan pencarian data yang paling sesuai dan yang berhubungan dengan karya ilmiah sehingga dapat dijadikan acuan yang jelas dalam pembuatan karya ilmiah.

3. Perancangan dan Pembuatan

Setelah mendapatkan acuan data yang jelas, dilakukan perancangan secara umum pada karya ilmiah terlebih dahulu dengan secara menyeluruh, kemudian setelah ditemukan kerangka secara menyeluruh maka dilanjutkan dengan pembuatan karya ilmiah yang sepenuhnya mengacu pada kerangka yang sudah dibuat.

4. Uji Coba dan Koreksi

Tahap ini digunakan untuk uji coba karya ilmiah secara keseluruhan yang sudah dibuat yaitu dengan melihat dari berbagai sisi mulai dari desain yang sesuai sampai kinerja dan fungsi yang tepat. Dari uji coba secara keseluruhan yang dilakukan, maka akan dicari titik lemahnya dan sekaligus akan dilakukan penyempurnaan apabila terdapat kekurangan.

1.6 Sistematika Penulisan

Dalam pembuatan karya ilmiah ini, terdiri dari lima bab dengan sistematika penulisan sebagai berikut:

BAB I : Pendahuluan

Membahas tentang latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian, beserta sistematika penulisan yang berhubungan dengan karya ilmiah.

BAB II : Landasan Teori

Membahas tentang landasan teori yang berkaitan dengan judul karya ilmiah beserta pemaparan komponen-komponen yang digunakan dalam menciptakan karya ilmiah.

BAB III : Perancangan dan Pembuatan Aplikasi

Berisikan tentang pembahasan kebutuhan media yang digunakan beserta kerangka secara global yang menyeluruh dalam pembuatan aplikasi.

BAB IV: Hasil Akhir dan Pengujian Aplikasi

Berisikan tentang hasil akhir dari aplikasi yang telah dibuat, uji coba aplikasi, beserta petunjuk operasional aplikasi.

BAB V: Penutup

Berisikan tentang kesimpulan dari terciptanya aplikasi beserta saran, guna mendapatkan aplikasi yang tepat guna.

BAB II

TINJAUAN PUSTAKA

2.1 Pendahuluan

Pada bab ini akan diterangkan mengenai media dan teori penunjang yang di gunakan untuk pembuatan aplikasi. Pembahasan dalam bab ini meliputi :

1. Borland Delphi 7.0
2. Komunikasi Serial
3. Distorsi
4. Tata Suara
5. Wireless
6. Mikrokontroller atmega8535

2.2 Borland Delphi 7.0

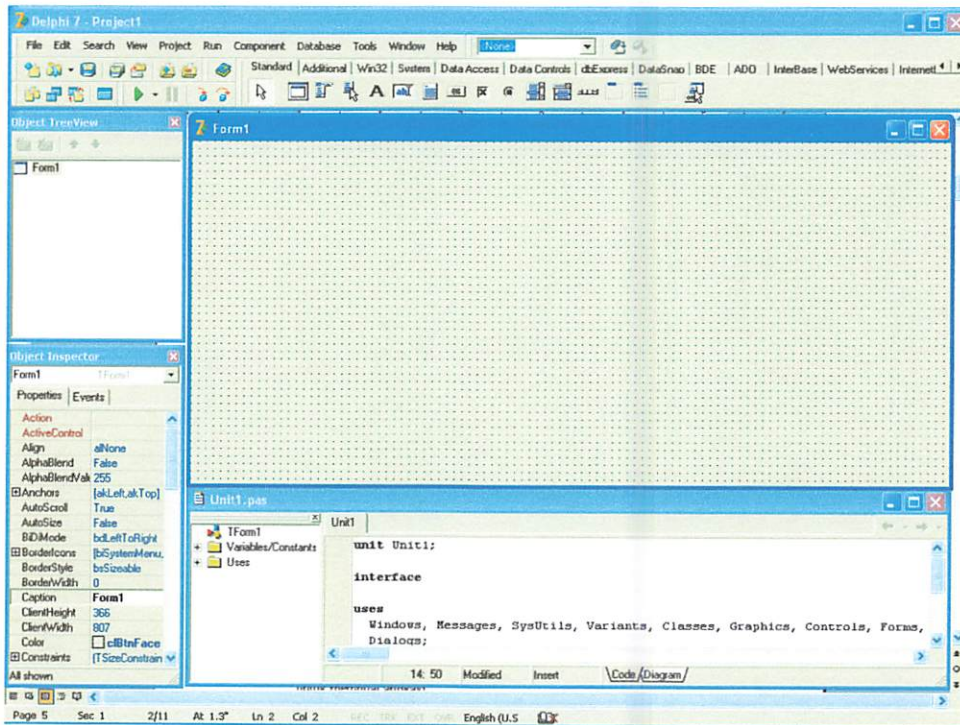
Borland Delphi 7.0 adalah suatu software yang digunakan untuk membuat aplikasi yang berbasis visual. Didalam software ini, listing program menggunakan bahasa pascal tetapi sudah dipadu dengan bahasa pemrograman yang terstruktur sehingga *programmer* dipermudah dalam pembuatan aplikasi, berbeda dengan pendahulu software ini yaitu *Turbo Pascal*.

Selain digunakan untuk membuat aplikasi database, Borland Delphi 7.0 juga dapat digunakan untuk membuat aplikasi kendali maupun monitoring. Selain dapat membuat tampilan aplikasi yang menarik, Borland Delphi 7.0 merupakan software yang mempunyai kemampuan *powerfull* dibandingkan software lain terutama untuk membuat aplikasi kendali.

Delphi sendiri adalah sebuah *software* yang digunakan untuk membuat aplikasi berbasis windows, aplikasi berbasis grafis, aplikasi berbasis jaringan dan aplikasi berbasis internet. Dengan fitur aplikasi visual pada Delphi, aplikasi yang dibangun menggunakan Delphi terlihat lebih menarik. Terdapat juga beberapa keunggulan Delphi diantaranya yaitu proses *compilasi* yang cepat dan kemudahan dalam membuat dan mendesain aplikasi.

2.2.1 IDE Delphi

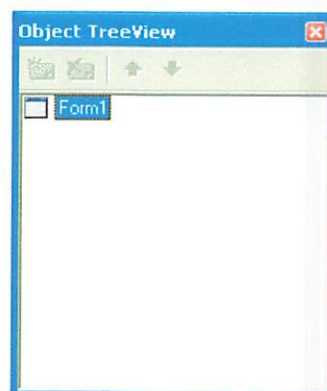
1. Jendela Utama



Gambar 2.1 Jendela Utama Delphi 7.0

Pada jendela utama Borland Delphi terdapat beberapa menu seperti yang terdapat pada aplikasi berbasis windows pada umumnya. Selain itu terdapat beberapa *toolbar* yang berfungsi sebagai *shortcut* menu, dan terdapat juga *component pallette* yaitu sebagai tempat terdapatnya komponen.

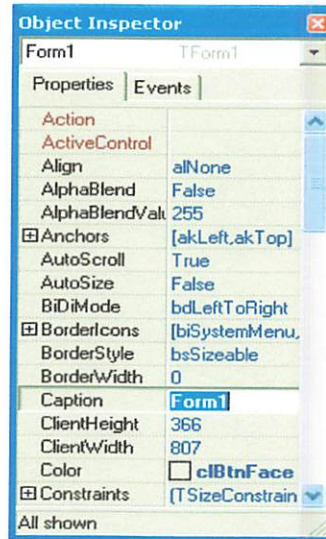
2. Object Treeview



Gambar 2.2 Object Treeview

Fasilitas *object treeview* dapat ditemukan mulai Borland Delphi 6.0 pada versi sebelumnya tidak ditemui fasilitas ini. Fasilitas ini digunakan untuk menampilkan daftar komponen yang sedang digunakan dalam membuat aplikasi.

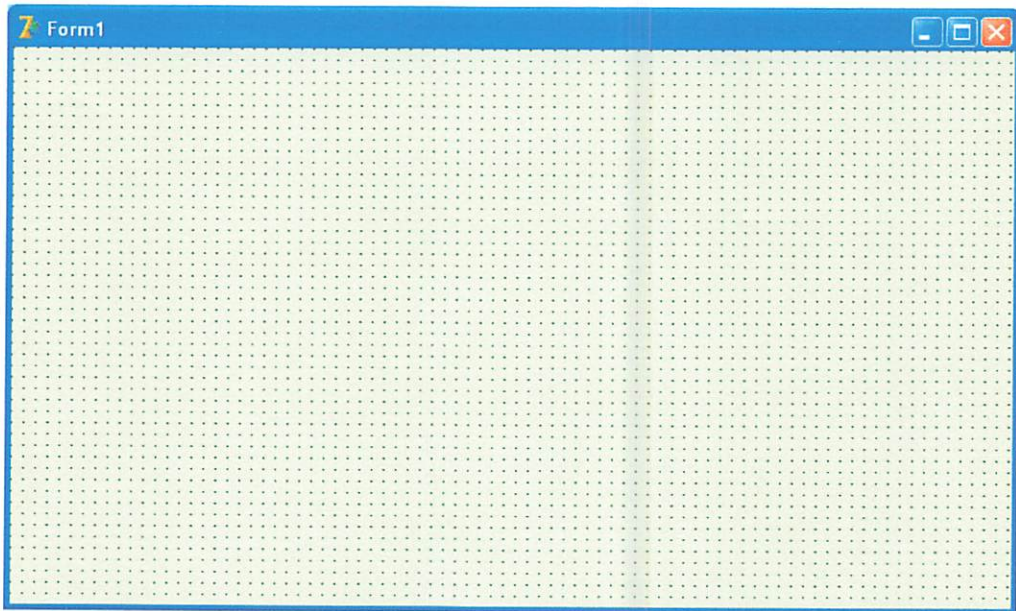
3. Object Inspector



Gambar 2.3 Object Inspector

Fasilitas ini digunakan untuk melihat atau mengubah nilai properti dan even suatu komponen. Dari sini nilai property suatu komponen dapat langsung diubah walaupun tidak secara keseluruhan, maksudnya disini adalah selain *object inspector*, nilai properti dapat diubah dari listing program. Selain itu, fasilitas ini digunakan untuk menentukan even atau kejadian apa listing program akan dijalankan.

4. Form Designer



Gambar 2.4 Form Designer

Form Designer adalah suatu komponen utama Borland Delphi. Form ini digunakan untuk meletakkan komponen yang lain ketika membuat sebuah aplikasi yang akan ditata sedemikain rupa sehingga akan diperoleh tampilan aplikasi yang menarik.

2.2.2 Menu Delphi

1. Menu File

Di dalam menu file ini terdapat beberapa menu di dalamnya sebagian contohnya adalah menu untuk membuat aplikasi baru, membuat *project* dan menutup aplikasi.

2. Menu Edit

Menu edit yang terdapat pada Delphi digunakan untuk mengubah segala sesuatu yang berhubungan dengan pembuatan *project*, misalnya adalah menu *undo*, *redo*, *copy*, *paste* dan beberapa menu lainnya.

3. Menu Search

Menu *search* adalah menu yang digunakan untuk pencarian, misalnya untuk mencari text di dalam listing program pada sebuah project.

4. Menu View

Digunakan untuk menampilkan kebutuhan apa saja yang ingin ditampilkan pada jendela utama pada saat membuat aplikasi, misalnya untuk menampilkan *component palette*, *object treeview*, *object inspector* dan juga beberapa kebutuhan lainnya.

5. Menu Project

Terdapat beberapa kegunaan di dalam menu ini, misalnya untuk mengatur kondisi *project* mulai dari form aplikasi yang ingin ditampilkan pertama kali ketika program dijalankan, menambah atau menggabungkan *project* yang terpisah dan menggabungkan form aplikasi yang terpisah untuk dikemas dalam satu project juga digunakan untuk *compile* aplikasi untuk dijadikan sebuah .exe. Selain itu ada beberapa fitur lain di dalam menu ini.

6. Menu Run

Didalam menu ini disediakan beberapa fitur, misalnya adalah untuk menjalankan aplikasi, juga terdapat fitur untuk *reset* aplikasi ketika ada *error* saat aplikasi sedang dijalankan.

7. Menu Component

Menu ini mempunyai beberapa kemampuan, misalnya untuk menambahkan komponen dari luar Delphi maupun dari Delphi itu sendiri. Hal ini dikarenakan tidak semua komponen yang dibutuhkan ketika membangun aplikasi disediakan pada *component palette*. Sebagai contoh adalah ketika membangun aplikasi kendali ini, programmer menggunakan komponen Tcomport yang berfungsi untuk menjembatani komunikasi serial antara Delphi dengan perangkat luar.

8. Menu Tools

Di dalam menu ini terdapat beberapa fitur, misalnya untuk memanggil database bawaan Delphi yaitu *Database Desktop* (Paradox).

9. Menu Window

Menu ini digunakan untuk mengarahkan kursor pada posisi yang dikehendaki.

10. Menu Help

Menu *help* adalah menu yang digunakan untuk memudahkan *programmer* untuk membuat aplikasi misalnya menemukan kesulitan pada saat membangun aplikasi.

2.2.3 Komponen Standart Delphi 7.0

Tabel 2.1 Komponen Standart Delphi

No	Icon	Name	Fungsi
1		Pointer	Mengembalikan fungsi mouse ke defaultnya
2		Frame	Membentuk suatu frame terhadap obyek yang ada didalamnya
3		Main menu	Membuat menu Utama
4		Popup Menus	
5		label	Hanya untuk menampilkan Teks
6		Edit	Untuk menampilkan dan input data (1 baris)
7		Memo	Sama seperti edit tetapi mempunyai kapasitas lebih besar (lebih dari 1 baris)
8		Button	Digunakan untuk melakukan eksekusi terhadap suatu proses
9		Checkbox	Digunakan untuk menentukan pilihan lebih dari satu
10		Radio Button	Digunakan untuk menentukan pilhan, tetapi hanya satu pilhan yang bisa digunakan
11		List Box	Menmpilkan pilihan dalam bentuk list
12		Combo Box	Menampilkan pilihan dalam bentuk popup
13		Scroll Bar	Merupakan icon yang berupa baris status
14		Group Box	Digunakan untuk mengelompokan suatu icon
15		Radio Group	Digunakan untuk mengelompokan pilihan

2.2.4 Tipe Data

Pada dasarnya, semua data yang tersimpan pada memori computer harus mempunyai tipe data. Pengertian tipe data disini adalah suatu aturan pada Delphi yang digunakan untuk menentukan suatu jenis variable. Berikut ini adalah beberapa tipe data yang digunakan dalam pemrograman Borland Delphi :

1. Tipe Data Text

a. Char

Adalah huruf atau angka yang terdiri dari satu karakter saja.

b. String

Adalah bentuk dari tipe data yang terdiri dari beberapa huruf atau angka.

Tabel 2.2 Tipe Data Text String

TIPE	JANGKAUAN
ShortString	255 karakter
AnsiString	-2^{31} karakter
WideString	-2^{30} karakter

2. Tipe Data Bilangan

a. Integer

Adalah bentuk tipe data untuk variable yang mempunyai nilai bilangan bulat.

Tabel 2.3 Tipe Data Bilangan Integer

TIPE	JANGKAUAN
ShortInt	-128 s/d 127

SmallInt	-32768 s/d 32767
LongInt	-21474836448 s/d 21474836447
Int64	$-2 \times E+63$ s/d $2 \times E-63-1$
Byte	0 s/d 255
Word	0 s/d 65535

b. Real

Tipe data real adalah tipe data untuk variable yang mempunyai nilai bilangan pecahan.

Tabel 2.4 Tipe Data Bilangan Real

TIPE	JANGKAUAN
Real48	$2.9 \times E-39$ s/d $1.7 \times E+38$
Singlet	$1.5 \times E-45$ s/d $3.4 \times E+38$
Double	$5.0 \times E-324$ s/d $1.7 \times E+308$
Extended	$3.6 \times E-4951$ s/d $1.1 \times E+4932$
Comp	-2^{63+1} s/d 2^{63-1}
Currency	-922337203685477.5808 s/d 922337203685477.5807

c. Boolean

Tipe data Boolean adalah tipe data yang mempunyai nilai true dan false.

Tabel 2.5 Tipe Data Bilangan Boolean

TIPE	JANGKAUAN
Real48	2.9 x E-39 s/d 1.7 x E+38
Singlet	1.5 x E-45 s/d 3.4 x E+38
Double	5.0 x E-324 s/d 1.7 x E+308
Extended	3.6 x E-4951 s/d 1.1 x E+4932
Comp	$-2^{63}+1$ s/d $2^{63}-1$
Currency	-922337203685477.5808 s/d 922337203685477.5807

2.2.5 Operator

Operator adalah bentuk penugasan pada sebuah ekspresi program, sebagai contoh adalah "5+6", maka yang disebut operator adalah (+). Dibawah ini adalah operator yang sering dikenal dalam Delphi.

1. Operator Aritmatik

Operator aritmatik adalah operator yang digunakan dalam operasi matematika yaitu penjumlahan, pengurangan, perkalian, pembagian, sisa bagi dan pembagian pada bilangan bulat. Dibawah ini adalah contoh tabelnya :

Tabel 2.6 Operator Aritmatik

OPERATOR	PENJELASAN	CONTOH
+	Penjumlahan	$1 + 2 = 3$
-	Pengurangan	$5 - 2 = 3$

*	Perkalian	$2 * 5 = 10$
/	Pembagian	$6 / 2 = 3$
mod	sis hasil bagi	$7 \text{ mod } 2 = 1$
div	pembagian bilangan bulat	$7 \text{ div } 2 = 3$

2. Operator Pembandingan

Operator pembandingan adalah operator yang digunakan untuk membandingkan dua buah nilai dan biasanya hasil dari pembandingan tersebut menghasilkan nilai *true* atau *false*. Berikut ini adalah tabelnya :

Tabel 2.7 Operator Pembandingan

OPERATOR	PENJELASAN
'>'	Lebih dari
'>='	Lebih dari sama dengan
'<'	Kurang dari
'<='	Kurang dari sama dengan
'=='	sama dengan
'<>'	Tidak sama dengan

3. Operator Logika

Bentuk dari operator logika ada tiga yaitu operator AND (pernyataan DAN), operator OR (pernyataan ATAU), kemudian operator NOT (pernyataan TIDAK). Ketiga operator ini apabila digunakan akan menghasilkan nilai *true* atau *false*.

2.3 Komunikasi Serial

Komunikasi serial adalah salah satu metode komunikasi data dimana hanya satu bit data yang dikirimkan melalui seuntai kabel pada suatu waktu tertentu. Pada dasarnya komunikasi serial adalah kasus khusus komunikasi paralel dengan nilai $n = 1$, atau dengan kata lain adalah suatu bentuk komunikasi paralel dengan jumlah kabel hanya satu dan hanya mengirimkan satu bit data secara simultan.

2.3.1 RS-232

RS-232 adalah nama tradisional untuk serangkaian standart untuk seri binner tunggal berakhir Data dan control sinyal menghubungkan antara DTE (peralatan terminal data) dan DCE (peralatan data circuit-terminating). Hal ini umumnya digunakan dalam komputer port serial. Standart ini mendefinisikan karakteristik listrik, waktu sinyal, arti sinyal, ukuran fisik dan pinout konektor. Versi saat ini dari standart adalah *TIA-232-F* Antarmuka antara Data Terminal Equipment dan Data Circuit-Terminating Equipment mempekerjakan Serial Binary Data interchange, yang di terbitkan pada tahun 1997.

Sebuah RS 232 port serial pernah menjadi fitur standart dari komputer pribadi, digunakan untuk koneksi ke modem, printer, mouse, penyimpanan data, pasokan listrik terputus dan perangkat peripheral lainnya. Namun kecepatan rendah transmisi, ayunan tegangan besar dan konektor standart besar termotivasi pengembangan Universal Serial Bus yang telah membuat RS-232 dari sebagian besar peran antarmuka peranti. Banyak komputer pribadi modern tidak memiliki port RS-232 dan harus menggunakan eksternal converter USB-to-RS-232 untuk menghubungkan ke RS-232 peripheral. Perangkat RS-232 masih ditemukan terutama di mesin industri, peralatan jaringan dan instrumen ilmiah.

2.3.2 Vacommm

Komponen VaComm merupakan komponen yang digunakan untuk mengakses port serial komputer, namun komponen tersebut tidak terdapat pada Delphi 7 sehingga harus di install terlebih dahulu. Dengan komponen VaComm maka proses pengiriman dan penerimaan data melalui port serial dapat dilakukan dengan mudah.



Gambar 2.24 Komponen VaComm pada IDE Delphi23

Setiap VaComm hanya dapat menangani satu port serial sehingga jika ingin menggunakan lebih dari satu port serial, maka harus menggunakan VaComm sebanyak port serial yang dipakai. Properti pada VaComm yang sering digunakan untuk mengatur konfigurasi port serial diantaranya:

- AutoOpen : Digunakan Untuk membuka komunikasi port secara otomatis.
- Baudrate : Sebagai pengaturan baudrate yang akan digunakan.
- Databits : Digunakan untuk menentukan jumlah data bit.
- DeviceName : Digunakan untuk menentukan nomor port serial yang akan dipakai.

Berikut merupakan pengaturan properti pada tombol untuk mengaktifkan port serial komputer, sehingga dapat melakukan pengiriman atau penerimaan data. Pengaturan Events untuk melakukan proses penerimaan satu karakter adalah VaCommRXchart. Setiap pembacaan karakter yang diterima (ReadText) akan disimpan pada variabel yang telah ditentukan, berikut prosedur programnya.

Prosedur VaCommRXChart akan membaca data yang diterima port serial dengan perintah ReadText, selanjutnya data disimpan pada variabel Tmp lalu menampilkannya di komponen Edit1.

2.4 Distorsi

Distorsi adalah sebuah perubahan suara yang terjadi ketika amplitude sinyal melebihi range yang tersedia. Hasilnya adalah timbulnya artifact harmonis tambahan seiring bentuk waveform berubah. Contoh distorsi adalah suara berisik yang dikeluarkan oleh speaker yang rusak. Distorsi biasanya ingin dihindari, namun untuk beberapa tujuan, distorsi justru di ringankan terutama dalam bidang music. Suara gitar cenderung bersih, menambah efek distorsi menghasilkan suara yang lebih menarik. Penggunaan efek ini tentu saja tidak harus untuk keperluan music, untuk game efek ini juga bias menghasilkan efek tertentu, misalnya suara radio yang agak rusak.

2.5 Tata Suara

Tata suara adalah suatu teknik pengaturan peralatan suara atau bunyi pada suatu acara pertunjukan, pertemuan, rekaman dan lain-lain. Tata suara memainkan peranan penting dalam suatu pertunjukan langsung dan menjadi satu bagian tak terpisahkan dari tata panggung, bahkan acara pertunjukan itu sendiri.

Tata suara erat kaitannya dengan pengaturan penguatan suara agar bias terdengar keras tanpa mengabaikan kualitas suara-suara yang dikuatkan. Pengaturan tersebut meliputi pengaturan mikropon-mikropon, kabel-kabel, prosessor, efek suara, pengaturan konsul mixer, audio power amplifier dan speaker-speakernya secara keseluruhan.

2.6 Wireless

Jaringan wireless adalah jaringan yang mengkoneksi dua komputer atau lebih menggunakan sinyal radio, cocok untuk berbagai file, printer atau akses internet. Teknologi wireless LAN menjadi sangat populer dibanyak aplikasi. Setelah dilakukan evaluasi terhadap teknologi tersebut, menjadikan para pengguna merasa puas dan meyakini realibility teknologi ini sudah siap untuk digunakan dalam skala luas dan komplek pada jaringan tanpa kabel.

Teknologi komunikasi data dengan tidak menggunakan kabel untuk menghubungkan antara klien dan server. Secara umum teknologi wireless LAN hamper sama dengan teknologi jaringan komputer yang menggunakan kabel Wire LAN (Local Area Network). Teknologi wireless LAN ada yang menggunakan frekuensi radio untuk mengirim dan menerima data yang tentunya mengurangi kebutuhan atau ketergantungan hubungan melalui kabel. Akibatnya pengguna mempunyai mobilitas atau fleksibilitas yang tinggi dan tidak tergantung pada suatu tempat atau lokasi. Teknologi wireless LAN juga memungkinkan untuk membentuk jaringan komputer yang mungkin tidak dapat dijangkau oleh jaringan komputer yang menggunakan kabel.

Wireless LAN bekerja dengan menggunakan gelombang radio. Sinyal radio menjalar dari pengirim ke penerima melalui free space, pantulan-pantulan, difraksi, line of sight dan obstructed tiap sinyal. Pada jalur yang berbeda-beda memiliki level kekuatan, delay dan frasa yang berbeda-beda. Awalnya teknologi ini di desain untuk aplikasi perkantoran dalam ruangan, namun sekarang wireless LAN dapat digunakan pada jaringan peer to peer dalam ruangan dan juga point to point diluar ruangan maupun point to multipoint pada aplikasi bridge wireless LAN di desain sangat modular dan fleksibel. Jaringan ini juga bisa di optimalkan pada lingkungan yang berbeda. Dapat mengatasi kendala geografis dan rumitnya instalasi kabel.

Mirip dengan jaringan Ethernet kabel, sebuah wireless LAN mengirim data dalam bentuk paket. Setiap adapter memiliki no ID yang permanen dan unik yang berfungsi sebagai sebuah alamat, tiap paket selain berisi data juga menyertakan alamat penerima dan pengirim paket tersebut. Sama dengan sebuah adapter Ethernet, sebuah kartu, wireless LAN akan memeriksa kondisi jaringan sebelum mengirim paket ke dalamnya. Bila jaringan dalam keadaan kosong, maka paket langsung dikirimkan. Apabila kartu mendeteksi adanya data lain yang sedang menggunakan frekuensi radio, maka ia menunggu sesaat kemudian memeriksanya kembali.

2.7 Mikrokontroler ATmega 8535

ATMega8535 adalah mikrokontroler CMOS 8 *bit* daya rendah berbasis arsitektur RISC. Instruksi dikerjakan pada satu siklus clock, ATMega8535 mempunyai throughput mendekati 1 MIPS per Mhz, hal ini membuat ATMega8535 dapat bekerja dengan kecepatan tinggi walaupun dengan penggunaan daya rendah. Mikrokontroler ATMega8535 memiliki beberapa fitur atau spesifikasi yang menjadikan sebuah solusi pengendali yang efektif untuk berbagai keperluan. Fitur-fitur tersebut antara lain:

1. Saluran I/O sebanyak 32 buah, yang terdiri atas port A,B,C dan D
2. ADC (Analog to Digital Converter)
3. Tiga buah timer/ Counter dengan kemampuan perbandingan
4. CPU yang terdiri atas 32 register
5. Watchdog timer dengan osilator internal
6. SRAM sebesar 512 byte
7. Memori flash sebesar 8kb dengan kemampuan read while write
8. Unit interupsi internal dan eksternal
9. Port antarmuka SPI untuk men-download program ke flash
10. EEPROM sebesar 512 byte yang dapat deprogram saat operasi
11. Antarmuka komparator analog
12. Port USART untuk komunikasi serial

BAB III

PERANCANGAN DAN DESAIN APLIKASI

3.1 Deskripsi Umum

Program aplikasi sistem monitoring tata suara panggung adalah aplikasi yang dibangun untuk memonitor speaker yang berada di atas panggung yang saling berlawanan arah dengan pengontrol guna untuk mempermudah mengetahui tingkatan speaker yang sudah melebihi *overload* dan menyesuaikan dengan sound *FOH (front of house)* yang cenderung lebih besar dan saling berhadapan dengan pengontrol.

Program aplikasi monitoring tata suara panggung dibangun sesuai dengan spesifikasi *hardware* yang sudah ada, sehingga apabila ingin menggunakan aplikasi ini pada perangkat lain yang mempunyai spesifikasi berbedah, maka program tidak akan berfungsi dan haru dilakukan perubahan pada program aplikasi agar aplikasi dapat digunakan pada *hadrware* yang memilki spesifikasi berbeda.

Beberapa hal yang akan di bahas dalam pembuatan aplikasi monitoring ini:

1. Perancangan sistem komunikasi dari *hardware* untuk di koneksikan ke dalam program Delphi.
2. Perancangan tampilan monitoring mulai dari pembuatan beberapa form menu yang akan di gunakan dalam aplikasi ini.
3. Melakukan perhitungan dari data yang masuk akan di ubah menjadi Tegangan (V) untuk mendapatkan nilai Desibel (dB).
4. Perancangan drag pada menu yang terdapat hasil monitoring level speaker.

3.2 Analisa kebutuhan sistem

Analisis kebutuhan dilakukan untuk mengetahui spesifikasi kebutuhan untuk sistem. Terdapat dua Spesifikasi kebutuhan yaitu kebutuhan perangkat lunak dan perangkat keras.

3.2.1 Analisa kebutuhan perangkat lunak (Software)

Spesifikasi perangkat lunak yang digunakan dalam pembuatan aplikasi, adalah sebagai berikut :

1. Sistem Operasi Windows 7

Alasan menggunakan Windows 7, karena penulis lebih terbiasa menggunakan sistem operasi pada windows.

2. Delphi 7

Alasan menggunakan Delphi adalah bahasa pemrograman tingkat tinggi yang mudah dipahami dan dimengerti oleh manusia (*Readable*). Jadi perlu diketahui bagi anda yang baru mengenal atau belum mengenal Delphi sebelumnya bahwa kode-kode yang digunakan dalam bahasa pemrograman ini banyak menggunakan istilah dalam kehidupan sehari-hari. Sebagai contoh *IntToStr*. Ini adalah fungsi yang digunakan untuk mengkonversi suatu variable atau nilai dengan tipe data *Integer* menjadi *string*.

3. Microsoft Access

Alasan menggunakan *Microsoft Access* mudah digunakan dan dikoneksikan dengan Delphi.

4. Komponen Vacomm

Untuk mempercepat sistem komunikasi dari USB serial yang merupakan jalur dari *hardware* untuk mengirim data dari mikrokontroler ke laptop.

5. Komponen Alpha skin

Untuk memperbagus tampilan aplikasi monitoring layaknya aplikasi pemutar musik yang bias dirubah warna dan skin pada tampilan.

3.2.2 Analisa kebutuhan perangkat keras (*Hardware*)

Spesifikasi perangkat keras yang digunakan dalam pembuatan aplikasi, adalah sebagai berikut :

1. Laptop minimum windows xp
2. USB RS 232
3. Mikorkontroler ATMEGA
4. Sepasang modul TLP RLP
5. Speaker

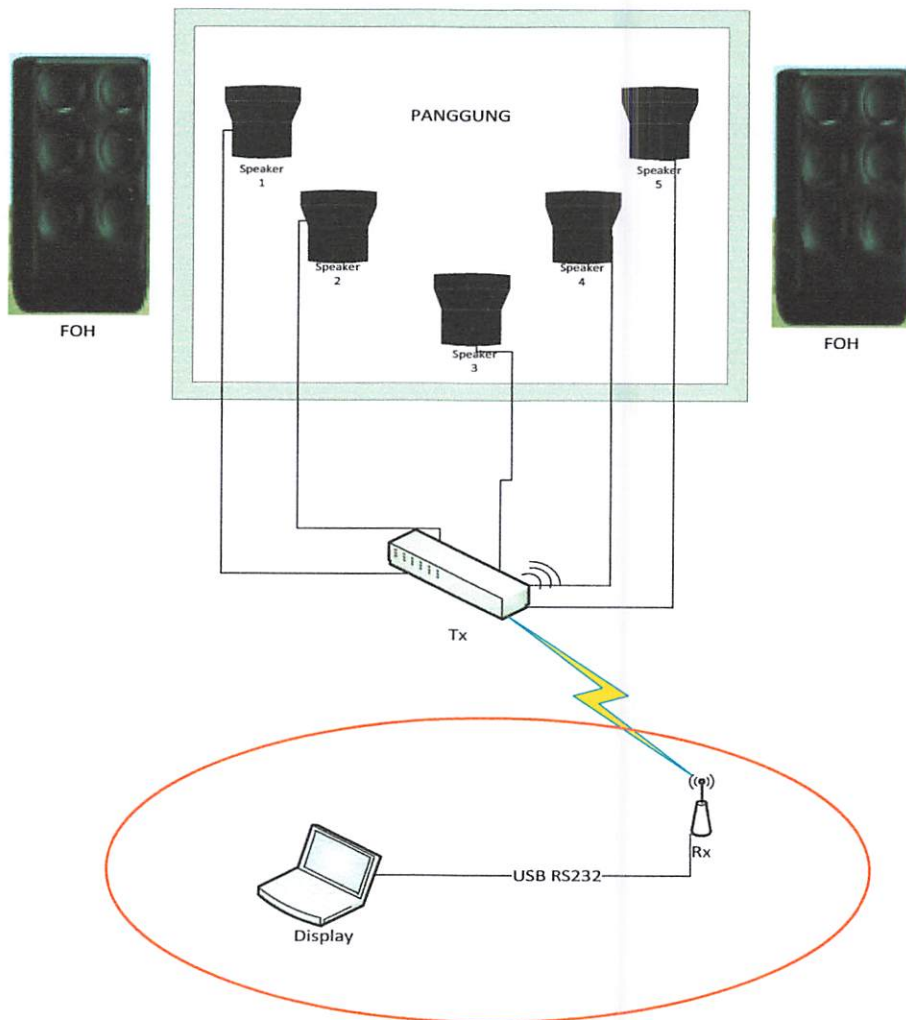
3.3 Desain Sistem

3.3.1 Gambaran umum aplikasi

Gambara umum dari aplikasi monitoring tata suara panggung yang akan dirancang adalah sebagai berikut :

1. Aplikasi monitoring ini berbentuk tampilan *user interface* yang menggunakan bahasa pemograman delphi 7 dan database *Microsoft Access* sebagai media penyimpanan data.
2. Dalam aplikasi ini pengguna pertama harus melakukan setting komunikasi untuk menentukan COMPORT dimana kabel USB RS232 terkoneksi, setelan itu melakukan proses komunikasi dengan *hardware*.
3. Aplikasi monitoring dapat digunakan untuk memonitoring speaker di panggung yang kalah dengan speaker *FOH (front of house)* yang saling berhadapan dengan pengontrol
4. Aplikasi ini juga dapat mengatur tata letak speaker karena pada bar speaker dapat di drag layaknya tampilan di dekstop.

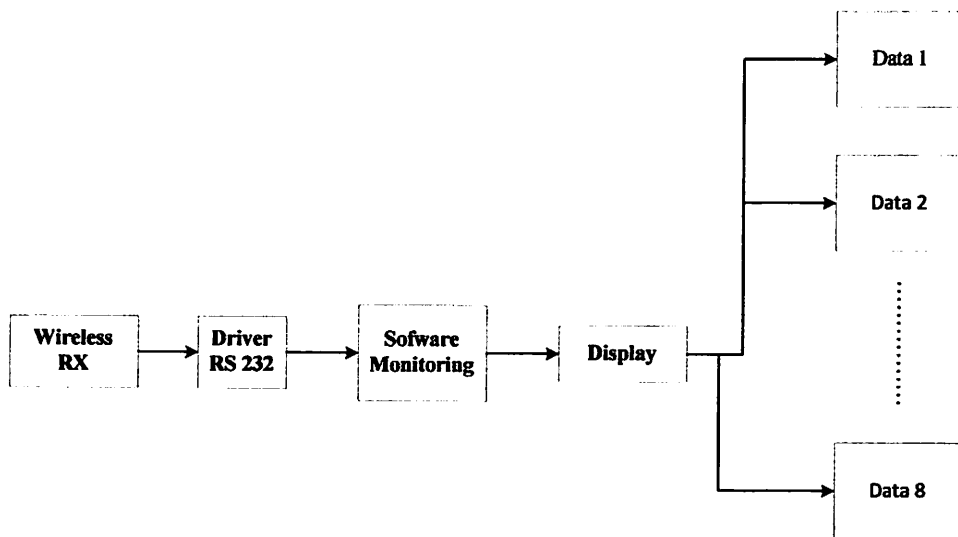
Jadi dari penjabaran di atas maka dibuat gambaran umum untuk membuat aplikasi monitoring ini yang merupakan desain awal dalam perancangan aplikasi remote monitoring tata suara panggung berikut merupakan tampilan gambaran umum :



Gambar 3.1 Gambaran umum secara keseluruhan

Pada gambar di atas tidak semua akan dibahas dalam perancangan aplikasi ini karena dalam perancangan ini saya hanya akan membahas tentang perancangan aplikasi monitoring dari modul RX sampai di tampilkan pada laptop seperti yang di tandai pada lingkaran merah.

jadi untuk melakukan perancangan ini maka dibuat gambaran secara detail dari proses perancangan aplikasi mulai dari RX sampai pada tampilan program aplikasi berikut gambar perancangan.



Gambar 3.2 Gambaran umum aplikasi yang akan di rancang

Dari gambar di atas merupakan rancangan aplikasi monitoring dari data yang diterima oleh RX dengan kabel USB RS232 akan dihubungkan pada laptop yang selanjutnya akan di proses didalam *Software* monitoring untuk menampilkan data yang di terima.

3.3.2 Pemodelan Aplikasi

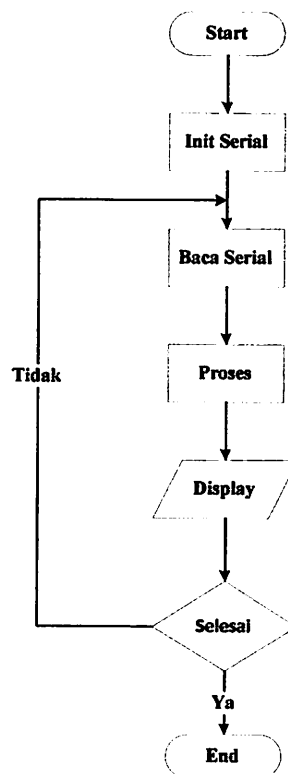
Dalam merancang program aplikasi, terlebih dahulu yaitu menggambar cara kerja aplikasi sesuai dengan fungsinya guna untuk mendapatkan algoritma program secara global yang digambarkan dalam sebuah *Flowchart* aplikasi.

3.3.3 Flowchart

Flowchart adalah suatu teknik untuk menyusun rencana program. Selain itu *Flowchart* adalah untaian simbol gambar (*chart*) yang menunjukkan aliran (*flow*) dari proses terhadap data.

3.3.3.1 Flowchart program aplikasi

Flowchart ini menggambarkan langkah-langkah dalam perancangan pembuatan aplikasi monitoring secara global, *flowchart* program aplikasi dapat dilihat pada gambar sebagai berikut.

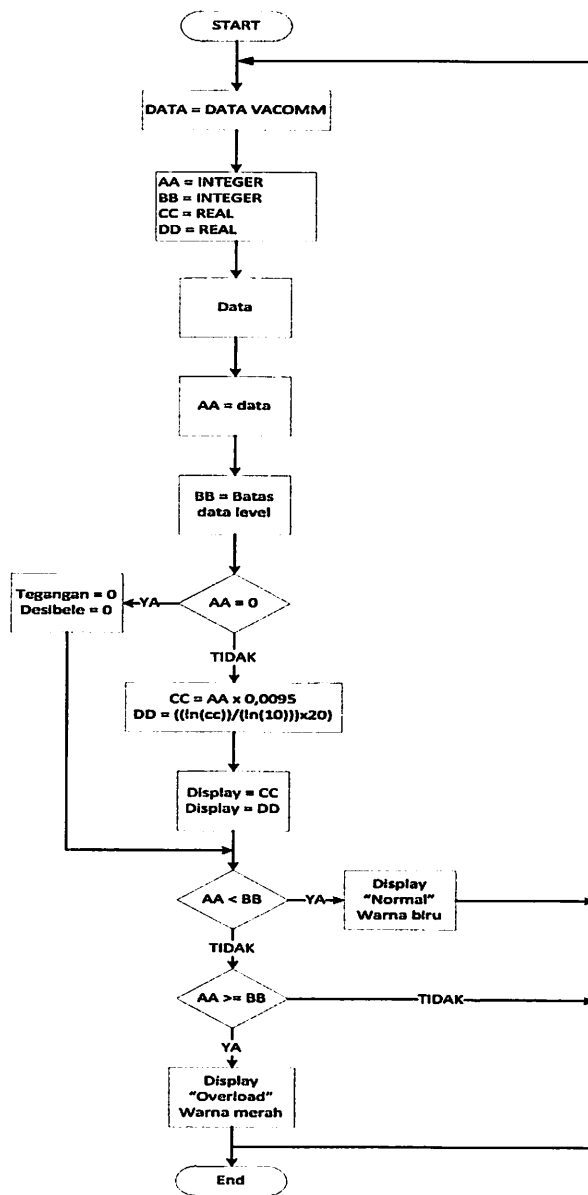


Gambar 3.3 Flowchart Aplikasi Monitoring

Dari *Flowchat* diatas dapat disimpulkan bahwa ada beberapa proses aliran pada proses pembuatan aplikasi monitoring mulai dari Init serial yang kemudian akan dibaca serial yang akan di proses untuk di tampilkan pada aplikasi monitoring tata suara panggung berupa level-level speaker, untuk menentukan keadaan YES/NO menunjukkan berakhirnya proses aplikasi jika proses tersebut berakhir maka monitoring telah selesai dilakukan jika tidak akan kembali lagi pada baca serial.

3.3.3.2 Flowchart proses aplikasi

Flowchat ini menggambarkan langkah-langkah yang terdapat dalam proses aplikasi untuk menampilkan data yang diterima berupa display bar level speaker dan dilakukan perhitungan untuk memperoleh tegangan (V) dan decibel (dB). Berikut gambar *flowchart* proses aplikasi.



Ket. Flowchart :

AA = DATA

BB = BATAS DATA

CC = TEGANGAN (V)

DD = DESIBELE (dB)

Gambar 3.4

Flowchart proses aplikasi

3.4 Perancangan dan pembuatan Desain Aplikasi

Pada tahap awal pembuatan program aplikasi adalah perancangan desain aplikasi. Di dalam merancang desain aplikasi dilakukan dengan teliti hal ini berkaitan dengan dimana dan bagaimana *source code* program akan diletakkan supaya mudah dan tidak membingungkan saat melakukan *coding* program.

Didalam membangun program aplikasi, pemrograman tidak dilakukan dengan metode *console* tetapi pemrograman dilakukan dengan memanfaatkan komponen visual

yang terdapat pada Borland Delphi dan ada juga tambahan beberapa komponen yang tidak ada di dalam Borland Delphi karena dengan metode ini program aplikasi akan lebih unggul dalam tampilan dan tentunya juga dalam hal waktu dalam pengerjaan program aplikasi. Adapun beberapa hal dan beberapa komponen yang bersangkutan dengan perancangan dan pembuatan program aplikasi monitoring tata suara panggung ini adalah sebagai berikut :

1. Coding

Yaitu proses pengkodean komponen yang digunakan sesuai dengan fungsi yang diinginkan. Disini coding dilakukan dengan menggunakan bahasa pemrograman pascal karena Delphi sendiri adalah pengembangan dari pascal.

2. Form

Form adalah jendela utama atau lembar kerja yang ada pada Delphi. Form ini digunakan untuk meletakkan komponen yang akan digunakan saat membangun aplikasi. Menarik dan tidaknya tampilan suatu aplikasi akan tergantung penataan komponen yang ada pada form ini.

3. Properties

Didalam IDE Delphi terdapat *object inspector* yang didalamnya terdapat menu properties. Menu ini digunakan untuk mengisikan nilai dari suatu komponen. Hal ini dimaksudkan agar programmer mudah dalam menginisial suatu komponen.

4. Event

Masih di lingkungan *object inspector*, selain properties juga terdapat menu event dimana fungsi dari menu ini adalah untuk menempatkan *source code* sesuai fungsi yang diinginkan.

5. Label

Merupakan salah satu jenis komponen yang digunakan untuk menginisial atau penamaan suatu tampilan.

6. Edit

Edit adalah satu komponen didalam Delphi yang berfungsi sebagai inputan. Edit juga sering dikenal dengan nama *Textbox*.

7. Button

Button adalah komponen yang terdapat pada Delphi yang digunakan sebagai tombol yang akan memproses suatu kejadian sesuai kode program yang dimasukkan. Komponen *Button* mempunyai beberapa keluarga, salah satunya adalah *bitbutton* dimana masing-masing mempunyai dasar fungsi yang sama tetapi masing-masing mempunyai kelebihan dan kekurangannya.

8. Image

Merupakan komponen yang digunakan untuk meletakkan gambar. Sebagai contohnya adalah untuk membuat *background* gambar atau menampilkan photo.

9. Groupbox

Groupbox adalah komponen yang digunakan untuk pengelompokan komponen-komponen yang lain supaya tampilan lebih rapi dan mudah dalam proses pengkodean program.

10. Vacomm

Komponen yang digunakan untuk mengakses *port* serial komputer, namun komponen tersebut tidak terdapat pada Delphi 7 sehingga harus di *install* terlebih dahulu. Dengan komponen VaComm maka proses pengiriman dan penerimaan data melalui *port* serial dapat dilakukan dengan mudah.

11. Alpha skin

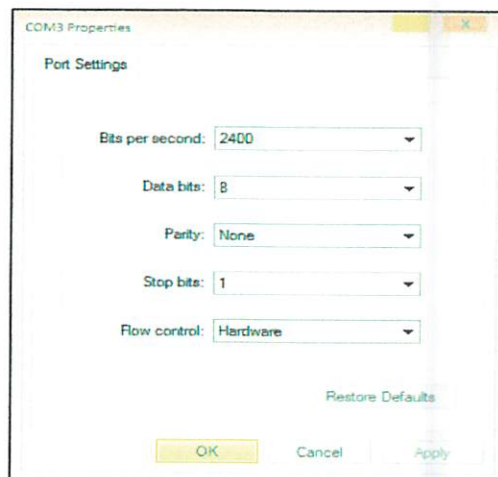
Merupakan komponen untuk membuat skin pada Delphi agar tampilan jadi lebih bagus dan menarik.

3.5 Perancangan Sistem Komunikasi

Pada tahap perancangan sistem komunikasi ini hal yang pertama dilakukan adalah uji koneksi *hardware* dengan laptop melalui kabel USB RS232 yang merupakan jalur komunikasi dalam pembuatan remote monitoring tata suara panggung.

Pada komunikasi ini data yang dikirim sudah disesuaikan dari masing-masing inputan dengan inisial data D1,D2,D3.....D8 karena sudah di tentukan di dalam mikrokontroler ATmega 8535 yang terdapat PIN untuk memilah data- data yang di terima oleh sebab itu data yang diterima dapat dibagi meskipun menggunakan satu jalur komunikasi.

Untuk Perancangan komunikasi dilakukan pengecekan kabel USB RS232 dengan menggunakan *hyper terminal* dengan settingan yang sudah di tetntukan seperti gambar dibawah ini.



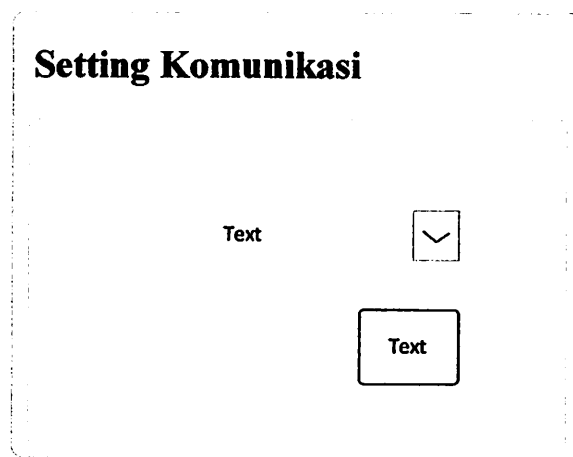
Gambar 3.5 Setting komunikasi hyper terminal.

Gambar diatas merupakan settingan pada hyper terminal untuk mengkoneksikan kabel USB RS232 sebagai perantara *hardware* dengan laptop.

3.6 Perancangan Tampilan Program Aplikasi

3.6.1 Perancangan Form Setting Komunikasi

Dalam perancangan Setting komunikasi ini adalah hal yang paling utama dan paling penting dalam pengerjaan remote monitoring tata suara panggung karena merupakan jalur untuk melakukan setting komunikasi dimana Comport USB RS232 terhubung berikut merupakan rancangan tampilan form setting komunikasi.

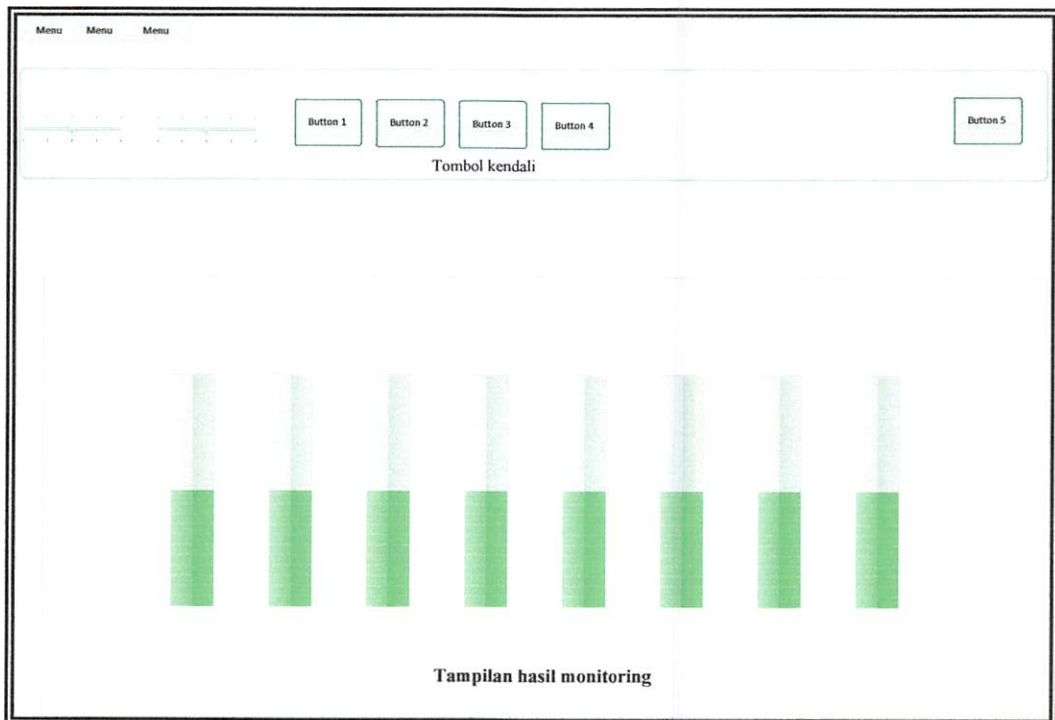


Gambar 5.6 Form perancangan setting komunikasi

Pada perancangan form setting komunikasi ada beberapa komponen yang digunakan dalam pembuatan form tersebut diantaranya *Button* yang digunakan untuk menyimpan setting komunikasi dalam aplikasi, *ComboBox* digunakan untuk menentukan di COM berapa kabel USB RS232 terhubung dan yang paling penting yaitu penggunaan komponen *Vacomm* untuk mengakses port serial pada komputer sebagai komunikasi aplikasi dengan port serial.

3.6.2 Perancangan Form Monitoring

Form monitoring merupakan form utama pada aplikasi remote monitoring tata suara panggung karena pada form ini akan di tampilkan hasil monitoring dari speaker-speaker yang sudah terhubung dan juga akan diketahui mana speaker yang mengalami *overload* dan tidak. Berikut adalah tampilan raancangan form monitoring



Gambar 3.7 Rancangan form monitoring

Didalam form ini terdapat beberapa komponen yang digunakan dalam perancangan aplikasi guna untuk mendapatkan hasil yang baik dan bagus misalnya komponen *button* pada perancangan form monitoring ini digunakan beberapa komponen Button yang digunakan untuk tombol proses pada aplikasi sesuai dengan fungsi dari tombol-tombol tersebut diantaranya :

1. Button 1

Yang akan digunakan untuk tombol membuka form setting komunikasi.

2. Button 2

Yang akan digunakan untuk melakukan koneksi dengan *hardware* jika sudah melakukan setting komunikasi dan laptop sudah terhubung dengan *hardware*.

3. Button 3

Yang akan digunakan untuk menampilkan database hasil data yang diterima.

4. Button 4

Yang akan digunakan untuk menampilkan border aplikasi monitoring.

5. Button 5

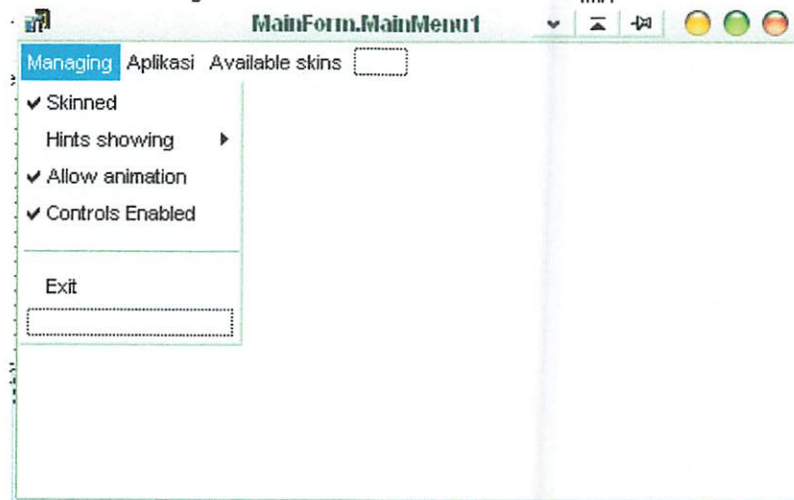
Yang akan digunakan untuk keluar aplikasi (Exit)

Digunakan juga komponen *Progresbar* yang akan digunakan untuk menampilkan data yang di terima yaitu berupa level-level speaker seperti spectrum, komponen *trackbar* untuk mengatur warna pada tampilan aplikasi monitoring tersebut dan komponen *alphaskin* guna untuk mempercantik tampilan aplikasi monitoring dengan cara merubah skin tampilan tersebut layaknya pemutar aplikasi pemutar musik yang sudah ada.

Pada perancangan form monitoring ini juga terdapat komponen *mainmenu* yang digunakan untuk membuat menu pada aplikasi diantaranya

1. Menu managing

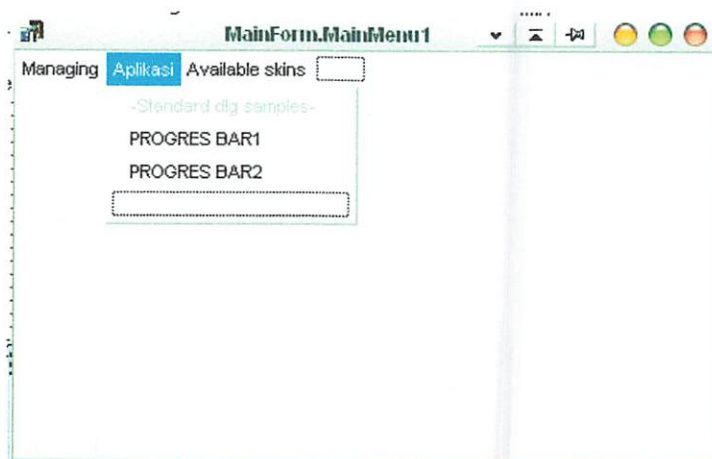
Pada menu ini ada beberapa pilihan untuk mengaktifkan beberapa skin.



Gambar 3.8 Menu Managing

2. Menu aplikasi

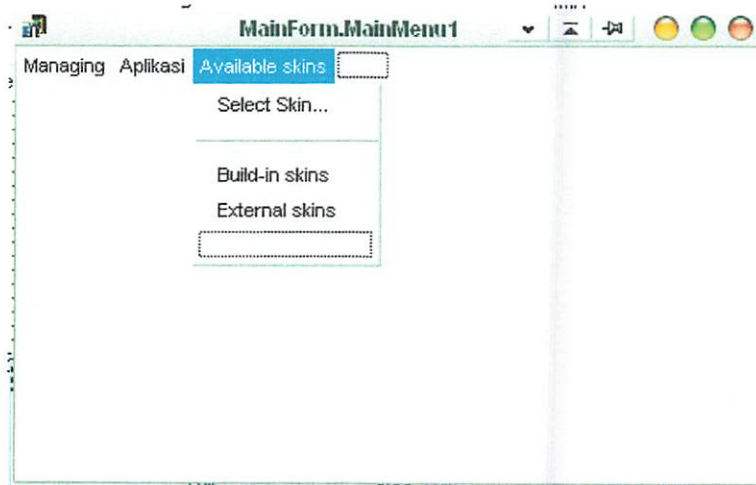
Pada menu ini adalah menu untuk menentukan model tampilan pada progresbar.



Gambar 3.9 Menu Aplikasi

3. Menu available skin

Pada menu ini adalah menu dimana untuk memilih jenis skin yang akan digunakan pada aplikasi monitoring



Gambar 3.10 Menu Available Skin

Pada tahap awal *desain* monitoring yang pertama kali dilakukan hanyalah percobaan koneksi beserta tampilan dasar yang akan digunakan pada aplikasi, pada tahap uji koneksi, data yang dibaca dari *hardware* akan di tampilkan pada *Progressbar*.

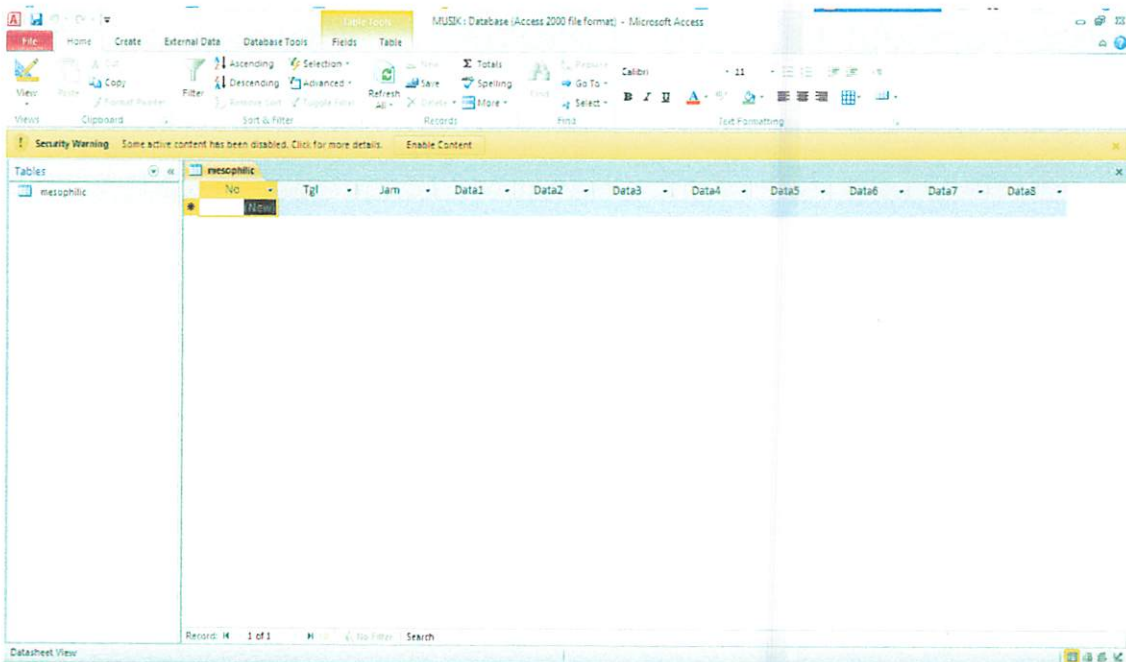
3.6.3 Perancangan database

Form *Database* adalah form yang digunakan untuk menyimpan data-data yang dikirim oleh *hardware* dan mengetahui berapa data yang telah diterima, database ini menggunakan *Microsoft Access* yang akan dihubungkan dengan *Delphi*.

Pada bagian ini terdiri tabel-tabel yang akan dibutuhkan dalam membangun database pada aplikasi monitoring

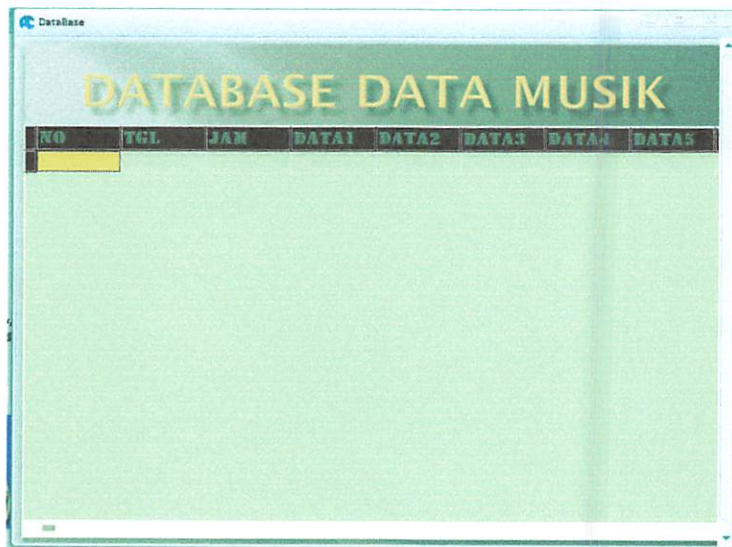
Nama Field	Panjang	Keterangan
No	Long Integer	
Tgl	10	
Jam	10	
Data 1	10	
Data 2	10	
Data 3	10	
Data 4	10	
Data 5	10	
Data 6	10	
Data 7	10	
Data 8	10	

Tabel 3.1 Aplikasi Monitoring



Gambar 3.11 Hasil akhir perancangan database pada Microsoft Access

Setelah melakukan pembuatan database pada Microsoft Access selanjutnya membuat mengkoneksikan database dari Access ke Delphi dengan menggunakan komponen *ADOconnection* yang digunakan untuk memanggil database dari Microsoft Access yang kemudian akan di tampilkan menggunakan komponen *DBgrid* untuk menampilkan database pada Delphi yang sudah di buat pada *Microsoft Access*. pada gambar merupakan hasil tampilan database pada Delphi.



Gambar 3.12 Tampilan Database pada Delphi

Pada form *database* dilakukan pengaturan untuk menyimpan data maksimal karena data yang dikirim terlalu banyak jadi diberi batasan untuk menyimpan data dalam database sehingga jika sudah melebihi batas maksimal maka database tidak akan melakukan penyimpanan data yang diterima.

3.7 Perhitungan untuk mendapatkan Outputan Tegangan (volt) dan Desibel (dB) pada aplikasi monitoring.

Pada aplikasi monitoring yang akan ditampilkan adalah data berupa *Decibel (dB)* jadi pada program aplikasi harus dimasukkan perumusan sebagai berikut.

Untuk mendapatkan *Tegangan (Volt)* yaitu dengan perumusan :

$$V = \text{Data yang masuk} \times \left(\frac{5V}{10bit} \right)$$

$$10bit = 1023$$

Data yang masuk didapat dari *hardware* yang merupakan data sinyal dari speaker yang berupa analog yang kemudian di konverter menggunakan ADC pada mikrokontroler supaya data menjadi digital dan dapat ditampilkan pada program aplikasi.

Contoh :

Data yang masuk : 613

$$V = 613 \left(\frac{5V}{1023} \right)$$

$$V = 3V$$

Dari hasil perhitungan tegangan maka untuk mendapatkan $dB = 20 \log 3$

Karena pada delphi tidak mengenal perumusan log maka pada delphi di ganti dengan menggunakan lan dengan perumusan

$$\frac{Lan10}{lanVin}$$

Jadi perumusan yang dimasukkan untuk mencari $dB = 20 \left(\frac{lan10}{lan3} \right)$

$$= 42 \text{ Db}$$

Jadi untuk Inputan tegangan 3V maka Outputan Decibel = 42 db

Setelah dilakukan perhitungan maka hasil perhitungan akan di tampilkan pada program aplikasi monitoring pada bagian bar speaker berikut tampilannya :



Gambar 3.13 Tampilan Bar Speaker

Pada tampilan *Bar Speaker* akan menentukan speaker yang mengalami overload jika data yang diterima melebihi dengan *settingan* yang sudah di buat pada level output maksimal, jika mengalami *overload* maka gambar *speaker* merah dan jika normal akan berwarna biru.

3.8 Perancangan Drag Pada Aplikasi Monitoring

Pembuatan *Drag* pada tampilan *Bar Speaker* guna untuk mempermudah mengetahui *speaker* yang mengalami gangguan atau *overload* dan dapat juga di fungsikan untuk menentukan tata letak *speaker*, jadi bar speaker yang terdapat di dalam aplikasi monitoring bisa di pindah-pindah sesuai dengan penataan speaker yang berada di atas panggung

BAB IV

IMPLEMENTASI DAN PENGUJIAN APLIKASI

4.1 Spesifikasi program

Program aplikasi monitoring tata suara panggung adalah program aplikasi yang dibuat hanya untuk memenuhi tugas akhir perkuliahan bukan untuk kehidupan yang sebenarnya karena *hardware* yang dibuat spesifikasinya terbatas dan kekuatannya tidak bisa di pakek untuk sound system yang mempunyai tegangan tinggi layaknya sound system yang biasanya terdapat pada sebuah acara dan konser. Program aplikasi ini mempunyai kemampuan untuk memonitoring hasil *Outputan* speaker.

Kemampuan memonitoring tata suara panggung adalah program aplikasi ini dapat digunakan untuk memonitoring *Outputan* speaker yang di kirim oleh *hardware* ke laptop dan aplikasi monitoring akan menampilkan data yang masuk melalui tampilan seperti *Spectrum* untuk mengetahui level data yang di terima jika data yang di keluarkan oleh Speaker melebihi data yang sudah di atur dalam aplikasi monitoring berarti data yang di terima mengalami *Overload* dan akan ada sebuah tanda yang akan menunjukkan bahwa data yang di terima melebihi data yang sudah di tetapkan

Program aplikasi ini juga dilengkapi pengaturan skin dan pengaturan warna guna untuk memperbaiki tampilan monitoring untuk menyamakan dengan program pemutar musik yang sudah ada.

4.2 Petunjuk Operasional dan uji coba

Dalam pembuatan program aplikasi terlebih dahulu di buat rancangan bagaimana agar program aplikasi mudah bagi operator untuk mengoprasikannya. Dengan melalui beberapa tahap percobaan yang akhirnya program aplikasi dibuat layaknya aplikasi pemutar musik yang menggunakan *spectrum* guna untuk mengetahui level dari speaker dengan mudah.

Pada tahap percobaan tidak dilakukan hanya dengan mencari ide yang tepat dan cocok untuk membuat tampilan aplikasi monitoring ini, tetapi di lakukan juga pengesetan dengan *hardware* guna untuk menentukan data yang dikirim harus sesuai dengan data yg di terima oleh aplikasi yang akan di tampilkan pada aplikasi tersebut.

Selain itu ada juga kebutuhan sistem yang sangat berperan penting dalam pengujian aplikasi ini diantaranya:

1. Kebutuhan perangkat keras (*hardware*)

Kebutuhan perangkat keras yang digunakan untuk mendukung proses pengoperasian sistem aplikasi monitoring tata suara panggung ini adalah.

1. Sepasang perangkat mikrokontroler TLP RLP yang sudah ada
2. Monitor
3. Kabel RS 232
4. Speaker

2. Kebutuhan perangkat lunak (*software*)

Kebutuhan perangkat lunak (*software*) yang digunakan untuk mendukung pembuatan program aplikasi monitoring ini, antara lain

1. Sistem operasi yang digunakan adalah microsoft windows minimum windows XP
2. Borland delphi 7

4.3 Implementasi

Tahap implementasi pengembangan perangkat lunak merupakan proses perubahan spesifikasi sistem menjadi sistem yang dapat dijalankan, tahapan ini merupakan tahapan lanjutan dari proses perancangan, yaitu proses pemrograman perangkat lunak sesuai dengan spesifikasi dan desain sistem.

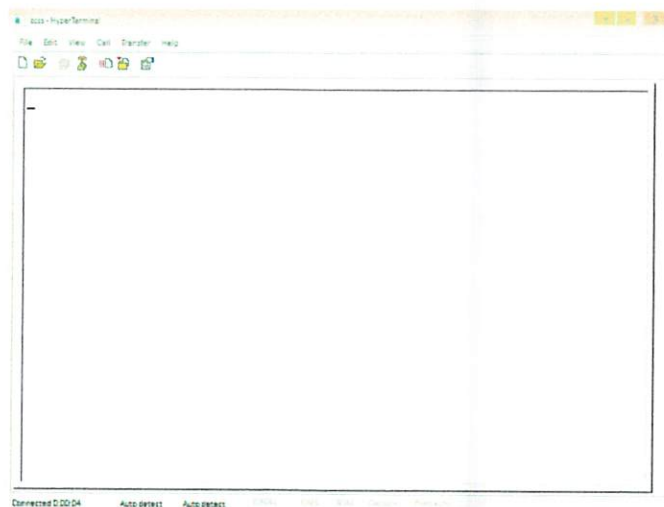
Dalam aplikasi pembuatan monitoring tata suara panggung ini, yang digunakan untuk mengimplementasikan monitoring tata suara panggung adalah Borland Delphi 7, hasil implementasi dari monitoring tata suara panggung meliputi setting komunikasi, tampilan monitoring, Database, perhitungan hasil outputan dan penerapan sistem drag pada aplikasi.

4.4 Pengujian Hasil Aplikasi Monitoring

4.4.1 Pengujian Komunikasi

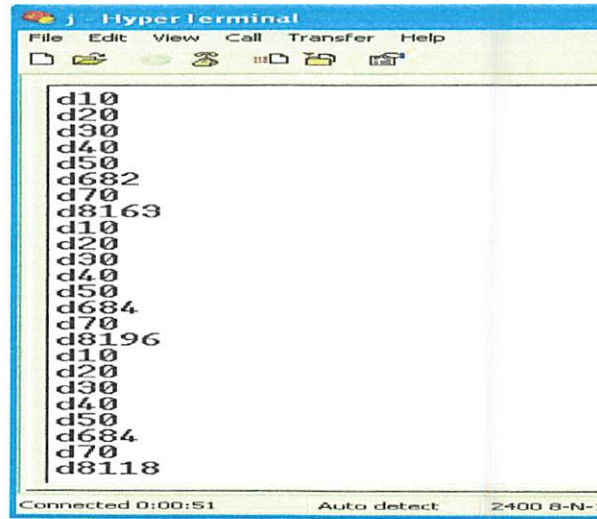
1. Pengujian komunikasi dengan hyper terminal

Untuk melakukan uji koneksi *hardware* pertama yang dilakukan adalah pengujian dengan menggunakan *hyper terminal* untuk mengetahui apakah *hardware* sudah dapat mengirim data pada laptop yang menggunakan kabel serial USB RS232 karena dalam pengujian ini merupakan hal yang paling pokok untuk melakukan komunikasi pada aplikasi remote monitoring tata suara panggung yang sudah dibuat. Berikut gambar hasil pengujian dengan menggunakan *hyper terminal*



Gambar 4.1 Pengujian *Hyper Terminal* tanpa *hardware*

Pada gambar di atas merupakan komunikasi data dengan *hyper terminal* yang terkoneksi dengan kabel RS 232 yang belum tersambung dengan *hardware*.



Gambar 4.2 Komunikasi dengan *Hyper Terminal* yang sudah terhubung dengan *hardware*

Pada gambar diatas adalah pengujian yang sudah terhubung dengan *hardware* yang berisi data-data yang sudah dikirim dari *hardware* dan data tersebut sudah di pilah-pilah oleh mikrokotroler pada *hardware* menjadi D1,D2.....D8 dan pembacaan data tersebut sebagai berikut D merupakan inisialisasi data yang diberikan, selanjutnya baris ke dua angka 1-8 merupakan chanel data yang sudah di pilah-pilah, pada baris ke 3 sampai 6 merupakan data dari speaker yang sudah di converter pada ADC dari analog menjadi digital

2. Pengujian komunikasi dengan aplikasi monitoring

Pengujian berikut adalah untuk melakukan Setting komunikasi pada aplikasi remote monitoring tata suara panggung guna untuk mengkoneksikan aplikasi dengan laptop yang sudah terhubung dengan *hardware* yang melalui USB RS232



Gambar 4.3 Setting komunikasi pemilihan COMPORT

Pada awal pengujian setting komunikasi ini yang dilakukan adalah menentukan pada COMPORT berapa USB serial RS232 terkoneksi pada laptop, dilakukan penentuan COMPORT karena pada laptop sering berubah-ubah pembacaan port serial.



Gambar 4.4 Setting komunikasi penyimpanan Comport

Setelah melakukan pemilihan Comport pada setting komunikasi maka hasil setting tersebut akan disimpan untuk diprose didalam komponen *Vacom* pada *Device name*.

4.4.2 Pengujian tampilan aplikasi monitoring

Dalam pengujian tampilan ini akan dilakukan beberapa percobaan untuk menentukan bahwa aplikasi monitoring ini sesuai dengan apa yang di inginkan karena ini merupakan pengujian paling pokok pada remote monitoring tata suara panggung , pengujian tampilan monitoring tersebut meliputi :

1. Pengujian skin aplikasi monitoring

Pengujian tampilan yang tidak menggunakan komponen *alphaskin* dan yang sudah menggunakan *alphaskin*, guna memakai komponen ini hanya untuk memperbagus tampilan layaknya pemutar musik yang bisa diubah skin dan warnanya agar pada saat memonitoring kita bisa melihata tampilan yang lebih menarik dan bagus.



Gambar 4.5 tampilan monitoring tanpa alpha skin



Gambar 4.6 Tampilan monitoring setelah menggunakan alphaskin

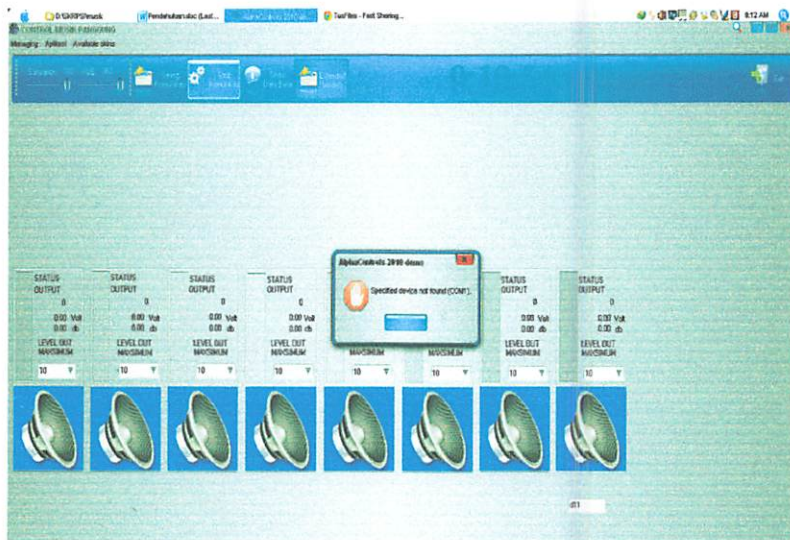
Jadi setelah dilakukan pengujian seperti di atas maka dapat dihasilkan tampilan yang lebih menarik karena menggunakan komponen alphaskin yang digunakan pada aplikasi monitoring.

2. Pengujian tampilan yang sudah terdapat komunikasi dan belum ada komunikasi yang masuk

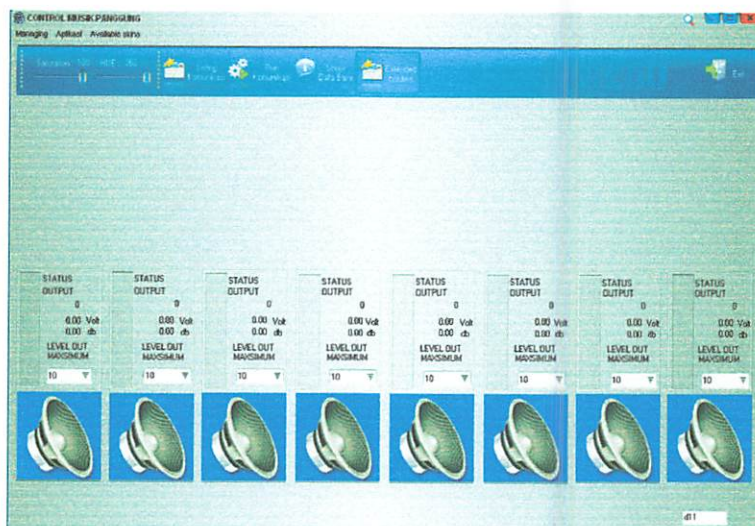
Pada pengujian ini yang dilakukan hanya mencoba aplikasi monitoring pada saat aplikasi terhubung dengan *hardware* dan tidak dengan menekan tombol komunikasi



untuk menjalankan aplikasi berikut pengujiannya.



Gambar 4.7 tampilan belum jika tidak ada komunikasi



Gambar 4.8 tampilan monitoring jika terdapat komunikasi

3. Pengujian tampilan monitoring dengan inputan dan tanpa inputan dari speaker

Pengujian ini merupakan paling penting dalam aplikasi monitoring karena pada proses pengujian ini akan menghubungkan langsung dengan *hardware* yang akan

mengirim data dari speaker dan akan di tampilan pada monitoring tata suara panggung berupa level-level seperti spectrum.



Gambar 4.9 tampilan monitoring sebelum ada inputan masuk dari speaker

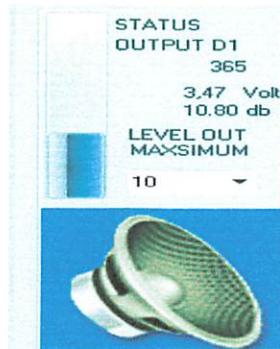


Gambar 4.10 Tampilan monitoring dengan inputan dari *hardware*

Pada gambar merupakan tampilan jika terdapat Inputan dari speaker, dan data yang di terima sudah disesuaikan dari *hardware* untuk menentukan masing-masing speaker yang di tampilkan levelnya pada aplikasi monitoring.

4.4.3 Pengujian Perhitungan Tegangan (Volt) dan Decibel (dB)

Pada pengujian hasil perhitungan ini terletak pada bar speaker yang hasil outputannya akan diketahui berapa decibel speaker akan mengalami gangguan (overload), jadi terdapat beberapa rumus untuk mengetahui hasil dari outputan yang rumus tersebut sudah dibuat di dalam aplikasi monitoring. Berikut merupakan tampilan bar speaker



Gambar 4.11 Tampilan bar speaker

Dalam hasil pengujian tampilan bar speaker dengan aplikasi monitoring ini outputan yang dikeluarkan adalah Tegangan = 3,47 V dan Decibel = 10,80 Db

Pada pengujian juga dilakukan pengecekan eror pengujian dengan perbandingan dari hasil perhitungan pada tampilan dan pada perhitungan dengan rumus dan avometer.

- Uji dengan perhitungan

$$V = (365 \times 0,0095) + 1,4$$

$$= 3,47 \text{ V}$$

$$\text{dB} = 20 \log 3,47$$

$$= 10,81$$

Diatas merupakan hasil perhitungan dengan menggunakan rumus yang kemudian dilakukan perhitungan untuk memperoleh eror pengujian yaitu dengan menggunakan rumus

Untuk pengukuran eror pengujian adalah :

$$= \frac{\text{ErorPengujian} - \text{ErorPerhitungan}}{\text{ErorPengujian}} \times 100\%$$

$$\frac{10,80-10,81}{10,80} \times 100\% = 0,001\%$$

Jadi untuk tingkat kesalahannya pada perhitungan dan tampilan adalah **0,001%**

4.4.4 Pengujian aplikasi dengan speaker pada panggung

Pada pengujian ini akan dilakukan pengujian untuk melakukan proses pengaplikasian program dengan speaker yang berada di panggung untuk mengetahui speaker yang mengalami kelebihan tingkatan suara dan speaker yang sedang aktif dan tidak pada.



Gambar 4.12 Pengujian Dengan 4 Speaker

Pada gambar tampilan diatas dilakukan pengujian dengan menggunakan 4 speaker guna untuk mengetahui mana speaker yang mengalami overload dan tidak ada juga speaker yang aktif dan tidak, berikut merupakan tampilan bar speaker yang sudah terhubung dengan masing-masing speaker.

1. Tampilan bar speaker yang pertama menunjukkan bahwa keadaan speaker sedang normal.
2. Tampilan bar speaker yang ke dua menunjukkan speaker sedang mati/tidak aktif, maka tidak terdapat inputan yang dikirim oleh speaker
3. Tampilan bar speaker yang ke tiga menunjukkan speaker dalam keadaan overload karena data yang diterima oleh program melebihi data yang sudah ditentukan pada program aplikasi motoring.
4. Tampilan bar speaker yang ke empat menunjukkan speaker sedang mati/tidak aktif, maka tidak terdapat inputan yang dikirim oleh speaker

4.4.5 Pengujian Jarak Pada Hardware

pada dasarnya jarak komunikasi hardware bekerja pada jarak 20 meter karena di dalam hardware komunikasi yang digunakan adalah modul TLP dan RLP yang menggunakan wireless untuk jalur komunikasi, karena itu pancaran sinyal yang dikirim dari hardware akan berkurang yang akan terjadi. Untuk membuktikan jarak pancar tersebut maka dilakukan pengujian komunikasi hardware dengan berbagai jarak pengujian. Data hasil pengujian disajikan dalam bentuk table.

No	Jarak (meter)	Kondisi (penghalang)	Kondisi (tampa penghalang)
1	1	Berhasil (data dikirim)	Berhasil (data dikirim)
2	2	Berhasil (data dikirim)	Berhasil (data dikirim)
3	4	Berhasil (data dikirim)	Berhasil (data dikirim)
4	6	Berhasil (data dikirim)	Berhasil (data dikirim)
5	8	Berhasil (data dikirim)	Berhasil (data dikirim)
6	10	Berhasil (data dikirim)	Berhasil (data dikirim)
7	12	Berhasil (data dikirim)	Berhasil (data dikirim)
8	14	Berhasil (data dikirim)	Berhasil (data dikirim)
9	16	Tidak berhasil	Berhasil (data dikirim)

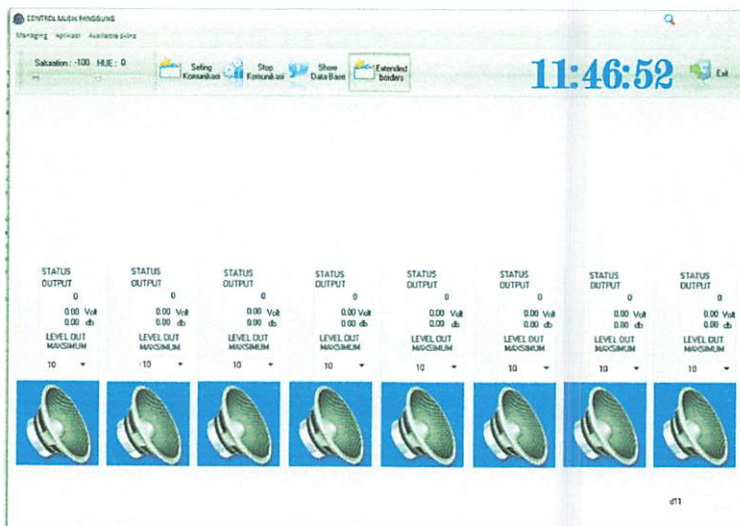
10	18	Tidak berhasil	Berhasil (data dikirim)
11	20	Tidak berhasil	Berhasil (data dikirim)

Table 4.1 hasil pengujian jarak

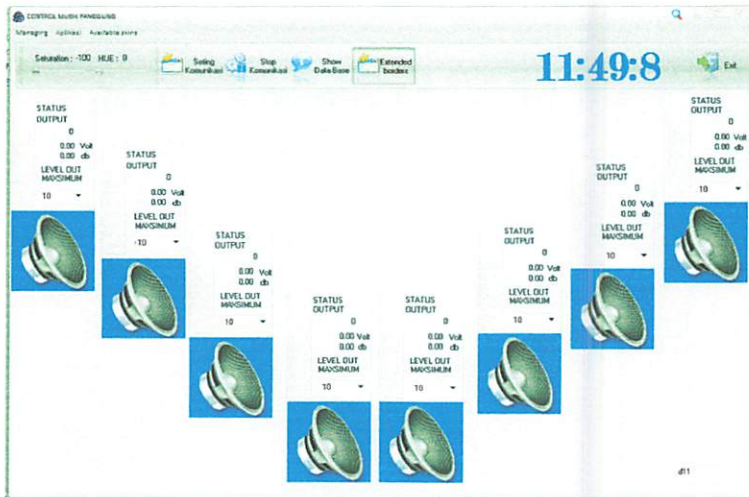
data hasil pengujian menunjukkan jika ada penghalang, pada jarak kurang dari 15 meter pengiriman data berhasil dengan baik, sedangkan jika tanpa penghalang, pengiriman data dapat dilakukan pada jarak lebih dari 15 meter hingga kurang 20 meter.

4.4.5 Pengujian drag pada bar speaker

Pada pengujian drag yang dilakukan dengan memindahkan *bar speaker* sesuai dengan tata letak panggung yang di harapkan dan juga dapat digunakan untuk mengetahui pada speaker yang mana yang mengalami Overload



Gambar 4.12 Tampilan monitoring normal.



Gambar 4.13 Tampilan setelah proses Drag

Dari gambar di atas adalah proses melakukan drag pada menu bar speaker untuk menentukan letak speaker pada panggung sehingga bentuk penataan speaker menjadi huruf V.

4.4.5 Tabel hasil pengujian

Pada table ini akan dilakukan pengecekan apakah aplikasi berhasil dan tidak sesuai dengan apa yang di inginkan

PENGUJIAN	BERHASIL	GAGAL
Komunikasi aplikasi dengan hardware	✓	
Tampilan aplikasi	✓	
Tampilan bar level untuk menampilkan data yang di terima	✓	
Perhitungan dB dan tegangan	✓	
Menerima data pada aplikasi	✓	
Drag	✓	

Table 4.2 hasil pengujian

BAB V

PENUTUP

5.1 Kesimpulan

Dengan dibangunnya program aplikasi monitoring tata suara panggung ini, maka akan muncul beberapa kesimpulan diantaranya :

1. Dengan memanfaatkan program aplikasi ini, proses menemukan gangguan pada speaker yang berada di atas panggung lebih mudah.
2. Kondisi normal dan overload pada speaker dapat di tampilkan pada bar speaker berupa level – level seperti equalizer sesuai dengan tegangan yang di keluarkan oleh speaker yang sudah di olah menjadi data digital, dengan kondisi batasan jika overload dapat di atur pada aplikasi.
3. Pengujian jarak pada aplikasi ini yang menggunakan modul TLP dan RLP di dalam *hardware*.
4. Pengujian pada aplikasi ini memiliki batasan jarak yaitu pada jarak 1 – 15 meter hardware mampu melakukan pengiriman data secara normal dan jika tanpa penghalang pengirim data dapat dilakukan melebihi dari 15 meter sampai kurang 20 meter.

5.2 Saran

Dalam pembuatan program aplikasi ini terdapat beberapa kendala yang sangat berpengaruh, sehingga program aplikasi ini masih jauh dari kesempurnaan. Untuk mendapatkan program aplikasi yang lebih baik bagi pengembang selanjutnya maka ada baiknya memperhatikan saran-saran sebagai berikut:

1. Sebaiknya hardware bisa melakukan komunikasi Dua arah agar bisa langsung di control dengan aplikasi monitoring

2. Untuk menyempurnakan program ini sebaiknya ditambahkan sebuah filter sehingga bisa menentukan nada-nada yang tidak sesuai dengan yang diinginkan.
3. Perancangan keamanan yang kuat apabila akan dikembangkan menjadi program kendali jarak jauh misalnya berbasis web.



LAMPIRAN



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

NAMA : IMAM BASHORI
NIM : 09.12.505
JURUSAN : Teknik Elektro S-1
KONSENTRASI : Teknik Komputer
MASA BIMBINGAN: SEMESTER GENAP 2012/2013
JUDUL : PERANCANGAN DAN PEMBUATAN USER INTERFACE
REMOTE MONITORING TATA SUARA PANGGUNG
MENGUNAKAN DELPHI 7.0

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Jum'at
Tanggal : 16 Agustus 2013
Dengan Nilai : 80,75 (A) *o*

PANITIA UJIAN SKRIPSI

Ketua Majelis Penguji

M. Ibrahim Ashari, ST, MT
NIP.P.1030100358

Sekretaris Majelis Penguji

Dr. Eng. Aryuanto S, ST, MT
NIP.P.1030800417

ANGGOTA PENGUJI

Dosen Penguji I

Irmalia Suryani Faradisa, ST, MT
NIP.P.103000365

Dosen Penguji II

Bambang Prio Hartono, ST, MT
NIP.Y.1028400082



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
 FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
 PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
 Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata Satu (S-1) Jurusan Teknik Elektro Konsentrasi Teknik Komputer, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : IMAM BASHORI
 NIM : 09.12.505
 JURUSAN : Teknik Elektro S-1
 KONSENTRASI : Teknik Komputer
 MASA BIMBINGAN: SEMESTER GENAP 2012/2013

**PERANCANGAN DAN PEMBUATAN USER INTERFACE
 REMOTE MONITORING TATA SUARA PANGGUNG
 MENGGUNAKAN DELPHI 7.0**

Tanggal	Uraian	Paraf
Penguji I 16 - 08 - 2013	1. Tata tulis sesuai silabus 2. Abstrak 3. Gambar 3.3 4. Rumus 5. Bab 4. Hal 48 dst	
Penguji II 16 - 08 - 2013	1. Penulisan di sesuaikan dengan panduan penulisan	

Disetujui,

Dosen Penguji I

Irmalia Suryani Faradisa, ST, MT
 NIP.P.103000365

Dosen Penguji II

Bambang Prio Hartono, ST, MT
 NIP.Y.1028400082

Mengetahui,

Dosen Pembimbing I

M. Ibrahim Ashari, ST, MT
 NIP.P.1030100358

Dosen Pembimbing II

Michael Ardita, ST, MT
 NIP.P.1031000434

**FORMULIR BIMBINGAN SKRIPSI**

Nama : Imam Bashori
Nim : 09.12.505
Masa Bimbingan : Semester Genap 2012-2013
Judul Skripsi : Perancangan Dan Pembuatan User Interface Remote Monitoring Tata Suara Panggung Menggunakan Delphi 7.0

No	Tanggal	Uraian	Paraf Pembimbing
1	24/4 13	konsultasi Laporan seminar proposal	
2	25/5 13	BAB I & II	
3	26/6 13	BAB III	
4	30/6 13	BAB IV	
5	10/7 13	Revisi penataan tulisan	
6	20/7 13	BAB V	
7	22/7 13	Acc Seminar Hasil	
8			
9			
10			

Malang,

Dosen pembimbing I

M. Ibrahim Ashari, ST, MT
NIP.P. 1030100358



FORMULIR BIMBINGAN SKRIPSI

Nama : Imam Bashori
Nim : 09.12.505
Masa Bimbingan : Semester Genap 2012-2013
Judul Skripsi : Perancangan Dan Pembuatan User Interface Remote Monitoring Tata Suara Panggung Menggunakan Delphi 7.0

No	Tanggal	Uraian	Paraf Pembimbing
1	12 / 13 / 05	konsultasi Latar Belakang	
2	20 / 13 / 05	konsultasi Tujuan & Batasan	
3	29 / 13 / 05	BAB I & II	
4	16 / 13 / 05	BAB III	
5	25 / 13 / 06	REVISI BAB III	
6	05 / 13 / 7	KONSULTASI BAB IV & V	
7	17 / 13 / 7	REVISI BAB IV & V	
8	22 / 13 / 7	Acc Seminar Hasil	
9			
10			

Malang, 23 September 2013

Dosen pembimbing II

Michael Ardhita, ST, MT
NIP.P.1031000434



PERMOHONAN PERSETUJUAN SKRIPSI

Yang Bertanda Tangan Dibawah Ini:

Nama : IMAM BASHORI
 N I M : 0912505
 Semester : 8 (delapan)
 Fakultas : Teknologi Industri
 Jurusan : Teknik Elektro S-I
 Konsentrasi : TEKNIK ENERGI LISTRIK
TEKNIK ELEKTRONIKA
TEKNIK KOMPUTER DAN INFORMATIKA
TEKNIK KOMPUTER
TEKNIK TELEKOMUNIKASI

Alamat :

Dengan ini kami mengajukan permohonan untuk mendapatkan persetujuan untuk membuat SKRIPSI Tingkat Sarjana. Untuk melengkapi permohonan tersenut, bersama ini kami lampirkan persyaratan-persyaratan yang harus dipenuhi.


Adapun persyaratan- persyaratan pengambilan SKRIPSI adalah sebagai berikut:

1. Telah melaksanakan semua praktikum sesuai dengan konsentrasinya (.....)
2. Telah lulus dan menyerahkan laporan Praktek Kerja (.....)
3. Telah lulus seluruh mata kuliah keahlian (MKB)sesuai konsentrasinya (.....)
4. Telah menempuh matakuliah > 134 sks dengan IPK > 2 dan tidak ada nilai E (.....)
5. Telah mengikuti secara aktif kegiatan seminar Skripsi yang diadakan Jurusan (.....)
6. Memenuhi persyaratan administrasi (.....)

Demikian permohonan ini untuk mendapatkan penyelesaian lebih lanjut dan atas perhatiannya kami ucapkan terima kasih.


Telah diteliti kebenarannya data tersebut diatas

Recording Teknik Elektro S-I


 (.....)

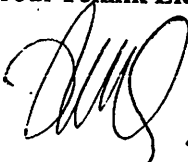
Malang,.....201

Pemohon


 (.....)

Disetujui

Ketua Prodi Teknik Elektro S-I

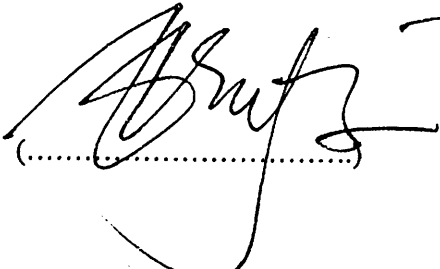


M. Ibrahim Ashari, ST, MT

NIP. P. 1030100358

Mengetahui

Dosen Wali


 (.....)

Catatan:

Bagi mahasiswa yang telah memenuhi persyaratan mengambil SKRIPSI agar membuat proposal dan mendapat persetujuan dari Ketua Prodi T. elektro S-I



PROGRAM STUDI TEKNIK ELEKTRO S-1

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI NASIONAL MALANG

Kampus II : Jl. Raya Karanglo Km. 2 Telp. (0341) 417636 Malang

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i :

Nama : **IMAM BASHORI**

Nim : **0912505**

Semester : **VIII (Delapan)**

Jurusan : **Teknik Elektro S-1**

Konsentrasi : **Teknik Komputer**

Dengan ini menyatakan bersedia/tidak bersedia*) Membimbing skripsi dari mahasiswa tersebut, dengan judul :

" PERANCANGAN DAN PEMBUATAN USER INTERFACE UNTUK REMOTE MONITORING TATA SUARA PANGGUNG MENGGUNAKAN DELPHI 7.0"

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Hormat Kami

M. Ibrahim Ashari, ST, MT

NIP.P. 1030100358

*) Coret yang tidak perlu



PROGRAM STUDI TEKNIK ELEKTRO S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
Kampus II : Jl. Raya Karanglo Km. 2 Telp. (0341) 417636 Malang

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i :

Nama : **IMAM BASHORI**
Nim : **0912505**
Semester : **VIII (Delapan)**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer**

Dengan ini menyatakan bersedia/tidak bersedia*) Membimbing skripsi dari mahasiswa tersebut, dengan judul :

" PERANCANGAN DAN PEMBUATAN USER INTERFACE UNTUK REMOTE MONITORING TATA SUARA PANGGUNG MENGGUNAKAN DELPHI 7.0"

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Hormat Kami

Michael Ardita, ST, MT

NIP.P. 1031000434

Catatan :

Setelah disetujui agar formulir ini Diserahkan mahasiswa/i yang bersangkutan kepada jurusan untuk diproses lebih lanjut

*) Coret yang tidak perlu



PROGRAM STUDI TEKNIK ELEKTRO S-1

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI NASIONAL MALANG

Kampus II : Jl. Raya Karanglo Km. 2 Telp. (0341) 417636 Malang

jumlah : 1 (satu) berkas

Pembimbing Skripsi

di atas : Yth. Bapak/Ibu **M. Ibrahim Ashari, ST, MT**

Dosen Teknik Elektro S-1

ITN Malang

Yang bertanda tangan dibawah

Nama : **IMAM BASHORI**

Nim : **0912505**

Jurusan : **Teknik Elektro S-1**

Konsentrasi : **Teknik Komputer**

Dengan ini mengajukan permohonan, kiranya Bapak/Ibu bersedia menjadi Dosen Pembimbing untuk penyusunan Skripsi dengan judul :

"PERANCANGAN DAN PEMBUATAN USER INTERFACE UNTUK REMOTE MONITORING TATA SUARA PANGGUNG MENGGUNAKAN DELPHI 7.0"

Demikian permohonan kami buat dan atas kesediaan Bapak kami ucapkan terima kasih.

Mengetahui

Program Studi Teknik Elektro S-1

M. Ibrahim Ashari, ST, MT

NIP.P. 1030100358

Hormat Kami

IMAM BASHORI

NIM. 0912505



PROGRAM STUDI TEKNIK ELEKTRO S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
Kampus II : Jl. Raya Karanglo Km. 2 Telp. (0341) 417636 Malang

Lampiran : 1 (satu) berkas
Pembimbing Skripsi

Kepada : Yth. Bapak/Ibu **Michael Ardita, ST, MT**
Dosen Teknik Elektro S-1
ITN Malang

Yang bertanda tangan dibawah

Nama : **IMAM BASHORI**
Nim : **0912505**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer**

Dengan ini mengajukan permohonan, kiranya Bapak/Ibu bersedia menjadi Dosen Pembimbing untuk penyusunan Skripsi dengan judul :

"PERANCANGAN DAN PEMBUATAN USER INTERFACE UNTUK REMOTE MONITORING TATA SUARA PANGGUNG MENGGUNAKAN DELPHI 7.0"

Demikian permohonan kami buat dan atas kesediaan Bapak kami ucapkan terima kasih.

Mengetahui

Ketua Program Studi Teknik Elektro S-1

M. Ibrahim Ashari, ST, MT

NIP.P. 1030100358



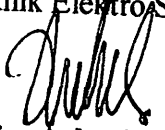

Hormat Kami

IMAM BASHORI

NIM. 0912505



BERITA ACARA SEMINAR PROPOSAL SKRIPSI
PROGRAM STUDI TEKNIK ELEKTRO S-1
Konsentrasi : Teknik Telekomunikasi

1.	Nim	: 0912713		
2.	Nama	: NELSON PEREIRA BELO		
3.	Konsentrasi Jurusan	: Teknik Telekomunikasi		
4.	Jadwal Pelaksanaan:	Waktu	Tempat	
		30 April 2013	09.00	III.1.3
5.	Judul proposal yang diseminarkan Mahasiswa	PERANCANGAN DAN PEMBUATAN SWEEP GENERATOR UNTUK PENGUJIAN RESPON FILTER IF AM 455 KHZ BERBASIS MIKROKONTROLER ATMEGA 16		
6.	Perubahan judul yang diusulkan oleh Kelompok Dosen Keahlian			
7.	Catatan :			
8.	Catatan :			
	Persetujuan judul Skripsi			
	Disetujui, Dosen Keahlian I	Disetujui, Dosen Keahlian II	Disetujui, Dosen Keahlian III	
	 (Bambang Priso H)	 (Michael Ardita)	(.....)	
	Mengetahui, Ketua Program Studi Teknik Elektro S-1	Disetujui, Calon Dosen Pembimbing ybs		
Pembimbing I		Pembimbing II		
 M. Ibrahim Ashari, ST, MT NIP. P 1030100358	(.....)	 (SOTYO HADI, ST)		

```
unit MainUnit;  
{ $WARNINGS OFF }  
{ $I sDefs.inc }
```

interface

uses

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
sSkinProvider, sSkinManager, sComboBoxes, ExtCtrls, sPanel, FileCtrl,  
sScrollBar,  
ImgList, sStatusBar, sGauge, sCheckbox, ExtDlgs, ComCtrls, StdCtrls,  
Buttons, sSpeedButton, sComboBox, sHintManager, sDialogs, sMemo, CheckLst,  
sFrameBar, sLabel, Menus,  
Mask, sMaskEdit, sCustomComboEdit, sTooledit, sTrackBar, sEdit, ActnList,  
StdActns, ToolWin, acMagn, XPMAN, sListBox,  
acProgressBar, sButton, acAlphaImageList, acAlphaHints, acPNG, sBitBtn,  
Grids, DBGrids, VaClasses, VaComm, DB, ADOdb;
```

type

```
TMainForm = class(TForm)  
  MainMenu: TMainMenu;  
  File1: TMenuItem;  
  Exit1: TMenuItem;  
  AvailableSkinsMenu: TMenuItem;  
  Skinned1: TMenuItem;  
  Enabled1: TMenuItem;  
  sSkinProvider1: TsSkinProvider;  
  N1: TMenuItem;  
  BuildinSkins: TMenuItem;  
  ExternalSkins: TMenuItem;  
  PopupMenu1: TPopupMenu;  
  Skinned2: TMenuItem;  
  Hints2: TMenuItem;  
  Enabled2: TMenuItem;  
  Exit2: TMenuItem;  
  MainActionList: TActionList;  
  ActionSkinned: TAction;  
  ActionHintsDisable: TAction;  
  ActionEnabled: TAction;  
  ActionClose: TWindowClose;  
  ActionOverView: TAction;  
  ActionAnimation: TAction;  
  Allowanimation1: TMenuItem;  
  Allowanimation2: TMenuItem;  
  N3: TMenuItem;  
  ActionHintsSkinned: TAction;  
  ActionHintsCustom: TAction;  
  Skinnedhints1: TMenuItem;  
  Customhints1: TMenuItem;  
  Hintskind1: TMenuItem;  
  Hintsshowing1: TMenuItem;  
  Custom1: TMenuItem;  
  Skinned3: TMenuItem;  
  Disabled1: TMenuItem;  
  sSkinManager1: TsSkinManager;  
  Dialogs1: TMenuItem;  
  Standard1: TMenuItem;  
  miOpenDialog1: TMenuItem;  
  miSaveDialog1: TMenuItem;  
  SelectSkin1: TMenuItem;  
  N2: TMenuItem;  
  ActionCloseForm: TAction;  
  ImageList32: TsAlphaImageList;  
  ActionHintsStd: TAction;  
  Standard2: TMenuItem;  
  Standard3: TMenuItem;  
  N4: TMenuItem;  
  N5: TMenuItem;  
  sCoolBar1: TsPanel;  
  sPanel1: TsPanel;  
  sStickyLabel1: TsStickyLabel;  
  sStickyLabel3: TsStickyLabel;  
  sLabel1: TsLabel;  
  sLabel2: TsLabel;  
  sTrackBar1: TsTrackBar;
```

```
sTrackBar2: TsTrackBar;
sSpeedButton3: TsSpeedButton;
sSpeedButton1: TsSpeedButton;
sSpeedButton5: TsSpeedButton;
sSpeedButton4: TsSpeedButton;
sBitBtn1: TsSpeedButton;
sPanel2: TsPanel;
sPanel3: TsPanel;
spnl1: TsPanel;
lb11: TsLabel;
lb12: TsLabel;
cbb1: TsComboBox;
spnl2: TsPanel;
spnl3: TsPanel;
img1: TImage;
img2: TImage;
img3: TImage;
spnl4: TsPanel;
spnl5: TsPanel;
img4: TImage;
spnl6: TsPanel;
img5: TImage;
spnl7: TsPanel;
spnl8: TsPanel;
img6: TImage;
img7: TImage;
img9: TImage;
sprgrsbr4: TsProgressBar;
sprgrsbr5: TsProgressBar;
sprgrsbr6: TsProgressBar;
sprgrsbr7: TsProgressBar;
sprgrsbr8: TsProgressBar;
sprgrsbr9: TsProgressBar;
sprgrsbr10: TsProgressBar;
lb13: TsLabel;
lb14: TsLabel;
lb15: TsLabel;
lb16: TsLabel;
lb17: TsLabel;
lb18: TsLabel;
lb19: TsLabel;
lb110: TsLabel;
lb111: TsLabel;
lb112: TsLabel;
lb113: TsLabel;
lb114: TsLabel;
lb115: TsLabel;
lb116: TsLabel;
cbb2: TsComboBox;
cbb3: TsComboBox;
cbb4: TsComboBox;
cbb5: TsComboBox;
cbb6: TsComboBox;
cbb7: TsComboBox;
cbb8: TsComboBox;
lb117: TsLabel;
lb118: TsLabel;
lb119: TsLabel;
lb120: TsLabel;
lb121: TsLabel;
lb122: TsLabel;
lb123: TsLabel;
lb124: TsLabel;
lb125: TsLabel;
lb126: TsLabel;
lb127: TsLabel;
lb128: TsLabel;
lb129: TsLabel;
lb130: TsLabel;
lb131: TsLabel;
lb132: TsLabel;
lb133: TsLabel;
lb134: TsLabel;
lb135: TsLabel;
lb136: TsLabel;
```

```
lbl37: TLabel;
lbl38: TLabel;
lbl39: TLabel;
lbl40: TLabel;
sprgrsbr1: TsProgressBar;
vcml: TVaComm;
ds1: TDataSource;
tbl1: TADOTable;
con1: TADOConnection;
atncfldtbl1No: TAutoIncField;
wdsrngfldtbl1Tgl: TWideStringField;
wdsrngfldtbl1Jam: TWideStringField;
wdsrngfldtbl1Data1: TWideStringField;
wdsrngfldtbl1Data2: TWideStringField;
wdsrngfldtbl1Data3: TWideStringField;
wdsrngfldtbl1Data4: TWideStringField;
wdsrngfldtbl1Data5: TWideStringField;
wdsrngfldtbl1Data6: TWideStringField;
wdsrngfldtbl1Data7: TWideStringField;
wdsrngfldtbl1Data8: TWideStringField;
spnl12: TsPanel;
spnl9: TsPanel;
btn1: TsButton;
edt2: TEdit;
edt3: TEdit;
edt4: TEdit;
edt5: TEdit;
edt6: TEdit;
edt7: TEdit;
edt8: TEdit;
mmo1: TMemo;
edt9: TEdit;
edt10: TEdit;
btn3: TButton;
pn11: TPanel;
lbl44: TsWebLabel;
tmr1: TTimer;
edt1: TsEdit;
lbl45: TLabel;
lbl46: TLabel;
lbl47: TLabel;
lbl48: TLabel;
lbl49: TLabel;
lbl50: TLabel;
lbl51: TLabel;
lbl52: TLabel;
lbl53: TLabel;
lbl54: TLabel;
lbl55: TLabel;
lbl56: TLabel;
lbl57: TLabel;
lbl58: TLabel;
lbl59: TLabel;
lbl60: TLabel;
lbl61: TLabel;
lbl62: TLabel;
lbl63: TLabel;
lbl64: TLabel;
lbl65: TLabel;
lbl66: TLabel;
lbl67: TLabel;
lbl68: TLabel;
lbl69: TLabel;
lbl70: TLabel;
lbl71: TLabel;
lbl72: TLabel;
lbl73: TLabel;
lbl74: TLabel;
lbl75: TLabel;
lbl76: TLabel;
procedure SpeedShow(Sender: TObject);
procedure sSpeedButton1Click(Sender: TObject);
procedure SkinMenuClick(Sender: TObject);
procedure ActionAnimationExecute(Sender: TObject);
procedure FormCreate(Sender: TObject);
```

```
procedure miOpenDialog1Click(Sender: TObject);
procedure miSaveDialog1Click(Sender: TObject);
procedure SelectSkin1Click(Sender: TObject);
procedure ActionCloseFormExecute(Sender: TObject);
procedure sTrackBar1Change(Sender: TObject);
procedure sTrackBar2Change(Sender: TObject);
procedure sSpeedButton3Click(Sender: TObject);
procedure sSpeedButton5Click(Sender: TObject);
procedure sSpeedButton4Click(Sender: TObject);
procedure sprgrsbr1MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure sprgrsbr1MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure sprgrsbr2MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure spn11MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure spn11MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure spn11MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure sprgrsbr6MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure spn12MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure spn13MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure spn14MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure spn15MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure spn16MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure spn18MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure spn17MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure spn12MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure spn13MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure spn14MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure spn15MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure spn18MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure spn17MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure spn12MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure spn13MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure spn14MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure spn15MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure spn16MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure spn18MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure spn17MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure spn16MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure btn1Click(Sender: TObject);
procedure vcmlRxChar(Sender: TObject; Count: Integer);
procedure tmr1Timer(Sender: TObject);
  private
  { Private declarations }
  GrabDeltaX, GrabDeltaY: integer;
  MouseDown: boolean;
  Pesan : string;
  simpan : Integer;
public
```

```
    PressedBtn : TObject;  
    procedure GenerateSkinsList;  
end;
```

var

```
    IniName : string;  
    MainForm: TMainForm;  
    Loading : boolean;  
    OldFrame, CurrentFrame : TFrame;  
    Animated : boolean = True;  
    DialogLoading : boolean = False;  
  
    AppLoading : boolean;  
    FormShown : boolean = False;
```

implementation

```
uses sMaskData, sStyleSimply, sSkinProps, ShellApi, sMessages, sStoreUtils,  
    sGraphUtils, sVclUtils,  
    sHintDesigner, acntUtils, sConst, acSelectSkin, sCommonData, Unit1,Unit2;
```

```
{ $R *.DFM }
```

```
procedure TMainForm.FormShow(Sender: TObject);
```

```
begin
```

```
    if FormShown then Exit;  
    AppLoading := True;  
    FormShown := True;  
    GenerateSkinsList;  
    AppLoading := False;
```

```
end;
```

```
procedure TMainForm.sSpeedButton1Click(Sender: TObject);
```

```
begin
```

```
    ActionAnimation.Execute;  
    vcml.Open;
```

```
end;
```

```
procedure TMainForm.SkinMenuClick(Sender: TObject);
```

```
begin
```

```
    sSkinManager1.SkinName := DelChars(TMenuItem(Sender).Caption, '&');
```

```
end;
```

```
procedure TMainForm.GenerateSkinsList;
```

```
var
```

```
    sl : TacStringList;  
    i : integer;  
    mi : TMenuItem;
```

```
begin
```

```
    sl := TacStringList.Create;  
    sSkinManager1.GetSkinNames(sl);  
    Buildinskins.Clear;  
    for i := 0 to sSkinManager1.InternalSkins.Count - 1 do begin  
        mi := TMenuItem.Create(Application);  
        mi.Caption := sSkinManager1.InternalSkins[i].Name;  
        if mi.Caption = sSkinManager1.SkinName then mi.Checked := True;  
        mi.OnClick := SkinMenuClick;  
        mi.RadioButton := True;  
        Buildinskins.Add(mi);  
    end;
```

```
    Externalskins.Clear;
```

```
    sl := TacStringList.Create;  
    sSkinManager1.GetExternalSkinNames(sl);
```

```
    if sl.Count > 0 then begin
```

```
        sl.Sort;  
        for i := 0 to sl.Count - 1 do begin  
            mi := TMenuItem.Create(Application);  
            mi.Caption := sl[i];  
            if mi.Caption = sSkinManager1.SkinName then mi.Checked := True;  
            mi.OnClick := SkinMenuClick;  
            mi.RadioButton := True;  
            if (i <> 0) and (i mod 20 = 0) then mi.Break := mbBreak;  
            Externalskins.Add(mi);  
        end;
```

```
    end;
```

```
    end;
    FreeAndNil(sl);
end;

procedure TMainForm.ActionAnimationExecute(Sender: TObject);
begin
    Animated := not ActionAnimation.Checked; // Saving option in variable
    ActionAnimation.Checked := Animated;
    if Animated then begin
        sSpeedButton1.Caption := 'Stop'#13#10'Komunikasi';
        sSpeedButton1.ImageIndex := 0;
    end
    else
        begin
            sSpeedButton1.Caption := 'Run'#13#10'Komunikasi';
            sSpeedButton1.ImageIndex := 1;
        end;
end;

procedure TMainForm.FormCreate(Sender: TObject);
begin
    simpan := 0;
    MouseDown := false;
    DoubleBuffered := true;
    spn12.Hide;
end;

procedure TMainForm.miOpenDialog1Click(Sender: TObject);
begin
    sprgrsbr4.SMOOTH := True;
    sprgrsbr1.SMOOTH := True;
    sprgrsbr5.SMOOTH := True;
    sprgrsbr6.SMOOTH := True;
    sprgrsbr7.SMOOTH := True;
    sprgrsbr8.SMOOTH := True;
    sprgrsbr9.SMOOTH := True;
    sprgrsbr10.SMOOTH := True;
end;

procedure TMainForm.miSaveDialog1Click(Sender: TObject);
begin
    sprgrsbr1.SMOOTH := False;
    sprgrsbr4.SMOOTH := False;
    sprgrsbr5.SMOOTH := False;
    sprgrsbr6.SMOOTH := False;
    sprgrsbr7.SMOOTH := False;
    sprgrsbr8.SMOOTH := False;
    sprgrsbr9.SMOOTH := False;
    sprgrsbr10.SMOOTH := False;
end;

procedure TMainForm.SelectSkin1Click(Sender: TObject);
begin
    SelectSkin(sSkinManager1);
end;

procedure TMainForm.ActionCloseFormExecute(Sender: TObject);
begin
    Close;
end;

procedure TMainForm.sTrackBar1Change(Sender: TObject);
begin
    if not aSkinChanging and (sSkinManager1.HueOffset <> sTrackBar1.Position) then
        begin // If not in a skin changing (global variable from AC package used)
            sSkinManager1.BeginUpdate;
            sLabell1.Caption := IntToStr(sTrackBar1.Position);
            sSkinManager1.HueOffset := sTrackBar1.Position;
            sSkinManager1.EndUpdate(True, False);
        end;
end;

procedure TMainForm.sTrackBar2Change(Sender: TObject);
```

```
begin
  if not aSkinChanging and (sSkinManager1.Saturation <> sTrackBar2.Position) then
    begin // If not in a skin changing (global variable from AC package used)
      sSkinManager1.BeginUpdate;
      sLabel2.Caption := IntToStr(sTrackBar2.Position);
      sSkinManager1.Saturation := sTrackBar2.Position;
      sSkinManager1.EndUpdate(True, False);
    end;
end;
```

```
procedure TMainForm.sSpeedButton3Click(Sender: TObject);
begin
  Form1.Show; // SelectSkin1.Click
end;
```

```
procedure TMainForm.sSpeedButton5Click(Sender: TObject);
begin
  sSkinManager1.ExtendedBorders := not sSkinManager1.ExtendedBorders;
end;
```

```
procedure TMainForm.sSpeedButton4Click(Sender: TObject);
begin
  form2.Show;
end;
```

```
procedure TMainForm.sprgrsbr1MouseDown(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
  GrabDeltaX := x;
  GrabDeltaY := y;
  MouseDown := true;
end;
```

```
procedure TMainForm.sprgrsbr1MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  MouseDown := False;
end;
```

```
procedure TMainForm.sprgrsbr2MouseDown(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
  GrabDeltaX := x;
  GrabDeltaY := y;
  MouseDown := true;
end;
```

```
procedure TMainForm.spnllMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  GrabDeltaX := x;
  GrabDeltaY := y;
  MouseDown := true;
end;
```

```
procedure TMainForm.spnllMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  if MouseDown then
    begin
      spnll.Left := spnll.Left + x - GrabDeltaX;
      spnll.Top := spnll.Top + y - GrabDeltaY;
      spnll.Repaint;
    end;
end;
```

```
procedure TMainForm.spnllMouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  MouseDown := False;
end;
```

```
procedure TMainForm.sprgrsbr6MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
  MouseDown := True;

end;
```

```
procedure TMainForm.spnl2MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  GrabDeltaX := x;
  GrabDeltaY := y;
  MouseDown := true;
end;
```

```
procedure TMainForm.spnl3MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  GrabDeltaX := x;
  GrabDeltaY := y;
  MouseDown := true;
end;
```

```
procedure TMainForm.spnl4MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  GrabDeltaX := x;
  GrabDeltaY := y;
  MouseDown := true;
end;
```

```
procedure TMainForm.spnl5MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  GrabDeltaX := x;
  GrabDeltaY := y;
  MouseDown := true;
end;
```

```
procedure TMainForm.spnl6MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  GrabDeltaX := x;
  GrabDeltaY := y;
  MouseDown := true;
end;
```

```
procedure TMainForm.spnl8MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  GrabDeltaX := x;
  GrabDeltaY := y;
  MouseDown := true;
end;
```

```
procedure TMainForm.spnl7MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  GrabDeltaX := x;
  GrabDeltaY := y;
  MouseDown := true;
end;
```

```
procedure TMainForm.spnl2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  if MouseDown then
  begin
    spnl2.Left := spnl2.Left + x - GrabDeltaX;
    spnl2.Top := spnl2.Top + y - GrabDeltaY;
    spnl2.Repaint;
    //sStatusbar1.Panels[0].Text := format( 'x=%d;x=%d', [sprgrsbr1.Left,
    sprgrsbr1.Top] );
```

```
    end;

end;

procedure TMainForm.spnl3MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  if MouseDown then
  begin
    spnl3.Left := spnl3.Left + x - GrabDeltaX;
    spnl3.Top := spnl3.Top + y - GrabDeltaY;
    spnl3.Repaint;
    //sStatusBar1.Panels[0].Text := format( 'x=%d;x=%d', [sprgrsbr1.Left,
    sprgrsbr1.Top] );
    end;

end;

procedure TMainForm.spnl4MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  if MouseDown then
  begin
    spnl4.Left := spnl4.Left + x - GrabDeltaX;
    spnl4.Top := spnl4.Top + y - GrabDeltaY;
    spnl4.Repaint;
    //sStatusBar1.Panels[0].Text := format( 'x=%d;x=%d', [sprgrsbr1.Left,
    sprgrsbr1.Top] );
    end;

end;

procedure TMainForm.spnl5MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  if MouseDown then
  begin
    spnl5.Left := spnl5.Left + x - GrabDeltaX;
    spnl5.Top := spnl5.Top + y - GrabDeltaY;
    spnl5.Repaint;
    //sStatusBar1.Panels[0].Text := format( 'x=%d;x=%d', [sprgrsbr1.Left,
    sprgrsbr1.Top] );
    end;

end;

procedure TMainForm.spnl8MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  if MouseDown then
  begin
    spnl8.Left := spnl8.Left + x - GrabDeltaX;
    spnl8.Top := spnl8.Top + y - GrabDeltaY;
    spnl8.Repaint;
    //sStatusBar1.Panels[0].Text := format( 'x=%d;x=%d', [sprgrsbr1.Left,
    sprgrsbr1.Top] );
    end;

end;

procedure TMainForm.spnl7MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  if MouseDown then
  begin
    spnl7.Left := spnl7.Left + x - GrabDeltaX;
    spnl7.Top := spnl7.Top + y - GrabDeltaY;
    spnl7.Repaint;
    //sStatusBar1.Panels[0].Text := format( 'x=%d;x=%d', [sprgrsbr1.Left,
    sprgrsbr1.Top] );
    end;

end;

procedure TMainForm.spnl2MouseDown(Sender: TObject; Button: TMouseButton;
```

```
    Shift: TShiftState; X, Y: Integer);
begin
MouseDown := False;
end;

procedure TMainForm.spnl3MouseUp(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
MouseDown := False;
end;

procedure TMainForm.spnl4MouseUp(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
MouseDown := False;
end;

procedure TMainForm.spnl5MouseUp(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
MouseDown := False;
end;

procedure TMainForm.spnl6MouseUp(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
MouseDown := False;
end;

procedure TMainForm.spnl8MouseUp(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
MouseDown := False;
end;

procedure TMainForm.spnl7MouseUp(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
MouseDown := False;
end;

procedure TMainForm.spnl6MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
begin
if MouseDown then
begin
    spnl6.Left := spnl6.Left + x - GrabDeltaX;
    spnl6.Top := spnl6.Top + y - GrabDeltaY;
    spnl6.Repaint;
    //sStatusbar1.Panels[0].Text := format( 'x=%d;y=%d', [sprgrsbr1.Left,
    sprgrsbr1.Top] );
end;

end;

procedure TMainForm.btn1Click(Sender: TObject);
var
    AA, BB : Integer;
begin
lbl19.Caption := edt1.Text;
AA := StrToInt(edt1.Text);
BB := StrToInt(cbbl1.Text);
if AA < BB then
begin
    lbl17.Caption := 'NORMAL';
    lbl20.Font.Color := clBlue;
    spnl1.Color := clBlue;
end;
if AA >= BB then
begin
    lbl17.Caption := 'OVERLOADE';
    lbl20.Font.Color := clMaroon;
    spnl1.Color := clMaroon;
end;
sprgrsbr1.Position:=StrToInt(Trim(lbl19.Caption));
```

```
//pbl.Position:=StrToInt(Trim(spnl9.Caption));  
end;
```

```
procedure TMainForm.vcm1RxChar(Sender: TObject; Count: Integer);
```

```
VAR
```

```
I: Integer;  
Tmp: string;  
data : string;  
AA,BB : Integer;  
cc,dd: Real;
```

```
begin
```

```
//baca data yang diterima oleh comport  
Tmp := vcm1.ReadText;
```

```
for I := 1 to Length(Tmp) do
```

```
case Tmp[I] of
```

```
#10:; //lewatkan
```

```
#13: //Tunggu Enter
```

```
begin
```

```
mm01.Lines.Add(pesan);;
```

```
pesan := '';
```

```
data := edt1.Text ;
```

```
if pos('d1',data)=1 then
```

```
begin
```

```
lbl19.Caption:=Copy(data,3,4);
```

```
AA := StrToInt(lbl19.Caption);
```

```
BB := StrToInt(cbb1.Text);
```

```
if AA = 0 then
```

```
begin
```

```
lbl45.Caption := '0.00';
```

```
lbl47.Caption := '0.00';
```

```
end
```

```
else
```

```
begin
```

```
cc := (AA * 0.0095)+1.4;
```

```
dd := (((Ln(cc))/(Ln(10)))* 20 );
```

```
lbl45.Caption := Copy(FloatToStr(cc),1,5);
```

```
lbl47.Caption := Copy(FloatToStr(dd),1,5);
```

```
end ;
```

```
if AA < BB then
```

```
begin
```

```
lbl17.Caption := 'NORMAL';
```

```
spn11.Color := clBlue;
```

```
end;
```

```
if AA >= BB then
```

```
begin
```

```
lbl17.Caption := 'OVERLOADE';
```

```
spn11.Color := clMaroon;
```

```
end;
```

```
sprgrsbr1.Position:=StrToInt(Trim(lbl19.Caption));
```

```
end;
```

```
if pos('d2',data)=1then
```

```
begin
```

```
lbl22.Caption:=Copy(data,3,4);
```

```
AA := StrToInt(lbl22.Caption);
```

```
BB := StrToInt(cbb2.Text);
```

```
if AA = 0 then
```

```
begin
```

```
lbl49.Caption := '0.00';
```

```
lbl56.Caption := '0.00';
```

```
end
```

```
else
```

```
begin
```

```
cc := (AA * 0.0095)+1.4;
```

```
dd := (((Ln(cc))/(Ln(10)))* 20 );
```

```
lbl49.Caption := Copy(FloatToStr(cc),1,5);
```

```
lbl56.Caption := Copy(FloatToStr(dd),1,5);
```

```
end ;
```

```
if AA < BB then
```

```
begin
```

```
lbl20.Caption := 'NORMAL';
```

```
        lbl20.Font.Color := clBlue;
        spnl2.Color := clBlue;
    end;
    if AA >= BB then
    begin
        lbl20.Caption := 'OVERLOADE';
        lbl20.Font.Color := clMaroon;
        spnl2.Color := clMaroon;
    end;
        sprgrsbr4.Position:=StrToInt(Trim(lbl22.Caption));
end;
if pos('d3',data)=1then
begin
    lbl25.Caption:=Copy(data,3,4);
    AA := StrToInt(lbl25.Caption);
    BB := StrToInt(cbb3.Text);
    if AA = 0 then
    begin
        lbl50.Caption := '0.00';
        lbl57.Caption := '0.00';
    end
    else
    begin
        cc := (AA * 0.0095)+1.4;
        dd := ((Ln(cc))/(Ln(10)))* 20 );
        lbl50.Caption := Copy(FloatToStr(cc),1,5);
        lbl57.Caption := Copy(FloatToStr(dd),1,5);
    end ;

    if AA < BB then
    begin
        lbl23.Caption := 'NORMAL';
        lbl20.Font.Color := clBlue;
        spnl3.Color := clBlue;
    end;
    if AA >= BB then
    begin
        lbl23.Caption := 'OVERLOADE';
        lbl20.Font.Color := clMaroon;
        spnl3.Color := clMaroon;
    end;
        sprgrsbr5.Position:=StrToInt(Trim(lbl25.Caption));
end;

if pos('d4',data)=1then
begin
    lbl28.Caption:=Copy(data,3,4);
    AA := StrToInt(lbl28.Caption);
    BB := StrToInt(cbb4.Text);
    if AA = 0 then
    begin
        lbl51.Caption := '0.00';
        lbl58.Caption := '0.00';
    end
    else
    begin
        cc := (AA * 0.0095)+1.4;
        dd := ((Ln(cc))/(Ln(10)))* 20 );
        lbl51.Caption := Copy(FloatToStr(cc),1,5);
        lbl58.Caption := Copy(FloatToStr(dd),1,5);
    end ;

    if AA < BB then
    begin
        lbl26.Caption := 'NORMAL';
        lbl20.Font.Color := clBlue;
        spnl4.Color := clBlue;
    end;
    if AA >= BB then
    begin
        lbl26.Caption := 'OVERLOADE';
        lbl20.Font.Color := clMaroon;
        spnl4.Color := clMaroon;
    end;
        sprgrsbr6.Position:=StrToInt(Trim(lbl28.Caption));
```

```
end;

if pos('d5',data)=1then
begin
  lbl31.Caption:=Copy(data,3,4);
  AA := StrToInt(lbl31.Caption);
  BB := StrToInt(cbb5.Text);
  if AA = 0 then
  begin
    lbl52.Caption := '0.00';
    lbl59.Caption := '0.00';
  end
else
begin
  cc := (AA * 0.0095)+1.4;
  dd := (((Ln(cc))/(Ln(10))) * 20 );
  lbl52.Caption := Copy(FloatToStr(cc),1,5);
  lbl59.Caption := Copy(FloatToStr(dd),1,5);
end ;

  if AA < BB then
  begin
    lbl29.Caption := 'NORMAL';
    lbl20.Font.Color := clBlue;
    spn15.Color := clBlue;
  end;
  if AA >= BB then
  begin
    lbl29.Caption := 'OVERLOADE';
    lbl20.Font.Color := clMaroon;
    spn15.Color := clMaroon;
  end;
  sprgrsbr7.Position:=StrToInt(Trim(lbl31.Caption));
end;

if pos('d6',data)=1then
begin
  lbl34.Caption:=Copy(data,3,4);
  AA := StrToInt(lbl34.Caption);
  BB := StrToInt(cbb6.Text);
  if AA = 0 then
  begin
    lbl53.Caption := '0.00';
    lbl60.Caption := '0.00';
  end
else
begin
  cc := (AA * 0.0095)+1.4;
  dd := (((Ln(cc))/(Ln(10))) * 20 );
  lbl53.Caption := Copy(FloatToStr(cc),1,5);
  lbl60.Caption := Copy(FloatToStr(dd),1,5);
end ;

  if AA < BB then
  begin
    lbl32.Caption := 'NORMAL';
    lbl20.Font.Color := clBlue;
    spn16.Color := clBlue;
  end;
  if AA >= BB then
  begin
    lbl32.Caption := 'OVERLOADE';
    lbl20.Font.Color := clMaroon;
    spn16.Color := clMaroon;
  end;
  sprgrsbr8.Position:=StrToInt(Trim(lbl34.Caption));
end;

if pos('d7',data)=1then
begin
  lbl37.Caption:=Copy(data,3,4);
  AA := StrToInt(lbl37.Caption);
  BB := StrToInt(cbb7.Text);
```

```
        if AA = 0 then
        begin
        lbl54.Caption := '0.00';
        lbl61.Caption := '0.00';
        end
    else
    begin
    cc := (AA * 0.0095)+1.4;
    dd := (((Ln(cc))/(Ln(10)))* 20 );
    lbl54.Caption := Copy(FloatToStr(cc),1,5);
    lbl61.Caption := Copy(FloatToStr(dd),1,5);
    end ;

        if AA < BB then
        begin
        lbl35.Caption := 'NORMAL';
        lbl20.Font.Color := clBlue;
        spnl8.Color := clBlue;
        end;
        if AA >= BB then
        begin
        lbl35.Caption := 'OVERLOADE';
        lbl20.Font.Color := clMaroon;
        spnl8.Color := clMaroon;
        end;
        sprgrsbr9.Position:=StrToInt(Trim(lbl37.Caption));
end;

if pos('d8',data)=1then
begin
    simpan := simpan + 1;
    lbl40.Caption:=Copy(data,3,4);
    AA := StrToInt(lbl40.Caption);
    BB := StrToInt(cbb8.Text);
    if AA = 0 then
    begin
    lbl55.Caption := '0.00';
    lbl62.Caption := '0.00';
    end
    else
    begin
    cc := (AA * 0.0095)+1.4;
    dd := (((Ln(cc))/(Ln(10)))* 20 );
    lbl55.Caption := Copy(FloatToStr(cc),1,5);
    lbl62.Caption := Copy(FloatToStr(dd),1,5);
    end ;

        if AA < BB then
        begin
        lbl38.Caption := 'NORMAL';
        lbl20.Font.Color := clBlue;
        spnl7.Color := clBlue;
        end;
        if AA >= BB then
        begin
        lbl38.Caption := 'OVERLOADE';
        lbl20.Font.Color := clMaroon;
        spnl7.Color := clMaroon;
        end;

if simpan <= 400 then
begin
    sprgrsbr10.Position:=StrToInt(Trim(lbl40.Caption));
    mainform.Tbl1.Open;
    mainform.Tbl1.Append;
    mainform.wdstrngfldtbl1Tgl.Text := edt3.Text;
    mainform.wdstrngfldtbl1Jam.Text := lbl44.Caption;

    mainform.wdstrngfldtbl1Data1.Text := lbl47.Caption;
    mainform.wdstrngfldtbl1Data2.Text := lbl56.Caption;
    mainform.wdstrngfldtbl1Data3.Text := lbl57.Caption;
    mainform.wdstrngfldtbl1Data4.Text := lbl58.Caption;
    mainform.wdstrngfldtbl1Data5.Text := lbl59.Caption;
```

```
        mainform.wdstrngfldtbl1Data6.Text := lbl60.Caption;
        mainform.wdstrngfldtbl1Data7.Text := lbl61.Caption;
        mainform.wdstrngfldtbl1Data8.Text := lbl62.Caption;
    tbl1.Post ;
end ;

end;

end
else
    pesan := pesan + Tmp[I];
    edt1.Text := Pesan;
    Edt10.Text := mmol.Text;
    mmol.Text := '';
    end;
end;

procedure TMainForm.tmr1Timer(Sender: TObject);
var
    Jam, Menit, Detik, MiliDetik : Word;
    tanggal, bulan, tahun : Word;
//    JAMI, MENITI          : INTEGER;
begin
    DecodeTime(Time, Jam , Menit, Detik, MiliDetik);
    lbl44.Caption := IntToStr(Jam) + ':' +
                    IntToStr(Menit) + ':' +
                    IntToStr(Detik);

    DecodeDate(date, tahun, bulan, tanggal);
    edt3.Text := inttostr(tanggal) + '-' + inttostr(bulan) + '-' + inttostr(tahun);
    //sLABEL12.Caption := inttostr(tanggal) + '-' + inttostr(bulan) + '-' +
    inttostr(tahun);
    end;
end.
```

```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, sComboBox, Buttons, sSpeedButton, sLabel, ExtCtrls,  
sPanel;
```

```
type
```

```
TForm1 = class(TForm)  
    spnl11: TsPanel;  
    lbl42: TsLabel;  
    lbl43: TsLabel;  
    btn2: TsSpeedButton;  
    cbb10: TsComboBox;  
    procedure btn2Click(Sender: TObject);  
private  
    { Private declarations }  
public  
    { Public declarations }  
end;
```

```
var
```

```
Form1: TForm1;
```

```
implementation
```

```
uses MainUnit;
```

```
{$R *.dfm}
```

```
procedure TForm1.btn2Click(Sender: TObject);
```

```
begin
```

```
MainForm.vcm1.DeviceName := cbb10.Text;
```

```
form1.Close;
```

```
end;
```

```
end.
```

```
unit Unit2;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, Grids, DBGrids, ExtCtrls, sPanel;
```

```
type
```

```
TForm2 = class(TForm)  
    spnl10: TsPanel;  
    img8: TImage;  
    dbgrd1: TDBGrid;  
    procedure img8Click(Sender: TObject);  
private  
    { Private declarations }  
public  
    { Public declarations }  
end;
```

```
var
```

```
Form2: TForm2;
```

```
implementation
```

```
uses MainUnit;  
{$R *.dfm}
```

```
procedure TForm2.img8Click(Sender: TObject);  
begin  
Close;  
end;  
  
end.
```



NATIONAL INSTITUTE OF TECHNOLOGY
Smart and Intelligent
EST. 1969

ITN MALANG 2013

LEMBAR PERSEMBAHAN

Alhamdulillah kupersembahkan kepada ALLAH SWT atas segala rahmat dan kesempatan untuk menyelesaikan tugas akhir dengan segala kekuranganku. Segala syukur aku ucapkan kepadaMu karena telah menghadirkan mereka yang selalu memberi semangat dan doa disaat kutertatih. KarenaMu lah mereka ada, dan karenaMu lah tugas akhir ini terselesaikan. Hanya padaMu tempat kumengadu dan mengucapkan syukur.

Kepada Abah dan Umi (H. Maskur Suparto dan Hj. Sri Haryati) tersayang, tugas akhir ini aku persembahkan. Tiada kata yang bisa menggantikan segala sayang, usaha, semangat, dan juga uang yang telah dicurahkan untuk penyelesaian tugas akhir ini. Teruntuk adik Ummu Latifah yang sedikit nakal dan malas belajar, tugas akhir ini kaka persembahkan untuk jadi motivasi dan pengingat semangatmu. Luluslah lebih cepat dan lebih baik dari kakamu ini, Um. (walaupun kaka agak meragukan itu ☺)

Kepada Pak M. Ibrahim dan Pak Michael yang telah membimbing saya selama penyelesaian tugas akhir ini. Saya ucapkan terima kasih atas ilmu, nasihat, cerita yang telah Bapak beri pada saya. Terima kasih atas kesabaran Bapak selama masa bimbingan saya walaupun saya banyak kekurangan dan kelalaian ☺

Segelas coklat untuk kalian para sahabat yang selalu menghangatkan hidupku dan memberikan semburat warna di tugas akhir ini. Kalian yang selalu ada (walaupun tidak sampai akhir) di setiap langkahku ☺ Tanpa kalian tugas akhir ini akan seratus kali lebih berat untuk diselesaikan. Terima kasih yang takkan pernah habis untuk kalian yang telah menciptakan sebuah cerita denganku di kota ini. Cerita tentang yang ada di setiap sudut kampus dan Malang akan selalu terkenang selamanya. Kalian adalah keluarga dan rumah kedua bagiku ☺

Untuk kamu yang selalu ada di setiap cerita. Untuk kamu yang selalu menjadi alasan aku tersenyum dan terus melangkah walaupun goyah untuk terus berusaha dan pantang menyerah. Teruntuk kamu yang selalu mendampingi di kala susah dan senang. Menjadi seseorang yang selalu mau mengerti aku walaupun terkadang kita suka berantem ga jelas Teruntuk kamu yang selalu berbagi cerita dan tawamu. Aku berterima kasih kepadanya karena mengizinkanmu untuk menemaniku ☺

WASSALAM.....!!!!

IMAM BASHORI

BIOGRAFI PENULIS



Penulis lahir di kabupaten Lumajang pada tanggal 22 juli 1990 dari ayah H. Maskur Suparto dan Ibu Hj. Sri Haryati. Penulis merupakan anak pertama dari dua bersaudara. Penulis memulai pendidikan pada tahun 1997 di SD Negeri Sawaran Lor kota Klakah, Lumajang. Pertengahan 2003 penulis menempuh pendidikan di SMPN 1 Klakah sampai tahun 2006. Kemudian melanjutkan pendidikan di MA-SMK Darut Taqwa, Sengonagung, Purwosari – Pasuruan bidang studi Jurusan

Bahasa mulai tahun 2006 dan lulus tahun 2009, kemudian pada tahun yang sama penulis melanjutkan studi di perguruan tinggi Institut Teknologi Nasional Malang. Penulis memilih program Studi Teknik Elektro S-1, Konsentrasi Teknik Komputer, Fakultas Teknologi Industri dan di wisuda tanggal 28 September 2013, dengan judul Skripsi “*Perancangan dan Pembuatan User Interface Remote Monitoring Tata Suara Panggung*”