

**PEMBUATAN USER INTERFACE ALAT PELATIHAN PLC
DENGAN STUDI KASUS WATER TREATMENT PLANT**

SKRIPSI



**Disusun Oleh :
ADI SETYAWAN
09.12.522**

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

LEMBAR PERSETUJUAN

**PEMBUATAN USER INTERFACE ALAT PELATIHAN PLC
DENGAN STUDI KASUS WATER TREATMENT PLANT**

SKRIPSI

*Disusun dan diajukan untuk melengkapi dan memenuhi persyaratan guna mencapai
gelar Sarjana Teknik*

Disusun Oleh :

ADI SETYAWAN

09.12.522

Mengetahui,

Ketua Program Studi Teknik Elektro S-1

M. Ibrahim Ashari, ST, MT

NIP.Y.1030100358

Diperiksa dan Disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Eng. Arruanto Soetedjo, ST, MT
NIP.P. 1030800417

Yuli Wahyuni, ST, MT
NIP.P. 1031200456

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

2013

SURAT PERNYATAAN ORISINALITAS

Yang bertandatangan di bawah ini :

Nama : Adi Setyawan
NIM : 09.12.522
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, 26-September-2012

Yang membuat pernyataan,

METERAI
TEMPEL
A41DFAAF00C232781
6000 DJP

Adi Setyawan
NIM :0912522



PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

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 : ADI SETYAWAN
NIM : 09.12.522
JURUSAN : Teknik Elektro S-1
KONSENTRASI : Teknik Komputer
MASA BIMBINGAN: SEMESTER GENAP 2012/2013
JUDUL : PEMBUATAN USER INTERFACE ALAT PELATIHAN PLC
DENGAN STUDI KASUS WATER TREATMENT PLANT

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

Hari : Jumat
Tanggal : 16 Agustus 2013
Dengan Nilai : 83 (A) *af*

PANITIA UJIAN SKRIPSI

Ketua Majelis Penguji

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

Sekretaris Majelis Penguji

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

ANGGOTA PENGUJI

Dosen Penguji I

Irmalia Suryani Faradisa, ST, MT
NIP.P.1030000365

Dosen Penguji II

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

Ucapan Terima Kasih Untuk:

Bapak dan Ibu : Bapak Siswanto Dan Ibu Haryati

Saudaraku :Eryanti Wahyuningtias, Silvia Margayanti

Guru Pemrograman :Mas Handi, Mas Juni, Erik Cosmas,

Afif, Muhamad Rifki

ABSTRAK

PEMBUATAN USER INTERFACE ALAT PELATIHAN PLC DENGAN STUDI KASUS ATER TREATMENT PLANT

AdiSetyawan (09.12.522)

Dosen Pembimbing I : Dr. Eng. Aryuanto Soetedjo, ST. MT

Dosen Pembimbing II : Yuli Wahyuni, ST. MT

Jurusan Teknik Elektro, Konsentrasi Teknik Komputer, Fakultas Teknik Industri,
Institut Teknologi Nasional Malang, Kampus II, Singosari-Malang, ,telp. (0341) 417636

adhi.ezh90@gmail.com

ABSTRAK

Air merupakan salah satu kebutuhan pokok manusia. Proses produksi air bersih (Water Treatment Plant)) sangat menarik di jadikan sebagai media pembelajaran dengan membuat prototipe atau miniatur plant yang menggunakan modul kontrol berbasis PLC. Akan tetapi didalam pembuatan miniatur atau prototipe plant tersebut terkendala dengan biaya yang cukup besar serta waktu pembuatan yang lama. Cara untuk mengatasi permasalahan tersebut adalah dengan dibuatnya user interface sebagai pengganti prototipe plant untuk mensimulasikan proses produksi air bersih yang akan divisualisasikan di komputer, aplikasi user interface ini diharapkan dapat digunakan untuk modul kontrol berbasis PLC.

Dalam pembuatan aplikasi user interface ini menggunakan bahasa pemrograman delphi 7.0 karena dinilai tepat untuk dihubungkan dengan modul kontrol berbasis PLC, selain menggunakan bahasa pemrograman delphi pembuatan aplikasi user interface juga menggunakan gambar gif untuk membuat tampilan proses- proses pengolahan air sehingga aplikasi ini terlihat seperti nyata.

Hasil dari pengujian aplikasi user interface modul kontrol berbasis PLC dinilai dapat digunakan untuk modul kontrol berbasis PLC, serta dapat mensimulasikan proses pengolahan air bersih. Rekayasa sensor yang digunakan aplikasi user interface dapat diterima dengan baik oleh PLC sehingga respon yang di kirim oleh PLC tepat. Aplikasi user interface membutuhkan banyak gambar untuk mensimulasikan tiap tahapan proses produksi air bersih.

Kata kunci :User Interface, delphi 7.0, simulasi, pengolahan air

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Kuasa atas segala limpahan berkat dan rahmat-Nya sehingga penelitian berjudul Pembuatan User Interface Alat Pelatihan PLC Dengan Studi Kasus Water Treatment Plant dapat terselesaikan.

Penulisan ini dibuat untuk memenuhi salah satu syarat dalam memperoleh gelar sarjana teknik. Ucapan terima kasih yang sebesar-besarnya kami sampaikan kepada:

1. Bapak Ir. Soeparno Djiwo, MT. selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Anang Subardi, MT. selaku Dekan Fakultas Teknologi Industri ITN Malang.
3. Bapak M. Ibrahim Ashari, ST, MT. selaku Ketua Jurusan Teknik Elektro ITN Malang.
4. Bapak Dr.Eng. Aryunto Soetedjo, ST, MT. Dan Ibu Yuli Wahyuni, ST, MT selaku dosen pembimbing.
5. Bapak Sotyohadi, ST. kepala laboratorium pemrograman komputer dan multimedia ITN Malang.
6. Teman-teman asisten laboratorium pemrograman komputer dan multimedia ITN Malang.
7. Semua pihak yang membantu dalam penyusunan penelitian 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, Juli 2012

Penulis

DAFTAR ISI

Lembar Persetujuan.....	ii
Surat Pernyataan	iii
Abstrak	iv
Kata Pengantar.....	v
Daftar Isi.....	vi
Daftar Gambar	viii
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 Metodologi Penelitian	3
1.6 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI	
2.1 Borland Delphi 7.0.....	5
2.1.1 IDE Delphi	6
2.1.2 Menu Delphi.....	8
2.1.3 Komponent Standart Delphi 7.0.....	10
2.1.4 Komponen Tambahan Delphi 7.0.....	10
2.1.5 Tipe Data.....	11
2.1.6 Operator.....	14
2.2 Gambar Gif.....	13
2.2.1 Mengenal Gif.....	14
2.2.2 Struktur File.....	16
2.2.3 Kompresi Gif.....	16
2.2.4 Versi.....	17
2.3 Proses Pengolahan Air Bersih Perusahaan Daerah Air Minum (PDAM).....	17
2.4 Pengendalian Sistem	20

	2.5 Modul Interface.....	22
BAB III	PERANCANGAN DAN DESAIN APLIKASI	
	3.1 Deskripsi Umum	24
	3.2 Tahap Identifikasi	24
	3.3 Tahap Konseptualisasi.....	24
	3.3.1 User Interface Untuk mensimulasikan Water Treatment Plant	25
	3.4 Penentuan Komponen Dasar.....	26
	3.5 Diagram Blok.....	26
	3.6 Diagram Alir.....	27
	3.7 Pengalamatan	29
	3.7.1 Pengalamatan Inputan Dari PLC 1 (Master).....	29
	3.7.2 Pengalamatn Inputan Dari PLC 2 (Slave).....	29
	3.7.3 Pengalamatan Mikrokontroler (Modul Interface)	29
	3.8 Perancangan Dan Desain Aplikasi.....	30
	3.8.1 Perancangan Gambar Gif.....	30
	3.8.2 Desain Aplikasi	51
	3.8.3 Form Simulasi	53
BAB IV	IMPLEMENTASI DAN PENGUJIAN	
	4.1 Spesifikasi Program	54
	4.2 Pengujian	54
	4.2.1 Form Simulasi	54
	4.2.2 Pengaturan Koneksi Comport	55
	4.2.3 Form Simulasi Setelah Koneksi Terhubung.....	55
	4.2.4 Pengujian Penerimaan Data Dari Interface.....	56
	4.2.5 Pengujian Sistem keseluruhan.....	69
BAB V	PENUTUP	
	5.1 Kesimpulan	75
	5.2 Saran.....	75
	Daftar Pustaka.....	77
	Lampiran	

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.....	7
Gambar 2.5	: Water Treatment Plant.....	18
Gambar 2.6	: Blok Diagram Modul Input Interface	22
Gambar 2.7	: Blok Diagram Modul Output Interface	23
Gambar 3.1	: Diagram Blok User Interface Modul Kontrol Berbasis PLC.....	27
Gambar 3.2	: Diagram Alir (Flowchart)	28
Gambar 3.3	: Perancangan Bak Prasedimentasi.....	30
Gambar 3.4	: Membuat Frame Animasi Bak Prasedimentasi.....	31
Gambar 3.5	: 19 Frame Animasi	31
Gambar 3.6	: Proses Pembuatan Animasi Bak Prasedimentasi	32
Gambar 3.7	: Penentuan Delay Time.....	32
Gambar 3.8	: Penentuan Looping Delay Time.....	33
Gambar 3.9	: Jendela Penyimpanan Gambar Gif.....	33
Gambar 3.10	: Perancangan Bak Koagulasi	34
Gambar 3.11	: Membuat Frame Animasi Bak Koagulasi.....	34
Gambar 3.12	: 18 Frame Animasi	35
Gambar 3.13	: Proses Pembuatan Animasi Bak Koagulasi	35
Gambar 3.14	: Penentuan Delay Time.....	36
Gambar 3.15	: Penentuan Looping Delay Time.....	36
Gambar 3.16	: Jendela Penyimpanan Gambar Gif.....	37
Gambar 3.17	: Perancangan Bak Flokulasi.....	37
Gambar 3.18	: Membuat Frame Animasi Bak Flokulasi.....	38
Gambar 3.19	: 32 Frame Animasi	38
Gambar 3.20	: Proses Pembuatan Animasi Bak Flokulasi	39
Gambar 3.21	: Penentuan Delay Time.....	39
Gambar 3.22	: Penentuan Looping Delay Time.....	40
Gambar 3.23	: Jendela Penyimpanan Gambar Gif.....	40
Gambar 3.24	: Perancangan Bak Sedimentasi	41

Gambar 4.13	: Penerimaan Data Indikator Level Air <i>Low</i>	60
Gambar 4.14	: Pengiriman Data Indikator Level Air <i>Medium</i>	61
Gambar 4.15	: Penerimaan Data Indikator Level Air <i>Medium</i>	61
Gambar 4.16	: Pengiriman Data Indikator Level Air <i>High</i>	62
Gambar 4.17	: Penerimaan Data Indikator Level Air <i>High</i>	62
Gambar 4.18	: Pengiriman Data Motor Koagulasi Cepat	63
Gambar 4.19	: Penerimaan Data Motor Koagulasi Cepat.....	63
Gambar 4.20	: Pengiriman Data Motor Koagulasi Lambat	64
Gambar 4.21	: Penerimaan Data Motor Koagulasi Lambat	64
Gambar 4.22	: Pengiriman Data Pompa Vakum	65
Gambar 4.23	: Penerimaan Data Pompa Vakum.....	65
Gambar 4.24	: Pengiriman Data Pompa Distribusi	66
Gambar 4.25	: Penerimaan Data Pompa Distribusi.....	66
Gambar 4.26	: Pengiriman Data Indikator Produksi	67
Gambar 4.27	: Penerimaan Data Indikator Produksi.....	67
Gambar 4.28	:(a) Kondisi Push Button ON Sistem	69
Gambar 4.28	:(b) Kondisi Push Button ON Sistem	69
Gambar 4.29	: Kondisi Air Di Level Rendah (<i>Low</i>)	70
Gambar 4.30	: Pompa Intake Menyala Bergantian	71
Gambar 4.31	: Sensor Koagulasi Menyala.....	72
Gambar 4.32	: Sensor Sedimentasi	72
Gambar 4.33	: Kondisi Air Berada Di Level <i>Medium</i>	73
Gambar 4.34	: Kondisi Air Berada Di Level <i>High</i>	74

Gambar 3.25 : Membuat Frame Animasi Bak Sedimentasi	41
Gambar 3.26 : 16 Frame Animasi	42
Gambar 3.27 : Proses Pembuatan Animasi Bak Sedimentasi.....	42
Gambar 3.28 : Penentuan Delay Time.....	43
Gambar 3.29 : Penentuan Looping Delay Time.....	43
Gambar 3.30 : Jendela Penyimpanan Gambar Gif.....	44
Gambar 3.31 : Perancangan Bak Filter	44
Gambar 3.32 : Membuat Frame Animasi Bak Filter.....	45
Gambar 3.33 : 27 Frame Animasi	45
Gambar 3.34 : Proses Pembuatan Animasi Bak Filter	46
Gambar 3.35 : Penentuan Delay Time.....	46
Gambar 3.36 : Penentuan Looping Delay Time.....	47
Gambar 3.37 : Jendela Penyimpanan Gambar Gif.....	47
Gambar 3.38 : Perancangan Bak Disinfeksi	48
Gambar 3.39 : Membuat Frame Animasi Bak Disinfeksi	48
Gambar 3.40 : 33 Frame Animasi	49
Gambar 3.41 : Proses Pembuatan Animasi Bak Disinfeksi.....	49
Gambar 3.42 : Penentuan Delay Time.....	50
Gambar 3.43 : Penentuan Looping Delay Time.....	50
Gambar 3.44 : Jendela Penyimpanan Gambar Gif.....	51
Gambar 3.45 : Perancangan Form Simulasi	53
Gambar 4.1 : From Simulasi Pertama Kali Dijalankan	54
Gambar 4.2 : Pengaturan Koneksi Comport	55
Gambar 4.3 : Form Simulasi Setelah Terhubung dengan Interface	55
Gambar 4.4 : Pengiriman Data Indikator Sistem	56
Gambar 4.5 : Penerimaan Data Indikator Sistem	56
Gambar 4.6 : Pengiriman Data Pompa 1	57
Gambar 4.7 : Penerimaan Data Pompa 1	57
Gambar 4.8 : Pengiriman Data Pompa 2	58
Gambar 4.9 : Penerimaan Data Pompa 2	58
Gambar 4.10 : Pengiriman Data Pompa 3	59
Gambar 4.11 : Penerimaan Data Pompa 3	59
Gambar 4.12 : Pengiriman Data Indikator Level Air <i>Low</i>	60

DAFTAR TABEL

Tabel 2.1	: Komponen Standar Delphi 7.0.....	10
Tabel 2.2	: Tipe Data Text String	11
Tabel 2.3	: Tipe Data Bilangan Integer.....	12
Tabel 2.4	: Tipe Data Bilangan Real.....	12
Tabel 2.5	: Tipe Data Bilangan Boolean.....	13
Tabel 2.6	: Operator Aritmatik	14
Tabel 2.7	: Operator Pembandingan	15
Tabel 2.8	: Maximum Hardware Configuration	21
Tabel 3.1	: Pengalamatan Inputan dari PLC 1 (Master)	29
Tabel 3.2	: Pengalamatan Inputan dari PLC 1 (Slave).....	29
Tabel 3.3	: Pengalamatan Dari PLC ke Mikrokontroler	29
Tabel 4.1	: Pengambilan Data Masukan Dari Modul Interface ke User Interface	68
Tabel 4.2	: Pengambilan Data Keluaran Modul Interface Dari User Interface	68



BAB I

PENDAHULUAN

1.1. Latar Belakang

Sciring berkembangannya populasi penduduk di Indonesia, kebutuhan air bersih akan semakin meningkat. Di setiap daerah terdapat perusahaan yang dapat mensuplai air bersih, perusahaan tersebut yaitu Perusahaan Daerah Air Minum atau yang sering kita dengar dengan nama PDAM. Secara umum sistem pengolahan air untuk menghasilkan air bersih pada Perusahaan Daerah Air Minum (PDAM) terdiri dari serangkaian proses. Pertama-tama air baku yang diterima ditampung pada water intake, kemudian dialirkan melalui pompa ke koagulasi. Dari koagulasi kemudian dialirkan ke flokulator dan sedimentation room. Selanjutnya air tersebut disaring melalui filter dan ditampung pada reservoir kemudian dipompa untuk didistribusikan ke pelanggan.

Rangkaian proses di atas dapat diaplikasikan sebagai miniatur atau prototipe plant untuk alat pelatihan PLC. Tetapi didalam pembuatan miniatur atau prototipe plant tersebut, terkendala dengan biaya yang cukup besar dan waktu pembuatan yang cukup lama. Salah satu cara untuk mengatasi permasalahan tersebut adalah dengan dibuatnya user interface untuk mensimulasikan proses produksi air bersih (Water Treatment Plant) yang akan di virtualisasikan di komputer. Dengan adanya user interface ini maka tidak perlu dibuat miniatur atau prototipe plant.

Berdasarkan latar belakang tersebut, maka peneliti mengemukakan sebuah ide "Pembuatan User Interface untuk Alat Pelatihan PLC Dengan Studi Kasus Water Treatment Plant" dengan menggunakan bahasa pemrograman delphi.

1.2. Rumusan Masalah

Bagaimana implementasi pemrograman Delphi dalam membuat user interface untuk alat pelatihan PLC dengan studi kasus Water Treatment Plant ?

1.3. Tujuan

Berdasarkan permasalahan yang dikemukakan diatas maka, tujuan dalam penulisan skripsi ini yaitu membuat user interface untuk alat pelatihan PLC dengan studi kasus Water Treatment Plant.

1.4. Batasan Masalah

Untuk mencapai tujuan maka penulis mengambil batasan masalah sebagai berikut:

1. Aplikasi yang dibuat dititik beratkan pada virtualisasi Water Treatment Plant.
 2. Tidak membahas perhitungan debit air dan kapasitas pompa pada Water Treatment Plant.
 3. Tidak membahas mekanisme pencampuran zat kimia yang dibutuhkan dalam proses produksi air bersih.
 4. Aplikasi user interface membutuhkan 23 I/O meliputi 11 input dan 12 output.
 5. Aplikasi user interface menggunakan dua buah PLC (PLC Twido tipe TWDLMDA20DTK) masing-masing PLC mempunyai 20 I/O.
 6. Untuk komunikasi dengan hardware menggunakan interface yang sudah tersedia
-

1.5. Metodologi Penelitian

Metode yang digunakan dalam penyusunan skripsi ini adalah :

1. Kajian literatur

Pengumpulan data dan informasi dilakukan dengan mencari bahan-bahan kepustakaan dan referensi dari berbagai sumber sebagai landasan teori yang ada hubungannya dengan permasalahan pada perancangan aplikasi.

2. Analisa kebutuhan sistem

Data dan informasi yang telah diperoleh akan dianalisa agar didapatkan kategori-kategori yang harus ditampilkan pada aplikasi yang akan dibuat.

3. Perancangan dan implementasi

Bersarkan data dan informasi yang diperoleh serta analisa kebutuhan aplikasi, akan dijadikan acuan dalam merancang kerangka secara global yang menggambarkan mekanisme dari aplikasi yang akan dibuat.

4. Eksperimen dan evaluasi

Setelah melalui beberapa tahap mulai dari pengumpulan data, pada tahap ini aplikasi yang telah selesai dibuat akan diuji coba, yaitu pengujian secara fungsional sistem dan akan dilakukan koreksi serta penyempurnaan aplikasi apabila diperlukan.

5. Kesimpulan.

1.6. Sistematik Penulisan

Untuk mempermudah dan memahami pembahasan penulisan karya skripsi ini, sistematika penulisan disusun sebagai berikut:

BAB I : PENDAHULUAN

Berisikan latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian, dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Membahas tentang dasar teori mengenai permasalahan yang berhubungan dengan penelitian.

BAB III : PERANCANGAN DAN PENGUJIAN PROGRAM

Bab ini berisi tentang kebutuhan media yang digunakan dalam membangun program dan menjelaskan kerangka program secara global.

BAB IV : PEMBUATAN DAN PENGUJIAN

Berisi tentang pembahasan mengenai langkah-langkah pembuatan program serta pengujian terhadap program.

BAB V : PENUTUP

Bab ini berisi ungkapan kembali pokok persoalan beserta hasilnya secara singkat serta keinginan penulis untuk menyampaikan suatu gagasan yang belum tercapai dalam tujuan penelitian demi perbaikan.

DAFTAR PUSTAKA



BAB II

TINJAUAN PUSTAKA

Dalam penyusunan laporan skripsi ini diperlukan landasan teori yang memiliki relevansi dengan masalah yang dibahas. Landasan teori ini untuk memberikan arah, persepsi dan landasan untuk menentukan solusi terhadap permasalahan yang sedang dibahas. Landasan teori tersebut diperoleh dengan membahas beberapa literatur yang mempublikasikan pendapat beberapa ilmuwan yang dipakai sebagai penunjang pembahasan masalah.

2.1. Delphi

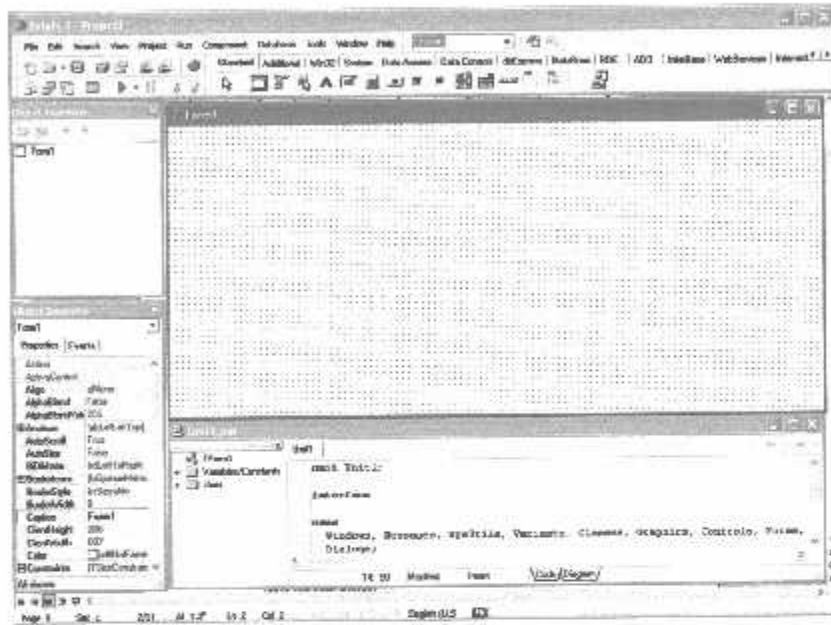
Borland Delphi 7.0 adalah suatu software yang digunakan untuk membuat aplikasi yang berbasis visual. Di dalam software ini, listing program menggunakan bahasa pascal tetapi sudah dipadu dengan bahasa pemrograman yang terstruktur sehingga *programmer* dipermudah dalam pembuatan aplikasi, berbeda dengan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

2.1.1. 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 palette* yaitu sebagai tempat terdapatnya komponen

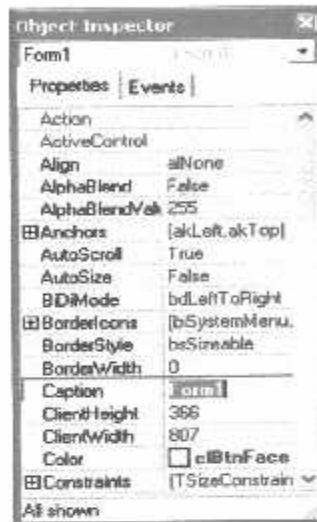
2. Object Treeview



Gambar 2.2
Object Treeview

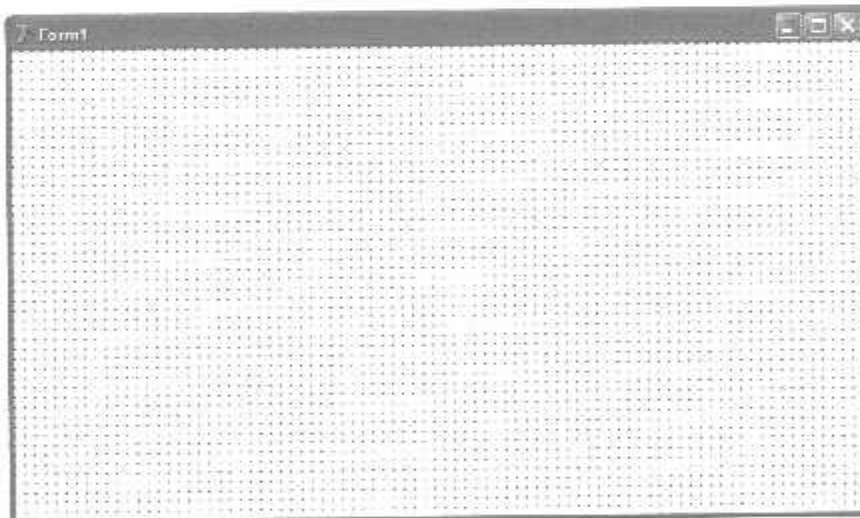
Fasilitas *object treeview* dapat ditemui mulai Borland Delphi 6.0, jadi pada versi sebelumnya tidak ditemui fasilitas ini. Fasilitas ini digunakan untuk menampilkan daftar komponen yang sedang digunakan dalam membuat aplikasi.

3. Object Inspector



Gambar 2.3
Object Inspector

4. Form Designer



Gambar 2.4
Form Designer

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

2.1.2. Menu Delphi

1. Menu File

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

2. Menu Edit

Menu edit yang terdapat pada Delphi digunakan untuk mengubah segala sesuatu yang berhubungan tentang 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 pallette*, *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

Di dalam menu ini disediakan beberapa fitur, misalnya adalah untuk menjalankan aplikasi dan juga untuk menghentikan aplikasi, juga terdapat fitur untuk *reset* aplikasi ketika ada *error* ketika 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 pallette*. 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 Data Base

Menu Database digunakan untuk membuat, mengubah atau melihat database.

9. Menu Tools

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

10. Menu Window

Menu ini digunakan untuk mengarahkan kursor pada posisi yang dikhendaki.

11. Menu Help

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

2.1.3. 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	A	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 pilihan, tetapi hanya satu pilhan yang bisa digunakan
11		List Box	Menampilkan 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

2.1.4. Komponen Tambahan Delphi 7.0

Dalam pembuatan aplikasi ini, selain menggunakan komponen standart yang disediakan di delphi 7.0 juga menggunakan komponen tambahan yang belum tersedia. Komponen tersebut diantaranya adalah.

- Comport

Biasanya untuk mengirim maupun menerima data secara serial kita memakai aplikasi bawaan Windows yaitu Hyperterminal. Terbatas memang karena hanya mengolah text. Untuk dapat berkreasi lebih, maka kita membutuhkan aplikasi semacam delphi. Tetapi di delphi belum ada komponen yang menangani komunikasi serial. Kita harus menambahkannya. Salah satu komponen yang dapat digunakan adalah ComPort. Sebelum kita menggunakan ComPort, install terlebih dahulu ComPort di Delphi 7.

2.1.5. Tipe Data

Pada dasarnya, semua data yang tersimpan pada memori komputer harus mempunyai tipe data. Pengertian tipe data di sini adalah suatu aturan pada Delphi yang digunakan untuk menentukan suatu jenis variabel. 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 variabel yang mempunyai nilai bilangan bulat.

Perhatikan tabel di bawah ini:

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 x E+63 s/d 2 x E- 63-1
Byte	0 s/d 255
Word	0 s/d 65535

b. Real

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

Tabel 2.4
Tipe Data Bilangan Real

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

c. Boeelan

Tipe data boolean adalah tipe data yang mempunyai nilai true dan false. Perhatikan tabel di bawah ini:

Tabel 2.5
Tipe Data Bilangan Boolean

TIBE	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

2.1.6. Operator

Operator adalah bentuk penugasan pada sebuah ekspresi program, sebagai contoh adalah "5+6", maka yang disebut operator adalah (+). Di bawah ini adalah beberapa 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. Di bawah 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	sisa hasil bagi	$7 \text{ mod } 2 = 1$
Div	pembagian bilangan bulat	$7 \text{ div } 2 = 3$

2. Operator Perbandingan

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

Tabel 2.7
Operator Perbandingan

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

3. Operator Logikal

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

2.2. Gambar Gif

Graphics Interchange Format (GIF) merupakan format grafis yang paling sering digunakan untuk keperluan desain website. GIF memiliki kombinasi warna lebih sedikit dibanding JPEG, namun mampu menyimpan grafis dengan latar belakang (*background*) transparan ataupun dalam bentuk animasi sederhana.

2.2.1 Mengenal GIF

GIF adalah format gambar asli yang dikompres dengan CompuServe. Bitmap jenis ini mendukung 256 warna dan bitmap ini juga sangat populer dalam internet. Format GIF hanya dapat menyimpan gambar dalam 8 bit dan hanya mampu digunakan mode *grayscale*, *bitmap*, dan *index color*. Format ini merupakan format standar Internet atau publikasi

elektronik. Selain itu format ini mendukung penggunaan *multiple bitmap* dalam satu file sehingga dapat menghasilkan gambar animasi dan merekam penggunaan *Transparency Masking*. GIF sering digunakan untuk mewarnai halaman HTML pada tag `<body background="back.gif">`, spanduk (banner), icon, dan lain-lain. GIF juga baik untuk menampilkan gambar dengan komposisi yang tidak menggunakan terlalu banyak warna, seperti gambar kartun. Sementara itu, GIF tidak cocok digunakan untuk menampilkan foto. GIF menggunakan metode *Lossless Compression*, untuk membuat ukuran file sekecil mungkin. *Lossless Compression* adalah kompresi yang tidak mengurangi kualitas pada gambar, namun dapat memperkecil besarnya jumlah file, jadi tidak ada penghilangan data pada saat dilakukan kompresi.

2.2.2 Struktur File

- Bagian kepala (Header): Menyimpan informasi identitas file GIF (3 bytes, harus string "GIF") dan versinya (3 bytes, harus string "87a" atau "89a")
- *Global Screen Descriptor*: Mendefinisikan logikal screen area di mana masing-masing file GIF ditampilkan.
- *Global Color Table*: Masing-masing gambar dalam GIF dapat menggunakan *global color table* atau tabel warnanya sendiri-sendiri. Penggunaan GCT akan memperkecil ukuran file GIF.
- gambar1, gambar2, gambar3, ... gambar-n: di mana masing-masing gambar memiliki struktur blok sendiri-sendiri dan terminator antar file.
- *Trailer*: Akhir dari sebuah file GIF.

2.2.3 Kompresi Gif

Format GIF menggunakan kompresi algoritma dari LZW (Lempel Zev Welch) yang dimiliki oleh Unisys. Pemegang hak cipta GIF kini dipegang oleh CompuServe Incorporated. Awalnya GIF adalah format yang bebas royalti bagi semua pengguna namun tahun 1995. Unisys memutuskan menarik royalti pada vendor pengguna GIF.

2.2.4 Versi

1. Versi Pertama

Versi awal GIF adalah GIF87a dirilis pada tahun 1987 yang memiliki fasilitas :

- 1) Kompresi LZW
- 2) Dapat memasukkan gambar lebih dari satu dalam satu file dan menampilkan secara bergantian, animasi.
- 3) Mengatur posisi gambar dalam latar GIF.
- 4) Interlacing

Interlacing adalah sebuah teknik menampilkan gambar secara gradual.

2. Versi Kedua

Versi Kedua GIF yaitu GIF GIF98a, penambahan fasilitas pada GIF89a ini adalah :

- 5) Dapat mengatur waktu dari tiap gambar.
- 6) mengatur masukkan dari pengguna.
- 7) Mengatur warna transparansi.
- 8) Menyimpan komentar.
- 9) Menampilkan baris dari teks.
- 10) Menambahkan spesifikasi aplikasi yang digunakan dalam MC.

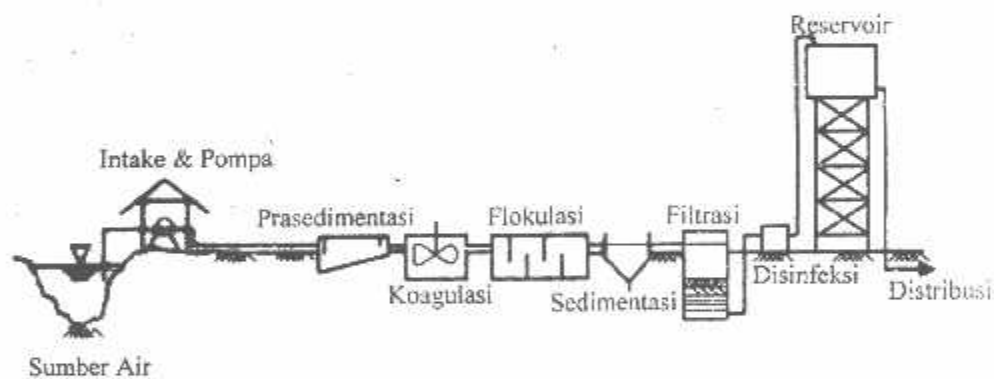
2.3. Proses pengolahan air bersih Perusahaan Daerah Air Minum (PDAM)

Air bersih adalah kebutuhan penting dalam kehidupan manusia. Dalam keseharian, air bersih digunakan untuk berbagai keperluan, dari minum, mandi, cuci, masak dan lainnya. Hasil dari aktivitas masyarakat tersebut adalah air buangan / air limbah. Selain dari rumah tangga, air buangan juga dapat berasal dari industri maupun kotapraja. Air buangan ini

adalah salah satu bahan baku dari sekian banyak jenis bahan baku yang dapat diolah menjadi air bersih.

Secara umum, pengolahan air bersih terdiri dari tiga aspek, yakni pengolahan secara fisika, kimia dan biologi. Pada pengolahan secara fisika, biasanya dilakukan secara mekanis, tanpa adanya penambahan bahan kimia. Contohnya adalah pengendapan fitrasi, adsorpsi, dan lain-lain. Pada pengolahan secara kimiawi, terdapat penambahan bahan kimia, seperti klor, tawas, dan lain-lain. Biasanya bahan ini digunakan untuk menyisihkan logam-logam berat yang terkandung dalam air. Sedangkan pada pengolahan secara biologis, biasanya memanfaatkan mikroorganisme sebagai media pengolahnya.

PDAM (Perusahaan Dagang Air Minum) memiliki WTP (Water Treatment Plant), yaitu bangunan pokok dari sistem pengolahan air bersih. Bangunan ini beberapa bagian. Secara umum, skema pengolahan air bersih dapat dilihat pada gambar 3.1 dan diagram blok dapat di lihat pada gambar 2.28 .



Gambar 2.5
Water Treatment Plant

sumber: PDAM Bandarmasih Kota Banjarmasin, Diklat Manajemen Air Minum Berbasih
Kompetensi Tingkat Muda Volume: 2

keterangan:

1. Bangunan Intake (Bangunan Pengumpul Air)

Bangunan intake berfungsi sebagai bangunan pertama untuk masuknya air dari sumber air. Sumber air utamanya diambil dari air sungai. Pada bangunan ini terdapat bar screen (penyaring kasar) yang berfungsi untuk menyaring benda-benda yang ikut tergenang dalam air, misalnya sampah, daun-daun, batang pohon, dsb.

2. Bak Prasedimentasi (optional)

Bak ini digunakan bagi sumber air yang karakteristik turbiditasnya tinggi (kekeruhan yang menyebabkan air berwarna coklat). Bentuknya hanya berupa bak sederhana, fungsinya untuk pengendapan partikel-partikel diskrit dan berat seperti pasir, dll.

3. Koagulasi

Disinilah proses kimiawi terjadi, pada proses koagulasi ini dilakukan proses destabilisasi partikel koloid, karena pada dasarnya air sungai atau air kotor biasanya berbentuk koloid dengan berbagai partikel koloid yang terkandung didalamnya. Tujuan proses ini adalah untuk memisahkan air dengan pengotor yang terlarut didalamnya, analoginya seperti memisahkan air pada susu kedelai. Pada unit ini terjadi rapid mixing (pengadukan cepat) agar koagulan dapat terlarut merata dalam waktu singkat. Bentuk alat pengaduknya dapat bervariasi, selain rapid mixing, dapat menggunakan hidrolis (hydrolic jump atau terjunan) atau mekanis (menggunakan batang pengaduk).

4. Flokulasi

Selanjutnya air masuk ke unit flokulasi. Tujuannya adalah untuk membentuk dan memperbesar flok (pengotor yang terendapkan). Di sini dibutuhkan lokasi yang alirannya tenang namun tetap ada pengadukan lambat (slow mixing) supaya flok menumpuk. Untuk meningkatkan efisiensi, biasanya ditambah dengan senyawa kimia yang mampu mengikat flok-flok tersebut.

5. Sedimentasi

Bangunan ini digunakan untuk mengendapkan partikel-partikel koloid yang sudah didestabilisasi oleh unit sebelumnya. Unit ini menggunakan prinsip berat jenis. Berat jenis partikel kolid (biasanya berupa lumpur) akan lebih besar daripada berat jenis air. Pada masa kini, unit koagulasi, flokulasi dan sedimentasi telah ada yang dibuat tergabung yang disebut unit aselator.

6. Filtrasi

Sesuai dengan namanya, filtrasi adalah untuk menyaring dengan media butiran. Media butiran ini biasanya terdiri dari antrasit, pasir silica dan kerikil silica dengan ketebalan berbeda. Cara ini dilakukan dengan metode gravitasi.

7. Disinfeksi

Setelah bersih dari pengotor, masih ada kemungkinan ada kuman dan bakteri yang hidup, sehingga ditambahkan senyawa kimia yang dapat mematikan kuman ini, biasanya berupa penambahan chlor, ozonisasi, UV, pemabasan, dan lain-lain sebelum masuk ke bangunan selanjutnya, yakni reservoir.

8. Reservoir

Reservoir berfungsi sebagai tempat penampungan sementara air bersih sebelum didistribusikan melalui pipa-pipa secara gravitasi. Namun ada juga yang sebaliknya air masuk ke reservoir secara gravitasi dan untuk pendistribusian dioperasikan menggunakan pompa yang langsung menuju pelanggan.

2.4. Pengendalian Sistem

Dalam pembuatan aplikasi kendali dibutuhkan sebuah alat untuk pengendali. Alat pengendali tersebut adalah PLC (Programmable Logic Controller). PLC (Programable Logic Controller) merupakan suatu sistem peralatan yang digunakan untuk megontrol suatu peralatan atau sistem lain dengan menggunakan suatu rangkaian logika yang dapat dapat dipogram sesuai kebutuhan.

PLC pertama kali diperkenalkan pada tahun 1960-an. Alasan peranan utama perancangan PLC adalah untuk menghilangkan beban ongkos perawatan dan penggantian sistem kontrol berbasis relay. Bedford Associates (Bedford, MA) mengajukan usulan yang diberi nama MODICON (kepanjangan dari Modular Digital Controller) untuk perusahaan-perusahaan mobil di Amerika Serikat. Sedangkan perusahaan lain mengajukan sistem berbasis komputer (PDP-8). MODICON 084 merupakan PLC pertama yang digunakan pada produk komersial.

PLC Twido adalah PLC yang dibuat oleh Schneider Telemecanique yang termasuk kategori PLC nano. PLC Twido tersedia dalam 2 model yaitu model compact dan model modular. Adapun model PLC Twido yang penulis bahas pada skripsi ini adalah model Twido modular dengan tipe TWDLMDA20DTK. Tabel maximum hardware configuration dari PLC TWDLMDA20DTK dapat dilihat pada tabel 2.8.

Tabel 2.8 Maximum Hardware Configuration

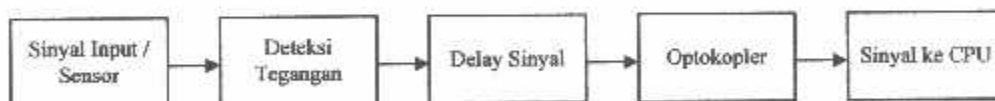
[Sumber: *Twido programmable controllers Hardware Reference Guide TWD USE 10AE eng Version 2.5*]

Controller Item	Compact controller TWDLMDA20DTK
Standard digital inputs	12
Standard digital outputs	8
Max expansion I/O modules (digital or analog)	4
Max digital inputs (controller I/O + exp I/O)	$12+(4 \times 32) = 140$
Max digital outputs (controller I/O + exp I/O)	$8+(4 \times 32) = 136$
Max digital I/O (controller I/O + exp I/O)	$20+(4 \times 32) = 148$
Max AS-Interface bus interface modules	2
Max I/O with ASInterface modules (7 I/O per slave)	$20+(2 \times 62 \times 7) = 888$

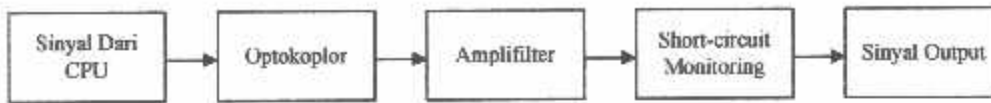
Max relay outputs	64 expansion only
Potentiometers	1
Built-in analog inputs	1
Max analog I/O (controller I/O + exp I/O)	9 in / 4 out
Remote controllers	7
Serial ports	2
Cartridge slots	2
Largest application/ backup size (KB)	32
Optional memory cartridge (KB)	32
Optional RTC cartridge	yes ²
Optional Operator Display	yes ²
Optional 2nd port	yes ²

2.5. Modul interface

Fungsi dari sebuah modul interface adalah untuk mengubah sinyal masukan ataupun sinyal keluaran dari sensor ataupun aktuator untuk diproses oleh CPU sesuai dengan kebutuhan aplikasi sistem.



Gambar 2.6
Blok Diagram Modul Input Interface



Gambar 2.7
Blok Diagram Modul Output Interface

Keterangan:

1. Pendeteksian tegangan error bertujuan meyakinkan bahwa tegangan masuk masih dalam batas yang diijinkan. Apabila tegangan terlalu tinggi maka akan diturunkan melalui diode breakdown.
 2. Delay sinyal meyakinkan tegangan yang diterima merupakan inputan yang sebenarnya. Rangkaian ini mempertahankan tegangan input sesaat (1-20 ms) untuk membedakan dengan sinyal-sinyal lain seperti tegangan interferensi.
 3. Optokopler mengirimkan informasi sensor berupa cahaya dan menciptakan isolasi elektronik antara kontrol dan rangkaian logika. Selanjutnya melindungi komponen elektronik yang sensitif dari naiknya tegangan luar secara tiba-tiba.
 4. Amplifier berguna untuk menguatkan arus listrik output sehingga nantinya cukup kuat untuk menggerakkan actuator.
 5. Short circuit monitoring berfungsi untuk memonitor terjadinya hubung singkat pada rangkaian luar dan memutuskan hubungan antara modul output dengan rangkaian luar.
-



BAB III

PERANCANGAN DAN DESAIN APLIKASI

3.1. Deskripsi Umum

User interface untuk Alat Pelatihan PLC Dengan Studi Kasus Water Treatment Plant adalah suatu aplikasi yang di bangun untuk melakukan simulasi sistem otomasi industri pada proses produksi air bersih, yang dikemas dalam bentuk simulasi sebagai pengganti miniatur atau prototipe plant.

Pada kasus ini aplikasi kusus di bangun untuk modul kontrol berbasis PLC yang nantinya akan digunakan sebagai media pembelajaran atau pelatihan PLC di salah satu laboratorium di jurusan teknik elektro Institute Teknologi Nasional Malang karena saat ini didalam pembelajaran atau pelatihan PLC tersebut hanya menggunakan lampu – lampu untuk mengindikasikan bahwa sistem tersebut telah beroperasi. Dengan adanya aplikasi ini kita dapat melihat secara riil bagaimana proses produksi tersebut berjalan.

3.2. Tahap Identifikasi

Pada tahap ini identifikasi permasalahan yang akan dibuat pada sistem berkaitan dan terbatas pada aplikasi yang dapat mensimulasikan proses produksi air bersih sebagai pengganti prototipe proses produksi air bersih, konfigurasi software simulasi → protokol komunikasi → modul PLC, serta peningkatan efisiensi dan kuantitas produksi air bersih..

3.3. Tahap Konseptualisasi

Dalam tahap konseptualisasi ini membuat gambaran cara kerja aplikasi sesuai dengan fungsinya dan menentukan apa saja yang terkait dengan aplikasi yang dirancang.

3.3.1. User Interface Untuk Mensimulasikan Water Treatment Plant

Didalam aplikasi user interface modul kontrol berbasis PLC ini akan mensimulasi proses produksi air bersih, berikut adalah komponen – komponen yang akan di simulasikan aplikasi user interface:

1. Bangunan Intake

Terdapat 3 buah pompa di bangunan Intake yang nantinya akan di kontrol PLC .

2. Bak Prasedimentasi

Seperti yang terdapat di water treatment yang sebenarnya, di aplikasi ini Bak Prasedimentasi akan menampung air yang di alirkan oleh pompa intake.

3. Bak Koagulasi

Setelah air penuh di Bak Prasedimentasi air akan mengalir ke Bak koagulasi, disini terdapat pompa, pompa ini berfungsi untuk memutar mixer untuk proses pengadukan. Pompa koagulasi adalah salah satu yang di kontrol PLC.

4. Bak Flokulasi

Setelah Bak koagulasi penuh air akan mengalir ke Bak flokulasi.

5. Bak Sedimentasi

Setelah Bak koagulasi penuh air akan mengalir ke Bak flokulasi. Di sedimentasi terdapat pompa vakum yang nantinya juga akan di kontrol PLC.

6. Bak Filtrasi

Setelah Bak flokulasi penuh air akan mengalir ke Bak filtrasi.

7. Bak Disinfeksi

Setelah Bak filtrasi penuh air akan mengalir ke Bak disinfeksi.

8. Bak reservoir

Setelah Bak disinfeksi penuh air akan mengalir ke Bak reservoir.

9. Distribusi

Terdapat pompa distribusi yang akan mendistribusikan air dari Bak reservoir ke pelanggan dengan menggunakan 3 buah kran. Pompa distribusi juga akan di kontrol oleh PLC.

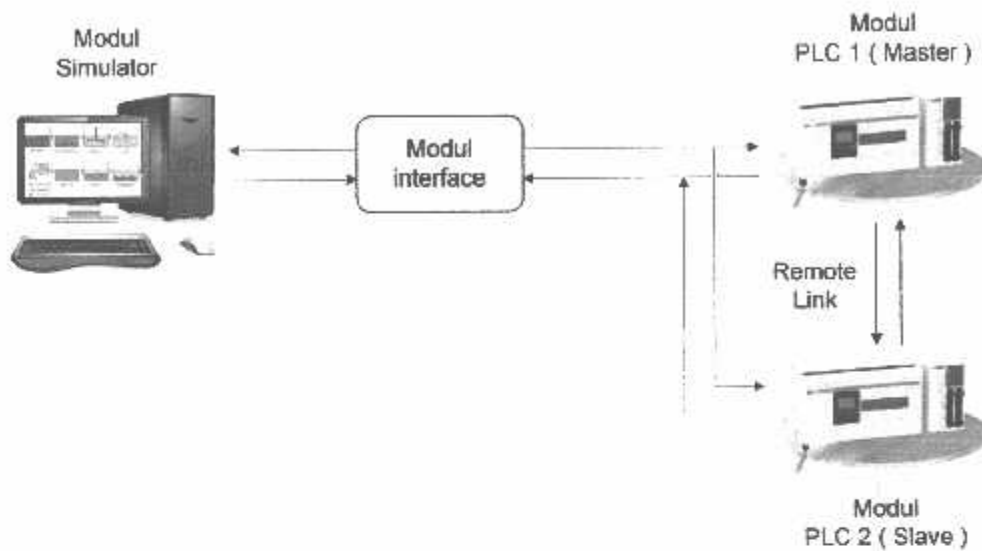
3.4. Penentuan Komponen Dasar

Berdasarkan konseptualisasi maka di butuhkan komponen dasar pembangun sebagai berikut :

1. Borland Delphi 7.0 yang akan digunakan untuk membuat aplikasi user interface ini.
2. Modul PLC type TWDLMDA20DTK dengan Jumlah inputan dan outputan yang digunakan untuk sistem adalah 23 I/O, masing-masing terdiri dari 11 inputan dan 12 outputan. Jumlah I/O yang tersedia pada PLC type TWDLMDA20DTK sebanyak 20 I/O, maka dibutuhkan 2 buah modul PLC untuk mencukupi kebutuhan I/O.
3. Karena aplikasi ini menggunakan piranti luar maka dibutuhkan modul protokol yang disebut dengan interface sebagai media komunikasi, protokol ini berfungsi menerjemahkan perintah modul PLC berupa sinyal analog menjadi sinyal digital untuk mengontrol aplikasi ini.
4. Mikrokontroler tipe ATMEGA 8535 komponen utama untuk membuat minimum sistem modul interface.

3.5. Diagram Blok

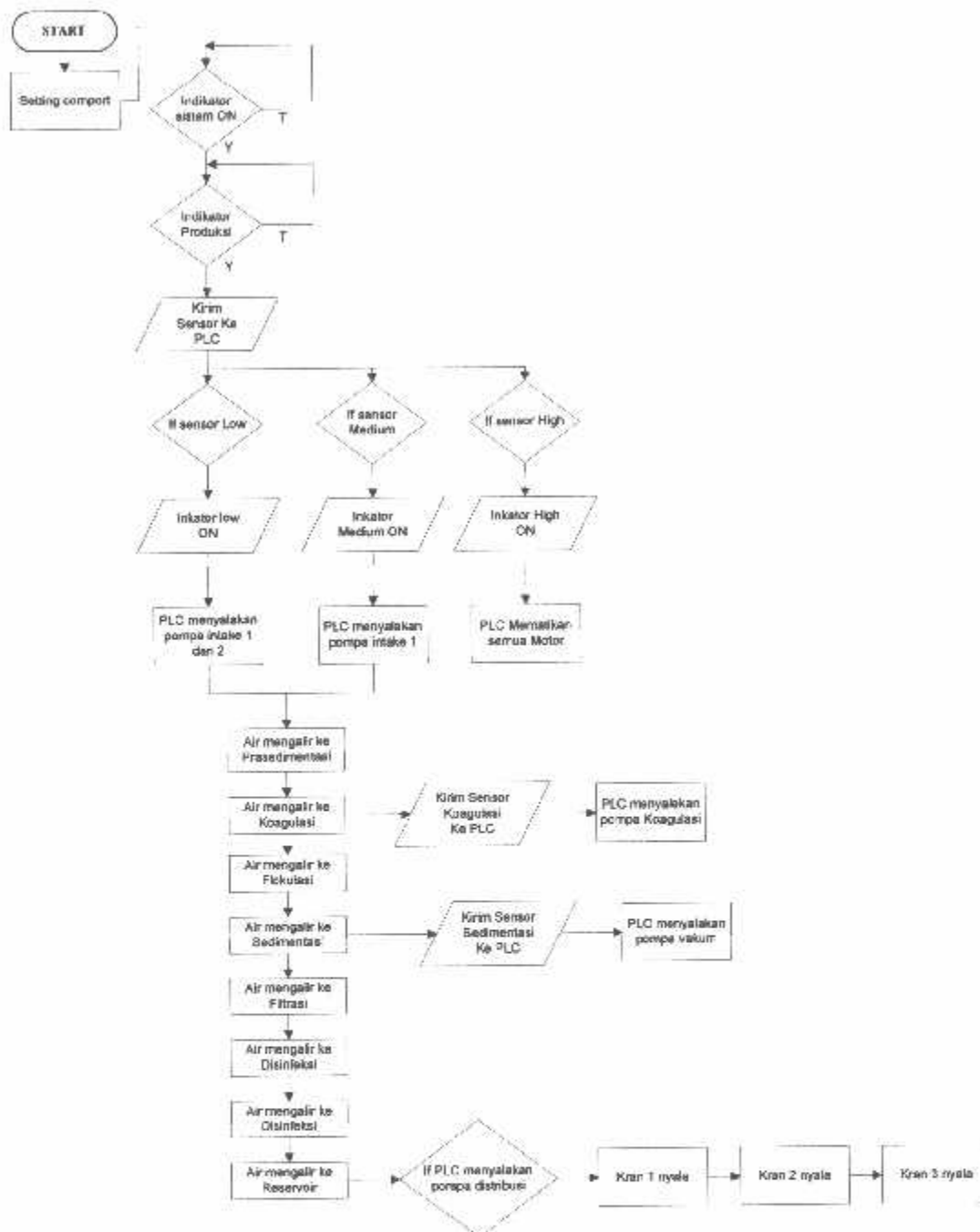
Dalam pembuatan sebuah aplikasi diagram blok adalah salah satu bagian penting, karena dari diagram blok dapat diketahui cara kerja aplikasi. Sehingga keseluruhan sistem dapat di fungsikan sesuai dengan perancangan. Diagram blok aplikasi user interface modul kontrol berbasis PLC dapat dilihat pada gambar



Gambar 3.1
Diagram Blok User Interface modul Kontrol Berbasis PLC

3.6. Diagram Alir (Flowchart)

Sebelum masuk ke tahap perancangan terlebih dahulu harus membuat diagram alir atau yang sering disebut dengan flowchart. Pembuatan diagram alir ini bertujuan untuk melihat gambaran bagaimana jalannya program dari awal (start) hingga akhir (end). Diagram alir aplikasi user interface modul control berbasis PLC dapat dilihat pada gambar 3.2



Gambar 3.2
 Diagram Alir User Interface modul Kontrol Berbasis PLC

3.7. Pengalamatan

3.7.1. Pengalamatan Inputan Dari PLC 1 (Master)

Tabel 3.1 Pengalamatan Inputan dari PLC 1 (Master)

No	Input	Inisial	Alias	Keterangan
1	Q0.0	0		Indikator Sistem
2	Q0.1	1		Indikator Pompa Intake 1
3	Q0.2	2		Indikator Pompa Intake 2
4	Q0.3	3		Indikator Pompa Intake 3
5	Q0.4	4		Indikator Low Reservoir
6	Q0.5	5		Indikator Medium Reservoir
7	Q0.6	6		Indikator High Reservoir
8	Q0.7	7		Indikator Motor Koagulasi Kecepatan Tinggi

3.7.2. Pengalamatan Inputan Dari PLC 2 (Slave)

Tabel 3.2 Pengalamatan Inputan dari PLC 2 (Slave)

No	Input	Inisial	Alias	Keterangan
1	Q0.0	0	Q1.0.0	Indikator Motor Koagulasi Kecepatan Rendah
2	Q0.1	1	Q1.0.1	Indikator Pompa Vakum
3	Q0.2	2	Q1.0.2	Indikator Pompa Distribusi
4	Q0.3	3	Q1.0.3	Indikator Indikator Produksi

3.7.3. Pengalamatan Mikrokontroler (Modul Interface)

Tabel 3.3 Pengalamatan Dari PLC Ke Mikrokontroler

No	Data	Port	Inputan PLC 1	Inputan PLC 2	Keterangan
1	0	PA0	Q0.0		Indikator Sistem
2	1	PA1	Q0.1		Pompa Intake 1
3	2	PA2	Q0.2		Pompa Intake 2
4	5	PA5	Q0.3		Pompa Intake 3
5	6	PA6	Q0.4		Indikator Low Reservoir
6	7	PA7	Q0.5		Indikator Medium Reservoir
7	8	PB0	Q0.6		Indikator High Reservoir
8	9	PB1	Q0.7		Motor Koagulasi Kecepatan Tinggi
9	10	PB2		Q0.0	Motor Koagulasi Kecepatan Rendah
10	11	PB3		Q0.1	Pompa Vakum
11	12	PB4		Q0.2	Pompa Distribusi
12	13	PB5		Q0.3	Indikator Produksi

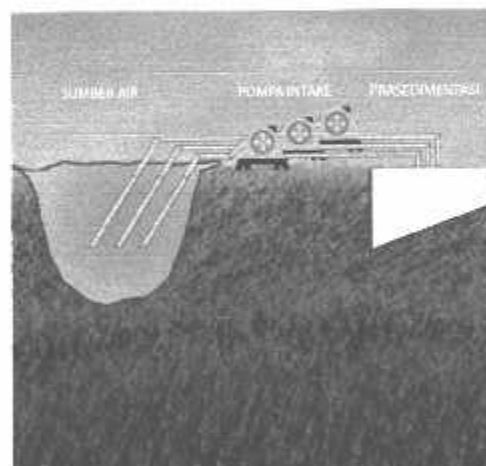
3.8. Perancangan Dan Desain Aplikasi

Pada tahap awal pembuatan aplikasi adalah perancangan desain aplikasi. Berikut adalah tahap- tahap dalam membuat aplikasi user interface module kontrol berbasis PLC untuk sistem kontrol proses produksi air bersih adalah sbb:

3.8.1. Perancangan Gambar Gif

Karena di Borland Delphi komponen yang dapat membuat animasi bergerak sangat terbatas, maka aplikasi ini menggunakan gambar gif untuk membuat animasi bergerak, berikut adalah perancangan gambar gif yang ada di aplikasi.

1. Perancangan animasi Bak Prasedimentasi
 - a. Membuat perancangan gambar Bak Prasedimentasi

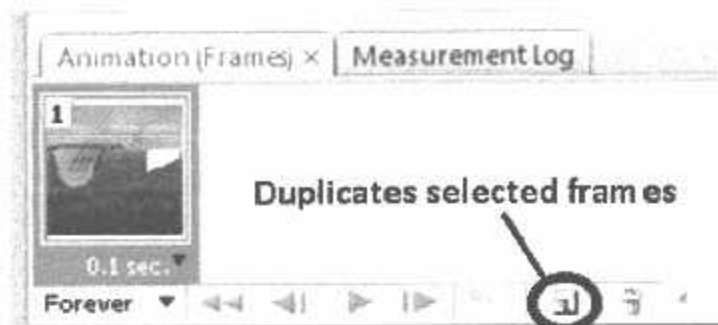


Gambar 3.3
Perancangan Bak Prasedimentasi

- b. Membuat animasi pengisian air Bak Prasedimentasi

Berikut adalah langkah-langkah membuat animasi pengisian air pada Bak Prasedimentasi.

- Pertama membuat frame animasi



Gambar 3.4
Membuat frame animasi Bak Prasedimentasi

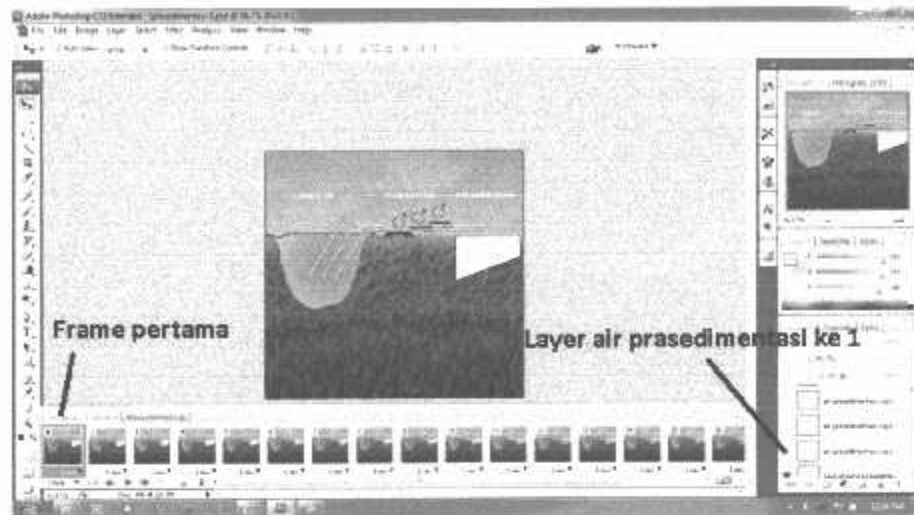
- Selanjutnya buat 19 frame dalam jendela animasi dengan mengklik icon “duplicates selected frames” (tanda lingkaran pada gambar 3.4) sehingga jadi seperti pada gambar 3.5



Gambar 3.5
19 frame animasi

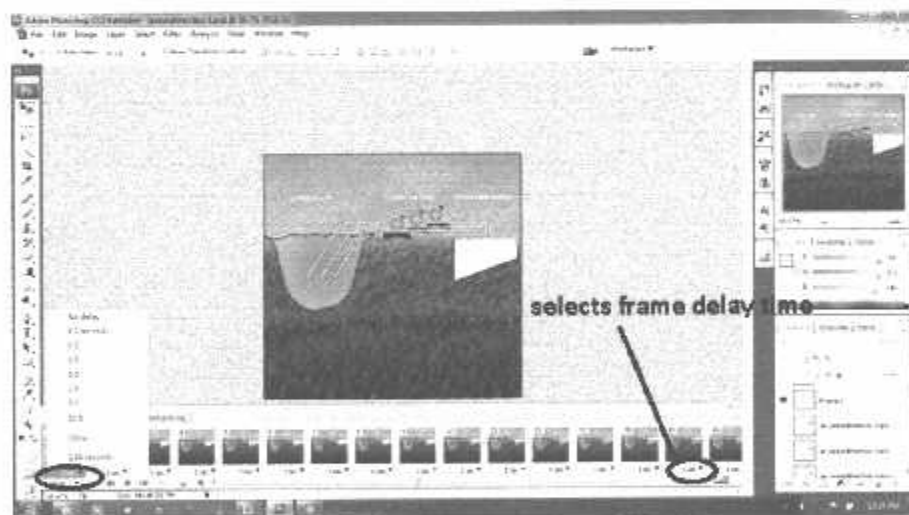
- Untuk membuat animasi Bak terisi air secara bertahap ,klik frame pertama lalu nonaktifkan semua layer air Prasedimentasi, kemudian di lanjutkan dengan mengklik frame kedua dan mengaktifkan layer air Prasedimentasi 1. Setelah itu dilanjutkan dengan mengklik frame ketiga dan mengaktifkan layer air Prasedimentasi 2, langkah ini dibuat sampai pada frame ke 19.

Proses pembuatan animasi pengisian air pada Bak Prasedimentasi dapat dilihat pada gambar 3.6



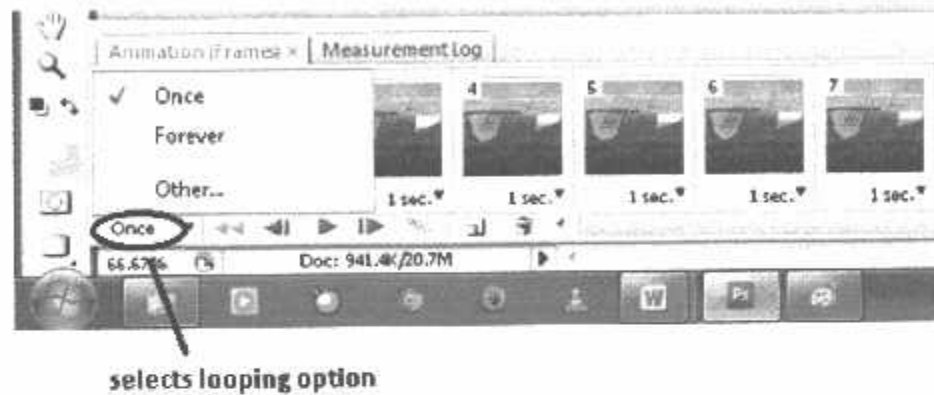
Gambar 3.6
Proses pembuatan animasi Bak Prasedimentasi

- Setelah proses pembuatan animasi selesai tentukan delay time pada jendela animation dengan mengklik icon yang di lingkari pada gambar 3.7, setelah itu pilih waktu delay. Pada animasi pengisian air Bak Prasedimentasi menggunakan delay time 1 second.



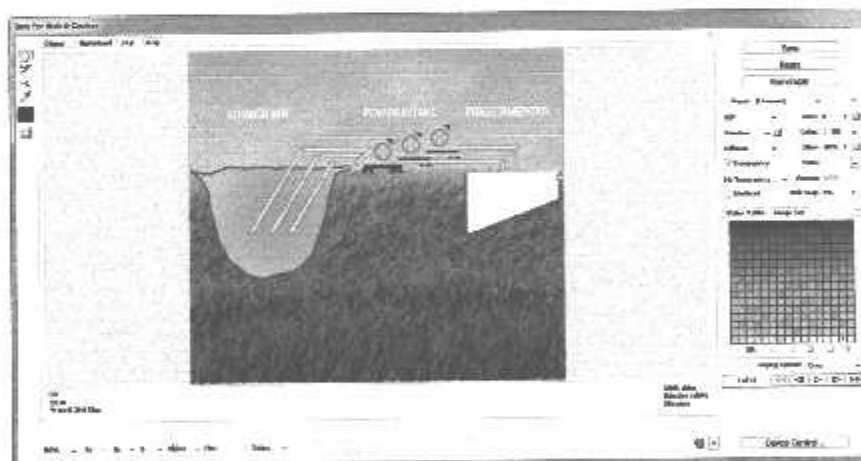
Gambar 3.7
Penentuan delay time

- Karena animasi pengisian air pada Bak Prasedimentasi bergerak satu kali maka harus di setting looping pada icon selects looping option menjadi "once".



Gambar 3.8
Penentuan looping delay time

- Setelah itu simpan gambar dalam bentuk ekstensi gif image dengan cara klik file → save for web & device atau shortcut CTRL+ALT+SHIFT+S. gambar jendela penyimpanan gambar gif dapat dilihat pada gambar 3.8



Gambar 3.9
Jendela penyimpanan gambar gif

2. Perancangan animasi Bak Koagulasi

a. Membuat perancangan gambar Bak Koagulasi



Gambar 3.10
Perancangan Bak Koagulasi

b. Membuat animasi pengisian air Bak Koagulasi

Berikut adalah langkah-langkah membuat animasi pengisian air pada Bak Koagulasi.

- Pertama membuat frame animasi



Gambar 3.11
Membuat frame animasi Bak Koagulasi

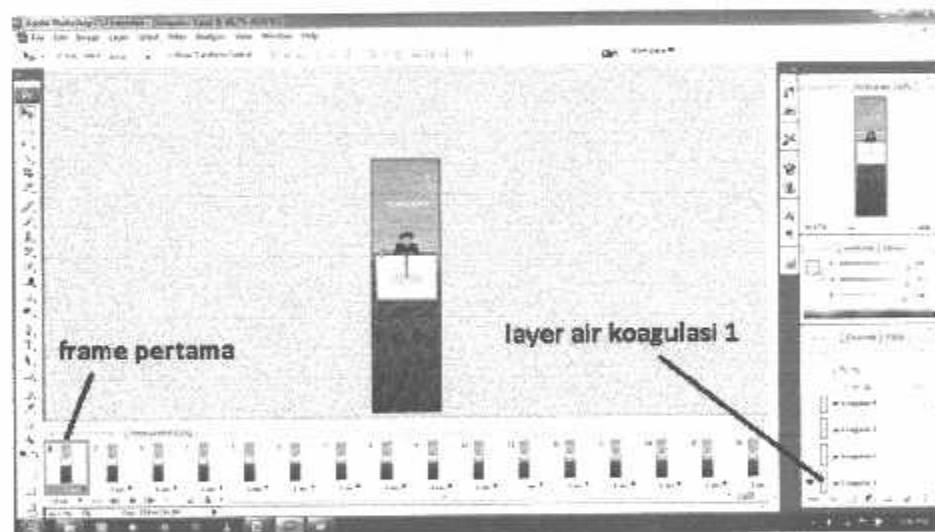
- Selanjutnya buat 18 frame dalam jendela animasi dengan mengklik icon "duplicates selected frames" (tanda lingkaran pada gambar 3.4) sehingga jadi seperti pada gambar 3.12



Gambar 3.12
18 frame animasi

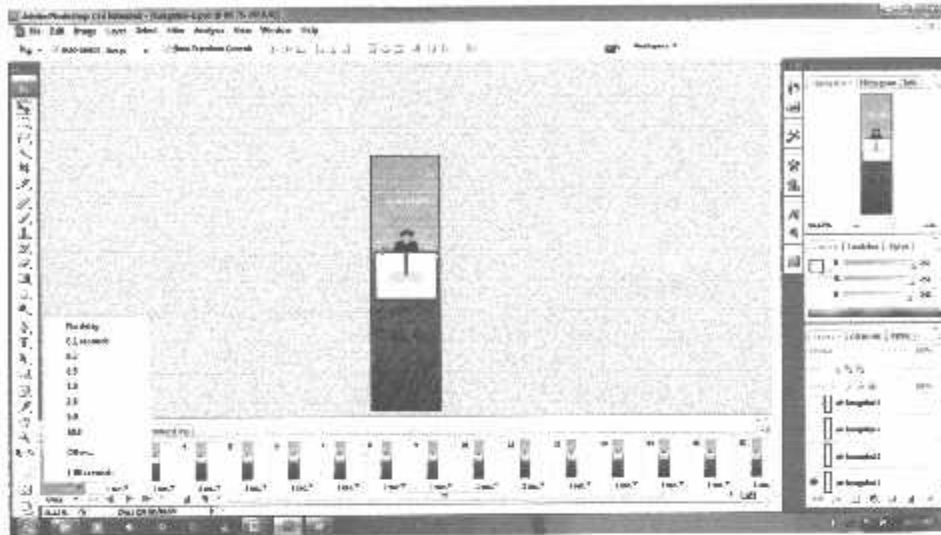
- Untuk membuat animasi Bak terisi air secara bertahap, klik frame pertama lalu aktifkan layer air koagulasi 1, kemudian dilanjutkan dengan mengklik frame kedua dan mengaktifkan layer air koagulasi 2. Setelah itu dilanjutkan dengan mengklik frame ketiga dan mengaktifkan layer air koagulasi 3, langkah ini dibuat sampai pada frame ke 18.

Proses pembuatan animasi pengisian air pada Bak Koagulasi dapat dilihat pada gambar 3.13



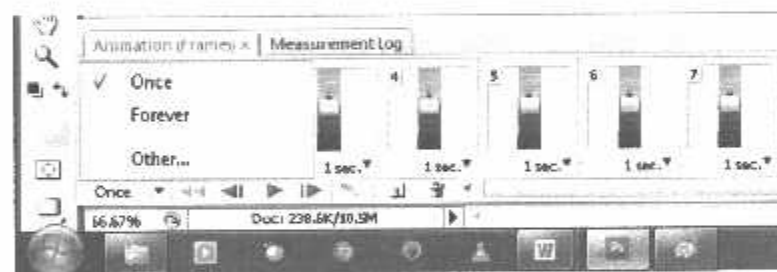
Gambar 3.13
Proses pembuatan animasi Bak Koagulasi

- Setelah proses pembuatan animasi selesai tentukan delay time pada jendela animation dengan mengklik icon "selects delay time" (seperti pada gambar 3.7), setelah itu pilih waktu delay. Pada animasi pengisian air Bak Koagulasi menggunakan delay time 1 second.



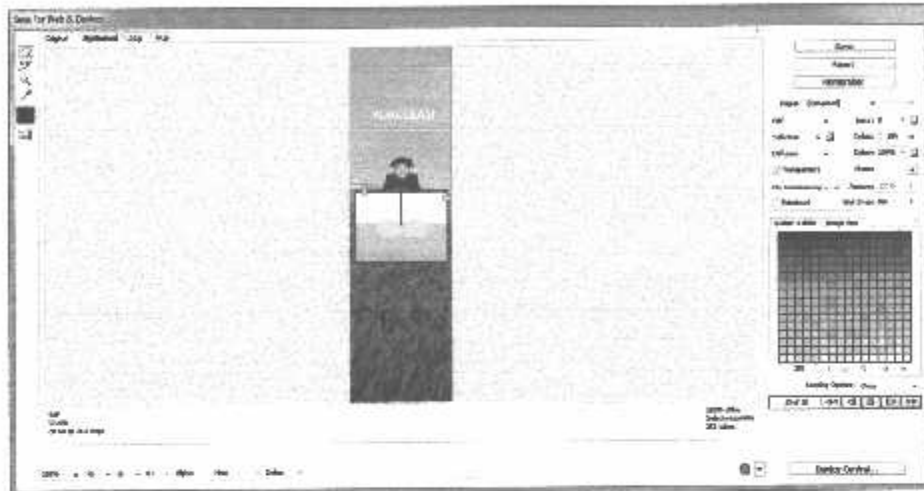
Gambar 3.14
Penentuan delay time

- Karena animasi pengisian air pada Bak Koagulasi bergerak satu kali maka harus di setting looping pada icon selects looping option (seperti pada gambar 3.8) menjadi "once". Gambar penentuan looping delay time pada animasi pengisian air Bak Koagulasi dapat dilihat pada gambar 3.15



Gambar 3.15
Penentuan looping delay time

- Setelah itu simpan gambar dalam bentuk ekstensi gif image dengan cara klik file → save for web & device atau shortcut CTRL+ALT+SHIFT+S. gambar jendela penyimpanan gambar gif dapat dilihat pada gambar 3.16



Gambar 3.16
Jendela penyimpanan gambar gif

3. Perancangan animasi Bak Flokulasi

a. Membuat perancangan gambar Bak Flokulasi

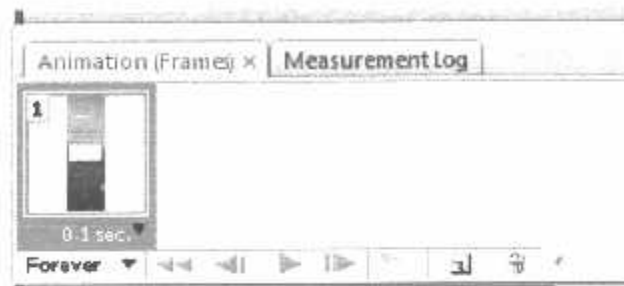


Gambar 3.17
Perancangan Bak Flokulasi

b. Membuat animasi pengisian air Bak Flokulasi

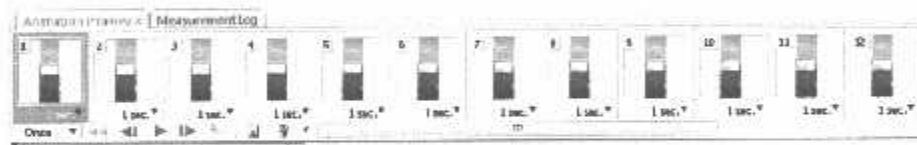
Berikut adalah langkah-langkah membuat animasi pengisian air pada Bak Flokulasi.

- Pertama membuat frame animasi



Gambar 3.18
Membuat frame animasi Bak Flokulasi

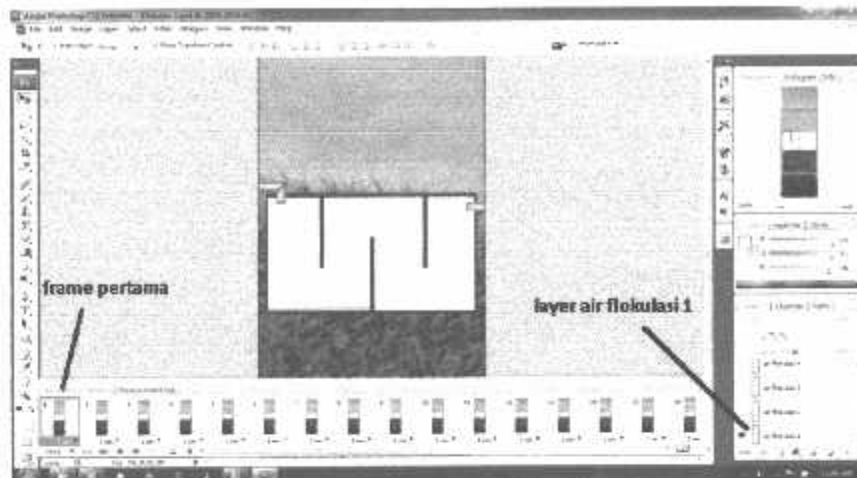
- Selanjutnya buat 32 frame dalam jendela animasi dengan mengklik icon “duplicates selected frames” (tanda lingkaran pada gambar 3.4) sehingga jadi seperti pada gambar 3.19



Gambar 3.19
32 frame animasi

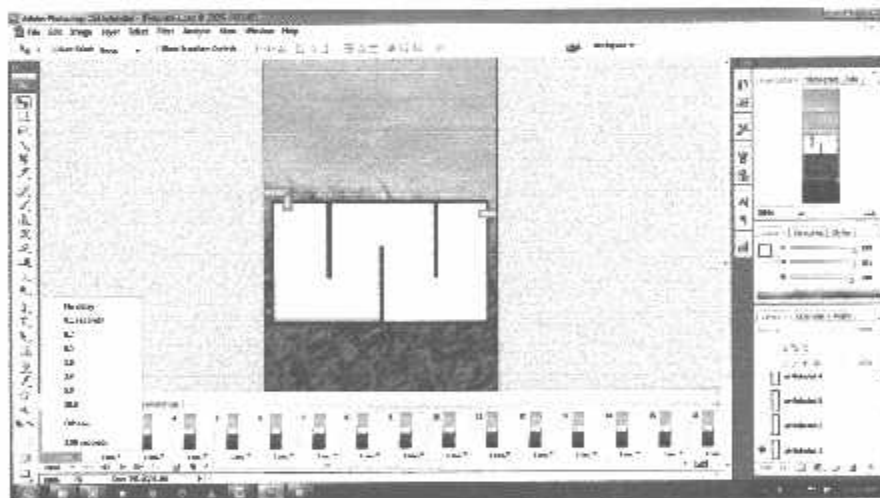
- Untuk membuat membuat animasi Bak terisi air secara bertahap ,klik frame pertama lalu aktifkan layer air Flokulasi 1, kemudian di lanjutkan dengan mengklik frame kedua dan mengaktifkan layer air Flokulasi 2. Setelah itu dilanjutkan dengan mengklik frame ketiga dan mengaktifkan layer air Flokulasi 3, langkah ini dibuat sampai pada frame ke 32.

Proses pembuatan animasi pengisian air pada Bak Flokulasi dapat dilihat pada gambar 3.20



Gambar 3.20
Proses pembuatan animasi Bak Flokulasi

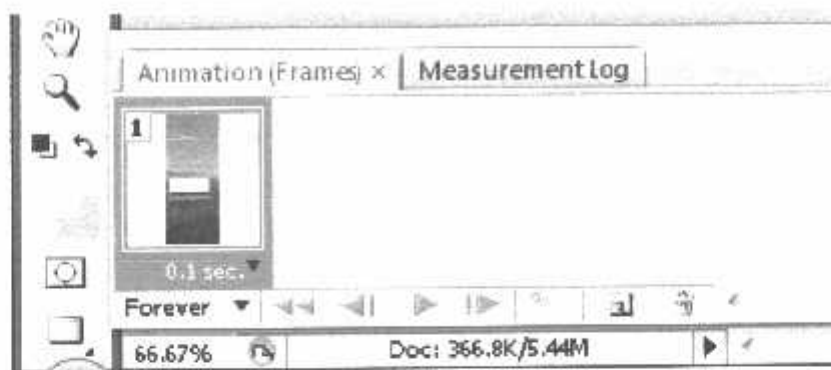
- Setelah proses pembuatan animasi selesai tentukan delay time pada jendela animation dengan mengklik icon “selects delay time” (seperti pada gambar 3.7), setelah itu pilih waktu delay. Pada animasi pengisian air Bak Flokulasi menggunakan delay time 1 second.



Gambar 3.21
Penentuan delay time

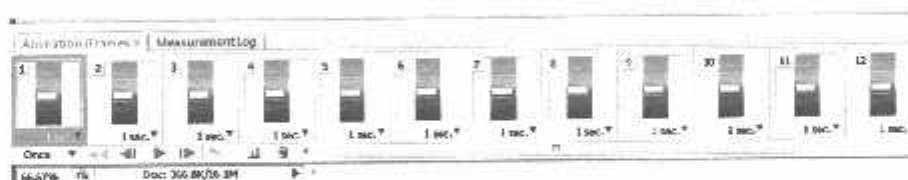
- Karena animasi pengisian air pada Bak Flokulasi bergerak satu kali maka harus di setting looping pada icon selects looping option (seperti pada

- Pertama membuat frame animasi



Gambar 3.32
Membuat frame animasi Bak Filter

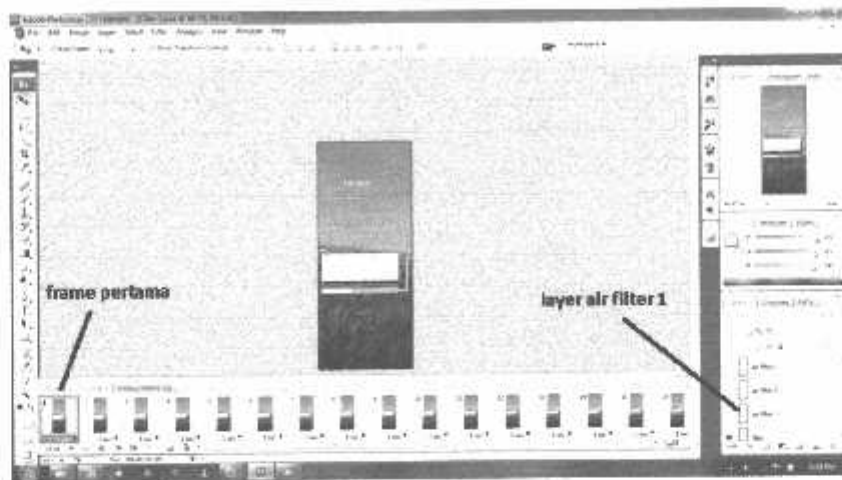
- Selanjutnya buat 27 frame dalam jendela animasi dengan mengklik icon “duplicates selected frames” (tanda lingkaran pada gambar 3.4) sehingga jadi seperti pada gambar 3.33



Gambar 3.33
27 frame animasi

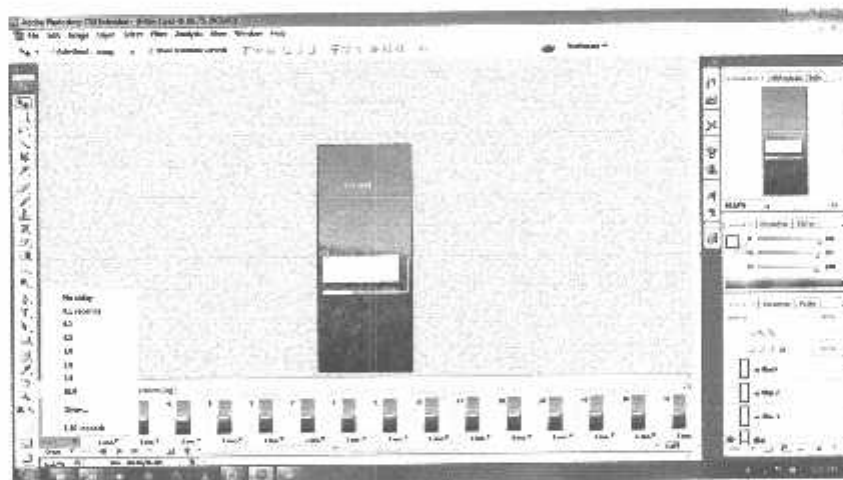
- Untuk membuat animasi Bak terisi air secara bertahap ,klik frame pertama lalu nonaktifkan semua layer air filter, kemudian di lanjutkan dengan mengklik frame kedua dan mengaktifkan layer air filter 1. Setelah itu dilanjutkan dengan mengklik frame ketiga dan mengaktifkan layer air Sedimentasi 2, langkah ini dibuat sampai pada frame ke 27.

Proses pembuatan animasi pengisian air pada Bak Filter dapat dilihat pada gambar 3.34



Gambar 3.34
Proses pembuatan animasi Bak Filter

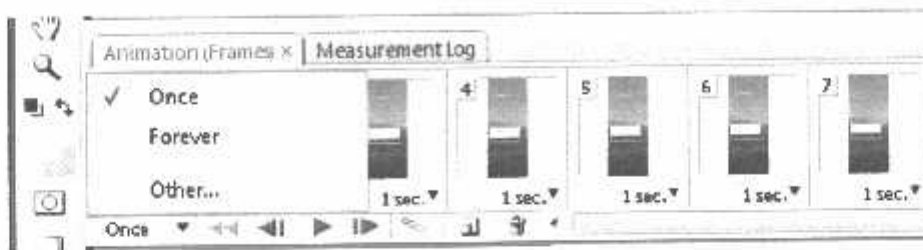
- Setelah proses pembuatan animasi selesai tentukan delay time pada jendela animation dengan mengklik icon “selects delay time” (seperti pada gambar 3.7), setelah itu pilih waktu delay. Pada animasi pengisian air Bak Filter menggunakan delay time 1 second.



Gambar 3.35
Penentuan delay time

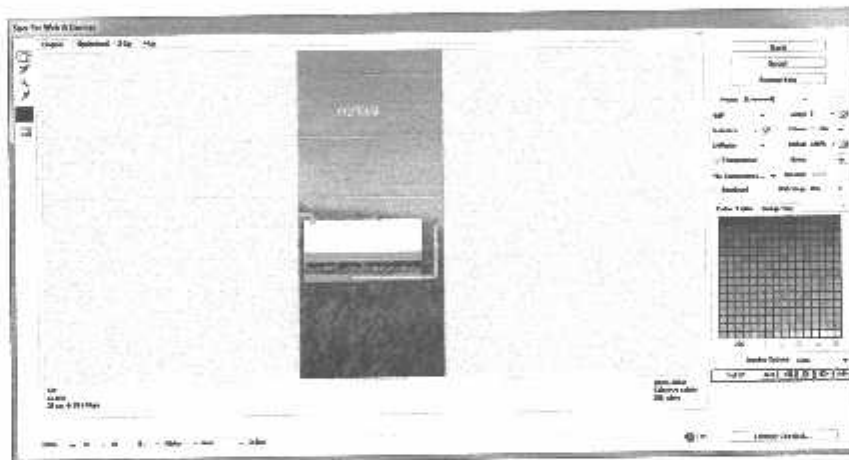
- Karena animasi pengisian air pada Bak Filter bergerak satu kali maka harus di setting looping pada icon selects looping option (seperti pada gambar 3.8)

menjadi “once”. Gambar penentuan looping delay time pada animasi pengisian air Bak Filter dapat dilihat pada gambar 3.36



Gambar 3.36
Penentuan looping delay time

- Setelah itu simpan gambar dalam bentuk ekstensi gif image dengan cara klik file → save for web & device atau shortcut CTRL+ALT+SHIFT+S. gambar jendela penyimpanan gambar gif dapat dilihat pada gambar 3.37



Gambar 3.37
Jendela penyimpanan gambar gif

6. Perancangan animasi Bak Disinfeksi
 - a. Membuat perancangan gambar Bak Disinfeksi

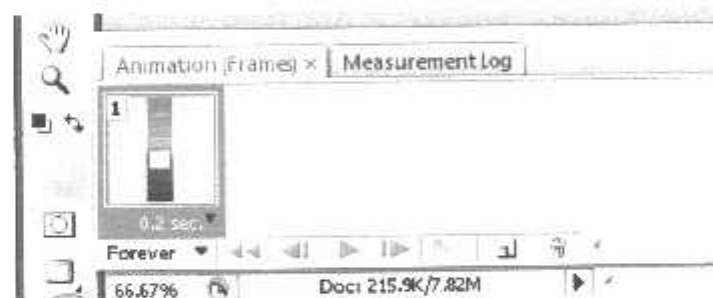


Gambar 3.38
Perancangan Bak Disinfeksi

b. Membuat animasi pengisian air Bak Disinfeksi

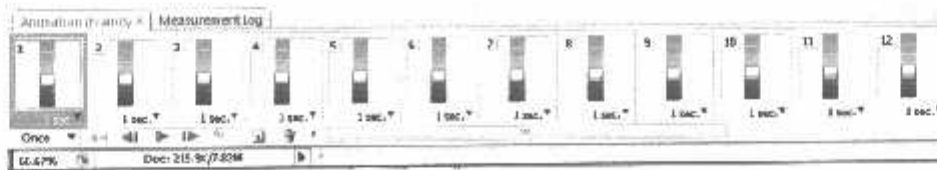
Berikut adalah langkah-langkah membuat animasi pengisian air pada Bak Disinfeksi.

- Pertama membuat frame animasi.



Gambar 3.39
Membuat frame animasi Bak Disinfeksi

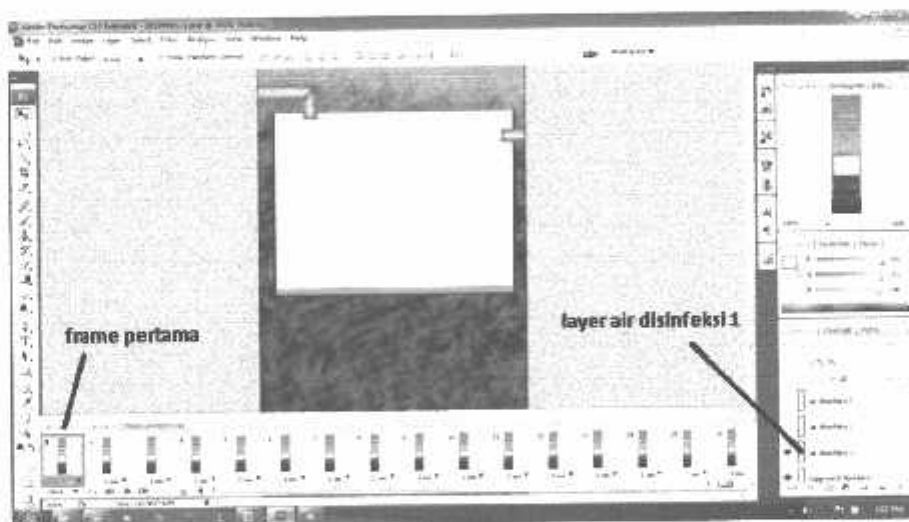
- Selanjutnya buat 33 frame dalam jendela animasi dengan mengklik icon "duplicates selected frames" (tanda lingkaran pada gambar 3.4) sehingga jadi seperti pada gambar 3.40



Gambar 3.40
33 frame animasi

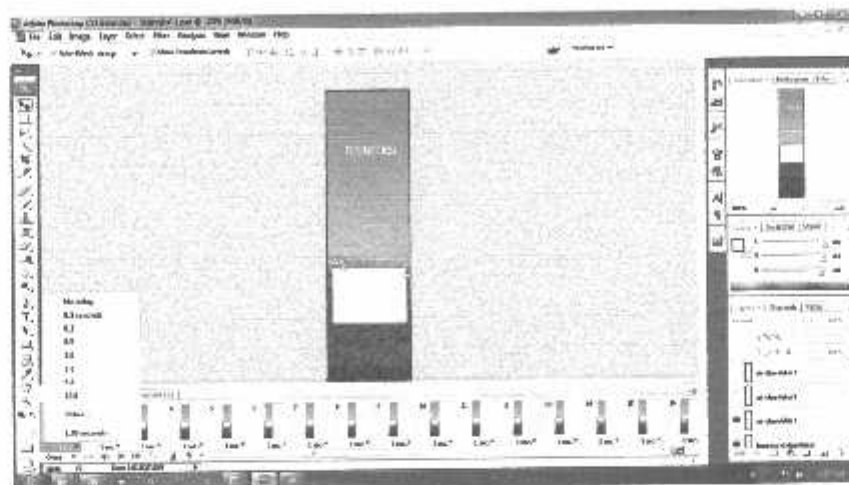
- Untuk membuat membuat animasi Bak terisi air secara bertahap ,klik frame pertama lalu aktifkan layer air Disinfeksi 1, kemudian di lanjutkan dengan mengklik frame kedua dan mengaktifkan layer air Disinfeksi 2. Setelah itu dilanjutkan dengan mengklik frame ketiga dan mengaktifkan layer air Disinfeksi 3, langkah ini dibuat sampai pada frame ke 33.

Proses pembuatan animasi pengisian air pada Bak Disinfeksi dapat dilihat pada gambar 3.41



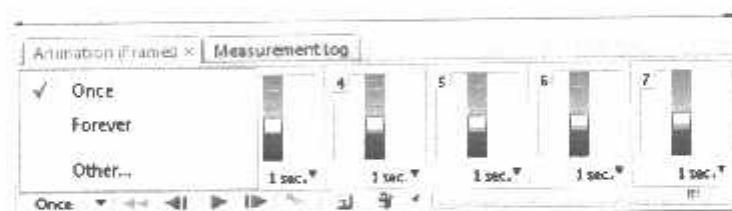
Gambar 3.41
Proses pembuatan animasi Bak Disinfeksi

- Setelah proses pembuatan animasi selesai tentukan delay time pada jendela animation dengan mengklik icon "selects delay time" (seperti pada gambar 3.7), setelah itu pilih waktu delay. Pada animasi pengisian air Bak Disinfeksi menggunakan delay time 1 second.



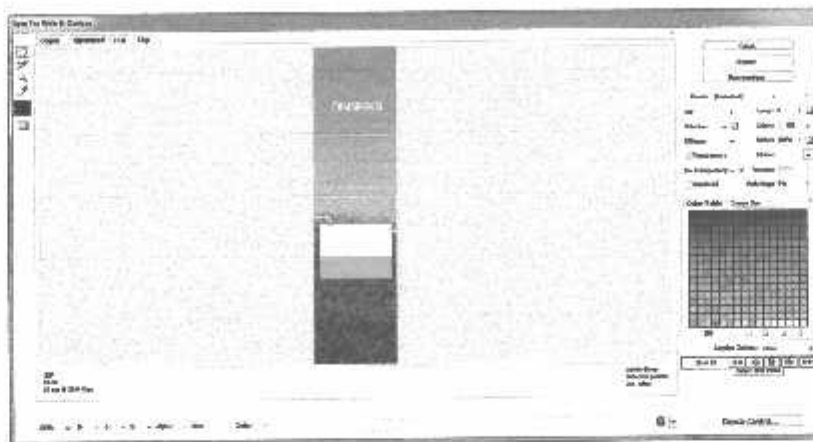
Gambar 3.42
Penentuan delay time

- Karena animasi pengisian air pada Bak Disinfeksi bergerak satu kali maka harus di setting looping pada icon selects looping option (seperti pada gambar 3.8) menjadi "once". Gambar penentuan looping delay time pada animasi pengisian air Bak Disinfeksi dapat dilihat pada gambar 3.43



Gambar 3.43
Penentuan looping delay time

- Setelah itu simpan gambar dalam bentuk ekstensi gif image dengan cara klik file → save for web & device atau shortcut CTRL+ALT+SHIFT+S. gambar jendela penyimpanan gambar gif dapat dilihat pada gambar 3.44



Gambar 3.44
Jendela penyimpanan gambar gif

3.8.2. Desain Aplikasi

Didalam membangun program aplikasi, pemrograman dilakukan dengan memanfaatkan komponen visual yang terdapat pada Borland Delphi. Dengan metode ini aplikasi akan lebih unggul dalam tampilan dan juga dalam hal waktu pengerjaan. Berikut adalah komponen – komponen yang digunakan untuk membuat aplikasi user interface untuk modul kontrol berbasis PLC:

1. Label

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

2. Edit

Adalah salah satu komponen di dalam Delphi yang berfungsi sebagai inputan. Edit juga sering dikenal dengan nama *textbox*.

3. 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

4. Timer

Komponen ini adalah komponen yang digunakan untuk masalah pewaktuan. Dalam komponen ini kejadian maupun pewaktuan bisa diatur sendiri, misalnya untuk membuat tampilan kalender, jam, atau kejadian lain yang berhubungan dengan pewaktuan.

5. Groupbox

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

6. Image

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

7. Comport

Komponen ini digunakan sebagai *driver* komunikasi serial antara Delphi dengan perangkat luar.

8. Comdatapaket

Komponen Delphi yang digunakan untuk mengakses kiriman data yang di terima comport.

9. Checkbox

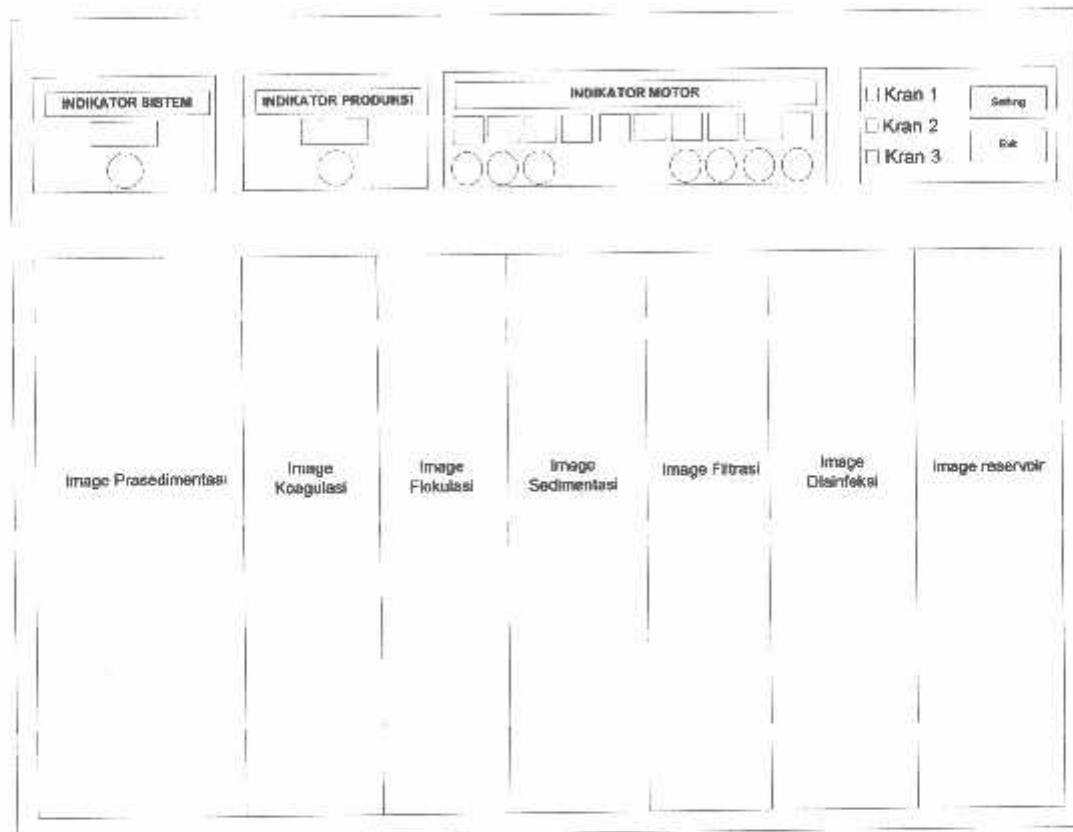
Checkbox adalah salah satu komponen Delphi yang fungsinya hamper mirip dengan button yaitu memproses suatu kejadian sesuai program yang dimasukan.

10. Xpmanifest

Komponen ini berfungsi untuk membuat tampilan button lebih bagus.

3.8.3. Form Simulasi

Form simulasi adalah form yang digunakan untuk menjalankan simulasi proses produksi air bersih, dari prsoses pengisian air Baku di Bak Prasedimentasi sampai di proses distribusi. Pada aplikasi user interface ini hanya menggunakan satu form. Gambar form simulasi dapat di lihat pada gambar 3.10



Gambar 3.45
Perancangan Form Simulasi

Didalam form simulasi terdapat beberapa komponen, seperti button yang digunakan untuk mengatur komunikasi dengan interface, edit digunakan untuk menampilkan data yang dikirim oleh interface, image yang digunakan untuk menampilkan animasi gif, checkbox yang digunakan untuk menyalakan dan menghidupkan kran, dan beberapa komponen yang lain.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

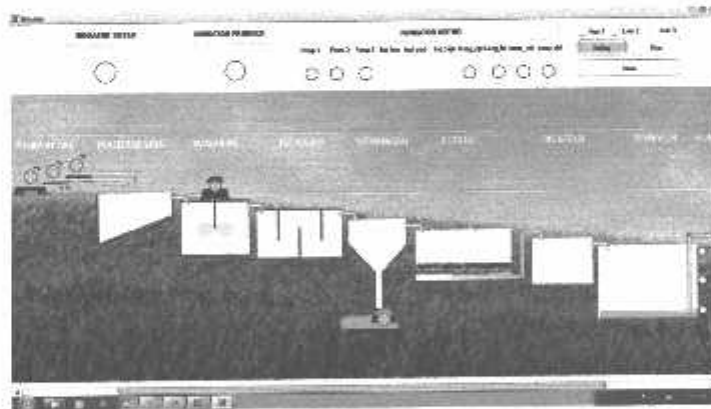
4.1. Spesifikasi Program

Program aplikasi user interface untuk Alat Pelatihan PLC Dengan Studi Kasus Water Treatment Plant adalah program aplikasi yang dibuat khusus untuk modul kontrol berbasis PLC yang akan digunakan sebagai media pembelajaran atau pelatihan PLC disalah satu laboratorium di jurusan Teknik Elektro Institut Teknologi Nasional Malang.

Aplikasi user interface ini dapat mensimulasikan tahap-tahap proses produksi air baku menjadi air bersih, mulai dari pengambilan air baku dari sungai yang dialirkan ke bak prasedimentasi melalui 3 buah pompa intake, sampai proses pendistribusian.

4.2. Pengujian

4.2.1. Form Simulasi



Gambar 4.1
Form simulasi pertama kali di jalankan

Gambar di atas adalah tampilan saat form simulasi di jalankan, pada kondisi ini komunikasi antara aplikasi dan interface belum terhubung. Untuk indikator sistem, pompa 1, pompa 2, pompa 3, level *low*, level *medium*, level *high*, motor koagulasi cepat, motor

koagulasi lambat, pompa vakum, pompa distribusi, dan indikator produksi lampu indikator berwarna putih serta textbox dalam keadaan kosong.

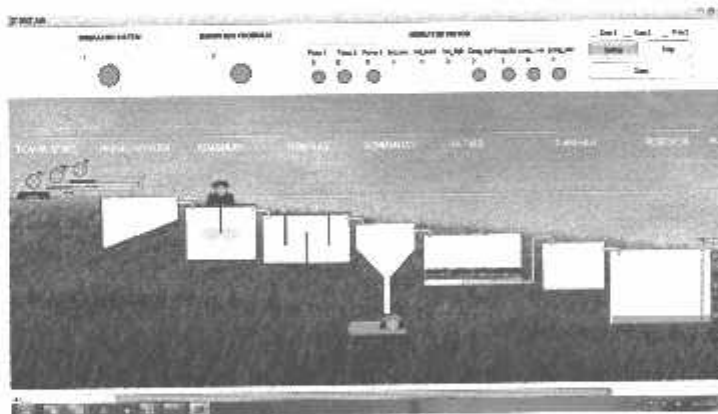
4.2.2. Pengaturan Koneksi Comport



Gambar 4.2
Pengaturan koneksi comport

Gambar diatas adalah gambar pengaturan koneksi comport, disini komunikasi antara aplikasi dan interface dilakukan secara serial. Port yang digunakan adalah com4.

4.2.3. Form Simulasi Setelah Koneksi Terhubung



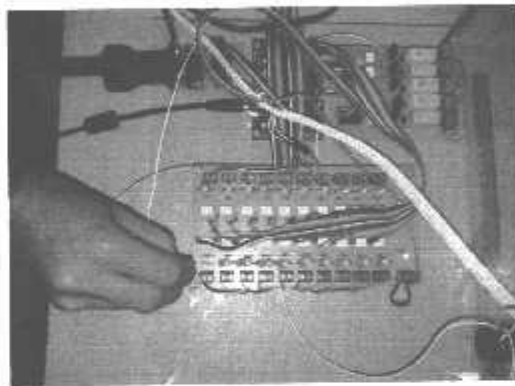
Gambar 4.3
Form simulasi setelah terhubung dengan interface

Setelah komunikasi sudah terhubung, maka lampu indikator pada indikator sistem, pompa 1, pompa 2, pompa 3, level air *low*, level air *medium*, level air *high*, motor koagulasi cepat, motor koagulasi lambat, pompa vakum, pompa distribusi, dan indikator produksi akan berwarna merah dan textbox masing2 indikator bernilai '0'.

4.2.4. Pengujian Penerimaan Data Dari Interface

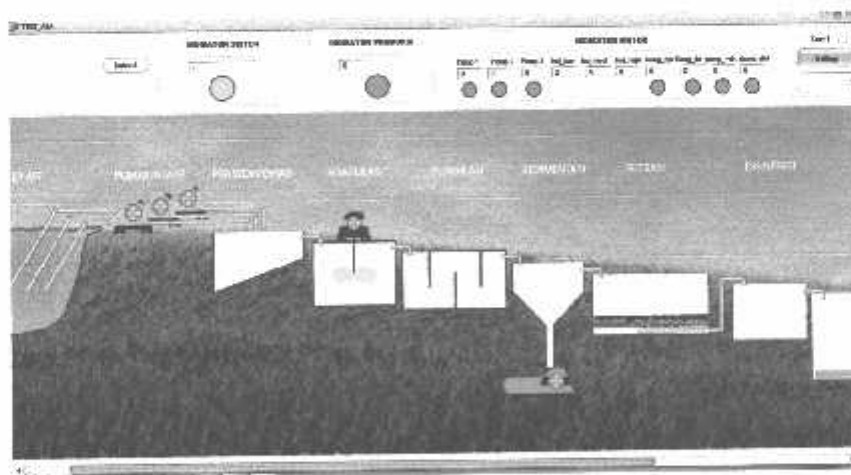
Sebelum melakukan uji coba aplikasi terlebih dahulu dilakukan pengujian penerimaan data karena aplikasi ini melibatkan perangkat luar. Pengujian penerimaan data ini bermaksud untuk memastikan data yang dikirim interface sesuai dengan inisial yang diterima aplikasi. Berikut adalah tahap uji penerimaan data :

1. Indikator Sistem



. Gambar 4.4
pengiriman data indikator sistem

Gambar di atas adalah gambar uji coba pengiriman data secara manual untuk indikator sistem. Sesuai dengan tabel yang dibuat di perancangan, indikator sistem berada di port Port A0.

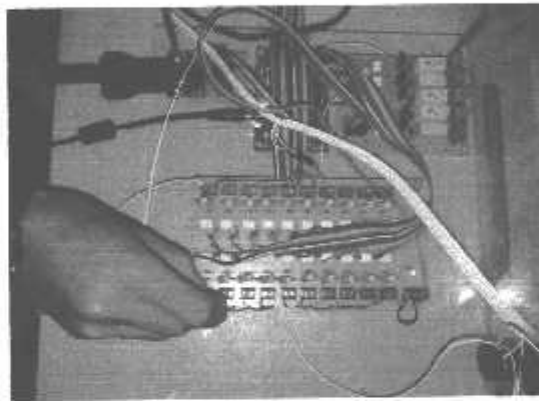


. Gambar 4.5

penerimaan data indikator sistem

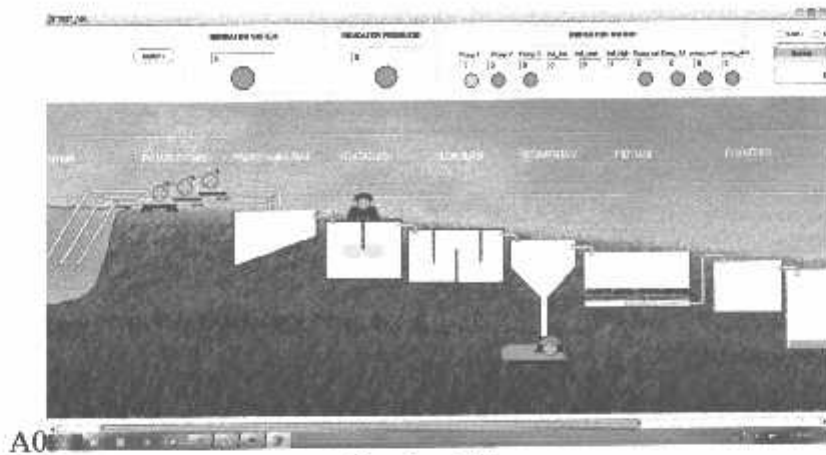
Gambar di atas adalah gambar uji penerimaan data yang di kirim oleh interface. Didalam pembuatan aplikasi ini untuk metode penerimaan data yang dikirim oleh interface menggunakan inisial string dataS0, dataS1, dataS2 ,....., sampai dataS11 sesuai jumlah inputan yang dikirim interface. Untuk indikator sistem menggunakan inisial dataS0. Jika data yang dikirim bernilai '1' maka lampu indikator pada indikator sistem akan berwarna kuning dan textbox akan bernilai sama seperti data yang dikirim oleh interface.

2. Pompa 1



Gambar 4.6
pengiriman data pompa 1

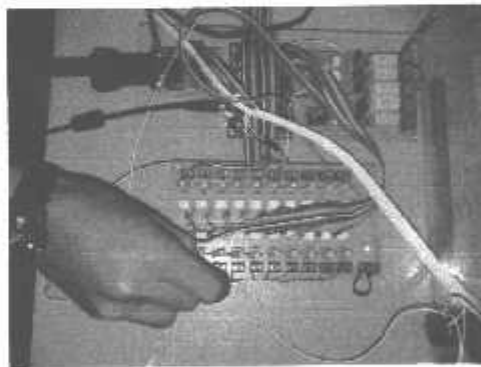
Gambar di atas adalah gambar uji coba pengiriman data secara manual untuk indikator sistem. Sesuai dengan tabel yang dibuat di perancangan, pompa 1 berada di Port A1.



Gambar 4.7
Penerimaan data pompa 1

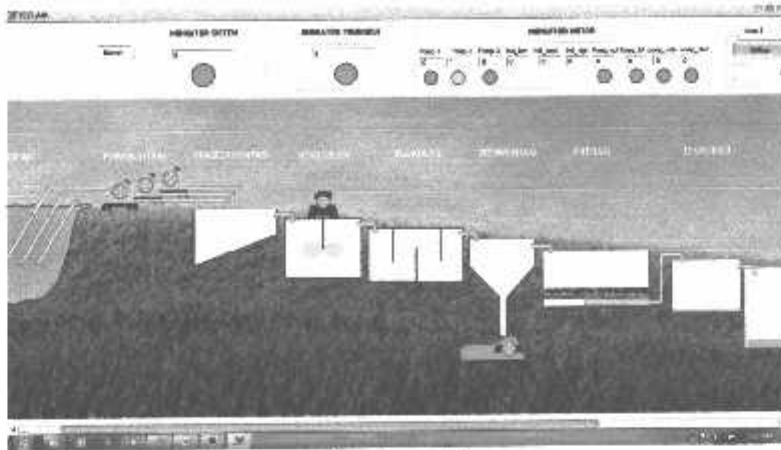
Gambar di atas adalah gambar uji penerimaan data yang di kirim oleh interface. Sama seperti indikator sistem pompa 1 juga menggunakan inisial string untuk penerimaan data. Untuk pompa 1 menggunakan inisial dataS1. Jika data yang dikirim bernilai '1' maka lampu indikator pada pompa 1 akan berwarna kuning dan textboxt akan bernilai sama seperti data yang dikirim oleh interface.

3. Pompa 2



Gambar 4.8
pengiriman data pompa 2

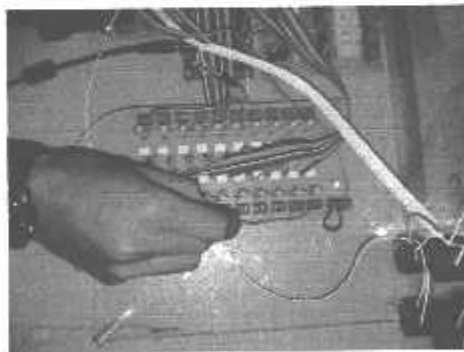
Gambar di atas adalah gambar uji pengiriman data secara manual untuk pompa 2. Sesuai dengan tabel yang dibuat di perancangan, pompa 2 berada di Port A2.



Gambar 4.9
Penerimaan data pompa 2

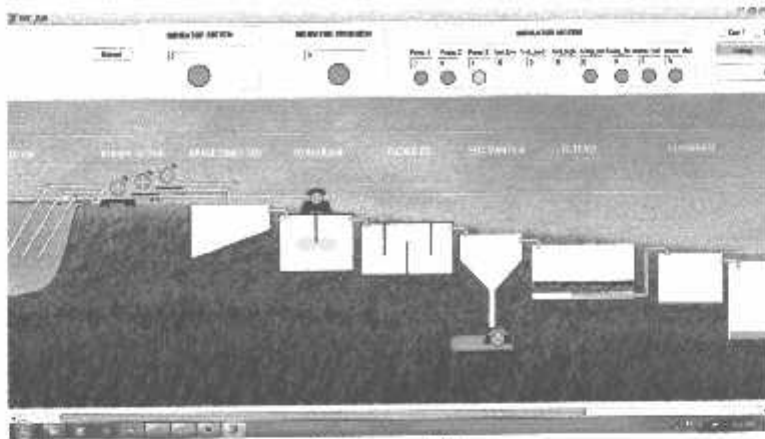
Gambar di atas adalah gambar uji penerimaan data yang di kirim oleh interface. Sama seperti indikator sistem dan pompa 1, pompa 2 juga menggunakan inisial string untuk penerimaan data. Untuk pompa 2 menggunakan inisial dataS2. Jika data yang dikirim bernilai '1' maka lampu indikator pada pompa 2 akan berwarna kuning dan textbox akan bernilai sama seperti data yang dikirim oleh interface.

4. pompa 3



Gambar 4.10
pengiriman data pompa 3

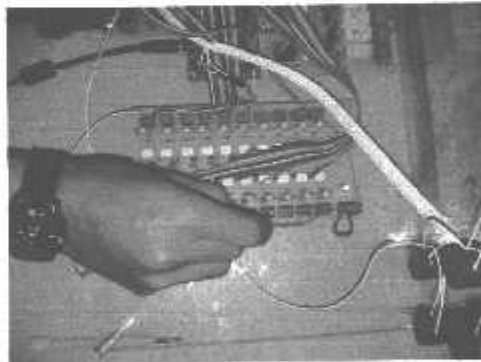
Gambar di atas adalah gambar uji pengiriman data secara manual untuk pompa 3. Sesuai dengan tabel yang dibuat di perancangan, pompa 3 berada di Port A5.



Gambar 4.11
Penerimaan data pompa 3

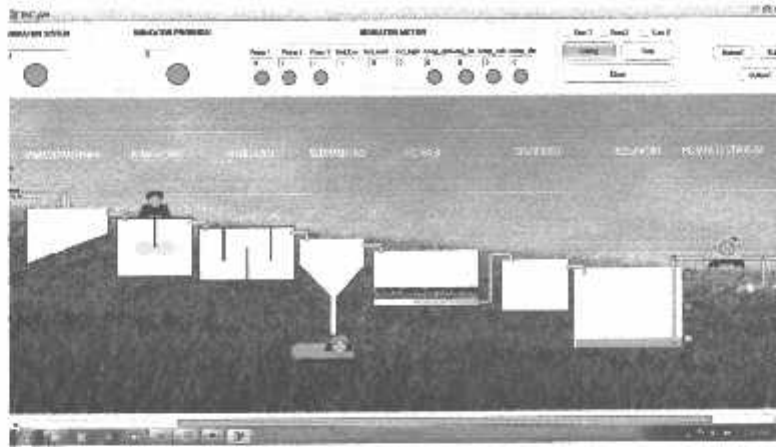
Gambar di atas adalah gambar uji penerimaan data yang di kirim oleh interface. Sama seperti indikator- indikator sebelumnya pompa 3 juga menggunakan inisial string untuk penerimaan data. Untuk pompa 1 menggunakan inisial dataS3.. Jika data yang dikirim bernilai '1' maka lampu indikator pada pompa 3 akan berwarna kuning dan textbox akan bernilai sama seperti data yang dikirim oleh interface..

5. Indikator Level Air *Low*



Gambar 4.12
pengiriman data indikator level air *low*

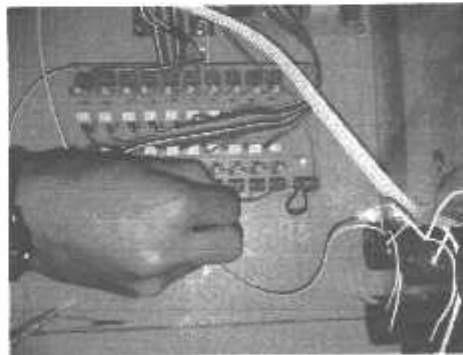
Gambar di atas adalah gambar uji pengiriman data secara manual untuk indikator level air *low*. Sesuai dengan tabel yang dibuat di perancangan, indikator level air *low* berada di Port A6.



Gambar 4.13
penerimaan data indikator level air *low*

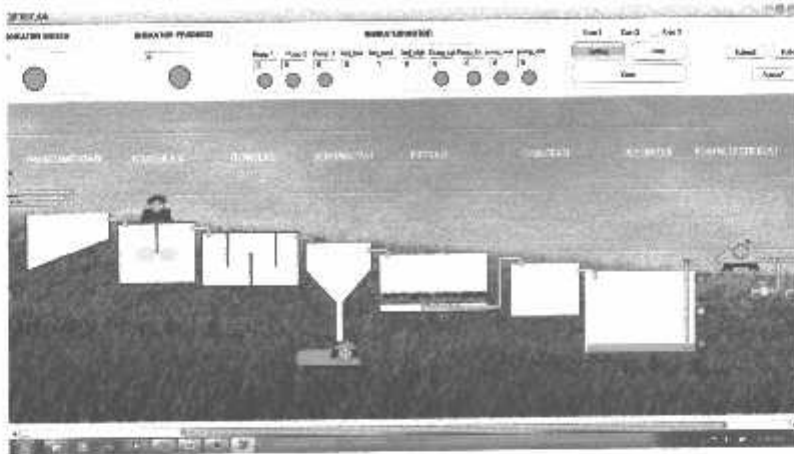
Gambar di atas adalah gambar uji penerimaan data yang di kirim oleh interface. Sama seperti indikator- indikator sebelumnya indikator level air *low* juga menggunakan inisial string untuk penerimaan data. Indikator level air *low* menggunakan inisial dataS4. Jika data yang dikirim bernilai '1' maka lampu indikator pada indikator level air *low* akan berwarna kuning dan textboxt akan bernilai sama seperti data yang dikirim oleh interface.

6. Indikator Level Air *Medium*



Gambar 4.14
pengiriman data indikator level air *medium*

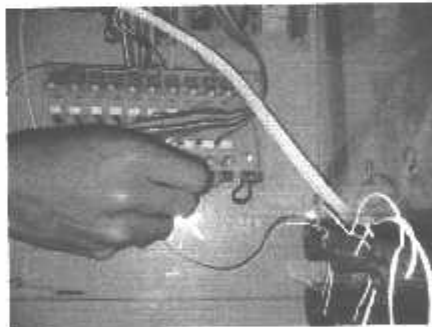
Gambar di atas adalah gambar uji pengiriman data secara manual untuk indikator level air *medium*. Sesuai dengan tabel yang dibuat di perancangan, indikator level air *medium* berada di Port A7.



Gambar 4.15
penerimaan data indikator level air *medium*

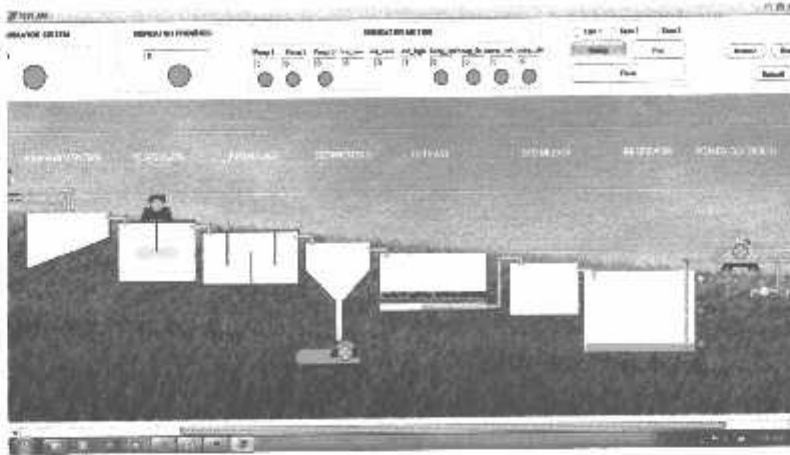
Gambar di atas adalah gambar uji penerimaan data yang di kirim oleh interface. Sama seperti indikator- indikator sebelumnya indikator level air *medium* juga menggunakan inisial string untuk penerimaan data. Indikator level air *medium* menggunakan inisial dataS5. Jika data yang dikirim bernilai '1' maka lampu indikator pada indikator level air *high* akan berwarna kuning dan textbox akan bernilai sama seperti data yang dikirim oleh interface.

7. Indikator Level Air *High*



Gambar 4.16
pengiriman data level air *high*

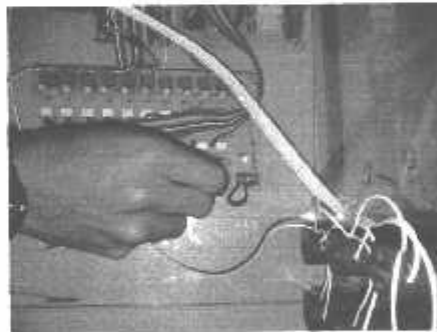
Gambar di atas adalah gambar uji pengiriman data secara manual untuk indikator level air *high*. Sesuai dengan tabel yang dibuat di perancangan, indikator level air *high* berada di Port B0.



Gambar 4.17
penerimaan data level air *high*

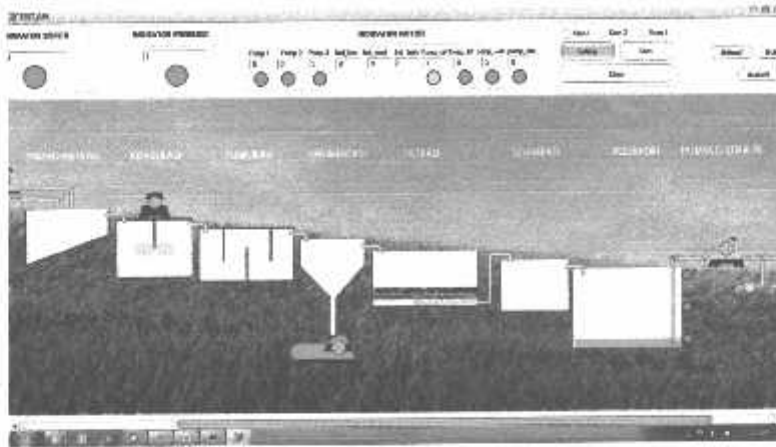
Gambar di atas adalah gambar uji penerimaan data yang di kirim oleh interface. Sama seperti indikator- indikator sebelumnya indikator level air *high* juga menggunakan inisial string untuk penerimaan data. Indikator level air *high* menggunakan inisial dataS6. Jika data yang dikirim bernilai '1' maka lampu indikator pada indikator level air *high* akan berwarna kuning dan textbox akan bernilai sama seperti data yang dikirim oleh interface.

8. Motor Koagulasi Cepat



Gambar 4.18
pengiriman data motor koagulasi cepat

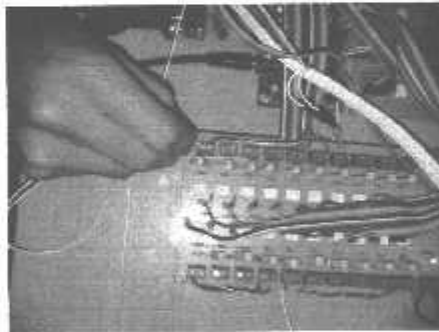
Gambar di atas adalah gambar uji pengiriman data secara manual untuk motor koagulasi cepat. Sesuai dengan tabel yang dibuat di perancangan, imotor koagulasi cepat berada di Port B1.



Gambar 4.19
penerimaan data motor koagulasi cepat

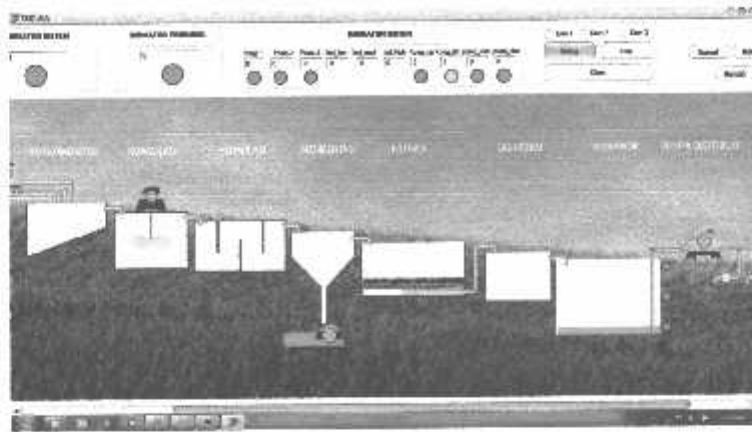
Gambar di atas adalah gambar uji penerimaan data yang di kirim oleh interface. Sama seperti indikator- indikator sebelumnya motor koagulasi cepat juga menggunakan inisial string untuk penerimaan data. Motor koagulasi cepat menggunakan inisial dataS7. Jika data yang dikirim bernilai '1' maka lampu indikator pada motor koagulasi lambat akan berwarna kuning dan textbox akan bernilai sama seperti data yang dikirim oleh interface.

9. Motor Koagulasi Lambat



Gambar 4.20
pengiriman data motor koagulasi lambat

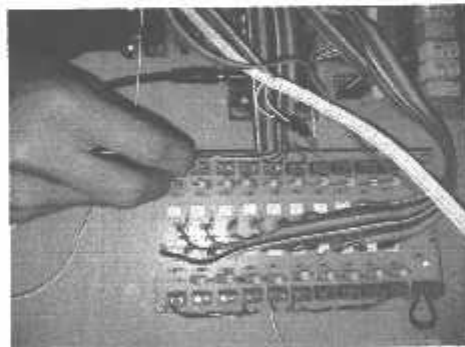
Gambar di atas adalah gambar uji pengiriman data secara manual untuk motor koagulasi lambat. Sesuai dengan tabel yang dibuat di perancangan, imotor koagulasi lambat berada di Port B2.



Gambar 4.21
penerimaan data motor koagulasi lambat

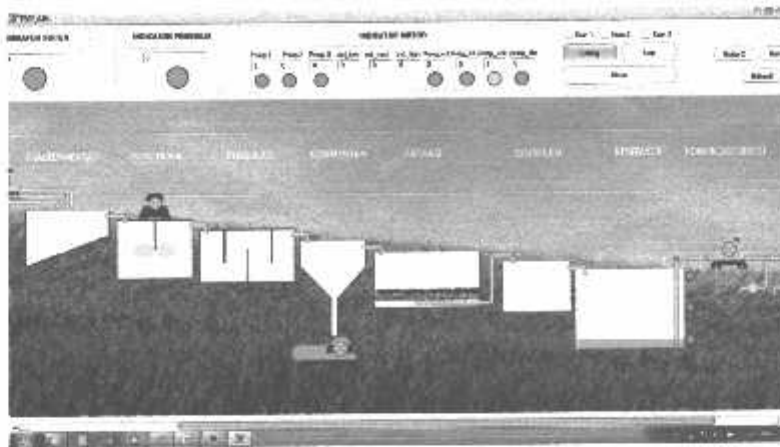
Gambar di atas adalah gambar uji penerimaan data yang di kirim oleh interface. Sama seperti indikator- indikator sebelumnya indikator koagulasi lambat juga menggunakan inisial string untuk penerimaan data. Motor koagulasi lambat menggunakan inisial dataS8. Jika data yang dikirim bernilai '1' maka lampu indikator pada motor koagulasi lambat akan berwarna kuning dan textbox akan bernilai sama seperti data yang dikirim oleh interface.

10. Pompa Vakum



Gambar 4.22
pengiriman data pompa vakum

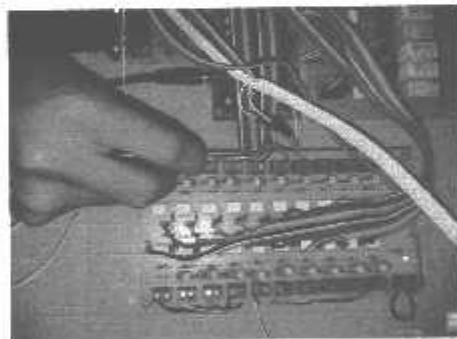
Gambar di atas adalah gambar uji pengiriman data secara manual untuk pompa vakum. Sesuai dengan tabel yang dibuat di perancangan, pompa vakum berada di Port B3.



Gambar 4.23
Penerimaan Data Pompa Vakum

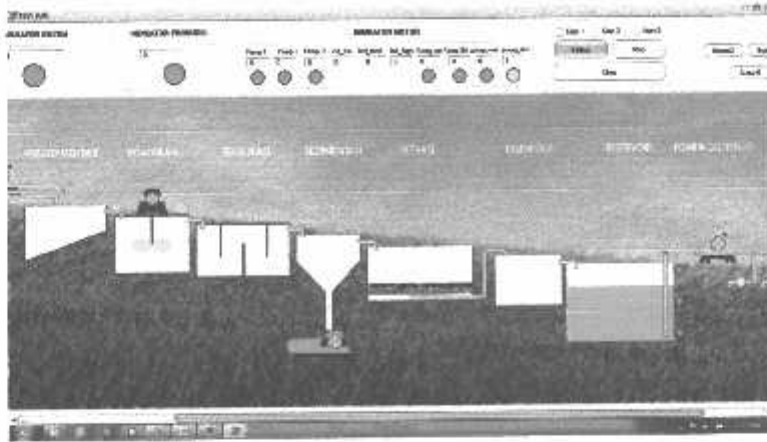
Gambar di atas adalah gambar uji penerimaan data yang di kirim oleh interface. Sama seperti indikator- indikator sebelumnya pompa vakum juga menggunakan inisial string untuk penerimaan data. Pompa vakum menggunakan inisial dataS9. Jika data yang dikirim bernilai '1' maka lampu indikator pada pompa vakum akan berwarna kuning dan textboxt akan bernilai sama seperti data yang dikirim oleh interface.

11. Pompa Distribusi



Gambar 4.24
Pengiriman Data Pompa Distribusi

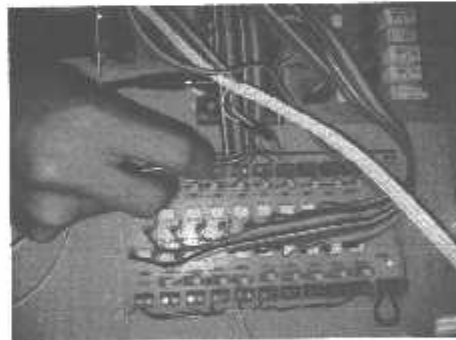
Gambar di atas adalah gambar uji pengiriman data secara manual untuk pompa distribusi. Sesuai dengan tabel yang dibuat di perancangan, pompa distribusi berada di Port B4.



Gambar 4.25
Penerimaan Data Pompa Distribusi

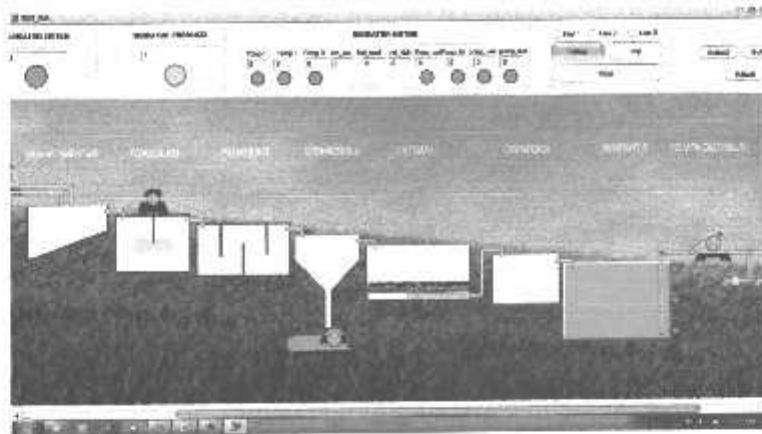
Gambar di atas adalah gambar uji penerimaan data yang di kirim oleh interface. Sama seperti indikator- indikator sebelumnya pompa distribusi juga menggunakan inisial string untuk penerimaan data. Pompa distribusi menggunakan inisial dataS10. Jika data yang dikirim bernilai '1' maka lampu indikator pada pompa distribusi akan berwarna kuning dan textbox akan bernilai sama seperti data yang dikirim oleh interface.

12. Indikator Produksi



Gambar 4.26
Pengiriman Data Indikator Produksi

Gambar di atas adalah gambar uji pengiriman data secara manual untuk indikator produksi. Sesuai dengan tabel yang dibuat di perancangan, indikator produksi berada di Port B5.



Gambar 4.27

Penerimaan Data Pompa Distribusi

Gambar di atas adalah gambar uji penerimaan data yang di kirim oleh interface. Sama seperti indikator- indikator sebelumnya indikator produksi juga menggunakan inisial string untuk penerimaan data. Indikator produksi menggunakan inisial dataS11. Jika data yang dikirim bernilai '1' maka lampu indikator pada indikator produksi akan berwarna kuning dan textboxt akan bernilai sama seperti data yang dikirim oleh interface.

Setelah melakukan pengujian pengiriman data masukan dan keluaran pada modul interface sebelum terhubung dengan modul PLC, maka didapatkan hasil pengambilan data seperti pada table 4.1 dan 4.2.

Tabel 4.1 Pengambilan Data Masukan dari Modul Interface Ke User Interface

No	Port	Status Saat Diberi Logika		Keterangan
		Input (Mikrokontroller)	Output (Software Simulator)	
1	PA0	1	1	Indikator Sistem On
		0	0	
2	PA1	1	1	Pompa Intake 1
		0	0	
3	PA2	1	1	Pompa Intake 2
		0	0	
4	PA5	1	1	Pompa Intake 3
		0	0	
5	PA6	1	1	Indikator <i>Low</i> Reservoir
		0	0	
6	PA7	1	1	Indikator <i>Medium</i>

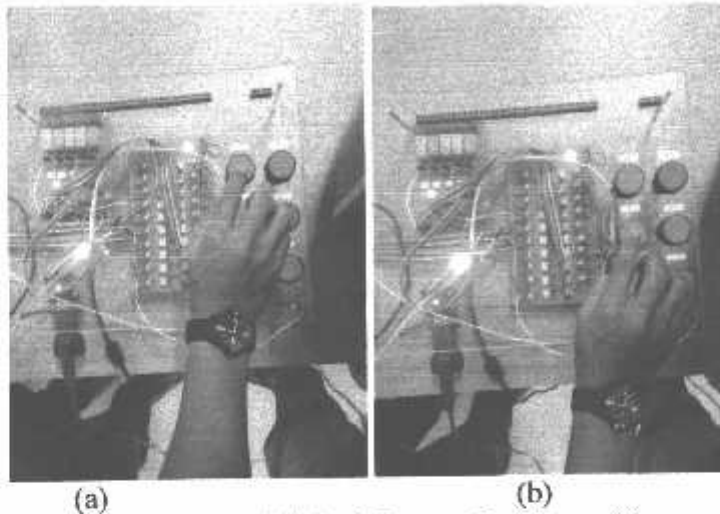
		0	0	Reservoir
7	PB0	1	1	Indikator <i>High</i>
		0	0	Reservoir
8	PB1	1	1	Motor Koagulasi On
		0	0	Kecepatan Tinggi
9	PB2	1	1	Motor Koagulasi On
		0	0	Kecepatan Rendah
10	PB3	1	1	Pompa Vakum On
10	PB3	0	0	Pompa Vakum On
11	PB4	1	1	Pompa Distribusi On
		0	0	
12	PB5	1	1	Indikator Produksi
		0	0	On

Tabel 4.3 Pengambilan Data Keluaran Modul Interface Dari User Interface

No	Kiriman Software Simulator	Status Saat Diberi Logika		Keterangan
		Input (Software Simulator)	Output (Mikrokontroler)	
1	Sensor <i>Low</i> Reservoir	1	1	Port PD2
		0	0	
2	Sensor <i>Medium</i> Reservoir	1	1	Port PD3
		0	0	
3	Sensor <i>High</i> Reservoir	1	1	Port PD4
		0	0	
4	Sensor Koagulasi	1	1	Port PD5
		0	0	
5	Sensor Koagulasi	1	1	Port PD6
		0	0	

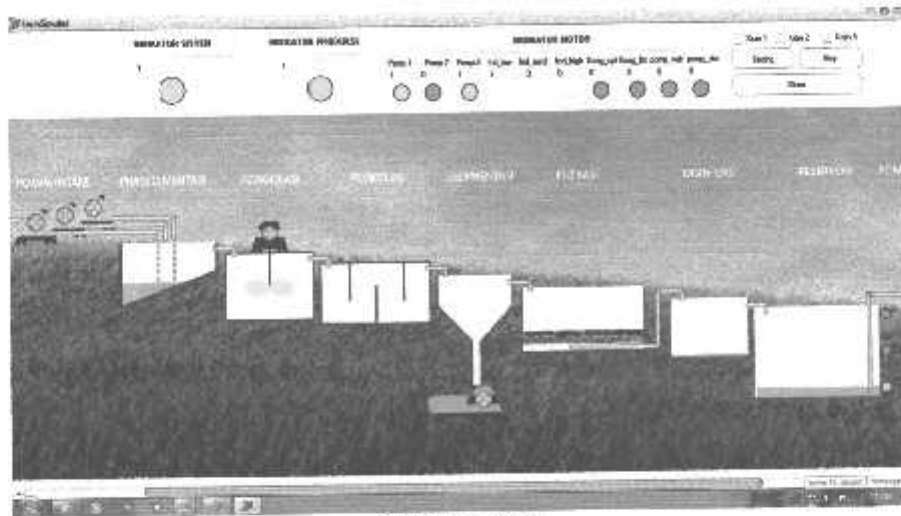
4.2.5. Pengujian sistem keseluruhan

Pada pengujian sistem keseluruhan ini sudah melibatkan modul PLC yang akan mengontrol aplikasi user interface.



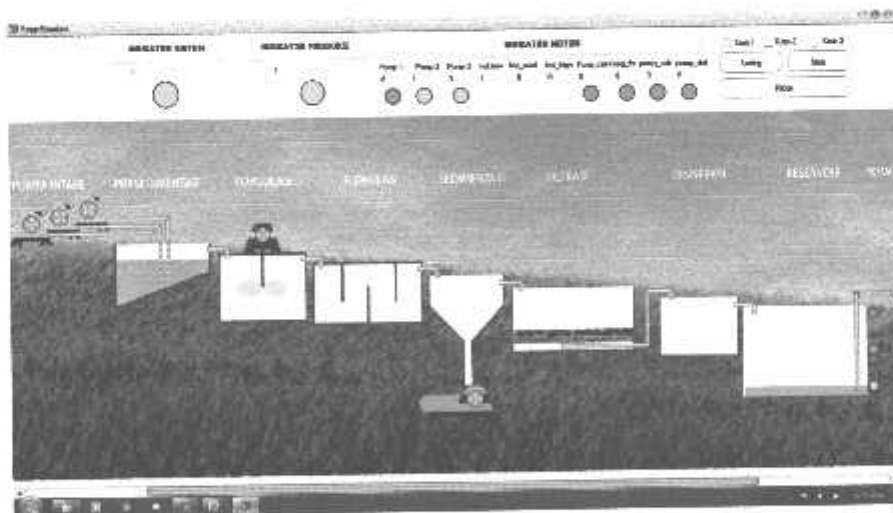
(a)
Gambar 4.28 Kondisi Push Button On Sistem (a)
Kondisi Push Button On Produksi (b)

Setelah melalui proses uji coba manual yang dilakukan pada uji coba sebelumnya, sekarang dapat dilakukan uji coba sistem keseluruhan. Pada tahap uji coba sistem keseluruhan melibatkan modul PLC yang akan mengontrol aplikasi user interface sepenuhnya. Proses pertama untuk menjalankan aplikasi user interface dengan menekan tombol push button ON sistem yang ada di modul PLC. Setelah sistem dinyalakan proses produksi air dapat disimulasikan dengan menekan push button ON produksi. Setelah ON produksi menyala aplikasi user interface akan mengirim sensor level air dalam kondisi *low*. PLC akan memberikan respon dengan menyalakan pompa 1 dan pompa 2.



Gambar 4.29
Kondisi air di level rendah *low*

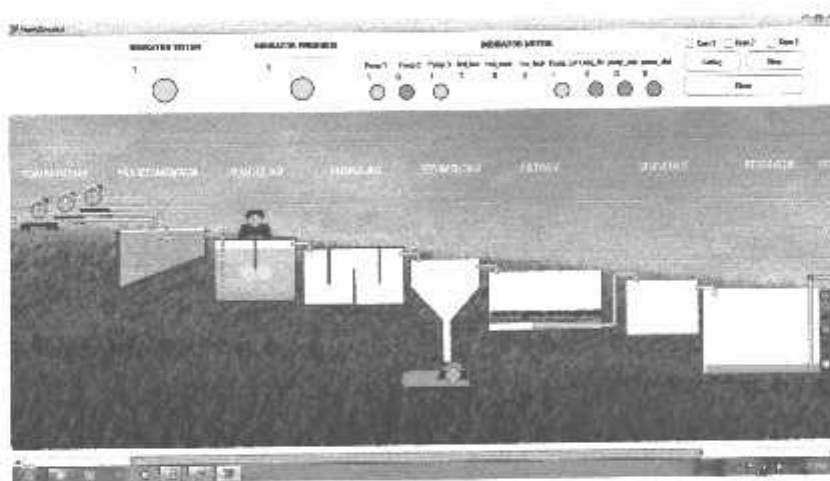
Pada gambar diatas tampak lampu indikator pada indikator sistem dan indikator produksi berwarna kuning, kondisi ini menandakan aplikasi sedang berjalan. Pada lampu indikator level air *low* juga berwarna kuning, ini menandakan air reservoir berada di kondisi level rendah (*low*). Lampu indikator pompa 1 dan pompa 3 berwarna kuning ini juga menandakan pompa sedang bekerja.



Gambar 4.30
Pompa intake menyala bergantian

Pada saat air berada di level rendah (*low*), proses produksi air menggunakan 2 pompa untuk mengalirkan air baku dari sumber air (sungai). Untuk memaksimalkan kinerja setiap pompa modul PLC menggunakan timer untuk mengatur pompa menyala secara bergantian, pada 10 second pertama modul PLC akan akan menyalakan pompa 1 dan 3 dan pada 10 second berikutnya modul PLC akan menyalakan pompa 2 dan 3. Proses ini akan berulang terus sampai air berada di level sedang (*medium*).

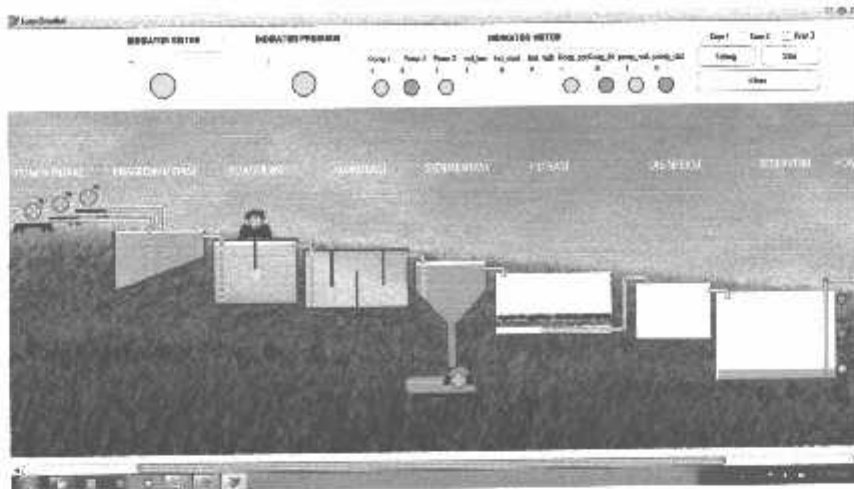
Gambar diatas tampak terlihat indikator pada pompa 2 berwarna kuning dan indikator pompa 1 berwarna merah. Kondisi ini mengindikasikan pompa 1 dan pompa 2 menyala secara bergantian.



Gambar 4.31
Sensor koagulasi menyala

Pada saat kondisi air hampir memenuhi bak koagulasi. Aplikasi user interface akan mengirimkan sensor. Modul PLC akan memberikan respon dengan menyalakan motor koagulasi untuk mengaduk air, pada kondisi ini PLC akan meyalakan motor koagulasi cepat. Motor koagulasi akan bekerja cepat sampai air berada di level sedang (*medium*).

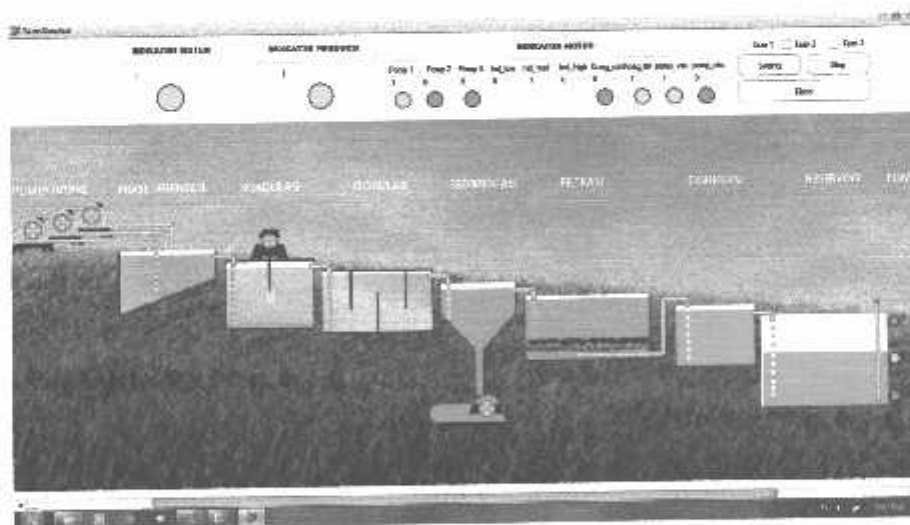
Gambar diatas tampak terlihat lampu indikator motor koagulasi cepat berwarna kuning dan juga textbox bernilai '1', kondisi seperti gambar diatas mengindikasi kalau motor koagulasi sedang berputar cepat.



Gambar 4.32
Sensor sedimentasi

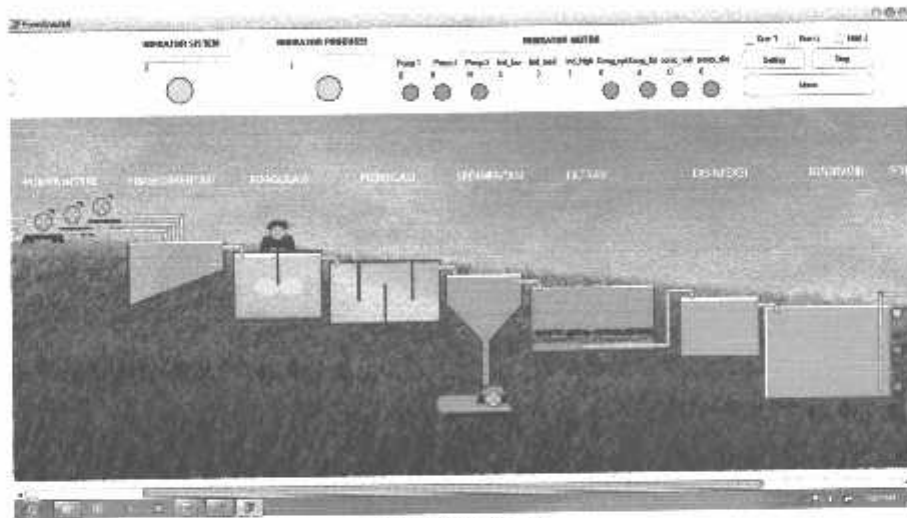
Pada saat air hampir memenuhi bak sedimentasi, aplikasi user interface akan mengirimkan sensor. Modul PLC akan memberikan respon dengan menyalakan pompa vakum. Modul PLC akan Menyalakan pompa vakum setiap 5 second.

Gambar diatas tampak terlihat lampu indikator pada pompa vakum berwarna kuning. Kondisi seperti gambar di atas mngindikasi kalau pompa vakum sedang menyala. proses ini akan berjalan sampai air berada di level tinggi (*high*).



Gambar 4.33
Kondisi air berada di level *medium*

setelah air memasuki bak reservoir disini terdapat tiga sensor yaitu sensor level air rendah (*low*), sensor level air sedang (*medium*), dan sensor level air tinggi (*high*). Gambar di atas tampak terlihat lampu indikator sensor level *medium* berwarna kuning, kondisi tersebut menngindikasikan bahwa air berada di level sedang (*medium*). Pada saat level air berada di *medium* motor koagulasi bekerja menjadi lambat, kondisi ini diindikasikan dengan menyalnya lampu indikator di motor koagulasi lambat.



Gambar 4.34
Kondisi air berada di level tinggi (*high*)

Pada saat kondisi air di level tinggi (*high*), air telah memenuhi bak reservoir. Aplikasi user interface akan mengirim imputan ke PLC untuk mematikan semua motor. Gambar diatas tampak terlihat lampu indikator sensor level air tinggi (*high*) berwarna kuning.



BAB V

PENUTUP

5.1. Deskripsi Umum

Setelah melihat hasil pengujian dari implementasi pemrograman Delphi dalam membuat user interface alat pelatihan PLC dengan studi kasus Water Treatment Plant maka ada beberapa point yang perlu diperhatikan yaitu:

1. Penginisialisasian untuk penerimaan data yang dikirim PLC sesuai dengan data yang dikirimkan PLC.
2. Sensor – sensor buatan yang dikirim aplikasi user interface dapat di terima dengan baik oleh PLC, sehingga respon yang dikirimkan PLC tepat.
3. Ketepatan dalam penentuan interval timer untuk proses *load* gambar sangat penting karena sangat berpengaruh pada proses simulasi.
4. Diharapkan aplikasi user interface alat pelatihan PLC dengan studi kasus Water Treatment Plant dapat digunakan dengan baik untuk pelatihan PLC sebagai pengganti miniatur atau prototipe plant.

5.2. Saran

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

1. Aplikasi user interface ini dirasa masih jauh dari kesempurnaan, masih banyak yang dapat dikembangkan dari segi tampilan, karena masih ada instrument yang belum menyerupai plant yang sebenarnya.

2. Bila dilakukan pengembangan sebaiknya menggunakan instrument yang dapat dihitung nilainya, sehingga debit air yang ada di setiap bak dapat dihitung nilainya..



DAFTAR PUSTAKA

- [1] Wahana Komputer. 2009. *Buku Panduan Aplikatif & Solusi (PAS) : Aplikasi Cerdas Menggunakan Delphi*. Yogyakarta:ANDI
- [2] *Pemrograman Borland Delphi 7 (jilid 1)*. Yogyakarta:ANDI.
- [3] PDAM Bandarmasih Kota Banjarmasin, *Diklat Manajemen Air Minum Berbasih Kompetensi Tingkat Muda Volume: 2*
- [4] Hanif Said, 2012, *Aplikasi Programmable Logic Controller (PLC) Dan Sistem Pneumatik Pada Manufaktur Industri*, Yogyakarta : CV Andi Offset (ANDI)
- [5] Satrya, Juni. 2012. *Rancang Bangun Aplikasi Monitoring Dan Kendali Terpusat Lampu Penerangan Gedung Teknik Elektro ITN Malang*. Malang
- [6] Mulwinda, Anggraini. *Simulasi Sistem Pengolahan Air Di Instalasi Pengolahan Air Kudu Semarang*. [PDF], (<http://eprints.undip.ac.id/25591/1/ML2F097612.pdf>, diakses tanggal 14 Maret 2013)
- [7] http://id.wikipedia.org/wiki/Graphics_Interchange_Format
- [8] www.google.com





LAMPIRAN

LAMPIRAN

Listing Program

```
unit ParsingData;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, CPort, ExtCtrls, VrControls, VrTank, jpeg, GIFImage, sCheckBox,
sButton, sSkinManager, XPMan;

type
TTEST_AJA = class(TForm)
    ComPort1: TComPort;
    ComDataPacket1: TComDataPacket;
    Timer_sensorLevel: TTimer;
    Timer_koagulasi: TTimer;
    Timer_sedimentasi: TTimer;
    GroupBox2: TGroupBox;
    sB_setting: TsButton;
    sCb_kran1: TsCheckBox;
    sB_stop: TsButton;
    GroupBox3: TGroupBox;
    Label13: TLabel;
    Edat1: TEdit;
    Panel1: TPanel;
```

Label1: TLabel;
Label4: TLabel;
Label5: TLabel;
Label6: TLabel;
Label7: TLabel;
Edat2: TEdit;
Edat3: TEdit;
Edat4: TEdit;
Label8: TLabel;
Label9: TLabel;
Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
Shap_pomp1: TShape;
Shap_pomp2: TShape;
Shap_pomp3: TShape;
Shap_koag_lbi: TShape;
Shap_koag_cpt: TShape;
Shap_pomp_vak: TShape;
Shap_pomp_dis: TShape;
GroupBox4: TGroupBox;
Shap_produk: TShape;
Panel4: TPanel;
Label14: TLabel;
Edat11: TEdit;
GroupBox1: TGroupBox;
ImagePrasedimentasi: TImage;

```
Shape_level_medium: TShape;
Shape_level_high: TShape;
Image_kran1: TImage;
Image_kran2: TImage;
Image_kran3: TImage;
GroupBox5: TGroupBox;
Panel3: TPanel;
Label3: TLabel;
Edat0: TEdit;
Shap_sist: TShape;
sCb_kran2: TsCheckBox;
Edat5: TEdit;
Edat6: TEdit;
Edat7: TEdit;
Edat8: TEdit;
Edat9: TEdit;
Edat10: TEdit;
sCb_kran3: TsCheckBox;
Tm_progres1: TTimer;
Tm_progres2: TTimer;
Tm_progres3: TTimer;
Tm_progres4: TTimer;
Tm_progres5: TTimer;
Tm_progres6: TTimer;
Tm_progres7: TTimer;
XPManifest1: TXPManifest;
procedure ComDataPacket1Packet(Sender: TObject; const Str: String);
```

```
procedure FormCreate(Sender: TObject);
procedure Timer_sensorLevelTimer(Sender: TObject);
procedure Timer_koagulasiTimer(Sender: TObject);
procedure Timer_sedimentasiTimer(Sender: TObject);
procedure sB_settingClick(Sender: TObject);
procedure sB_stopClick(Sender: TObject);
procedure indikator_Sistem(Sender: TObject);
procedure indikator_Produksi(Sender: TObject);
procedure indikator_low (Sender : TObject);
procedure indikator_Medium(Sender: TObject);
procedure indikator_High(Sender: TObject);
procedure Pompa_1(Sender: TObject);
procedure Pompa_2(Sender: TObject);
procedure Pompa_3(Sender: TObject);
procedure Koagulasi_lambat(Sender: TObject);
procedure Koagulasi_cepat(Sender: TObject);
procedure Pompa_vakum(Sender: TObject);
procedure Pompa_distribusi(Sender: TObject);
procedure Tm_progres1Timer(Sender: TObject);
procedure Tm_progres2Timer(Sender: TObject);
procedure Tm_progres3Timer(Sender: TObject);
procedure Tm_progres4Timer(Sender: TObject);
procedure Tm_progres5Timer(Sender: TObject);
procedure Tm_progres6Timer(Sender: TObject);
procedure Tm_progres7Timer(Sender: TObject);
procedure FormActivate(Sender: TObject);
private
```

```

{ Private declarations }
dataIN : string;

//Ini variabel penampung data
Ind_sist, Ind_Prod, Ind_low, Ind_med, Ind_high, Pomp_1, Pomp_2, Pomp_3,
Mot_koaglow, Mot_koaghigh, Pomp_vak, Pomp_dis : integer;

public
{ Public declarations }
end;

var
TEST_AJA: TTEST_AJA;
//variabel data buffer
datS0, datS1, datS2, datS3, datS4, datS5, datS6, datS7, datS8, datS9, datS10, datS11:
string;
sendBuffer : string;
adi:String;

implementation

{$R *.dfm}

procedure TTEST_AJA.FormCreate(Sender: TObject);
begin
ImagePrasedimentasi.Picture.LoadFromFile('prasedimentasi/prasedimentasi.gif');
ImageKoagulasiasi.Picture.LoadFromFile('koagulasi/koagulasi.gif');
ImageFlokulasi.Picture.LoadFromFile('flokulasi/flokulasi.gif');

```

```
ImageSedimentasi.Picture.LoadFromFile('sedimentasi/sedimentasi.gif');
ImageFiltrasi.Picture.LoadFromFile('filter/filter.gif');
ImageDisinfeksi.Picture.LoadFromFile('disinfeksi/disinfeksi.gif');

setlength(sendBuffer,6);

sendBuffer:='00000';

end;

procedure TTEST_AJA.indikator_Sistem(Sender: TObject);
begin
  if Ind_sist = 1 then
    begin
      Shap_sist.Brush.Color:=clYellow;
    end;
  if Ind_sist = 0 then
    begin
      Shap_sist.Brush.Color:=clRed;
    end;
end;

procedure TTEST_AJA.indikator_Produksi(Sender: TObject);
begin
  if Ind_Prod=1 then
    begin
      if adi='ON' then
```

```
begin
  Shap_Produk.Brush.Color:=clYellow;
  Tm_progres1.Enabled:=True;
  Timer_sensorLevel.Enabled:=True;
  ImagePrasedimentasi.Picture.LoadFromFile('prasedimentasi/prasedimentasi-1.gif');
  Timer_koagulasi.Enabled:=True;
  Timer_sedimentasi.Enabled:=True;
  adi:="";
end;

end;

if Ind_Prod=0 then
  begin
    Shap_Produk.Brush.Color:=clRed;
  end;
end;

Procedure TTEST_AJA.indikator_low(Sender: TObject);
begin
  if Ind_low=1 then
    begin
      Shape_level_low.Brush.Color:=clYellow;
    end;
  if Ind_low=0 then
    begin
      Shape_level_low.Brush.Color:=clRed;
    end;
  end;
end;
```

```
procedure TTEST_AJA.indikator_Medium(Sender: TObject);
begin
  if Ind_med = 1 then
  begin
    Shape_level_medium.Brush.Color:=clYellow;
    ImagePrasedimentasi.Picture.LoadFromFile('prasedimentasi/prasedimentasi-2.gif');
    ImageKoagulasiasi.Picture.LoadFromFile('koagulasi/koagulasi-2.gif');
    ImageFlokulasi.Picture.LoadFromFile('flokulasi/flokulasi-2.gif');
    ImageSedimentasi.Picture.LoadFromFile('sedimentasi/sedimentasi-2.gif');
    ImageFiltrasi.Picture.LoadFromFile('filter/filter-2.gif');
    ImageDisinfeksi.Picture.LoadFromFile('disinfeksi/disinfeksi-2.gif');
    ImageMengalir1.Visible:=True;
    ImageMengalir2.Visible:=True;
    ImageMengalir3.Visible:=True;
    Image_mengalir_koagulasi.Visible:=True;
    Image_mengalir_flokulasi.Visible:=True;
    Image_mengalir_sedimentasi.Visible:=True;
    Image_mengalir_filter.Visible:=True;
    Image_mengalir_disinfeksi.Visible:=True;
    Image_mengalir_reservoir.Visible:=True;
    ImageMotor.Visible:=True;
    ImageBaling2.Visible:=True;
    ImagePompaVakum.Visible:=True;
  end;
  if Ind_med = 0 then
  begin
```

```

    Shape_level_medium.Brush.Color:=clRed;
end;
end;

procedure TTEST_AJA.indikator_High(Sender: TObject);
begin
    if Ind_high = 1 then
    begin
        Shape_level_high.Brush.Color:=clYellow;
        ImagePompa3.Visible:=False;
        ImagePompa1.Visible:=False;
        ImageMengalir3.Visible:=False;
        ImageMengalir1.Visible:=False;
        ImagePrasedimentasi.Picture.LoadFromFile('prasedimentasi/prasedimentasi-
        penuh.gif');
        ImageKoagulasilasi.Picture.LoadFromFile('koagulasi/koagulasi-penuh.gif');
        ImageFlokulasi.Picture.LoadFromFile('flokulasi/flokulasi-penuh.gif');
        ImageSedimentasi.Picture.LoadFromFile('sedimentasi/sedimentasi-penuh.gif');
        ImageFiltrasi.Picture.LoadFromFile('filter/filter-penuh.gif');
        ImageDisinfeksi.Picture.LoadFromFile('disinfeksi/disinfeksi-penuh.gif');
        ImageMengalir1.Visible:=False;
        ImageMengalir2.Visible:=False;
        ImageMengalir3.Visible:=False;
        Image_mengalir_koagulasi.Visible:=False;
        Image_mengalir_flokulasi.Visible:=False;
        Image_mengalir_sedimentasi.Visible:=False;
        Image_mengalir_filter.Visible:=False;
        Image_mengalir_disinfeksi.Visible:=False;
    end;
end;

```

```

Image_mengalir_reservoir.Visible:=False;
ImageMotor.Visible:=False;
ImageBaling2.Visible:=False;
ImagePompaVakum.Visible:=False;
end;
if Ind_high = 0 then
begin
Shape_level_high.Brush.Color:=clRed;
end;
end;

procedure TTEST_AJA.Pompa_1(Sender: TObject);
begin
if Pomp_1 = 1 then
begin
Shap_pomp1.Brush.Color:=clYellow;
ImagePompa1.Visible:=True;
ImageMengalir1.Visible:=True;
ImagePompa1.Picture.LoadFromFile('prasedimentasi/pompa-1.gif');
ImageMengalir1.Picture.LoadFromFile('prasedimentasi/mengalir.gif');
end;
if Pomp_1 = 0 then
begin
Shap_pomp1.Brush.Color:=clRed;
ImagePompa1.Visible:=False;
ImageMengalir1.Visible:=False;
end;
end;

```

end;

procedure TTEST_AJA.Pompa_2(Sender: TObject);

begin

if Pomp_2 = 1 then

begin

Shap_pomp2.Brush.Color:=clYellow;

ImagePompa2.Visible:=True;

ImageMengalir2.Visible:=True;

ImagePompa2.Picture.LoadFromFile('prasedimentasi/pompa-2.gif');

ImageMengalir2.Picture.LoadFromFile('prasedimentasi/mengalir.gif');

end;

if Pomp_2 = 0 then

begin

Shap_pomp2.Brush.Color:=clRed;

ImagePompa2.Visible:=False;

ImageMengalir2.Visible:=False;

end;

end;

procedure TTEST_AJA.Pompa_3(Sender: TObject);

begin

if Pomp_3 = 1 then

begin

Shap_pomp3.Brush.Color:=clYellow;

ImagePompa3.Visible:=True;

ImageMengalir3.Visible:=True;

```

    ImagePompa3.Picture.LoadFromFile('prasedimentasi/pompa-3.gif');
    ImageMengalir3.Picture.LoadFromFile('prasedimentasi/mengalir.gif');
end;
if Pomp_3 = 0 then
begin
    Shap_pomp3.Brush.Color:=clRcd;
    ImagePompa3.Visible:=False;
    ImageMengalir3.Visible:=False;
end;
end;

procedure TTEST_AJA.Koagulasi_Jambat(Sender: TObject);
begin
    if Mot_koagLow = 1 then
begin
    Shap_koag_lbt.Brush.Color:=clYellow;
    ImageMotor.Visible:=True;
    ImageBaling2.Visible:=True;
    ImageMotor.Picture.LoadFromFile('koagulasi/Motor-Koagulasi3.gif');
    ImageBaling2.Picture.LoadFromFile('koagulasi/Baling3.gif');
end;
    if Mot_koagLow = 0 then
begin
    Shap_koag_lbt.Brush.Color:=clRcd;
//    ImageMotor.Visible:=False;
//    ImageBaling2.Visible:=False;
end;
end;

```

end;

procedure TTEST_AJA.Koagulasi_cepat(Sender: TObject);

begin

 if Mot_koagHigh = 1 then

 begin

 Shap_koag_cpt.Brush.Color:=clYellow;

 ImageMotor.Visible:=True;

 ImageBaling2.Visible:=True;

 ImageMotor.Picture.LoadFromFile('koagulasi/Motor-Koagulasi.gif');

 ImageBaling2.Picture.LoadFromFile('koagulasi/Baling2.gif');

 end;

 if Mot_koagHigh = 0 then

 begin

 Shap_koag_cpt.Brush.Color:= clRed;

 // ImageMotor.Visible:=False;

 // ImageBaling2.Visible:=False;

 end;

end;

procedure TTEST_AJA.Pompa_vakum(Sender: TObject);

begin

 if Pomp_vak = 1 then

 begin

 Shap_pomp_vak.Brush.Color:=clYellow;

 ImagePompaVakum.Visible:=True;

 ImagePompaVakum.Picture.LoadFromFile('sedimentasi/pompa-vakum.gif');

```

end;
if Pomp_vak = 0 then
begin
    Shap_pomp_vak.Brush.Color:= clRed;
    ImagePompaVakum.Visible:=False;
end;
end;

procedure TTEST_AJA.Pompa_distribusi(Sender: TObject);
begin
    if Pomp_dis = 1 then
        begin
            Shap_pomp_dist.Brush.Color:=clYellow;
            ImagePompaDistribusi.Visible:=True;
            ImagePompaDistribusi.Picture.LoadFromFile('reservoir/pompa-distribusi.gif');
        end;
    if Pomp_dis = 0 then
        begin
            Shap_pomp_dist.Brush.Color:= clRed;
            ImagePompaDistribusi.Visible:=False;
        end;
    end;

    procedure TTEST_AJA.ComDataPacket1Packet(Sender: TObject; const Str: String);
    var i:integer;
        error : integer;
    begin

```

```
dataIN := Str;
```

```
// Reset data
```

```
datS0:=' ';
```

```
datS1:=' ';
```

```
datS2:=' ';
```

```
datS3:=' ';
```

```
datS4:=' ';
```

```
datS5:=' ';
```

```
datS6:=' ';
```

```
datS7:=' ';
```

```
datS8:=' ';
```

```
datS9:=' ';
```

```
datS10:=' ';
```

```
datS11:=' ';
```

```
for i:-1 to length(dataIN) do
```

```
begin
```

```
  if dataIN[i]=';' then break;
```

```
  datS0:=datS0+dataIN[i];
```

```
end;
```

```
for i:i+1 to length(dataIN) do
```

```
begin
```

```
  if dataIN[i]=';' then break;
```

```
  datS1:=datS1+dataIN[i];
```

```
end;
```

```
for i:-i+1 to length(dataIN) do
begin
  if dataIN[i]=';' then break;
  datS2:=datS2+dataIN[i];
end;
```

```
for i:=i+1 to length(dataIN) do
begin
  if dataIN[i]=';' then break;
  datS3:=datS3+dataIN[i];
end;
```

```
for i:-i+1 to length(dataIN) do
begin
  if dataIN[i]=';' then break;
  datS4:=datS4+dataIN[i];
end;
```

```
for i:=i+1 to length(dataIN) do
begin
  if dataIN[i]=';' then break;
  datS5:=datS5+dataIN[i];
end;
```

```
for i:-i+1 to length(dataIN) do
begin
```

```
    if dataIN[i]~';' then break;
    datS6:=datS6+dataIN[i];
end;
```

```
for i:=i+1 to length(dataIN) do
begin
    if dataIN[i]~';' then break;
    datS7:=datS7+dataIN[i];
end;
```

```
for i:=i+1 to length(dataIN) do
begin
    if dataIN[i]~';' then break;
    datS8:=datS8+dataIN[i];
end;
```

```
for i:=i+1 to length(dataIN) do
begin
    if dataIN[i]~';' then break;
    datS9:=datS9+dataIN[i];
end;
```

```
for i:=i+1 to length(dataIN) do
begin
    if dataIN[i]~';' then break;
    datS10:=datS10+dataIN[i];
end;
```

```
for i:-i+1 to length(dataIN) do
begin
    if dataIN[i]=';' then break;
    datS11:=datS11+dataIN[i];
end;
```

```
//assign value
```

```
Edat0.Text:=datS0;
```

```
Edat1.Text:=datS1;
```

```
Edat2.Text:=datS2;
```

```
Edat3.Text:=datS3;
```

```
Edat4.Text:=datS4;
```

```
Edat5.Text:=datS5;
```

```
Edat6.Text:=datS6;
```

```
Edat7.Text:=datS7;
```

```
Edat8.Text:=datS8;
```

```
Edat9.Text:=datS9;
```

```
Edat10.Text:=datS10;
```

```
Edat11.Text:=datS11;
```

```
// Konversi dari variabel string ke integer
```

```
val(datS0,Ind_sist,error);
```

```
val(datS1,Pomp_1,error);
```

```
val(datS2,Pomp_2,error);
```

```
val(datS3,Pomp_3,error);
```

```
val(datS4,Ind_low,error);
```

```
val(datS5,Ind_med,error);
val(datS6,Ind_high,error);
val(datS7,Mot_koagHigh,error);
val(datS8,Mot_koagLow,error);
val(datS9,Pomp_vak,error);
val(datS10,Pomp_dis,error);
val(Edat11.Text,Ind_Prod,error);
```

```
indikator_Sistem(Sender);
indikator_Produksi(Sender);
indikator_Medium(Sender);
indikator_High(Sender);
Pompa_1(Sender);
Pompa_2(Sender);
Pompa_3(Sender);
Koagulasi_lambat(Sender);
Koagulasi_cepat(Sender);
Pompa_vakum(Sender);
Pompa_distribusi(Sender);
indikator_low(Sender);
```

```
end;
```

```
procedure TTEST_AJA.Timer_sensorLevelTimer(Sender: TObject);
```

```
begin
```

```
  if (VrTank1.Position > 34 ) and (VrTank1.Position < 64 ) then
```

```
  begin
```

```
    sendBuffer[1] := '1';
    sendBuffer[2] := '0';
    sendBuffer[3] := '0';
end;
// sendBuffer[6]:='!';
comport1.WriteStr(sendBuffer);

if (VrTank1.Position > 65) and (VrTank1.Position < 99 ) then
begin
sendBuffer[2] := '1';
sendBuffer[3] := '0';
end;
// sendBuffer[6]:='!';
comport1.WriteStr(sendBuffer);

if VrTank1.Position =100 then
begin
sendBuffer[3] := '1';
end ;
// sendBuffer[6]:='!';
comport1.WriteStr(sendBuffer);
end;

procedure TTEST_AJA.Timer_koagulasiTimer(Sender: TObject);
begin
sendBuffer[4] := '1';
```

```
    sendBuffer[6]:='';  
    comport1.WriteStr(sendBuffer);  
end;  
  
procedure TTEST_AJA.Timer_sedimentasiTimer(Sender: TObject);  
begin  
    sendBuffer[5] := '1';  
  
    sendBuffer[6]:='';  
    comport1.WriteStr(sendBuffer);  
end;  
  
procedure TTEST_AJA.sB_settingClick(Sender: TObject);  
begin  
    ComPort1.ShowSetupDialog;  
    comport1.Open;  
    ComPort1.Connected:=true;  
  
    comport1.WriteStr(sendBuffer);  
end;  
  
procedure TTEST_AJA.sB_stopClick(Sender: TObject);  
begin  
    ComPort1.Connected:=False;  
    ComPort1.Close;  
    Application.Terminate;  
end;
```

```
procedure TTEST_AJA.Tm_progres1Timer(Sender: TObject);
begin
  ImagePrasedimentasi.Picture.LoadFromFile('prasedimentasi/prasedimentasi-2.gif');
  Tm_progres2.Enabled := True;
  imageKoagulasilasi.Picture.LoadFromFile('koagulasi/koagulasi-1.gif');
  Image_mengalir_koagulasi.Picture.LoadFromFile('koagulasi/mengalir-Koagulasi.gif');

  Tm_progres1.Enabled := False;
end;
```

```
procedure TTEST_AJA.Tm_progres2Timer(Sender: TObject);
begin
  ImageKoagulasilasi.Picture.LoadFromFile('koagulasi/koagulasi-2.gif');

  Tm_progres3.Enabled:=True;
  Imageflokulasi.Picture.LoadFromFile('flokulasi/flokulasi-1.gif');
  Image_mengalir_flokulasi.Picture.LoadFromFile('flokulasi/mengalir-Flokulasi.gif');

  Tm_progres2.Enabled:=False;
end;
```

```
procedure TTEST_AJA.Tm_progres3Timer(Sender: TObject);
begin
  ImageFlokulasi.Picture.LoadFromFile('flokulasi/flokulasi-2.gif');

  Tm_progres4.Enabled :=True;
```

```

ImageSedimentasi.Picture.LoadFromFile('sedimentasi/sedimentasi-1.gif');
Image_mengalir_sedimentasi.Picture.LoadFromFile('sedimentasi/mengalir-
sedimentasi.gif');

Tm_progres3.Enabled:=False;
end;

procedure TTEST_AJA.Tm_progres4Timer(Sender: TObject);
begin
ImageSedimentasi.Picture.LoadFromFile('sedimentasi/sedimentasi-2.gif');

Tm_progres5.Enabled:=True;
ImageFiltrasi.Picture.LoadFromFile('filter/filter-1.gif');
Image_mengalir_filter.Picture.LoadFromFile('filter/mengalir-filter.gif');

Tm_progres4.Enabled:=False;
end;

procedure TTEST_AJA.Tm_progres5Timer(Sender: TObject);
begin
ImageFiltrasi.Picture.LoadFromFile('filter/filter-2.gif');

Tm_progres6.Enabled:=True;
ImageDisinfeksi.Picture.LoadFromFile('disinfeksi/disinfeksi-1.gif');
Image_mengalir_disinfeksi.Picture.LoadFromFile('disinfeksi/mengalir-disinfeksi.gif');

Tm_progres5.Enabled:=False;

```

```
end;
```

```
procedure TTEST_AJA.Tm_progres6Timer(Sender: TObject);
```

```
begin
```

```
    ImageDisinfeksi.Picture.LoadFromFile('disinfeksi/disinfeksi-2.gif');
```

```
    Tm_progres7.Enabled:=True;
```

```
    Image_mengalir_reservoir.Picture.LoadFromFile('reservoir/mengalir-reservoir.gif');
```

```
    Tm_progres6.Enabled:=False;
```

```
end;
```

```
procedure TTEST_AJA.Tm_progres7Timer(Sender: TObject);
```

```
begin
```

```
    if (VrTank1.Position >=35) and (VrTank1.Position <=65) then
```

```
        begin
```

```
            VrTank1.Position:=VrTank1.Position +4;
```

```
        end else
```

```
            VrTank1.Position:=VrTank1.Position +2;
```

```
    if Pomp_dis = 1 then
```

```
        begin
```

```
            if sCb_kran1.Checked then
```

```
                begin
```

```
                    Image_kran1.Visible:=True;
```

```
//                    ImagePompaDistribusi.Visible:=True;
```

```
//                    ImagePompaDistribusi.Picture.LoadFromFile('reservoir/pompa-distribusi.gif');
```



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 : ADI SETYAWAN
 NIM : 09.12.522
 JURUSAN : Teknik Elektro S-1
 KONSENTRASI : Teknik Komputer
 MASA BIMBINGAN: SEMESTER GENAP 2012/2013
 JUDUL : **PEMBUATAN USER INTERFACE ALAT PELATIHAN PLC
 DENGAN STUDI KASUS WATER TREATMENT PLANT**

Tanggal	Uraian	Paraf
Penguji I 16-08-2013	<ol style="list-style-type: none"> Bab 3 perancangan pembuatan animasi dan Inisialisa nya harus jelas Bab 4 gambar-gambar diganti warna seperti visualisasinya jelas. Bab 5 kesimpulan diambil di bab 4 dan saran disesuaikan 	
Penguji II 16-08-2012		

Disetujui,

Dosen Penguji I

Irmalia Suryani Faradisa, ST, MT
 NIP.P.1030000365

Dosen Penguji II

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

Mengetahui,

Dosen Pembimbing I

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

Dosen Pembimbing II

Yuli Wahyuni ST, MT
 NIP.P. 1031200456



Lampiran : 1 (satu) berkas
Pembimbing Skripsi

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

Yang bertanda tangan dibawah

Nama : **ADI SETYAWAN**
Nim : **0912522**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer**

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

"RANCANG BANGUN USER INTERFACE MODULE KONTROL BERBASIS PLC UNTUK SISTEM KONTROL PROSES PRODUKSI AIR BERSIH PERUSAHAAN DAERAH AIR MINUM (PDAM)"

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. 1030100358

Hormat Kami

ADI SETYAWAN
NIM. 0912522