

**PERANCANGAN DAN PEMBUATAN STORAGE LOGIC
ANALYZER**

**SKRIPSI
JURUSAN TEKNIK ELEKTRO**

Diajukan Untuk Memenuhi Persyaratan
Memperoleh Gelar Sarjana Teknik



Disusun oleh:
M. ULINUHA PUJA D. S.
NIM. 0710633025 - 63

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG
2014**

LEMBAR PERSETUJUAN

PERANCANGAN DAN PEMBUATAN *STORAGE LOGIC ANALYZER*

SKRIPSI

KONSENTRASI TEKNIK ENERGI ELEKTRIK

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh:
M. ULINUHA PUJA D. S.
NIM. 0710633025 – 63

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

Waru Djuriatno, ST., MT.
NIP. 19690725 199702 1 001

Moch. Rif'an, ST., MT.
NIP.19710301 200012 1 001

LEMBAR PENGESAHAN

PERANCANGAN DAN PEMBUATAN *STORAGE LOGIC ANALYZER*

SKRIPSI

JURUSAN TEKNIK ELEKTRO

Diajukan Untuk Memenuhi Persyaratan

Memperoleh Gelar Sarjana Teknik

Disusun oleh:

M.ULINUHA PUJA D.S.

NIM. 0710633025 - 63

Skripsi ini telah diuji dan dinyatakan lulus pada

Tanggal 11 Agustus 2014

DOSEN PENGUJI

Ir. Muhammad Aswin, M.T.

NIP.19640626 1999002 1 001

Adharul Muttaqin, ST., M.T.

NIP.19760121 200501 1 001

R. Arief Setyawan, M.T.

NIP.19750819 199903 1 001

Mengetahui

Ketua Jurusan Teknik Elektro

Muhammad Aziz Muslim, ST., MT., Ph.D.

NIP. 19741203 200012 1 001

ABSTRAK

M. ULINUHA PUJA D.S., Jurusan Elektro, Fakultas Teknik Universitas Brawijaya, September 2013, *Perancangan dan Pembuatan Storage Logic Analyzer*, Dosen Pembimbing: Waru Djuriatno, ST., MT. dan Mochammad Rif'an, ST., MT.

Teknologi yang berkembang pesat saat ini telah mendorong percepatan di berbagai bidang. Peralatan yang menggunakan sistem digital semakin banyak diminati masyarakat. Secara tidak langsung teknologi digital telah menjadi hal penting dari berbagai kehidupan. Dengan semakin diminatinya alat berbasis digital mengharuskan adanya alat ukur yang mampu menganalisis dan mengukur sinyal digital dari rangkaian digital ataupun IC. Dalam hal ini alat yang ada adalah *Logic Analyzer*.

Logic Analyzer memiliki *channel* masukan yang dimanfaatkan untuk membaca logika keluaran dari rangkaian yang diuji lalu ditampilkan dalam bentuk grafik berupa gelombang kotak yang mewakili logika 0 ketika gelombang pada level rendah dan 1 ketika gelombang pada level tinggi.

Karena adanya logika 0 pada gelombang rendah dan 1 pada gelombang yang tinggi, pada tugas akhir ini dibuat sebuah aplikasi yang dapat membaca data dari rangkaian IC TTL menggunakan bahasa pemrograman *Delphi 7*. Proses pembacaan ini dimulai dari komunikasi serial antara *microcontroller* dan *Personal Computer* kemudian dilanjutkan dengan proses menampilkan grafik dan proses penyimpanan data dan penyimpanan gambar.

Dari pengujian dan analisa perancangan *logic analyzer*, diperoleh hasil bahwa *logic analyzer* dapat menyimpan data dan menyimpan gambar grafik pada *Personal Computer*.

KATA PENGANTAR

Puji syukur kehadiran Allah SWT, Sang Maha Pencipta yang telah memberikan limpahan rahmat dan hidayah sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul “Perancangan dan Pembuatan *Storage Logic Analyzer*” sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya Malang.

Tidak banyak yang bisa penulis sampaikan kecuali ungkapan terima kasih kepada berbagai pihak yang telah dengan tulus ikhlas memberikan bimbingan, arahan, dan dukungan hingga penulisan tugas akhir ini dapat terselesaikan. Pada kesempatan kali ini, penulis mengucapkan banyak terima kasih kepada:

1. Bapak M. Azis Muslim, S.T., M.T., Ph.D selaku Ketua Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya.
2. Bapak Hadi Suyono, S.T., M.T., Ph.D selaku Sekretaris Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya.
3. Bapak Mochammad Rif'an, ST., MT. selaku Ketua Program Studi Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya.
4. Bapak Waru Djuriatno, ST., MT. selaku dosen pembimbing I yang telah banyak memberikan bimbingan, masukan dan arahan dalam penyusunan tugas akhir ini.
5. Bapak Mochammad Rif'an. ST., MT. selaku dosen pembimbing II yang telah banyak memberikan bimbingan, masukan dan arahan dalam penyusunan tugas akhir ini.
6. Bapak dan Ibu Dosen serta karyawan jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya.
7. Ayahanda M. Arief Mulyanto, Ibunda Niniek Tarbiyatun, dan Saudaraku M. Bagus H.W. dan M. Arie H.T serta keponakanku Amirul yang telah memberikan dukungan serta do'a dan semangat untuk terus maju hingga terselesaikan skripsi ini.

8. Kekasihku Khairunisa G M P yang selalu ada di saat susah dan senang.
9. Teman-temanku Indra, Emre, Iwan, Tora dan yang lainnya yang gak bisa disebutkan satu persatu.
10. Serta semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya tugas akhir ini.

Penulis menyadari sepenuhnya bahwa Tugas Akhir ini masih banyak kekurangan. Segala kritik dan saran yang membangun akan penulis terima demi kesempurnaan skripsi ini. Harapan penulis semoga skripsi ini bermanfaat bagi pembaca dan khususnya mahasiswa jurusan teknik elektro dimasa yang akan datang.



Malang, Mei 2014

Penyusun

DAFTAR ISI

ABSTRAK	i
KATA PENGANTAR	ii
DAFTAR ISI	iii
BAB I PENDAHULUAN	1
1.1 LATAR BELAKANG	1
1.2 RUMUSAN MASALAH	1
1.3 RUANG LINGKUP	2
1.4 TUJUAN	2
BAB II TINJAUAN PUSTAKA	3
2.1 LOGIC ANALYZER	3
2.1.1 ARSITEKTUR LOGIC ANALYZER	5
BAB III METODOLOGI	8
3.1 BLOCK DIAGRAM SISTEM	8
3.2 PERANCANGAN ALAT	9
3.3 PERANCANGAN PERANGKAT LUNAK	10
BAB IV IMPLEMENTASI	13
4.1 SET UP PERALATAN	13
4.1.1 PERANGKAT KERAS	13
4.1.2 PERANGKAT LUNAK	14
4.1.2.1 DIAGRAM ALIR AT MEGA 16	14
4.1.2.2 DIAGRAM ALIR SOFTWARE PENAMPIL	15
4.1.3 DESKRIPSI ALAT	16
4.1.3.1 JENIS DATA	17
4.1.3.2 INSTRUMEN PENGUMPULAN DATA	17
4.1.3.2.1 RANGKAIAN LOGIC ANALYZER	18
4.1.3.2.2 KOMUNIKASI SERIAL HYPERTERMINAL	19



4.1.3.2.3 KOMUNIKASI SERIAL DELPHI	20
4.1.3.2.4 TEGANGAN LOGIKA	20
4.1.3.2.5 PROGRAM DELPHI	20
4.1.4 TEKNIK ANALISA DATA.....	21
4.2 IMPLEMENTASI SISTEM	22
4.2.1 LINGKUNGAN IMPLEMENTASI	22
4.2.2 PENGOLAHAN DATA.....	22
4.2.2.1 PROSES PENGAMBILAN DATA	22
4.2.2.2 PROSES KONVERSI DARI DESIMAL KE BINER	23
4.2.2.3 PROSES PEMILIHAN MODE.....	24
4.2.2.4 PROSES MENAMPILKAN GRAFIK.....	26
4.2.2.5 PROSES MENYIMPAN DATA DAN GAMBAR	27
BAB V PENGUJIAN.....	28
5.1 PENYAJIAN DATA	28
5.1.1 PENGUJIAN RANGKAIAN MINIMUM.....	28
5.1.2 PENGUJIAN KOMUNIKASI SERIAL HYPERTERMINAL.....	30
5.1.3 PENGUJIAN KOMUNIKASI SERIAL DELPHI 7	32
5.1.4 PENGUJIAN TEGANGAN LOGIKA	33
5.1.5 PENGUJIAN PROGRAM LOGIC ANALYZER DELPHI 7	34
5.1.6 PENGUJIAN KESELURUHAN.....	37
5.2 PEMBAHASAN.....	40
BAB VI PENUTUP.....	42
6.1 KESIMPULAN	42

BAB I

PENDAHULUAN

I. LATAR BELAKANG

Pada zaman yang semakin maju ini teknologi informasi selalu berkembang cukup pesat. Banyak sekali aplikasi - aplikasi yang muncul seiring dengan kemajuan tersebut. Cara manual pun semakin tidak berarti dengan majunya teknologi informasi ini. Kita bisa mengerjakan pekerjaan apapun secara lebih gampang dan otomatis.

Kemajuan teknologi informasi ini menyebabkan teknologi elektronika menjadi ikut berkembang pula. Kedua teknologi ini saling terkait satu sama lain. Banyak sekali aplikasi dalam teknologi yang biasa kita nikmati saat ini dan itu mempermudah kita untuk melakukan segala sesuatu hal yang berhubungan dengan elektronika.

Dengan adanya kemudahan tersebut, setiap orang semakin bisa meneliti suatu penelitian dengan mudah. Setiap kita melakukan praktikum atau menjalani kuliah kita selalu direpotkan dengan melihat keluaran atau *output* dari suatu rangkaian yang tidak bisa disimpan hasil keluarannya.

Dari permasalahan diatas, penulis mendapatkan ide untuk mencoba mengatasi permasalahan tersebut yaitu membuat program *storage logic analyser* yang dimana program ini bisa menyimpan dan menampilkan data melalui monitor *personal computer* atau laptop sehingga bisa memudahkan pengajar untuk menampilkan keluaran dari rangkaian *microcontroller* kepada peserta mahasiswa yang mengikuti kelas tersebut.

II. RUMUSAN MASALAH

Berdasarkan latar belakang yang telah dipaparkan secara jelas diatas, maka rumusan masalah ditekankan pada:

1. Bagaimana cara membuat alat yang bisa mendeteksi tegangan logika.
2. Bagaimana membuat alat yang dapat berkomunikasi dengan PC.
3. Bagaimana membuat software pada *microcontroller* untuk mendeteksi logika rangkaian.

4. Bagaimana membuat software dengan bahasa pemrograman delphi pada PC untuk menerima, menyimpan dan menampilkan data dalam bentuk grafik maupun tabel data digital

III. RUANG LINGKUP

Penyusunan penelitian ini menggunakan batasan masalah sebagai berikut:

1. *Microcontroller* yang digunakan AT MEGA 16.
2. Program pada *microcontroller* menggunakan bahasa pemrograman Bahasa C/C++.
3. Program pada PC menggunakan bahasa pemrograman Delphi.
4. Rangkaian dihubungkan ke PC menggunakan USB to TTL (CP2102).
5. Frekuensi tidak *disampling*

IV. TUJUAN

Membuat *logic analyser* menggunakan AT MEGA 16 yang akan ditampilkan pada layar monitor *personal computer*.

BAB II

TINJAUAN PUSTAKA

2.1 Logic Analyser

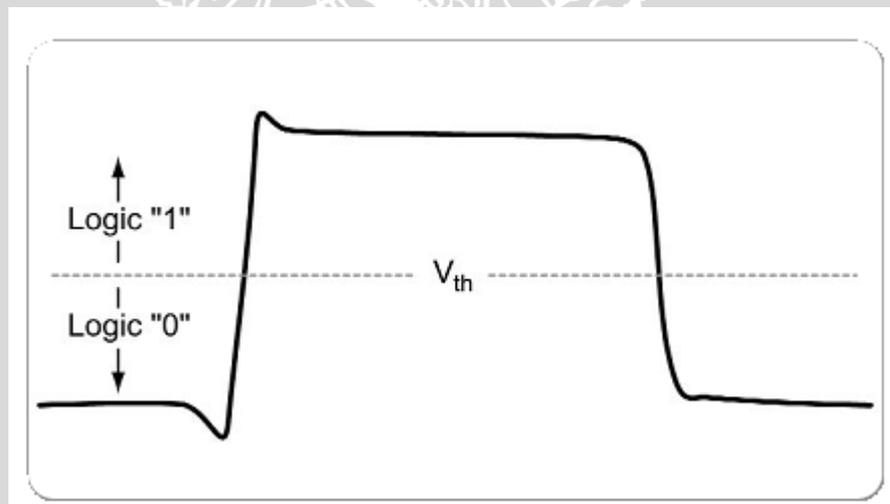
Logic Analyser memiliki kemampuan berbeda dari osiloskop. Perbedaan paling jelas antara dua instrumen adalah jumlah saluran (input). Osiloskop digital memiliki hingga sampai empat input sinyal. Analisa logika memiliki antara 34 dan 136 saluran. Setiap saluran input satu sinyal digital. Beberapa kompleks desain sistem memerlukan ribuan saluran input. Tepat analisa logika skala yang tersedia untuk tugas – tugas juga. Sebuah analisa logika mengukur dan menganalisa sinyal berbeda dari osiloskop.

Logic Analyzer berfungsi untuk membaca waktu dan keadaan dari pengambilan *sample input* dalam bentuk gelombang (*input wave-form*) untuk menentukan tinggi atau rendahnya *amplitude* gelombang. Jika sinyal *sample* berada diatas tegangan ambang yang digunakan maka akan ditampilkan sebagai gelombang dengan *amplitude* tinggi atau 1 pada *Logic Analyzer* dan sebaliknya, jika sinyal berada dibawah batas tegangan ambang yang digunakan akan ditampilkan sebagai gelombang dengan *amplitude* rendah atau 0. Dari *sample* ini dihasilkan serangkaian angka 0 dan 1 yang mewakili gambar 1 bit pada *input wave-form*. Serangkaian angka inidisimpan dalam memori dan digunakan untuk menampilkan gambar 1 bit pada *input wave-form* dalam bentuk grafik dengan sumbu x mewakili waktu dan sumbu y mewakili amplitudo tegangan.

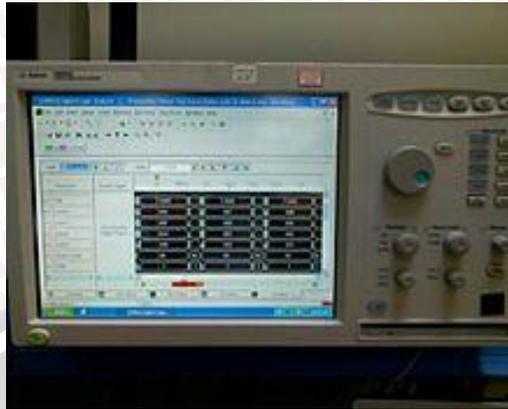
Penganalisa logika tidak mengukur detail analog. Sebaliknya, mendeteksi tingkat logika ambang batas. Ketika Anda menghubungkan *logika analyzer* ke sirkuit digital, Anda hanya peduli dengan logika keadaan sinyal. Sebuah analisa logika mencari hanya dua tingkat logika, seperti yang ditunjukkan pada Gambar 6.1.1 Ketika input berada di atas ambang tegangan (V) tingkat dikatakan menjadi "tinggi" atau "1," sebaliknya, tingkat bawah V_{Th} adalah "0" "rendah" atau Ketika logika analisa sampel masukan, itu menyimpan sebuah "1" atau "0" tergantung pada tingkat sinyal relatif terhadap ambang tegangan. Tampilan

gelombang waktu sebuah *logic analyzer* sama dengan waktu Diagram yang ditemukan dalam lembar data atau diproduksi oleh simulator. Semua sinyal berkorelasi waktu, sehingga pengaturan dan *hold time*, lebar pulsa, data asing atau data yang hilang dapat dilihat. Selain tinggi di channel hitungan, analisa logika menawarkan fitur penting yang mendukung desain digital verifikasi dan *debugging*. Di antaranya adalah :

1. Pemicu yang canggih memungkinkan Anda menentukan kondisi dimana *logic analyzer* memperoleh Data.
2. *High-density probe* dan *adapter* yang menyederhanakan koneksi ke sistem yang diuji atau *system under test (SUT)*.
3. Kemampuan analisis yang menerjemahkan data yang diambil menuju ke prosesor instruksi dan berkorelasi kepada kode sumber



Gambar 2.1. *logic analyzer* menentukan nilai logika relatif terhadap ambang tegangan tingkat.



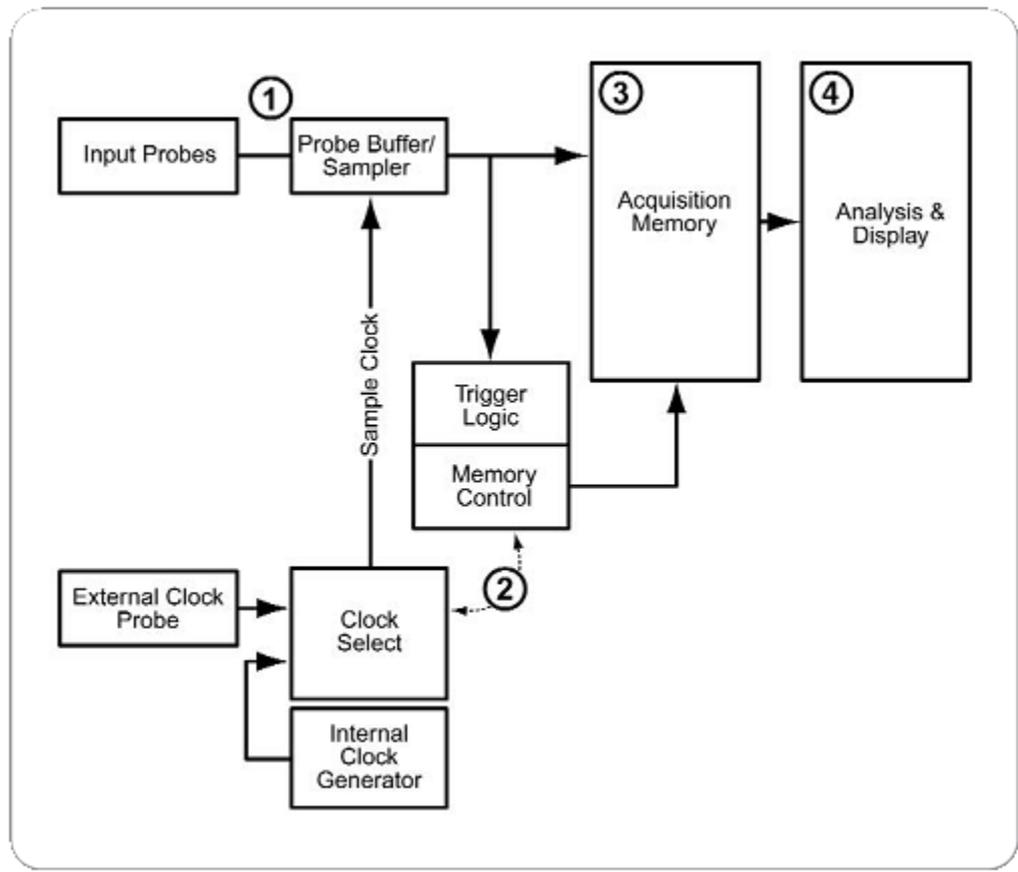
Gambar 2.2 Logic Analyser

2.1.1. Arsitektur *Logic Analyzer* dan Pengoperasiannya

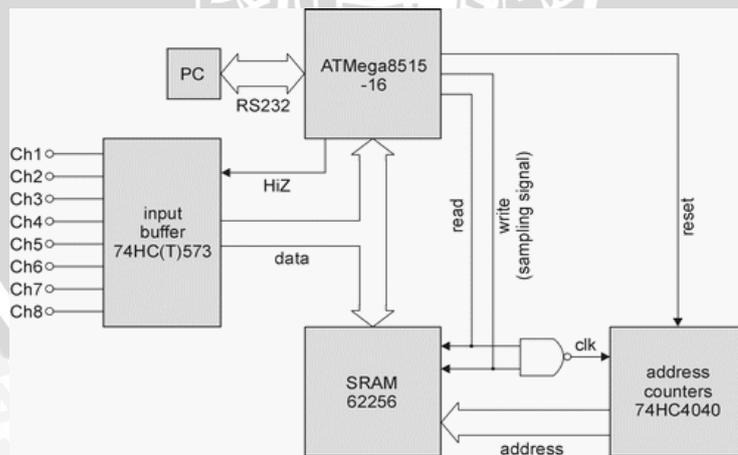
Logic analyzer untuk menyambung, memperoleh, dan menganalisa sinyal digital. Ini adalah empat langkah untuk menggunakan penganalisis logika :

1. Probe: tersambung ke *the System Under Test* – *SUT*
2. *Setup (clock mode and triggering)*
3. Memperoleh data
4. *Menganalisis dan menampilkan data (Analyze and display)*

Gambar pada gambar 2.3. adalah *logic analyzer* diagram blok sederhana. setiap blok melambangkan beberapa hardware dan/atau elemen lunak. Blok nomor sesuai dengan empat langkah yang tercantum di atas.

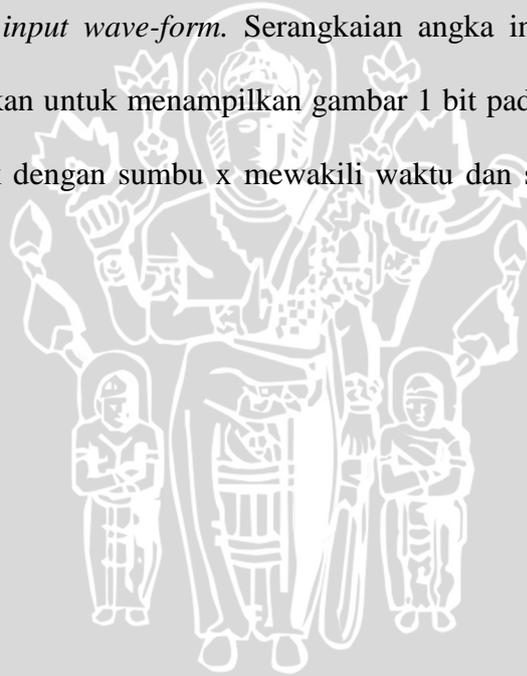


Gambar 2.3 Diagram blok *logic analyzer* sederhana
(Sumber :http://www.tektronix.com/logic_analyzers)



Gambar 2.4 Block Diagram Logic Analyser
(sumber :ep.com.pl)

Logic Analyzer adalah alat yang berfungsi untuk membaca waktu dan keadaan dari pengambilan *sample input* dalam bentuk gelombang (*input wave-form*) untuk menentukan tinggi atau rendahnya *amplitude* gelombang. Jika sinyal *sample* berada diatas tegangan ambang yang digunakan maka akan ditampilkan sebagai gelombang dengan *amplitude* tinggi atau 1 pada *Logic Analyzer* dan sebaliknya, jika sinyal berada dibawah batas tegangan ambang yang digunakan akan ditampilkan sebagai gelombang dengan *amplitude* rendah atau 0. Dari *sample* ini dihasilkan serangkaian angka 0 dan 1 yang mewakili gambar 1 bit pada *input wave-form*. Serangkaian angka ini disimpan dalam memori dan digunakan untuk menampilkan gambar 1 bit pada *input wave-form* dalam bentuk grafik dengan sumbu x mewakili waktu dan sumbu y mewakili amplitudo tegangan.



BAB III METODOLOGI PENELITIAN

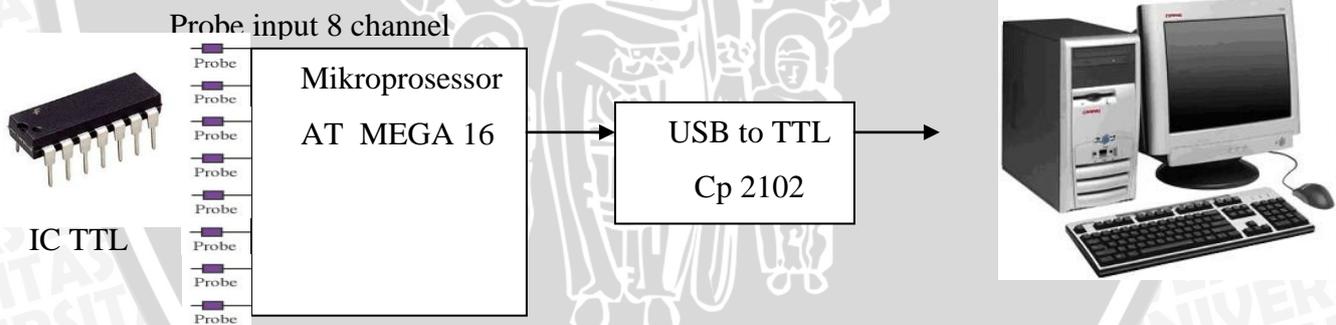
Metode penelitian yang digunakan dalam penelitian ini secara umum tersusun sebagai berikut:

1. Studi literatur
2. Perancangan alat
3. Pembuatan alat
4. Pengujian alat
5. Pengambilan kesimpulan dan saran

menghasilkan perangkat yang sesuai dengan tujuan awal penelitian.

3.1 Blok Diagram Sistem

Berdasarkan spesifikasi yang telah ditentukan, sistem *storage logic analyser* digambarkan secara garis besar dalam sebuah blok diagram seperti ditunjukkan pada Gambar 3.1



Gambar 3.1. Blokdiagram keseluruhan sistem

Keterangan blok diagram yang ditunjukkan pada Gambar 3.1 sebagai berikut:

1. Rangkaian pada gambar 3.1 adalah rangkaian yang akan dihubungkan dengan mikrokontroller
2. *Probe input 8* adalah *channel* yang sesuai dengan masukannya sejumlah 8 buah *probe*

3. Mikrokontroler *AT Mega 16* sebagai pengolah data utama yang bersumber dari rangkaian sebelumnya dan meneruskan hasil perhitungan internal berupa logika digital sebagai bentuk sinyal pemicuan dan diteruskan ke rangkaian selanjutnya.
4. *USB to TTL* tipe CP 2102 adalah jalan komunikasi antara mikrokontroler dan *Personal Computer*
5. *Personal Computer* sebagai alat untuk menampilkan hasil sinyal digital pada rangkaian yang diproses oleh mikrokontroler

3.2 Perancangan Alat

Perancangan alat yang digunakan dalam pembuatan *storage logic analyser* harus memenuhi spesifikasi dari peralatan yang akan digunakan. Spesifikasi peralatan yang digunakan antara lain:

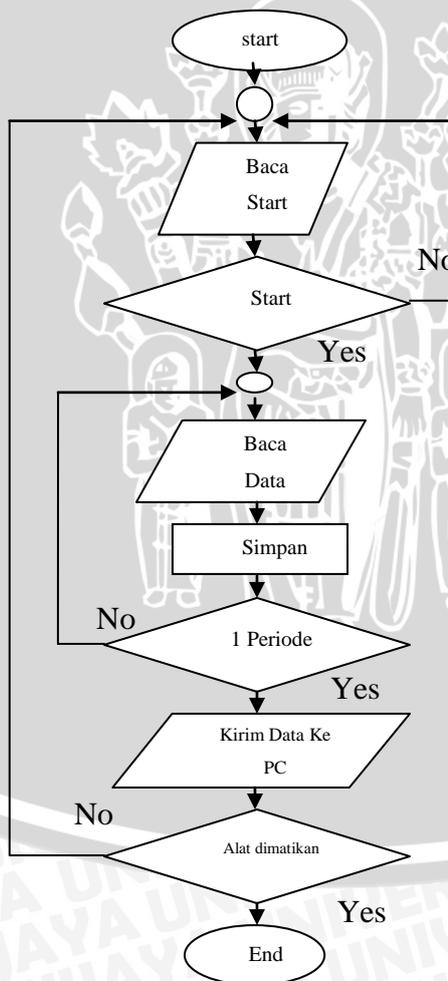
Logic Analyzer

- Signal input 8 *channel*
- 5 mode *trigger*
- Frekuensi 0 -11,0592 MHz
- Tampilan untuk kondisi logika keluaran IC TTL/ Rangkaian elektronika dalam bentuk grafik di layar komputer

3.3 Perancangan Perangkat Lunak

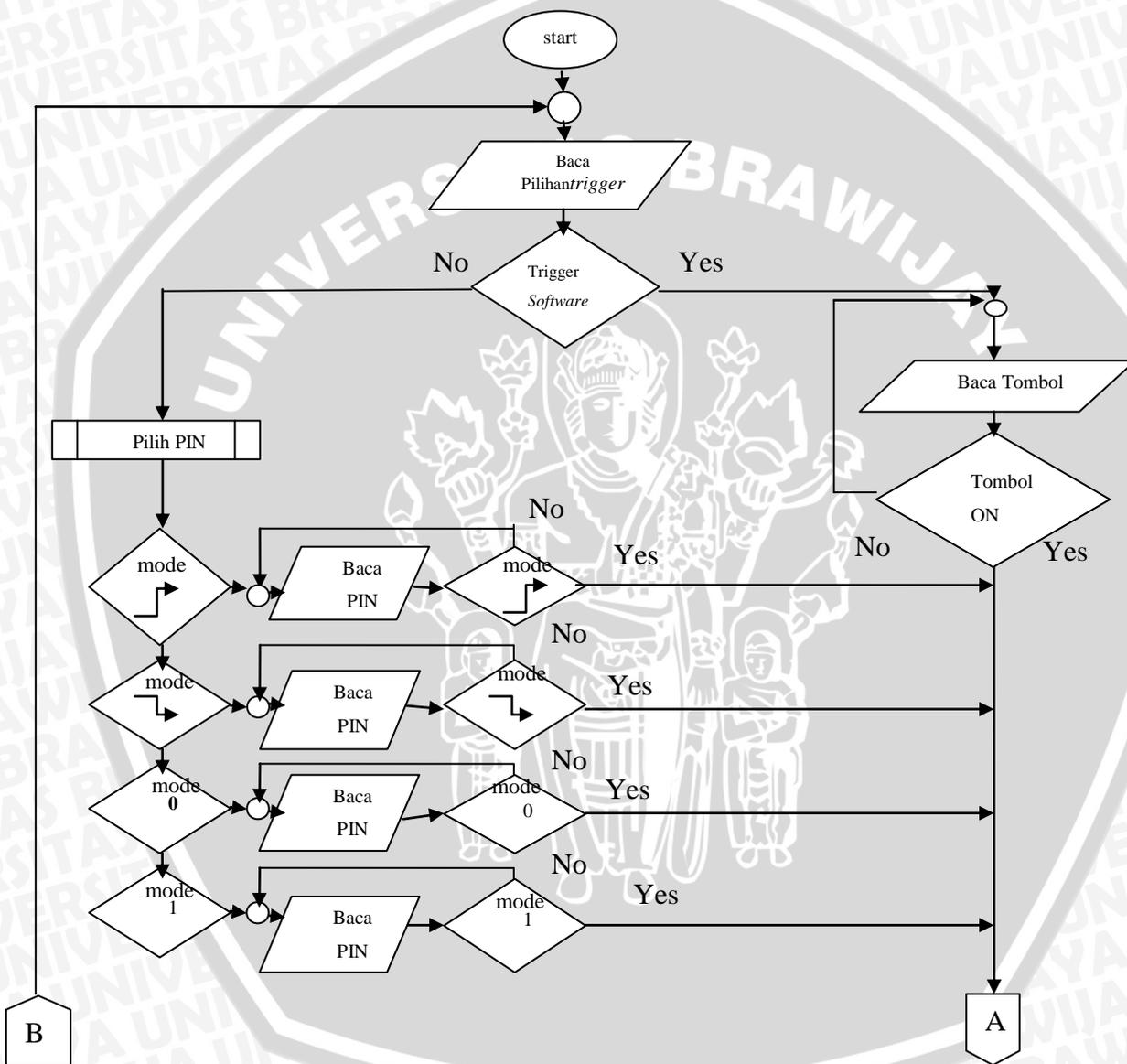
Perancangan perangkat lunak dibuat untuk mendukung perangkat keras yang telah dirancang sebelumnya. Perangkat lunak yang dirancang diharapkan sesuai dengan tujuan awal penelitian mengenai *storage logic analyser*. Perancangan perangkat lunak bertujuan agar mikrokontroler mampu berkomunikasi dengan *personal computer*. Perangkat lunak yang akan dirancang adalah perangkat lunak pada mikrokontroler ATmega16 menggunakan bahasa C/C++ dan perangkat lunak pada *personal computer* menggunakan pemrograman delphi.

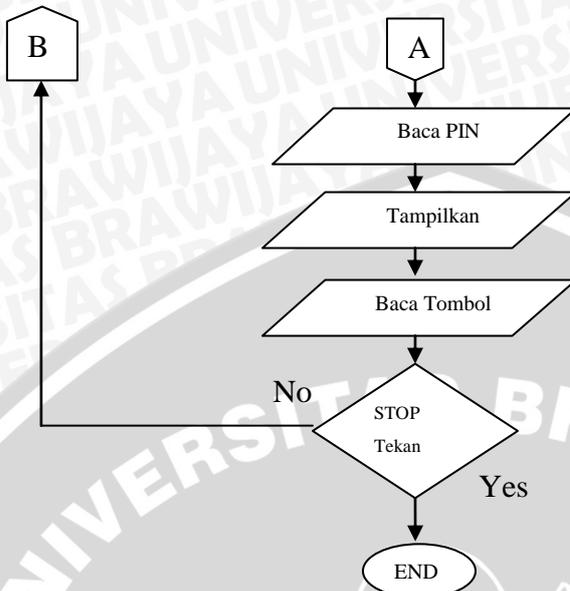
1. Perancangan diagram alir pada mikrokontroler



Gambar 3.2 Diagram alir pada MK

2. Perancangan diagram alir program *logic analyzer* pada *personal computer*





(Lanjutan)

Gambar 3.3 Diagram alir Software logic analyzer

BAB IV

IMPLEMENTASI

4.1 Set Up Peralatan

Set up peralatan merupakan hal yang wajib dilakukan dalam pembuatan alat, agar proses pembuatan alat dapat berjalan dengan baik dan sistematis. *Set up* peralatan ini dibagi menjadi dua perancangan yaitu:

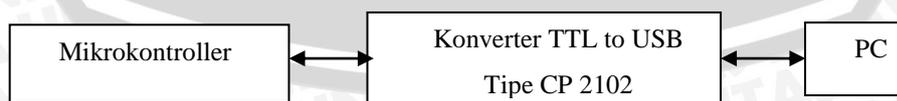
4.1.1 Perangkat Keras (*Hardware*)

Secara umum perancangan ini terdiri dari dua unit yaitu unit pada mikrokontroler dan unit pada komputer dengan masing-masing unit terdiri dari beberapa blok rangkaian. Diagram blok unit pada mikrokontroler dan PC *logic analyzer* yang dirancang adalah seperti gambar 4.1:



Gambar 4.1 Diagram Blok Unit Pada Mikrokontroler

Diagram blok unit pada PC *Logic Analyzer* yang dirancang adalah seperti gambar 4.2:

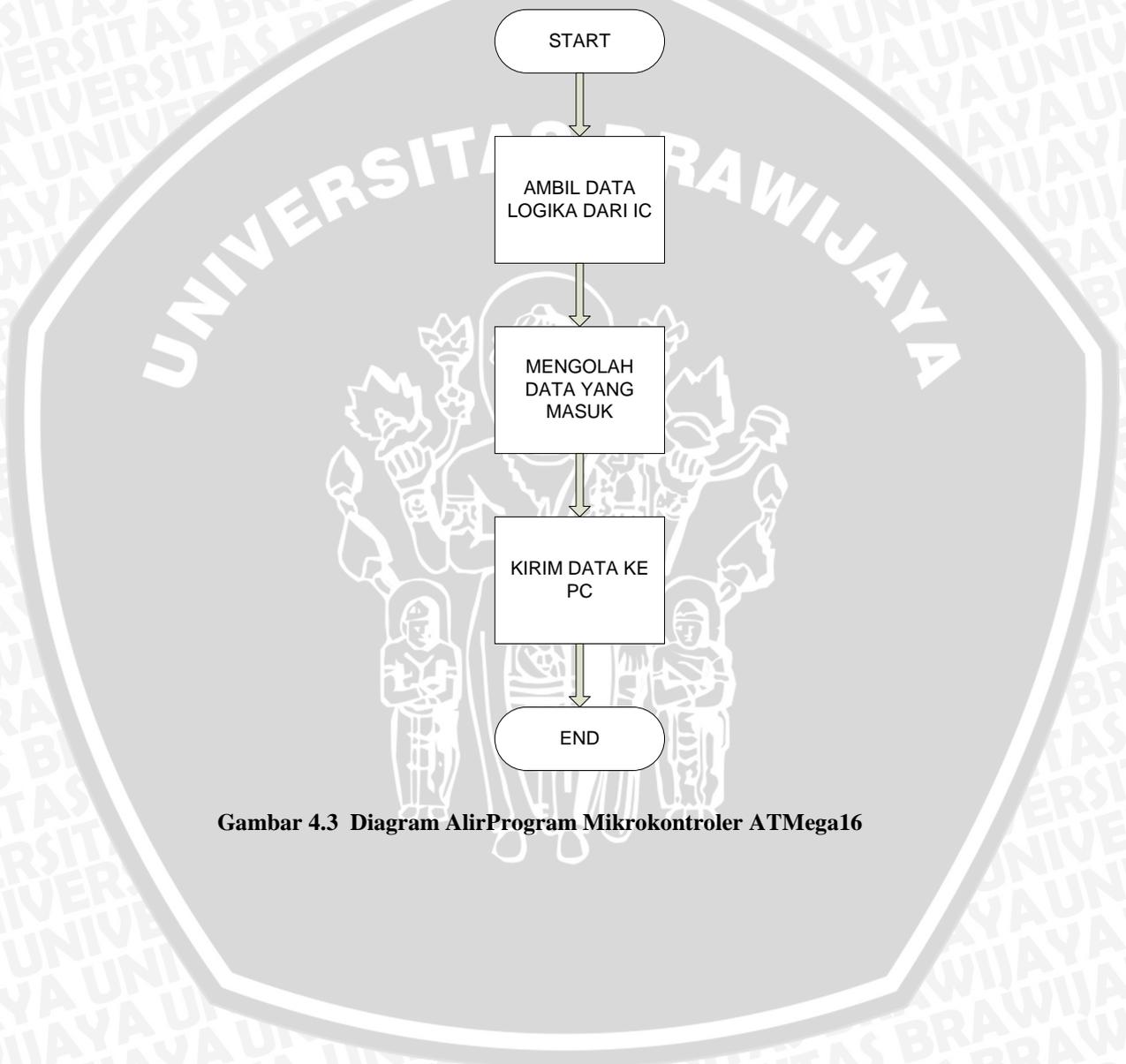


Gambar 4.2 Diagram Blok Unit Pada Komputer

4.1.2 Perangkat Lunak (*software*)

Implementasi *software* dilakukan pada dua bagian antara lain:

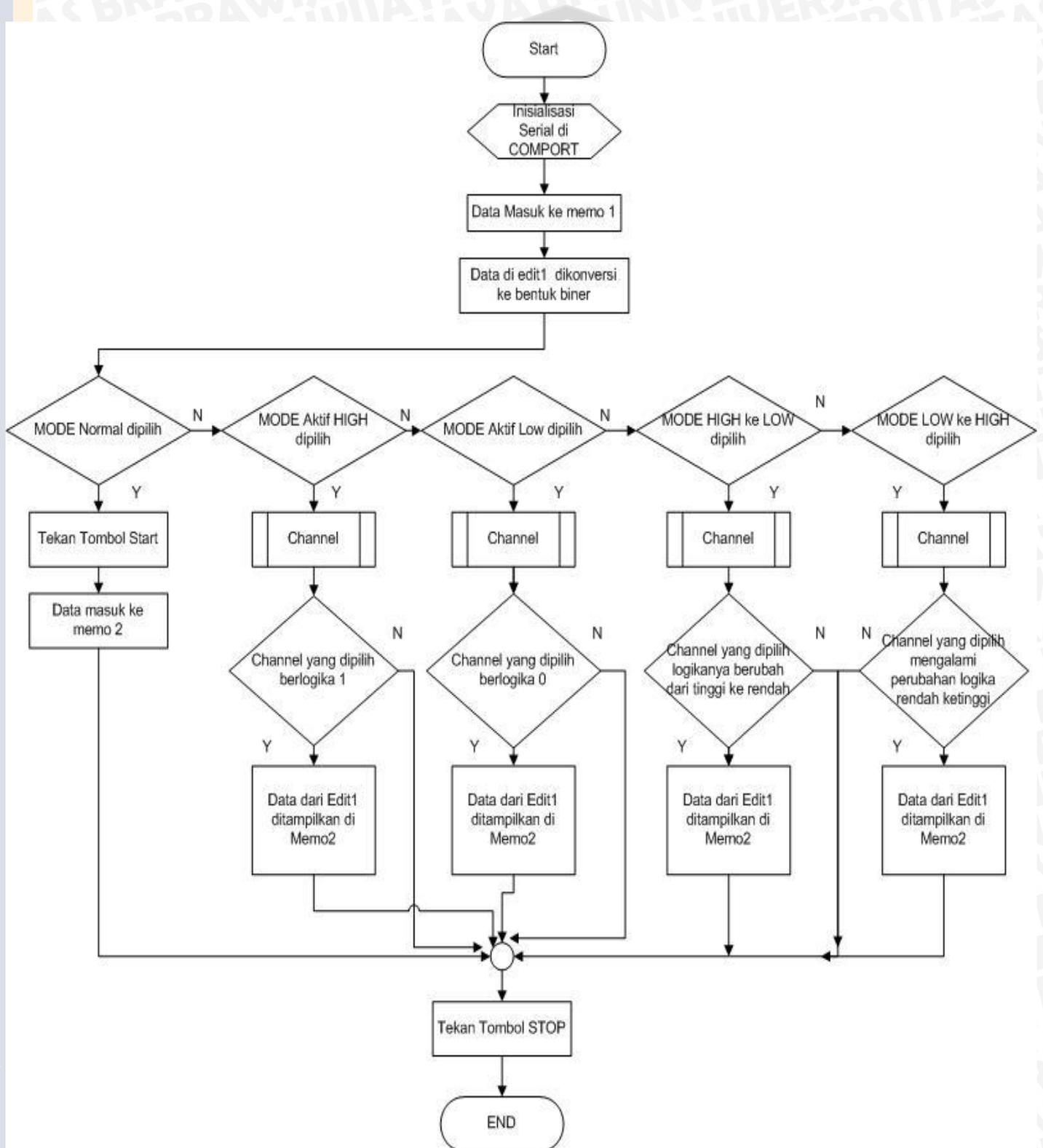
4.1.2.1 Diagram Alir Mikrokontroler Atmega 16



Gambar 4.3 Diagram Alir Program Mikrokontroler ATMega16

4.1.2.2 Diagram Alir Software Penampil

Diagram Alir pada Delphi



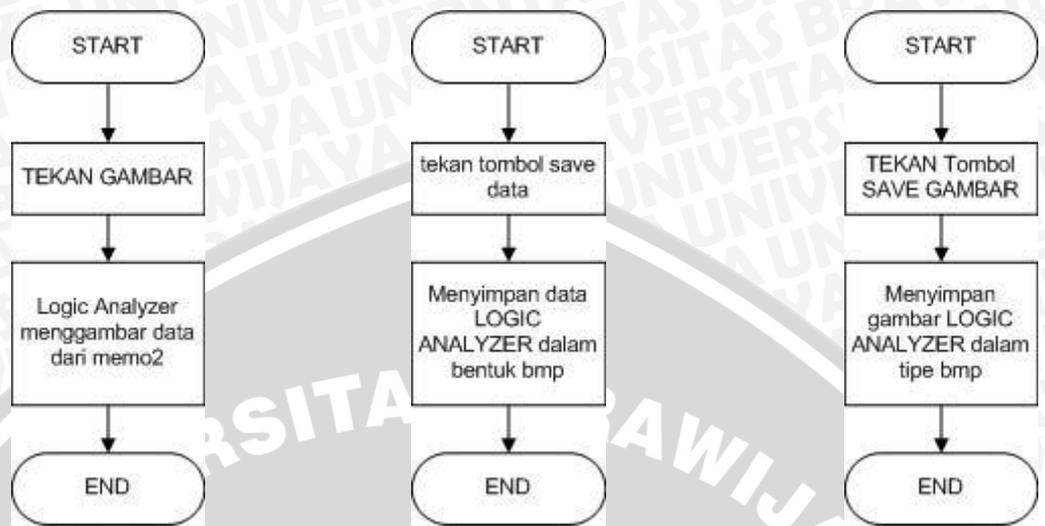


Diagram Alir
Penampikan grafik
(a)

Diagram Alir
Penyimpanan data
(b)

Diagram Alir
Penyimpanan gambar
(c)

Gambar 4.4 Diagram Alir Program Delphi pada PC

Pada penelitian ini diagram alir yang dibutuhkan untuk mempermudah pembuatan *software Logic Analyzer* menggunakan *Delphi* meliputi:

1. Pengambilan data mode Real Time, pada mode ini pengambilan data yang dikirim dari mikrokontroler akan ditampilkan secara langsung. Mode ini diperlukan dalam pengujian yang tidak memerlukan frekuensi tinggi
2. Proses menampilkan data dalam bentuk grafik, pada proses ini data yang diterima akan dikonversi menjadi bentuk biner yang selanjutnya digunakan untuk menggambar grafik.

4.1.3 Deskripsi Alat

Alat *Logic Analyzer* ini berfungsi untuk membaca dan memberi data logika dari suatu IC atau alat elektronika. Alat ini menggunakan satu sistem minimum ATmega 16 dan satu PC. *Logic Analyzer* ini memiliki 8 *channel* masukan, 1 *channel* vcc dan 1 *channel* ground. Jumlah *channel* ini memanfaatkan port pada Atmega 16. *Port A* memiliki 8 pin yang dimanfaatkan sebagai *channel* masukan, sedangkan *Port D* dimanfaatkan sebagai komunikasi serial.

Data logika masukan diukur dan dikirim melalui komunikasi serial asinkron dengan *baudrate* 115200 bps menuju PC, begitu juga sebaliknya data logika keluaran dari PC dikirim menuju mikrokontroler. Data masukan dan keluaran ditampilkan pada PC dengan menggunakan *software Borland Delphi 7*. Arah komunikasi alat ini yaitu *Half Duplex*. Pada saat program *Logic Analyzer* dijalankan pertama kali maka harus melakukan *setting* komunikasi yang terdeteksi pada PC. Untuk menjalankan program yang kedua kali dan seterusnya tidak perlu melakukan *setting* komunikasi melainkan langsung dapat menjalankan program.

4.1.3.1 Jenis Data

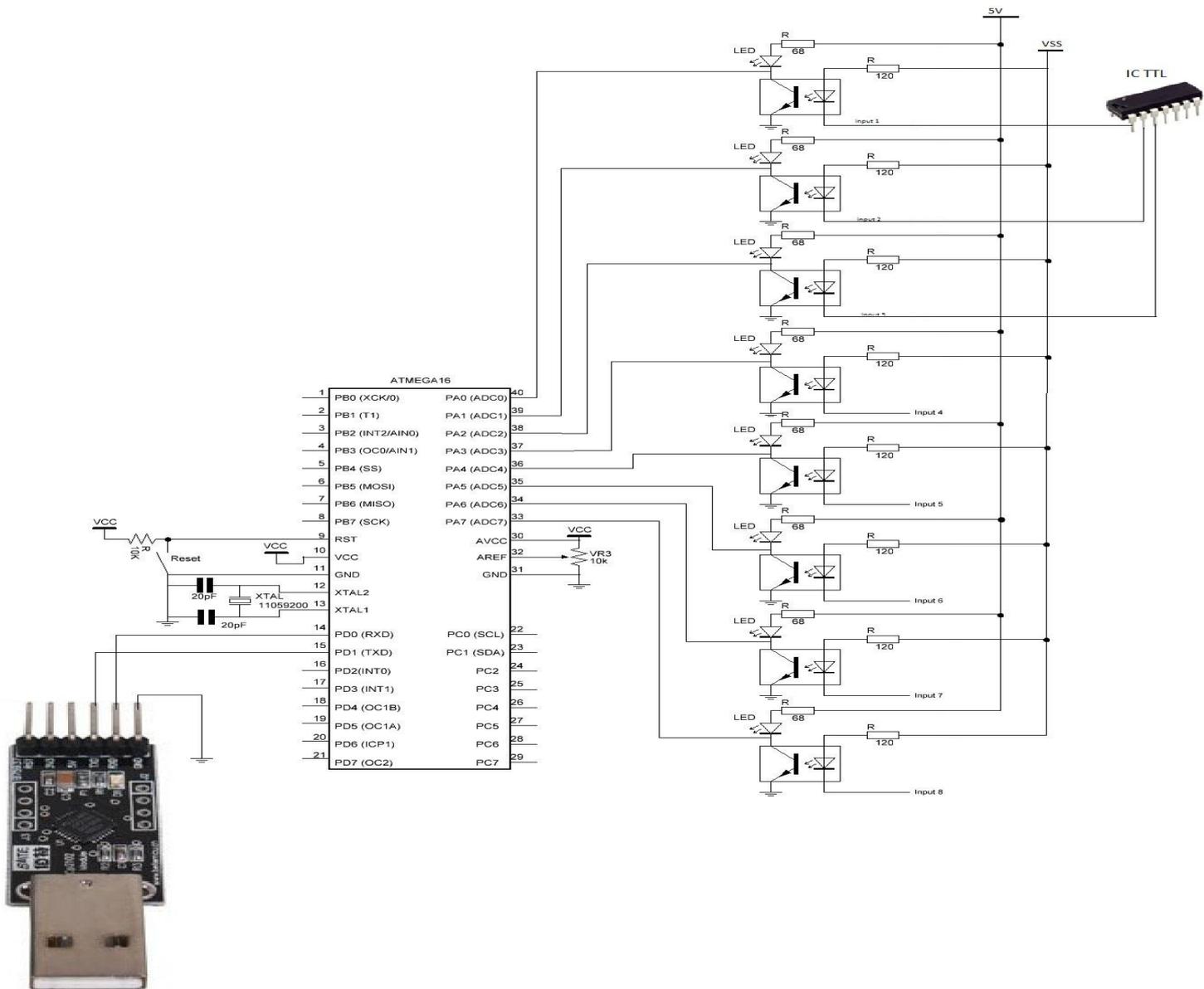
Jenis data pada penelitian dan pengembangan ini adalah data kuantitatif yaitu data uji komunikasi serial dengan *hyperterminal*, pengukuran tegangan logika, dan pengukuran hasil keluaran *Logic Analyzer*. Pengumpulan data dilakukan dengan cara pengukuran, pengujian data uji mekanis dan hasil keluaran *Logic Analyzer*.

4.1.3.2 Instrumen Pengumpulan Data

Data dalam penelitian dan pengembangan ini didapatkan dengan menggunakan aplikasi program *user interface*.

4.1.3.2.1 Rangkaian Logic Analyzer

1. Rangkaian Logic Analyzer digambarkan seperti pada gambar 4.5



Gambar 4.5 Skema Rangkaian *Logic Analyzer*

2. Probe input dihubungkan ke kaki IC TTL yang akan diuji
3. Membuat program pengujian sistem minimum berupa nyala led seperti listing dibawah ini

```
while (1)
{
// Place your code here
PORTC=255;
delay_ms(200);
};
```

4. Mengisikan program pada sistem minimum
5. Melakukan Pengujian dan Pengamatan
6. Mengambil data hasil Pengujian

4.1.3.2.2 Komunikasi Serial dengan *Hyperterminal*

1. Membuat program pada mikrokontroler ATmega16.
2. Menginisialisasi *comport* dengan *baud rate* pada mikrokontroler ATmega16, dan pada PC melalui *hyperterminal*.
3. Menghubungkan *transmitter* dan *Receiver* pada sistem minimum mikrokontroler ATmega16 menuju PC melalui *converter USB to TTL*.
4. Mengirimkan karakter huruf 'ULINUHA' yang akan ditampilkan pada PC melalui *hyper terminal* menggunakan instruksi:

```
while (1)
{
```

```
// Place your code here
puts("\n\rULINUHA\n\r");
delay_ms(1000);
}
```

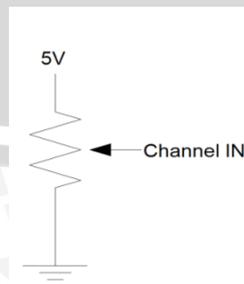
Menampilkan data hasil pengujian.

4.1.3.2.3 Komunikasi Serial dengan *Borland Delphi7*

1. Membuat program pada *mikrocontroller* ATmega16
2. Menginisialisasi *comport* dengan *baud rate* pada mikrokontroler ATmega16, dan pada PC melalui *Delphi*.
3. Menghubungkan *transmitter* dan *Receiver* pada sistem minimum mikrokontroler ATmega16 menuju PC melalui *converter* USB to TTL.
4. Mengirimkan karakter huruf tertentu yang akan ditampilkan pada PC melalui memo pada *Delphi7*.

4.1.3.2.4 Mengukur Tegangan Logika

1. Merangkai alat *Logic Analyzer*.
2. Menghubungkan *power supply* pada alat.
3. Menghubungkan *Channel* dengan *power supply* yang tegangannya dapat divariasikan dengan Potensio 10K seperti pada gambar 4.6.



Gambar 4.6 Rangkaian Pengambilan Data Logika

4.1.3.2.5 Program *Delphi* 7

1. Membuat program pada mikrokontroler ATmega16
2. Menginisialisasi *comport* dengan *baud rate* pada mikrokontroler ATmega16, dan pada PC melalui *Delphi*.
3. Menghubungkan *transmitter* dan *Receiver* pada sistem minimum mikrokontroler ATmega16 menuju PC melalui *converter* USB to TTL.
4. Menerima data logika masukan dari mikrokontroler dan mengirim data logika keluaran ke mikrokontroler
5. Memilih Mode penerimaan data pada *Logic Analyzer*
6. Menampilkan data logika masukan, keluaran dan waktu dalam memo2
Menampilkan data logika masukan, keluaran dalam bentuk grafik pada *chart1*.
7. Menekan tombol *save data* dan *save picture* untuk menyimpan data dan gambar ke PC

Implementasi ini dilakukan untuk mengetahui apakah alat telah berfungsi sesuai dengan perancangan atau tidak. Pengujian ini dilakukan dengan melibatkan keseluruhan rangkaian *hardware* dan *software* yang telah dibuat. Data masukan dan keluaran yang ditampilkan pada PC dibandingkan dengan datasheet IC dan Sumber data rangkaian elektronika yang diuji.

4.1.4 Teknik Analisa Data

Proses analisis data dilakukan untuk mengetahui apakah sistem bekerja sesuai dengan perencanaan yang dilakukan pada masing-masing pengujian

a. **Analisis *Bit Error Rate* (BER) pada Komunikasi Serial.**

Pengujian BER yaitu dengan cara pengiriman karakter huruf tertentu sebanyak 100 kali menuju PC melalui hyperterminal. Jumlah data yang tidak sesuai dengan yang dikirimkan pada hyperterminal disebut data error. Analisis data error (BER) yaitu:

$$\% \text{BER} = \frac{\text{Data Error}}{\text{Data Keseluruhan}} \times 100\%$$

b. **Sistem Keseluruhan**

Analisis pada sistem ini bertujuan untuk mengetahui apakah alat yang dibuat sudah sesuai dengan spesifikasi yang direncanakan. Perbandingan antara datasheet dengan data yang ditampilkan oleh komputer.

4.2 Implementasi Sistem

Setelah tahap perancangan tahap selanjutnya adalah tahap implementasi. Implementasi ini merupakan proses transformasi hasil perancangan perangkat lunak yang telah dibuat ke dalam kode (*coding*) sesuai dengan sintaks dari bahasa pemrograman yang digunakan.

4.2.1 Lingkungan Implementasi

Aplikasi dibuat dengan menggunakan IDE Netbeans7.0. Sistem diimplementasikan dengan menggunakan spesifikasi sebagai berikut:

- 1.) Perangkat keras
AT MEGA 16
Spesifikasi:
 - frekuensi 0 – 11,0592 MHz
 - Signal input 8 *channel*

- Tampilan untuk kondisi logika keluaran IC TTL/ Rangkaian elektronika dalam bentuk grafik di layar komputer

- 5 mode *trigger*

2. Perangkat Lunak

- Bahasa pemrograman : C/C++ dan Delphi

4.2.2 Pengolahan Data

Aplikasi pengolahan data ini menggunakan teknik konversi data desimal menjadi biner, proses pemilihan mode untuk mengambil data teknik konversi biner ke grafik, dan teknik menyimpan data dan gambar grafik.

4.2.2.1 Proses Pengambilan Data

Bagian script Delphi sebagai berikut untuk mengambil data :

```
procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
var
  terima:string ;
begin
  comport1.ReadStr(terima,count) ;
  memo1.Text:=memo1.Text+terima ;
end;

// Comport membaca data dari MK dan dimasukkan dalam variable
terima dan di masukkan ke memo 1
```

4.2.2.2 Proses konversi data dari Desimal ke Biner

Bagian script Delphi sebagai berikut untuk mengkonversi data desimal yang dikirimkan dari mikroprosesor menjadi bilangan biner :

```
procedure TForm1.Memo1Change(Sender: TObject);
var
  i,ulang:integer ;
```

```

begin
if memo1.Lines.Count>10 then
begin
i:=strtoint(memo1.Lines.Strings[9]);
  biner1:= "";
for ulang:=0 to 7 do
begin
if (i mod 2) = 1 then biner1 := '1' + biner1
else biner1 := '0' + biner1 ;
i:=i div 2 ;
end ;
edit1.Text := biner1

```

4.2.2.3 Proses Pemilihan Mode

Bagian script Delphi sebagai berikut berfungsi untuk memilih mode penangkapan:

```

if radiogroup1.itemindex= 1 then
begin
if(edit1.text[8-radiogroup2.itemindex]='1') then
timer1.enabled:=true
else
timer1.Enabled:=false ;
end ;
if radiogroup1.ItemIndex= 2 then
begin
if(edit1.text[8-radiogroup2.itemindex]='0') timer1.enabled:=true
then
timer1.enabled:=true

```

```

else
    timer1.Enabled:=false ;
end ;
if radiogroup1.ItemIndex= 3 then
begin
if(edit2.text[8-radiogroup2.itemindex]='1')
    timer1.enabled:=true
and (edit1.text[8-radiogroup2.itemindex]='0')
then
    timer1.enabled:=true;
end ;
if radiogroup1.ItemIndex= 4 then
begin
if(edit2.text[8-radiogroup2.itemindex]='0')
and (edit1.text[8-radiogroup2.itemindex]='1')
    timer1.enabled:=true
then
    timer1.enabled:=true;
end ;
MEMO1.Lines.Delete(0);
end ;
    edit2.Text:=edit1.Text ;
end ;

```

4.2.2.4 Proses menampilkan Grafik

```

procedure TForm1.Button4Click(Sender: TObject);
var
    baris : integer ;
    binner : string ;
begin
    for baris := 0 to memo2.lines.count - 1 do

```

```

begin
binner := memo2.lines.Strings [baris] ;
if length(binner) =8 then
begin
if (binner[1]='0') then series1.add (0) else series1.Add(1) ;
if (binner[2]='0') then series2.add (2) else series2.Add(3) ;
if (binner[3]='0') then series3.add (4) else series3.Add(5) ;
if (binner[4]='0') then series4.add (6) else series4.Add(7) ;
if (binner[5]='0') then series5.add (8) else series5.Add(9) ;
if (binner[6]='0') then series6.add (10) else series6.Add(11) ;
if (binner[7]='0') then series7.add (12) else series7.Add(13) ;
if (binner[8]='0') then series8.add (14) else series8.Add(15) ;
end ;
end ;

end;

```



4.2.2.5 Proses menyimpan data dan gambar

Prosedur save picture

```

procedure TForm1.Button5Click(Sender: TObject);
begin
if SavePictureDialog1.Execute then Chart1.SaveToBitmapFile
(SavePictureDialog1.FileName) ;

```

end;

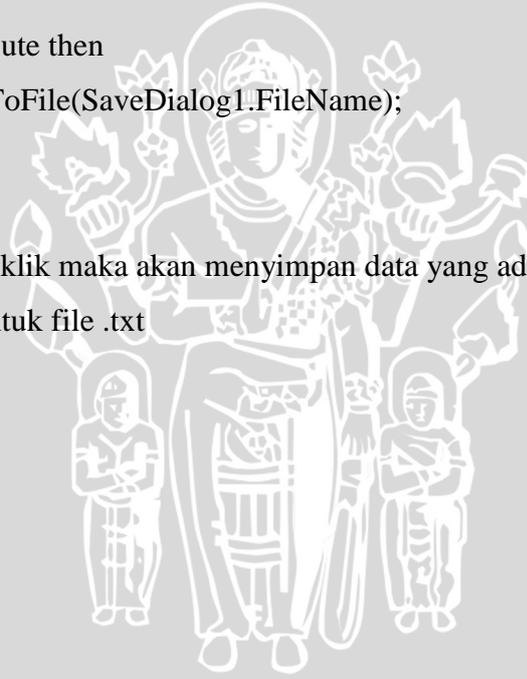
NB:

jika button5 atau button save picture di klik maka melakukan proses penyimpanan data chart ke dalam file bitmap

Prosedur save data

```
procedure TForm1.Button6Click(Sender: TObject);  
begin  
if SaveDialog1.Execute then  
Memo2.Lines.SaveToFile(SaveDialog1.FileName);  
end;
```

NB: jika button 6 di klik maka akan menyimpan data yang ada di memo 2 ke dalam PC dalam bentuk file .txt



BAB V

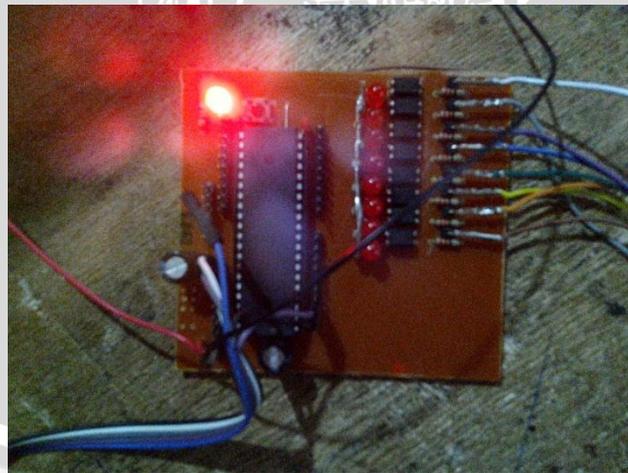
PENGUJIAN

5.1 Penyajian Data

Pada bab ini akan ditampilkan data pengujian selama penelitian yang terdiri dari pengujian rangkaian *logic analyzer*, pengujian komunikasi serial dengan *hypeterminal*, Pengujian komunikasi serial dengan *Delphi7*, pengujian tegangan logika dan pengujian keseluruhan.

5.1.1. Pengujian Rangkaian *Logic Analyzer*

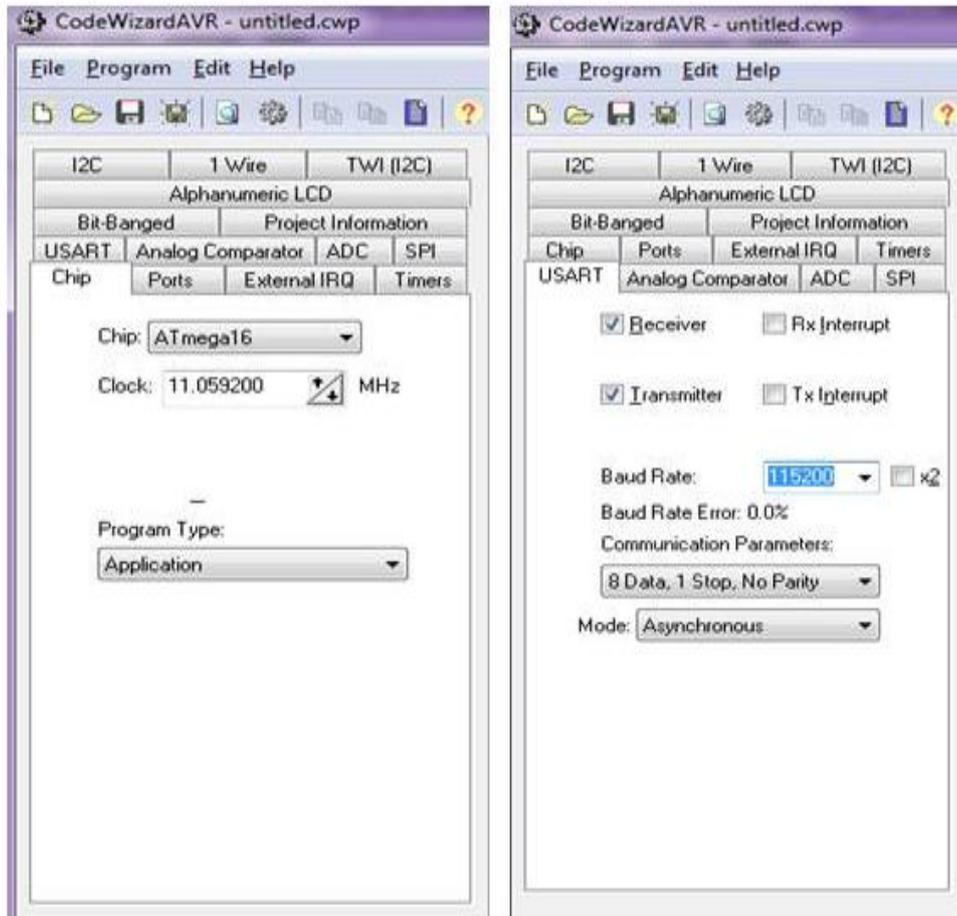
Pengujian rangkaian *logic analyzer* ditunjukkan seperti pada gambar 5.2 dengan cara mengirimkan perintah nyala LED dari PC ke Port C didapatkan hasil seperti gambar5.1.



Gambar 5.1 *Logic Analyzer*

5.1.2 Pengujian Komunikasi Serial dengan Hyperterminal

Proses Inisialisasi Mikrokontroller, Kristal dan *baud rate* diperlukan pengaturan awal pada program CV AVR seperti gambar 5.3



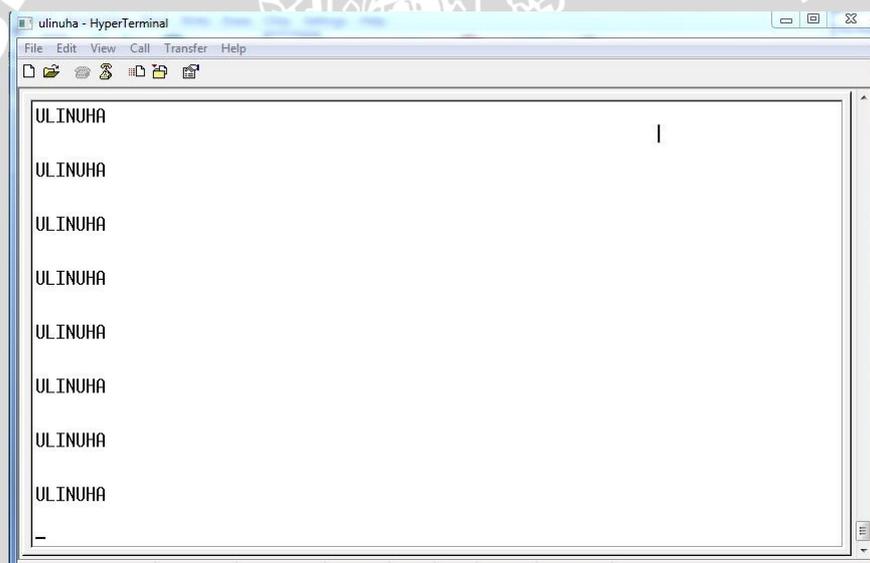
Gambar 5.3 Inisialisasi Mikrokontroler, pengaturan Kristal dan *Baudrate* AT Mega 16

Kecepatan pengiriman data (*baudrate*) yaitu 115200 bps yang diperoleh dari rumusan yaitu:

$$\text{Baudrate} = \frac{f_{osc}}{16(\text{UBRR}+1)}$$

Nilai *system oscillator clock frequency* (f_{osc}) pada Mikroprosesor sebesar 11,0952 MHz. dengan tujuan mendapatkan *error* 0 % dan pengiriman data yang paling cepat yang dapat diterima mikokontroler ATmega16 dan *Delphi 7* maka dapat menggunakan *baud rate* sebesar 115200 bps. Selain itu *baud rate* 115200 bps.

Pengujian komunikasi serial dari sistem minimum ATmega 16 menuju PC dengan *Hyperterminal* dengan mengirimkan kata "ULINUHA". Dari hasil pengujian didapatkan hasil seperti gambar 5.4.



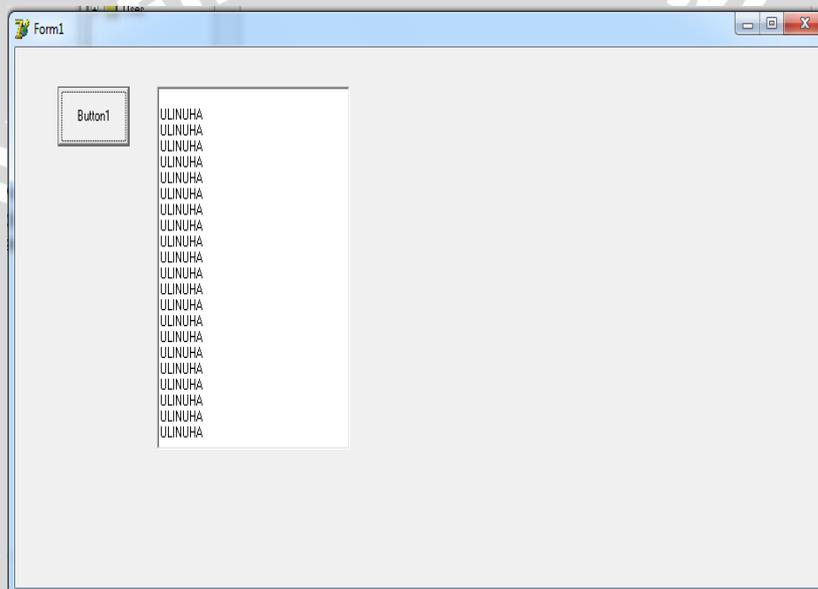
Gambar 5.4 Hasil Pengujian *Hyperterminal* dengan PC

Dari hasil pengujian *Hyperterminal* nilai *Bit Error Rate (BER)* dengan pengiriman karakter "ULINUHA" yaitu :

$$\begin{aligned} \% \text{ BER} &= \frac{\text{Data Error}}{\text{Data Keseluruhan}} \times 100 \% \\ &= \frac{0}{100} \times 100 \% = 0 \% \end{aligned}$$

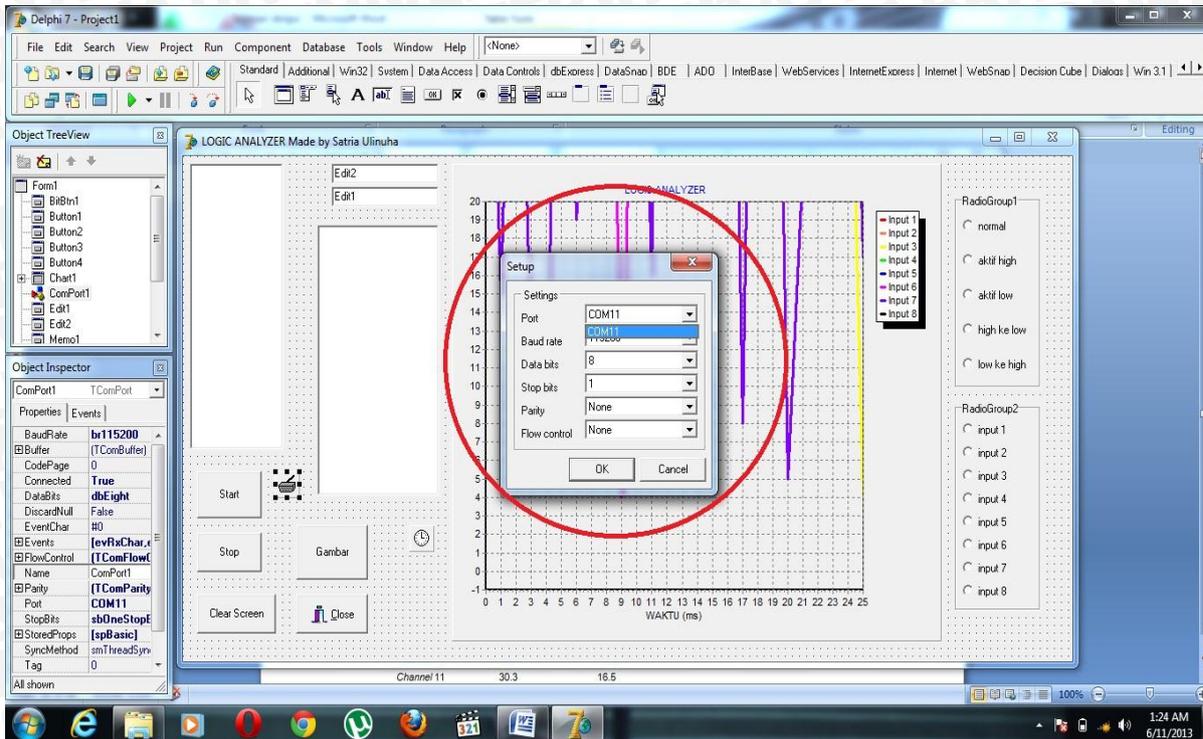
5.1.3 Pengujian Komunikasi Serial dengan Borland Delphi7

Pengujian komunikasi serial dari sistem minimum ATmega 16 menuju PC dengan *Borland delphi 7* dengan mengirimkan kata "ULINUHA" dengan instruksi sama seperti pengujian komunikasi serial dengan PC dan didapatkan hasil seperti gambar 5.5.



Gambar 5.5 Hasil Pengujian *Hyperterminal* dengan *Delphi7*

Untuk *setting* komunikasi serial pada program *Logic Analyzer* yaitu dengan memilih *com* pada *setting com* yang terdeteksi pada PC seperti gambar 5.6



Gambar 5.6 Gambar Tampilan *Set Up* Komunikasi Serial Program *Logic Analyzer* pada *Delphi7*



Gambar 5.7 Gambar Tampilan *Exception notification* ketika ada error

Dari pengujian komunikasi serial diketahui bahwa komunikasi serial sudah dapat digunakan.

5.1.4 Pengujian Tegangan Logika

Pengujian ini dilakukan dengan memberikan tegangan dari 5V menjadi 0V.maka didapatkan hasil sebagai berikut

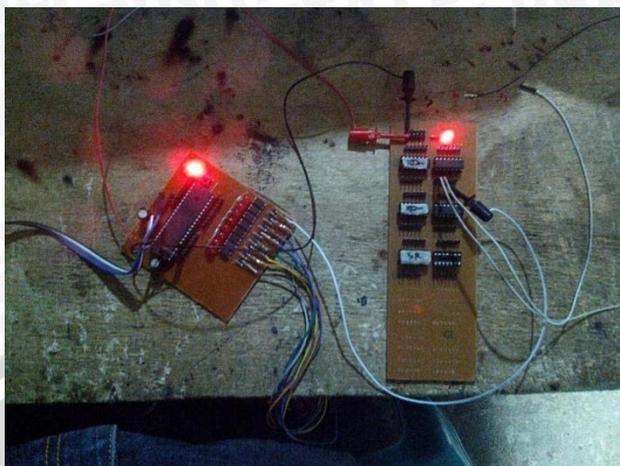
Tabel 4.1.3 Hasil Pengujian Perubahan Tegangan dari 5 volt

Tegangan(V)	Logika
5	1
4.9	1
4.85	1
4.73	1
4.64	1
4.45	1
4.37	1
4.2	1
3.39	1
3.18	0
0	0

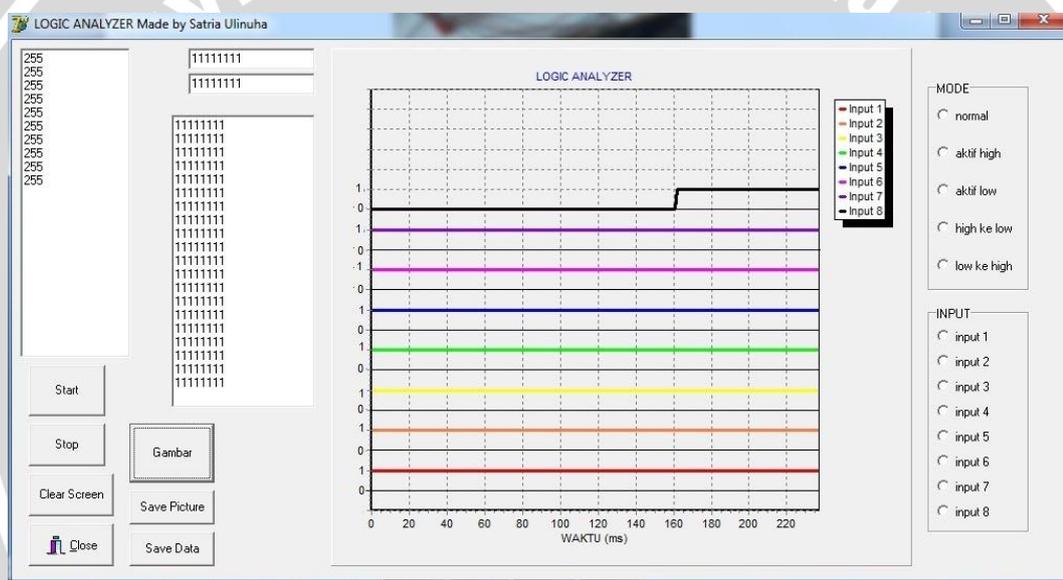
Dari grafik diatas dapat diketahui bahwa channel masukan akan berlogika 1 ketika mendapat tegangan 3.39 – 5 V dan akan berlogika 0 ketika mendapat tegangan 0-3.18V.

5.1.5 Pengujian Program *Logic Analyzer* Pada *Delphi 7*

Pengujian ini menampilkan gambar-gambar hasil pembuatan *software Logic Analyzer* menggunakan *Delphi 7*. Pada *Hardware channel 1-8* dihubungkan dengan rangkaian IC TTL disesuaikan dengan warna kabelnya seperti pada gambar 5.8.



Gambar 5.8 Pengujian *Hardware Logic Analyzer*

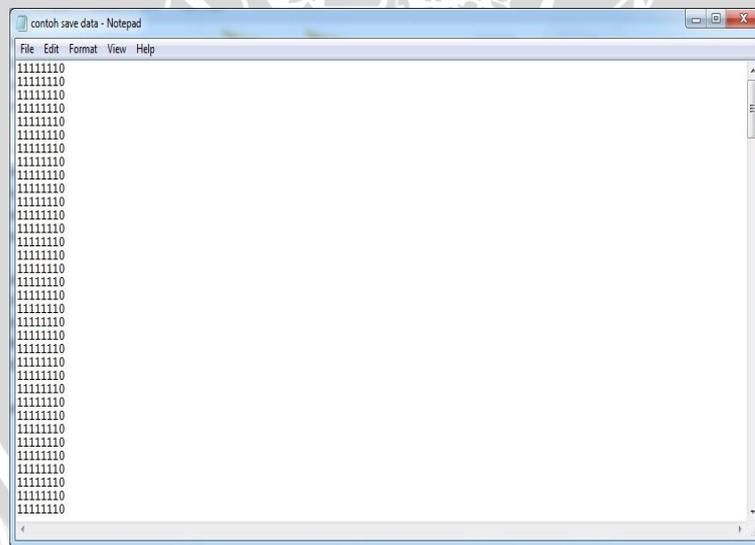


Gambar 5.9 Hasil Pengujian *Hardware Logic Analyzer* pada *Delphi7*

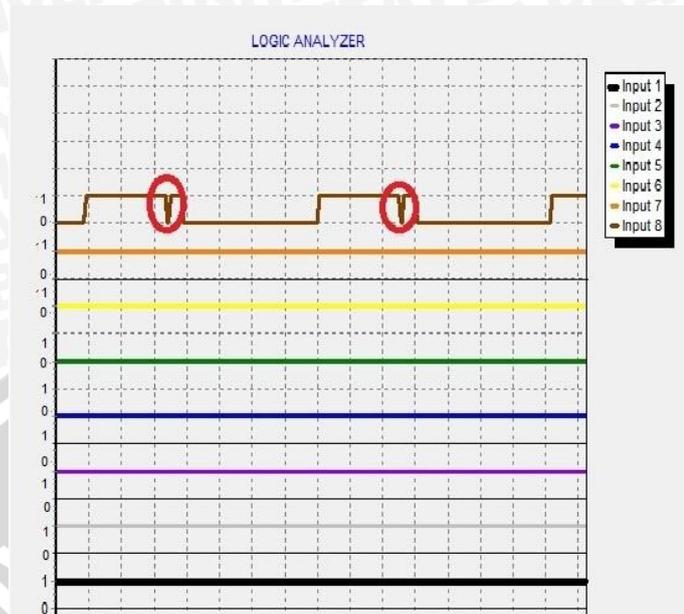
Pada mode pemilihan normal maka signal masukan belum bisa diproses sebelum kita menekan tombol start untuk memulai. Untuk mode aktif high, mode aktif low, mode high ke low dan low ke high signal masukan akan langsung diproses tanpa menekan tombol start terlebih dahulu berdasarkan mode yang dipilih.

Pada aktif high data akan langsung diproses pada keadaan logika '1'. Untuk aktif low data akan langsung diproses pada keadaan logika '0'. Dan mode low ke high akan memproses perubahan logika dari '0' ke logika '1'. Terakhir mode high ke low adalah sebaliknya dari mode low ke high yaitu perubahan logika dari '1' ke logika '0'.

Hasil pengujian dapat disimpan baik data maupun gambar. Jika ingin menyimpan data maka tekan menu '*Save Data*'. Isikan nama file yang diinginkan, data yang tersimpan akan ditampilkan dalam bentuk *Notepad* dan jika ingin menyimpan gambar hasil pengujian maka tekan menu '*Save Picture*'. Isikan nama *image* yang diinginkan. gambar yang tersimpan dalam bentuk bitmap. Pada pengujian ini data dan gambar disimpan dengan nama file 'data' dan gambar 'data'. Seperti yang ditunjukkan pada gambar 5.10 dan gambar 5.11.



Gambar 5.10 Hasil Penyimpanan Data Program *Logic Analyzer* pada *Delphi7*



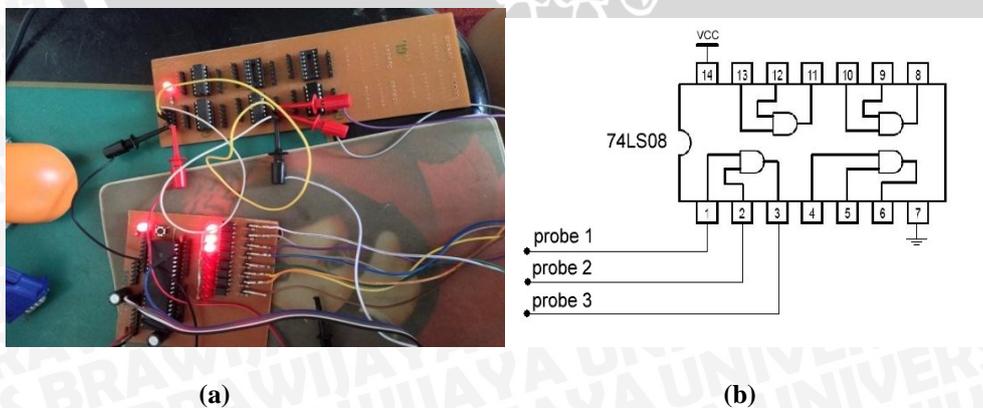
Gambar 5.11 Hasil Penyimpanan Gambar Program *Logic Analyzer* pada *Delphi7*

5.1.6 Pengujian Keseluruhan

Pengujian ini melibatkan semua komponen *hardware* dan *software*. alat hasil rancang bangun digunakan untuk menguji IC dan hasilnya ditampilkan dalam *Logic Analyzer*. Beberapa IC TTL yang telah diuji menggunakan alat hasil rancang bangun adalah:

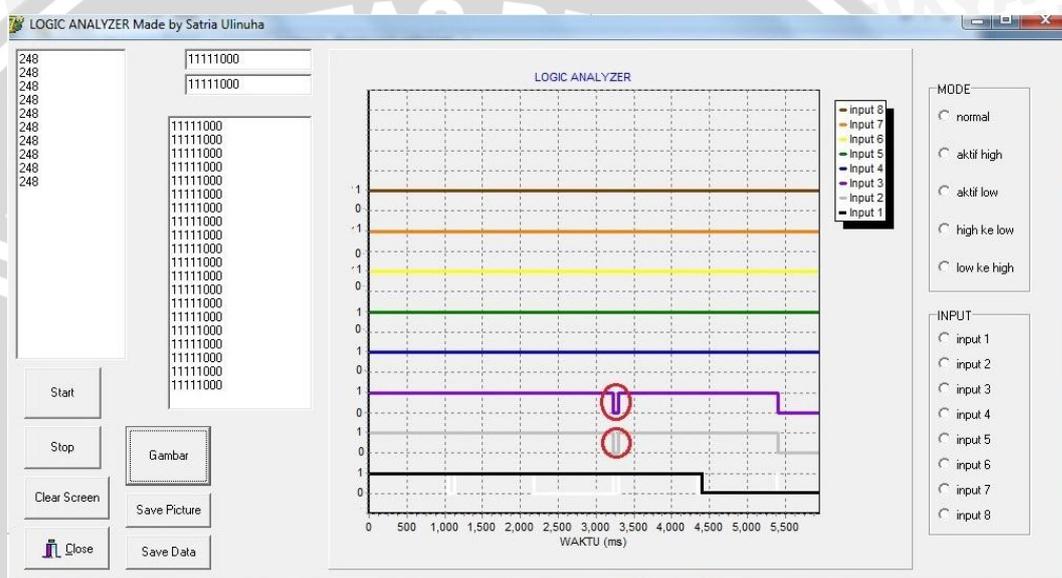
- (1) IC 74LS32 (Gerbang OR)

Pengujian IC ini dilakukan seperti pada gambar:



Gambar 5.12 (a) Pengujian Gerbang OR, (b) Simbol Gerbang OR

Pada pengujian gerbang OR 74LS32, kaki 1 IC diberi logika '0' dan dihubungkan dengan probe *input 1*, kemudian kaki 2 diberi logika '0' dan dihubungkan dengan *input 2* dan kaki 3 dihubungkan dengan *input 3* dimana merupakan keluaran gerbang OR. hasil pengujian dapat dilihat pada gambar berikut ini:



Gambar 5.13 Hasil Pengujian Gerbang OR Pada Program Logic Analyzer

Menurut datasheet tabel logika dari gerbang OR adalah sebagai berikut:

$Y = A + B$

Inputs		Output
A	B	Y
L	L	L
L	H	H
H	L	H
H	H	H

H = HIGH Logic Level
L = LOW Logic Level

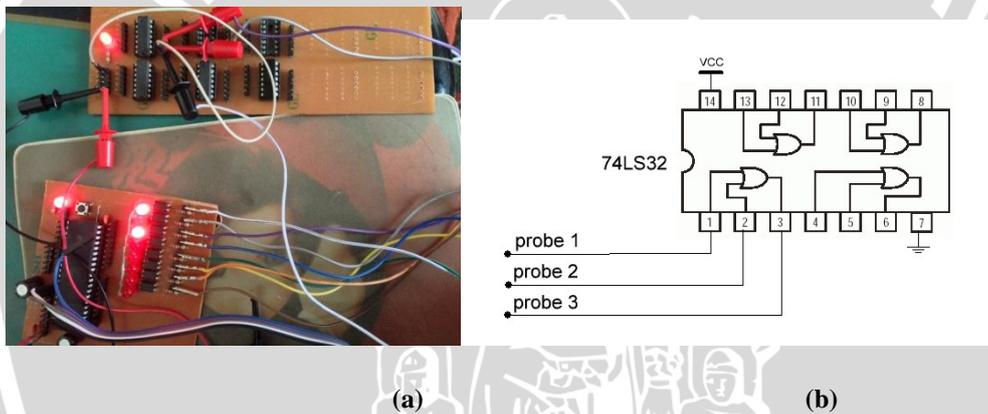
Gambar 5.14 Tabel Logika Gerbang OR (IC74LS32)

(Sumber Fairchild datasheet)

Berdasarkan logika gerbang OR jika dibandingkan dengan gambar grafik 5.14 hasilnya sama. Di rangkaian terbukti probe 1, 2 dan 3 led menyala dimana kondisi ini berlogika '0'. Untuk gambar 5.13 grafik yang dilingkari merah menjelaskan terjadi perubahan tegangan karena pada waktu awal pemasangan probe yang kurang tepat atau pas.

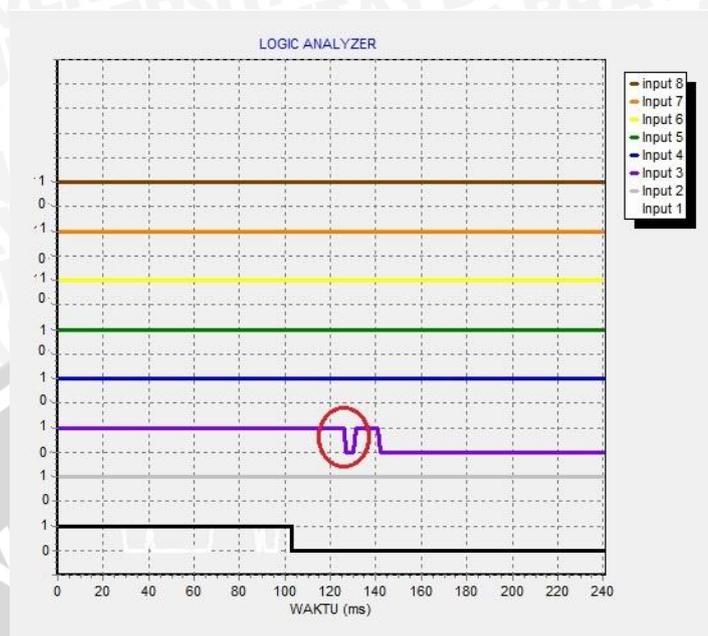
(2) IC 74LS08 (gerbang AND)

Pengujian IC ini dilakukan seperti pada gambar



Gambar 5.15 (a) Pengujian Gerbang AND, (b) IC Gerbang AND

Pada pengujian gerbang AND 74LS08, kaki 1 IC yang diberi logika '0' dihubungkan dengan *input1* dan kaki 2 yang diberi logika '1' dihubungkan dengan *input2*. kaki 3 dihubungkan dengan *probe input 3*. hasil pengujian dapat dilihat pada gambar 5.16.



Gambar 5.16 Hasil pengujian Gerbang AND pada Program Logic Analyzer

Menurut data sheet table logika dari gerbang AND adalah sebagai berikut:

$Y = AB$

Inputs		Output
A	B	Y
L	L	L
L	H	L
H	L	L
H	H	H

H = HIGH Logic Level
L = LOW Logic Level

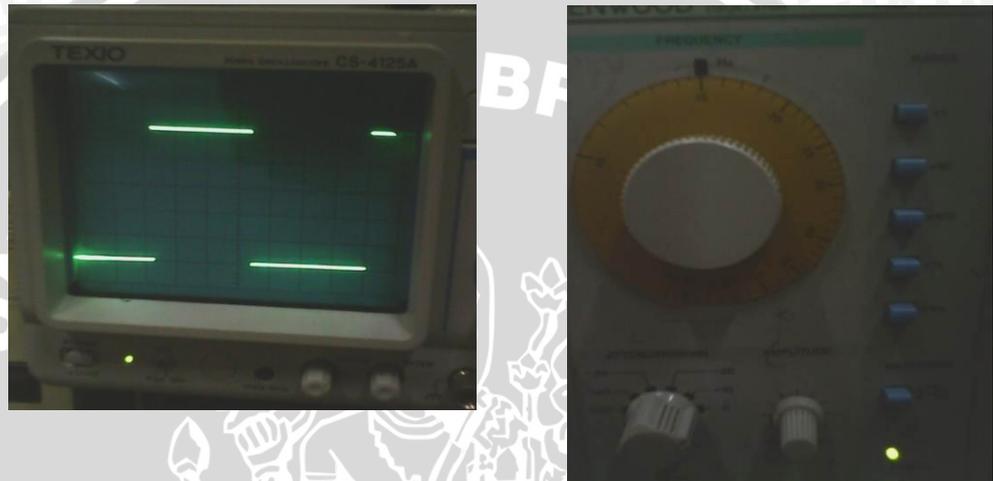
Gambar 5.17 Tabel Logika Gerbang AND (IC74LS08)

(Sumber: Fairchild datasheet)

Berdasarkan logika gerbang AND pada gambar 5.15 jika dibandingkan dengan gambar grafik 5.17 hasilnya sama. Untuk gambar 5.16 grafik yang dilingkari merah menjelaskan terjadi naik turun tegangan karena pada waktu awal pemasangan probe yang kurang tepat atau pas.

5.1.7 Pengujian Kecepatan Alat Dalam Menerima data

Pengujian ini dilakukan dengan memberi masukan pada alat *Logic analyzer* menggunakan *oscilloscop*.. Dihitung frekuensi maksimum yang dapat *logic analyzer* terima. Frekuensi maksimum yang dapat terukur adalah 11.0592 MHz seperti yang ditunjukkan pada gambar 5.18



Gambar 5.18. Gambar pengujian frekuensi menggunakan oskiloskop

5.2 Pembahasan

Alat *Logic Analyzer* berfungsi untuk membaca dan memberi data logika dari suatu IC atau alat elektronika digital. *Logic Analyzer* menggunakan satu sistem minimum ATmega 16 dan satu PC. *Logic Analyzer* ini memiliki 8channelmasukan , 1 channel vcc dan 1 channel ground. Jumlah channel ini memanfaatkan port pada Atmega 16. Port A masing-masing memiliki 8 pin yang dimanfaatkan sebagai channel masukan, sedangkan Port D dimanfaatkan sebagai komunikasi serial. Warna kabel yang berbeda sesuai dengan warna logika pada grafik di simulasi *Logic Analyzer* . *Logic Analyzer* juga dilengkapi indikator berupa led. Untuk data logika masukan led berwarna merah dan *active low*(mati ketika mendapat logika 1 dan nyala ketika mendapat

logika 0). Tegangan logika bernilai 1 ketika mendapat tegangan 3.39 – 5 V dan akan bernilai 0 ketika mendapat tegangan 0-3.18V.

Data logika masukan dibaca dan dikirim melalui komunikasi serial asinkron dengan *baudrate* 115200 bps menuju PC, begitu juga sebaliknya data logika keluaran dari PC dikirim menuju mikrokontroler. Data masukan dan keluaran ditampilkan pada PC dengan menggunakan *software Borland Delphi 7*.

Arah komunikasi alat ini yaitu *Half Duplex*. Pada saat program *Logic Analyzer* dijalankan pertama kali maka harus melakukan *setting* komunikasi yang terdeteksi pada PC. Untuk menjalankan program yang kedua kali dan seterusnya tidak perlu melakukan *setting* komunikasi melainkan langsung dapat menjalankan program. *Mode Real Time* pengambilan data pada program *Logic Analyzer*:

Mode Real Time, pada *mode* ini data yang masuk pada mikro langsung dikirim ke PC dan tertampil pada program *Logic Analyzer*. *Mode* ini digunakan untuk membaca dan memberi logika keluaran IC atau rangkaian elektronika yang tidak menggunakan kecepatan tinggi.

Program *Logic Analyzer* ini memiliki beberapa fasilitas, yaitu:

- a. Mampu menyimpan data hasil pengujian
- b. Mampu menyimpan gambar hasil pengujian
- c. Mampu mengambil data pengujian yang telah tersimpan

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, pengujian dan analisis sistem maka dapat diambil kesimpulan sebagai berikut:

1. *Logic Analyzer* dapat membaca logika keluaran IC dan mengeluarkan logika yang digunakan sebagai masukan IC dan rangkaian elektronika digital.
2. Perangkat lunak dapat mengolah data serta menampilkannya pada komputer dengan beberapa fasilitas yaitu:
 - a. Mampu menyimpan data hasil pengujian dalam format *.txt*
 - b. Mampu menyimpan gambar hasil pengujian dalam format *.bmp*
 - c. Mampu mengambil data pengujian yang telah tersimpan

DAFTAR PUSTAKA

http://www.tektronix.com/logic_analyzers

Atmel. 2002. *AVR182: Zero Cross Detector*. California: Atmel.

Atmel. 2006. *ATmega8535/ATmega8535L, 8-bit AVR Microcontroller with 8 Kbytes in-System Programmable Flash*. California: Atmel.

Mismail, Budiono.1992.*Dasar – Dasar Rangkaian Logika Digital*. Bandung:Penerbit ITB Bandung.

Abdul, H.A.Thamrin.2001. *komunikasi Data dan Komputer*.Jakarta:Salemba Teknika.

Andrianto, Heri. 2008. *Pemrograman Mikrokontroler AVR ATmega16 menggunakan Bahasa C (CodeVision AVR)*. Bandung. Penerbit: Informatika

Arius, Donny.dkk.2008.Komunikasi Data. Yogyakarta: Andi Offset.

Malvino.1992.*Prinsip-Prinsip Elektronik*. Jakarta: Erlangga.

<http://www.allegromicro.com/~Media/Files/Datasheets/ACS712-Datasheet.ashx>

Team Agilent Tecnology Inc.*Felling Comfortable With Logic Analyzer*.2006.USA:

Agilent Technology

Team Agilent Tecnology Inc. *Datasheet Agilent 16800 Series Portable Logic*

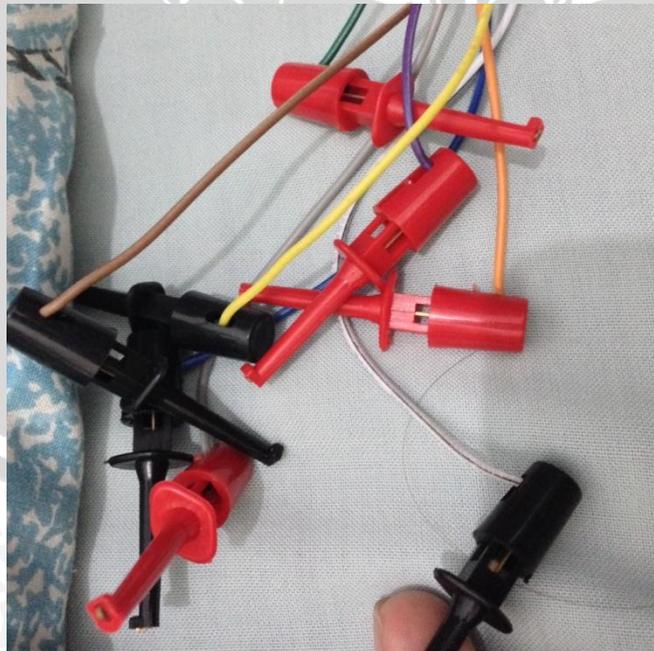
Analyzer.2006.USA: Agilent Technology

LAMPIRAN 1

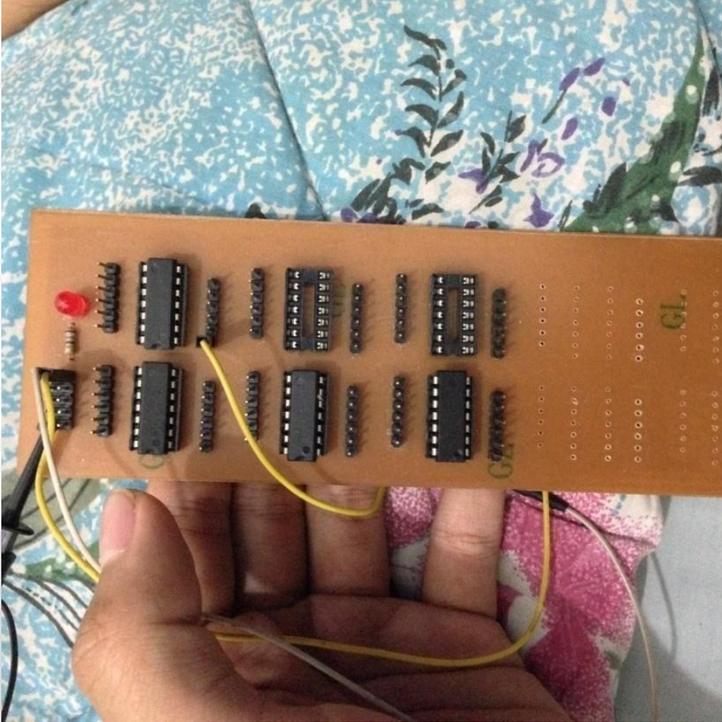
GAMBAR ALAT LOGIC ANALYZER KESELURUHAN



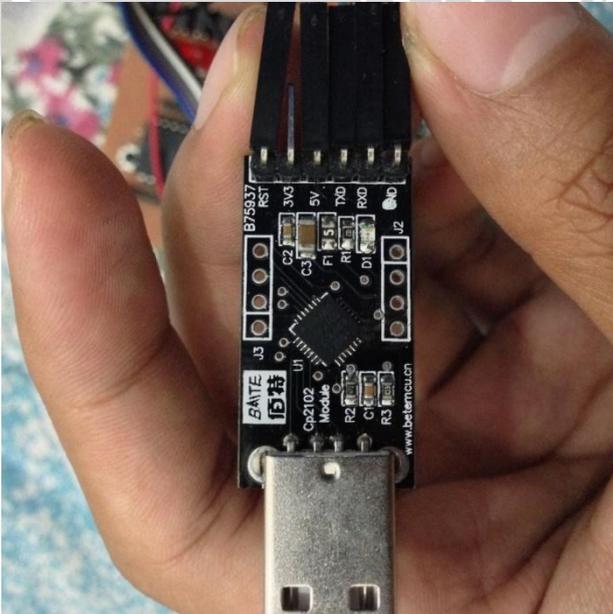
PROBE



RANGKAIAN IC YANG DIUJI



USB TO TTL CP2102



UNIVERSITAS BRAWIJAYA

Datasheet
74LS32



UNIVERSITAS BRAWIJAYA

Datasheet

74LS08

