

Pengembangan Hardware Untuk Praktikum Digital 1 Dalam Remote Laboratory

Nasrullah^{#1}, Edy Satriyanto, S.Si, M.Si-1^{#2},

[#]Jurusan Teknik Elektronika, Politeknik Elektronika Negeri Surabaya
Kampus PENS-ITS Sukolilo, Surabaya

¹elf.nash8@gmail.com

Abstrak—Pesatnya ilmu pengetahuan zaman sekarang mengharuskan setiap individu dapat bersaing dan dapat mengaplikasikan ilmu pengetahuan yang dimiliki untuk memudahkan dan mempercepat suatu penyelesaian masalah. Dengan alasan tersebut maka dibangunlah sebuah **Pengembangan Hardware Untuk Praktikum Digital-1 dalam Remote Laboratory**. Modul ini dibangun atas lima modul praktikum, yaitu Gerbang Logika Dasar, Half Adder, Rangkaian Multilevel NAND dan Rangkaian XOR dan Encoder. Modul ini dibangun dengan menggunakan mikrokontroler ATmega128 dan dikomunikasikan secara serial ke komputer server dengan menggunakan IC MAX232. Hasil yang diperoleh setelah pengujian sesuai dengan yang direncanakan dan sesuai dengan tabel kebenaran dari setiap jenis praktikum, error yang terjadi adalah kerusakan pada led yang terletak pada modul. Dengan dibangunnya modul ini diharapkan dapat membantu dan memperlancar proses belajar-mengajar khususnya pada praktikum Elektronika Digital.

Kata kunci : Modul Praktikum Digital, mikrokontroler.

I. PENDAHULUAN

Perkembangan ilmu pengetahuan pada zaman sekarang ini sangat cepat, khususnya dalam dunia pendidikan. setiap individu dituntut untuk sanggup beradaptasi dengan perkembangan yang ada. dalam setiap perkembangan ilmu pengetahuan ini diharapkan semua akan menjadi semakin mudah, cepat dan flexibel. begitu juga didalam praktikum diharapkan akan mempermudah mahasiswa, dosen maupun teknisi yang menanganinya untuk memudahkannya maka direncanakan suatu alat yang diharapkan dapat membantu proses belajar-mengajar tersebut. dimana alat ini adalah pengembangan hardware untuk praktikum elektronika digital-1 dalam remote laboratory. didalamnya terdapat modul untuk praktikum digital-1 diantaranya cara membangun logika gerbang-gerbang dasar, rangkaian dasar dan penyederhanaan boolean, rangkain dasar dan penyederhanaan dengan karnaugh maap, rangkaian multilevel nand dan nor, rangkaian aritmetika, code converter dan rangkaian kombinasi dengan fungsi khusus. modul praktikum ini dirancang untuk pembelajaran secara jarak jauh yang terkoneksi langsung ke internet dengan menggunakan real-time web yang pada proses akhirnya antar server dapat saling terhubung dengan adanya gambar dan suara. pada prosesnya modul praktikum elektronika digital-1 terlebih dahulu harus ter-connect ke pc dan dari pc inilah data dikirimkan melalui real-time web dan

dengan begitu proses koneksi berlangsung. pembuatan modul praktikum elektronika digital-1 ini diharapkan dapat membantu proses belajar mengajar didalam laboratorium elektronika digital dan memudahkan mahasiswa ataupun dosen dalam melaksanakan praktikum.

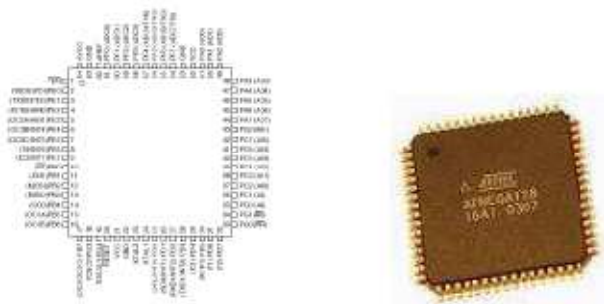
II. TEORI PENUNJANG

Pada tahun 1854 Geoge Boole menciptakan logika simbolik yang sekarang dikenal sebagai aljabar Boole. Setiap peubah (variabel) dalam aljabar Boole hanya memiliki dua keadaan atau dua harga, yaitu keadaan benar yang dinyatakan dengan 1 atau keadaan salah yang dinyatakan dengan 0. Aljabar Boole yang memiliki dua keadaan ini semula dimaksudkan untuk menyelesaikan persoalan-persoalan logika.

Aljabar Boole diwujudkan berupa sebuah piranti atau sistem yang disebut dengan Gerbang Logika. Gerbang Logika adalah blok bangunan dasar untuk membentuk rangkaian elektronika digital, yang digambarkan dengan simbol-simbol tertentu yang telah ditetapkan. Sebuah gerbang logika memiliki beberapa masukan tetapi hanya memiliki satu keluaran. Keluarannya akan HIGH (1) atau LOW (0) tergantung pada level digital pada terminal masukan. Beberapa gerbang logika yang akan dibahas adalah gerbang OR, gerbang logika AND, dan gerbang logika NOT (INVERTER).

2.1 Arsitektur ATmega128

Mikrokontroler ATmega 128 merupakan mikrokontroler keluarga AVR yang mempunyai kapasitas flash memori 128KB. AVR (*Alf and Vegard's Risc Processor*) merupakan seri mikrokontroler CMOS 8-bit buatan Atmel, berbasis arsitektur RISC (*Reduced Instruction Set Computer*). Secara umum, AVR dapat terbagi menjadi 4 kelas, yaitu keluarga ATtiny, keluarga AT90Sxx, keluarga AT-Mega, dan AT86RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah memori, *peripheral*, dan fungsinya. Dari segi arsitektur dan instruksi yang digunakan, bisa dikatakan hampir sama. Semua jenis AVR dilengkapi dengan *flash* memori sebagai memori program. Kapasitas dari *flash* memori ini berbeda antara *chip* yang satu dengan *chip* yang lain. Tergantung dari jenis IC yang digunakan. Untuk *flash* memori yang paling kecil adalah 1 kbytes (ATtiny11, ATtiny12, dan ATtiny15) dan paling besar adalah 128 kbytes (AT-Mega128).



Gambar 2.1 Atmega128

Mikrokontroler AVR ATmega-128 memiliki spesifikasi sebagai berikut:

1. Saluran I/O sebanyak 56 buah, yaitu Port A, Port B, Port C, Port D, Port E, Port F dan Port G.
2. ADC 10 bit sebanyak 8 saluran.
3. 2 buah *Timer/Counter* 8 bit dan 2 buah *Timer/Counter* 16 bit.
4. Dua buah PWM 8 bit.
5. *Watchdog Timer* dengan osilator internal.
6. Internal SRAM sebesar 4 kbyte.
7. Memori *flash* sebesar 128 kBytes.
8. Interupsi Eksternal.
9. Port antarmuka SPI.
10. EEPROM sebesar 4 kbyte.
11. Real time counter.
12. 2 buah Port USART untuk komunikasi serial.
13. Enam kanal PWM.
14. Tegangan operasi sekitar 4,5 V sampai dengan 5,5V.

2.1.1 USART

Pada mikrokontroler ATMEGA128 menggunakan serial komunikasi USART. Sehingga Data Rx berfungsi sebagai pengirim dan penerima data sedangkan Tx sebagai output clock. Output dari serial pada mikrokontroler ATMEGA 128 berupa tegangan TTL 0 – 5 Volt.

2.2 Komunikasi Serial

Dalam komunikasi data serial, data dikirim dalam bentuk pulsa kontinyu yang disebut bit. Data dikirim bit per bit secara berurutan melalui kanal komunikasi yang telah ditentukan. Penerima juga menerima data dalam bentuk bit-bit yang sama.

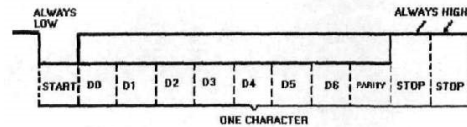
Ada tiga metode yang sering dijumpai pada komunikasi data serial, yaitu *simplex*, *half duplex*, dan *full duplex*. Pada transmisi data *simplex* data hanya dapat dikirim satu arah saja, sedangkan *half duplex* data dapat ditransmisikan secara bergantian dan transmisi *full duplex* merupakan transmisi dua arah dimana data dapat diterima oleh system dan system tersebut dapat mengirimkan data secara

bersamaan. Transmisi *full duplex* dapat dijumpai pada sistem telepon.

Dalam komunikasi data serial, ada dua metode dasar yang digunakan yaitu komunikasi serial asinkron dan komunikasi serial sinkron.

2.2.1 Komunikasi Serial Asinkron

Komunikasi serial asinkron adalah komunikasi data yang memerlukan start bit untuk menunjukkan mulainya data dan stop bit untuk menunjukkan selesainya data. Dalam teknik komunikasi serial asinkron, serial output line selalu dalam keadaan logika '1' (mark) kecuali pada saat data dikirim dan keadaan ini dinamakan *marking*. Setiap karakter dimulai dengan satu bit logika '0' (low). Bit yang pertama dinamakan start bit dan digunakan untuk sinkronisasi antara pengirim dan penerima.



Gambar 2.8 komunikasi serial asinkron

2.2.2 Komunikasi Serial Sinkron

Telah dijelaskan bahwa pada komunikasi asinkron, setiap data memerlukan start bit tanda awali dan stop bit sebagai tanda akhir karakter. Dengan demikian untuk setiap karakter sebesar 8 bit data maka harus dikirim minimum 10 bit (1 bit start, 8 bit data, 1 bit stop). Sehingga terlihat bahwa 20% dari waktu transmisi terbuang. Bila data yang dikirim dalam jumlah besar, kerugian ini semakin terasa. Alternative lain untuk menambahkan kecepatan transmisi adalah dengan komunikasi data sinkron. Pada mode ini setiap karakter tidak dikirim secara terpisah dengan dibatasi oleh start bit dan stop bit. Melainkan karakter dikirim dalam blok data yang dibatasi oleh karakter sinkronisasi. Jumlah karakter sinkronisasi tersebut dapat deprogram.

2.2.3 Protokol Komunikasi Serial

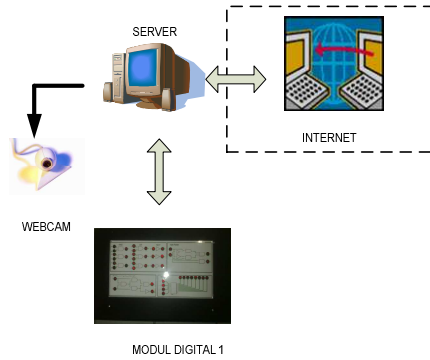
Dalam komunikasi serial, pengirim dan penerima harus memenuhi protocol komunikasi. Berikut ini beberapa istilah protocol komunikasi serial RS232

1. Boud periode (Boudrate)
Kecepatan transmisi yang diukur dalam satuan bit per sekon. *Clock* pengirim dan penerima harus disinkronisasi pada boud periode yang sama.
2. Marking State
Periode waktu selama data tidak dikirim, selama kondisi marking, output line pengirim selalui logic '1' (*High*).
3. Start bit
Logika '0' (*Low*) menunjukkan data transmisi dimulai. Kondisi *Low* pada starting bit dinamakan *spacing state*.
4. Karakter bit
Berisi data dengan jumlah 5, 6, 7 atau 8 bit. Bit pertama dikirim adalah LSB.
5. Parity bit
Parity bit adalah bit pilihan yang dikirim setelah karakter bit untuk mendeteksi *error* transmisi. Ada dua macam *parity*, *parity* genap dan *parity* ganjil.

- 6. Stop bit
Sebagai tanda akhir data yang dikirimkan.

III. PERANCANGAN SYSTEM

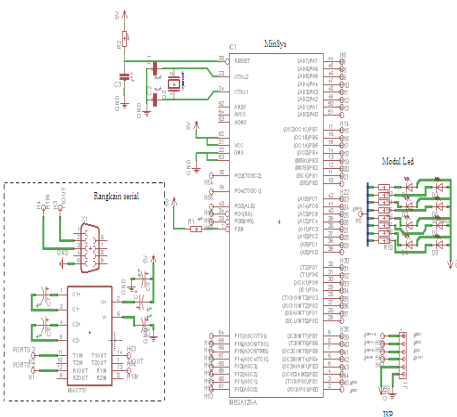
Perencanaan sistem ini terdiri dari pembuatan *hardware* yang terdiri dari rangkaian *minimum system* ATmega 128 dan rangkaian serial serta perancangan box untuk praktikum digital.



Gambar 3.1 Diagram Sistem

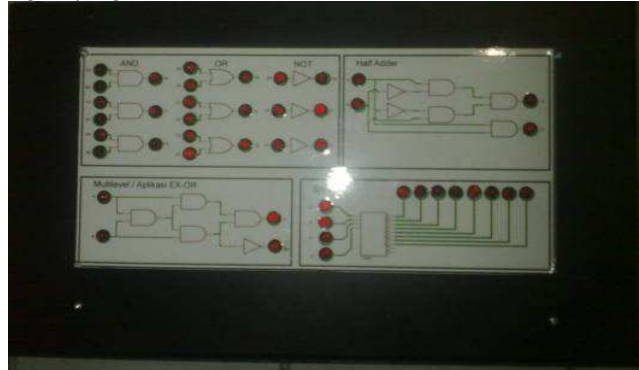
3.1 Perencanaan Hardware

Pada tahap ini dilakukan perencanaan *hardware* yang terdiri dari rangkaian minimum system ATmega128, rangkaian led serta rangkaian komunikasi serial RS232. Perancangan perangkat keras (*hardware*) dimulai dengan merancang rangkaian minimum sistem mikrokontroler yang terdiri atas rangkaian mikrokontroler serta rangkaian pembangkit clock eksternal kemudian rangkaian komunikasi serial RS232. Perancangan rangkaian dilakukan dengan bantuan software Eagle yang juga dapat digunakan sebagai pembuat layout PCB. Setelah skema rangkaian terbuat dilanjutkan ke tahap berikutnya yakni membuat layout PCB yang juga menggunakan software Eagle. Pada tahap berikutnya adalah pembuatan PCB dan perangkaian komponen. Apabila semua telah selesai yang terakhir adalah merancang dan menanam program pada mikrokontroler agar dapat bekerja sesuai dengan yang diinginkan. Gambar keseluruhan rangkaian dapat dilihat dibawah ini :



Gambar 3.2 Rangkaian Sistem

Pada rangkaian diatas semua rangkaian *led* yang berjumlah 44 terhubung secara *Pull-up* langsung ke port mikrokontroler. Dengan memanipulasi I/O pada mikrokontroler maka fungsi dari mikrokontroler tersebut menjadi fungsi gerbang-gerbang logika yang akan diakses oleh server.

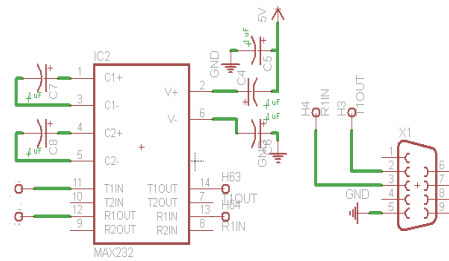


Gambar 3.3 Tampak atas Modul praktikum Digital I

Pada gambar diatas terdapat 3 gerbang AND 2 input, 3 gerbang OR 2 input, 3 gerbang NOT serta rangkaian *Multilevel NAND*, rangkaian *Half adder* dan rangkaian *Encoder*. Sedangkan untuk komunikasi data dari mikrokontroler menggunakan IC RS232 yaitu komunikasi secara serial. Metode yang digunakan adalah komunikasi serial asinkron yang mana memerlukan start bit untuk menunjukkan mulainya data dan stop bit untuk menunjukkan selesainya data. Untuk lebih memudahkan membuat fungsi gerbang logika pada mikrokontroler maka terlebih dahulu mengartur konfigurasi port yang telah dihubungkan dengan rangkaian *led* dengan tampilan pada modul. Berikut adalah konfigurasi port untuk setiap praktikum digital 1:

3.2 Perencanaan Komunikasi

Untuk memudahkan pengiriman data, maka direncanakan protokol. Metode komunikasi yang digunakan adalah metode komunikasi asinkron.

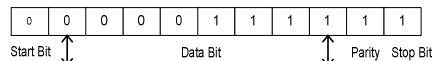


Gambar 3.4 Rangkaian Serial RS232

Susunan protokol dapat dilihat dibawah ini :

- 1. Boud periode (Boudrate) yang digunakan adalah 9600

2. Start bit '0'
3. Karakter bit berisi data berjumlah 8 bit.
4. Parity bit
6. Stop bit sebagai tanda akhir data yang dikirimkan.

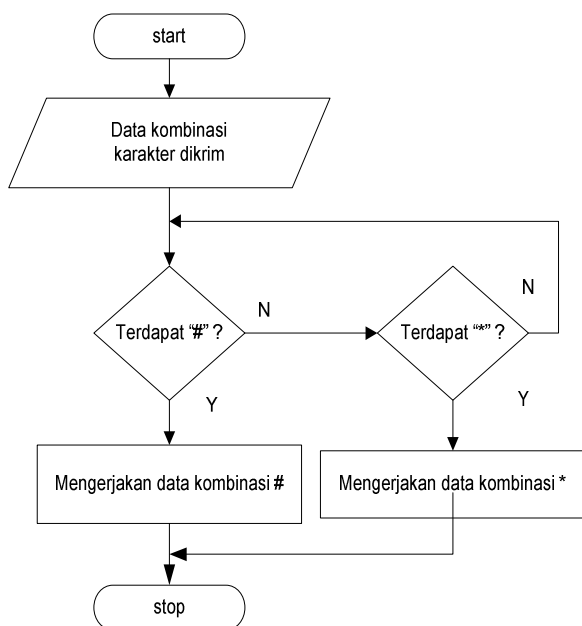


Gambar 3.5 Protokol komunikasi serial

IV. PENGUJIAN

Pada tahap pengujian dilakukan pengujian komunikasi serial dari mikrokontroler ke komputer. Setiap pengujian dilakukan secara bertahap samapai didapat kondisi hardware yang benar. Berikut ini adalah gambar praktikum yang disimulasikan dengan *visual basic 6.0*.

Dari pengujian pengiriman data kombinasi maka dapat dianalisa bahwa pengiriman data kombinasi berhasil dilakukan, flowchart pengiriman data kombinasi dapat dilihat dibawah ini



Gambar 4.1 Flowchart Pengiriman Kombinasi Data

Dari flowchart maka dapat dijelaskan algoritma program nya adalah ketika mikrokontroler menerima data dari komunikasi serial maka mikro akan membandingkan, jika kombinasi data yang diterima terdapat karakter tanda pagar (#) maka akan menjalankan program yang telah dikondisikan, dan jika data yang diterima terdapat karakter bintang (*) maka akan menjalankan program yang telah dikondisikan. Data karakter pagar (#) dan karakter bintang (*) terletak diawal data kombinasi yang dikirimkan. Dari pengujian ini led yang dihubungkan ke mikrokontroler berhasil diaktifkan.

Untuk potongan programnya dapat dilihat dibawah ini :

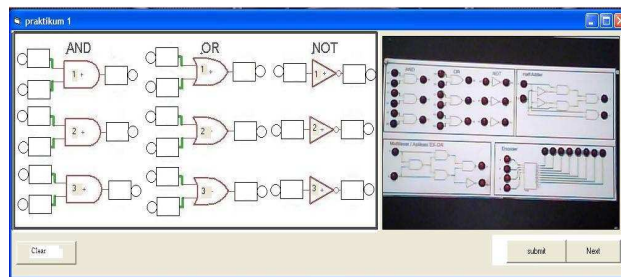
```

while (1)
{
switch(rx_buffer1[0])
{
if(rx_buffer1[0]=='#')
{
switch(rx_buffer1[1])
{
case 'A':  A0=on;  break;
}
}
}

if(rx_buffer1[0]=='*')
{
switch(rx_buffer1[1])
{

case '1':  PORTF=0b11111110;  break; //AM
case '2':  PORTF=0b11111111;  break; //AM
}
}
}
}

```



Gambar 4.2 Tampilan Praktikum 1

Dari gambar diatas hasil yang terjadi sesuai dengan yang diharapkan yaitu pada praktikum 1 gerbang AND maka hasilnya sesuai dengan tabel kebenaran gerbang AND, dan pada gerbang OR hasilnya sama dengan tabel kebenarannya serta pada gerbang NOT hasilnya juga sama dengan table kebenarannya

Tabel 4.1 Hasil Pengujian Gerbang AND

| INPUT | | OUTPUT |
|-------|---|--------|
| A | B | Y |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

Gerbang AND adalah hubungan yang menghasilkan output 1 apabila semua input berlogika 1. Apabila salah satu atau semua input terputus maka output rangkaian menjadi 0.

Rangkaian AND dapat dianalogikan sebagai rangkaian seri dari beberapa inputnya.

| | | | |
|---|---|---|---|
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

Tabel 4.2 Hasil Pengujian Gerbang OR

| INPUT | | OUTPUT |
|-------|---|--------|
| A | B | C |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

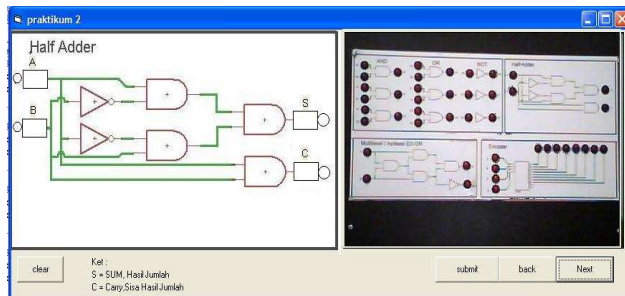
Rangkaian OR adalah hubungan yang menghasilkan output 1 apabila salah satu atau semua input berlogika 1. Untuk menghasilkan logika 0 maka semua input harus 0. Rangkaian OR dapat dianalogikan sebagai rangkaian parallel dari beberapa inputnya.

Tabel 4.3 Hasil Pengujian Gerbang NOT

| INPUT | | OUTPUT |
|-------|---|--------|
| A | B | C |
| 0 | 0 | 0 |
| 0 | 1 | 1 |

Yang membedakan hasil dari modul praktikum ini dengan IC TTL adalah setiap IC TTL mempunyai *propagation delay* yang berbeda dengan delay pada mikrokontroler.

Pada praktikum 2 yaitu praktikum half adder hasil dari pengujian sama dengan tabel kebenaran. Half adder mempunyai dua input biner (A, B) dan dua output biner (S, C). S merupakan hasil penjumlahan dan C adalah sisa. Half Adder Adalah suatu operasi penjumlahan dua bit biner tanpa menyertakan carry-in nya. Dikarenakan menggunakan mikrokontroler maka tegangan pada setiap led sama yaitu 1.89 volt. Untuk gambar GUI praktikum 2 dapat dilihat dibawah ini.



Gambar 4.3 Tampilan Praktikum 2

Tabel 4.4 Hasil Pengujian Rangkaian Half Adder

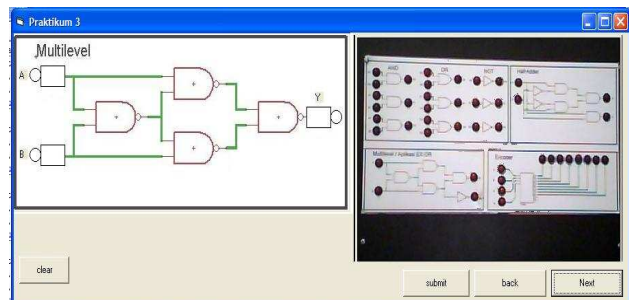
| INPUT | | OUTPUT | |
|-------|---|--------|---|
| A | B | S | C |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |

Pada praktikum 3 yaitu praktikum multilevel yang dibangun menggunakan gerbang NAND kondisi-kondisi yang terjadi adalah sesuai dengan tabel kebenaran rangkaian sesungguhnya, logika yang terjadi adalah dikendalikan melalui software dan dikirimkan pada mikrokontroler sehingga indikator pada alat membuktikan hasil dari perhitungan logika yang diberikan, selanjutnya trainer indicator itu sendiri mengirimkan umpan balik pada software sehingga software dapat menindikasikan output yang terjadi sebenarnya. Hal ini telah membuktikan table hasil pengujian pada tabel 4.6 dibawah ini

Tabel 4.5 Hasil Pengujian Rangkaian Multilevel NAND

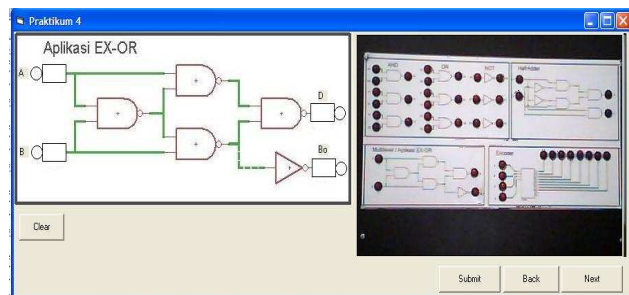
| INPUT | | OUTPUT |
|-------|---|--------|
| A | B | C |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Untuk gambar GUI praktikum 3 dapat dilihat dibawah ini .



Gambar 4.4 Tampilan Praktikum 3

Pada pengujian praktikum 4 yaitu praktikum aplikasi EX-OR yang dibangun dari gerbang NAND untuk mengecek jumlah dari bilangan 1 dalam suatu kode bilangan itu ganjil atau genap disebut parity checker. Maka kondisi tersebut berhasil dicapai pada perancangan modul ini. Untuk gambar GUI dan tabel hasil dari pengujian dapat dilihat dibawah ini.



Gambar 4.5 Tampilan Praktikum 4

Tabel 4.6 Hasil Pengujian Rangkaian Aplikasi EX-OR

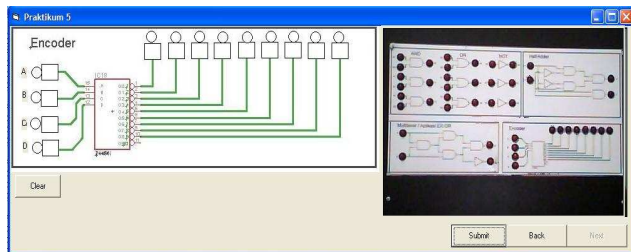
| INPUT | | OUTPUT | |
|-------|---|--------|----------------|
| A | B | D | D ₀ |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

Pada pengujian praktikum 5 yaitu praktikum encoder desimal ke biner (BCD) yang memiliki sembilan input dan empat output berjalan sesuai dengan tabel kebenaran. logika yang terjadi adalah dikendalikan melalui software dan dikirimkan pada mikrokontroler sehingga indikator pada alat membuktikan hasil dari perhitungan logika yang diberikan, selanjutnya trainer indicator itu sendiri mengirimkan umpan balik pada software sehingga software dapat menindikasikan output yang terjadi sebenarnya. Hal ini telah membuktikan table hasil pengujian pada tabel 4.8 dibawah ini

Tabel 4.7 hasil Pengujian Encoder

| Input | | | | | | | | | Output | | | |
|-------|---|---|---|---|---|---|---|---|--------|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | D | C | B | A |
| L | H | H | H | H | H | H | H | H | H | H | H | L |
| X | L | H | H | H | H | H | H | H | H | H | L | H |
| X | X | L | H | H | H | H | H | H | H | H | L | L |
| X | X | X | L | H | H | H | H | H | H | L | H | H |
| X | X | X | X | L | H | H | H | H | H | L | H | L |
| X | X | X | X | X | L | H | H | H | H | L | L | H |
| X | X | X | X | X | X | L | H | H | H | L | L | L |
| X | X | X | X | X | X | X | L | H | L | H | H | H |
| X | X | X | X | X | X | X | X | L | L | H | H | L |

Untuk gambar GUI praktikum 5 dapat dilihat dibawah ini



Gambar 4.6 Tampilan Praktikum 5

Untuk semua data komunikasi yang dikirimkan server dikarenakan menggunakan komunikasi serial asinkron dan baudrate yang digunakan 9600 bit per second maka 1 bit membutuhkan 1/9600 detik atau 0,000104 detik atau 104 uS (microsecond= 10⁻⁶ det

V. KESIMPULAN

Setelah melakukan tahap perancangan sampai dengan pengujian, maka dapat ditarik kesimpulan sebagai berikut.

- Dikarenakan banyaknya I/O mikrokontroler yang harus di komunikasikan melalui komunikasi serial maka untuk mengatasi masalah tersebut karakter yang dikirimkan server dibagi menjadi dua paket pengiriman, yaitu paket pengiriman data yang diawali dengan tanda pagar (#) dan paket pengiriman data yang diawali dengan tanda bintang (*).
- Untuk PORT F dan PORT G tidak bisa diprogram per bit ini dikarenakan tidak adanya library pada software compiler *Codevision* untuk ATmega128, jadi bit-bit yang akan diaktifkan akan dimanipulasi dengan program OR dan program AND.
- Ada beberapa output yang terjadi *error* pada saat modul dijalankan, ini disebabkan oleh program pada mikrokontroler tidak berjalan semestinya , sebab yang lain adalah rusaknya *led* pada modul.

DAFTAR PUSTAKA

- [1] CodeVisionAVR, help.
- [2] Datasheet ATmega128
- [3] Fletcher, William I; 1980; *Engineering approach to Digital Design*; PHI; America
- [4] Kristalina, Prima, dkk; 2003; *Praktikum Elektronika Digital Program Diploma IV*; PENS-ITS
- [5] Manual Book *Digital Trainer Experiment Manual*; LABTECH
- [6] McGraw-Hill, Tata; 2005; *2000 Solved Problems in Digitals Electronics*; Sigma Series; New Delhi
- [7] Uyemura, John P; 1999; *Digital System Design*; ITP; USA