



**TUGAS AKHIR - TE 145561**

## **Sistem *Billing* pada Meja Billiard**

Muhammad Nurudin  
NRP 10311500000053

Dosen Pembimbing  
Slamet Budiprayitno S.T.,MT

PROGRAM STUDI KOMPUTER KONTROL  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**FINAL PROJECT - TE 145561**

***Billing System on Billiard Table***

Muhammad Nurudin  
NRP 1031150000053

Supervisor  
Slamet Budiprayitno S.T.,MT

COMPUTER CONTROL STUDY PROGRAM  
Electrical and Automation Engineering Department  
Vocational Faculty  
Sepuluh Nopember Institute of Technology  
Surabaya 2018

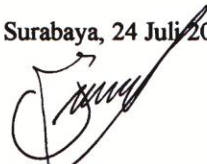
## PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**Sistem *Billing* pada Meja *Billiard***” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 24 Juli 2018



Muhammad Nurudin  
NRP 1031150000053

--- Halaman ini sengaja dikosongkan ---

**SISTEM BILLING PADA MEJA BILLIARD**

**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Memperoleh Gelar Ahli Madya Teknik**

**Pada  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember**

**Menyetujui:**

**Dosen Pembimbing :**

**Slamet Budi Prayitno S.T., MT**

**NIP. 197811132010121002**

**SURABAYA  
JULI, 2018**

--- Halaman ini sengaja dikosongkan ---

## **Sistem *Billing* pada Meja *Billiard***

**Nama : Muhammad Nurudin**  
**Pembimbing : Slamet Budiprayitno S.T.,MT**

### **ABSTRAK**

Pada bisnis penyewaan meja *billiard* yang ada, masih banyak yang menggunakan sistem pencatatan manual. Dalam pencatatan manual ini proses perhitungan tarif bisa dimanipulasi dan tidak dapat dipantau secara *real time* sehingga laporan transaksi tidak valid. Manipulasi data bisa berupa pengurangan biaya tarif atau penambahan waktu penyewaan dengan tarif normal atau kedua-duanya. Dengan demikian semua pihak akan dirugikan. Kerugian dari pihak pemilik yaitu pendapatan yang tidak sesuai dengan transaksi yang ada, sedangkan dari pihak konsumen tidak puas karena ketidakcocokan antara perhitungan waktu yang disewa dengan perhitungan waktu yang berjalan. Pada sistem *biling billiard* ini menggunakan mikrokontroler sebagai antar muka antara perangkat komputer dengan lampu meja *billiard*. Tipe mikrokontroler yang digunakan adalah ATmega16. Dalam aplikasi perangkat lunak, menggunakan LabVIEW sebagai HMI (*Human Machine Interface*). Dalam perangkat lunak ini, sistem mampu menyalakan atau mematikan lampu pada meja *billiard* dan melakukan proses *billing*. Proses *billing* dimulai saat lampu pada meja *billiard* dinyalakan, dan akan berhenti ketika lampu pada meja *billiard* dimatikan. Selain mikrokontroler, perangkat tambahan yang digunakan adalah rangkaian relay dan optocoupler. Hasil dari pengujian yang telah dilakukan, dalam rentan waktu 1 sampai 60 menit, kondisi lampu padam sesuai dengan input yang diberikan pada HMI (*human machine interface*), sedangkan pengujian dalam rentan waktu 90 sampai 360 menit terdapat *delay* selama 2 detik. *Delay* ini terlihat ketika indikator lampu pada HMI telah mati, selang 2 detik lampu pada *prototipe* meja *billiard* baru padam.

Kata kunci: Sistem *Billing*, *Billiard*, LabVIEW.

--- Halaman ini sengaja dikosongkan ---



## ***BILLING SYSTEM ON BILLIARD TABLE***

**Name : Muhammad Nurudin**  
**Supervisor : Slamet Budiprayitno S.T.,MT**

### ***ABSTRACT***

*In billiard desk rental business that exist, there are use manual system. In this manual system, the cost calculation process can be manipulated and can't be monitored in real time so the transaction report is invalid. Data manipulation may include cost reduction or additional rental time at normal rates or both. Losses from the owner of the income that is not in accordance with the existing transactions. While the consumer is not satisfied because of the mismatch between the calculation of time rented with the calculation of running time. In this billiard billing system using microcontroller as interface between computer device with billiard table lamp. The type of microcontroller used is ATmega16. For software applications, using LabVIEW. In this software, the system is able to turn on or turn off the lights on the billiard table and do the billing process. The billing process begins when the lights on the billiard table are turned on, and will stop when the lights on the billiard table are turned off. In addition to the microcontroller, the enhancements used optocoupler and relay. The results of the tests that have been performed, within 1 to 60 minutes, the lamp condition goes off according to the input given to the HMI (human machine interface), whereas the test within 90 to 360 minutes is delayed for 2 seconds. This delay is visible when the light indicator on the HMI have died, after 2 seconds the lights on the protocol of the billiard table went off.*

*Keywords : Billing System, Billiard, LabVIEW*

--- Halaman ini sengaja dikosongkan ---

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma-3 pada Program Studi Komputer Kontrol, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

### **“Sistem *Billing* pada Meja *Billiard*”**

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Slamet Budiprayitno S.T., MT. selaku dosen pembimbing atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 24 Juli 2018

Penulis

--- Halaman ini sengaja dikosongkan ---

## DAFTAR ISI

Halaman judul .....	i
Pernyataan keaslian Tugas akhir .....	iii
Persetujuan Dosen Pembimbing .....	v
Abstrak .....	vii
Abstrac .....	ix
Kata pengantar .....	xi
Daftar isi .....	xiii
Daftar gambar .....	xv
Daftar tabel .....	xvii
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar belakang .....	1
1.2 Permasalahan .....	2
1.3 Batasan masalah .....	2
1.4 Tujuan .....	2
1.5 Metodologi penelitian .....	2
1.6 Sistematika laporan .....	3
<b>BAB II TEORI DASAR</b> .....	<b>5</b>
2.1 ATmega16 .....	5
2.1.1 Arsitektur ATmega16 .....	6
2.1.2 Konfigurasi pin ATmega16 .....	6
2.2 Relay .....	9
2.3 Optocoupler .....	12
2.4 MCB (Miniature Circuit Breaker) .....	13
2.4.1 <i>Thermal Tripping</i> .....	15
2.4.2 <i>Magnetic Tripping</i> .....	15
2.5 Fuse .....	16
2.6 CodeVision AVR .....	17
2.7 <i>Billing</i> .....	17
2.8 Billiard .....	18
2.9 Komunikasi Serial .....	18
2.9.1 USB to TTL PL2303 .....	19
2.10 LabVIEW .....	20
<b>BAB III PERANCANGAN ALAT</b> .....	<b>25</b>
3.1 Perancangan Hardware .....	27
3.1.1 Perancangan Rangkaian Relay dan Optocoupler .....	28
3.1.2 Desain alat .....	30
3.2 Perancangan Software .....	35

3.2.1	Flowchart Program .....	35
3.2.2	Perancangan program pada CodeVisionAVR .....	36
3.2.3	Perancangan HMI menggunakan LabVIEW .....	38
BAB IV	UJI UKUR DAN UJI ALAT .....	41
4.1	Pengujian Relay.....	41
4.2	Pengujian Alat dan Program .....	45
BAB V	PENUTUP.....	47
5.1	Kesimpulan .....	47
5.2	Saran .....	47
Daftar Pustaka	.....	49
Lampiran A	<i>Listing</i> program pada CodeVisionAVR.....	51
Lampiran B	<i>Screenshot</i> HMI pada LabVIEW .....	57
Lampiran C	Datasheet ATMega16 .....	59
Lampiran D	Datasheet modul relay 4 channel.....	65
Lampiran E	Datasheet PL2303 .....	71
Boidata Penulis	.....	81

## DAFTAR GAMBAR

Gambar 2.1 ATmega16.....	5
Gambar 2.2 Susunan kaki ATmega16.....	7
Gambar 2.3 Relay .....	9
Gambar 2.4 Bagian-bagian Relay.....	10
Gambar 2.5 Modul Relay 4 <i>Channel</i> .....	11
Gambar 2.6 <i>Pole</i> dan <i>Throw Relay</i> .....	12
Gambar 2.7 Simbol dan bentuk-bentuk Optocoupler .....	12
Gambar 2.8 Rangkaian dalam Optocoupler .....	13
Gambar 2.9 MCB .....	14
Gambar 2.10 <i>Thermal Tripping</i> .....	15
Gambar 2.11 <i>Magnetic Tripping</i> .....	15
Gambar 2.12 Fuse .....	16
Gambar 2.13 Tampilan awal CodeVision AVR.....	17
Gambar 2.14 USB to TTL PL2303 .....	19
Gambar 2.15 <i>Front panel</i> pada LabVIEW.....	20
Gambar 2.16 Blok diagram pada LabVIEW.....	21
Gambar 2.17 <i>Control pallette</i> pada LabVIEW .....	21
Gambar 2.18 <i>Function pallette</i> pada LabVIEW.....	22
Gambar 2.19 Konfigurasi VISA untuk port serial.....	22
Gambar 2.20 Konfigurasi pada port serial .....	23
Gambar 3.1 Skema proses sistem <i>billing</i> .....	25
Gambar 3.2 Skema proses sistem <i>billing</i> .....	27
Gambar 3.3 Schematic rangkaian relay dan optocoupler .....	29
Gambar 3.4 Layout PCB rangkaian optocoupler dan relay .....	29
Gambar 3.5 Rangkaian relay dan optocoupler .....	30
Gambar 3.6 Akrilik.....	31
Gambar 3.7 Desain bagian belakang .....	31
Gambar 3.8 Desain bagian depan .....	32
Gambar 3.9 Desain bagian samping dan atas.....	32
Gambar 3.10 Desain bagian bawah .....	33
Gambar 3.11 Bagian depan alat .....	33
Gambar 3.12 Bagian belakang alat.....	34
Gambar 3.13 Bagian dalam alat .....	34

Gambar 3.14 Flowchart program.....	36
Gambar 3.15 Pemrograman menggunakan CodeVisionAVR.....	37
Gambar 3.16 HMI menggunakan LabVIEW.....	38



## **DAFTAR TABEL**

Tabel 2.1 Fungsi Khusus Port B Atmega 16.....	7
Tabel 2.2 Fungsi Khusus Port C Atmega 16.....	8
Tabel 2.3 Fungsi Khusus Port D Atmega 16.....	8
Tabel 3.1 Fungsi-fungsi indikator pada HMI.....	38
Tabel 4.1 Pengujian relay 1.....	41
Tabel 4.2 Pengujian relay 2.....	42
Tabel 4.3 Pengujian relay 3.....	43
Tabel 4.4 Pengujian relay 4.....	44
Tabel 4.5 Pengujian Alat dan Program.....	45

--- Halaman ini sengaja dikosongkan ---

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Berbicara di bidang teknologi, maka kita akan mendapatkan suatu perkembangan yang sangat pesat dimana hampir semua bidang pekerjaan dilakukan dengan menggunakan komputer. Dengan menggunakan komputer, maka suatu pekerjaan dapat terselesaikan dengan cepat dan tepat. Maka dari pada itu dapat dikatakan bahwa komputer adalah suatu alat yang sangat penting di era saat ini.

Dengan adanya komputer, kita dapat mengolah data dengan tepat dan cepat serta dapat mengontrol suatu alat. Untuk itu dibutuhkan teknologi guna meningkatkan mutu kerja dan pelayanan sehingga dapat terciptanya efisiensi kerja.

Perkembangan tempat hiburan saat ini juga semakin banyak, seperti pusat-pusat perbelanjaan dan makin menjamurnya tempat-tempat hiburan. Mulai dari tempat hiburan khusus anak-anak, remaja, orang dewasa bahkan untuk keluarga. Tapi seiring dengan perkembangan tempat hiburan, tidak sepenuhnya diikuti dengan perkembangan teknologi untuk membantu usaha bisnis hiburan.

Perkembangan usaha bisnis tempat hiburan tidak lagi hanya sebatas diskotik atau tempat-tempat rekreasi, tapi lebih luas lagi. Seperti halnya dengan dunia olahraga juga bisa dijadikan salah satu bisnis di dunia hiburan. Contohnya adalah permainan olahraga billiard. Billiard adalah salah satu permainan olah raga yang dimainkan oleh 2 (dua) orang atau lebih. Dalam perkembangannya billiard bukan lagi menjadi permainan yang hanya dimainkan oleh orang dewasa maupun atlet olah raga billiard, tapi permainan billiard saat ini banyak dimainkan oleh anak-anak remaja, pelajar maupun mahasiswa.

Bisnis billiard semakin-hari semakin berkembang. Hal ini dapat dilihat dengan semakin banyaknya tempat persewaan billiard yang menyediakan penyewaan tempat dan fasilitas untuk permainan billiard.

Setelah melakukan observasi pada beberapa tempat persewaan billiard, penulis melihat masih adanya kekurangan yaitu masih banyak tempat persewaan billiard yang belum menyediakan aplikasi khusus yang digunakan untuk mengontrol waktu permainan dan biaya sewa pemakaian meja billiard. Untuk tempat persewaan billiard yang belum menggunakan aplikasi, masih banyak terjadi kesalahan dalam pencatatan biaya dan waktu penyewaan. Hal ini sangat merugikan

pemilik usaha persewaan billiard karena berpengaruh terhadap pendapatan serta memungkinkan terjadinya kesalahan dalam pembuatan laporan, selain itu penggunaan listrik yang tidak terkontrol dapat mengakibatkan pemborosan biaya.

Melihat permasalahan tersebut, penulis berinisiatif untuk membuat suatu program aplikasi *billing* persewaan meja billiard yang dapat membantu pemilik usaha persewaan billiard dalam mengontrol waktu permainan dan biaya sewa pemakaian meja untuk permainan billiard serta mengontrol pemakaian lampu untuk masing-masing meja.

## **1.2 Permasalahan**

Identifikasi masalah yang didapat berdasarkan dari uraian tersebut diatas yaitu:

1. Belum tersedianya aplikasi yang mampu mengontrol waktu permainan dan biaya sewa pemakaian meja untuk permainan billiard serta mengontrol pemakaian lampu untuk masing-masing meja.
2. Penggunaan fasilitas oleh pihak yang tidak melakukan penyewaan, yang dapat mengakibatkan kerugian dalam pendapatan serta pemborosan dalam penggunaan listrik.
3. Masih banyaknya terjadi kesalahan dalam pencatatan waktu dan biaya sewa yang mengakibatkan kesalahan dalam pembuatan laporan.

## **1.3 Batasan Masalah**

Perancangan dan pembuatan program aplikasi *billing* billiard dibatasi pada penggunaan untuk 4 (empat) meja, dan juga untuk lampu meja billiard yang digunakan menggunakan prototipe.

## **1.4 Tujuan**

Penelitian ini bertujuan untuk untuk membuat Aplikasi Billing Billiard yang dapat memberi kemudahan dalam pengontrolan lama waktu permainan dan biaya sewa pemakaian meja untuk permainan billiard serta mengontrol pemakaian lampu untuk masing-masing meja.

## **1.5 Metodologi Penelitian**

Dalam pelaksanaan penelitian ini penulis melaksanakan beberapa tahapan, yaitu:

1. Pengumpulan Data
  - a. Penelitian Literatur.

Pada langkah pertama ini penulis mempelajari dan meneliti hal-hal yang berhubungan dengan permasalahan, melalui buku-buku dan referensi yang ada.
  - b. Penelitian Lapangan  

Pada langkah kedua ini, penulis mengumpulkan data-data yang berhubungan dengan sistim penyewaan melalui observasi dit tempat persewaan meja billiard.
2. Analisa Data  

Pada tahap ini penulis melakukan analisa terhadap data-data atau dokumen yang telah diperoleh, dimana untuk selanjutnya data-data tersebut akan digunakan dalam pembuatan program aplikasi.
3. Membuat Program Aplikasi  

Pada tahap ini penulis melakukan proses pembuatan program aplikasi dengan menggunakan *LabView*, sesuai dengan hasil analisa dari data-data yang telah diperoleh.

## 1.6 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

### **Bab I      Pendahuluan**

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, dan sistematika laporan.

### **Bab II     Teori Dasar**

Pada bab ini akan dibahas teori dasar dan teori penunjang terkait perangkat dan bahan yang digunakan dalam Tugas Akhir ini.

### **Bab III    Perancangan Sistem**

Membahas perencanaan dan pembuatan perangkat keras yang meliputi rangkaian-rangkaian, desain prototipe, dan perangkat lunak yang akan digunakan untuk menjalankan alat tersebut

### **Bab IV    Uji Ukur dan Uji Alat**

Pada bab ini dilakukan pengujian dari sistem yang telah dibuat pada bab 3.

## **Bab V Penutup**

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

## **BAB II**

### **TEORI DASAR**

Beberapa teori penunjang yang dipaparkan dalam buku Tugas Akhir ini adalah teori dasar mengenai ATMega16, relay, optocoupler, MCB (*Miniature Circuit Braker*), fuse, power supply, dan Visual Basic, CV AVR, *billing*, billiard.

#### **2.1 ATMega16**

Mikrokontroler adalah sebuah sistem komputer lengkap dalam satu chip. Mikrokontroler lebih dari sekedar sebuah mikroprosesor karena sudah terdapat atau berisikan ROM (*Read-Only Memory*), RAM (*Read-Write Memory*), beberapa plot masukan maupun keluaran, dan beberapa peripheral seperti pencacah/pewaktu, ADC (*Analog to Digital converter*), DAC (*Digital to Analog converter*) dan serial komunikasi.

Mikrokontroler AVR (*Alf and Vegard's Risc processor*) merupakan pengontrolan utama standar industri dan riset saat ini. Hal ini dikarenakan berbagai kelebihan yang dimilikinya dibandingkan mikrokontroler antara lain lebih murah, dukungan software dan dokumentasi yang memadai dan memerlukan komponen pendukung yang sangat sedikit. Salah satu tipe mikrokontroler AVR untuk aplikasi standar yang memiliki fitur memuaskan ialah ATMega16.

Mikrokontroler AVR standar memiliki arsitektur 8 bit, dimana semua instruksi dikemas dalam kode 16 bit. Secara internal mikrokontroler ATMega16 terdiri atas unit-unit fungsionalnya ALU (*Arithmetic and Logical Unit*), himpunan register kerja, register dan decoder instruksi, dan pewaktu beserta komponen kendali lainnya.



**Gambar 2.1** ATMega16

### 2.1.1 ARSITEKTUR ATMEGA16

Mikrokontroler ini menggunakan arsitektur *Harvard* yang memisahkan memori program dari memori data, baik port alamat maupun port data, sehingga pengaksesan program dan data dapat dilakukan secara bersamaan (*concurrent*). Fitur-fitur yang dimiliki ATmega16 sebagai berikut:

- 1) Mikrokontroler AVR 8 Bit yang memiliki kemampuan tinggi, dengan daya rendah.
- 2) Memiliki kapasitas Flash memori 16 KByte.
- 3) Saluran I/O sebanyak 32 buah, yaitu Port A, Port B, Port C dan Port D.
- 4) CPU terdiri atas 32 register.
- 5) Unit Interupsi internal dan eksternal.
- 6) ADC internal dengan *fidelitas* 10 bit 8 channel.
- 7) Sistem mikroprosesor 8 bit berbasis RISC dengan kecepatan maksimal 16 MHz.
- 8) Port USART untuk komunikasi serial.

Dengan fitur-fitur seperti diatas, pembuatan alat menggunakan ATmega16 menjadi lebih sederhana dan tidak memerlukan IC pendukung yang banyak.

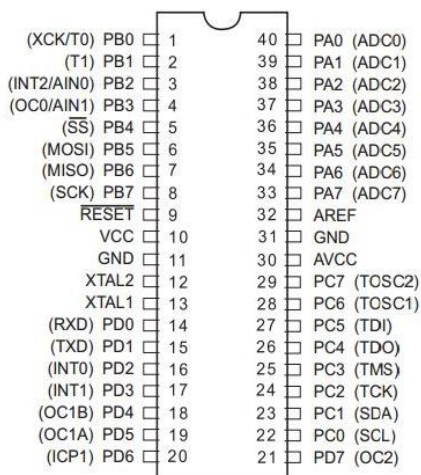
### 2.1.2 KONFIGURASI PIN ATMEGA16

Konfigurasi pin mikrokontroler Atmega16 dengan kemasan 40 pin dapat dilihat pada gambar 2.2. ATmega16 memiliki 32 pin yang digunakan untuk input/output. Pin-pin tersebut terdiri dari 8 pin sebagai port A, 8 pin sebagai port B, 8 pin sebagai port C dan 8 pin sebagai port D. Dalam komunikasi serial, maka hanya port D yang dapat digunakan kerana fungsi khusus yang dimilikinya. Berikut ini adalah penjelasan umum susunan kaki dari ATmega16:

1. VCC merupakan pin masukan positif catu daya. Setiap peranti elektronika digital membutuhkan sumber daya yang umumnya sebesar 5V
2. GND sebagai pin Ground,
3. Port A (PA0-PA7),  
Merupakan pin I/O dua arah dan dapat diprogram sebagai pin masukan ADC. Port A berfungsi sebagai input analog pada konverter A/D. Port A juga sebagai suatu port I/O 8-bit dua arah, jika A/D konverter tidak digunakan.



Untuk lebih jelas akan ditunjukkan pada tabel 2.1. Berikut ini adalah susunan pin Mikrokontroler ATmega16 :



**Gambar 2.2** Susunan kaki ATMega16

4. Port B (PB0-PB7),  
Merupakan pin I/O dua arah dan pin fungsi khusus, yaitu *Timer/Counter*, komparator analog dan SPI. Port B adalah suatu port I/O 8-bit dua arah dengan resistor internal pull-up (yang dipilih untuk beberapa bit)

**Tabel 2.1.** Fungsi Khusus Port B Atmega 16

Port Pin	Fungsi Khusus
PB 7	SCK: SPI serial waktu
PB 6	MISO: SPI master input / slave output
PB 5	MOSI: SPI master output / slave input
PB 4	SS: SPI slave select input
PB 3	AIN1: pembanding analog, input negatif OC0: Timer / counter 0 output (pembanding output)
PB 2	AIN0: Pembanding analog, input positif INT2: <i>External interrupt 2</i> input
PB 1	T1: Timer / counter1 external counter input

PB 0	T0: Timer / counter 0 external counter input XCK: USART waktu eksternal input / output
------	---

5. Port C (PC0-PC7),

Merupakan pin I/O dua arah dan pin fungsi khusus, yaitu komparator TWI, komparator analog dan timer osilator. Port C adalah suatu port I/O 8-bit dua arah dengan resistor *internal pull-up*,

**Tabel 2.2** Fungsi Khusus Port C Atmega 16

Port Pin	Fungsi Khusus
PC 7	OC2: Timer / counter 2 output (hasil output)
PC 6	ICP1: Timer / counter 1 input Pin
PC 5	OC1A: Timer / counter 1 hasil output A (hasil output A)
PC 4	OC1B: Timer / counter 1 hasil output B (hasil Output B)
PC 3	INT1: <i>External interrupt</i> 1 input
PC 2	INT0: <i>External interrupt</i> 0 input
PC 1	TXD: USART output Pin
PC 0	RXD: USART input Pin

6. Port D (PD0-PD7),

Merupakan pin I/O dua arah dan pin fungsi khusus, yaitu komparator analog, interupsi eksternal dan komunikasi serial. Port D adalah suatu port I/O 8-bit dua arah dengan resistor *internal pull-up*,

**Tabel 2.3** Fungsi Khusus Port D Atmega 16

Port Pin	Fungsi Khusus
PD 7	OC2: Timer / counter 2 output (hasil output)
PD 6	ICP1: Timer / counter 1 input Pin
PD 5	OC1A: Timer / counter 1 hasil output A (hasil output A)
PD 4	OC1B: Timer / counter 1 hasil output B (hasil Output B)
PD 3	INT1: <i>External interrupt</i> 1 input
PD 2	INT0: <i>External interrupt</i> 0 input
PD 1	TXD: USART output Pin
PD 0	RXD: USART input Pin

7. Reset merupakan pin yang digunakan untuk me-reset mikrokontroler,
8. XTAL 1 dan XTAL 2 sebagai pin masukan clock eksternal. Suatu mikrokontroler membutuhkan sumber detak (clock) agar dapat mengeksekusi instruksi yang ada di memori. Semakin tinggi nilai kristalnya maka semakin cepat mikrokontroler tersebut,
9. AVCC sebagai pin masukan tegangan untuk ADC,
10. AREF sebagai pin masukan tegangan referensi.

## 2.2 Relay

Relay adalah Saklar (Switch) yang dioperasikan secara listrik dan merupakan komponen *Electromechanical* (Elektromekanikal) yang terdiri dari 2 bagian utama yakni Elektromagnet (Coil) dan Mekanikal (seperangkat Kontak Saklar/Switch). Relay menggunakan prinsip elektromagnetik untuk menggerakkan kontak saklar sehingga dengan arus listrik yang kecil (low power) dapat menghantarkan listrik yang bertegangan lebih tinggi.

Sebagai contoh, Relay 5V: 50 mA mampu menggerakkan Armature Relay (yang berfungsi sebagai saklar) untuk menghantarkan listrik 220V 2A. Karena Relay merupakan salah satu jenis dari saklar, maka istilah *pole* dan *throw* yang dipakai dalam saklar juga berlaku pada relay. *Pole* yaitu banyaknya kontak (Contact) yang dimiliki oleh sebuah relay, sedangkan *throw* adalah banyaknya kondisi yang dimiliki oleh sebuah kontak (Contact).

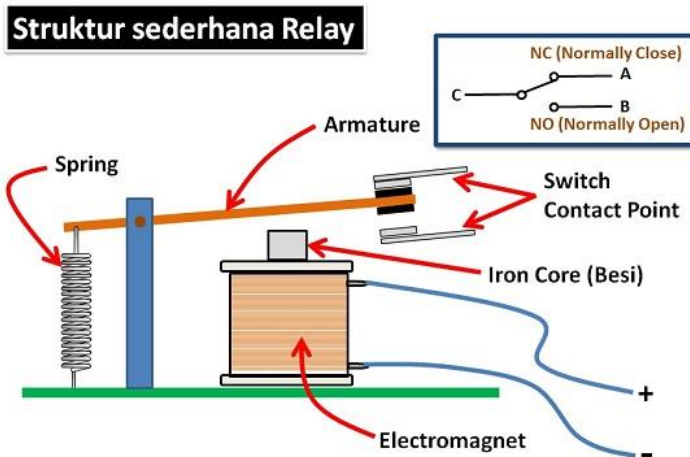


**Gambar 2.3** Relay

Pada dasarnya, Relay terdiri dari 4 komponen dasar yaitu :

1. Elektromagnet (Coil)
2. *Armature*
3. *Switch Contact Point* (Saklar)
4. *Spring*

Pada Gambar 2.4 merupakan gambar dari bagian-bagian Relay.



Gambar 2.4 Bagian-bagian relay

Kontak Poin (*Contact Point*) Relay terdiri dari 2 jenis yaitu :

1. *Normally Close* (NC) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi *close* (tertutup).
2. *Normally Open* (NO) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi *open* (terbuka)

Berdasarkan gambar 2.4, sebuah Besi (*Iron Core*) yang dililit oleh sebuah kumparan berfungsi untuk mengendalikan besi tersebut. Apabila kumparan *coil* diberikan arus listrik, maka akan timbul gaya elektromagnet yang kemudian menarik *armature* untuk berpindah dari posisi sebelumnya (NC) ke posisi baru (NO) sehingga menjadi saklar yang dapat menghantarkan arus listrik di posisi barunya (NO). Posisi dimana *Armature* tersebut berada sebelumnya (NC) akan menjadi *open* atau tidak terhubung. Pada saat tidak dialiri arus listrik, *armature* akan kembali lagi ke posisi Awal (NC). Coil yang digunakan oleh relay untuk menarik *contact point* ke posisi *close* pada umumnya hanya membutuhkan arus listrik yang relatif kecil.

Relay yang digunakan untuk Tugas Akhir ini yaitu Modul Relay 4 channel, dimana modul relay ini dapat digunakan sebagai *switch* untuk menjalankan lampu AC. Kendali *ON / OFF switch* (relay), sepenuhnya ditentukan oleh mikrokontroler. Relay merupakan salah satu jenis dari saklar, maka istilah *pole* dan *throw* yang dipakai dalam saklar juga berlaku pada relay. Berdasarkan penggolongan jumlah *pole* dan *throw*-nya, sebuah modul *relay* tersebut dapat digolongkan dalam golongan *relay* SPDT karena memiliki 1 *Common*, 1 *Normally Open* (NO), dan 1 *Normally Close* (NC). Berikut pada Gambar 2.5 adalah gambar Modul Relay 4 Channel.



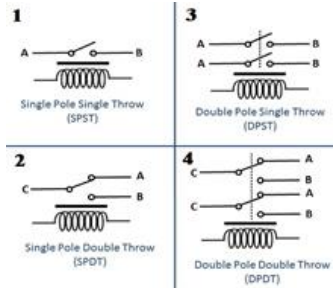
**Gambar 2.5** Modul Relay 4 Channel

Terdapat pula golongan lainnya yaitu :

1. *Single Pole Single Throw* (SPST) : Relay golongan ini memiliki terminal 4 terminal, 2 terminal untuk *saklar* dan 2 terminal untuk *coil*.
2. *Double Pole Single Throw* (DPST) : Relay golongan ini memiliki 6 terminal, diantaranya 4 terminal yang terdiri dari 2 pasang terminal saklar, sedangkan 2 terminal lainnya untuk *coil*.
3. *Double Pole Double Throw* (DPDT) : Relay golongan ini memiliki 8 terminal, diantaranya 6 terminal yang merupakan 2

pasang *Relay SPDT* yang dikendalikan oleh 1 *coil*, sedangkan 2 terminal lainnya untuk *coil*.

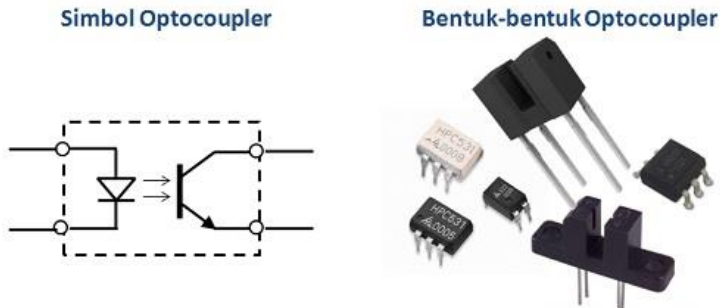
Golongan *relay* berdasarkan *pole* dan *throw* dapat dilihat pada Gambar 2.6.



**Gambar 2.6** Pole dan Throw Relay

### 2.3 Optocoupler

Dalam dunia elektronika, optocoupler juga dikenal dengan sebutan opto-isolator, photocoupler atau optical Isolator. Optocoupler adalah komponen elektronika yang berfungsi sebagai penghubung berdasarkan cahaya optik. Pada dasarnya optocoupler terdiri dari 2 bagian utama yaitu transmitter yang berfungsi sebagai pengirim cahaya optik dan receiver yang berfungsi sebagai pendeteksi sumber cahaya. Masing-masing bagian optocoupler (transmitter dan receiver) tidak memiliki hubungan konduktif rangkaian secara langsung tetapi dibuat sedemikian rupa dalam satu kemasan komponen.

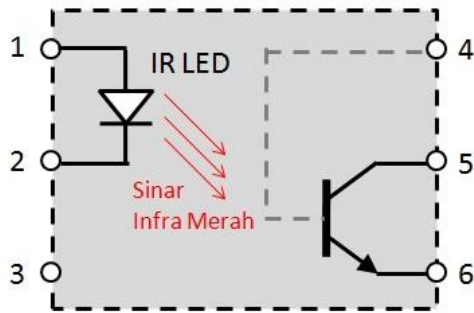


**Gambar 2.7** Simbol dan bentuk-bentuk optocoupler

Jenis-jenis optocoupler yang sering ditemukan adalah optocoupler yang terbuat dari bahan semikonduktor dan terdiri dari

kombinasi LED (*light emitting diode*) dan phototransistor. Dalam kombinasi ini, LED berfungsi sebagai pengirim sinyal cahaya optik (Transmitter) sedangkan phototransistor berfungsi sebagai penerima cahaya tersebut (Receiver).

Pada prinsipnya, optocoupler dengan kombinasi LED-phototransistor adalah optocoupler yang terdiri dari sebuah komponen LED (*light emitting diode*) yang memancarkan cahaya inframerah (IR LED) dan sebuah komponen semikonduktor yang peka terhadap cahaya (phototransistor) sebagai bagian yang digunakan untuk mendeteksi cahaya infra merah yang dipancarkan oleh IR LED. Untuk lebih jelas mengenai Prinsip kerja Optocoupler, dapat dilihat pada gambar 2.6 dibawah ini :



**Gambar 2.8** Rangkaian dalam optocoupler

Dari gambar 2.6, dapat dijelaskan bahwa arus listrik yang mengalir melalui IR LED akan menyebabkan IR LED memancarkan sinyal cahaya Infra merah. Intensitas cahaya tergantung pada jumlah arus listrik yang mengalir pada IR LED tersebut.

Cahaya infra merah yang dipancarkan tersebut akan dideteksi oleh phototransistor dan menyebabkan terjadinya hubungan atau *switch ON* pada phototransistor. Prinsip kerja phototransistor hampir sama dengan transistor bipolar, yang membedakan adalah terminal basis (*base*) phototransistor merupakan penerima yang peka terhadap cahaya.

#### **2.4 MCB (*Miniature Circuit Breaker*)**

MCB (*Miniature Circuit Breaker*) atau miniatur pemutus sirkuit adalah sebuah perangkat elektromekanikal yang berfungsi sebagai

pelindung rangkaian listrik dari arus yang berlebihan. Dengan kata lain, MCB dapat memutuskan arus listrik secara otomatis ketika arus listrik yang melewati MCB tersebut melebihi nilai yang ditentukan. Namun saat arus dalam kondisi normal, MCB dapat berfungsi sebagai saklar yang bisa menghubungkan atau memutuskan arus listrik secara manual.

MCB pada dasarnya memiliki fungsi yang hampir sama dengan Sekering (fuse) yaitu memutuskan aliran arus listrik rangkaian ketika terjadi gangguan kelebihan arus. Terjadinya kelebihan arus listrik ini dapat dikarenakan adanya hubung singkat (*short circuit*) ataupun adanya beban lebih (*overload*). Namun MCB dapat di-ON-kan kembali ketika rangkaian listrik sudah normal, sedangkan fuse/sekering yang terputus akibat gangguan kelebihan arus tersebut tidak dapat digunakan lagi.

Pada kondisi Normal, MCB berfungsi sebagai saklar manual yang dapat menghubungkan (ON) dan memutuskan (OFF) arus listrik. Pada saat terjadi kelebihan beban (*overload*) ataupun hubung singkat rangkaian (*short circuit*), MCB akan beroperasi secara otomatis dengan memutuskan arus listrik yang melewatinya. Secara visual, kita dapat melihat perpindahan knob atau tombol dari kondisi ON menjadi kondisi OFF. Pengoperasian otomatis ini dilakukan dengan dua cara, yaitu dengan cara *magnetic tripping* (pemutusan hubungan arus listrik secara magnetik) dan *thermal tripping* (pemutusan hubungan arus listrik secara thermal/suhu).

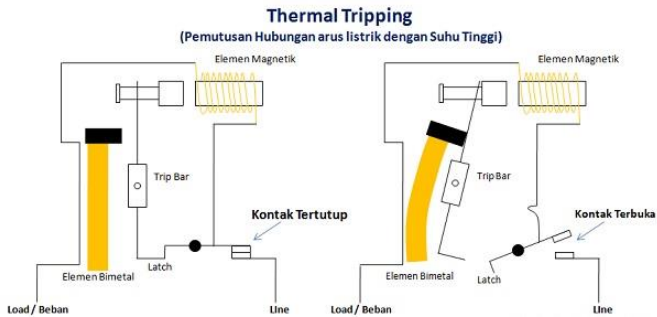


**Gambar 2.9** MCB



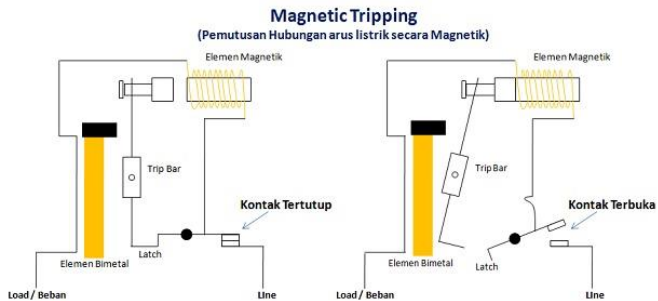
### 2.4.1 Thermal Tripping

Pada saat kondisi overload (kelebihan beban), arus yang mengalir melalui bimetal menyebabkan suhu bimetal itu sendiri menjadi tinggi. Suhu panas tersebut mengakibatkan bimetal melengkung sehingga memutuskan kontak MCB (*Trip*).



### 2.4.2 Magnetic Tripping

Ketika terjadi hubungan singkat rangkaian (*short circuit*) secara mendadak ataupun kelebihan beban yang sangat tinggi (*heavy overload*), *magnetic tripping* atau pemutusan hubungan arus listrik secara magnetik akan diberlakukan. Pada saat terjadi hubungan singkat ataupun kelebihan beban berat, medan magnet pada solenoid MCB akan menarik *latch* (palang) sehingga memutuskan kontak MCB (*trip*).



## 2.5 Fuse

Fuse atau dalam bahasa Indonesia disebut dengan sekering adalah komponen yang berfungsi sebagai pengaman dalam rangkaian elektronika maupun perangkat listrik. Fuse (Sekering) pada dasarnya terdiri dari sebuah kawat halus pendek yang akan meleleh dan terputus jika dialiri oleh arus listrik yang berlebihan ataupun terjadinya hubungan arus pendek (*short circuit*) dalam sebuah peralatan listrik / elektronika. Dengan putusnya fuse (sekering) tersebut, arus listrik yang berlebihan tersebut tidak dapat masuk ke dalam rangkaian elektronika sehingga tidak merusak komponen-komponen yang terdapat dalam rangkaian elektronika yang bersangkutan. Karena fungsinya yang dapat melindungi peralatan listrik dan peralatan elektronika dari kerusakan akibat arus listrik yang berlebihan.

Fuse (sekering) terdiri dari 2 terminal dan biasanya dipasang secara seri dengan rangkaian elektronika / listrik yang akan dilindunginya sehingga apabila fuse (sekering) tersebut terputus maka akan terjadi “*Open Circuit*” yang memutuskan hubungan aliran listrik agar arus listrik tidak dapat mengalir masuk ke dalam Rangkaian yang dilindunginya.



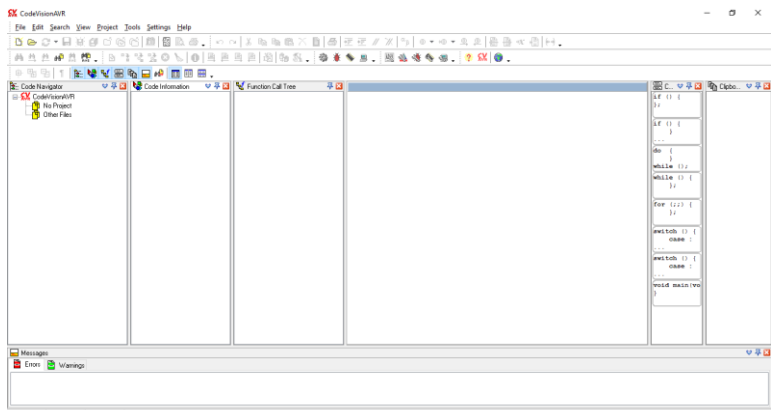
**Gambar 2.12** Fuse

Bentuk fuse (sekering) yang paling sering ditemukan adalah berbentuk tabung (silinder) dan pisau (*blade type*). Fuse yang berbentuk tabung atau silinder sering ditemukan di peralatan listrik rumah tangga maupun pada rangkaian elektronika sedangkan fuse yang berbentuk pisau (*blade*) lebih sering digunakan di bidang otomotif (kendaraan bermotor).

Nilai fuse biasanya tertera pada badan fuse itu sendiri ataupun diukir pada terminal fuse. Nilai fuse diantaranya terdiri dari arus listrik (dalam satuan Ampere (A) ataupun miliAmpere (mA) dan tegangan (dalam satuan Volt (V) ataupun miliVolt (mV)).

## 2.6 CodeVision AVR

CodeVision AVR merupakan sebuah software yang digunakan untuk memprogram mikrokontroler. Mulai dari penggunaan untuk kontrol sederhana sampai kontrol yang cukup kompleks. Mikrokontroler dapat berfungsi jika telah diisi sebuah program. Pengisian program ini dapat dilakukan menggunakan compiler yang selanjutnya diprogram ke dalam mikrokontroler menggunakan fasilitas yang sudah di sediakan oleh program tersebut. Salah satu compiler program yang umum digunakan sekarang ini adalah CodeVision AVR yang menggunakan bahasa pemrograman C. Dengan menggunakan pemrograman bahasa C, diharapkan waktu desain (*deleloping time*) akan menjadi lebih singkat. Setelah program dalam bahasa C ditulis dan dilakukan kompilasi tidak terdapat kesalahan (*error*) maka proses download dapat dilakukan. Mikrokontroller AVR mendukung sistem download secara ISP (*In-System Programming*). Gambar Tampilan CodeVision AVR adalah sebagai berikut :



Gambar 2.13 Tampilan awal CodeVision AVR

## 2.7 Billing

*Billing* berasal dari bahasa Inggris yaitu *bill* (*noun*), yang artinya bukti transaksi pembayaran. Maka *billing* (*adv*) dapat juga diartikan mengirimkan bukti transaksi, atau mengumumkan bukti transaksi. Setiap bidang usaha kini selalu melakukan transaksi, apalagi bidang-bidang yang selalu melakukan transaksi dalam jumlah besar seperti rumah sakit atau departement store atau bidang usaha yang

transaksinya berbentuk abstrak seperti usaha warung internet maupun persewaan billiard, dan disinilah sistem *billing* bekerja.

Sistem *billing* merupakan sistem yang membantu para usahawan untuk mengatur dan mencatat segala transaksi yang terjadi. Contohnya bagi pengusaha persewaan meja billiard, *billing sistem* digunakan untuk memonitor penggunaan meja billiard yang digunakan., menghitung biaya yang harus dibayar *customer*, serta memberikan informasi sebagai analisa pengambilan keputusan secara cepat dan akurat.

Pada dasarnya sistem *billing* merupakan sistem pencatat dan pemonitor transaksi berbentuk software. Untuk pembuatan Sistem *Billing* pada Meja Billiard ini menggunakan software Visual Basic, sehingga untuk menggunakan sistem *billing* cukup dengan meng-*install* software tersebut pada sistem operasi.

## 2.8 Billiard

Billiard atau bola sodok adalah sebuah cabang olahraga yang masuk dalam kategori cabang olahraga konsentrasi, sehingga sangat dibutuhkan ketahanan dan pemahaman mental yang benar serta harus ditunjang oleh kemampuan fisik yang prima agar mampu berprestasi lebih tinggi dan stabil.

Cabang olahraga ini dimainkan di atas meja dan dengan peralatan bantu khusus serta peraturan tersendiri. Permainan ini terbagi dari beberapa Jenis, antara lain jenis *Carom*, *English Billiard* dan *Pool*. Olahraga ini dapat dimainkan secara perorangan maupun tim.

Sebagai contoh, jenis *Carom* dimainkan di meja yang tidak memiliki lubang sama sekali. Ini berbeda dengan jenis *English Billiard* dan *Pool* yang dimainkan di meja yang memiliki lubang sebanyak 6 buah. Meski sama-sama memiliki 6 buah lubang, ukuran atau luas meja antara *English Billiard* dan *Pool*-pun berbeda, lebih luas meja jenis *English Billiard*. Sampai saat ini, yang berkembang di Indonesia adalah jenis *Pool* yang itu pun masih terbagi dalam nomor bola 15, bola 8 dan bola 9.

## 2.9 Komunikasi Serial

Komunikasi serial adalah sebuah komunikasi yang terjadi dengan mengirimkan data per-bit secara berurutan dan bergantian. Komunikasi ini hanya membutuhkan satu jalur atau kabel yang sedikit dibandingkan dengan komunikasi paralel, di mana hanya satu bit data

yang dikirimkan melalui seuntai jalur atau kabel pada waktu tertentu secara berurutan. Manfaat paling dasar adanya komunikasi serial ialah memungkinkan terjadinya pertukaran informasi antara papan ATMega16 dan komputer. Hal ini memungkinkan untuk dapat memantau status pengerjaan yang sedang dilakukan langsung di komputer. Sebagai contoh, dapat mengendalikan mikrokontroler menggunakan komputer. Komunikasi serial pada ATMega16 dapat dilakukan secara dua arah, artinya baik komputer yang kita gunakan maupun papan ATMega16 yang sedang kita program dapat saling berkomunikasi dan saling mengirim maupun menerima informasi.

### 2.9.1 USB to TTL PL2303

USB to TTL merupakan alat untuk berkomunikasi dengan perangkat eksternal menggunakan port USB dengan level tegangan mikrokontroler adalah berbasis 5 V.

Kabel USB TTL menggunakan chip IC PL2303 yang stabil saat dipakai komunikasi dengan kecepatan tinggi. *Chip Prolific PL2303TA* merupakan versi improvement dari PL2303HX. Terdapat 2 LED SMD yang terpasang sehingga bisa menjadi indikator ketika transmisi aktif. Selain itu juga terdapat sirkuit pelindung yang berfungsi untuk melindungi dan menjaga peralatan dari kerusakan yang diakibatkan oleh kelebihan beban. Kelebihan dari konverter *USB to TTL* adalah dapat digunakan sebagai modul komunikasi serial antara mikrokontroler dengan development board yang belum punya fitur USB terintegrasi seperti pada ATMega16 yang digunakan pada pembuatan Sitem *Billing* Pada Meja Billiard ini.



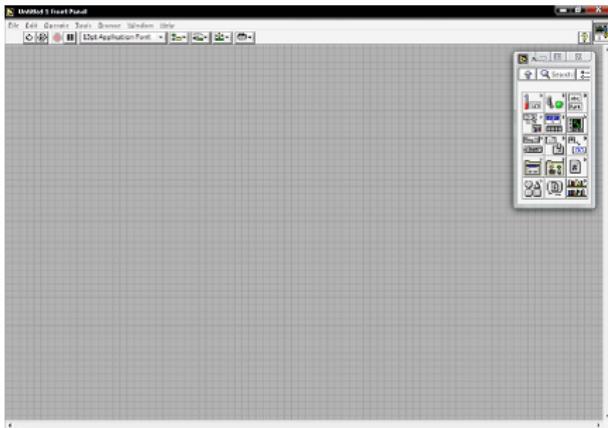
**Gambar 2.14** USB to TTL PL2303

## 2.10 LabVIEW

LabVIEW merupakan singkatan dari *Laboratory Virtual Instrument Engineering Workbench*. Program LabVIEW adalah sebuah software pemrograman yang diproduksi oleh *National Instruments* dengan konsep yang berbeda. Seperti bahasa pemrograman lainnya yaitu C++, matlab atau Visual Basic, LabVIEW juga mempunyai fungsi dan peranan yang sama, perbedaannya bahwa labVIEW menggunakan bahasa pemrograman berbasis grafis atau block diagram sementara bahasa pemrograman lainnya menggunakan basis text. Program labVIEW dikenal dengan sebutan Vi atau *Virtual Instruments* karena penampilan dan operasinya dapat meniru sebuah instrument. Pada labVIEW, user pertama-tama membuat *user interface* atau *front panel* dengan menggunakan kontrol dan indikator, yang dimaksud dengan kontrol adalah *knobs*, *push button*, *dials* dan peralatan input lainnya sedangkan yang dimaksud dengan indikator adalah *graphs*, *LEDs*, dan peralatan display lainnya. Setelah menyusun user interface, lalu user menyusun block diagram yang berisi kode-kode Vi untuk mengontrol front panel. NI (2005) menyebutkan Software LabVIEW terdiri dari tiga komponen utama yaitu:

### 1. *Front panel*

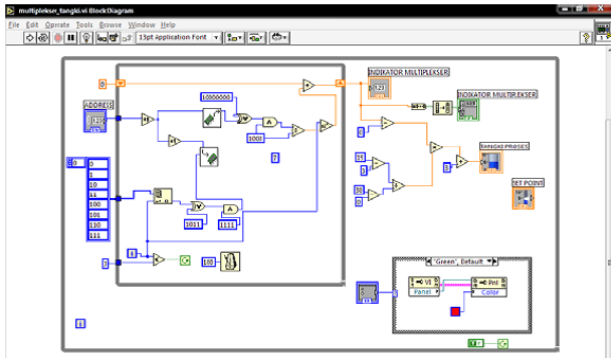
Front panel adalah bagian *window* yang berlatar belakang abu-abu serta mengandung kontrol dan indikator. *Front panel* digunakan untuk membangun sebuah VI, menjalankan program dan mendebug program.



**Gambar 2.15** *Front Panel* pada LabVIEW

2. Blok diagram dari Vi

Blok diagram adalah bagian window yang berlatar belakang putih berisi *source code* yang dibuat dan berfungsi sebagai instruksi untuk *front panel*.



Gambar 2.16 Blok diagram pada LabVIEW

3. Control palette

*Control palette* merupakan tempat beberapa kontrol dan indikator pada *front panel*, *control palette* hanya tersedia di *front panel*, untuk menampilkan *control palette* dapat dilakukan dengan mengklik *windows >> show control palette* atau klik kanan *front panel*.



Gambar 2.17 Control Palette pada LabVIEW

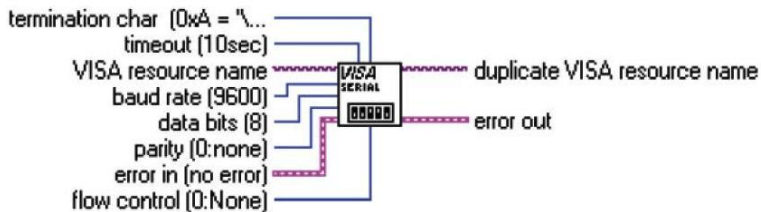
#### 4. *Function palette*

*Function palette* digunakan untuk membangun sebuah block diagram. *Function palette* hanya tersedia pada blok diagram, untuk menampilkannya dapat dilakukan dengan mengklik *windows >> show control palette* atau klik kanan pada lembar kerja block diagram.



**Gambar 2.18** Function *Palette* pada LabVIEW

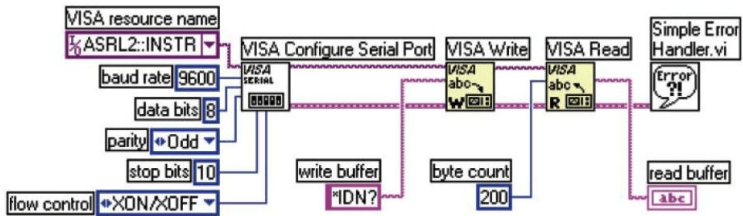
Instrument I/O serial pada LabVIEW pada *subpalette* yang merupakan fungsi pada *Virtual Instrument (VI)* digunakan sebagai komunikasi port serial. Fungsi *VISA Write* dan *VISA Read* akan bekerja dengan ragam tipe komunikasi instrument pada komunikasi serial. Karena komunikasi serial membutuhkan konfigurasi tambahan. Yang pertama dilakukan adalah memulai komunikasi serial dengan konfigurasi VISA port serial. Konfigurasinya dapat dilihat pada gambar 2.20.



**Gambar 2.19** Konfigurasi VISA untuk port serial



Konfigurasi *VISA port serial* menginisialisasi port yang diidentifikasi oleh *VISA resource name setting* yang lebih spesifik. *Timeout set* merupakan nilai *timeout* pada komunikasi serial. *Baud rate*, *data bits*, *parity*, dan *flow control* merupakan parameter spesifik pada port serial. *Clustes error in* dan *error out* memperbaiki kondisi gangguan pada VI.

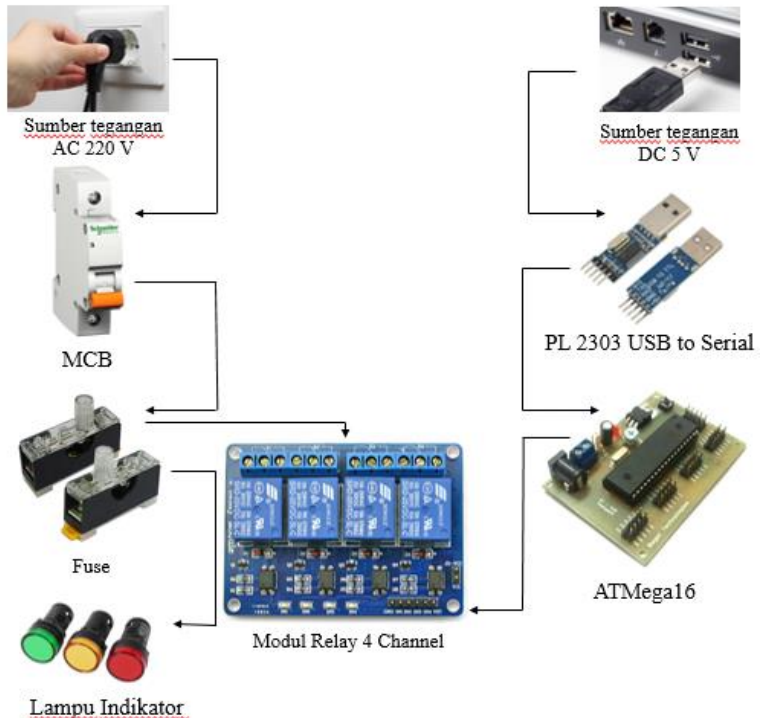


**Gambar 2.20** Konfigurasi pada port serial

--- Halaman ini sengaja dikosongkan ---

### BAB III PERANCANGAN ALAT

Pada bab ini yang akan dibahas adalah tentang perancangan hardware dan software. Dalam perancangan hardware yang dilakukan adalah pembuatan rangkaian relay dan optocoupler, desain akrilik, dan juga *wiring*. Sedangkan dalam perancangan software yang dilakukan adalah melakukan pemrograman tentang pembuatan sistem *billing*, dimana sistem tersebut dapat mengatur nyala atau mati lampu pada meja billiard dan melakukan proses *billing*. Berikut ini adalah Skema proses pada Sistem *Billing* pada Meja Billiard yang ditunjukkan pada gambar 3.1:

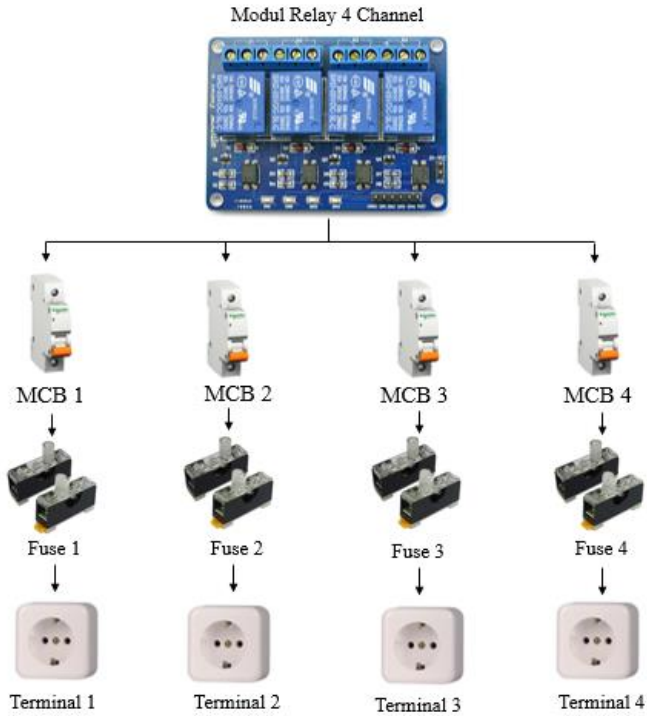


Gambar 3.1 Skema proses pada Sistem *Billing*

Pada gambar 3.1, sumber yang digunakan adalah 220 VAC dan 5 VDC. Sumber 220 VAC akan melewati MCB. MCB disini berfungsi sebagai pengaman ketika terjadi kelebihan arus (*overload*) maupun terjadi hubung singkat (*short circuit*). Selain itu MCB juga berfungsi sebagai saklar utama untuk ON/OFF alat. Selanjutnya masuk ke fuse yang juga berfungsi untuk menjadi pengaman ketika terjadi kelebihan arus maupun hubungan singkat. Perbedaannya dengan fuse adalah MCB dapat di-ON-kan kembali ketika rangkaian listrik sudah normal, sedangkan fuse/sekering yang terputus akibat kelebihan arus tersebut tidak dapat digunakan lagi. Terdapat lampu indikator yang menyatakan bahwa adanya arus yang mengalir ketika MCB dalam keadaan ON, dan akan mati ketika MCB dalam keadaan OFF.

Tegangan 220 VAC digunakan untuk memberi tegangan AC pada modul relay 4 channel. Sedangkan sumber 5 VDC digunakan untuk memberi tegangan sumber pada ATMega16 melalui USB to TTL PL2303. Selain itu tegangan 5 VDC digunakan untuk memberi tegangan DC (logika 0 dan 1) pada rangkaian relay. Relay pada awalnya berada pada kondisi NO (*normally open*), namun ketika diberi logika 1 oleh ATMega16, maka relay akan aktif dengan kondisi NC (*normally close*). Dalam rangkaian modul relay 4 channel, terdapat empat buah relay, dimana masing-masing relay mewakili satu terminal yang nantinya akan disambungkan dengan lampu pada meja billiard. Seperti contoh ketika relay A diberi logika 1 oleh ATMega16, maka terminal A akan aktif dan menyalakan lampu meja billiard A. Pada saat lampu meja billiard menyala, maka akan terjadi proses *billing*, dan akan mati ketika waktu yang telah ditentukan telah habis. Begitu juga dengan relay B, C, dan D.

Pada gambar 3.2 adalah skema proses pada Sistem *Billing* pada Meja Billiard lanjutan dari gambar 3.1. Relay berfungsi untuk mengendalikan sirkuit tegangan tinggi (lampu AC) dengan bantuan dari signal tegangan rendah dari mikrokontroler. Relay akan aktif ketika diberikan logika 1 oleh mikrokontroler. Namun untuk menyalakan lampu AC pada meja Billiard, diberikan MCB dan fuse pada masing-masing meja yang digunakan sebagai pengaman adanya arus berlebih (*overload*) maupun adanya hubung singkat rangkaian (*short circuit*).



**Gambar 3.2** Skema proses pada Sistem *Billing*

### 3.1 Perancangan Hardware

Perancangan hardware merupakan kegiatan menyatukan semua komponen berdasarkan rangkaian yang telah disusun sesuai sistem rancang bangun. Komponen yang digunakan antara lain adalah:

1. Terminal (5 buah)
2. Fuse kubur (4 buah)
3. MCB (5 buah)
4. Modul Relay (1 buah)
5. Kabel power
6. Indikator lampu AC
7. Kabel Pelangi
8. Mikrokontroler ATmega16
9. Komunikasi serial USB to TTL PL2303 (1 buah)

## 10. Lampu 10.5 W (4 buah)

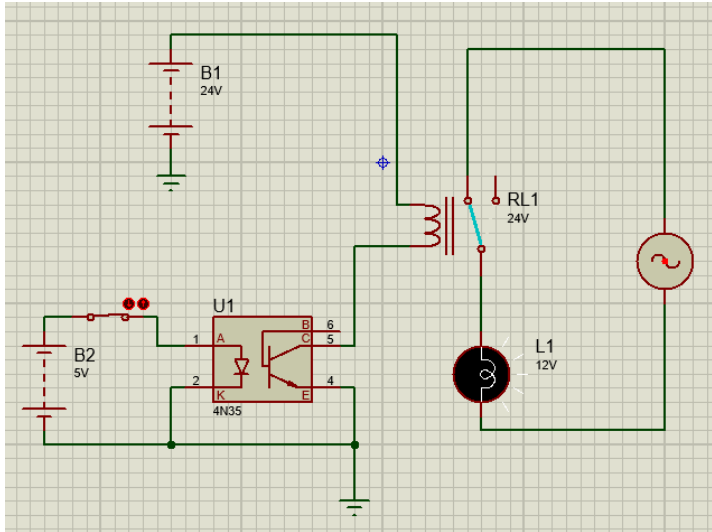
### 3.1.1 Perancangan Rangkaian Relay dan Optocoupler

Optocoupler adalah sebuah komponen semi konduktor atau alat yang terdiri dari LED (*Light Emitting Diode*) dan komponen yang sensitif terhadap cahaya. Biasanya digunakan untuk isolasi rangkaian satu ke rangkaian yang lainnya. Optocoupler juga sering di kenal dengan nama *optical coupler* dan opto isolator. Disebut sebagai opto-isolator karena LED dengan komponen sensitif cahaya terpisah oleh udara, namun dua komponen di *package* dalam satu komponen.

Optocoupler ini di hubungkan ke sistem kontroler atau rangkaian elektronik dengan tegangan rendah dan sensitif terhadap beban tegangan besar. Jadi untuk memproteksi adanya gangguan tegangan *feedback* maka di gunakanlah isolator ini. Komponen sensitif cahaya ini (biasanya photo transistor) dihubungkan untuk mengendalikan lampu AC. Jadi dengan tegangan yang kecil mampu mengendalikan beban lampu AC yang membutuhkan daya yang besar.

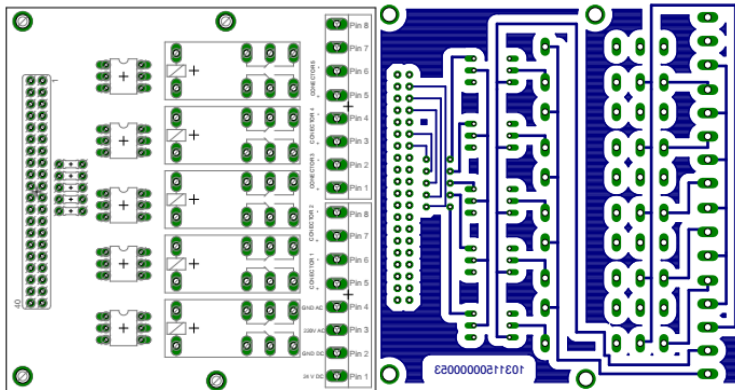
Rangkaian driver relay menggunakan optocoupler lebih bagus kinerjanya dibandingkan menggunakan transistor. Hal ini dikarenakan LED pada optocoupler yang terhubung ke kontroller hanya di hubungkan oleh cahaya saja, maka ketika ada kerusakan pada rangkaian photo transistor (komponen sensitif cahaya) maka tidak akan merusak fungsi rangkaian kontroller.

LED pada optocoupler (U1) dihubungkan ke pin baterai sebagai sumber 5 VDC (B2). Kemudian pada sisi Photo transistornya (*receiver*) kita hubungkan ke relay (RL1) untuk dikendalikan. Ketika LED pada optocoupler mendapatkan tegangan dari baterai, maka lampu (L1) akan menyala. Memberi tegangan 5 VDC dari baterai (B2) akan menyebabkan LED pada optocoupler (U1) aktif sehingga mengaktifkan photo transistor mengakibatkan lilitan pada rangkaian dalam relay (RL1) bersifat magnet dan menarik mekanik kontaktor di dalam relay yang artinya relay menjadi terhubung/aktif. Selain itu sumber tegangan relay juga didapatkan dari baterai 24 VDC (B1) dan tegangan AC 220 V. Rangkaian *schematic* relay dan optocoupler yang di simulasikan pada program “Proteus” ditunjukkan pada gambar 3.3.



**Gambar 3.3** Schematic rangkaian relay dan optocoupler

Setelah diuji pada program proteus dan programnya berjalan dengan baik, maka dilakukan *routing* rangkaian relay dan optocoupler menggunakan program eagle dan didapatkan hasil layout PCB yang ditunjukkan pada gambar 3.4 dibawah ini:



**Gambar 3.4** Layout PCB rangkaian optocoupler dan relay

Dari layout PBC yang telah dibuat, didapatkan rangkaian relay dan optocoupler yang ditunjukkan oleh gambar 3.4. Komponen yang digunakan adalah relay hrm1h-s-dc24v (5 buah), optocoupler 4N35 (5 buah), resistor 330  $\Omega$  (5 buah), *socket connector* 20 pin (1 buah), dan terminal block 8 kaki (2 buah).



**Gambar 3.5** Rangkaian relay dan optocoupler

### 3.1.2 Desain Alat

Untuk merangkai semua komponen yang telah disiapkan, diperlukan desain untuk peletakan komponen-komponen. Bahan yang digunakan adalah akrilik. Akrilik merupakan plastik yang bentuknya menyerupai kaca. Namun, akrilik mempunyai sifat-sifat yang membuatnya lebih unggul dibandingkan dengan kaca. Salah satu perbedaannya adalah kelenturan yang dimiliki oleh akrilik. Akrilik merupakan bahan yang tidak mudah pecah, ringan, dan juga mudah untuk dipotong, dikikir, dibor, dihaluskan, dikilapkan atau dicat. Akrilik dapat dibentuk secara thermal menjadi berbagai macam bentuk yang rumit. Karena sifatnya mudah dipotong, dibor, dan tidak mudah pecah, maka akrilik adalah material yang ideal untuk dipergunakan sebagai tempat peletakan komponen-komponen yang dibutuhkan.

Untuk memotong akrilik sesuai dengan desain yang kita harapkan, cara yang paling efisien adalah menggunakan laser *cutting*. Laser *cutting* adalah teknik pemotongan bahan dengan permukaan rata menggunakan sinar laser. Pemotongan dengan menggunakan teknik laser *cutting* ini desainnya dapat disesuaikan

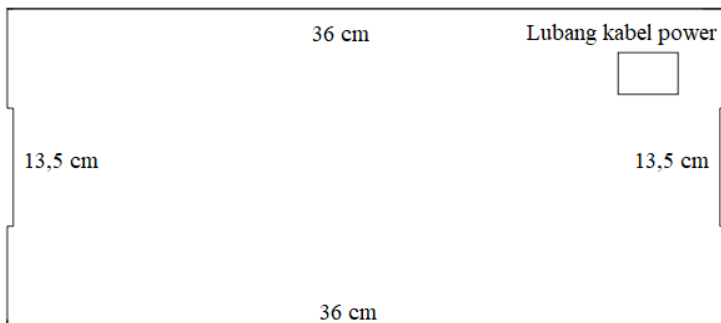


dengan kebutuhan (desain khusus) sampai pada detail yang sangat kecil dan hasil potong yang sangat presisi.



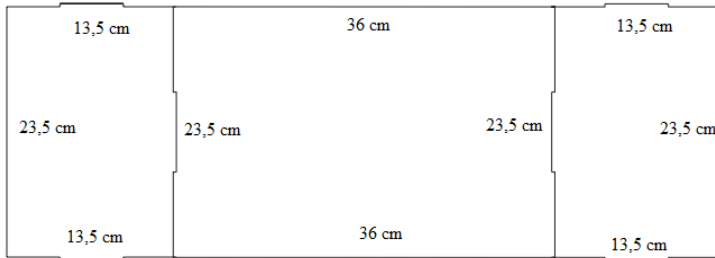
**Gambar 3.6** Akrilik

Sebelum melakukan pemotongan akrilik dengan teknik laser *cutting*, diperlukan desain terlebih dahulu. Desain ini dibuat pada software “Corel Draw”. Corel draw adalah sebuah program komputer yang melakukan editing pada garis vektor. Berikut ini adalah hasil desain di software corel draw :



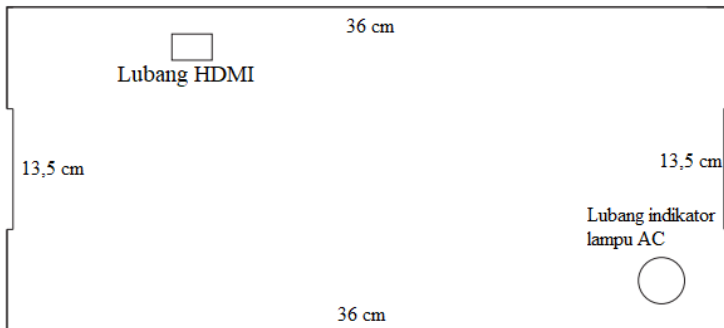
**Gambar 3.7** Desain bagian belakang

Pada gambar 3.7, desain yang dibuat adalah ukuran 36cm x 13,5 cm. Pada bagian belakang terdapat lubang untuk kabel power (220 VAC) dan juga nantinya akan di bor untuk peletakan 4 buah terminal.



**Gambar 3.8** Desain bagian depan

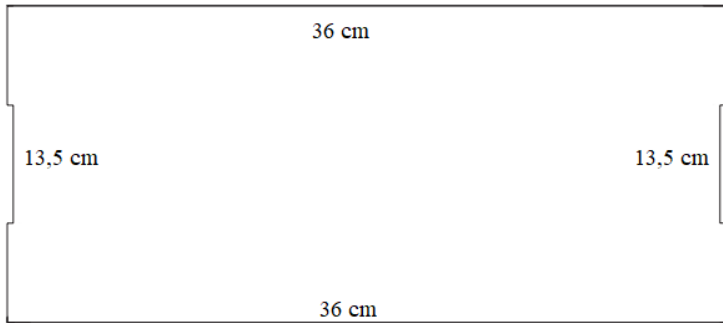
Pada gambar 3.8, desain yang dibuat adalah ukuran 36cm x 13,5 cm. Pada desain bagian depan ini terdapat dua lubang, yaitu lubang untuk indikator lampu AC dan lubang untuk masuknya kabel mikrokontroler.



**Gambar 3.9** Desain bagian samping dan atas

Pada gambar 3.9, merupakan desain bagian samping dan bagian atas. Pada bagian samping ukuran yang dibuat adalah 23,5cm x 13,5cm, sedangkan pada bagian atas ukurannya adalah 36cm x 23,5cm.

Pada gambar 3,10, merupakan desain bagian bawah dengan ukuran 36cm x 13,5cm. Namun ada perbedaan mengenai penggunaan ukuran akrilik yang digunakan. Pada bagian bawah ini yang digunakan adalah akrilik dengan ketebalan 5mm, sedangkan untuk ukuran atas, samping, depan maupun belakang menggunakan akrilik dengan ketebalan 3mm.



**Gambar 3.10** Desain bagian bawah

Dari desain yang telah dibuat, didapatkan hasil alat yang ditunjukkan oleh gambar 3.11, 3.12, dan 3.13 di bawah ini:



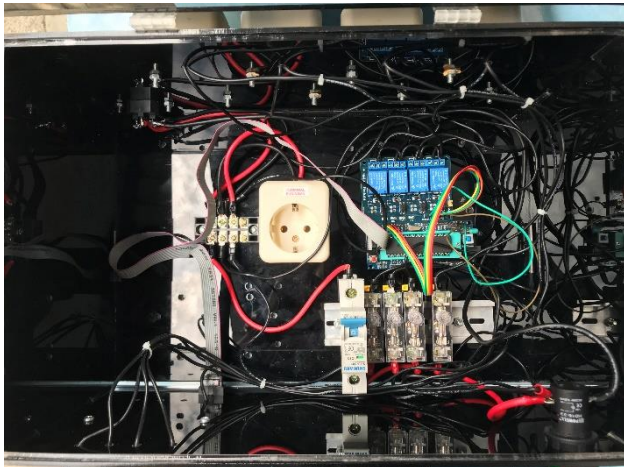
**Gambar 3.11** Bagian depan alat

Pada gambar 3.11, merupakan bagian depan alat. Pada bagian depan terdapat lampu indikator AC yang berfungsi untuk mengetahui bahwa alat sudah mendapatkan power 220 VAC dan juga terdapat 4 buah MCB yang digunakan sebagai pengaman pada setiap lampu meja billiard yang dikontrol.



**Gambar 3.12** Bagian belakang alat

Pada gambar 3.12 merupakan bagian belakang alat. Pada bagian belakang terdapat empat buah terminal yang digunakan sebagai stop kontak pada lampu meja billiard.



**Gambar 3.13** Bagian dalam alat

Pada gambar 3.13 merupakan bagian dalam alat. Pada bagian dalam merupakan tempat peletakan komponen yang digunakan. Terdapat beberapa komponen yang digunakan yaitu, 1 buah terminal, 4

buah fuse, 1 buah MCB, modul relay 4 *channel* dengan optocoupler, dan mikrokontroler ATmega16.

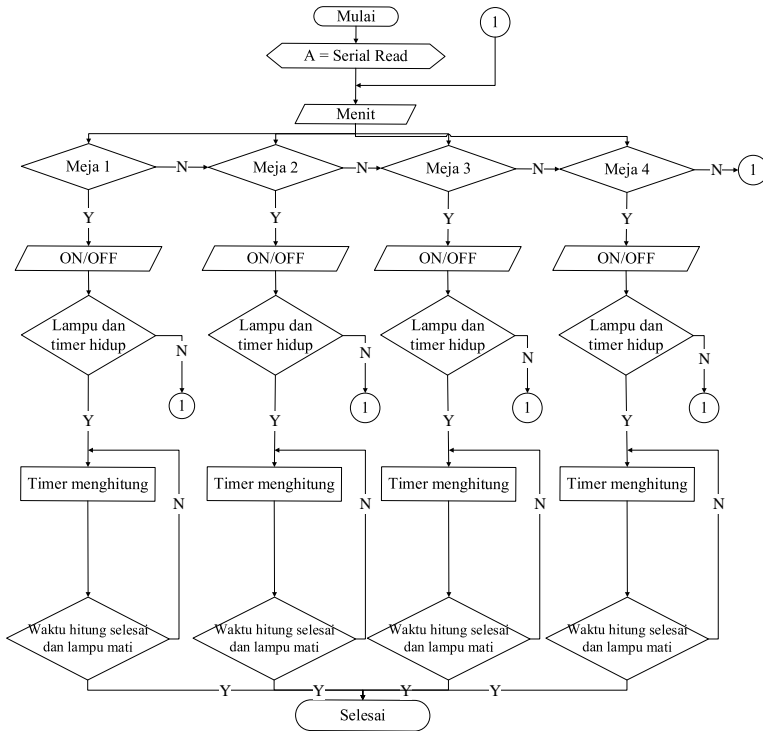
### **3.2 Perancangan Software**

Pada perancangan software untuk alat ini ada pada bagian mengaktifkan beban lampu AC melalui mikrokontroler ATmega16 yang di rangkai dengan rangkaian relay dan optocoupler. Program tersebut diharapkan dapat melakukan kontrol nyala dan matinya lampu AC, dan juga dapat melakukan proses *billing* ketika lampu mulai dinyalakan. Sehingga nantinya dapat diketahui jumlah yang harus dibayar *customer* dalam persewaan meja billiard. Pembuatan program pada mikrokontroler ATmega16 menggunakan program CodeVisionAVR. Selain itu diperlukan HMI (*human machine interface*) untuk tampilan penghubung antara manusia dengan mesin. Untuk membuat HMI ini menggunakan program LabVIEW.

#### **3.2.1 Flowchart Program**

Program ini dimulai dengan pemberian harga awal variabel menit bernilai 0. Selanjutnya input berupa menit dimasukkan. Apakah input menit diberikan pada meja 1, jika ya selanjutnya diberikan input pada tombol ON/OFF. Jika tidak apakah input menit diberikan pada meja 2, jika ya selanjutnya diberikan input pada tombol ON/OFF. Jika tidak apakah input menit diberikan pada meja 3, jika ya selanjutnya diberikan input pada tombol ON/OFF. Jika tidak apakah input menit diberikan pada meja 4, jika ya selanjutnya diberikan input pada tombol ON/OFF. Jika tidak maka akan kembali pada input berupa menit.

Ketika diberi input ON, apakah lampu dan timer hidup. Jika ya maka timer akan menghitung. Jika tidak kembali ke input menit. Lampu dan timer hidup berdasarkan input menit dan tombol ON sampai pada batas menit yang diberikan pada menit. Apakah waktu hitung dan lampu mati, jika ya maka program selesai, jika tidak maka timer akan tetap berjalan. Begitu juga dengan meja 2,3, dan 4.



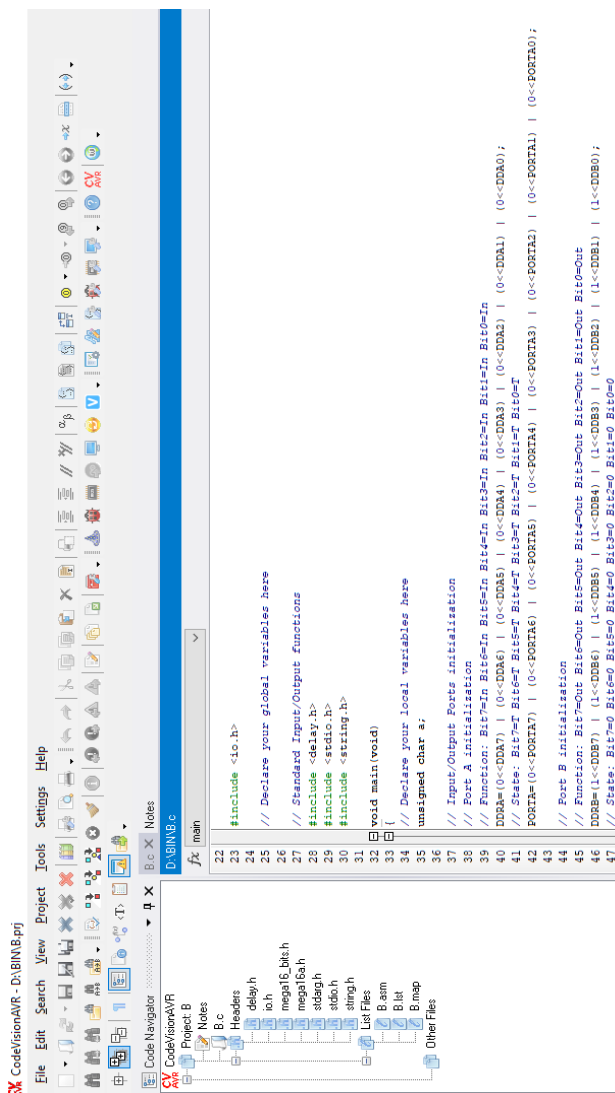
**Gambar 3.14** Flowchart program

### 3.2.2 Perancangan program pada CodeVisionAVR

CodeVisionAVR adalah sebuah compiler C yang telah dilengkapi dengan fasilitas *Integrated Development Environment* (IDE) dan didesain agar dapat menghasilkan kode program secara otomatis untuk mikrokontroler Atmel AVR. *Integrated Development Environment* (IDE) telah dilengkapi dengan fasilitas pemrograman chip melalui metode *In-System Programming* sehingga dapat secara otomatis mentransfer file program ke dalam chip mikrokontroler AVR setelah sukses dikompilasi.

CodeVisionAVR dapat menghasilkan kode program secara otomatis melalui fasilitas *CodeWizardAVR Automatic Program Generator*. Dengan adanya fasilitas ini maka penulisan program

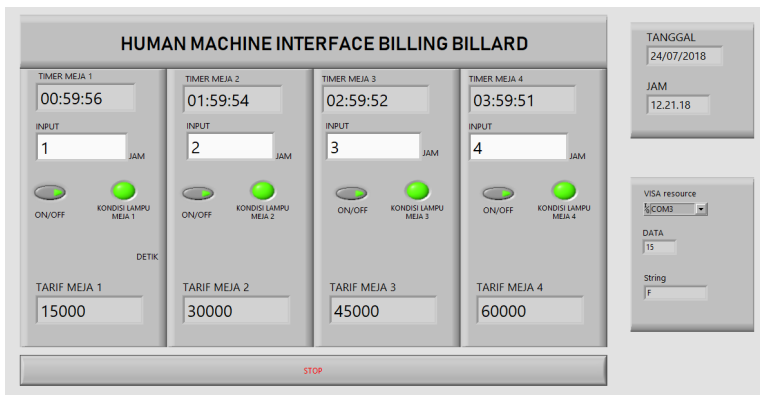
dapat dilakukan dengan cepat dan lebih efisien. Gambar 3.15 ini adalah pemrograman ATmega16 menggunakan CodeVisionAVR.



Gambar 3.15 Pemrograman menggunakan CodeVisionAVR

### 3.2.2 Perancangan HMI menggunakan LabVIEW

Perancangan HMI (*Human Machine Interface*) menggunakan LabVIEW adalah untuk mempermudah pengontrolan alat. Pada HMI di desain sedemikian rupa agar mudah dalam penggunaannya. Desain HMI yang dibuat berisi lama waktu yang akan digunakan setiap meja, kondisi lampu setiap meja, tombol ON/OFF untuk mengaktifkan proses *billing* dan menyelakan lampu pada meja billiard, biaya/tarif yang akan muncul, dan tombol “STOP” untuk mengakhiri semua proses yang berjalan. Pada gambar 3.16 adalah hasil HMI yang telah dibuat pada program LabVIEW.



**Gambar 3.16** HMI menggunakan LabView

Pada gambar 3.16 terdapat beberapa indikator yang digunakan. Diantaranya adalah :

**Tabel 3.1** Fungsi-fungsi indikator pada HMI

No	Indikator	Fungsi
1	Timer	Untuk mengetahui waktu yang berjalan
2	Set Point (menit)	Untuk menentukan waktu yang digunakan
3	Tombol ON/OFF	Untuk memulai dan mengakhiri proses <i>billing</i>
4	Kondisi lampu	Untuk mengetahui kondisi lampu dalam keadaan menyala atau mati
5	Stop	Untuk menghentikan semua proses <i>billing</i>
6	Port	Untuk memilih <i>port</i> yang akan digunakan



7	Data	Untuk mengetahui data yang dikirimkan ke mikrokontroler
8	Tarif Meja	Untuk mengetahui biaya yang perlu dibayar oleh <i>customer</i>
9	Tanggal	Tanggal <i>real time</i>
10	Jam	Jam <i>real time</i>

--- Halaman ini sengaja dikosongkan ---

## BAB IV UJI UKUR DAN UJI ALAT

Pengujian alat yang dilakukan merupakan pengujian terhadap perangkat keras (*hardware*) dan perangkat lunak (*software*) dari sistem secara keseluruhan yang telah selesai dibuat untuk mengetahui komponen-komponen apakah berjalan dengan baik dan sesuai yang diharapkan.

### 4.1 Pengujian Relay

Pada pengujian relay ini yang diuji adalah kondisi relay apakah aktif atau tidak. Aktif atau tidaknya relay dipengaruhi oleh logika mikrokontroler dan kondisi MCB. Dari hasil uji relay, nantinya akan diketahui kondisi lampu apakah nyala atau tidak. Berikut ini adalah data dari pengujian relay yang digunakan dijelaskan pada tabel 4.1, 4.2, 4.3, dan 4.4.

**Tabel 4.1** Pengujian relay 1

No	Logika Mikrokontroler	Tegangan output Mikrokontroler (VDC)	MCB	Relay	Kondisi Lampu
1.	0	0	OFF	Tidak Aktif	Mati
2.	0	0	ON	Tidak Aktif	Mati
3.	1	5	OFF	Tidak Aktif	Mati
4.	1	4,9	ON	Aktif	Nyala

Dari tabel 4.1 diatas, didapatkan data yaitu :

1. Ketika logika mikrokontroler bernilai 0, maka output dari mikrokontroler adalah 0 V. Begitu juga dengan keadaan MCB mati, maka relay tidak akan aktif karena tidak mendapat sumber tegangan dari mikrokontroler (VDC) maupun tegangan 220 VAC, sehingga lampu tidak akan menyala
2. Ketika logika mikrokontroler bernilai 0, maka output dari mikrokontroler adalah 0 V. Meskipun MCB dalam keadaan

nyala, relay tetap tidak akan aktif karena tidak mendapat sumber tegangan dari mikrokontroler (VDC). Sehingga lampu tidak akan menyala

3. Ketika logika mikrokontroler bernilai 1, output mikrokontroler adalah 4,7 VDC. Namun MCB dalam keadaan mati, maka relay tidak akan aktif karena tidak mendapat sumber tegangan 220 VAC, sehingga lampu tidak akan menyala
4. Ketika logika mikrokontroler bernilai 1, output dari mikrokontroler adalah 4,6 VDC. Begitu juga dengan MCB dalam keadaan menyala, maka relay akan aktif karena mendapat sumber dari mikrokontroler (VDC) maupun tegangan 220 VAC, sehingga lampu akan menyala.

**Tabel 4.2** Pengujian relay 2

No	Logika mikrokontroler	Tegangan output mikrokontroler (VDC)	MCB	Relay	Kondisi Lampu
1.	0	0	OFF	Tidak Aktif	Mati
2.	0	0	OFF	Tidak Aktif	Mati
3.	1	5,2	OFF	Tidak Aktif	Mati
4.	1	5	ON	Aktif	Nyala

Dari tabel 4.2 diatas, didapatkan data yaitu :

1. Ketika logika mikrokontroler bernilai 0, maka output dari mikrokontroler adalah 0 V. Begitu juga dengan keadaan MCB mati, maka relay tidak akan aktif karena tidak mendapat sumber tegangan dari raspberry (VDC) maupun tegangan 220 VAC. Sehingga lampu tidak akan menyala
2. Ketika logika mikrokontroler bernilai 0, maka output dari mikrokontroler adalah 0 V. Meskipun MCB dalam keadaan nyala, relay tetap tidak akan aktif karena tidak mendapat sumber tegangan dari raspberry (VDC). Sehingga lampu tidak akan menyala

3. Ketika logika mikrokontroler bernilai 1, output mikrokontroler adalah 4,1 VDC. Namun MCB dalam keadaan mati, maka relay tidak akan aktif karena tidak mendapat sumber tegangan 220 VAC. Sehingga lampu tidak akan menyala
4. Ketika logika mikrokontroler bernilai 1, output dari mikrokontroler adalah 4,1 VDC. Begitu juga dengan MCB dalam keadaan menyala, maka relay akan aktif karena mendapat sumber dari mikrokontroler (VDC) dan juga tegangan 220 VAC. Sehingga lampu akan menyala

**Tabel 4.3** Pengujian relay 3

No	Logika mikrokontroler	Tegangan output mikrokontroler (VDC)	MCB	Relay	Kondisi Lampu
1.	0	0	OFF	Tidak Aktif	Mati
2.	0	0	ON	Tidak Aktif	Mati
3.	1	4,9	OFF	Tidak Aktif	Mati
4.	1	5,1	ON	Aktif	Nyala

Dari tabel 4.3 diatas, didapatkan data yaitu :

1. Ketika logika mikrokontroler bernilai 0, maka output dari mikrokontroler adalah 0 V. Begitu juga dengan keadaan MCB mati, maka relay tidak akan aktif karena tidak mendapat sumber tegangan dari mikrokontroler (VDC) maupun tegangan 220 VAC. Sehingga lampu tidak akan menyala
2. Ketika logika mikrokontroler bernilai 0, maka output dari mikrokontroler adalah 0 V. Meskipun MCB dalam keadaan nyala, relay tetap tidak akan aktif karena tidak mendapat sumber dari mikrokontroler (VDC). Sehingga lampu tidak akan menyala
3. Ketika logika mikrokontroler bernilai 1, output mikrokontroler adalah 3,9 VDC. Namun MCB dalam keadaan mati, maka relay tidak akan aktif karena tidak mendapat sumber tegangan 220 VAC. Sehingga lampu tidak akan menyala

4. Ketika logika mikrokontroler bernilai 1, output dari mikrokontroler adalah 4 VDC. Begitu juga dengan MCB dalam keadaan menyala, maka relay akan aktif karena mendapat sumber dari mikrokontroler (VDC) maupun tegangan 220 VAC. Sehingga lampu akan menyala

**Tabel 4.4** Pengujian relay 4

No	Logika mikrokontroler	Tegangan output mikrokontroler (VDC)	MCB	Relay	Kondisi Lampu
1.	0	0	OFF	Tidak Aktif	Mati
2.	0	0	ON	Tidak Aktif	Mati
3.	1	4,9	OFF	Tidak Aktif	Mati
4.	1	5,1	ON	Aktif	Nyala

Dari tabel 4.4 diatas, didapatkan data yaitu :

1. Ketika logika mikrokontroler bernilai 0, maka output dari mikrokontroler adalah 0 V. Begitu juga dengan keadaan MCB mati, maka relay tidak akan aktif karena tidak mendapat sumber dari mikrokontroler (VDC) maupun tegangan 220 VAC. Sehingga lampu tidak akan menyala
2. Ketika logika mikrokontroler bernilai 0, maka output dari mikrokontroler adalah 0 V. Meskipun MCB dalam keadaan nyala, relay tetap tidak akan aktif karena tidak mendapat sumber dari mikrokontroler (VDC). Sehingga lampu tidak akan menyala
3. Ketika logika mikrokontroler bernilai 1, output raspberry adalah 4,1 VDC. Namun MCB dalam keadaan mati, maka relay tidak akan aktif karena tidak mendapat sumber tegangan 220 VAC. Sehingga lampu tidak akan menyala
4. Ketika logika mikrokontroler bernilai 1, output dari mikrokontroler adalah 4,3 V. Begitu juga dengan MCB dalam keadaan menyala, maka relay akan aktif karena mendapat

sumber dari mikrokontroler (VDC) maupun tegangan 220 VAC. Sehingga lampu akan menyala

#### 4.2 Pengujian Alat dan Program

Untuk mengetahui program HMI yang telah dibuat berjalan dengan baik, diperlukan adanya pengujian alat dan program. Pengujian ini membutuhkan *stopwatch* untuk membandingkan kondisi timer pada program dengan waktu pada *stopwatch*. Berikut ini adalah hasil pengujian alat dan program yang ditunjukkan oleh tabel 4.5.

**Tabel 4.5** Pengujian alat dan program

Timer pada HMI (menit)	<i>Stopwatch</i> (menit)	Waktu lampu padam	Biaya yang keluar
1	1	Sesuai	250
2	2	Sesuai	500
3	3	Sesuai	750
4	4	Sesuai	1000
5	5	Sesuai	1250
6	6	Sesuai	1500
7	7	Sesuai	1750
8	8	Sesuai	2000
9	9	Sesuai	2250
10	10	Sesuai	2500
11	11	Sesuai	2750
12	12	Sesuai	3000
13	13	Sesuai	3250
14	14	Sesuai	3500
15	15	Sesuai	3750
30	30	Sesuai	7500
60	60	Sesuai	15000
90	92	<i>Delay 2 detik</i>	22500
120	122	<i>Delay 2 detik</i>	30000
150	152	<i>Delay 2 detik</i>	37500
180	182	<i>Delay 2 detik</i>	45000
210	212	<i>Delay 2 detik</i>	50000
240	242	<i>Delay 2 detik</i>	60000
270	272	<i>Delay 2 detik</i>	67500
300	301	<i>Delay 2 detik</i>	75000

Timer pada HMI (menit)	<i>Stopwatch</i> (menit)	Waktu lampu padam	Biaya yang keluar
330	332	<i>Delay</i> 2 detik	82500
360	362	<i>Delay</i> 2 detik	90000

Berdasarkan tabel 4.5, pengujian dilakukan selama 6 jam. Didapatkan data bahwa pengujian selama 1 menit hingga 60 menit, kondisi lampu padam sesuai dengan input pada HMI. Namun pada rentan waktu pengujian selama 90 menit sampai 360 menit terjadi *delay* selama 2 detik. *Delay* ini terlihat ketika indikator lampu pada HMI telah mati, selang 2 detik lampu pada *prototipe* meja billiard baru padam.



## **BAB V**

### **PENUTUP**

Bab penutup berisi tentang kesimpulan-kesimpulan yang didapatkan selama proses pembuatan Tugas Akhir ini beserta saran-saran untuk perbaikan dan pengembangannya.

#### **5.1 Kesimpulan**

Dari pengujian dan analisa perancangan dan implementasi Sistem *Billing* pada Meja Billiard, diperoleh beberapa kesimpulan, antara lain:

1. Perancangan dan implementasi sistem billing billard ini menggunakan mikrokontroler ATmega16 dengan bahasa C.
2. HMI *billing* billiard mampu mengendalikan lampu pada meja billiard.
3. Software *billing billiard* dapat menghitung dengan benar biaya penggunaan/penyewaan meja billiard.
4. Pengujian dalam rentan waktu 1-60 menit, kondisi lampu padam sesuai dengan input yang diberikan pada HMI (*human machine interface*), sedangkan pengujian dalam rentan waktu 90-360 menit terdapat *delay* selama 2 detik. *Delay* ini terlihat ketika indikator lampu pada HMI telah mati, selang 2 detik lampu pada *protitipe* meja billiard baru padam.

#### **5.2 Saran**

Berikut ini adalah saran-saran yang dapat diberikan untuk implementasi dan pengembangan lebih lanjut dari sistem ini :

1. Untuk pengembangan lebih lanjut peralatan ini dapat dikembangkan menjadi peralatan dengan sistem kendali jarak jauh dan juga dapat dikendalikan melalui android.
2. Perlu ditambahkan *database*, sehingga dapat menyimpan file transaksi yang telah dilakukan
3. Perlu ditambahkan program cetak struk, sehingga transaksi yang dilakukan memiliki nota
4. Perlu ditambahkan program pemesanan makanan dan minuman dengan biaya total masuk pada sistem *billing*

--- Halaman ini sengaja dikosongkan ---

## DAFTAR PUSTAKA

Budiharto, Widodo. 2010. *Elektronika Digital + Mikroprosesor*. Jakarta: Andi

<https://teknikelektronika.com/pengertian-relay-fungsi-relay/> (diakses pada Maret 2018)

<https://teknikelektronika.com/pengertian-optocoupler-fungsi-prinsip-kerja-optocoupler/> (diakses pada Maret 2018)

<https://teknikelektronika.com/pengertian-mcb-miniature-circuit-breaker-prinsip-kerja-mcb/> (diakses pada Maret 2018)

<https://teknikelektronika.com/mengukur-pengertian-fungsi-fuse-sekring/> (diakses pada Maret 2018)

--- Halaman ini sengaja dikosongkan ---

## Lampiran A – Listing program pada CodeVisionAVR

```
/******  
Chip type      : ATmega16A  
Program type   : Application  
AVR Core Clock frequency: 8,000000 MHz  
Memory model   : Small  
External RAM size : 0  
Data Stack size : 256  
*****/  

```

```
#include <io.h>
```

```
// Declare your global variables here
```

```
// Standard Input/Output functions
```

```
#include <delay.h>
```

```
#include <stdio.h>
```

```
#include <string.h>
```

```
void main(void)
```

```
{
```

```
// Declare your local variables here
```

```
unsigned char a;
```

```
// Input/Output Ports initialization
```

```
// Port A initialization
```

```
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In  
Bit0=In
```

```
DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) |  
(0<<DDA3) | (0<<DDA2) | (0<<DDA1) | (0<<DDA0);
```

```
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T  
PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) |  
(0<<PORTA4) | (0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) |  
(0<<PORTA0);
```

```
// Port B initialization
```

```
// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out  
Bit2=Out Bit1=Out Bit0=Out
```

```

DDRB=(1<<DDB7) | (1<<DDB6) | (1<<DDB5) | (1<<DDB4) |
(1<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0);
// State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) |
(0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) |
(0<<PORTB0);

// Port C initialization
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In
Bit0=In
DDRC=(1<<DDC7) | (1<<DDC6) | (1<<DDC5) | (1<<DDC4) |
(1<<DDC3) | (1<<DDC2) | (1<<DDC1) | (1<<DDC0);
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) |
(0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) |
(0<<PORTC0);

// Port D initialization
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In
Bit0=In
DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) |
(0<<DDD3) | (0<<DDD2) | (0<<DDD1) | (0<<DDD0);
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) |
(0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) |
(0<<PORTD0);

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=0xFF
// OC0 output: Disconnected
TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01)
| (0<<CS02) | (0<<CS01) | (0<<CS00);
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock

```

```

// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Disconnected
// OC1B output: Disconnected
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) |
(0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12)
| (0<<CS12) | (0<<CS11) | (0<<CS10);
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0<<AS2;
TCCR2=(0<<PWM2) | (0<<COM21) | (0<<COM20) | (0<<CTC2) |
(0<<CS22) | (0<<CS21) | (0<<CS20);
TCNT2=0x00;
OCR2=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) |
(0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);

// External Interrupt(s) initialization

```

```

// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
MCUCSR=(0<<ISC2);

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=(0<<RXC) | (0<<TXC) | (0<<UDRE) | (0<<FE) | (0<<DOR)
| (0<<UPE) | (0<<U2X) | (0<<MPCM);
UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (1<<RXEN) |
(1<<TXEN) | (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);
UCSRC=(1<<URSEL) | (0<<UMSEL) | (0<<UPM1) | (0<<UPM0) |
(0<<USBS) | (1<<UCSZ1) | (1<<UCSZ0) | (0<<UCPOL);
UBRRH=0x00;
UBRRL=0x33;

// Analog Comparator initialization
// Analog Comparator: Off
// The Analog Comparator's positive input is
// connected to the AIN0 pin
// The Analog Comparator's negative input is
// connected to the AIN1 pin
ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE)
| (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);
SFIOR=(0<<ACME);

// ADC initialization
// ADC disabled
ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) |
(0<<ADIE) | (0<<ADPS2) | (0<<ADPS1) | (0<<ADPS0);

// SPI initialization
// SPI disabled

```



```

SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) |
(0<<CPOL) | (0<<CPHA) | (0<<SPR1) | (0<<SPR0);

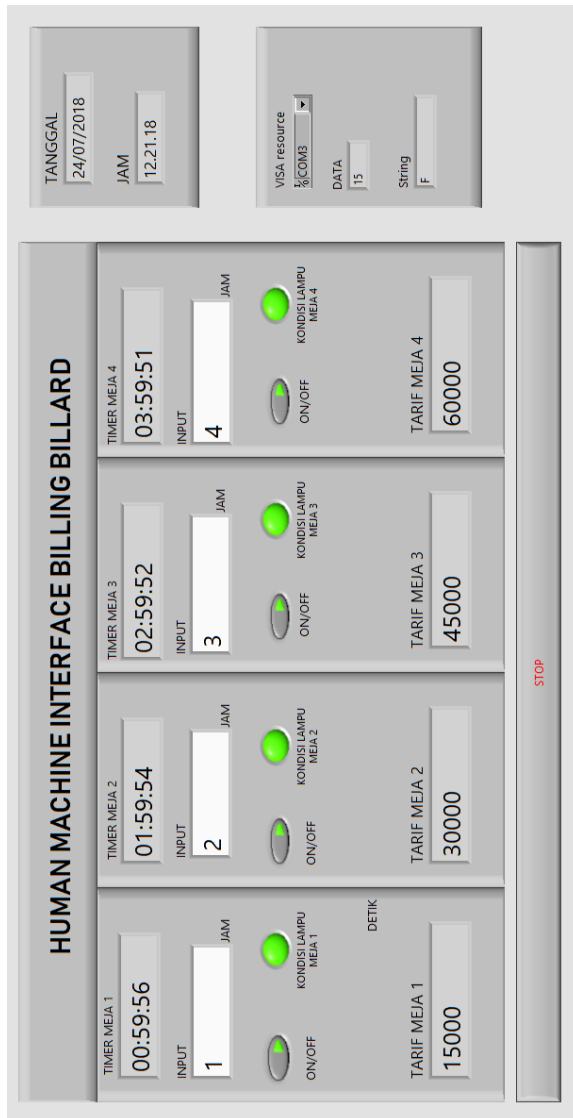
// TWI initialization
// TWI disabled
TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) |
(0<<TWIE);
while (1)
{
    // Place your code here
    while(UCSRA.7){
        int a;
        a=UDR;
        PORTC=a;
        switch (a) {
            case 0:
                PORTB=255;
                break;
            case 1:
                PORTB=112;
                break;
            case 2:
                PORTB=176;
                break;
            case 3:
                PORTB=48;
                break;
            case 4:
                PORTB=208;
                break;
            case 5:
                PORTB=80;
                break;
            case 6:
                PORTB=144;
                break;
            case 7:
                PORTB=16;
                break;
        }
    }
}

```

```
case 8:
    PORTB=224;
break;
case 9:
    PORTB=96;
break;
case 10:
    PORTB=160;
break;
case 11:
    PORTB=32;
break;
case 12:
    PORTB=192;
break;
case 13:
    PORTB=64;
break;
case 14:
    PORTB=128;
break;
case 15:
    PORTB=0;
break;
};
delay_ms(200);

}
}
```

**Lampiran B – Screenshot HMI (Human machine interface) pada LabVIEW**



--- Halaman ini sengaja dikosongkan ---

## Lampiran C – Datasheet ATMega16

### Features

- High-performance, Low-power AVR<sup>®</sup> 8-bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single-clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
  - 16K Bytes of In-System Self-Programmable Flash
    - Endurance: 10,000 Write/Erase Cycles
    - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - 512 Bytes EEPROM
    - Endurance: 100,000 Write/Erase Cycles
    - 1K Byte Internal SRAM
    - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four PWM Channels
  - 8-channel, 10-bit ADC
    - 8 Single-ended Channels
    - 7 Differential Channels in TQFP Package Only
    - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
  - Byte-oriented Two-wire Serial Interface
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
  - 32 Programmable I/O Lines
  - 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF
- Operating Voltages
  - 2.7 - 5.5V for ATmega16L
  - 4.5 - 5.5V for ATmega16
- Speed Grades
  - 0 - 8 MHz for ATmega16L
  - 0 - 16 MHz for ATmega16
- Power Consumption @ 1 MHz, 3V, and 25°C for ATmega16L
  - Active: 1.1 mA
  - Idle Mode: 0.35 mA
  - Power-down Mode: < 1 µA



8-bit AVR<sup>®</sup>  
Microcontroller  
with 16K Bytes  
In-System  
Programmable  
Flash

ATmega16  
ATmega16L

### Summary

2466NS-AVR-10/06

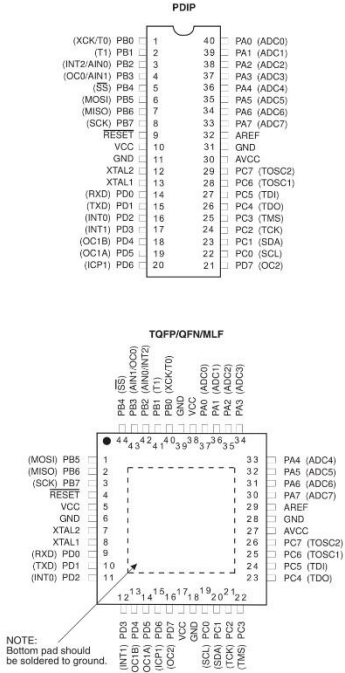


Note: This is a summary document. A complete document is available on our Web site at [www.atmel.com](http://www.atmel.com).



## Pin Configurations

Figure 1. Pinout ATmega16



## Disclaimer

Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

## 2 ATmega16(L)

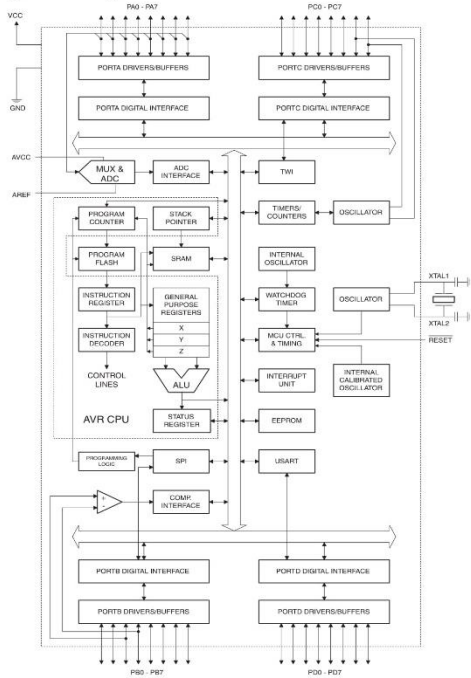
2466NS-AVR-10/06

## Overview

The ATmega16 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega16 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

## Block Diagram

Figure 2. Block Diagram





The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega16 provides the following features: 16K bytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 512 bytes EEPROM, 1K byte SRAM, 32 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary-scan, On-chip Debugging support and programming, three flexible Timer/Counters with compare modes, Internal and External Interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain (TQFP package only), a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the USART, Two-wire interface, A/D Converter, SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next External Interrupt or Hardware Reset. In Power-save mode, the Asynchronous Timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel's high density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega16 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The ATmega16 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

## Pin Descriptions

VCC	Digital supply voltage.
GND	Ground.
Port A (PA7..PA0)	Port A serves as the analog inputs to the A/D Converter.  Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.



<b>Port B (PB7..PB0)</b>	<p>Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port B also serves the functions of various special features of the ATmega16 as listed on page 56.</p>
<b>Port C (PC7..PC0)</b>	<p>Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs.</p> <p>Port C also serves the functions of the JTAG interface and other special features of the ATmega16 as listed on page 59.</p>
<b>Port D (PD7..PD0)</b>	<p>Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port D also serves the functions of various special features of the ATmega16 as listed on page 61.</p>
<b>RESET</b>	<p>Reset Input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 36. Shorter pulses are not guaranteed to generate a reset.</p>
<b>XTAL1</b>	<p>Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.</p>
<b>XTAL2</b>	<p>Output from the inverting Oscillator amplifier.</p>
<b>AVCC</b>	<p>AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to <math>V_{CC}</math>, even if the ADC is not used. If the ADC is used, it should be connected to <math>V_{CC}</math> through a low-pass filter.</p>
<b>AREF</b>	<p>AREF is the analog reference pin for the A/D Converter.</p>

**Resources**

A comprehensive set of development tools, application notes and datasheets are available for download on <http://www.atmel.com/avr>.

## ATmega16(L)

### Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
8	2.7 - 5.5V	ATmega16L-8AC	44A	Commercial (0°C to 70°C)
		ATmega16L-8PC	40P6	
		ATmega16L-8MC	44M1	
		ATmega16L-8AI <sup>(1)</sup>	44A	Industrial (-40°C to 85°C)
		ATmega16L-8AU <sup>(1)</sup>	44A	
		ATmega16L-8PI <sup>(1)</sup>	40P6	
ATmega16L-8PU <sup>(1)</sup>	40P6			
16	4.5 - 5.5V	ATmega16-16AC	44A	Commercial (0°C to 70°C)
		ATmega16-16PC	40P6	
		ATmega16-16MC	44M1	
		ATmega16-16AI <sup>(1)</sup>	44A	Industrial (-40°C to 85°C)
		ATmega16-16AU <sup>(1)</sup>	44A	
		ATmega16-16PI <sup>(1)</sup>	40P6	
ATmega16-16PU <sup>(1)</sup>	40P6			
		ATmega16-16MI <sup>(1)</sup>	44M1	
		ATmega16-16MU <sup>(1)</sup>	44M1	

Note: 1. Pb-free packaging alternative, complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.

Package Type	
<b>44A</b>	44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)
<b>40P6</b>	40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)
<b>44M1</b>	44-pad, 7 x 7 x 1.0 mm body, lead pitch 0.50 mm, Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)

## Lampiran D – Datasheet Modul Relay 4 channel

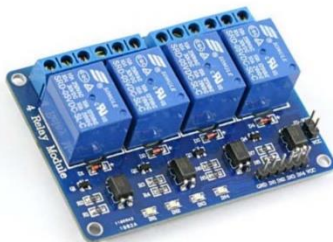


# Handson Technology

### User Guide

## 4 Channel 5V Optical Isolated Relay Module

This is a LOW Level 5V 4-channel relay interface board, and each channel needs a 15-20mA driver current. It can be used to control various appliances and equipment with large current. It is equipped with high-current relays that work under AC250V 10A or DC30V 10A. It has a standard interface that can be controlled directly by microcontroller. This module is optically isolated from high voltage side for safety requirement and also prevent ground loop when interface to microcontroller.



### Brief Data:

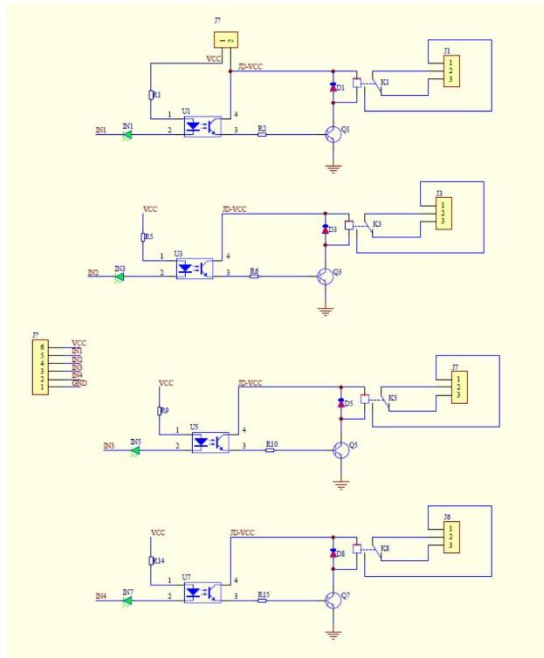
- Relay Maximum output: DC 30V/10A, AC 250V/10A.
- 4 Channel Relay Module with Opto-coupler. LOW Level Trigger expansion board, which is compatible with Arduino control board.
- Standard interface that can be controlled directly by microcontroller ( 8051, AVR, \*PIC, DSP, ARM, ARM, MSP430, TTL logic).
- Relay of high quality low noise relays SPDT. A common terminal, a normally open, one normally closed terminal.
- Opto-Coupler isolation, for high voltage safety and prevent ground loop with microcontroller.

**Schematic:**

VCC and RY-VCC are also the power supply of the relay module. When you need to drive a large power load, you can take the jumper cap off and connect an extra power to RY-VCC to supply the relay; connect VCC to 5V of the MCU board to supply input signals.

NOTES: If you want complete optical isolation, connect "Vcc" to Arduino +5 volts but do NOT connect Arduino Ground. Remove the Vcc to JD-Vcc jumper. Connect a separate +5 supply to "JD-Vcc" and board Gnd. This will supply power to the transistor drivers and relay coils.

If relay isolation is enough for your application, connect Arduino +5 and Gnd, and leave Vcc to JD-Vcc jumper in place.



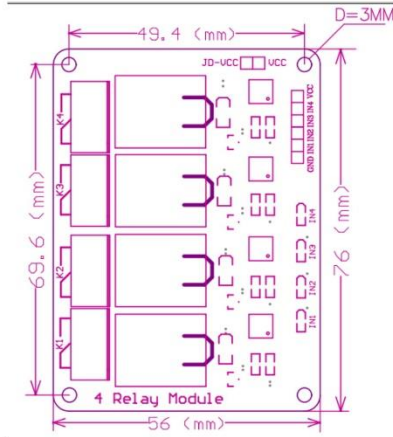
4 Channel Relay Module Schematic

It is sometimes possible to use this relay boards with 3.3V signals, if the JD-VCC (Relay Power) is provided from a +5V supply and the VCC to JD-VCC jumper is removed. That 5V relay supply could be totally isolated from the 3.3V device, or have a common ground if opto-isolation is not needed. If used with isolated 3.3V signals, VCC (To the input of the opto-isolator, next to the IN pins) should be connected to the 3.3V device's +3.3V supply.

NOTE: Some Raspberry-Pi users have found that some relays are reliable and others do not actuate sometimes. It may be necessary to change the value of R1 from 1000 ohms to something like 220 ohms, or supply +5V to the VCC connection.

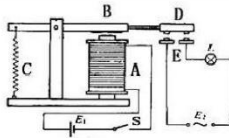
NOTE: The digital inputs from Arduino are Active LOW: The relay actuates and LED lights when the input pin is LOW, and turns off on HIGH.

**Module Layout:**



**Operating Principle:**

See the picture below: A is an electromagnet, B armature, C spring, D moving contact, and E fixed contacts. There are two fixed contacts, a normally closed one and a normally open one. When the coil is not energized, the normally open contact is the one that is off, while the normally closed one is the other that is on.



Supply voltage to the coil and some currents will pass through the coil thus generating the electromagnetic effect. So the armature overcomes the tension of the spring and is attracted to the core, thus closing the moving contact of the armature and the normally open (NO) contact or you may say releasing the former and the normally closed (NC) contact. After the coil is de-energized, the electromagnetic force disappears and the armature moves back to the original position, releasing the moving contact and normally closed contact. The closing and releasing of the contacts results in power on and off of the circuit.

### **Input:**

VCC : Connected to positive supply voltage (supply power according to relay voltage)

GND : Connected to supply ground.

IN1: Signal triggering terminal 1 of relay module

IN2: Signal triggering terminal 2 of relay module

IN3: Signal triggering terminal 3 of relay module

IN4: Signal triggering terminal 4 of relay module

### **Output:**

Each module of the relay has one NC (normally close), one NO (normally open) and one COM (Common) terminal. So there are 4 NC, 4 NO and 4 COM of the channel relay in total. NC stands for the normal close port contact and the state without power. NO stands for the normal open port contact and the state with power. COM means the common port. You can choose NC port or NO port according to whether power or not.

### **Testing Setup:**

When a low level is supplied to signal terminal of the 4-channel relay, the LED at the output terminal will light up. Otherwise, it will turn off. If a periodic high and low level is supplied to the signal terminal, you can see the LED will cycle between on and off.

For Arduino:

Step 1:

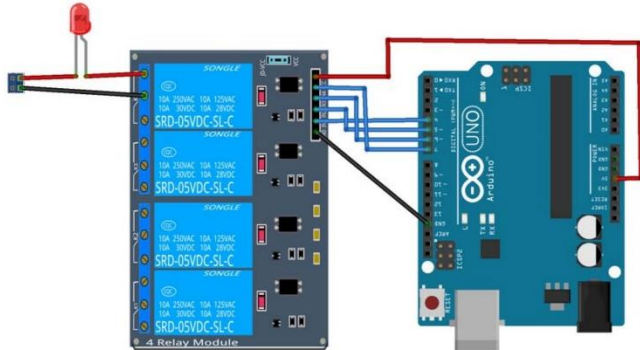
Connect the signal terminal IN1, IN2, IN3 & IN4 of 4-channel relay to digital pin 4, 5, 6, 7 of the Arduino Uno or ATmega2560 board, and connect an LED at the output terminal.

IN1> 4; IN2> 5; IN3>6; IN4>7

Step 2:

Upload the sketch "4 Channel Relay Demo " to the Arduino Uno or ATmega2560 board. Then you can see the LED cycle between on and off.

The actual figure is shown below:



#### Arduino Sketch: 4 Channel Relay Demo

```
/*
*****
Name: 4 channel_relay
Description: control the 4 channel relay module to ON or OFF
Website: www.handsontec.com
Email: techsupport@handsontec.com
*****
//the relays connect to

int RelayControl1 = 4; // Digital Arduino Pin used to control the motor
int RelayControl2 = 5;
int RelayControl3 = 6;
int RelayControl4 = 7;

void setup()
{
  Serial.begin(9600);
  pinMode(RelayControl1, OUTPUT);
  pinMode(RelayControl2, OUTPUT);
  pinMode(RelayControl3, OUTPUT);
  pinMode(RelayControl4, OUTPUT);
}

void loop()
{
  digitalWrite(RelayControl1,HIGH);// NO1 and COM1 Connected (LED on)
  delay(1000);
}
*/
```

```
digitalWrite(RelayControl1,LOW);// NO1 and COM1 disconnected (LED off)
delay(1000);
digitalWrite(RelayControl2,HIGH);
delay(1000);
digitalWrite(RelayControl2,LOW);
delay(1000);
digitalWrite(RelayControl3,HIGH);
delay(1000);
digitalWrite(RelayControl3,LOW);
delay(1000);
digitalWrite(RelayControl4,HIGH);
delay(1000);
digitalWrite(RelayControl4,LOW);
delay(1000);
}
```



## Lampiran E – Datasheet PL2303



### **PL-2303 USB to RS-232 Bridge Controller** **Product Datasheet**

---

Document Revision 1.4

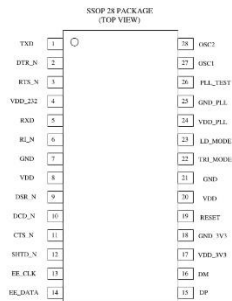
Document Release: August, 2002

**Prolific Technology Inc.**

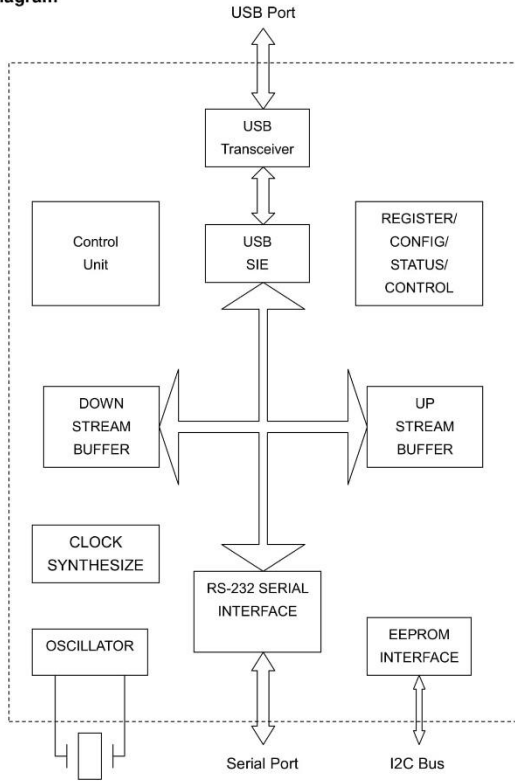
## PL-2303 USB to Serial RS232 Bridge Controller

### Features

- Full compliance with the USB Specification v1.1 and USB CDC v1.1
- Support the RS232 Serial interface
- Support automatic handshake mode
- Support Remote wake-up and power management
- 256 bytes buffer each for upstream and downstream data flow
- Support default ROM or external EEPROM for device configuration
- On chip USB transceiver
- On chip crystal oscillator running at 12M Hz
- Supports Windows 98/SE, ME, 2000, XP, Windows CE3.0, CE .NET, Linux, and Mac OS
- 28 Pins SOIC package



**Block Diagram**



## Overview

The PL-2303 operates as a bridge between one USB port and one standard RS232 Serial port. The two large on-chip buffers accommodate data flow from two different buses. The USB bulk-type data is adopted for maximum data transfer. Automatic handshake is supported at the Serial port. With these, a much higher baud rate can be achieved compared to the legacy UART controller.

This device is also compliant with USB power management and remote wakeup scheme. Only minimum power is consumed from the host during Suspend. By integrating all the function in a SOIC-28 package, this chip is suitable for cable embedding. Users just simply hook the cable into PC or hub's USB port, and then they can connect to any RS-232 devices.

## Pin Description

**Table 1. Pins Description**

Pin No.	Name	Type	Description
1	TXD	O	Data output to Serial port
2	DTR_N	O	Data Terminal Ready, active low
3	RTS_N	O	Request To Send, active low
4	VDD_232	P	RS-232 VDD. The RS-232 output signals (Pin 1 ~ Pin 3) are designed for 5V, 3.3V or 3V operation. VDD_232 should be connected to the same power level of the RS-232 interface. (The RS-232 input signals are always 5V-3V tolerant.) <i>Note: This document version only provides 5V DC characteristic information. Refer to future revisions for updates.</i>
5	RXD	I	Data input from Serial Bus
6	RI_N	I	Ring Indicator, active low
7	GND	P	Ground
8	VDD	P	Power
9	DSR_N	I	Data Set Ready, active low
10	DCD_N	I	Data Carrier Detect, active low
11	CTS_N	I	Clear To Send, active low
12	SHTD_N	O	Shut Down RS232 Transceiver
13	EE_CLK	I/O	During Reset, this pin is input for simulation purpose. During normal operation, this pin is Serial ROM clock
14	EE_DATA	I/O	Serial ROM data signal
15	DP	I/O	USB DPLUS signal
16	DM	I/O	USB DMINUS signal
17	VDD_3V3	P	3.3V power for USB transceiver
18	GND_3V3	P	3.3V ground
19	RESET	I	System Reset
20	VDD	P	Power
21	GND	P	Ground

Pin No.	Name	Type	Description
22	TRI_STATE	I	Tri-State This pin is referred after reset. High: RS-232 output inactive during Suspend. Low: RS-232 output tri-state during Suspend.
23	LD_MD/ SHTD	I/O	Load Mode/SHTD This pin is input during reset. Pull high with a 220K resistor to indicate the heavy load USB device (500mA). Pull down with a 220K resistor to indicate the light load USB device 100mA).  After reset, this pin becomes output. It output the inverse of SHTD_N.
24	VDD_PLL	P	5V power for PLL
25	GND_PLL	P	Ground for PLL
26	PLL_TEST	I	PLL test mode control
27	OSC1	I	Crystal oscillator input
28	OSC2	O	Crystal oscillator output

Type: *I* – Input signal    *O* – Output signal    *I/O* – Bi-directional signal    *P* – Power/Ground

### Supported Data Formats and Programmable Baud Rate Generator

The PL2303 USB-to-RS232 bridge controller supports versatile data formats and has a programmable baud rate generator. The supported data formats are shown on Table 2. The programmable baud rate generator supports baud rates up to 1.2M bps as shown in Table 3.

**Table 2. Supported Data Formats**

	Description
Stop bits	1 1.5 2
Parity type	None Odd Even Mark Space
Data bits	5, 6, 7, 8, or 16

**Table 3. Baud Rate Setting**

dwDTERate	Baud Rate
0012C000h	1228800
000E1000h	921600
00096000h	614400
00070800h	460800
00038400h	230400
0001C200h	115200
0000E100h	57600
00009600h	38400
00007080h	28800
00004B00h	19200
00003840h	14400
00002580h	9600
00001C20h	7200
000012C0h	4800
00000E10h	3600
00000960h	2400
00000708h	1800
000004B0h	1200
00000258h	600
0000012Ch	300
00000096h	150
0000004Bh	75

### External EEPROM and Device Configuration

PL-2303 allows storing the configuration data in an external EEPROM. After reset, the first two bytes of EEPROM are checked. If the value is 067Bh, the EEPROM is valid and the contents of the EEPROM are loaded as the chip's default parameters. Otherwise, the chip's default setting is used. The content of EEPROM is shown in Table 4 below.

The Device Configuration Register is used to control some vendor-specific functions. The meaning of each bit in Device Configuration Register is shown in Table 5. Reserved and unused pins always set to the default value.

**Table 4. EEPROM Content**

Bytes	Name	Description
1:0	EECHK	When the EEPROM is programmed, these two bytes is configured as 067B. After reset, they will be checked for the value. If matched, the following information will be loaded as the default parameters.
3:2	VID	USB Vendor ID
5:4	PID	Product ID
7:6	RN	Release number (BCD)
10:8	DCR	Device Configuration Register

**Table 5. Device Configuration Register**

Name	Bits	Definition	Default
23	RESERVED	Reserved	0
22	TRI_OUT	RS-232 Output Tri-state: 1: RS-232 output tri-state 0: RS-232 output in output mode	0
21	RW_MODE	Remote Wakeup Mode: 0: When engages remote wakeup, the device issues disconnect signal 1: When engages remote wakeup, the device issues resume signal	1
20	WURX	Enable Wake Up Trigger on RXD: 0 – Disabled; 1 – Enable Wake Up Trigger on RXD state changes.	0
19	WUDSR	Enable Wake Up Trigger on DSR: 0 – Disabled; 1 – Enable Wake Up Trigger on DSR state changes.	0
18	WURI	Enable Wake Up Trigger on RI: 0 – Disabled; 1 – Enable Wake Up Trigger on RI state changes.	1
17	WUDCD	Enable Wake Up Trigger on DCD: 0 – Disabled; 1 – Enable Wake Up Trigger on DCD state changes.	0
16	WUCTS	Enable Wake Up Trigger on CTS: 0 – Disabled; 1 – Enable Wake Up Trigger on CTS state changes.	0
15	RESERVED	Always set to one	1
14	RESERVED	Always set to zero	0
13	RESERVED	Always set to zero	0
12	RW_INH	Remote Wake Inhibit: 1 – Inhibit the USB Remote Wakeup function 0 – Enable the USB Remote Wakeup function	0
11:6	RESERVED	Always set to zero	0
5:4	RTSM	RTS Control Method: 00b – RTS is controlled by ControlBitMap. Signal is active low; 01 – RTS is controlled by ControlBitMap. Signal is active high; 10 – Drive RTS active when Downstream Data Buffer is NOT EMPTY; otherwise Drive RTS inactive. 11 – Drive RTS inactive when Downstream Data Buffer is NOT EMPTY; otherwise Drive RTS active.	0
3:1	RESERVED	Always set to zero	0
0	RSPDM	RS-232 Transceiver Shut-Down Mode: 1: Shut down the transceiver when USB SUSPEND is engaged 0: Do not shut down the transceiver even when USB SUSPEND is engaged	1

## Electrical Characteristics

### Absolute Maximum Ratings

Item	Ratings
Power Supply Voltage	-0.3 to 6.0 V
Input Voltage	-0.3 to VDD+0.3 V
Output Voltage	-0.3 to VDD+0.3 V
Storage Temperature	-55 to 150 °C

### DC Characteristics

Parameter	Symbol	Min	Typ	Max	Units
Power Supply Current	I <sub>DD</sub>	0.5	19	24	mA
Input Voltage					
Low	V <sub>IL</sub>	--	--	0.3* V <sub>DD</sub>	V
High	V <sub>IH</sub>	0.7* V <sub>DD</sub>	--	--	V
Output Voltage					
Low	V <sub>OL</sub>	--	--	0.4	V
High	V <sub>OH</sub>	3.5	--	--	V
Schmitt Trigger Threshold Voltage <sup>*1</sup>					
Negative going	V <sub>I-</sub>	--	1.10	--	V
Positive going	V <sub>I+</sub>	--	1.87	--	V
Output Voltage, 3.3V <sup>*2</sup>					
Low	V <sub>OL</sub>	--	--	0.4	V
High	V <sub>OH</sub>	2.4	--	--	V
Input Leakage Current	I <sub>L</sub>	-1	--	1	µA
Tri-state Leakage Current	I <sub>oz</sub>	-10	--	10	µA
Input Capacitance	C <sub>IN</sub>	--	3	--	Pf
Output Capacitance	C <sub>OUT</sub>	--	3	--	Pf
Bi-directional Buffer Capacitance	C <sub>BID</sub>	--	3	--	Pf
Operating Voltage Range	--	4.75	5	5.25	V
Suspend Current	I <sub>SUS</sub>	--	400	490	µA

\*1. RS232 pins RXD\_I, RI\_I, DSR\_I, DCD\_I, CTS\_I are 5V TTL Schmitt Trigger inputs.

\*2. RS232 pins TXD, DTR\_N, RTS\_N are 3.3V tri-state outputs.

### Temperature Characteristics

Parameter	Symbol	Min	Typ	Max	Units
Operating Temperature	--	-40	--	85	°C
Junction Operation Temperature	T <sub>J</sub>	0	25	115	°C



### USB Transceiver Characteristics

Parameter	Symbol	Min	Typ	Max	Units
Rise and Fall Times: (10%–90%)	$T_{R}$	4	8	15	ns
	$T_{F}$	4	8	15	ns
Cross Point	$V_{CR}$	1.3	--	2.0	V
Output Impedance	$R_{O}$	23	28	33	ohm
High Level Output	$V_{OH}$	2.8	--	--	V
Low Level Output	$V_{OL}$	--	--	0.7	V
High Level Input	$V_{IH}$	2.0	--	--	V
Low Level Input	$V_{IL}$	--	--	0.8	V

•  $C_L$ : 50pF

### Clock Characteristics

Parameter	Min	Typ	Max	Units
Frequency of Operation	11.976	12.000	12.024	MHz
Clock Period	83.1	83.3	83.5	ns
Duty Cycle	45	50	55	%

### Package Dimensions (28-Pin SSOP)

Symbol	Millimeters			Inch		
	Min	Nom	Max	Min	Nom	Max
b	0.22		0.38	0.009		0.015
E	7.40	7.80	8.20	0.291	0.307	0.323
E1	5.00	5.30	5.60	0.197	0.209	0.220
L	0.55	0.75	0.95	0.021	0.030	0.037
R1	0.09			0.004		
D	9.9	10.2	10.5	0.390	0.402	0.413
A			2.0			0.079
e		0.65			0.0256	
L1		1.25			0.050	
A1	0.05			0.020		
A2	1.65	1.75	1.85	0.065	0.069	0.073

--- Halaman ini sengaja dikosongkan ---

## **BIODATA PENULIS**



Nama : Muhammad Nurudin  
TTL : Mojokerto, 19 Juli 1995  
Jenis Kelamin : Laki-laki  
Agama : Islam  
Alamat : Jl. Kayu VII no.22 Gresik  
Telp/HP : 082141235223  
Email : uudnewbie@gmail.com

### **RIWAYAT PENDIDIKAN :**

1. 2001-2007 : SD NU 1 Gresik
2. 2007-2010 : SMPN 1 Gresik
3. 2010-2013 : SMA Darul Ulum 2 Unggulan BPPT Jombang
4. 2015-2018 : Program Studi Komputer Kontrol, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember , Surabaya

### **PENGALAMAN KERJA :**

1. Pendidikan Loka Latihan Keterampilan (LOLAPIL) angkatan 8 di PT. Petrokimia Gresik
2. Marketing Rizky and Comp. Gresik
3. Kerja Praktek di PT. Petrokimia Gresik

### **PENGALAMAN ORGANISASI**

1. Anggota Labolatorium Elektronika Dasar tahun ajaran 2016/2017