**PROYEK AKHIR – VE180626**

# RANCANGAN SISTEM PARKIR KENDARAAN DENGAN REKOMENDASI LOKASI SLOT PARKIR PADA CLOSED-SPACE MENGGUNAKAN ARDUINO MEGA 2560

Maulana Dwi Ghozali
NRP 10311500000078

Pembimbing
Ir. Joko Susila, M.T.
Ciptian Weried, S.ST, M.T.

DEPARTEMEN TEKNIK ELEKTRO OTOMASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2020

-----Halaman ini sengaja dikosongkan-----

**PROYEK AKHIR – VE180626**

# VEHICLE PARKING SYSTEM DESIGN WITH RECOMMENDED PARKING SLOT LOCATIONS IN CLOSED-SPACE USING ARDUINO MEGA 2560

Maulana Dwi Ghozali
NRP 10311500000078

Supervisor
Ir. Joko Susila, M.T.
Ciptian Weried, S.ST, M.T.

ELECTRICAL ENGINEERING AUTOMATION DEPARTMENT
Vocational Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2020

-----Halaman ini sengaja dikosongkan-----

# PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan hasil ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **"Rancangan Sistem Parkir Kendaraan Dengan Rekmomendasi Lokasi Slot Parkir Pada Closed-Space Menggunakan Arduino Mega 2560"** adalah benar-benar 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 peraturan yang berlaku.

Surabaya, Juni 2020

Maulana Dwi Ghozali

-----Halaman ini sengaja dikosongkan-----

# Rancangan Sistem Parkir Kendaraan Dengan Rekmomendasi Lokasi Slot Parkir Pada Closed-Space Menggunakan Arduino Mega 2560

## PROYEK AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan Untuk
Memperoleh Gelar Ahli Madya Teknik
Pada
Program Studi Elektro Otomasi
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Menyetujui:

| Dosen Pembimbing 1, | Dosen Pembimbing 2, |
|---|---|
| Ir. Joko Susila, M.T. | Ciptian Weried, S.ST, M.T. |
| NIP. 196606061991021001 | NPP. 1990201711060 |

**SURABAYA,**
**JUNI 2020**

-----Halaman ini sengaja dikosongkan-----

# Rancangan Sistem Parkir Kendaraan Dengan Rekmomendasi Lokasi Slot Parkir Pada Closed-Space Menggunakan Arduino Mega 2560

**Nama : Maulana Dwi Ghozali**
**Pembimbing 1 : Ir. Joko Susila, M.T.**
**Pembimbing 2 : Ciptian Weried, S.ST, M.T.**

**ABSTRAK**

Melihat keadaan area parkir setiap mall di kota Surabaya saat ini yang masih mengharuskan pengemudi untuk mencari *slot* parkir kosong secara manual. Apalagi hanya sebagian mall yang area parkirnya terdapat informasi jumlah slot parkir yang tersedia. Hal tersebut membuang waktu pengunjung yang ingin ke mall. Kejadian yang sering terjadi adalah pengendara akan berputar-putar di area parkir untuk mencari slot parkir yang kosong, terkadang pengendara tidak menemukan slot parkir yang kosong, sehingga waktu terbuang sia-sia. Proyek akhir ini dirancang sistem parkir dengan rekomendasi lokasi *slot* parkir. Sistem ini dibuat dengan menggabungkan *mikrokontroler* Arduino Mega 2560, *infrared barrier sensor*, motor servo, OLED dan *printer thermal*. Sistem ini akan didesain menyerupai tempat parkir pada umumnya yang berada pada *closed-space*. Untuk mendapatkan rekomendasi lokasi parkir, pengendara menekan *Push button* masuk. Nomor slot parkir yang terdekat dari pintu masuk dan kosong akan dicetak dengan *printer thermal*. *Infrared barrier sensor* dipasang pada setiap slot parkir untuk mendeteksi keberadaan kendaraan pada tempat tersebut. Data keberadaan kendaraan pada setiap slot parkir akan diolah oleh mikrokontroler Arduino Mega 2560. Jumlah slot parkir yang tersedia akan ditampilkan pada OLED yang terletak pada pintu masuk. Hasil dari proyek akhir ini diharapkan dapat mengurangi waktu yang terbuang untuk mencari tempat parkir.

**Kata kunci : rekomendasi lokasi slot parkir , mikrokontroler , sensor parkir**

-----Halaman ini sengaja dikosongkan-----

# Vehicle Parking System Design With Recommended Parking Slot Locations In Closed-Space Using Arduino Mega 2560

**Nama : Maulana Dwi Ghozali**
**Pembimbing 1 : Ir. Joko Susila, M.T.**
**Pembimbing 2 : Ciptian Weried, S.ST, M.T.**

**ABSTRACT**

*Seeing the current state of the parking area of every mall in the city of Surabaya which still requires drivers to manually search for empty parking slots. Moreover, only some malls have parking areas with information on the number of parking slots available. This wastes time for motorists who will visit the mall. An incident that often happens is that motorists will circle around the parking area looking for an empty parking slot, sometimes drivers do not find an empty parking slot, so that time is wasted. This final project is designed a parking system with a parking slot location recommendation. This system is made by combining the Arduino Mega 2560 microcontroller, infrared barrier sensor, servo motor, OLED and thermal printer. This system will be designed to resemble a parking lot in closed-space. To get a parking location recommendation, the driver presses the push button enter. The parking slot number closest to the entrance and empty will be printed on the thermal printer. Infrared barrier sensors are installed in each parking slot to detect the presence of vehicles in that place. The data on the presence of vehicles in each parking slot will be processed by the Arduino Mega 2560 microcontroller. The number of available parking slots will be displayed on the OLED located at the entrance. The results of this final project are expected to reduce the time wasted looking for a parking space.*

***Keywords: recommended parking slot location, microcontroller, parking sensor***

-----Halaman ini sengaja dikosongkan-----

# KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Proyek 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.

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

**Rancangan Sistem Parkir Kendaraan Dengan Rekmomendasi Lokasi Slot Parkir Pada Closed-Space Menggunakan Arduino Mega 2560**

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Ir. Joko Susila, M.T. dan Bapak Ciptian Weried, S.ST, M.T. 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 Proyek Akhir ini.

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

Surabaya, Mei 2020

Penulis

-----Halaman ini sengaja dikosongkan-----

# DAFTAR ISI

# DAFTAR GAMBAR

# DAFTAR TABEL

-----Halaman ini sengaja dikosongkan-----

# BAB I
# PENDAHULUAN

## 1.1 Latar Belakang

Melihat keadaan area parkir setiap mall di kota Surabaya saat ini yang masih mengharuskan pengemudi untuk mencari *slot* parkir kosong secara manual. Apalagi hanya sebagian mall yang area parkirnya terdapat informasi jumlah slot parkir yang tersedia. Hal tersebut membuang waktu pengunjung yang ingin ke mall. Kejadian yang sering terjadi adalah pengendara akan berputar-putar di area parkir untuk mencari slot parkir yang kosong, terkadang pengendara tidak menemukan slot parkir yang kosong, sehingga waktu terbuang sia-sia.

Untuk parkir, pengemudi mencari tempat (slot) parkir yang kosong untuk memarkirkan mobil mereka. Persoalan yang timbul adalah saat pencarian slot parkir yang masih kosong, dimana pengendara akan berputar-putar atau naik-turun untuk mencari slot parkir yang kosong tersebut. Menurut survey yang dilaksanakan di kota-kota besar di Eropa, dalam setahun penduduk Jerman menghabiskan waktu 44 jam untuk mencari parkir[1]. Masalah ini membuat penduduk Jerman menghasilkan kerugian sebesar 40.4 biliun euro. Pada proyek sebelumnya yang berjudul "Purwarupa Sistem Parkir Cerdas Berbasis Arduino Sebagai Upaya Mewujudkan *Smart City"* membahas tentang membuat sistem smart parking berbasis arduino di area parkir tidak bertingakat menggunakan visualisasi LED untuk membantu pengendara mencari slot parkir[2] dan proyek yang berjudul "*Car Parking System Using IR Sensor"* membahas tentang sistem *smart parking* menggunakan *microcontroller* dibantu dengan sensor inframerah untuk mendeteksi slot parkir yang kosong, informasi tersebut ditampilkan ke layar LCD[3]. Untuk mengatasi masalah ini dibutuhkan sebuah sistem parkir yang dapat memberitahu posisi slot parkir yang kosong, untuk mengurangi waktu yang terbuang dalam mencari slot parkir.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, maka maka masalah yang akan dipecahkan oleh penulis adalah bagaimana merancang dan membuat Sistem Parkir Kendaraan Dengan

Rekmomendasi Lokasi Slot Parkir Pada Closed-Space Menggunakan Arduino Mega 2560

## 1.3 Batasan Masalah

Batasan masalah dalam Proyek Akhir ini adalah sebagai berikut:

1. Kendaraaan yang disimulasikan pada sistem ini adalah kendaraan roda empat/ mobil penumpang
2. Sistem parkir akan dibuat dengan model miniatur yang mempresentasikan lahan parkir
3. Semua pengendara mobil diasumsikan mematuhi saran rekomendasi tempat parkir yang muncul pada karcis
4. Rancang bangun dibuat dengan Arduino Mega 2560

## 1.4 Tujuan

Tujuan utama dari Proyek Akhir ini adalah mampu merancang dan membuat sistem parkir kendaraan dengan rekomendasi slot parkir pada closed-space menggunakan arduino mega 2560. Serta mengimplementasikan alat pada masyarakat dengan efektif dan efisien.

## 1.5 Metodologi Penelitian

Dalam pelaksanaan proyek akhir berupa rancang bangun sistem parkir kendaraan dengan rekmomendasi lokasi slot parkir pada closed-space menggunakan arduino mega 2560, ada beberapa tahap yang perlu dipersiapakan yaitu sebagai berikut:

1. Pengamatan Persmasalahan

   Pada kegiatan ini penulis mendalami latar belakang persmasalahan terkait adanya sistem parkir konvensional yang ada pada sebagian besar gedung seperti mall, apartemen, perkantoran.Sistem parkir konvensional adalah sebuah sistem parkir yang mengharuskan kita sebagai pengendara untuk mencari slot kosong untuk memarkirkan kendaraan kita dengan cara mencari satu persatu. Hal tersebut sangat menghambat aktivitas dari pengendara itu sendiri dan kurang efisien.

2. Studi Literatur

Studi literatur merupakan tahapa pencarain data dan literatur untuk mencari sumber-sumber yang relevan dan dapat dipercaya sehingga dapat memperkuat penulisan proyek akhir ini. Literatur yang digunakan berasal dari jurnal, buku ilmiah, dan beberapa artikel dari internet. Selain itu, dalam kegiatan ini juga dilkakukan beberapa survey lapangan atau analisa kondisi lingkungan, guna mengetahui seberapa besar alat ini mempunyai nilai manfaat jika direalisasikan dalam kehidupan sehari-hari

3. Perancangan Alat dan Pembuatan Alat

Pada tahapan ini dilakukan perancangan dan permodelan alat, mulai dari bagian elektriknya sampai dengan bagian mekaniknya. Dan alat akan didesain sedemikian rupa agar nantinya dapat direalisasikan dalam kehidupan sehari-hari. Setelah melakukan perancangan bentuk, desain dan struktur sistem,barulah dilakukan pembuatan hardware dari sistem tersebut. Yang meliputi pembuatan sistem mekanik dan sistem elektriknya. Pembuatan alat ini menggunakan beberapa komponen meliputi Arduino Mega 2560, *Infrared Barrier Sensor*, motor servo, *printer thermal*.

4. Pengujian Alat

Pengujian ini dimaksudkan untuk memastikan bahan kinerja masing-masing komponen dari hasil pembuatan alat dapat befungsi sesuai yang diharapkan. Untuk proses pengujian *hardware* dilakukan dengan melakukan atau pengetesan *Infrared Barrier Sensor* dengan Arduino Mega 2560, push button, *printer thermal*, dan motor servo serta beberpa komponen lainnya yang telah dibuat bisa beroperasi dengan baik dan sesuai dengan bahasa pemrograman pada *software* Arduino IDE. Setelah pembuatan alat semuanya terselesaikan dengan baik, pengambilan data dan analisa data terpenuhi, maka tahap selanjutnya yaitu penyusunan laporan untuk buku tugas akhir sebagai bukti dan hasil yang telah dicapai selama pengerjaan alat hingga selesai.

## 1.6 Sistematika Laporan

Pembahasan Proyek Akhir ini akan dibagi menjadi lima bab dengan sistematika sebagi berikut:

**Bab I Pendahuluan**

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

**Bab II Teori Penunjang**

Bab ini menjelaskan tentang tinjauan pusaka, konsep dari kerja Sistem Parkir Kendaraan Dengan Rekmomendasi Lokasi Slot Parkir Pada Closed-Space Menggunakan Arduino Mega 2560, sistem program yang terfokuskan dalam arduino.

**Bab III Perncangan dan Implementasi**

Bab ini membahas percangan dan implementasi *software* dan *hardawre* Sistem Parkir Kendaraan Dengan Rekmomendasi Lokasi Slot Parkir Pada Closed-Space Menggunakan Arduino Mega 2560

**Bab IV Pengujian dan Analisis**

Bab ini memuat hasil simulasi dan pengujian serta analisis dari hasil tersebut

**Bab V Penutup**

Bab ini berisi kesimpulan dan saran hasil pembahasan yang telah diperoleh

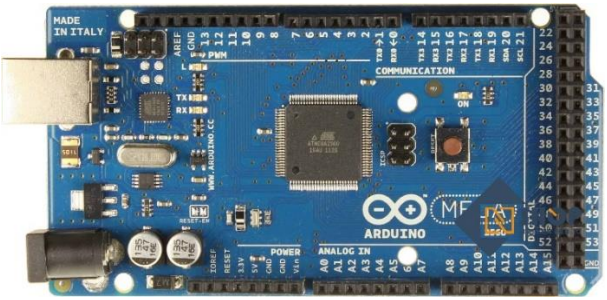
## 1.7 Relevansi

Membantu dan memudahkan pengguna kendaraan dalam mencari slot parkir untuk kendaraannya dengan cara yang efisien dengan adanya Sistem Parkir Kendaraan Dengan Rekmomendasi Lokasi Slot Parkir Pada Closed-Space Menggunakan Arduino Mega 2560.

# BAB II
# TEORI PENUNJANG

## 2.1 Arduino Mega 2560

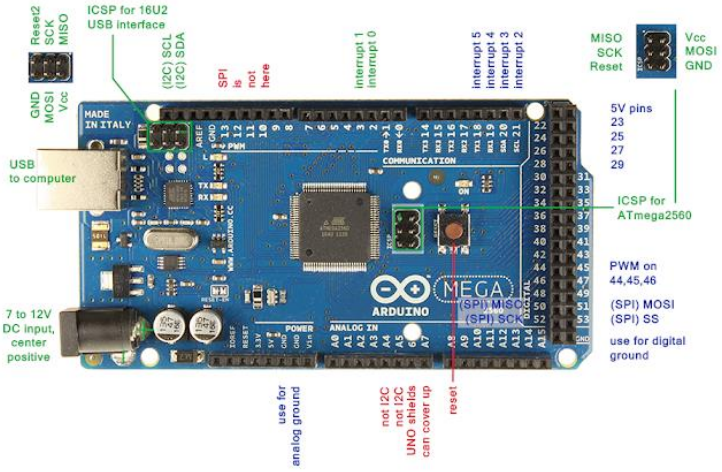
Arduino adalah board mikrokontroler dengan port Universal Serial Bus (USB) untuk menghubungkan komputer dan sejumlah socket yang terhubung dengan elektronik eksternal seperti motor, relay,sensor cahaya, dioda laser, loudspeaker, microphone, dan lain-lain[4]. Alat-alat tersebut diberi daya melalui USB dari komputer, dari baterai atau dari sebuah power supply. Alat-alat ini dapat dikendalikan oleh komputer atau diprogram oleh komputer, lalu diputus sambungannya dengan komputer sehingga dapat bekerja secara independen. Design dari *board* Arduino adalah *open-source*.[5] Datasheet Arduino Mega 2560 dapat dilihat pada Lampiran B-70. Bentuk dari *board* Arduino dapat dilihat pada Gambar 2.1




**Gambar 2.1** Board Arduino Mega 2560

Pin Digital Arduino Mega 2560 ada 54 Pin yang dapat digunakan sebagai Input atau Output dan 16 pin analog berlabel A0 sampai A15 sebagai ADC, setiap pin analog memiliki resolusi sebesar 10 bit. Arduino Mega 2560 dilengkapi dengan pin dengan fungsi khusus, sebagai berikut :

1. **Serial 4 buah** : Port Serial : Pin 0 (RX) dan Pin 1 (TX) ; Port Serial 1 : Pin 19 (RX) dan Pin 18 (TX); Port Serial 2 : Pin 17 (RX) dan Pin 16 (TX); Port Serial 3 : Pin 15 (RX) dan Pin 14 (TX). Pin RX digunakan untuk menerima data serial TTL dan Pin TX digunakan untuk mengirim data serial TTL.
2. **External *Interupts* 6 buah** : Pin 2 (Interrupt 0), Pin 3 (Interrupt 1), Pin 18 (Interrupt 5), Pin 19 (Interrupt 4), Pin 20 (Interrupt 3) dan Pin 21 (Interrupt 2).
3. **PWM 15 buah** : Pin 2,3,4,5,6,7,8,9,10,11,12,13 dan 44,45,46 pin-pin tersebut dapat digunakan sebagai Output PWM 8 bit.
4. **SPI** : Pin 50 (MISO), Pin 51 (MOSI), Pin 52 (SCK), Pin 53 (SS) digunakan untuk komunikasi SPI meggunakan SPI *Library.*
5. **I2C** : Pin 20 (SDA) dan Pin 21 (SCL) . Komunikasi I2C menggunakan *wire library*.
6. **LED :** 13. Built-in LED terhubung dengan Pin Digital 13

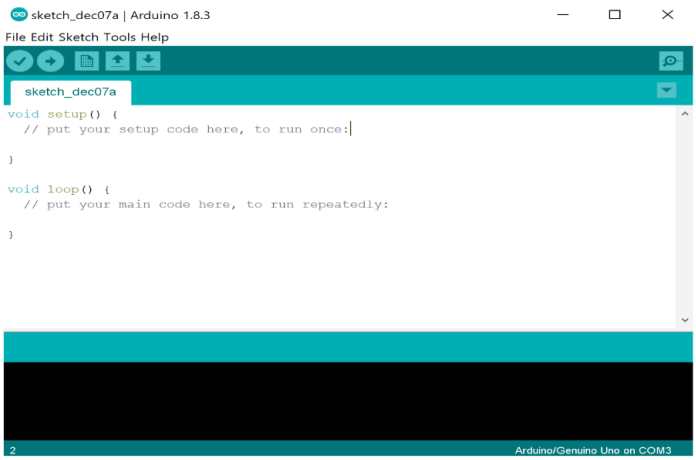



**Gambar 2.2** Arduino Mega 2560 Pinout

## 2.2 Arduino IDE

Arduino *Integrated Development Environment* (IDE) adalah aplikasi *cross-platform* (Windows, macOS, Linux) yang dibuat dengan bahasa pemrograman Java. IDE ini digunakan untuk menulis program dan mengupload ke board Arduino.

Arduino IDE mendukung bahasa C dan C++ [6]. Arduino layaknya IDE yang lain menyediakan fitur library yang mana terdiri dari class-class memliki tujuan tertentu yang dapat ditulis oleh siapa saja. Program Arduino yang ditulis oleh pengguna minimal terdiri dari 2 fungsi, fungsi *setup()* sebagai persiapan / inisialisasi pin-pin yang akan ditancapakan ke Arduino, dan fungsi *loop()* ini akan menjalankan perintah program didalamnya sampai board Arduino tersebut dimatikan[8].*Interface* dari program Arduino IDE dapat dilihat pada Gambar 2.3




**Gambar 2.3** Program Arduino IDE

## 2.3 Infrared Barrier Sensor

*Infrared Barrier Sensor* adalah sensor yang berfungsi untuk mendeteksi adanya rintangan atau benda yang berada pada jarak sensor.

Modul sensor ini memiliki sepasang pemancar dan penerima inframerah. Cara kerja dari *Infrared Barrier Sensor* sebagai berikut Frekuensi inframerah yang dipancarkan mengenai permukaan halangan/rintangan ( objek terdeteksi ) akan dipantulkan kembali dan diterima oleh bagian penerima inframerah. Setelah diproses oleh rangkaian pembanding (comparator ) , lampu hijau akan menyala dan mengeluarkan sinyal digital ( Digital Output) rendah. Jarak deteksi dapat diatur dengan potensiometer, dengan jarak efektrif 2-30 cm , tegangan kerja 3.3V – 5V. Ketika modul ini mendeteksi halangan di depan sinyal inframerah, lampu indikator warna hijau akan menyala dan port output mengeluarkan sinyal rendah secara terus menerus. Modul ini dapat mendeteksi 2-30 cm dengan sudut deteksi 35 derajat. Jarak deteksi dapat dinaikkan dengan memutar potensio searah jarum jam dan untuk mengurangi jarak deteksi dapat diputar berlawanan arah jarum jam. Sensor aktif inframerah mendeteksi pantulan, oleh karenanya bentuk pantulan dari obejek sangat penting. Permukaan warna hitam memiliki permukaan pantulan yang paling kecil dan permukaan putih memiliki pantulan yang paling besar. 3. Port output dapat dihubungkan langsung dengan IO port pada microcontroller. Dapat juga langsung dihubungkan dengan relav 5V. Menggunakan pembanding LM393, ini merupakan comparator yang stabil. Bentuk dari *Infrared Barrier Sensor* dapat dilihat pada Gambar 2.4

**Gambar 2.4** Infrared Barrier Sensor

## 2.4 Motor Servo

Motor Servo adalah aktuator putar atau aktuator linier yang mampu mengontrol posisi, kecepatan, dan percepatan sudut putar secara tepat[7] . Terdiri dari motor penggerak dan sensor untuk mengetahui posisi saat ini.Motor servo digerakkan pengontrol khusus yang dapat digerakkan dengan modul khusus untuk motor servo tersebut. Penggunaan dari motor servo antara lain : robotika, mesin numerik, dan mesin otomasi. Dalam peneletiaan ini, motor servo digunakan untuk menggerakan gerbang pintu masuk parkir. Bentuk dari motor servo dapat dilihat pada Gambar 2.5

**Gambar 2.5** Motor Servo

## 2.5 Printer Thermal

*Printer Thermal* adalah salah satu jenis printer dari keluarga printer impact, dimana printer impact ini biasanya sering digunakan sebagai printer untuk keperluan kasir pembelanjaan dan mesin fax. Proses pencetakan dari printer thermal ini dengan menggunakan gulungan kertas. Dimana proses bekerjanya dengan mengambil gulungan kertas tersebut kemudian berubah menjadi gelap saat dipanaskan.

Saat mencetak kertas akan ditarik bersebelahan dengan kepala cetak yang berisi pemanas elektronik. Mekanisme dokumen yang dicetak adalah teks dan grafis sederhana. Printer thermal ini tidak menggunakan tinta toner atau perlengkapan lainnya seperti yang dibutuhkan pada printer inkjet. Sehingga penggunaan dan pemeliharaan *printer thermal* cukup mudah. Pada proyek ini, *printer thermal* digunakan untuk mencetak tiket parkir yang berisi rekomendasi lokasi slot parkir yang tersedia. Bentuk dari *printer thermal* dapat dilihat pada Gambar 2.6


**Gambar 2.6** *Printer Thermal*

## 2.6 Display I2C OLED 0,96 inch

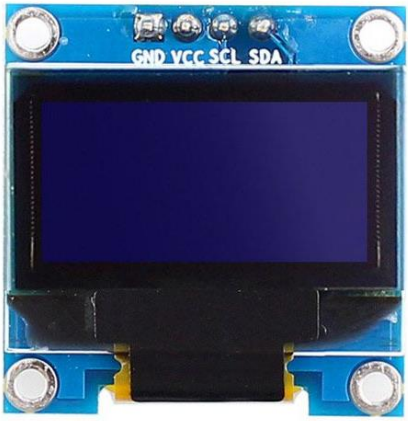
OLED 0,96 atau Organic Led adalah display grafik dengan ukuran 0,96 inci dan resolusi 128x64 piksel menggunakan teknologi OLED. Display OLED biasanya terbuat dari karbon dan hidrogen.Untuk komunikasi dengan mikrokontroler Arduino menggunakan komunikasi I2C, menggunakan 2 pin yaitu pin Sda dan pin Scl, sehingga menghemat pin.

Berbeda dengan teknologi LCD, layar OLED dapat menghasilkan cahaya sendiri dari masing-masing pikselnya dan tidak membutuhkan tambahan backlight lagi, sehingga tampilan dari layar OLED terlihat lebih terang dan jernih dan warna hitamnya benar-benar hitam pekat. Sehingga pemakaian daya relatif lebih hemat OLED

dibandingkan dengan LCD. Layar OLED 0,96 inch dapat dilihat pada Gambar 2.7

Spesifikasi Display I2C OLED 0.96 inch :

- Ukuran LCD + Board : 2,7 x 2,7 cm
- Ukuran layar LCD : 2,65 x 1,5 cm
- Resolusi layar : 128 x 32 pixel
- Warna pixel : Kombinasi kuning-biru ,full putih, full biru.
- Komunikasi : I2C / IIC
- VCC : 3,3 – 5V




**Gambar 2.7** OLED 0,96 inch

-----Halaman ini sengaja dikosongkan-----

# BAB III
# PERENCANAAN DAN IMPLEMENTASI

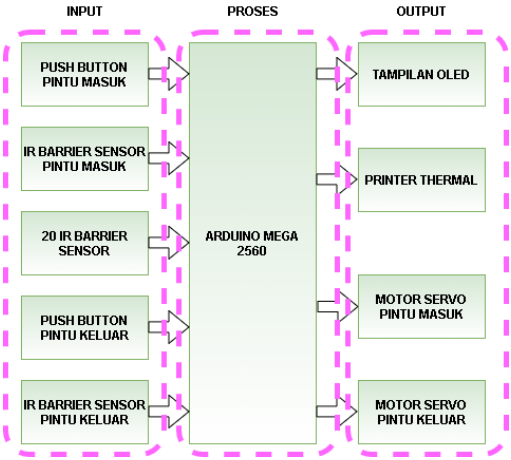
## 3.1 Gambaran Umum

Pada bab perencanaan dan implementasi akan dibahas menganai perencanaan perangkar keras (*Hardware*) dan perangkat lunak (*Software*)/ Pembahasan perangkat keras meliputi perancangan mekanik alat dan modul elektronik sedangkan perancangan pemrograman menggunakan *Software* Arduino IDE.

Perencanaan perangkat keras terdiri dari perencanaan desain mekanis dan elektronik yang mendukung IR barrier sensor dan output berupa rekomendasi lokasi yang dicetak oleh *printer thermal*. Sedangkat perangkat lunak meliputi pembuatan program yang akan di upload ke dalam board arduino.

### 3.1.1 Diagram Blok Sistem Parkir

Secara umum diagram blok perancangan sistem parkir kendaraan dengan rekomendasi slot parkir menggunakan arduino mega 2560 dapat dilihat pada Gambar 3.1 :




**Gambar 3.1** Diagram Blok Sistem Parkir

1. Push Button digunakan untuk menginisialisasi berjalannya sistem ini

2. OLED berfungsi untuk menampilkan sisa slot parkir yang tersedia.

3. Arduino mega merupakan pusat kendali dari seluruh sistem. Dimana arduino akan mengambil data yang dikirimkan oleh sensor.

4. IR barrier sensor berfungsi sebagai pendeteksi adanya kendaraan disetiap slot parkir. Data dari IR barrier sensor ini diteruskan ke arduino mega. Data yang berupa array ini, diolah oleh arduino mega sehingga mendapatkan rekomendasi slot parkir.

5. *Printer thermal* berfungsi untuk mencetak hasil data yang sudah diolah oleh arduino mega yang sudah menjadi rekomendasi slot parkir.

6. Servo berfungsi sebagai penggerak palang pintu masuk dan pintu keluar

### 3.1.2 Cara Kerja Alat

Seperti yang terlihat pada Gambar 3.1 . Sistem ini terdiri dari tiga bagian yaitu *input*, proses, dan *output*. Bagian *input* terdiri dari *push button* dan IR barrier sensor. Bagian pemroses adalah *board* Arduino Mega 2560. Bagian *output* terdiri dari tampilan OLED, *printer thermal*, dan motor servo.

Lokasi parkir terdekat dengan pintu masuk gedung diatur oleh program dari mikrokontroler dengan pengecekan ururtan. Urutan terdekat dengan pintu masuk gedung harus diisi terlebih dahulu.Untuk mendeteksi kendaraan sudah masuk ke *slot* parkir atau keluar dari *slot* parkir, dalam proyek ini digunakan IR barrier sensor yang berfungsi untuk mendeteksi kendaraan. Sensor mengirimkan data ke mikrokontroler selanjutnya akan diketahui sebagai keberadaan pada *slot* tertentu. Pada proyek ini saya asumsikan semua pengemudi parkir sesuai dengan nomor parkir yang tertera pada *printout* karcis, maka sistem ini akan dapat berjalan dengan baik sesuai yang sudah diprogram.

Pintu masuk dilengkapi dengan *push button, printer* thermal, dan motor servo. Ketika *push button* ditekan motor servo akan membuka gerbang dan *printer* mengeluarkan tiket parkir yang bertuliskan lokasi parkir yang disarankan. Palang pintu masuk dilengkapi dengan IR barrier sensor untuk mengkonfirmasi kendaraan telah melewati gerbang dan portal pintu masuk akan tertutup. Pintu keluar dilengkap dengan push button sebelum palang, motor servo, dan IR barrier sensor pintu keluar untuk konfirmasi kendaraan telah keluar dan palang pintu keluar akan tertutup. Tampilan OLED sebelum pintu masuk akan menampilkan jumlah *slot* parkir yang tersedia.

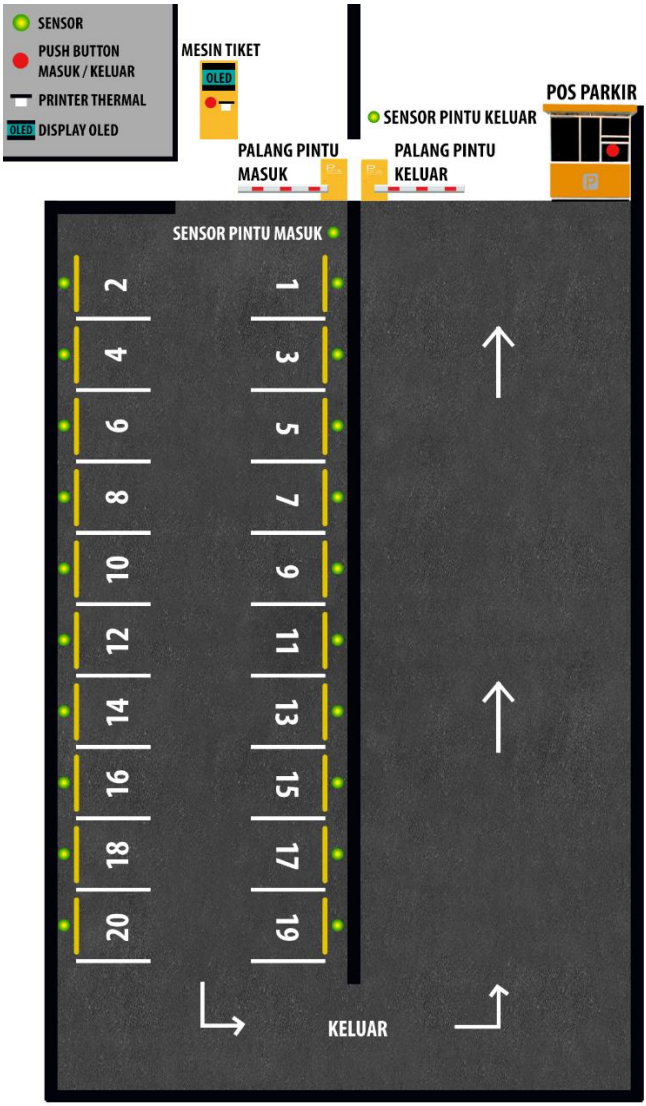
## 3.2 Perancangan Perangkat Keras

Perancangan perangkat keras meliputi perancangan modul arduino yang digunakan untuk mendukung sistem kerja alat ini.Pada input dipilih IR barrier sensor sebagai *receiver* untuk mendeteksi adanya kendaraan pada slot parkir. Arduino Mega 2560 dipilih sebagai prosesor untuk membaca dan mengolah data yng dikirim oleh IR barrier sensor. Untuk menampilkan simulasi tiap komponen, kita menggunakan *software* yang bernama Fritzing.

### 3.2.1 Desain Keseluruhan

Dalam proyek ini dirancang miniatur sistem parkir yang terdapat pada *closed-space* yang terdiri dari 1 lantai, 2 kolom dan setiap kolomnya dapat menampung 10 mobil. Jadi total kendaraan yang dapat ditampung dalam parkir ini sebanyak 20 mobil. Desain lokasi parkir diperlihatkan pada Gambar 3.2. Pada pintu masuk tempat parkir kami tempatkan push button bersebelahan dengan *printer thermal*, untuk memudahkan *user* untuk mengambil karcis parkir yang telah dicetak. Untuk palang pintu masuk akan digerakkan oleh motor servo. Setelah melewati palang, kami pasang IR barrier sensor untuk mengkonfirmasi bahwa badan kendaraan sudah sepenuhnya melewati portal. Setiap slot parkir terdapat nomor parkirnya. Kolom kiri untuk yang bernomor ganjil dan kolom kanan untuk yang bernomor genap. Karena miniatur mobil yang kami gunakan berukuran panjang 4cm,lebar 2,5cm,tinggi 2cm maka ukuran tiap slot parkir adalah panjang 4,5cm dan lebar 2,7 cm. Jarak

deteksi IR barrier sensor dengan miniatur mobil kita atur sejauh 2 cm. Pada pintu keluar kita asumsikan terdapat pos pembayaran parkir yang dioperasikan oleh seseorang. Yang dimana orang tersebut bertugas untuk mengurus pembayaran biaya parkir dan mengecek STNK (Surat Tanda Nomor Kendaraan) sudah sesuai dengan plat nomor kendaraan yang keluar.Di dalam pos pembayaran ini terdapat push button untuk membuka palang pintu keluar. Dan sama seperti pada pintu masuk, disini dipasang IR barrier sensor untuk mengkonfirmasi bahwa semua badan kendaraan sudah melewati palang pintu keluar.



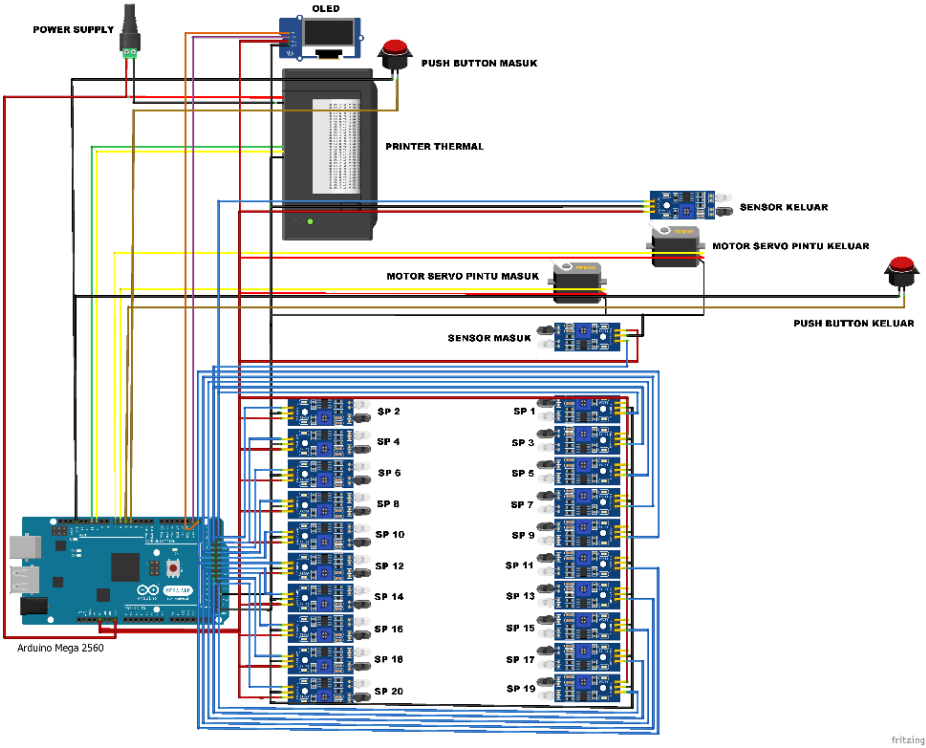

**Gambar 3.2** Desain Keseluruhan Tempat Parkir

### 3.2.2 Implementasi Keseluruhan

Jika semua bagian-bagian diatas diimplementasikan menjadi satu alat, maka wiring diagramnya menjadi seperti yang terlihat pada Gambar 3.3. Untuk keterangan koneksi pin arduino dan seluruh komponen dapat dilihat pada Tabel 3.1

**Tabel 3.1** Keterangan Wiring Seluruh Komponen

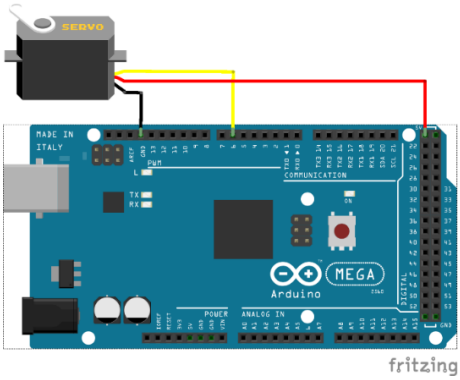
| No Pin | Keterangan Wiring Komponen |
|---|---|
| 4 | PB Masuk |
| 5 | PB Keluar |
| 6 | Servo Masuk |
| 7 | Servo Keluar |
| 10 | Kabel Kuning Printer (Tx) |
| 11 | Kabel Hijau Printer (Rx) |
| 20 | SDA (Oled) |
| 21 | SCL (Oled) |
| 22 | IR Barier Sensor Pintu Masuk |
| 23 | IR Barier Sensor Pintu Keluar |
| 26 | IR Barier Sensor Parkir 1 |
| .. | ... |
| .. | ... |
| 45 | IR Barier Sensor Parkir 20 |



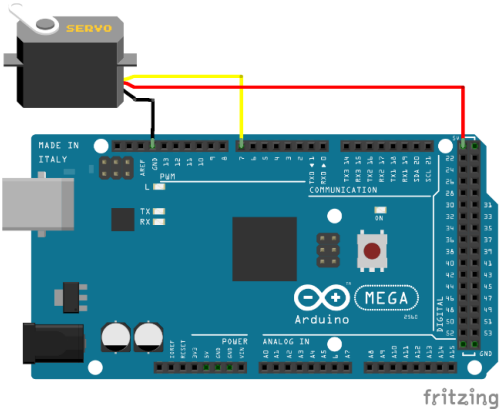

**Gambar 3.3** Wiring Keseluruhan

### 3.2.3 Implementasi Palang Pintu Dengan Motor Servo

Wiring motor servo dengan arduino mega 2560 untuk bagian palang pintu masuk dan palang pintu keluar terdapat pada Gambar 3.4 dan 3.5. VCC dan GND motor servo dihubungkan dengan pin VCC dan GND arduino. Untuk output motor servo dihubungkan dengan socket arduino nomor 6 dan 7. Socket nomor 6 untuk Motor servo pada palang pintu masuk, dan pin nomor 7 untuk motor servo pada pintu keluar.




**Gambar 3.4** Wiring Motor Servo Pada Pintu Masuk

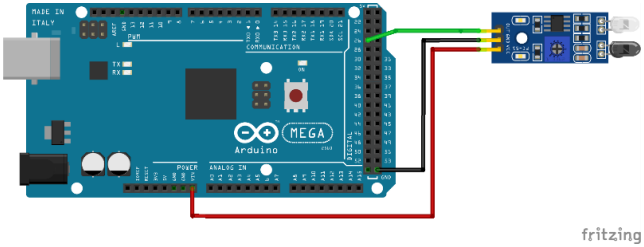



**Gambar 3.5** Wiring Motor Servo Pada Pintu Keluar

### 3.2.4 Implementasi Sensor Parkir Dengan IR Barier Sensor

Wiring IR barrier sensor dengan arduino mega 2560 terdapat pada Gambar 3.6. IR barrier sensor ini ditempatkan pada setiap titik slot parkir , dibelakang palang pintu masuk dan palang pintu keluar. VCC dan GND IR barrier sensor dihubungkan dengan socket VCC

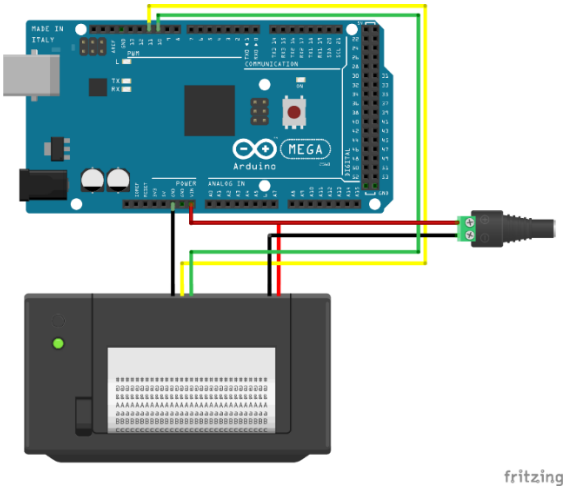
dan GND arduino mega 2560. Untuk pin output IR barrier sensor dihubungkan dengan socket 22 ( bagian palang pintu masuk ), 23 (bagian palang pintu keluar ), 26 sampai dengan 45 untuk setiap titik slot parkir.




**Gambar 3.6** Wiring IR Barrier Sensor

### 3.2.5 Implementasi Print Karcis

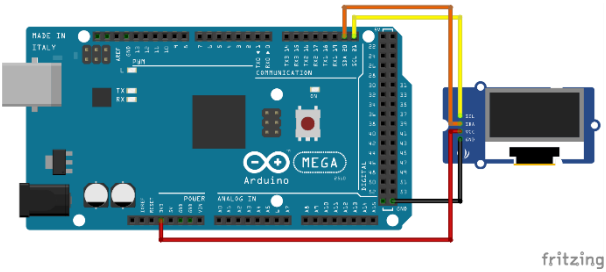
Wiring *printer thermal* dengan arduino mega 2560 terdapat pada Gambar 3.7. *Printer thermal* memerlukan tambahan daya dari *powersupply* sebesar 7 volt untuk dapat bekerja dengan baik. *Printer thermal* ini memiliki 4 buah pin, yakni pin TX, pin RX, pin VCC, dan pin GND. Untuk *printer thermal* kita memerlukan jack AC to DC sebagai power supply. Kutub positif dari jack DC diparalelkan dengan pin VCC *printer thermal* dan socker Vin arduino mega 2560. Kutub negatif dari jack DC dihubungkan dengan pin GND *printer thermal*. Kemudian pin GND printer thermal dihubungkan ke socket GND arduino mega 2560. Pin TX dan pin RX pada printer dihubungkan ke socket TX dan RX pada arduino mega 2560.




**Gambar 3.7** Wiring *Printer Thermal*

### 3.2.6 Implementasi Penampilan Kuota Dengan OLED

Wiring OLED dengan arduino mega 2560 terdapat pada Gambar 3.8. Dimana Pin VCC dan GND dari akan dihubungkan dengan socket VCC dan GND pada arduino. Pin Sda dan Scl dihubungkan dengan socket Sda dan Scl pada arduino. Lewat pin Sda dan Scl ini, data akan diolah sehingga menjadi tampilan pada OLED yang berisi informasi jumlah slot parkir yang tersedia. .




**Gambar 3.8** Wiring OLED 0,96 inch

## 3.3 Pemrograman *Software* Arduino

*Listing* program keseluruhan dari alat ini dapat diperlihatkan pada Lampiran A-54 Diagram alir sistem keseluruhan diperlihatkan pada Lampiran A-69. Awal program adalah inisialisasi setiap *port* dan pin mikrokontroler untuk penentuan pin mana yang akan digunakan sebagai *input, output,* dan komunikasi. Parkir yang disediakan untuk 20 kendaraan akan tampil pada OLED. Jika *pushbutton* masuk ditekan, maka jumlah parkir akan berkurang (*counter down)* dan ditampilan pada OLED. Kemudian dilakukan pengecekan lokasi parkir terdekat dengan pintu masuk gedung dengan metode urutan. Setelah lokasi parkir yang disarankan diperoleh, *printer thermal* akan mencetak lokasi parkir, kemudian pintu masuk dibuka. Setelah palang pintu masuk terbuka program akan menunggu konfirmasi kendaraan lewat dari IR barrier sensor pintu masuk. Setelah seluruh badan kendaraan melwati IR barrier sensor pintu masuk, kemuduian palang pintu masuk tertutup kembali.

Selanjutnya program akan melakukan pengecekan pintu keluar. Jika *push button* keluar ditekan, palang pintu keluar dibuka. Kemudian menunggu konfirmasi IR barrier sensor pintu keluar bahwa seluruh badan kendaraan telah melewati portal pintu keluar. Jika sudah melewati, maka palang pintu keluar akan ditutup kembali.

Subprogram untuk pengecekan lokasi parkir terdekat dilakukan dengan pengecekan urutan, pengecekan *slot* parkir awal jika kosong maka lokasi tersebut akan diberikan untuk dicetak oleh printer tetapi jika *slot* terisi maka dilakukan pengecekan *slot* selanjutnya, begitu seterusnya hingga urutan *slot* ke-20.

-----Halaman ini sengaja dikosongkan-----

# BAB IV

# PENGUJIAN DAN ANALISA

Pada bab ini akan dibahas mengenai pengujian alat dan analisa data dari hasil rancangan alat yang telah dibuat. Pengujian alat ini ditunjukkan untuk memastikan agar peralatan dapat berfungsi dengan baik.

Pengujian dan analisa dari alat ini meluputi analisa program arduino, pengujian alat keseluruhan. Setelah melakukan pengujian tersebut, data yang diperoleh akan dianalisa untuk mengetahui proses kerja dari seluruh sistem alat yang dibuat

## 4.1 Pengujian Motor Servo

Pengujian ini dilakukan untuk mengecek apakah motor servo dapat berputar sesuai dengan posisi yang diinginkan dengan tepat. Kami coba dengan memasukkan program yang.menginstruksikan motor servo berputar dari sudut 0 ke sudut 90 kemudian ke sudut 180. Untuk motor servo, *library* yang digunakan adalah **servo.h,** tidak perlu diinstall, cukup ditambahkan pada awal program saja. Wiring motor servo dilakukan seperti pada Gambar 3.3.

Cara menghubungkan motor servo dengan Arduino adalah sebagai berikut :

1. Motor servo mempunyai konektor *female* dengan tiga pin. Pin berwarna hitam merupakan *Ground*. Hubungkan dengan socket Arduino GND.
2. Hubungkan kabel power yang berwarna merah ke 5V pada Arduino
3. Hubungkan kabel yang tersisa pada konektor servo ke pin digital Arduino.

Setelah sudah menghubungkan motor servo dengan arduino, kita coba dengan memasukkan program yang tertera pada Gambar 4.1 berikut :

motortest

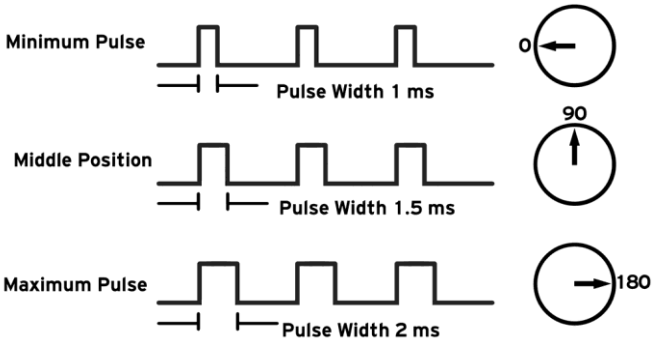
```
// Include the Servo library
#include <Servo.h>
// Declare the Servo pin
int servoPin = 6;
// Create a servo object
Servo Servo1;
void setup() {
   // We need to attach the servo to the used pin number
   Servo1.attach(servoPin);
}
void loop(){
   // Make servo go to 0 degrees
   Servo1.write(0);
   delay(1000);
   // Make servo go to 90 degrees
   Servo1.write(90);
   delay(1000);
   // Make servo go to 180 degrees
   Servo1.write(180);
   delay(1000);
}
```

**Gambar 4.1** Listing Program Uji Coba Motor Servo

Hasil dari program tersebut adalah mengubah motor servo ke 0 derajat, tunggu satu detik, lalu berputar ke 90 derajat, tunggu satu detik lagi, kemudian berputar ke 180 derajat, dan kembali ke posisi awal berulang-ulang.

Motor servo adalah perangkat yang pintar. Dengan hanya menggunakan satu pin input, mereka menerima posisi dari Arduino dan mereka menerapkannya pada gerakan motor servo sesuai dengan yang diterima. Secara internal, motor servo memiliki driver motor dan sirkuit *feedbadck* yang memastikan bahwa lengan servo mencapai posisi yang diinginkan. Tetapi jenis sinyal yang diterima pada input adalah gelombang persegi yang mirip dengan PWM. Setiap siklus dalam sinyal berlangsung selama 20 ms dan untuk sebagaian waktu, nilainya LOW. Pada awal setiap siklus, sinyalnya HIGH untuk waktu antara 1 dan 2 ms. Pada 1ms itu mewakili 0 derajat dan pada 2 milidetik itu mewakili 180 derajat. Diantaranya, ini mewakili nilai 0-180. Metode ini adalah metode

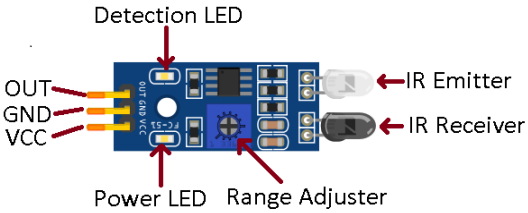
yang sangat bagus.Untuk lebih jelasnya lihat grafik yang tertera pada Gambar 4.2




**Gambar 42** Grafik Cara Kerja Motor Servo

## 4.2 Pengujian IR Barrier Sensor

Pengujian ini dilakukan untuk mengecek apakah *IR Barrier Sensor* dapat mendeteksi adanya benda pada jarak yang ditentukan.Pada pengujian ini kita atur jarak deteksi *IR Barrier Sensor Sensor* terhadap miniatur mobil sejauh 1 cm. Untuk mengatur jarak deteksi *IR Barrier Sensor*, putar sekrup pengatur jarak ( *range adjuster )* lihat pada Gambar 4.2. Jika diputar searah jarum jam, jarak deteksinya semakin besar.Jika diputar berlawanan dengan arah jarum jam, jarak deteksinya semakin kecil.




**Gambar 4.2** Keterangan Komponen *IR Barrier Sensor*

Pertama kita putar pengatur jaraknya (*range adjuster*) searah jarum jam sampai penuh, putar searah berlawanan dengan jarum jam sampai mendapatkan jarak yang diinginkan yaitu 1cm. Lakukan wiring seperti pada Gambar 3.6 dan tambahkan LED, kaki (+) positif LED pada pin nomor 22dan kaki (-) pada GND Arduino Mega 2560. Kemudian kita masukkan program pada Gambar 4.3 untuk mengetahui benda yang menghalangi *IR Barrier Sensor* terdeteksi sebagai input.

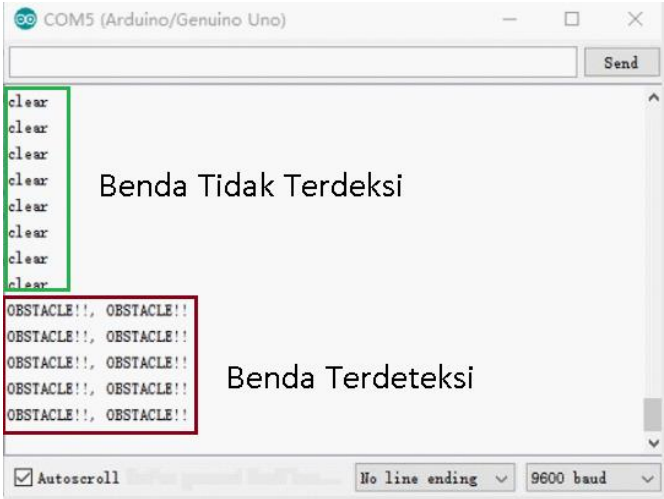
```
int LED = 22; //  LED
int isObstaclePin = 23; // This is our input pin
int isObstacle = HIGH; // HIGH MEANS NO OBSTACLE

void setup() {
    pinMode(LED, OUTPUT);
    pinMode(isObstaclePin, INPUT);
    Serial.begin(9600);
}

void loop() {
    isObstacle = digitalRead(isObstaclePin);
    if (isObstacle == LOW) {
        Serial.println("OBSTACLE!!, OBSTACLE!!");
        digitalWrite(LED, HIGH);
    } else {
        Serial.println("clear");
        digitalWrite(LED, LOW);
    }
    delay(200);
}
```

**Gambar 4.3** Listing Program Uji Coba *IR Barrier Sensor*

Cara kerja dari program pada Gambar 4.3 yaitu jika sensor mendeteksi adanya benda,maka LED akan menyala dan pada layar Arduino IDE terdapat teks “OBSTACLE!!, OBSTACLE!!”. Jika sensor tidak mendeteksi adanya benda, maka LED tidak menyala dan pada layar Arduino IDE terdapat teks “CLEAR”. Hasil program tersebut dilihat pada Gambar 4.4. Dari hasil tersebut bisa kita simpulkan bahwa *IR Barrier Sensor* yang digunakan bisa mendeteksi adanya benda , dan nilai tersebut bisa digunakan sebagai input.



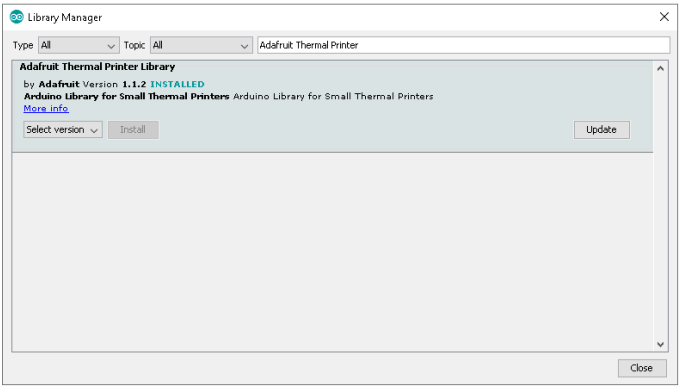

**Gambar 4.4** Hasil Uji Coba *IR Barrier Sensor*

### 4.3 Pengujian Printer Thermal

Pengujian ini dilakukan untuk mengecek apakah *printer thermal* dapat mencetak teks dengan baik. Kami uji coba dengan mencetak teks yang terdapat pada program **printer_test** yang tersedia pada *library* arduino. Wiring *printer thermal* dilakukan seperti pada Gambar 3.x

Untuk dapat mencetak menggunakan *printer thermal* dengan arduino, kita membutuhkan *library* **Adafruit Thermal Printer**. Cara menginstall *library* tersebut adalah sebagai berikut :

1. Buka Arduino IDE dan cari pada bagian **Sketch > Include Library > Manage Libraries**. Selanjutnya *Library Manager* akan terbuka.
2. Ketik “Adafruit Thermal Printer” pada kotak pencarian dan install. Untuk lebih jelasnya lihat Gambar 4.5

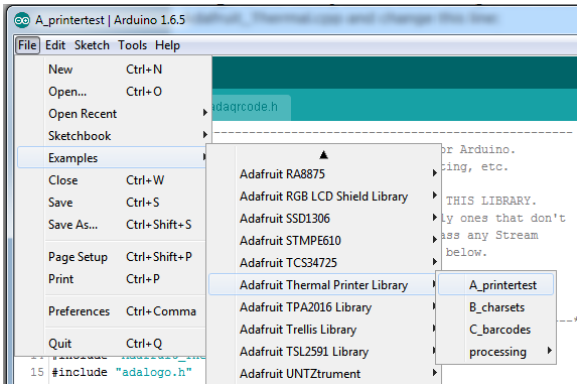



**Gambar 4.5** Instalasi *Library* Adafruit *Thermal Printer*

3. *Restart* Arduino IDE.

Setelah melakukan wiring *printer thermal* dan menginstall *library* Adafruit *thermal printer* , kita uji coba dengan mencetak teks yang terdapat pada program **printer_test** yang tersedia pada *library* arduino.Caranya adalah sebagai berikut :

1. Buka Arduino IDE dan cari pada bagian **File > Examples > Adafruit Thermal Printer Library**. Selanjutnya pilih **A_printertest.** Untuk lebih jelas lihat pada Gambar 4.6
2. Upload program dan jalankan program.




**Gambar 4.6** Program PrinterTest

Hasil dari program tersebut bisa dilihat pada Gambar 4.7. Hasil tersebut menunjukkan bahwa *printer thermal* dapat mencetak teks dengan jelas dan siap digunakkan.


**Gambar 4.7** Hasil Uji Coba Printer Thermal

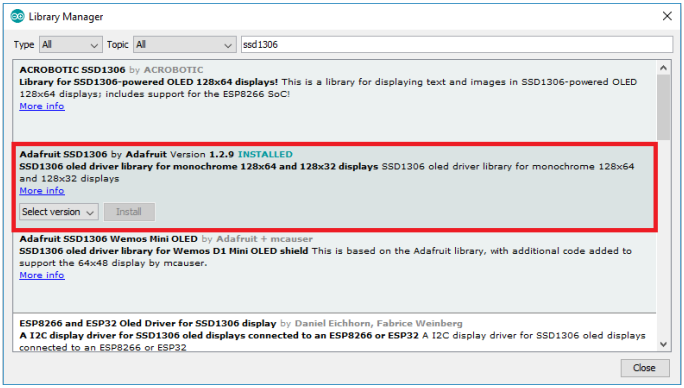
## 4.4 Pengujian Tampilan OLED

Pengujian ini dilakukan untuk mengecek apakah komponen OLED 0,96 inch dapat menampilkan tulisan pada layar OLED dengan baik. Kami coba dengan menampilkan tulisan sederhana yaitu “Hello, world!”. Wiring OLED dilakukan seperti pada Gambar 3.7.

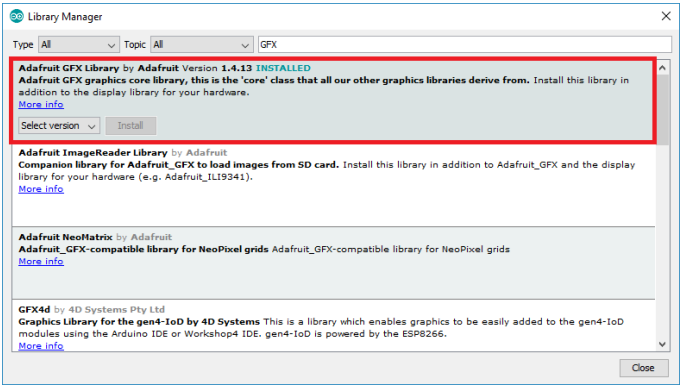
Untuk mengontrol tampilan pada OLED, kita membutuhkan *library* **adafruit_SSD1306.h** dan **adafruit_GFX.h .** Cara menginstall *library* tersebut adalah sebagai berikut :

3. Buka Arduino IDE dan cari pada bagian **Sketch** > **Include Library** > **Manage Libraries**. Selanjutnya *Library Manager* akan terbuka.
4. Ketik “SSD1306” pada kotak pencarian dan install *Library* SSD 1306 dari Adafruit. Untuk lebih jelasnya lihat pada Gambar 4.8

5. Setelah menginstall *Library* SSD1306 dari Adafruit, ketik "GFX" pada kotak pencarian dan install library tersebut. Untuk lebih jelasnya lihat pada Gambar 4.9
6. Setelah sudah menginstall kedua *library* tersebut, *restart* Arduino IDE.




**Gambar 4.8** Instalasi *Library* SSD1306




**Gambar 4.9** Instalasi *Library* GFX

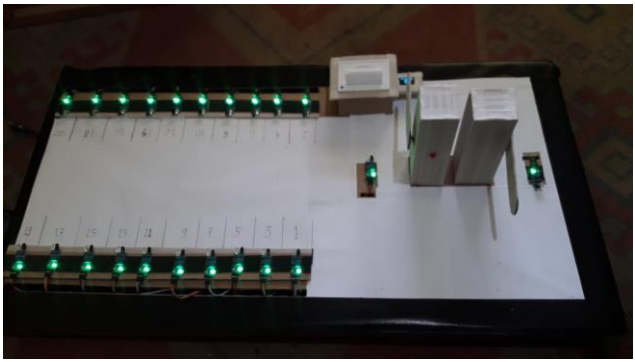
Setelah wiring OLED dan sudah menginstall *library* yang dibutuhkan, kita bisa menggunakan salah satu contoh program dari *library* ataupun memasukkan program sederhana yang kita buat sendiri. Contohnya menampilkan tulisan teks sederhana "Hello, world!". *Listing*

program untuk menampilkan teks sederhana “Hello,world!” dapat dilihat pada Lampiran A-53 . Setelah kita upload program tersebut, hasilnya dapat dilihat pada Gambar 4.10 Hasil tersebut menunjukkan bahwa komponen OLED bekerja dengan baik dan siap untuk digunakkan.


**Gambar 4.10** Tampilan Hasil Uji Coba OLED

## 4.5 Pengujian Pengisian Tempat Parkir Sampai Penuh

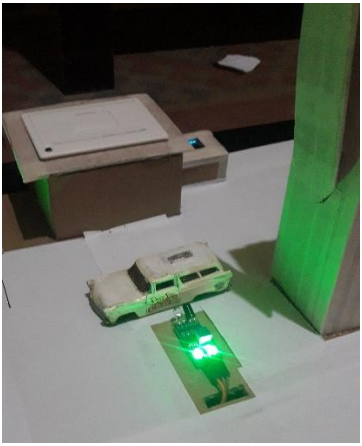
Pengujian *case* pertama dilakukan dengan kondisi awal OLED menunjukkan jumlah kapasitas parkir yaitu 20. Pengendara menekan *push button* di pintu masuk, palang pintu masuk terbuka, *printout* tiket keluar berisi lokas i parkir urutan pertama (nomor 1), kendaraan melewati gerbang dengan dideteksi IR barrier sensor pada pintu masuk, kemudian gerbang tertutup dan OLED menunjukkan sisa parkir yang tersedia sebanyak 19. Selanjutnya kendaraan terus memasuki area parkir hingga 20 kendaraan dan OLED menunjukkan angka 0, pada kondisi ini pengendara sudah tidak bisa memasuki area parkir lagi karena portal pintu masuk tidak akan terbuka. Hasil pengujian diperlihatkan pada Gambar 4.11 sampai dengan Gambar 4.16



**Gambar 4.11** Kondisi Awal

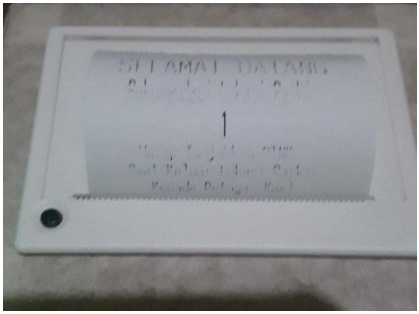

**Gambar 4.12** OLED Menampilkan Kuota Parkir 20 Unit

**Gambar 4.13** Mobil Masuk



**Gambar 4.14** Mobil Melewati IR Barrier Pintu Masuk

**Gambar 4.15** Mobil Setelah Melewati IR Barrier Pintu Masuk




**Gambar 4.16** *Printout* Tiket Mobil Pertama

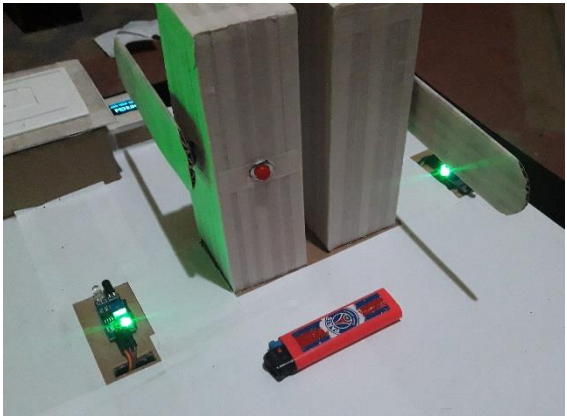
## 4.6 Pengujian Pengosongan Tempat Parkir

Pengujian *case* kedua dilakukan dengan kondisi awal OLED menunjukkan angka 0 atau pakrkir penuh. Pengendara menekan *push button* di pintu keluar, palang pintu keluar terbuka, kendaraan melewati gerbang dengan dideteksi IR barrier sensor pada pintu keluar, kemudian

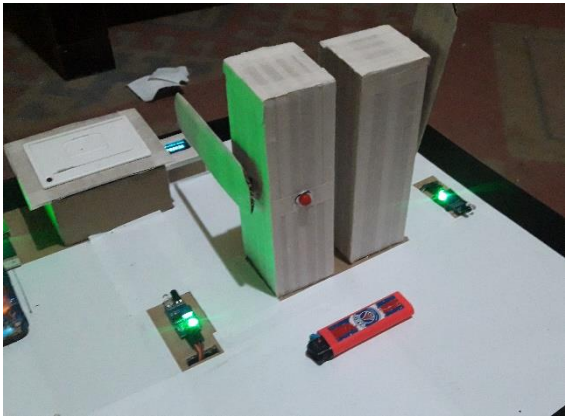
gerbang tertutup dan OLED menunjukkan tempat parkir yang tersedia sebanyak 1. Selanjutnya kendaraan terus keluar area parkir hingga OLED menunjukkan angka 20, pada kondisi ini lokasi parkir dalam keadaan kosong. Hasil pengujian diperlihatkan pada Gambar 4.17 sampai dengan Gambar 4.24

**Gambar 4.17** Kondisi Tempat Parkir Penuh

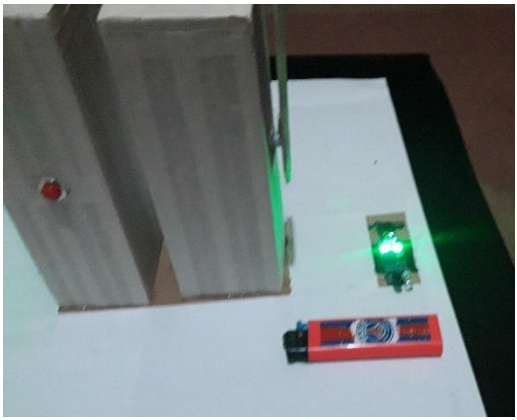

**Gambar 4.18** Tampilan OLED Saat Konidisi Parkir Penuh



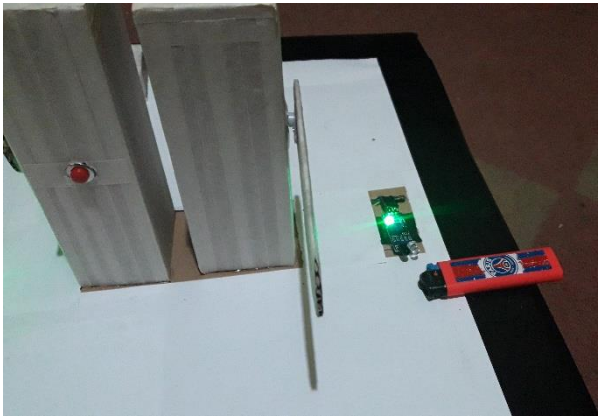
**Gambar 4.19** Mobil Akan Keluar Area Parkir



**Gambar 4.20** Palang Pintu Keluar Terbuka



**Gambar 4.21** Mobil Melewati IR Barrier Pintu Keluar



**Gambar 4.22** Mobil Setelah Melewati IR Barrier Pintu Keluar

**Gambar 4.23** Kondisi Slot Parkir Nomor 20 Setelah Ditinggalkan


**Gambar 4.24** Tampilan OLED Setelah Mobil Keluar Area Parkir

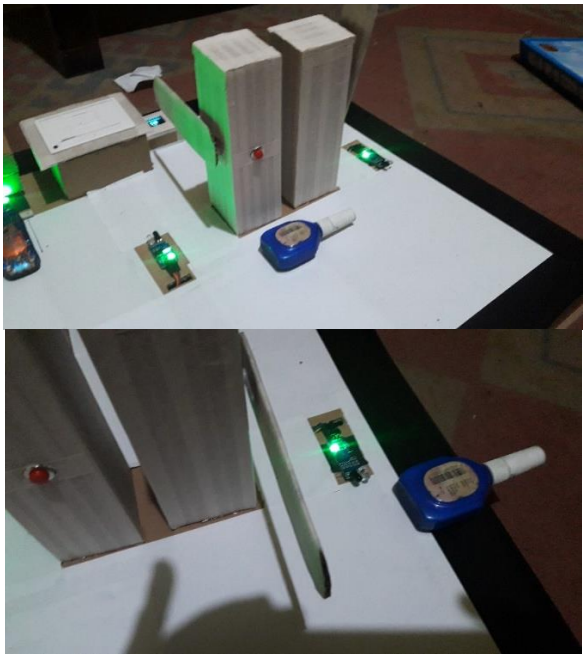
## 4.7 Pengujian Mobil Keluar Di Nomor Acak

Pengujian *case* ketiga dilakukan dengan kondisi awal lokasi parkir sudah terisi 10 kendaraan yang sudah parkir sesuai dengan urutan (1 sampai 10), satu pengendara keluar dari lokasi parkir yaitu pengendara pada urutan ke-5, OLED menunjukkan penjumlahan pada slot parkir yang tersedia yaitu 11. Beberapa saat kemudian satu

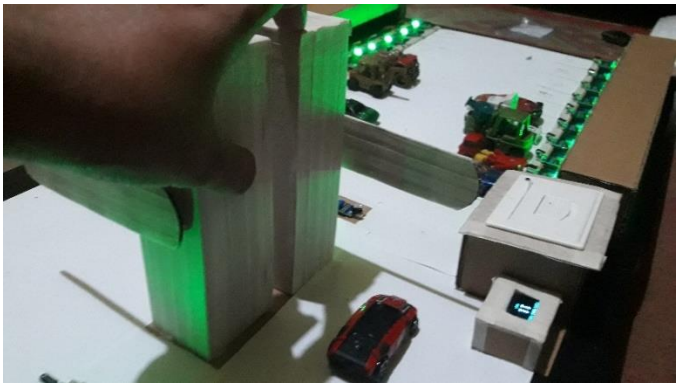
pengendara masuk dengan menekan *push button* pintu masuk, *printout* tiket yang diberikan bertuliskan lokasi slot parkir ke-5yaitu lokasi slot parkir yang ditinggalkan oleh kendaraan sebelumnya. Hasil percobaan ditunjukkan pada Gambar 4.25 sampai dengan Gambar 4.29


**Gambar 4.25** Tampilan OLED Saat Slot Parkir Tersedia 10 Unit

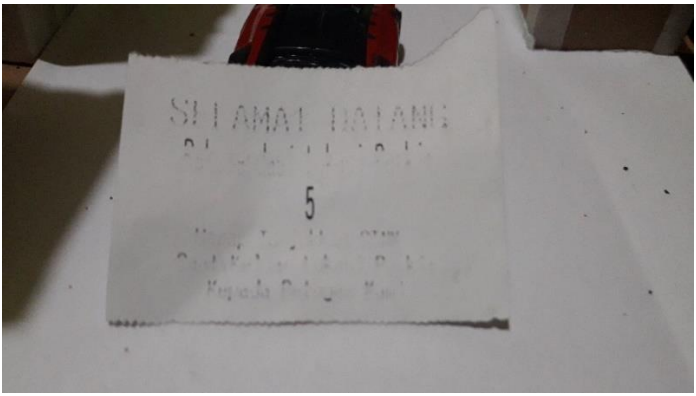
**Gambar 4.26** Posisi Kendaraan Saat Memenuhi Slot Parkir 1 Sampai Dengan Slot Parkir 10



**Gambar 4.27** Mobil Slot Parkir Nomor 5 Keluar Dari Area Parkir



**Gambar 4.28** Pengendara Lain Menekan *Push Button* Masuk




**Gambar 4.29** *Printout* Tiket Slot Parkir Nomor 5

## 4.8 Pengujian Kondisi Antrian Parkir Panjang

Pengujian *case* keempat dilakukan dengan kondisi antrian parkir sangat ramai. Pertama OLED menunjukkan sisa slot parkir adalah 20 unit. Pengendara mobil pertama yang menekan *push button* masuk, *printout* tiket yang diberikan bertuliskan lokasi slot parkir ke-1 , lalu palang pintu masuk terbuka. Kemudian pengemudi ini masuk, dideteksi oleh IR barrier sensor pintu masuk,dan palang pintu masuk tertutup kembali. Tetapi pada saat pengendara pertama belum sampai menempati slot parkir ke-1 , pengendara kedua menekan *push button* masuk. *Printout* tiket yang diberikan bertuliskan lokasi slot parkir ke-2 , lalu palang pintu masuk terbuka. Kemudian pengemudi ini masuk, dideteksi oleh IR barrier sensor pintu masuk,dan palang pintu masuk tertutup kembali. Untuk lebih jelasnya bisa dilihat pada Gambar 4.30 sampai dengan Gambar 4.34

**Gambar 4.30** Antrian Masuk Panjang

**Gambar 4.31** Mobil Pertama Menekan *Push Button*


**Gambar 4.32** *Printout* Karcis Mobil Pertama

**Gambar 4.33** Mobil Kedua Masuk Saat Mobil Pertama Belum Menempati Slot Parkir Nomor 1


**Gambar 4.34** *Printout* Karcis Mobil Kedua

## 4.9 Pengujian Push Button Masuk dan Push Button Keluar Ditekan Secara Bersama

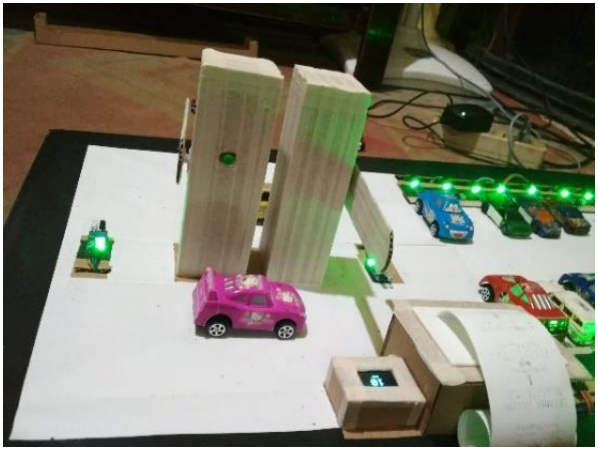
Pengujian *case* kelima ini merupakan pengujian yang dilakukan dengan cara menekan *pushbutton* masuk dan *pushbutton* keluar secara bersamaan. Tujuan pengujian kelima ini untuk mengetahui apakah pintu masuk dan pintu keluar dapat dioperasikan secara paralel. Kondisi awal tempat parkir sudah terisi sebagian ditunjukkan pada Gambar 4.35.

**Gambar 4.35** Kondisi Awal Pengujian Kelima

Kendaraan yang sudah berada dalam tempat parkir ingin keluar dari tempat parkir dan disaat yang bersamaan ada mobil lain yang ingin

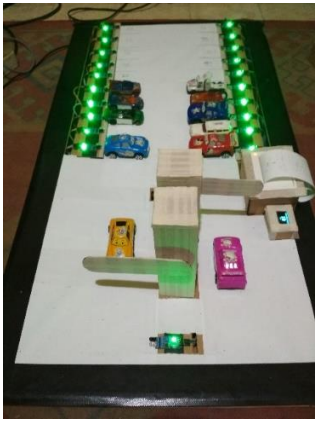
masuk ke tempat parkir, ditunjukkan pada Gambar 4.36 dan Gambar 4.37.

**Gambar 4.36** Mobil Yang Berada Dalam Area Parkir Menuju Pintu Keluar



**Gambar 4.37** Disisi Lain Mobil Ingin Masuk Area Parkir

Kedua mobil tersebut secara bersamaan menekan *pushbutton*. Ditunjukkan pada Gambar 4.38.



**Gambar 4.38** Kedua Mobil Menekan Push Button Bersamaan

Hasilnya adalah kondisi pada pintu masuk melakukan proses pencetakan tiket parkir yang berlangsung selama kurang lebih 12 detik, dilanjutkan dengan terbukanya palang pintu masuk, lihat pada Gambar 4.39. Kemudian palang pintu keluar terbuka., lihat pada Gambar 4.40

**Gambar 4.39** Palang Pintu Masuk Terbuka Dahulu

**Gambar 4.40** Selanjutnya Palang Pintu Keluar Terbuka

Kami mengambil data lama pencetakan tiket parkir dengan menggunakan *stopwatch*. Data tersebut dapat dilihat pada tabel 4.1

**Tabel 4.1** Data Waktu Pencetakan Tiket Parkir

| Nomor | Pengujian | Waktu Pencetakan Tiket Parkir (s) |
|---|---|---|
| 1 | Pengujian Pertama | 12,093 |
| 2 | Pengujian Kedua | 12,114 |
| 3 | Pengujian Ketiga | 12,039 |
| 4 | Pengujian Keempat | 12,083 |

Kami menduga hal tersebut terjadi karena terkendala oleh lama proses pencetakan tiket parkir yang memakan waktu 12 detik. Kemudiaun kami coba dengan mengurangi tulisan yang ada pada tiket parkir. Yang semula terdapat tulisan “Selamat Datang Rekomendasi Lokasi slot parkir ke-x dan jam masuk area parkir. Harap tunjukkan STNK pada petugas kami saat keluar lokasi parkir” menjadi hanya nomor lokasi parkir yang direkomendasikan dan jam masuk area parkir. Hasilnya sebagian besar sama, tetapi waktu pencetakan tiket parkir menjadi lebih singkat yang semula 12 detik menjadi 5 detik saja.

Kemudian kami coba menghilangkan perintah pencetakan tiket parkir pada program untuk sementara. Lalu kami simulasikan kembali pengujian 4.5. Kami tekan secara bersamaan *pushbutton* pintu masuk dan pintu keluar. Hasilnya adalah pintu masuk dan pintu keluar terbuka secara bersamaan juga. Sehingga dapat kami simpulkan hal yang terjadi pada pengujian 4.5 adalah proses terbuka nya pintu masuk dan pintu keluar sebenarnya terjadi secara paralel. Hanya terkendala pada kecepatan pencetakan tiket parkir oleh *printer thermal*.

# BAB V
# PENUTUP

Bab penutup ini berisi tentang kesimpulan yang diperoleh selama proses pembuatan rancagan sistem parkir kendaraan dengan rekomendasi slot parkir pada closed space menggunakan arduino mega 2560, kesimpulan dari hasil pengujian dan analisa data, serta saran untuk modul sistem parkir kendaraan dengan rekomendasi slot parkir pada closed space menggunakan arduino mega 2560 ini kedepannya.

## 5.1 Kesimpulan

1. Dengan adanya rekomendasi lokasi slot parkir, dapat menghemat waktu pengendara hingga 15 menit untuk mencari lokasi parkir
2. IR barrier sensor dapat diimplementasikan pada sistem parkir ini dengan sangat baik
3. IR barrier sensor dapat menggantikan Proximity Sensor yang digunakan pada sistem parkir pada umumnya.

## 5.2 Saran

1. Diharapkan kedepannya dapat memilih komponen-komponen yang dibutuhkan dengan lebih tepat. Karena selama perancangan alat yang lalu masih sering terjadi alat fungsi tidak maksimul maupun rusak.
2. Gunakan *printer thermal* dengan spesifikasi kecepatan pencetakan yang lebih tinggi untuk mendapatkan hasil yang lebih baik.
3. Keseluruhan alat baik *box casing* maupun tatanan lebih baik dibuat dari bahan yang kuat dan tidak mudah bengkok. Agar sewaktu dijalankan tidak menggangu kerja dari IR barrier sensor itu sendiri.
4. Untuk pembuatan miniatur tempat parkir selanjutnya diharapkan bisa lebih baik dari miniatur tempat parkir yang kami buat

# DAFTAR PUSTAKA


[1] M. Simmons, “Germans Waste 41 Hours a Year Searching for Parking,” INRIX.

[2] C. Iswahyudi, … A. P.-P., and undefined 2017, “Purwarupa Sistem Parkir Cerdas Berbasis Arduino Sebagai Upaya Mewujudkan Smart City,” Jurnal.Unmuhjember.Ac.Id, no. November, 2017.

[3] A. Agrawal, “Car Parking System Using IR Sensors,” vol. 4, no. 2, pp. 5–8, 2017.

[4] S. Monk, Programming Arduino : getting started with sketches. McGraw, 2012.

[5] J. J. (Jack J. Purdum, Beginning C for Arduino : learn C programming for the Arduino. .

[6] J. R. Castro, Building a home security system with Arduino : design, build, and maintain a home security system with Arduino Uno. 2015.

[7] D. Sawicz, “Hobby Servo Fundamentals,” 2012.

# LAMPIRAN A

***LISTING PROGRAM* TEKS "*HELLO WORLD*"**

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL
pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT,
&Wire, -1);

void setup() {
  Serial.begin(115200);

  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address
0x3D for 128x64
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
  }
  delay(2000);
  display.clearDisplay();

  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0, 10);
  // Display static text
  display.println("Hello, world!");
  display.display();
}

void loop() {

}
```

***LISTING* PROGRAM KESELURUHAN**

```
#include <SPI.h>

#include <Wire.h>

#include <Servo.h>

#include <Adafruit_GFX.h>

#include <Adafruit_SSD1306.h>

#include "Adafruit_Thermal.h"

#include "SoftwareSerial.h"

Servo myservo1,myservo2;

SoftwareSerial mySerial(10, 11);

Adafruit_Thermal printer(&mySerial);


#define SCREEN_WIDTH 128 // OLED display width, in pixels

#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL
pins)

#define OLED_RESET     12 // Reset pin # (or -1 if sharing Arduino
reset pin)

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT,
&Wire, OLED_RESET);

 int pbmasuk=5;

 int pbkeluar=4;

 int i=20;

//--------------variable Palang------------------
```

```
int penuh=0;
int kondisiA=0; // Kondisi palang masuk tertutup
int kondisiB=0; // Kondisi palang keluar tertutup
int pos1 =0;
int pos2 =0;
int s_tutuppalang_masuk=HIGH;
int s_tutuppalang_keluar=HIGH;
int state_palang_masuk=0;
int state_palang_keluar=0;
int state_lewat_pintukeluar=0;
unsigned long interval=1000; // the time we need to wait
unsigned long previousMillis=0; // millis() returns an unsigned long.
bool ledState = false; // state variable for the LED
int detik=0;
int sat_menit=0;
int pul_menit=0;
int sat_jam=8;
int pul_jam=0;
#define pin_tutuppalang_masuk 22
#define pin_tutuppalang_keluar 23
//-----------------------------------------------
int sena=HIGH;
int sens_pin[20];
```

```
int booking[20];

int sudah_scan=0;

//-----------------------------------------------


void servo_init(){

  myservo1.attach(7); // Servo masuk

  myservo2.attach(6); // Servo Keluar

}

void portalmasuk_close(){

  for (pos1 = 0; pos1 <= 90; pos1 += 1) { // goes from 0 degrees to 90
degrees

    // in steps of 1 degree

    myservo1.write(pos1);              // tell servo to go to position in
variable 'pos'

    delay(15);                         // waits 15ms for the servo to reach the
position

  }

}

void portalkeluar_close(){

  for (pos2 = 0; pos2 <= 90; pos2 += 1) { // goes from 0 degrees to 90
degrees

    // in steps of 1 degree

    myservo2.write(pos2);              // tell servo to go to position in
variable 'pos'

    delay(15);                         // waits 15ms for the servo to reach the
position
```

```
  }
}
//--------------------------
void portalmasuk_open(){
  for (pos1 = 90; pos1 >= 0; pos1 -= 1) { // goes from 180 degrees to 0
degrees
    myservo1.write(pos1);              // tell servo to go to position in
variable 'pos'
    delay(15);                         // waits 15ms for the servo to reach the
position
  }
}
void portalkeluar_open(){
  for (pos2 = 90; pos2 >= 0; pos2 -= 1) { // goes from 180 degrees to 0
degrees
    myservo2.write(pos2);              // tell servo to go to position in
variable 'pos'
    delay(15);                         // waits 15ms for the servo to reach the
position
  }
}
void init_printer() {
mySerial.begin(9600);
printer.begin();
for(int j=26;j<46;j++){
  int a=j-26;
```

```
  sens_pin[a]=j;
  pinMode(j, INPUT);
}
for(int j=0;j<20;j++){
  booking[j]=0;
}
//Serial.begin(9600);
}
void printer_awal(){
  printer.justify('C');
  printer.setSize('L');
  printer.println(" PRINTER READY ");
  printer.setSize('M');
  printer.println("CEK KONDISI SEMUA ELEMEN");
  printer.println(" ");
  printer.println(" ");
}
void print_karcis(){
//inisialisasi kode booking
for(int j=0;j<=20;j++){
sena=digitalRead(sens_pin[j]);
if((booking[j]==1)&&(sena==LOW)){ // Apakah kode booking 1
(dipesan) dan ada mobil yang parkir dilokasi j ?
```

```
  booking[j]=0;                    // Jika ya, maka atur ulang kode booking
menjadi 0

}

}

for(int j=0;j<=20;j++){

sena=digitalRead(sens_pin[j]);

if(sudah_scan==0){

if((booking[j]==0)&&(sena==HIGH)){ // Kode booking 0 (tidak ada
yang pesan tempat) dan Lokasi Parkir di j kosong.

  printer.justify('C');

  printer.setSize('L');

  printer.println(" SELAMAT DATANG ");

  printer.setSize('M');

  printer.println(" Rekomendasi Lokasi Parkir ");

  printer.setSize('M');

  printer.println(j+1);

  printer.setSize('S');

  printer.println("Jam Masuk Parkir :");

  printer.print(pul_jam);

  printer.print(sat_jam);

  printer.print(":");

  printer.print(pul_menit);

  printer.println(sat_menit);

  printer.setSize('S');
```

```
  printer.println("Harap Tunjukkan STNK  ");
  printer.println("Saat Keluar Lokasi Parkir");
  printer.println(" Kepada Petugas Kami ");
  printer.setSize('M');
  printer.println(" ");
  sudah_scan=1;
  booking[j]=1;
}
}
}
if(sudah_scan==1){
  sudah_scan=0;
}
if(sudah_scan==1){
  sudah_scan=0;
}
}
void init_pb(){
  pinMode(pbmasuk,INPUT);
  pinMode(pbkeluar,INPUT);
  digitalWrite(pbmasuk,HIGH);
  digitalWrite(pbkeluar,HIGH);
}
```

```
void pb_keluar(){
s_tutuppalang_keluar=digitalRead(pin_tutuppalang_keluar);
if(kondisiB==0){
  if(!digitalRead(pbkeluar)==HIGH){
    portalkeluar_open();
    kondisiB=1;
  }
 }
 if(kondisiB==1){
  if(s_tutuppalang_keluar==LOW){
    state_palang_keluar=1;
  }
 }
 if((state_palang_keluar==1)&&(s_tutuppalang_keluar==HIGH)){
    i++;
    state_palang_keluar=0;
    kondisiB=0;
    portalkeluar_close();
  }
}
void pb_masuk(){
 s_tutuppalang_masuk=digitalRead(pin_tutuppalang_masuk);
 if(kondisiA==0){
```

```
 if(!digitalRead(pbmasuk)==HIGH){ // Apakah PB Masuk ditekan ?
   if(penuh==0){
     print_karcis();
     portalmasuk_open();
     kondisiA=1;
   }
  }
 }
 if(kondisiA==1){
  if(s_tutuppalang_masuk==LOW){ // Apakah ada yang melintas ?
    state_palang_masuk=1; // Ada mobil masih lewat
  }
 }
 if((state_palang_masuk==1)&&(s_tutuppalang_masuk==HIGH)){ //
Apakah sudah selesai lewat mobilnya ?
    i--;
    state_palang_masuk=0;
    kondisiA=0;
    portalmasuk_close();
  }
}
void sensor_tutup_init(){
  pinMode(pin_tutuppalang_masuk,INPUT); // Sensor untuk palang
masuk
```

```
  pinMode(pin_tutuppalang_keluar,INPUT); // sensor untuk palang
keluar

}



void oled_setup() {

  // put your setup code here, to run once:

  Serial.begin(9600);


  // SSD1306_SWITCHCAPVCC = generate display voltage from
3.3V internally

  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {

    Serial.println(F("SSD1306 allocation failed"));

  //  for(;;); // Don't proceed, loop forever

  }


  // Show initial display buffer contents on the screen --

  // the library initializes this with an Adafruit splash screen.

  display.display();

  delay(2000); // Pause for 2 seconds


  // Clear the buffer

  display.clearDisplay();
```

```
  // Draw a single pixel in white
  display.drawPixel(10, 10, SSD1306_WHITE);

  // Show the display buffer on the screen. You MUST call display()
after
  // drawing commands to make them visible on screen!
  display.display();
  delay(2000);
}

void setup(){
  oled_setup();
  init_pb();
  init_printer();
  printer_awal();
  servo_init();
  sensor_tutup_init();
 pinMode(13, OUTPUT);
 digitalWrite(13, ledState);
}

void loop() {
  // put your main code here, to run repeatedly:
unsigned long currentMillis = millis(); // grab current time
```

```
 // check if "interval" time has passed (1000 milliseconds)
if ((unsigned long)(currentMillis - previousMillis) >= interval) {
   detik++;
   previousMillis = millis();
 }
if (detik>60){
   sat_menit++;
   detik=0;
 }
if (sat_menit>9){
   pul_menit++;
   sat_menit=0;
   detik=0;
 }
if(pul_menit>5){
   sat_jam++;
   sat_menit=0;
   detik=0;
   pul_menit=0;
}
if(sat_jam>9){
   sat_menit=0;
```

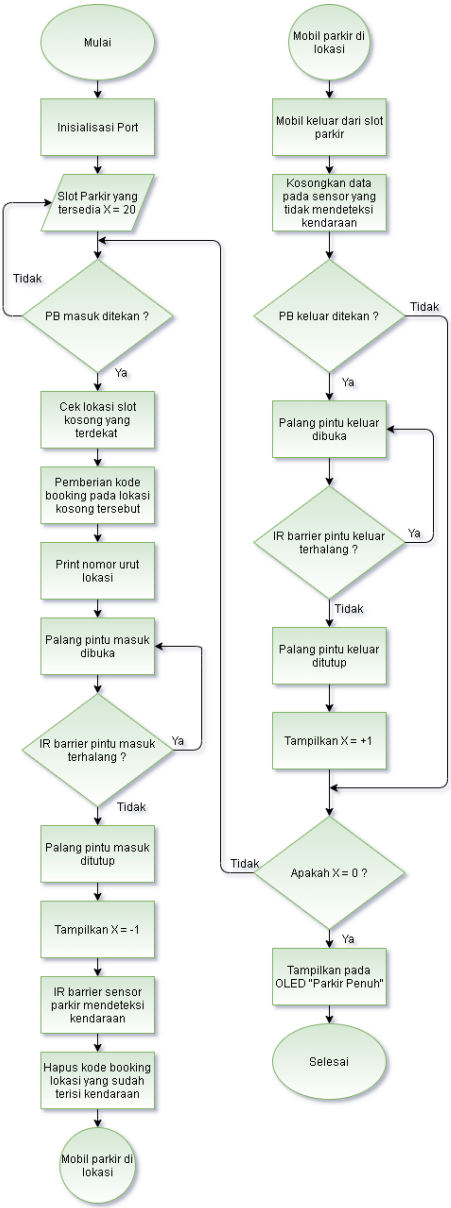
```
   detik=0;
   pul_menit=0;
   sat_jam=0;
   pul_jam++;
}

if (pul_jam==2){
if (sat_jam==4){
  detik=0;
  sat_menit=0;
  sat_jam=0;
  pul_menit=0;
  pul_jam=0;
  }
}

if(i<=0){
  i=0;
}
if(i==0){
  print_layar_penuh();
  penuh=1;
}
```

```
if(i>=20){
  i=20;
}
if(i>=1){
  print_layar_kuota(i);
  penuh=0;
}
pb_masuk();
pb_keluar();
}
void print_layar_penuh(void){
  display.clearDisplay();
  display.setTextSize(1);
  display.setCursor(10,0);            // Start at top-left corner
  display.println(F("KUOTA TEMPAT PARKIR"));
  display.setTextSize(3);             // Normal 1:1 pixel scale
  display.setTextColor(SSD1306_WHITE);        // Draw white text
  display.setCursor(20,25);            // Start at top-left corner
  display.println(F("PENUH"));
  display.display();
}
void print_layar_kuota(int i) {
 int koordinat;
```

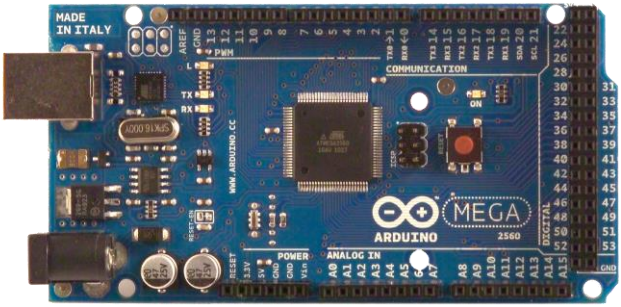
```
if(i>9){
  koordinat=45;
  }else{
  koordinat=55;
 }
  display.clearDisplay();
  display.setTextSize(1);             // Normal 1:1 pixel scale
  display.setTextColor(SSD1306_WHITE);        // Draw white text
  display.setCursor(10,0);             // Start at top-left corner
  display.println(F("KUOTA TEMPAT PARKIR"));
  display.setCursor(koordinat,17);
  display.setTextSize(4);
  display.println(i);
  display.setTextSize(1);
  display.setCursor(55,53);
  display.println("Unit");
  display.display();
}
```


Mulai
Inisialisasi Port
Slot Parkir yang tersedia X = 20
Tidak
PB masuk ditekan ?
Ya
Cek lokasi slot kosong yang terdekat
Pemberian kode booking pada lokasi kosong tersebut
Print nomor urut lokasi
Palang pintu masuk dibuka
IR barrier pintu masuk terhalang ?
Ya
Tidak
Palang pintu masuk ditutup
Tampilkan X = -1
IR barrier sensor parkir mendeteksi kendaraan
Hapus kode booking lokasi yang sudah terisi kendaraan
Mobil parkir di lokasi
Mobil parkir di lokasi
Mobil keluar dari slot parkir
Kosongkan data pada sensor yang tidak mendeteksi kendaraan
PB keluar ditekan ?
Tidak
Ya
Palang pintu keluar dibuka
IR barrier pintu keluar terhalang ?
Ya
Tidak
Palang pintu keluar ditutup
Tampilkan X = +1
Tidak
Apakah X = 0 ?
Ya
Tampilkan pada OLED "Parkir Penuh"
Selesai

# LAMPIRAN B

## DATASHEET ARDUINO MEGA 2560

# Arduino MEGA 2560




## Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

## Index

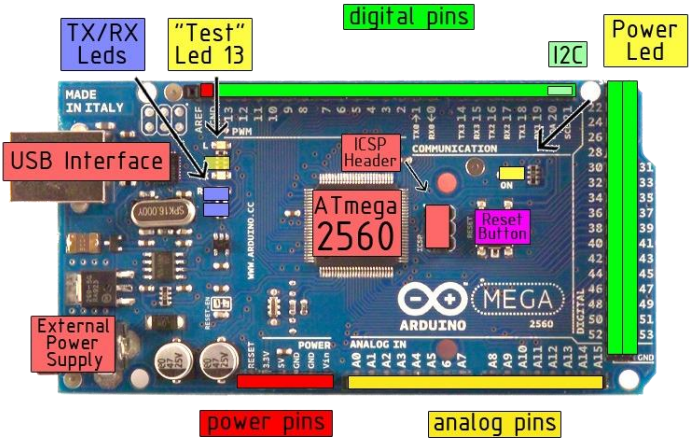
# Technical Specification

EAGLE files: arduino-mega2560-reference-design.zip Schematic: arduino-mega2560-schematic.pdf

## Summary

| | |
|---|---|
| Microcontroller | ATmega2560 |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limits) | 6-20V |
| Digital I/O Pins | 54 (of which 14 provide PWM output) |
| Analog Input Pins | 16 |
| DC Current per I/O Pin | 40 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 256 KB of which 8 KB used by bootloader |
| SRAM | 8 KB |
| EEPROM | 4 KB |
| Clock Speed | 16 MHz |

## the board



## Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

## Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

## Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **$I^2C$: 20 (SDA) and 21 (SCL).** Support $I^2C$ (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the $I^2C$ pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

## Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Mega's digital pins.

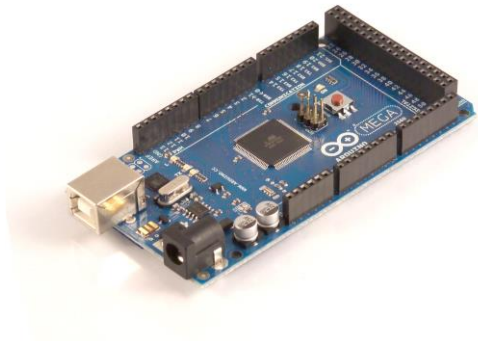
The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation on the Wiring website for details. To use the SPI communication, please see the ATmega2560 datasheet.

## Programming

The Arduino Mega2560 can be programmed with the Arduino software (download). For details, see the reference and tutorials.

The Atmega2560 on the Arduino Mega comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.



## Automatic (Software) Reset

Rather then requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

## USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

## Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. **Please note that $I^2C$ is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).**

# How to use Arduino

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platoform program. You'll have to follow different instructions for your personal OS. Check on the Arduino site for the latest instructions. *http://arduino.cc/en/Guide/HomePage*

**Linux Install** **Windows Install** **Mac Install**

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

## Blink led

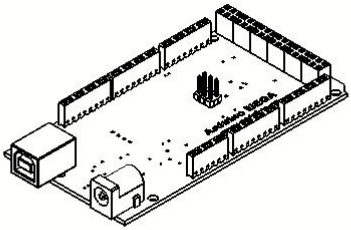
Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

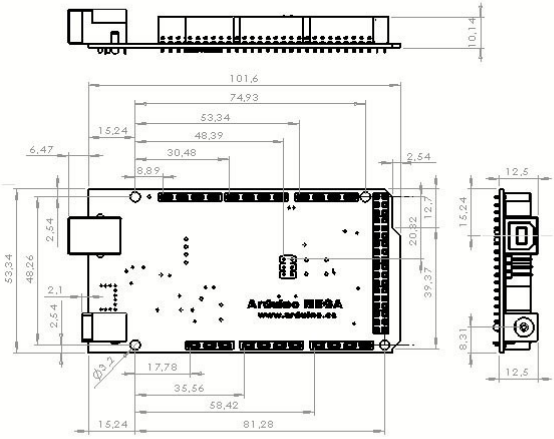
**File>Sketchbook>**
**Arduino-0017>Examples>**
**Digital>Blink**

Once you have your skecth you'll see something very close to the screenshot on the right.

In **Tools>Board** select MEGA

Now you have to go to
**Tools>SerialPort**
and select the right serial port, the one arduino is attached to.


10,14
101,6
74,93
53,34
15,24
48,39
6,47
30,48
2,54
8,89
12,5
15,24
12,7
20,82
2,54
53,34
48,26
39,37
2,1
2,54
8,31
Arduino MEGA
www.arduino.cc
Ø3,2
17,78
35,56
58,42
15,24
81,28
12,5

# Terms & Conditions

## 1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warrantyperiod.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino ™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

## 2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

## 3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

## 4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

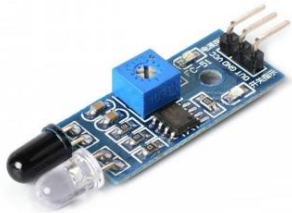
# Enviromental Policies

The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.

# DATASHEET IR BARRIER SENSOR

FC-51: IR Infrared Obstacle Detection Sensor Module 2 - 30cm

Giorgio De Nunzio - Giovanni Marsella



**0 Foreword: infrared (IR)**

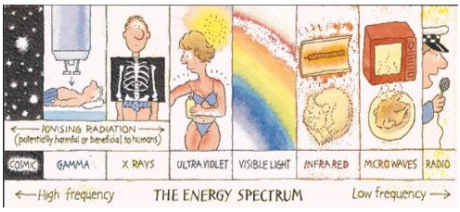



Fig. 0 - The spectrum of electromagnetic waves, in which the portion of infrared rays is present ( *infrared)* ( from http://galileo.phys.virginia.edu/outreach/8thgradesol/RadiationProtectionFrm.htm, image from the Uranium Information Center, Melbourne, Australia)

**1. Introduction**

The presence of an object (for example, an obstacle in front of a robot) can be detected by an infrared system consisting of an IR transmitter and receiver.

In more detail, a **IR transmitter,** or IR LED, sends an infrared signal at a particular wavelength compatible with a **IR receiver,** who has the task of detecting it.

There are different types of IR sensors for different types of applications. IR technology is used, for example, in proximity sensors to detect an object in the immediate vicinity, in contrast sensors to identify a path traced on the floor, or in counting sensors to count objects that pass in front of the sensor.

## 2 Operating principle

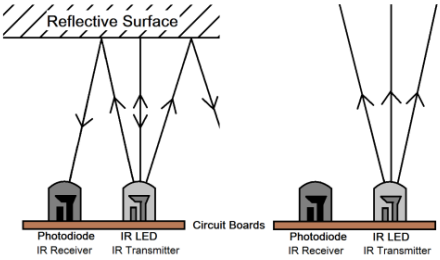



Fig. 1 - Operating principle of an IR sensor with and without an object in the vicinity.

The IR transmitter sends an infrared signal which, in the presence of a reflective surface (especially if white in color), "bounces" in various directions, including that along which the radiation hits the IR receiver, which captures the signal by detecting the object, and signaling it through one of its pins.

In the case of an absorbent surface (for example black in color) the IR signal is minimally reflected, with the consequence that the object is hardly detected by the sensor.

### 2.1 IR transmitter and IR receiver

The IR transmitter is a particular LED that emits radiation in the infrared frequency range, invisible to the naked eye. An infrared LED works exactly like an LED in the visible, except for the wavelength emitted.

The working voltage is 3.3-5V DC and the current consumption of about 20mA. The IR receiver, (a photodiode), is able to detect infrared radiation emitted by an IR transmitter. Aesthetically it is similar to an LED but the outer capsule can be wrapped in a dark colored film.

### 3 FC-51 IR sensor

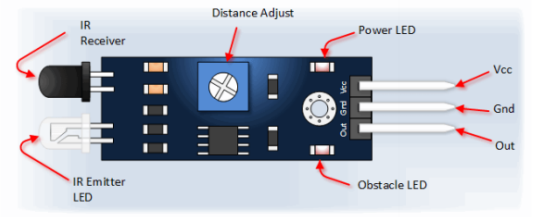



Fig. 2 - The Pin map of the FC-51 sensor.

The sensor used in these demos is the model **FC-51.** It is an economic sensor easily available on the internet for less than € 2.

#### 3.1 Pin configuration and electronic scheme

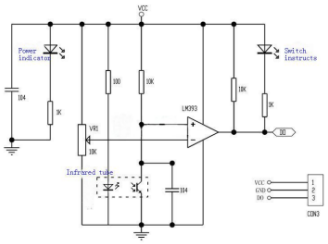



Fig. 3 - Scheme of an FC-51 IR sensor. The LM393 integrated circuit is an open collector voltage comparator which provides an output if there is a pull-up resistance between the IC output (DO) and the power supply Vcc (R = 10KΩ ): the DO output is high if the object is not detectable, low if the object is detectable.

The component has three connection pins:

1. Vcc for 3.3-5V DC power supply;

2. Gnd for mass reference;
3. Out for the sensor digital output signal.

**The Power LED on the device lights up when it is powered. The Obstacle LED lights up when an obstacle is detected.**

The sensor output is high (HIGH) if an obstacle is not detected (the LED receiver does not receive reflected signals), it is low (LOW) in the presence of an obstacle.

This sensor detects **objects at a distance between 2 and 30cm.** Using the potentiometric trimmer it is possible to calibrate the sensitivity according to the application and the environmental conditions (for example the brightness). **By turning the potentiometer counterclockwise, the distance at which the sensor detects the object decreases, turning it clockwise this distance increases.**

It should be noted that this is a rather "rough" device, so it can happen that an obstacle is not detected if it is dark in color or if it still absorbs the IR waveforms, and that the accuracy and effective range depend on the sensor quality (highly variable) and the material from which the object to be detected is made.

## 4 Some demos

### 4.1 IR sensor FC-51 test via serial terminal (Demo 01)

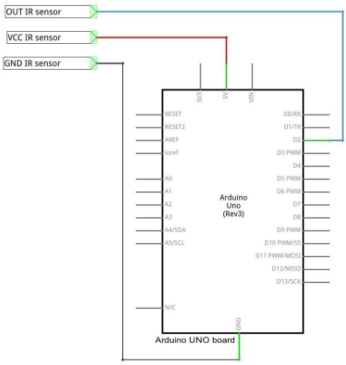



Fig. 4 - Demo scheme 1.

In the first demo, through the connection between the Arduino serial port and the PC, we will read the information on the detection of the object.

Let's take a look at the steps required by this demo:

1. We connect the sensor output pin to the Arduino digital pin 2 which call **IR.**
2. The function **setup ()** is performed only once before the loop function. We insert here the initialization code that enables the Arduino serial port and sets digital pin 2 as an input.

3. The function **loop ()** it is the main function and is cyclically repeated until it is turns off the Arduino board. We convert the functioning of the electronic circuit analyzed previously into C language. We save in the variable **detection** the value taken from the IR pin by the function **digitalRead:** if the value is low then there is an object in front of the sensor, otherwise it is not there (or is not detectable).

Here is the code:

```
# define IR 2 int detection;
void setup () {

    Serial.begin (9600); pinMode (IR,
  INPUT); }

void loop () {
    detection = digitalRead (IR); if (detection == LOW) {

        Serial.print ("There is an obstacle! \ N"); }

    else {
        Serial.print ("No obstacle! \ N"); }

    delay (500);                    // in ms
}
```

**4.2 IR sensor FC-51 and LED (Demo 02)** ( from site [2])

This demo adds the lighting of the LED on the Arduino board when an obstacle is detected. The Arduino pin used is different, but only for the different origin of the application example. Any digital pin is usable.

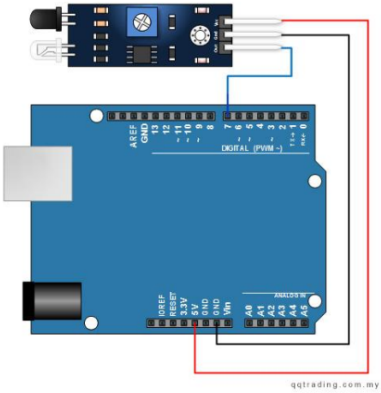



Fig. 5 - Demo scheme 2.

```
// IR Obstacle Collision Detection Module

int LED = 13; // Use the onboard Uno LED int isObstaclePin = 7; // This is our input pin
int isObstacle = HIGH; // HIGH MEANS NO OBSTACLE


void setup () {
        pinMode (LED, OUTPUT);
        pinMode (isObstaclePin, INPUT); Serial.begin (9600); }



void loop () {
        isObstacle = digitalRead (isObstaclePin); if (isObstacle == LOW) {

                Serial.println ("OBSTACLE !!, OBSTACLE !!"); digitalWrite (LED, HIGH); }
            else {

                Serial.println ( "clear"); digitalWrite (LED,
            LOW); }

        delay (200); }
```

4.3 IR sensor FC-51 and LED (Demo 03)

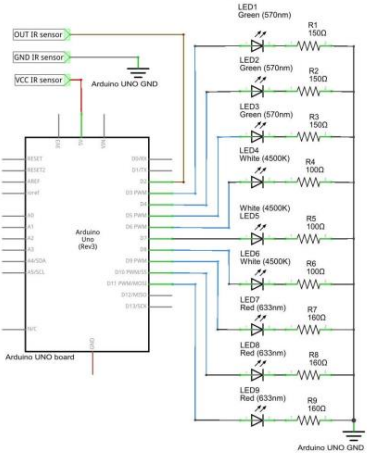



Fig. 6 Demo wiring diagram 3

In this demo we associate an output to each operating state of the IR sensor. The required components are (see also the note at the end of the paragraph!):

- IR sensor FC-51;
- 3 x green LEDs;
- 3 x R = 150Ω;
- 3 x white LEDs;
- 3 x R = 100Ω;
- 3 x red LEDs;
- 3 x R = 160Ω.

Remember that the I / O pins are able to absorb / deliver 40mA max, in total 200mA max (see the datasheet of the **ATmega32**
**).** Let's take a look at the steps required by this demo:

1. We connect the sensor output pin to Arduino digital pin 2. Let's define the pins digital LEDs as an array of pins, from 3 to 11 that we call **LedPIN.**
2. The function **setup {}** it is performed only once before the main loop. In addition to the code of initialization already seen, we declare as output the 9 LEDs through a for loop.
3. The function **loop {}** it is the main function and is cyclically repeated until it is turns off the Arduino board. We save in the variable **detection** the value taken from the pin **IR** through the specific function **digitalRead {}. T**his value can be low if there is an object or high if it is not there.

Let's take a look at the code:

```
# define IR 2                    // digital pin input for ir sensor
int detection; int i;

// digital pin array for:
// green led (3,4,5) - white led (6,7,8) - red led (9,10,11) int LedPIN [] = {3, 4, 5, 6, 7, 8, 9 , 10, 11};


void setup () {
    pinMode (IR, INPUT); for (i = 0; i <9; i ++) {

        pinMode (LedPIN [i], OUTPUT); }}



void loop () {
    detection = digitalRead (IR);
    if (detection == LOW) {// there is an object!
        BlinkLED (); }

    else {
        LedOFF (); }


void BlinkLED () {
    for (i = 0; i <9; i ++) {
        digitalWrite (LedPIN [i], LOW); // turn all the LEDs off}

    delay (150);
    for (i = 0; i <9; i ++) {
        digitalWrite (LedPIN [i], HIGH); // turn all the LEDs on}

    delay (150); }


void LedOFF () {
    for (i = 6; i <9; i ++) {
        digitalWrite (LedPIN [i], LOW); }

    delay (150);
    for (i = 3; i <6; i ++) {
        digitalWrite (LedPIN [i], LOW); }

    delay (150);
    for (i = 0; i <3; i ++) {
```

```
        digitalWrite (LedPIN [i], LOW); }

    delay (150); }
```

**Note:** the code is only one of the possible variants: when an obstacle is present, it flashes the nine LEDs, when there is no obstacle, if they are on it switches them off in sequence, in groups of three. Of course, having fewer LEDs available or wanting to change the code, it is possible to create a circuit that uses only an LED (which flashes if there is an obstacle, is turned off otherwise) or with any number.

**Sitography**

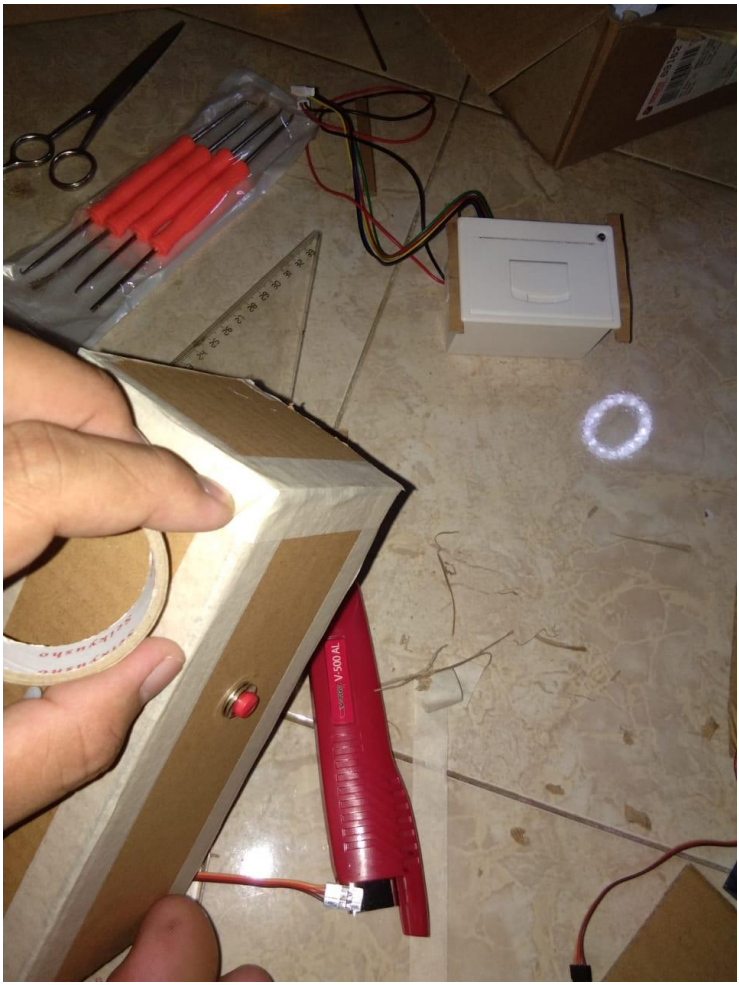

[1] http://www.playembedded.org/blog/it/2016/01/08/rilevare-un-ostacolo-con-il-sensor-ir-e-arduino/

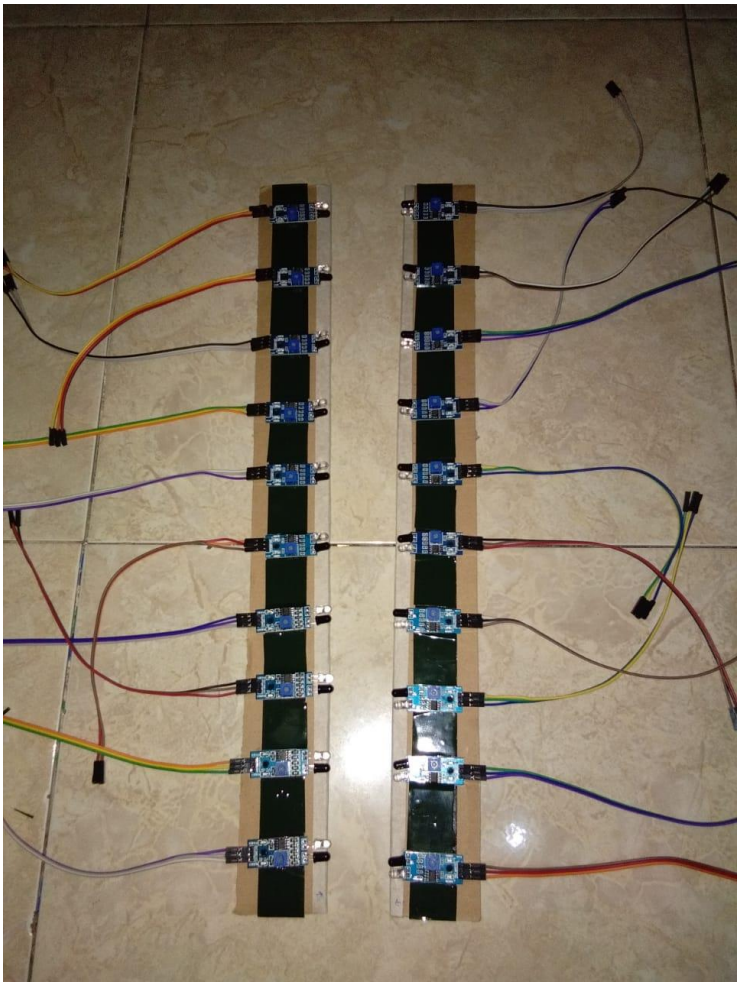
[2] http://qqtrading.com.my/ir-infrared-obstacle-detaction-sensor-module-fc-5

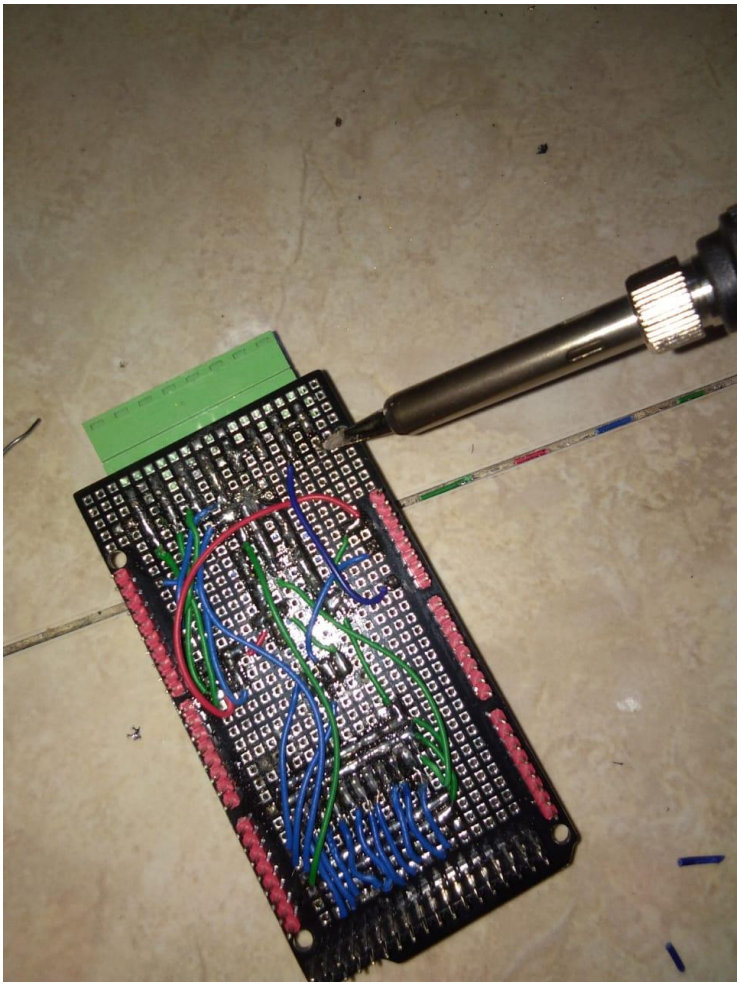
[3] http://www.playembedded.org/blog/it/2016/02/29/un-contatore-di-oggetti-using-un-sensor-ire-arduino/

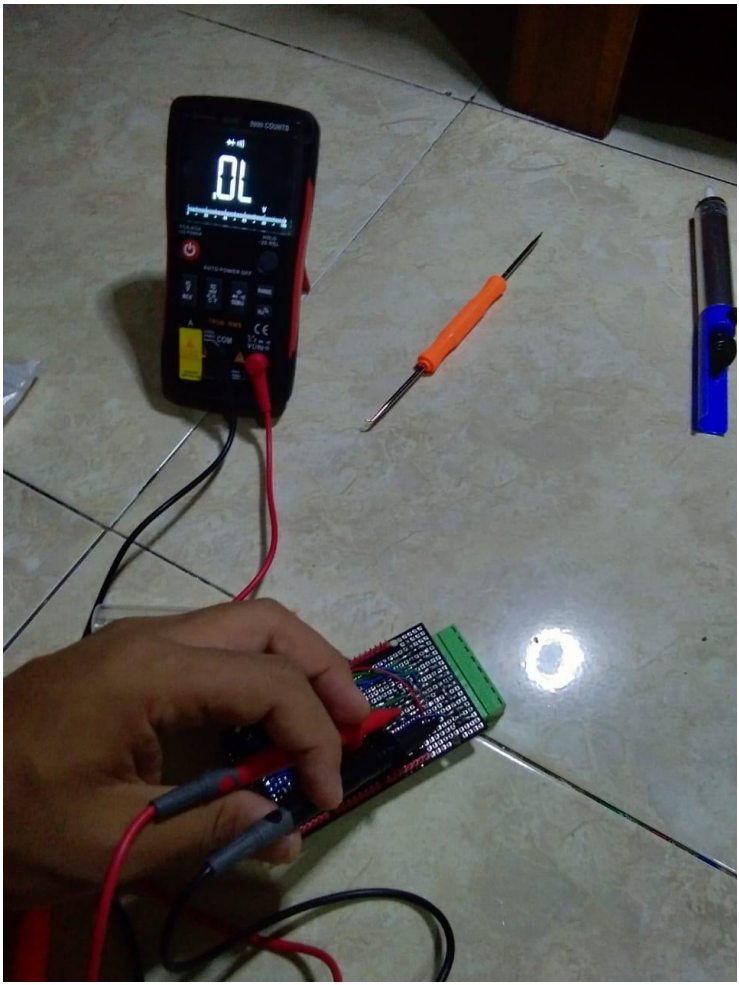
# LAMPIRAN C

# DOKUMENTASI


Cadenas V-100 A
Polipropileno






5999 COUNTS
.OL
AUTO POWER OFF
RANGE
TRUE RMS
COM

# RIWAYAT PENULIS

Nama : Maulana Dwi Ghozali

TTL : Surabaya, 18 Juli 1997

Jenis Kelamin : Laki - Laki

Agama : Islam

Alamat Asal : Jl. Jojoran 5/19 C Surabaya

Telp/HP : 08113067055

E-mail : *maulana.dwi73@gmail.com*

*RIWAYAT PENDIDIKAN*

- *2003 – 2009 : SDN MOJO 3 Surabaya*
- *2009 – 2012 : SMPN 41 Surabaya*
- *2012 – 2015 : SMAN 1 Surabaya*
- *2015 – Sekarang : Departemen Teknik Elektro Otomasi Institut Teknologi Sepuluh Nopember.*

*PENGALAMAN ORGANISASI*

- *OSIS SMPN 41 Surabaya (2011-2012)*
- *SEKRETARIS PB SMASA (2013-2014)*
- *HIMAD3TEKTRO DEPARTEMEN KESMA (2016-2017)*