

ITS
Institut
Teknologi
Sepuluh Nopember

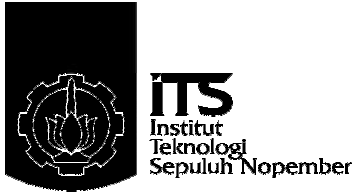
PROYEK AKHIR - VE180626

DATA LOGGING BERBASIS WIRELESS MENGGUNAKAN ESP8266

Wana Abdul Ghoni
NRP 10311400010021

Dosen Pembimbing
Fajar Budiman, ST., M.Sc.
Eko Pujiyatno Matni, S.Pd

PROGRAM STUDI ELEKTRONIKA INDUSTRI
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019



FINAL PROJECT - VE180626

DATA LOGGING BASED ON WIRELESS USING ESP8266

Wana Abdul Ghoni
NRP 10311400010021

Supervisor
Fajar Budiman, ST., M.Sc.
Eko Pujiyatno Matni, S.Pd

INDUSTRIAL ELECTRICITY STUDY PROGRAM
Electrical and Automation Engineering Department
Vocational Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2019

PERNYATAAN KEASLIAN PROYEK AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Proyek Akhir saya dengan judul “**DATA LOGGING BERBASIS WIRELESS MENGGUNAKAN ESP8266**” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

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

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

Surabaya, Januari 2019

Wana Abdul Ghoni
NRP 10311400010021

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

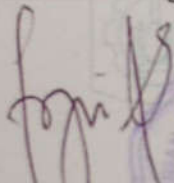
**DATA LOGGING BERBASIS WIRELESS MENGGUNAKAN
ESP8266**

PROYEK AKHIR

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

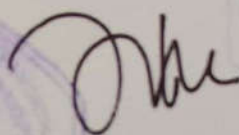
Menyetujui :

Dosen Pembimbing 1



Fajar Budiman, ST., M.Sc.
NIP. 19860707 201404 1 001

Dosen Pembimbing 2



Eko Pujiyatno Matni, S.Pd
NIP. 19710330 199403 1 002

**SURABAYA
JANUARI, 2019**

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

DATA LOGGING BERBASIS WIRELESS MENGGUNAKAN ESP8266

Nama Mahasiswa : Wana Abdul Ghoni
NRP : 10311400010021
Dosen Pembimbing 1 : Fajar Budiman,ST., M.Sc.
NIP : 19860707 201404 1 001
Dosen Pembimbing 2 : Eko Pujiyatno Matni, S.Pd
NIP : 19710330 199403 1 002

ABSTRAK

Pada Industri pekerjaan perhitungan jumlah barang membutuhkan suatu alat bantu agar pekerjaan tersebut lebih efisien dan efektif. Oleh karena itu, kami membuat suatu alat yang dapat membantu dalam proses perhitungan barang produksi. Alat ini dapat membantu meringankan pekerjaan manusia dalam menggunakan tenaga secara terus menerus karena perhitungan barang produksi apalagi di pabrik akan memakan tenaga pekerja secara berkelanjutan. Alat ini menggunakan *Barcode Scanner* sebagai penghitung sekaligus mengidentifikasi barang, sehingga Pencatatan produksi dapat lebih mudah. Barang tersebut akan terscan ketika melewati *barcode scanner*, kemudian data yang didapat akan diolah oleh Arduino Mega, setelah itu data tersebut akan dikirim ke komputer serta ditampilkan di layar komputer secara wireless melalui ESP 8266. Pengujian alat didapatkan kesimpulan bahwa jarak yang ideal untuk pemindaian *barcode* adalah 6 cm. Untuk pemindaian skala industri yang melakukan penghitungan barang dalam jumlah besar, alat ini kurang efisien. Hal ini dikarenakan spesifikasi kecepatan pembacaan *barcode* hanya 100 pembacaan/detik. Maka untuk kedepan bisa disempurnakan dengan tipe *barcode scanner* yang lebih tinggi untuk standar industri.

Kata Kunci : Penghitung Produk, Produk Identifikasi, *Data Logging, Wireless, Arduino, ESP8266*

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

DATA LOGGING BASED ON WIRELESS USING ESP8266

Name of Student : Wana Abdul Ghoni
Number of Registration : 103114 0001 0021
Supervisor 1 : Fajar Budiman, ST., M.Sc.
ID Number : 19860707 201404 1 001
Supervisor 2 : Eko Pujiyatno Matni, S.Pd
ID Number : 19710330 199403 1 002

ABSTRACT

In the industry the work of counting the number of goods requires a system so that the work is more efficient and effective. Therefore, a tool that can assist in the process of calculating production items should be designed. This tool can help to ease the work of humans in using energy continuously because the calculation of production goods especially at the factory will consume workers continuously. This tool uses a Barcode Scanner as a calculator while identifying items, so that recording production can be easier. The item will be detected when passing the barcode scanner, then the data obtained will be processed by Arduino Mega, after that the data will be sent to the computer and displayed on a computer screen wirelessly through ESP 8266. The test of the designed system concluded that the ideal distance for barcode scanning is 6 cm. For industrial-scale scanning of large quantities of goods, this tool is less efficient. This is because the specification of the barcode reading speed is only 100 readings / second. So for the future it can be improved with a higher type scanner method for industry standards.

Keywords : Counter Product, Identification Product, Data Logging, Wireless , Arduino, ESP8266

-----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 Teknik Elektronika Industri, Departemen Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

DATA LOGGING BERBASIS WIRELESS MENGGUNAKAN ESP8266

Dalam Proyek Akhir ini dirancang *Product Data Logging* berbasis *Wireless* dengan menggunakan ESP8266.

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Fajar Budiman, ST., M.Sc., dan Eko Pujiyatno Matni, S.Pd selaku Dosen Pembimbing serta Bapak Eko Setijadi, ST., MT., Ph.D., selaku Kepala Departemen atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Proyek Akhir ini, yang selalu memberikan doa, semangat, dan dukungannya kepada penulis. 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, Januari 2019

Penulis

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

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL.....	i
HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN PROYEK AKHIR.....	v
HALAMAN PENGESAHAN.....	vii
ABSTRAK.....	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Permasalahan.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Metodologi Penelitian.....	2
1.6 Sistematika Laporan.....	4
1.7 Relevansi.....	4
BAB II TEORI DASAR.....	5
2.1. Tinjauan Pustaka.....	5
2.2. Data Logging.....	5
2.3 Motor DC.....	5
2.4. Arduino.....	6
2.4.1 Spesifikasi Arduino Mega 2560.....	7
2.5 Barcode Scanner.....	8
2.5.1 Spesifikasi <i>Barcode Scanner</i> YONGLI XYL-901.....	9
2.6 <i>Barcode ITF-14</i>	10
2.6.1 Struktur <i>Barcode ITF-14</i>	11
2.7 ESP8266.....	12
2.7.1 WEMOS Mini D1.....	12
2.8 LAMP Server.....	13
2.9 Limit Switch.....	15

BAB III PERANCANGAN SISTEM KONTROL	17
3.1 Blok Diagram Skenario Kerja Alat	17
3.2 Perancangan Perangkat Mekanik	18
3.2.1 Perancangan Prototipe Conveyor	18
3.3 Perancangan Perangkat Elektronik	20
3.3.1 Skematik Modul WiFi WEMOS D1 Mini pada Arduino	20
3.3.2 Skematik Detektor Pembanding Berbasis <i>Limit Switch</i>	21
3.3.3 Perancangan Pada Panel Box	22
3.4 Perancangan Perangkat Lunak (Software)	22
3.5 Penyetelan Alamat IP Pada Server dengan WiFi dan Program Pengiriman dan Pembacaan antara Arduino dengan PHP.	24
3.6 Perancangan Web Server	24
BAB IV HASIL SIMULASI DAN IMPLEMENTASI	25
4.1 Pengujian Detektor Pembanding Berbasis Limit Switch	25
4.2 Pengujian Barcode Scanner	27
4.3 Pengujian Kecepatan Pemindaian XYL-901	30
BAB V PENUTUP	33
5.1 Kesimpulan	33
5.2 Saran	33
DAFTAR PUSTAKA	35
LAMPIRAN A (FOTO ALAT).....	A-1
LAMPIRAN B (PROGRAM)	B-1
LAMPIRAN C (DATASHEET).....	C-1
LAMPIRAN D (DAFTAR RIWAYAT HIDUP).....	D-1

DAFTAR GAMBAR

	HALAMAN
Gambar 2.1 Motor DC Sederhana	6
Gambar 2.2 Arduino Mega 2560	7
Gambar 2.3 <i>Barcode Scanner</i> Tipe YONGLI XYL-901	9
Gambar 2.4 Contoh <i>Barcode ITF-14</i>	11
Gambar 2.5 WEMOS D1 Mini	13
Gambar 2.6 Tampilan pertama <i>LAMP Server</i>	14
Gambar 2.7 <i>Limit Switch</i>	15
Gambar 3.1 Blok Diagram Kerja Alat	18
Gambar 3.2 Rancangan <i>Conveyor</i>	19
Gambar 3.3 Rancangan <i>Conveyor</i> dengan ukuran	19
Gambar 3.4 Prototipe <i>Conveyor</i>	20
Gambar 3.5 Skematik Modul WiFi WEMOS D1 Mini pada Arduino ..	21
Gambar 3.6 Skematik Detektor Perbandingan Berbasis <i>Limit Switch</i> ..	21
Gambar 3.7 Perancangan Panel Box	22
Gambar 3.8 <i>FlowChart</i> Program Arduino	23
Gambar 3.9 Penyetelan Alamat IP	24
Gambar 4.1 Perbandingan Detektor Barang dengan <i>Limit Switch</i>	26
Gambar 4.2 Grafik Perbandingan Pemindaian Sesuai Jarak	31

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

DAFTAR TABEL

HALAMAN

Tabel 2.1 Spesifikasi Arduino Mega 2560	8
Tabel 2.2 Spesifikasi <i>Barcode Scanner</i> YONGLI XYL-901	10
Tabel 2.3 Spesifikasi WEMOS D1 Mini	13
Tabel 4.1 Detektor Pembanding Berbasis Limit Switch.....	25
Tabel 4.2 Data pemindaian <i>barcode scanner</i> dari arah depan.....	27
Tabel 4.3 Data pemindaian dari arah kiri ke arah kanan	28
Tabel 4.4 Data pemindaian dari arah kanan ke kiri	28
Tabel 4.5 Data pemindaian dari arah bawah keatas	29
Tabel 4.6 Data pemindaian dari arah atas ke bawah.....	29
Tabel 4.7 Hasil Pengukuran Kecepatan XYL-901 Menscan <i>Barcode</i> Jarak 2cm	30
Tabel 4.8 Hasil Pengukuran Kecepatan XYL-901 Menscan <i>Barcode</i> Jarak 3cm	30
Tabel 4.9 Hasil Pengukuran Kecepatan XYL-901 Menscan <i>Barcode</i> Jarak 4cm	30
Tabel 4.10 Hasil Pengukuran Kecepatan XYL-901 Menscan <i>Barcode</i> dengan Jarak 5 cm	30
Tabel 4.11 Hasil Pengukuran Kecepatan XYL-901 Menscan <i>Barcode</i> dengan Jarak 6 cm.....	31

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

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di dalam dunia industri terdapat suatu proses dimana barang yang telah diproduksi akan dihitung jumlahnya. Proses perhitungan jumlah barang produksi tersebut membutuhkan suatu alat yang dapat membantu proses perhitungan agar hasil yang didapat tidak menimbulkan banyak kesalahan perhitungan dan lebih efektif dalam pengerjaannya. Karena proses perhitungan yang dilakukan secara manual menimbulkan kesalahan perhitungan yang disebabkan oleh *human error* dan data hasil perhitungan jumlah barang produksi tersebut membutuhkan suatu wadah untuk menyimpan dan memproses.[1]

Selain menghitung jumlah barang proses identifikasi barang juga dibutuhkan untuk pengumpulan dan penyimpanan data barang yang dihitung. Untuk pengidentifikasian barang menggunakan *barcode* yang tertera pada bagian luar barang dan penghitung jumlah barang menggunakan *limit switch* yang dipasang pada tepi *conveyor*. Setiap barang yang lewat, *limit switch* akan “on” dan mengirim data ke dalam sistem. Sistem akan menghitung jumlah namun belum mampu menyimpan data barang tersebut, sehingga butuh secara manual oleh manusia untuk memindai data tersebut.

Untuk mengatasi masalah tersebut kami membuat suatu alat yang dapat membantu proses perhitungan jumlah barang serta menampilkan data barang yang dihitung, yang disebut *datalogger*. Kemudian data tersebut diinputkan ke komputer dan akan ditampilkan dilayar komputer. Alat ini diletakkan dibagian luar dari *packaging (outer box)* lalu discan menggunakan *barcode scanner* lalu data hasil perhitungan barang produksi dikirim ke komputer dan ditampilkan di layar komputer. Sehingga dalam proses penghitungan dan pengidentifikasian barang bisa dipantau secara *remote* oleh operator menggunakan perangkat komputer.

1.2 Permasalahan

Permasalahan pada Proyek Akhir ini adalah sebagai berikut :

1. Meminimalisir kesalahan perhitungan seperti *human error* saat perhitungan jumlah barang.
2. Sulitnya melakukan pemantauan jumlah barang dengan cara manual jika barang yang dihitung banyak.
3. Bagaimana membuat sistem *datalogger* barang yang mampu menghitung (*counter*) dan memindai datanya.

1.3 Batasan Masalah

Dari perumusan masalah diatas, maka batasan masalah dari Proyek Akhir ini adalah :

1. Objek yang dihitung berupa barang yang digerakkan oleh Konveyor.
2. *Database* deprogram dengan MySQL dan *Barcode* yang digunakan ITF-14.
3. Data akan ditampilkan dilaptop/PC.

1.4 Tujuan

Tujuan kami dalam membuat Proyek Akhir ini adalah :

1. Mempermudah proses perhitungan jumlah barang produksi.
2. Mempermudah pemantauan data hasil perhitungan jumlah barang.
3. Mengurangi kesalahan perhitungan karena *human error*.

1.5 Metodologi Penelitian

Dalam pelaksanaan Proyek Akir yang berupa *DATA LOGGING BERBASIS WIRELESS MENGGUNAKAN ESP8266*, ada beberapa kegiatan yang dapat diuraikan sebagai berikut :

1. Studi Pustaka dan Survey Data Awal

Materi yang terkait dengan program arduino, *data logging* yang akan diukur dicari dari berbagai sumber literature yang dicari melalui media internet, ruang baca, dan perpustakaan. Materi tersebut membahas tentang *arduino*, ESP8266, production *data logging*, macam teknologi yang dipakai, dan cara analisis pengukuran.

2. Perencanaan Pembuatan Alat

Tahapan ini dilakukan setelah mendapat informasi dari referensi apa yang dibutuhkan pada alat diatas, pada tahapan ini alat akan dirancang dengan target telah siap digunakan tinggal mengkoordinasikan saja sesuai dengan parameter-parameter yang telah ditentukan sesuai referensi. Parameter tersebut antar lain :

1. Rancangan sistem *interface Data Logging* berbasis tampilan *localhost web* menggunakan *LAMP Server*.
2. Desain Alat
3. Bahan yang digunakan
4. Lama waktu pembuatan

3. Perencanaan Pembuatan Software

Agar ahsil rancangan dan pembuatan alat bisa bekerja dengan normal maka perlu algoritma yang sesuai untuk mengkoordinasikan kerja masing-masing komponen dengan menggunakan mikrokontroler jenis *Arduino*.

4. Uji Coba dan Analisa Data

Uji Coba dilakukan di Laboratorium dengan beberapa peralatan penunjang. Setelah mendapat informasi data-data yang diperlukan, penganalisaan data dapat dilakukan. Analisa sisem dapat dilakukan dengan cara membandingkan data-data yang didapat dari pengujian dengan alat yang telah dibuat, sehingga dapat diketahui apakah alat yang telah dibuat memiliki kinerja yang baik atau tidak, serta dapat mengetahui kelebihan serta kekurangan alat tersebut, agar teknologi ini nantinya dapat diterapkan dengan baik di seluruh lapisan masyarakat.

5. Penyusunan Laporan

Penyusunan laporan dilakukan seteelah beberapa data pengukuran yang didapat sudah mencukupi. Sehingga hasil yang didapatkan dari pembuatan Produk *Data Logging* berbasis *Wireless* ini dapat dijelaskan secara detail sesuai dengan data-data yang diperoleh.

1.6 Sistematika Laporan

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

Bab I Pendahuluan

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

Bab II Teori Dasar

Bab ini menjelaskan tentang tinjauan pustaka, konsep dari *Data Logging*, *ESP8266*, *LAMP Server*, model referensi.

Bab III Perancangan dan Desain

Bab ini membahas desain dan perancangan alat berupa *conveyor scanner* berbasis *Barcode Scanner*, modul *logger* yang menggunakan *ESP8266*.

Bab IV Hasil dan Implementasi

Berisi data-data pengujian alat pada hardware dan software secara keseluruhan beserta analisisnya. Data yang di ukur. Pengujian yang di lakukan antara lain pengujian alat, software dan komunikasi.

Bab V Penutup

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

1.7 Relevansi

Hasil yang diperoleh dari Proyek Akhir ini diharapkan menjadi alternatif lain sebagai pengitung barang pada industri yang lebih efisien dan efektif.

BAB II

TEORI DASAR

2.1. Tinjauan Pustaka

Metode yang diusulkan untuk menyelesaikan *Counter Product Logging*. Dengan menggunakan mikrokontroler ATMEGA16 dengan menggunakan komunikasi *Bluetooth*. Hasil yang dicapai terdapat kekurangan yaitu ketika jarak *Bluetooth* sangat terbatas dan data yang dikirimkan membutuhkan waktu yang cukup lama. Oleh karena itu, kita memerlukan adanya inovasi dengan media komunikasi *Wi-Fi*.

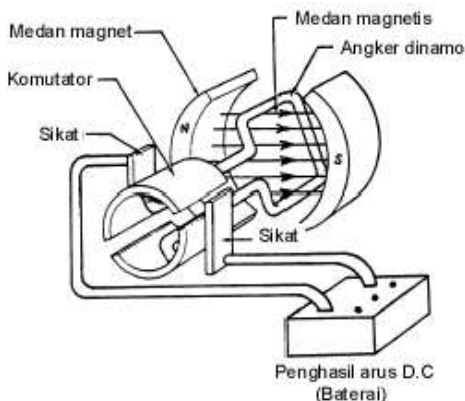
2.2. Data Logging

Proses yang menggunakan komputer untuk mengumpulkan data melalui sensor, menganalisis data, menyimpan keluaran hasil pengumpulan dan analisis. *Data logging* juga berarti kontrol menjelaskan bagaimana komputer mengumpulkan dan menganalisis data. *Data logging* umumnya digunakan dalam eksperimen ilmiah dan sistem pemantauan di mana ada kebutuhan untuk mengumpulkan informasi lebih cepat dari manusia dan mungkin bisa mengumpulkan informasi dan dalam kasus di mana ketepatan dipentingkan. Contoh jenis informasi yang *data logging* sistem dapat mengumpulkan termasuk suhu, suara frekuensi, getaran, kali, intensitas cahaya, arus listrik, tekanan dan perubahan keadaan dari suatu materi.

2.3 Motor DC

Motor listrik merupakan perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Kumparan medan pada motor DC disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Jika terjadi putaran pada kumparan jangkar dalam pada medan magnet, maka akan timbul tegangan (GGL) yang berubah-ubah arah pada setiap setengah putaran, sehingga merupakan tegangan bolak-balik. Prinsip kerja dari arus searah adalah

membalik fasa tegangan dari gelombang yang mempunyai nilai positif dengan menggunakan komutator, dengan demikian arus yang berbalik arah dengan kumparan jangkar yang berputar dalam medan magnet. Bentuk motor paling sederhana memiliki kumparan satu lilitan yang bisa berputar bebas di antara kutub-kutub magnet permanen.



Gambar 2.1 Motor DC Sederhana

Catu tegangan DC dari baterai menuju ke lilitan melalui sikat yang menyentuh komutator, dua segmen yang terhubung dengan dua ujung lilitan. Kumparan satu lilitan pada gambar di atas disebut angker dinamo. Angker dinamo adalah sebutan untuk komponen yang berputar di antara medan magnet.

2.4. Arduino

Arduino adalah pengendali *mikro single board* yang bersifat *open source*, diturunkan dari *wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang hardwarenya memiliki prosesor Atmel AVR dan softwarenya memiliki bahasa pemrograman sendiri. Pada bagian *software* Arduino yang akan digunakan adalah *driver* dan IDE, walaupun masih ada beberapa *software* lain yang sangat berguna selama pengembangan Arduino.

IDE Arduino adalah *software* yang sangat canggih dapat ditulis dengan menggunakan Java. IDE Arduino terdiri dari:

1. **Editor** Program yaitu sebuah *window* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *processing*.
2. **Compiler** yaitu sebuah modul yang mengubah kode program (bahasa *processing*) menjadi kode biner. Bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa *processing* yang bisa dipahami oleh mikrokontroler adalah kode biner itulah sebabnya compiler diperlukan dalam hal ini.
3. **Uploader** yaitu sebuah modul yang membuat kode biner dari komputer ke dalam memori di dalam papan arduino.

2.4.1 Spesifikasi Arduino Mega 2560

Arduino Mega 2560 adalah papan mikrokontroler ATmega2560 berdasarkan (*datasheet*) memiliki 54 digital pin input/output (dimana 15 dapat digunakan sebagai output PWM), 16 analog input, 4 UART (*Hardware port serial*), osilator Kristal 16 MHz, koneksi USB, Jack listrik, *header ICSP*, dan tombol *reset*. [7]. Modul seperti pada Gambar 2.1. Spesifikasi dapat dilihat pada Tabel 2.1.



Gambar 2.2 Arduino Mega 2560

Tabel 2.1 Spesifikasi Arduino Mega 2560

Jenis Spesifikasi	Nilai Spesifikasi
<i>Microcontroller</i>	<i>ATmega 2560</i>
Tegangan Operasi	5 Volt
Tegangan <i>Input</i> (Rekomendasi)	7-12 Volt
Tegangan <i>Input (Limit)</i>	6-20 Volt
<i>Pin Digital I/O</i>	54 (15 dapat sebagai PWM)
<i>Pin Analog Input</i>	16
Arus DC Per <i>Pin I/O</i>	20 mA
Arus DC untuk <i>Pin 3.3 Volt</i>	50 mA
<i>Flash Memory</i>	258 KB (8 KB digunakan oleh <i>bootloader</i>)
<i>SRAM</i>	8 KB
<i>EEPROM</i>	4 KB
<i>Clock Speed</i>	16 MHz
<i>Pin Digital I/O</i>	54 (15 dapat sebagai PWM)
<i>LED_BUILTIN</i>	13
Panjang	101.52 mm
Luas	50 mA
Berat	37 gram

2.5 Barcode Scanner

Barcode scanner adalah alat yang digunakan untuk membaca/memindai kode-kode berbentuk garis-garis vertikal (disebut dengan *BARCODE*) yang terdapat pada kebanyakan produk-produk consumer good. Penggunaan barcode scanner ini mempunyai dua keuntungan tambahan. Yang pertama akan memperkecil kesalahan input yang disebabkan kesalahan operator komputer atau kasir. Yang kedua, penggunaan *barcode scanner* mempercepat proses entry data, sehingga mengurangi jumlah antrian yang panjang.

Seperangkat *barcode scanner* terdiri dari *scanner*, *decoder* dan kabel yang menyambungkan decoder dengan komputer. Barcode scanner tersebut memindai simbol, menangkap dan merubah kode bar menjadi data elektrik lalu mengirimkannya ke komputer dengan format data yang sederhana.

Ada beberapa jenis *barcode scanner* yang umum dipasaran: *Pen Type Reader* atau *Bar Code Wands*; *Laser Barcode Scanner*; *CCD Barcode Scanner*; *Camera Based Barcode Scanner*.

2.5.1 Spesifikasi *Barcode Scanner* YONGLI XYL-901

Gambar 2.2 adalah *Barcode Scanner* Tipe YONGLI XYL-901 yang dipakai dalam pembuatan sistem pada Proyek Akhir ini. *Barcode Scanner* ini memiliki spesifikasi yang terangkum pada tabel 2.2. Dimana kecepatan memindai barcode adalah 100 ± 2 kali perdetik. Dalam sistem ini digunakan *trigger-mode* otomatis karena banyaknya barang yang terpindai oleh *scanner*.



Gambar 2.3 *Barcode Scanner* Tipe YONGLI XYL-901

Tabel 2.2 Spesifikasi *Barcode Scanner* YONGLI XYL-901

Jenis Spesifikasi	Nilai Spesifikasi
Tipe Scanner	<i>Wired Barcode Scanner</i>
Sumber Cahaya	650 nm <i>wavelength</i>
Kinerja Pindai	- Scanning Speed : 100±2 times / second - Width of Scan Field : 30mm at scanning window, 220mm at 200mm - Depth of Scan Field : 0--250mm (0.33mm PCS 90%) - Scanning Angle : Rotation angle of 30 degree; Dip angle of 45 degree; Declination angle of 60 degree - Reading Distance : 2.5 - 600mm
Kemampuan Dekode	EAN-8, EAN-13, UPC-A, UPC-E, Code 39, Code 93, Code 128, EAN128, Codaber, Industrial 2 of 5, Interleave 2 of 5, Matrix 2 of 5, MSI, China Post Code etc
Koneksi	USB 2.0
Dimensi	6.14 in x 3.74 in x 2.8 in (15.6 cm x 9.5 cm x 7.1 cm)
Dukungan Perangkat Lunak	Win xp, Linux, Win 2000, Win 2008, Win vista, Win7 32, Win7 64, Win8 32, Win8 64, Mac OS X
Lainnya	Trigger Mode : Auto, Manual

2.6 *Barcode ITF-14*

ITF-14 (Interleaved Two of Five) adalah implementasi *GSI* dari *Interleaved 2 of 5 barcode* untuk mengkodekan *Item Number* Perdagangan Global. *ITF-14* simbol yang umumnya digunakan pada kemasan tingkat produk, seperti kotak kardus. *ITF-14* akan selalu mengkodekan 14 digit. Alat *GSI GEPiR* dapat digunakan untuk mengetahui identifikasi perusahaan untuk diberikan *GTIN-14* yang dikodekan dalam *ITF-14 Symbol*.

Perbatasan hitam tebal di sekitar simbol disebut Pembawa Bar. Tujuan dari Pembawa Bar adalah untuk menyamakan tekanan yang diberikan oleh pelat cetak atas seluruh permukaan simbol, dan untuk meningkatkan membaca keandalan dengan membantu mengurangi kemungkinan salah membaca atau *scan* pendek yang mungkin terjadi saat pemindai ditahan *barcode* pada terlalu besar sudut. kasus seperti pemindaian miring menyebabkan pemindaian sinar untuk masuk atau keluar simbol *barcode* melalui Pembawa *Bar* di tepi atas atau bawah, memaksa *scanner* untuk mendeteksi scan tidak valid karena *Bar* Pembawa yang jauh lebih luas daripada bar hitam yang sah. Contoh *Barcode ITF-14* ditunjukkan pada Gambar2.5.



Gambar 2.4 Contoh *Barcode ITF-14*

2.6.1 Struktur *Barcode ITF-14*

Struktur dari *Barcode Interleaved two of five* sebagai berikut:

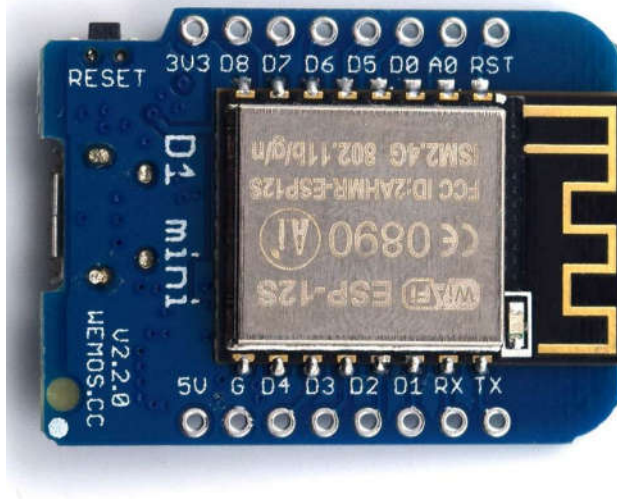
1. *Indicator* - ini menunjukkan tingkat kemasan untuk karton tertentu. awalan satu digit ini dapat berkisar 0-8.
2. *GSI Company Prefix* -Tergantung pada jumlah item perusahaan perlu untuk mengidentifikasi, sebuah *GSI* Perusahaan Awalan mungkin 7 sampai 10 digit.
3. *Item Reference* - Referensi nomor produk yang sama digunakan untuk *GTIN* tingkat item (*GTIN-8*, *GTIN-12*, *GTIN-13*) ketika karton terdiri dari *item* yang sama. Untuk karton yang berisi bermacam-macam barang sejumlah produk baru yang diProyекkan.
4. *Check Digit* - *Digit* terakhir dari *GTIN-14* adalah cek digit dihitung. Menggunakan algoritma *MOD10* cek digit, cek digit dihitung mencegah kesalahan substitusi.

2.7 ESP8266

ESP8266 adalah *low-cost WiFi chip* dengan *full TCP/IP stack* dan mempunyai kemampuan sebagai MCU (*Microcontroller Unit*) yang diproduksi oleh produsen China berbasis Shanghai, Espressif Systems. Chip pertama kali menjadi perhatian para makers di Barat pada Agustus 2014 dengan modul ESP-01. Yang dibuat oleh pihak ketiga, AI-Thinker. Modul mungil ini dapat memungkinkan MCU untuk tersambung pada jaringan *WiFi* dan membuat koneksi simple TCP/IP dengan menggunakan perintah *Hayes-style*. Namun pada saat itu hanya ada dokumentasi *non-English* untuk chip tersebut. Harga yang murah, volume penyimpanan besar dan hanya butuh beberapa komponen tambahan untuk menggunakannya, menjadi daya tarik untuk para penggemar dan penggiat elektronika dan mulai menerjemahkan dokumentasi ke bahasa Inggris.

2.7.1 WEMOS Mini D1

WEMOS D1 Mini adalah perangkat mikrokontroler yang didasarkan dari ESP8266, banyak komunitas elektronik yang menyebutnya "*little Arduino with Wi-Fi*" atau Arduino kecil dengan Wi-Fi. Dengan itu komunikasi Arduino dengan perangkat atau sensor-sensor lainnya dapat lebih mudah dan efisien. Gambar 2.4 adalah WEMOS D1 Mini ESP-12S yang dipakai pada Proyek akhir ini, dengan tegangan operasi 3.3 volt. Sehingga untuk menyambungkan pin antara Arduino MEGA 2560 dengan WEMOS D1 Mini menggunakan *Logic-Level Shifter* berfungsi sebagai pengubah tegangan dari tegangan operasi 5 Volt ke 3.3 Volt dan sebaliknya. Agar data yang dikirimkan dan diterima bisa dibaca oleh Arduino MEGA maupun WEMOS D1 Mini. Dimana spesifikasi WEMOS D1 Mini ini terangkum pada Tabel 2.3.



Gambar 2.5 WEMOS D1 Mini

Tabel 2.3 Spesifikasi WEMOS D1 Mini

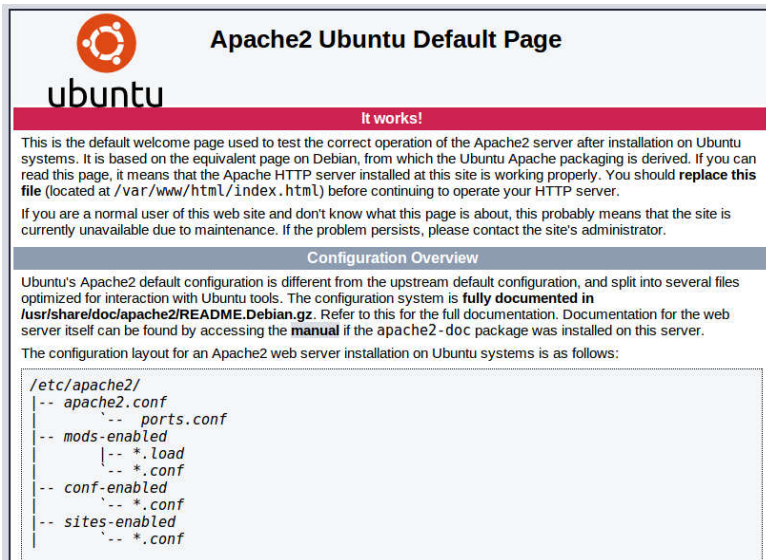
Jenis Spesifikasi	Nilai Spesifikasi
Tegangan Operasi	3.3 Volt
Pin Digital I/O	11
Pin Analog Input	1 (Max input: 3.2V)
Clock Speed	80MHz/160MHz
Flash	4M bytes
Panjang	34.2 mm
Lebar	25.6 mm
Berat	10 gram

2.8 LAMP Server

LAMP adalah istilah yang merupakan singkatan dari *Linux*, *Apache*, *MySQL* dan *PHP* (atau menggunakan *Perl* dan *Python*). Merupakan sebuah paket perangkat lunak bebas yang digunakan untuk menjalankan sebuah aplikasi secara lengkap. *Linux* sebagai Sistem

Operasi yang digunakan. *Apache* sebagai *Web Server*. Data disimpan pada *MySQL* dan diproses menggunakan bahasa pemrograman *PHP*. [2]

Linux adalah sistem operasi komputer mirip *UNIX* yang dirangkai dalam model pengembangan dan distribusi perangkat lunak bebas dan sumber terbuka (*open source*). Peran server web LAMP telah disediakan yang biasanya memakai *Apache*. *Apache HTTP Server* telah menjadi server web paling populer di Internet publik. Untuk penyimpanan *database* ini memakai *MySQL* sebagai sistem manajemen basis data relasional (RDBMS) dan untuk bahasa pemrograman aplikasi LAMP ini menggunakan *PHP* atau bisa juga bahasa lain seperti Perl dan Python, *PHP* adalah bahasa skrip sisi server (*server-side scripting*) yang dirancang untuk pengembangan web tetapi juga digunakan sebagai bahasa pemrograman tujuan umum (*general-purpose programming language*).



Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
    |-- ports.conf
-- mods-enabled
    |-- *.load
    |-- *.conf
-- conf-enabled
    |-- *.conf
-- sites-enabled
    |-- *.conf
```

Gambar 2.6 Tampilan pertama *LAMP Server*.

2.9 Limit Switch

Limit switch merupakan jenis saklar yang dilengkapi dengan katup yang berfungsi menggantikan tombol. Prinsip kerja limit switch sama seperti saklar Push ON yaitu hanya akan menghubungkan pada saat katupnya ditekan pada batas penekanan tertentu yang telah ditentukan dan akan memutuskan saat katup tidak ditekan. Limit switch termasuk dalam kategori sensor mekanis yaitu sensor yang akan memberikan perubahan elektrik saat terjadi perubahan mekanik pada sensor tersebut. Penerapan dari limit switch adalah sebagai sensor posisi suatu benda (objek) yang bergerak.

Limit switch umumnya digunakan untuk: Memutuskan dan menghubungkan rangkaian menggunakan objek atau benda lain; Menghidupkan daya yang besar, dengan sarana yang kecil; Sebagai sensor posisi atau kondisi suatu objek.

Prinsip kerja limit switch diaktifkan dengan penekanan pada tombolnya pada batas/daerah yang telah ditentukan sebelumnya sehingga terjadi pemutusan atau penghubungan rangkaian dari rangkaian tersebut. *Limit switch* memiliki 2 kontak yaitu NO (*Normally Open*) dan kontak NC (*Normally Close*) dimana salah satu kontak akan aktif jika tombolnya tertekan.



Gambar 2.7 *Limit Switch*

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

BAB III

PERANCANGAN DAN IMPLEMENTASI

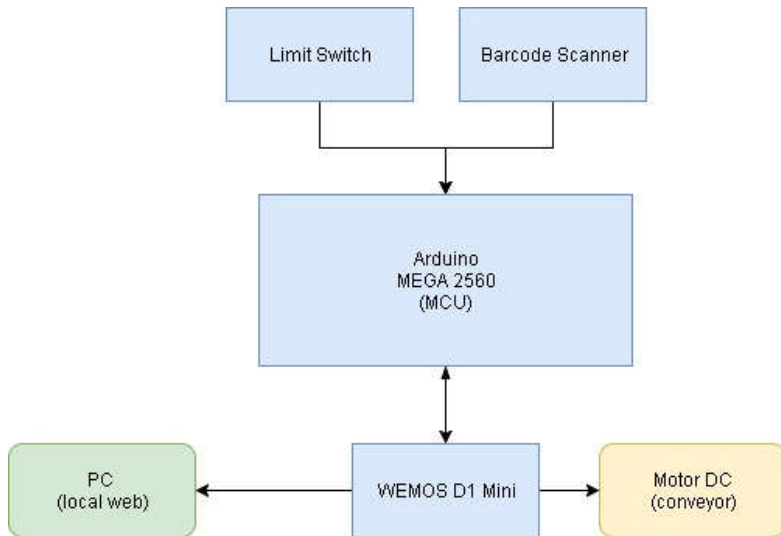
Dalam bab ini dibahas perancangan alat yang meliputi perencanaan perangkat keras (*hardware*) dan perangkat lunak (*software*). Hal tersebut guna mewujudkan Proyek Akhir yang berjudul “Perancangan dan Realisasi *DATA LOGGING BERBASIS WIRELESS MENGGUNAKAN ESP8266*”. Perancangan alat dibahas perbagian disertai dengan gambar skematik. Sedangkan penjelasan *software* akan dijelaskan mengenai pembuatan program *server* dengan menggunakan *database* pada *website*, pembuatan program *client* dengan menggunakan program *Arduino IDE* dan *Arduino* sebagai aplikasi pemrograman Mikrokontroler agar nantinya antara sistem pembacaan dan komputer dapat saling tersambung dengan baik.

Untuk memudahkan dalam pembahasan bab ini akan dibagi menjadi dua yaitu:

- a) Perancangan *hardware* (perangkat keras) yang terdiri dari pembuatan rangkaian *detector limitswitch*, perancangan *hardware* kontroler, serta perancangan *hardware plant* yang terdiri dari prototipe *conveyor*
- b) Perancangan *software* (perangkat lunak) meliputi perancangan pembuatan program *server localhost* dengan menggunakan *Apache* dan *MySQL* serta *PHP* pada *LAMP Server*

3.1 Blok Diagram Skenario Kerja Alat

Sistem ini bekerja dengan menggunakan *Arduino MEGA 2560* sebagai *MCU*, dengan *Barcode Scanner* sebagai pengidentifikasi, limit switch sebagai pembanding jumlah barang yang terdeteksi oleh *Barcode Scanner*. Kemudian *Arduino* sebagai mikrokontroler mengirim data yang terbaca oleh *Barcode Scanner* dan *limitswitch* menuju *WEMOS D1 Mini* yang bertindak sebagai modul *WiFi*. Dan tahap akhir *WEMOS D1 Mini* mengirim data ke *PC/Laptop* dengan nirkabel. Adapun desain skenario dapat dilihat pada Gambar 3.1.



Gambar 3.1 Blok Diagram Kerja Alat

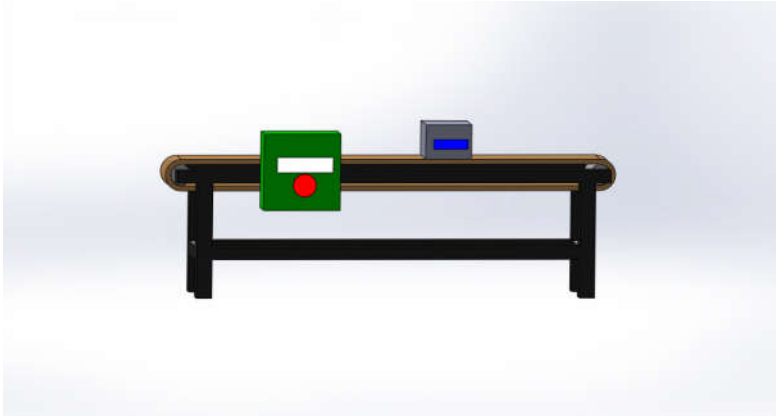
3.2 Perancangan Perangkat Mekanik

Pada keseluruhan sistem pada Gambar 3.1 dapat dilihat bahwa *conveyor* (digerakkan motor DC) terpisah sendiri, maksudnya *conveyor* membutuhkan sumber tegangan sendiri diluar sistem diatas sehingga tidak dapat dikendalikan melalui sistem. Karena *conveyor* akan berjalan terus dalam sistem ini.

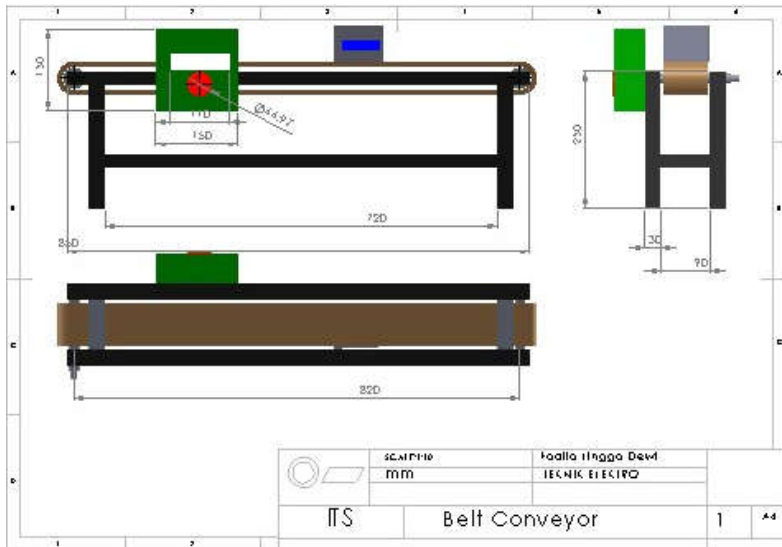
3.2.1 Perancangan Prototipe Conveyor

Conveyor adalah suatu sistem mekanik yang mempunyai fungsi memindahkan barang dari satu tempat ke tempat yang lain. Conveyor banyak dipakai di industri untuk transportasi barang yang jumlahnya sangat banyak dan berkelanjutan. Conveyor memiliki banyak jenis seperti *roller conveyor* dan *belt conveyor*. Untuk Proyek Akhir ini menggunakan Belt Konveyor karena belt conveyor dapat digunakan untuk mengangkut material baik yang berupa “*unit load*” atau “*bulk material*” baik secara mendatar ataupun miring. dimaksud dengan *unit load* adalah benda yang biasanya dapat dihitung jumlahnya satu per

satu, misalnya kotak, kantong, balok dan lain sebagainya. Sedangkan *Bulk Material* adalah material yang berupa butir-butir, bubuk atau serbuk, misalnya pasir, semen dan lain sebagainya.



Gambar 3.2 Rancangan *Conveyor*



Gambar 3.3 Rancangan *Conveyor* dengan ukuran

Realisasi prototype conveyor dengan menggunakan bahan dasar akrilik dengan tebal 8mm guna mengurangi kerusakan pada saat dilakukannya pengambilan data



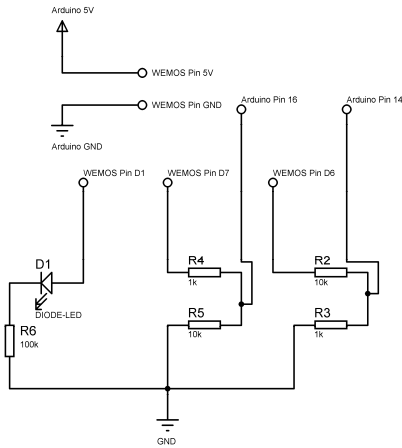
Gambar 3.4 Prototipe Conveyor

3.3 Perancangan Perangkat Elektronik

Perancangan perangkat elektronik ini terdiri dari alat yang membutuhkan sumber tegangan untuk mengaktifkannya diantaranya adalah:

3.3.1 Skematik Modul WiFi WEMOS D1 Mini pada Arduino

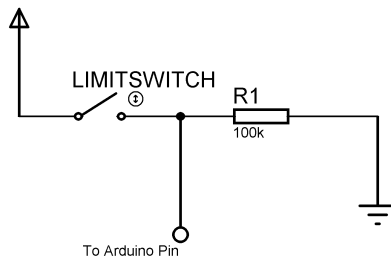
Pada rangkaian skematik ini digunakan *Logic-Level Shifter*, yang digunakan untuk menurunkan tegangan output pin Arduino yang bernilai 5 Volt menjadi 3.3 Volt. Karena ESP8266 mempunyai tegangan operasi (*Operation Voltage*) bernilai 3.3 Volt, apabila tidak diturunkan maka tegangan 5 Volt yang berasal dari output pin Arduino akan merusak papan sirkuit dan elemen-elemen yang ada pada modul WiFi. Serta pengiriman data dari barcode scanner melalui Arduino MEGA menuju WEMOS D1 Mini tidak akan bekerja dengan baik, karena akan menampilkan data yang tak beraturan.



Gambar 3.5 Skematik Modul WiFi WEMOS D1 Mini pada Arduino

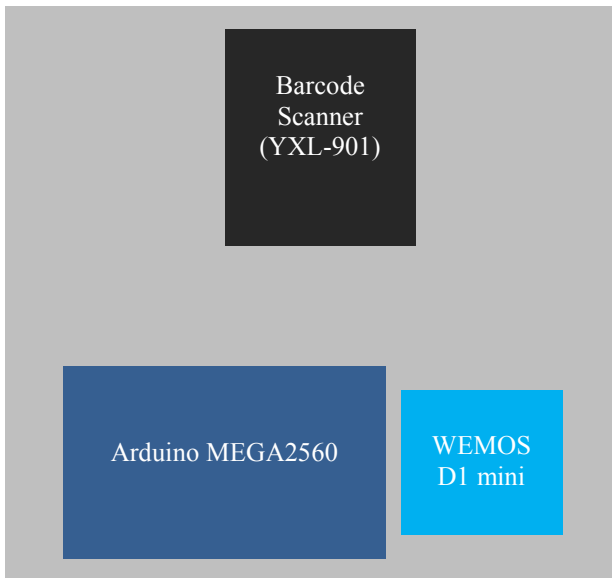
3.3.2 Skematik Detektor Pembanding Berbasis *Limit Switch*

Detektor Pembanding berbasis *Limit Switch* digunakan untuk menjadi pembanding pada pemindaian berapa jumlah barang yang sudah dipindai/dibaca oleh *Barcode Scanner*. Hasil penjumlahan total Detektor Pembanding dibandingkan apa sudah sama jumlah barang yang telah dipindah atau ada selisih antara jumlah barang hasil pemindaian oleh *Barcode Scanner* dan Detektor Pembanding berbasis *Limit Switch*.



Gambar 3.6 Skematik Detektor Pembanding Berbasis *Limit Switch*

3.3.3 Perancangan Pada Panel Box



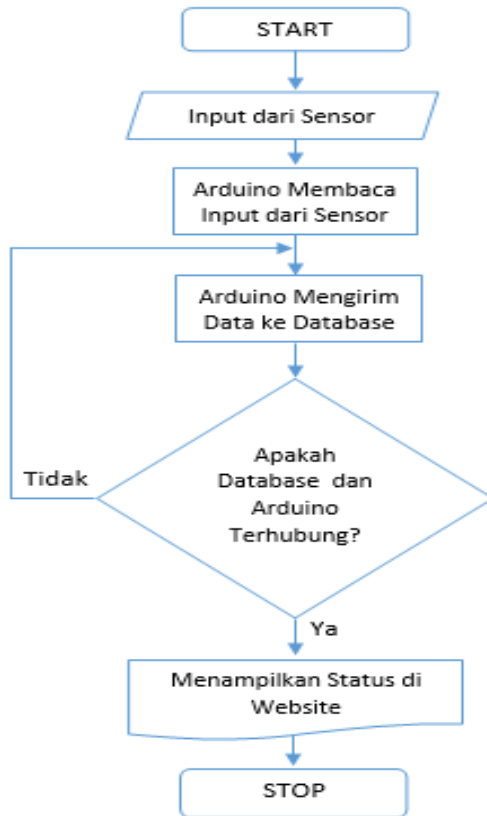
Gambar 3.7 Perancangan Panel Box

Pada Perancangan Panel Box, seluruh komponen elektronik ditempatkan didalam Panel Box, agar seluruh komponen terlindungi dari gangguan dari lingkungan luar. Gambar 3.7 memperlihatkan skema penempatan pada panel box sistem yang dibuat pada Proyek Akhir ini. Penempatan didesain agar dibuat *robust*.

3.4 Perancangan Perangkat Lunak (Software)

Gambar 3.8 menunjukkan Flowchart pada sistem perangkat lunak yang didesain pada sistem datalogger ini. Ketika barang yang digerakkan oleh *conveyor* akan dideteksi oleh *limitswitch* untuk dihitung dan discan langsung secara otomatis oleh *barcode scanner* sebagai input sensor. Arduino Mega akan membaca input sensor ini dan mengirim ke database di website yang bisa dilihat melalui browser laptop/handphone. Jika database belum terhubung maka akan kembali ke status pembacaan

sensor. Jika terhubung maka akan mengirim ke database di website untuk ditampilkan.



Gambar 3.8 *FlowChart* Program Arduino

3.5 Penyetelan Alamat IP Pada Server dengan WiFi dan Program Pengiriman dan Pembacaan antara Arduino dengan PHP.

Gambar 3.9 menunjukkan proses setting alamat IP pada ESP 8266 untuk diakses oleh komputer lain secara lokal.

```
const char* ssid = "WHYSOSERIOUS?"; //SSID Name
const char* pass = "jokervsbatman"; //password for SSID
const char host[] = "192.168.43.49"; //ip address localhost
```

Gambar 3.9 Penyetelan Alamat IP

3.6 Perancangan Web Server

Perancangan *web server* digunakan sebagai media *interface* dengan alamat IP 192.168.43.49. *Web server* tersebut berbasis *database* MySQL. Berikut ini merupakan tampilan utama *database* pada halaman phpMyAdmin pada Gambar 3.

Perancangan *web server* secara *offline* dilakukan dengan menggunakan aplikasi LAMPP. XAMPP merupakan suatu paket server yang tergolong sangat lengkap. XAMPP yang merupakan singkatan dari *Linux* (L), *Apache* (A), *MySQL* (M), dan *PHP* (P) dibuat dengan tujuan memudahkan bari para pengembang web server local untuk tujuan pengujian. LAMP menyediakan semua yang dibutuhkan dalam pengujian web. Karena sebagian besar penyebaran web server yang sebenarnya (secara online) menggunakan komponen yang sama seperti LAMP, sehingga membuat transisi dari server local ke *server online* sangat mudah.

BAB IV

HASIL PENGUJIAN ALAT

Bab ini dibagi menjadi tiga bagian. Pada bagian pertama dijelaskan mengenai hasil pengujian pada perangkat keras dan pada bagian kedua akan dijelaskan mengenai hasil pengujian pada media komunikasi yang merupakan tampilan pada *smartphone* dan yang terakhir adalah pengujian keseluruhan dari cara kerja alat. pada masing-masing bagian akan ditunjukkan implementasi dan mekanisme dari pengambilan data, untuk gambar hasil pengujian alat terlampir.

4.1 Pengujian Detektor Pembanding Berbasis Limit Switch

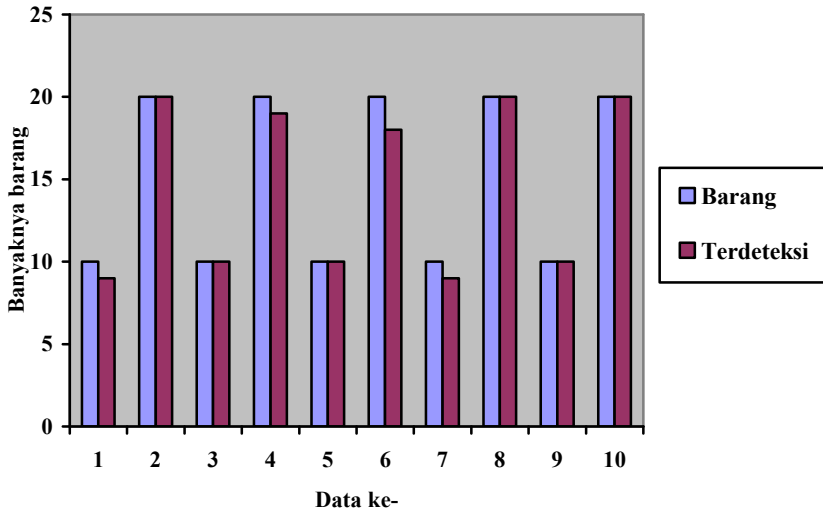
Dalam pengambilan data Detektor diambil dengan menempatkan barang yang akan dideteksi oleh limit switch. Dengan pengambilan data ini dapat diambil kesimpulan berapa banyak selisih barang yang terdeteksi dengan jumlah barang yang melewati konveyor.

Tabel 4.1 Detektor Pembanding Berbasis Limit Switch

No	Waktu (detik)	Banyaknya Barang	Banyaknya yang terdeteksi
1	10	10	9
2	20	20	20
3	10	10	10
4	20	20	19
5	10	10	10
6	20	20	18
7	10	10	9
8	20	20	20
9	10	10	10
10	20	20	20
11	10	10	9
12	20	20	19

Setelah dilakukan pengujian dan pengambilan data serta hasil seperti table diatas maka jarak 6 cm bisa dijadikan acuan dalam pengambilan data yang baik dan ideal karena pada beberapa jarak selain 6cm terjadi kesalahan pendeteksiian dengan jumlah selisih antara 1 sampai 2 barang.

Dapat dilihat pada Gambar 4.1 bahwa terdapat selisih jumlah barang yang lewat dengan jumlah barang terdeteksi oleh Detektor *limit switch* dikarenakan kurang sensitifnya *limit switch* yang dipergunakan dan juga posisi barang yang melewati *conveyor* serta massa barang tersebut.



Gambar 4.1 Perbandingan Detektor Barang dengan *Limit Switch*

4.2 Pengujian Barcode Scanner

Untuk pengujian Barcode Scanner diambil data dengan cara melakukan pemindaian dari arah depan, kiri, kanan, atas dan bawah dengan jarak dalam satuan cm dengan kecepatan motor DC pada conveyor ± 150 RPM. Dengan tujuan menguji spesifikasi pemindaian *barcode scanner* bisa digunakan dalam berapa jarak maksimal dan minimalnya. Terbaca atau tidaknya bisa diambil dengan melihat *barcode scanner* berbunyi *beep* dan masuknya data kedalam Arduino.

Tabel 4.2 Data pemindaian *barcode scanner* dari arah depan.

No	Jarak (cm)	Keterangan
1	0	Tidak Terbaca
2	1	Terbaca
3	2	Terbaca
4	3	Terbaca
5	4	Terbaca
6	5	Terbaca
7	6	Terbaca
8	7	Terbaca
9	8	Terbaca
10	9	Tidak Terbaca

Tabel 4.3 Data pemindaian dari arah kiri ke arah kanan

No	Jarak (cm)	Keterangan
1	0	Tidak Terbaca
2	1	Tidak Terbaca
3	2	Terbaca
4	3	Terbaca
5	4	Terbaca
6	5	Terbaca
7	6	Terbaca
8	7	Terbaca
9	8	Tidak Terbaca
10	9	Tidak Terbaca

Tabel 4.4 Data pemindaian dari arah kanan ke kiri

No	Jarak (cm)	Keterangan
1	0	Tidak Terbaca
2	1	Tidak Terbaca
3	2	Terbaca
4	3	Terbaca
5	4	Terbaca
6	5	Terbaca
7	6	Terbaca
8	7	Terbaca
9	8	Tidak Terbaca
10	9	Tidak Terbaca

Tabel 4.5 Data pemindaian dari arah bawah keatas

No	Jarak (cm)	Keterangan
1	0	Tidak Terbaca
2	1	Tidak Terbaca
3	2	Terbaca
4	3	Terbaca
5	4	Terbaca
6	5	Terbaca
7	6	Terbaca
8	7	Terbaca
9	8	Tidak Terbaca
10	9	Tidak Terbaca

Tabel 4.6 Data pemindaian dari arah atas ke bawah

No	Jarak (cm)	Keterangan
1	0	Tidak Terbaca
2	1	Tidak Terbaca
3	2	Terbaca
4	3	Terbaca
5	4	Terbaca
6	5	Terbaca
7	6	Terbaca
8	7	Terbaca
9	8	Tidak Terbaca
10	9	Tidak Terbaca

Dapat dilihat pada hasil pengukuran pada bel 4.2, Tabel 4.3, Tabel 4.4, Tabel 4.5, Tabel 4.6 bisa diambil kesimpulan antara jarak 2 – 7 cm dapat terbaca dengan baik sehingga bisa menjadi acuan dalam penempatan barang terhadap *barcode scanner*.

4.3 Pengujian Kecepatan Pemindaian XYL-901

Pada pengujian Kecepatan XYL-901 dalam menscan kali ini dilakukan pengukuran dengan menggunakan bantuan timer untuk mengetahui jarak antar *barcode* yang telah terscan.

Tabel 4.7 Hasil Pengukuran Kecepatan XYL-901 Menscan Barcode Jarak 2cm

No	Jarak (cm)	Waktu (detik)
1	2	00.9
2	2	01.0
3	2	01.4
4	2	01.9

Tabel 4.8 Hasil Pengukuran Kecepatan XYL-901 Menscan Barcode Jarak 3cm

No	Jarak (cm)	Waktu (detik)
1	3	01.0
2	3	01.0
3	3	01.4
4	3	01.9

Tabel 4.9 Hasil Pengukuran Kecepatan XYL-901 Menscan Barcode Jarak 4cm

No	Jarak (cm)	Waktu (detik)
1	4	01.2
2	4	01.3
3	4	01.2
4	4	01.0

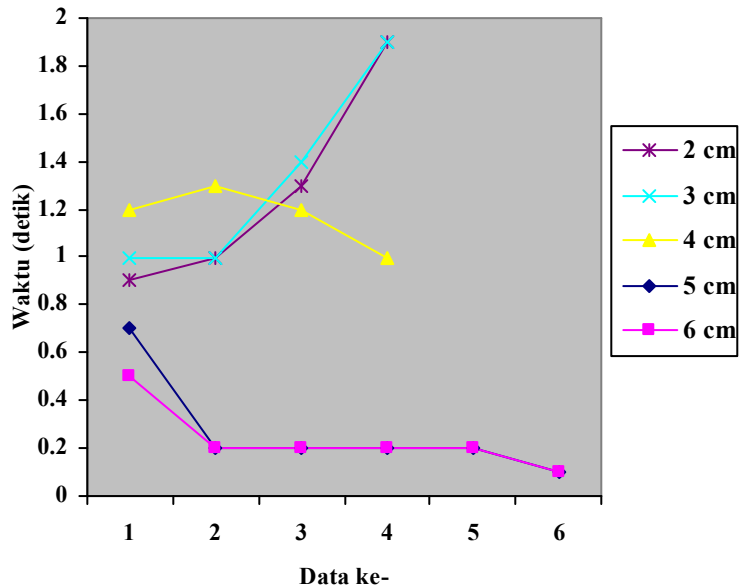
Tabel 4.10 Hasil Pengukuran Kecepatan XYL-901 Menscan Barcode dengan Jarak 5 cm

No	Jarak (cm)	Waktu (detik)
1	5	00.7
2	5	00.2
3	5	00.2
4	5	00.2
5	5	00.2
6	5	00.1

Tabel 4.11 Hasil Pengukuran Kecepatan XYL-901 Menscan *Barcode* dengan Jarak 6 cm

No	Jarak (cm)	Waktu (detik)
1	6	00.5
2	6	00.2
3	6	00.2
4	6	00.2
5	6	00.2
6	6	00.1

Dapat dilihat pada hasil pengukuran pada Tabel 4.7, Tabel 4.8, Tabel 4.9, Tabel 4.10 dan Tabel 4.11 bahwa setelah dilakukan pengambilan data didapatkan hasil kecepatan XYL-901 dalam memindai *barcode* yang baik adalah dengan jarak 6 cm. Dan Gambar 4.1 menunjukkan pemindaian barcode dengan waktu tercepat ialah pada jarak 6 cm



Gambar 4.2 Grafik Perbandingan Pemindaian Sesuai Jarak

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

BAB V

PENUTUP

Setelah melakukan perancangan dan pembuatan alat serta pengujian dan analisa, maka dapat ditarik kesimpulan dan saran dari kegiatan yang telah dilakukan untuk pengembangan Proyek Akhir ini.

5.1 Kesimpulan

Dari seluruh tahapan yang sudah dilaksanakan pada penyusunan Proyek Akhir ini, mulai dari studi *literature*, perancangan dan pembuatan sampai pada pengujiannya maka dapat disimpulkan bahwa:

1. Kesalahan perhitungan barang sudah dapat ditekan, namun masih tetap adanya selisih perhitungan barang dan detektor pembandingan saat alat bekerja.
2. Dengan pengiriman data menggunakan ESP8266 menuju ke komputer, pemantauan jumlah barang dapat teratasi sekaligus indentifikasi barang tersebut.
3. Pemakaian Limit Switch kurang sesuai yang diharapkan karena ada beberapa data yang terdapat selisih dalam pendeteksian barang dan pembandingan dengan hasil barcode scanner.

5.2 Saran

Untuk memperbaiki dan menyempurnakan kinerja alat ini, maka perlu disarankan:

1. Untuk XYL-901 terlebih dahulu lakukan penyetelan untuk membuat *continuous/auto power on* agar flash dapat hidup secara terus menerus tanpa menekan tombol setiap akan menscan (lihat di XYL-901 *Configuration*).
2. Pengaturan jarak antara barcode scanner dan barcode di barang sangatlah penting dalam menentukan kecepatan menscan.

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

DAFTAR PUSTAKA

- [1] Syarif Hidayatullah, “*Rancang Bangun Alat Penghitung Jumlah Produksi Semen Berbasis Data Logger Menggunakan Sensor Infrared*”, **Proyek Akhir**, Jurusan Teknik Elektro FTI-ITS, Surabaya, 2010
- [2] James B. Lee, Brent Ware,. **Open Source Web Development with LAMP: Using Linux, Apache, MySQL, Perl, and PHP**, Addison Wesley, 2010.
- [3], **Datasheet USB HOST SHIELD**, +5V-Powered Multichannel RS232 Drivers/Receivers, MAXIM, 2010.
- [4], **Datasheet XYL-901**, Specification of XYL-901, Champtek, 2010.
- [5] Kadir,Abdul.**Buku Pintar Pemograman Arduino**, Penerbit MediaKom, Yogyakarta, 2015.
- [6] Sewiawardhana,Sigit Wasista dan Delima Ayu Saraswati,**Belajar Arduino**, Penerbit Bumi Aksara, Jakarta, 2016.
- [7], **Datasheet ATmega 2560**, Microcontroller , Atmel, 2010.

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

LAMPIRAN A FOTO

A.Sistem Alat Keseluruhan



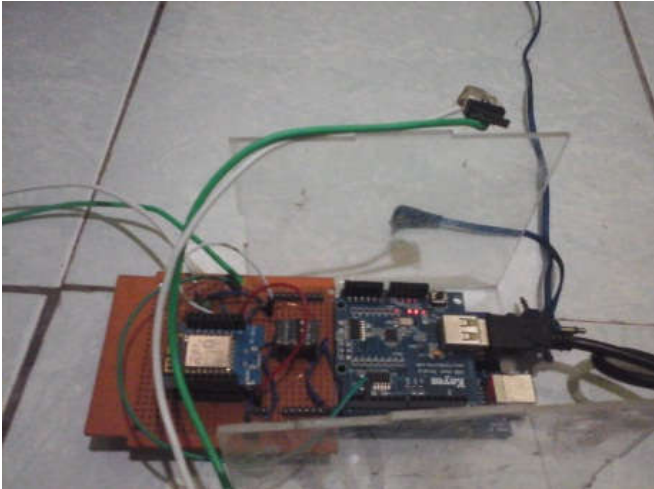
A.1 Saat Menscan Barcode



A.2 Module Scanner Tampak Atas



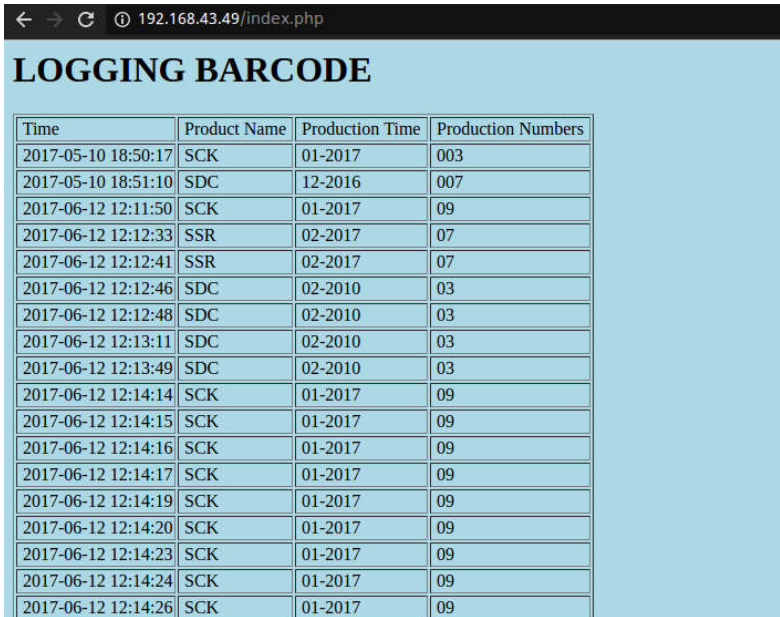
A.3 Mdule Scanner



B.Contoh Barang



C. Tampilan di Web Browser



The screenshot shows a web browser interface with a dark header bar containing navigation icons and the URL "192.168.43.49/index.php". Below the header, the page title "LOGGING BARCODE" is displayed in large, bold, black letters. The main content area features a table with four columns: "Time", "Product Name", "Production Time", and "Production Numbers". The table contains 20 rows of data, listing various production events with their corresponding timestamps, product names (SCK, SDC, SSR), production years, and numbers.

Time	Product Name	Production Time	Production Numbers
2017-05-10 18:50:17	SCK	01-2017	003
2017-05-10 18:51:10	SDC	12-2016	007
2017-06-12 12:11:50	SCK	01-2017	09
2017-06-12 12:12:33	SSR	02-2017	07
2017-06-12 12:12:41	SSR	02-2017	07
2017-06-12 12:12:46	SDC	02-2010	03
2017-06-12 12:12:48	SDC	02-2010	03
2017-06-12 12:13:11	SDC	02-2010	03
2017-06-12 12:13:49	SDC	02-2010	03
2017-06-12 12:14:14	SCK	01-2017	09
2017-06-12 12:14:15	SCK	01-2017	09
2017-06-12 12:14:16	SCK	01-2017	09
2017-06-12 12:14:17	SCK	01-2017	09
2017-06-12 12:14:19	SCK	01-2017	09
2017-06-12 12:14:20	SCK	01-2017	09
2017-06-12 12:14:23	SCK	01-2017	09
2017-06-12 12:14:24	SCK	01-2017	09
2017-06-12 12:14:26	SCK	01-2017	09

127.0.0.1 / localhost / x

127.0.0.1/phpmyadmin/index.php?token=edb104497eb76f1ea86b90e451d4e98#PMAURL-1:db_struct

Wana Abdul

Database: esp8266

Structure SQL Search Query Export Import Operations Privileges More

Table	Action	Rows	Type	Collation	Size	Overhead
Data Table	Browse Structure Search Empty Drop Insert	-9	InnoDB	latin1_swedish_ci	16 K	16 K
LOGGING BARCODE	Browse Structure Search Empty Drop Insert	-27	InnoDB	latin1_swedish_ci	16 K	16 K
LOGGING DETECTOR	Browse Structure Search Empty Drop Insert	-18	InnoDB	latin1_swedish_ci	16 K	16 K
3 tables	Sum	37	InnoDB	latin1_swedish_ci	48	48

Check All With selected:

Print view Data Dictionary

Create table

Name: Number of columns:

Go

(Recent tables) ...

- esp8266
 - New
 - Data Table
 - LOGGING BARCODE
 - LOGGING DETECTOR
 - information_schema
 - mysql
 - performance_schema
 - phpmyadmin

127.0.0.1 / localhost / ... READ DATA ... READ DATA ... Wana Abdul

phpMyAdmin

(Recent tables) ...

- esp8266
 - New
 - DataTable
 - LOGGING BARCODE
 - LOGGING DETECTOR
 - information_schema
 - mysql
 - performance_schema
 - phpmyadmin

Server: localhost Database: esp8266 Table: LOGGING BARCODE

Structure SQL Search Insert Export Import Operations More

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	No	int(20)			No	None	
2	Time	timestamp		on update CURRENT_TIMESTAMP	No	CURRENT_TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP
3	Data	varchar(11)	latin1_swedish_ci		No	None	
4	Product Name	varchar(20)	latin1_swedish_ci		No	None	
5	Production Time	varchar(20)	latin1_swedish_ci		No	None	
6	Production Numbers	varchar(20)	latin1_swedish_ci		No	None	

Check All
 With selected:
 Browse
 Change
 Drop
 Primary
 Unique
 Index

Print view
 Relation view
 Propose table structure
 Track table
 Move columns

Add
 1
 column(s)
 At End of Table
 At Beginning of Table
 After
 No
 Go

+ Indexes

Information

Space usage	Row statistics
Data	Format
16 KiB	Compact
Index	Collation
8 B	latin1_swedish_ci
Total	Creation
16 KiB	Feb 06, 2018 at 12:18 PM

LAMPIRAN B PROGRAM

B.1 Pemrograman Arduino MEGA 2560

```
#include <SoftwareSerial.h>
#include <ESP8266WiFi.h>

SoftwareSerial MS(12,SW_SERIAL_UNUSED_PIN, false,64);

const char* ssid = "SPR G-21";
const char* pass = "dotamania";
const char host[] = "1927.0.0.1";
char dt;

void setup() {
  Serial.begin(115200);
  delay(10);

  // We start by connecting to a WiFi network

  //Serial.println();
  Serial.println("");
  Serial.print("Connecting to ");
  Serial.println(ssid);
  String data;
  data = "";

  WiFi.begin(ssid, pass);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
```

```

    Serial.println(WiFi.localIP());
}
void loop() {

String data = "data1="+ dt;

// Use WiFiClient class to create TCP connections
WiFiClient client;
const int httpPort = 80;
if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
}

    if(MS.available()){
        dt=(char)MS.read();
    }

    // We now create a URI for the request
    String url = "POST /add.php? HTTP/1.1";
client.println(url);
client.print(data);
client.println("Host: 192.168.1.200");
client.println("Connection: close");
client.println();
delay(500);

// Read all the lines of the reply from server and print them to
Serial
while(client.available()){
    String line = client.readStringUntil('\r');
    Serial.print(data);
}

}

```


B.2 Pemrograman WEMOS D1 Mini

```
#include <hidboot.h>
#include <usbhub.h>

///include <avrpins.h>
// Satisfy the IDE, which needs to see the include statement in the
ino too.
#ifdef dobogusinclude
#include <spi4teensy3.h>
#include <SPI.h>
#endif
#define LEVER_SWITCH_PIN 2
int pressSwitch = 0;

long previousMillis = 0;
unsigned long currentMillis = 0;
long interval = 200;

class KbdRptParser : public KeyboardReportParser
{
    void PrintKey(uint8_t mod, uint8_t key);

protected:

    void OnKeyDown (uint8_t mod, uint8_t key);
    void OnKeyPressed(uint8_t key);
};

void KbdRptParser::OnKeyDown(uint8_t mod, uint8_t key)
{
    uint8_t c = OemToAscii(mod, key);

    if (c)
```

```

    OnKeyPressed(c);
}

void KbdRptParser::OnKeyPressed(uint8_t key)
{
    static uint32_t next_time = 0;    //watchdog
    static uint8_t current_cursor = 0; //tracks current cursor position

    Serial.print( (char)key );
    Serial3.print((char)key);
};

USB Usb;
//USBHub Hub(&Usb);
HIDBoot<USB_HID_PROTOCOL_KEYBOARD>
HidKeyboard(&Usb);

uint32_t next_time;

KbdRptParser Prs;

void detector() //For Limitswitch as detector
{
    pinMode(LEVER_SWITCH_PIN, INPUT);
    pressSwitch = digitalRead(LEVER_SWITCH_PIN);
    if (pressSwitch == 1)
    {
        Serial2.print("Detected");
        Serial.print("SEND");
    }
}

void setup()
{
    Serial.begin( 115200 );
    Serial2.begin(115200);
    Serial3.begin(115200);
}

```

```

    pinMode(LEVER_SWITCH_PIN, INPUT);
    pinMode(3, OUTPUT);
    #if !defined(__MIPSEL__)
        while (!Serial); // Wait for serial port to connect - used on
Leonardo, Teensy and other boards with built-in USB CDC serial
connection
    #endif
    Serial.write("Start");

    if (Usb.Init() == -1)
        Serial.write("OSC did not start.");

    delay( 200 );

    HidKeyboard.SetReportParser(0, &Prs);

}

void loop()
{
    Usb.Task();//For Barcode

    currentMillis = millis();
    if (currentMillis - previousMillis > interval) { // READ ONLY
ONCE PER INTERVAL
        previousMillis = currentMillis;
        detector();
    }
}

```

B.3 Pemrograman MySQL

```
SELECT SUBSTRING_INDEX( `Barcode` , '-', 1 ) AS `Product
Name` , SUBSTRING_INDEX( SUBSTRING_INDEX( `Barcode` , '-'
, -2 ) , '-', 1 ) AS `Production Time` , SUBSTRING_INDEX(
`Barcode` , '-', -1 ) AS `Production Numbers`
FROM `LOGGING BARCODE`
ORDER BY `Time` DESC , `Data` DESC
LIMIT 0 , 30
```

```
SELECT `Time`,LEFT(`Barcode`,3) AS `Product
Name`,CONCAT(MID(`Barcode`,5,2),-20,MID(`Barcode`,7,2)) AS
`Production Time`,RIGHT(`Barcode`,4) AS `Production Numbers`
FROM `LOGGING BARCODE`
```

```
SELECT REPLACE( LEFT( `Data` , 4 ) , '2214', 'Semen Cepat
Kering' ) AS `ProdName`
FROM `LOGGING BARCODE`
```

```
SELECT REPLACE( LEFT( `Data` , 4 ) , '2214', 'fourteen' OR
LEFT( `Data` , 4 ) , '2213', 'thirteen') AS `ProdName` FROM
`LOGGING BARCODE`
```

```
SELECT CASE
WHEN (LEFT(`Data` , 4))='2214' THEN
`ProdName`=(REPLACE( LEFT( `Data` , 4 ) , '2214', 'fourteen' ))
WHEN (LEFT(`Data` , 4 ))='2213' THEN
`ProdName`=(REPLACE( LEFT( `Data` , 4 ) , '2214', 'thirteen' ))
END AS `ProdName`
```

```
SELECT
CASE
WHEN (LEFT(`Data`,2))=10 THEN 'Barang 1'
WHEN (LEFT(`Data`,2))=20 THEN 'Barang 2'
WHEN (LEFT(`Data`,2))=30 THEN 'Barang 3'
WHEN (LEFT(`Data`,2))=40 THEN 'Barang 4'
WHEN (LEFT(`Data`,2))=50 THEN 'Barang 5'
```

```
ELSE 'none'  
END AS `ProdName`
```

```
CASE  
WHEN (MID(`Data`,3,2))=01 THEN 'Januari'  
WHEN (MID(`Data`,3,2))=02 THEN 'Februari'  
WHEN (MID(`Data`,3,2))=03 THEN 'Maret'  
WHEN (MID(`Data`,3,2))=04 THEN 'April'  
WHEN (MID(`Data`,3,2))=05 THEN 'Mei'  
WHEN (MID(`Data`,3,2))=06 THEN 'Juni'  
WHEN (MID(`Data`,3,2))=07 THEN 'Juli'  
WHEN (MID(`Data`,3,2))=08 THEN 'Agustus'  
WHEN (MID(`Data`,3,2))=09 THEN 'September'  
WHEN (MID(`Data`,3,2))=10 THEN 'Oktober'  
WHEN (MID(`Data`,3,2))=11 THEN 'November'  
WHEN (MID(`Data`,3,2))=12 THEN 'Desember'  
ELSE 'none'  
END AS `ProdTime`
```

```
FROM `LOGGING BARCODE`
```

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

LAMPIRAN C DATASHEET

Datasheet USB Host Shield

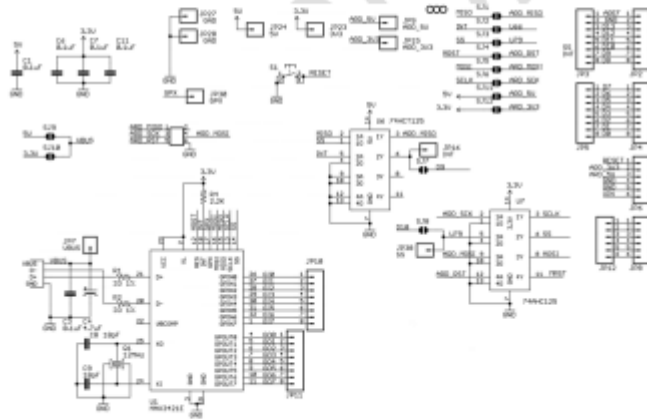
Specifications

- Works with standard (dual 5/3.3V) and 3.3V-only (for example, Arduino Pro) boards.
- Operates over the extended -40°C to +85°C temperature range
- Complies with USB Specification Revision 2.0 (Full-Speed 12Mbps Peripheral, Full/Low-Speed 12Mbps/1.5Mbps Host)

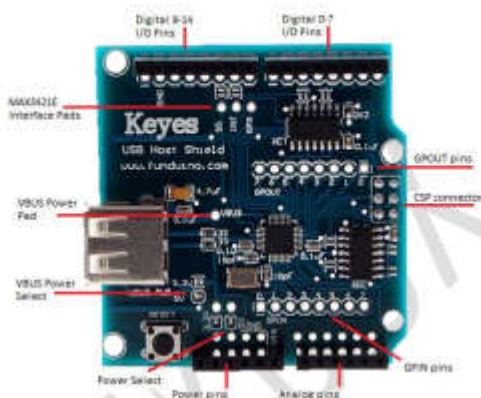
The following device classes are currently supported by the shield:

- HID devices, such as keyboards, mice, joysticks, etc.
- Mass storage devices, such as USB sticks, memory card readers, external hard drives (FAT32 Type File System - Arduino Mega only)

Schematic



Shield Overview



- **Power Select 2** solder jumpers marked "5V" and "3.3V". They are used for different power configurations. The configuration shown, when both jumpers are closed, is suitable for official Arduinos, such as UNO, Duemilanove, Mega and Mega 2560. See Power Options section for detailed explanation.
- **Power pins** are used to connect to power pins of Arduino board. RESET, 3.3V, 5V and GROUND signals from this connector are used.
- **Analog pins** are not used by the shield. They are provided to simplify mounting and provide pass-through for shields mounted atop of USB Host Shield in a stack.
- **GPIN pins**. Eight 3.3V general-purpose digital input pins of MAX3421E. They are used primarily to interface with buttons, rotary encoders and such. GPIN pins can also be programmed as a source of MAX3421E interrupt. An example of GPIN use can be seen in [digital camera controller](#) project.

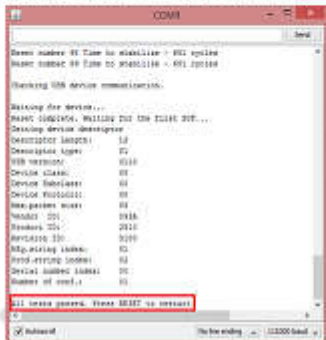
- ICSP connector is used by the shield to send/receive data using SPI interface. SCK, MOSI, MISO and RESET signals from this connector are used.
- GPOUT pins are eight 3.3V general-purpose digital output pins of MAX3421E. They can be used for many purposes; I use it to drive HD44780-compatible character LCD, as can be seen in digital camera controller circuit, as well as this [keyboard example](#). Max_LCD library which is part of standard USB Host library software package uses some of GPOUT pins.
- Digital I/O pins 0-7, like already mentioned analog pins are not used by the shield and provided only for convenience.
- Digital I/O pins 8-13. In this group, the shield in its default configuration uses pins 9 and 10 for INT and SS interface signals. However, standard-sized Arduino boards, such as Duemilanove and UNO have SPI signals routed to pins 11-13 in addition to ICSP connector, therefore shields using pins 11-13 combined with standard-sized Arduinos will interfere with SPI. INT and SS signals can be re-assigned to other pins (see below); SPI signals cannot.
- MAX3421E interface pads are used to make shield modifications easier. Pads for SS and INT signals are routed to Arduino pins 10 and 9 via solder jumpers. In case pin is taken by other shield a re-routing is necessary, a trace is cut and corresponding pad is connected with another suitable Arduino I/O pin with a wire. To undo the operation, a wire is removed and jumper is closed. See interface modifications section for more information. GPX pin is not used and is available on a separate pad to facilitate further expansion. It can be used as a second interrupt pin of MAX3421E.
- VBUS power pad. This pad is used in advanced power configurations, described in Power Options section.

How to Test

1. Mount the Keyes USB Host Shield into your Arduino.
2. Download [USB Host Shield Library](#) and extract it to library folder in your Arduino directory.
3. Open board_qc then click upload. You can find this sketch at Arduino IDE File> Examples > USB_Host_Shield > board_qc.
4. Click Serial Monitor and set the baud rate to 115200.
5. When prompted, plug in a USB device to the USB Host Shield's USB port.

```
Checking USB device communication.
Waiting for device...
Reset complete. Waiting for the first DDF...
```

6. If the shield is fully functional, you should be seeing this Serial Monitor:



Sample Program

Track a mouse actions through Arduino. This program will display the status of the mouse connected to the Keyes USB Host shield. This sketch is also included in the USB Host Library.

You need:

Arduino
Keyes USB Host Shield

Sketch

```
#include <hidboot.h>
#include <usbhub.h>
// Satisfy IDE, which only needs to see the include statement in the in
// o...
#ifdef debug
#include <SPI4teensy3.h>
#endif

class MouseRptParser : public MouseReportParser
```

```

protected:
    virtual void OnMouseMove (MOUSEINFO *mi);
    virtual void OnLeftButtonUp (MOUSEINFO *mi);
    virtual void OnLeftButtonDown (MOUSEINFO *mi);
    virtual void OnRightButtonUp (MOUSEINFO *mi);
    virtual void OnRightButtonDown (MOUSEINFO *mi);
    virtual void OnMiddleButtonUp (MOUSEINFO *mi);
    virtual void OnMiddleButtonDown (MOUSEINFO *mi);
};

void MouseRptParser::OnMouseMove (MOUSEINFO *mi)
{
    Serial.print ("dx=");
    Serial.print (mi->dx, DEC);
    Serial.print (" dy=");
    Serial.println (mi->dy, DEC);
};

void MouseRptParser::OnLeftButtonUp (MOUSEINFO *mi)
{
    Serial.println ("L Butt Up");
};

void MouseRptParser::OnLeftButtonDown (MOUSEINFO *mi)
{
    Serial.println ("L Butt Dn");
};

void MouseRptParser::OnRightButtonUp (MOUSEINFO *mi)
{
    Serial.println ("R Butt Up");
};

void MouseRptParser::OnRightButtonDown (MOUSEINFO *mi)
{
    Serial.println ("R Butt Dn");
};

void MouseRptParser::OnMiddleButtonUp (MOUSEINFO *mi)
{
    Serial.println ("M Butt Up");
};

void MouseRptParser::OnMiddleButtonDown (MOUSEINFO *mi)
{
    Serial.println ("M Butt Dn");
};

};

USB_Web;
USBHub Hub (4Web);
HIDBoot<HID_PROTOCOL_MOUSE> HidMouse (4Web);

uint32_t next_time;

MouseRptParser Prc;

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

```

```

while (!Serial); // Wait for serial port to connect - used on
Leonardo, Teensy and other boards with built-in USB CDC serial
connection
Serial.println("Start");

if (Usb.Init() == -1)
  Serial.println("OSC did not start.");

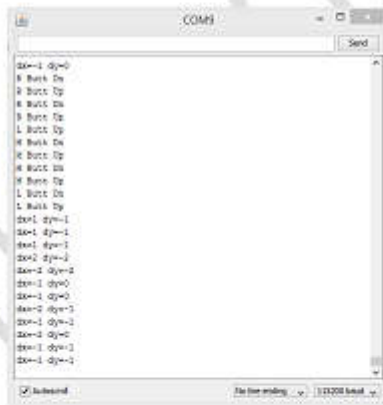
-delay( 200 );

next_time = millis() + 5000;
HidMouse.SetReportParser(0, (HIDReportParser*)&Pars);
}

void loop()
{
  Usb.Task();
}

```

Results



LAMPIRAN D

DAFTAR RIWAYAT HIDUP



Nama : Wana Abdul Ghoni
TTL : BDG, 04 September 1996
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat : Tawangsari Selatan RT 10
RW 02 Taman Sidoarjo
Telp/HP : 085732001755
E-mail : wana.and18@gmail.com

RIWAYAT PENDIDIKAN

1. 2002 – 2008 : SD Negeri Tawangsari 1
2. 2008 – 2011 : SMP Negeri 2 Taman
3. 2011 – 2014 : SMA Muhammadiyah 1 Taman
4. 2014 – 2019 : D3 Teknik Elektro, Program Studi Teknik Elektro Industri – Fakultas Vokasi Institut Teknologi Sepuluh Nopember (ITS)

PENGALAMAN KERJA

1. Kerja Praktek di PT. Pelindo Marine Service Surabaya

PENGALAMAN ORGANISASI

-

PENGALAMAN LAINNYA

-

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