PROYEK AKHIR – VE 180415

SISTEM PENCATAT DAN PELAPORAN *DOWNTIME* PADA AREA *CONVEYOR* MENGGUNAKAN ARDUINO DENGAN KOMUNIKASI *WIFI* DI PT. JATIM AUTOCOMP INDONESIA

Lugas Jabar Waskito
NRP 10311600000059

Dosen Pembimbing
Ir. Arif Musthofa, MT
Lucky Putri Rahayu, S.Si., M.Si.

Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya
2019

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

FINAL PROJECT – VE 180415

DOWNTIME RECORDING AND REPORTING SYSTEMS IN CONVEYOR AREAS USING ARDUINO WITH WIFI COMMUNICATION IN PT. JATIM AUTOCOMP INDONESIA

Lugas Jabar Waskito
NRP 10311600000059

Dosen Pembimbing
Ir. Arif Musthofa, MT
Lucky Putri Rahayu, S.Si., M.Si.

DEPARTEMENT OF ELECTRICAL AUTOMATION ENGINEERING
Vocational Faculty
Institut Teknologi Sepuluh Nopember
Surabaya
2019

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

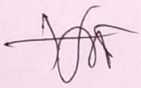
PERNYATAAN KEASLIAN
PROYEK AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Proyek Akhir saya dengan judul "**Sistem Pencatat Dan Pelaporan *Downtime* Pada Area *Conveyor* Menggunakan Arduino Dengan Komunikasi *Wifi* Di Pt. Jatim Autocomp Indonesia**" 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, 30 Juli 2019



Lugas Jabar Waskito
NRP 10311600000059

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

SISTEM PENCATAT DAN PELAPORAN *DOWNTIME* PADA AREA *CONVEYOR* MENGGUNAKAN ARDUINO DENGAN KOMUNIKASI *WIFI* DI PT. JATIM AUTOCOMP INDONESIA

PROYEK AKHIR

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

Menyetujui:

Pembimbing 1	Pembimbing 2
Ir. Arif Musthofa	Lucky Putri Rahayu, S.Si., M.Si.
NIP. 19660811 199203 1 004	NPP. 1991201712058

SURABAYA
JULI, 2019

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

SISTEM PENCATAT DAN PELAPORAN *DOWNTIME* PADA AREA *CONVEYOR* MENGGUNAKAN ARDUINO DENGAN KOMUNIKASI *WIFI* DI PT. JATIM AUTOCOMP INDONESIA

Nama Mahasiswa : Lugas Jabar Waskito
NRP : 10311600000059
Dosen Pembimbing I : Ir. Arif Musthofa, MT
NIP : 19660811 199203 1 004
Dosen Pembimbing II : Lucky Putri Rahayu, S.Si., M.Si.
NPP : 1991201712058

ABSTRAK

Di PT. Jatim Autocomp Indonesia dalam pencatatan *downtime* menggunakan sistem manual yaitu dengan menggunakan *stopwatch* untuk menghitung lama waktunya *downtime* dan dicatat di papan laporan yang ada didekat *conveyor* dan ditulis tiap 2 jam sekali. Dengan sistem manual tersebut membuat kerja GL (*Group Leader*) menjadi kewalahan dan akan mempengaruhi hasil produksi *wire harness* dan hal ini diperlukannya sistem dan pencatat laporan *downtime* agar mempermudah GL (*Group Leader*) dan Supervisor dalam mencatat dan memantau waktu *downtime* di *conveyor*

Maka dari itu pada Proyek Akhir ini dibuatlah sistem pencatat dan pelaporan *downtime* di area *conveyor*. Dengan dibentuknya sistem ini dapat mempermudah kerja GL (*Group Leader*) dan Supervisor dalam memantau waktu *downtime* pada *conveyor*. Metode yang digunakan untuk mendapatkan waktu selama *downtime* yaitu dengan cara menghitung selisih anatara waktu selesai *downtime* dikurangi dengan waktu pertama mengalami *downtime*. Data yang didapat akan dikelola oleh *Arduino Mega* dan akan ditampilkan dalam *database* pada *web*.

Setelah dilakukan pengujian, hasil dari pengujian menunjukan bahwa kecepatan dalam mencatat waktu *downtime* lebih cepat dari biasanya, yang biasanya mencatat tiap 2 jam sekali sekarangan hanya 5 Detik dalam pengiriman data ke *web.*

Kata Kunci : *Data logger, Downtime* Area *Conveyor,* PT. Jatim Autocomp Indonesia, *localhost,*

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

DOWNTIME RECORDING AND REPORTING SYSTEMS IN CONVEYOR AREAS USING ARDUINO WITH WIFI COMMUNICATION IN PT. JATIM AUTOCOMP INDONESIA

Student's Name	**: Lugas Jabar Waskito**
Registration Number	**: 10311600000059**
Supervisor I	**: Ir. Arif Musthofa, MT**
ID	**: 19660811 199203 1 004**
Supervisor II	**: Lucky Putri Rahayu, S.Si., M.Si.**
ID	**: 1991201712058**

ABSTRACT

At PT. East Java Autocomp Indonesia in determining downtime still uses a manual system, namely by calculating using stopwatch assistance as a tool to calculate the length of time downtime and recorded on a report board that has a conveyor and is written every 2 hours. With this manual system makes GL (Group Leader) work become chaotic and will affect production in wire harness production and this requires a system and report keeper of downtime to make it easier for GL (Group Leaders) and Supervisors to record and monitor downtime on the conveyor

Therefore in this Final Project a recording system was created and downtime reporting on the conveyor area. With the establishment of this system, it can simplify the work of GL (Group Leaders) and Supervisors in monitoring downtime on the conveyor. The method used to get time during downtime is by using the difference method between the time downtime is reduced and the first time experiencing downtime. The data obtained will be managed by Arduino Mega and will be displayed in the database on the Web.

After testing, the results of the testing show that the speed in recording downtime is faster than usual, which usually records every 2 hours at a time for only 5 seconds in sending data to the web

Keywords ***:*** *Data Logger, Downtime Conveyor Area,* PT. Jatim Autocomp Indonesia, *localhost*

-----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, shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW yang telah menunjukkan jalan kebanaran.

Proyek Akhir ini disusun untuk memenuhi persyaratan guna menyelesaikan pendidikan Diploma di Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya.

Dengan terselesaikannya Proyek Akhir ini tidak akan terwujud tanpa bantuan, dukungan, doa, serta dorongan semangat dari semua pihak yang terkait. Oleh karena itu, penulis mengucapkan terimakasih kepada:

1. Kedua orang tua yang senantiasa mendoakan dan memberikan dukungan dengan tulus tiada henti.
2. Bapak Ir. Arif Musthofa, MT selaku dosen pembimbing pertama.
3. Ibu Lucky Putri Rahayu, S.Si., M.Si. selaku dosen pembimbing dari kedua.
4. Bapak David Agung Fitriyanto, ST. selaku pembimbing Lapangan.
5. Kepada Bapak / Ibu. Staf dan Karyawan PT. JATIM AUTOCOMP INDONESIA
6. Teman-teman Teknik Elektro Otomasi dan Foxvire DE-11 yang selalu memberikan doa, semangat serta dukungan.
7. 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 ilmu di kemudian hari.

Surabaya, 30 Juli 2019

Lugas Jabar Waskito
10311600000059

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

DAFTAR ISI

HALAMAN

DAFTAR GAMBAR

HALAMAN

DAFTAR TABEL

HALAMAN

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

BAB I
PENDAHULUAN

1.1 Latar Belakang

Downtime adalah merupakan sumber utama yang menyebabkan kehilangan produktifitas di sebagian besar perusahaan manufaktur. *Downtime* sendiri merupakan waktu dimana suatu peralatan tidak dapat beroperasi lancar disebabkan ada suatu masalah. Contohnya masalah terjadi di perusahaan manufaktur sendiri seperti di PT. Jatim Autocomp Indonesia yaiu ketika pada saat proses produksi berjalan tiba – tiba rel kereta pada *conveyor* terputus dan itu menyebabkan proses produksi terhambat. Dikarenakan rel terputus pihak operator harus menekan tombol untuk memberhentikan sementara *conveyor* untuk di perbaiki oleh teknisi agar proses produksi berjalan normal kembali. Dengan kejadian masalah tersebut banyak waktu yang terbuang untuk melakukan perbaikan yang seharusnya bisa dilakukan untuk produksi barang. Apalagi didalam industri, waktu satu detik saja sangat bernilai. Untuk menghasilkan produk sesuai target perusahaan, hitungan produksi per Detiknya sangat berharga.

Untuk contoh masalah lainnya yang sering terjadi di PT. Jatim Autocomp Indonesia yaitu *error* pada bagian mesin PLC jadi harus mengunggah program kembali, kesalahan pada proses produksi kabel, atau kewalahan pada saat produksi dan masih banyak lagi. Untuk parameter kapan harus menekan tombol pada saat *downtime* terjadi yaitu ketika operator melihat proses produksi berjalan tidak normal atau melihat masalah yang sudah dijelaskan di atas dan ketika melihat kejadian tersebut terjadi maka operator langsung menekan tombol dan langsung memanggil teknisi agar masalah tersebut atau kejadian yang tidak normal dapat teratasi dan proses produksi berjalan normal kembali. Dalam menekan tombol pun ada peraturannya masing - masing. Peraturannya yaitu jadi terdapat di setiap *conveyor* terdapat 2 tombol yaitu tombol kuning dan tombol merah. Menekan tombol kuning ketika operator melihat masalah ringan seperti kewalahan dalam produksi atau bisa juga kesalahan memasukkan konektor dalam kabel. Sedangkan menekan tombol merah ketika operator melihat kerusakan parah seperti putusnya rel kereta pada *conveyor*, *error* mada mesin utama jadi harus menguploan ulang program, kesalahan dalam produksi kabel.

Jadi parameter menekan tombol merah dan tombol kuning dilihat dari seberapa masalah yang menganggu pada saat proses produksi. Juga di area *conveyor* tersebut terdapat 2 sirine untuk penanda bahwa area *conveyor* tersebut sedang mengalami *downtime*.

Tetapi sayangnya di PT. Jatim Autocomp Indonesia sendiri, sebuah industri manufaktur yang bekerja secara *non stop* dan tentunya sering terjadi *downtime* masih belum memiliki sistem pencatatan *downtime* yang otomatis pada area *conveyor*. Padahal di area *conveyor* tersebut bukan hanya 1 *conveyor* saja yang di catat *downtime*nya tetapi terdapat 26 *conveyor* yang saling berhubungan jadi kalau salah satu *conveyor* mengalami *downtime* akan berpengaruh ke kerja *conveyor* lainnya dan tentunya menghambat produksi kabel. PT. Jatim Autocomp Indonesia sendiri sampai sekarang masih menggunakan sistem manual dalam pencatatan *downtime* pada area *conveyor*. Yang dimaksud manual yaitu ketika operator melihat masalah maka operator akan menekan tombol dan bersamaan juga menekan tombol *stopwatch*.

Ketika masalah tersebut telah ditangani oleh pihak teknisi kemudian dari operator mematikan *stopwatch* dan kemudian mencatat di papan laporan yang berada di dekat area *conveyor*. Untuk pencatatan di papan laporan sendiri tiap 2 jam sekali sedangkan untuk operator yang mencatat *downtime* adalah GL (*Group Leader*). Papan laporan tersebut bertujuan untuk memberikan informasi ke pada supervisor. Jadi supervisor harus mengecek langsung ke area *conveyor* untuk mengetahui *downtime*,tidak bisa memantau dari jauh. Dengan sistem manual tersebut dapat kita lihat dengan *conveyor* yang banyak dan disana hanya terdapat 1 GL (*Group Leader*) membuat GL kewalahan dalam bekerja.

Maka dari itu untuk membantu GL (*Group Leader*) dalam pencatatan *downtime* pada area *conveyor* dan juga membantu Supervisor dalam memantau *downtime* pada area *conveyor* tersebut, untuk proyek akhir ini dibuatlah sistem pencatatan *downtime* pada area *conveyor* menggunakan *Arduino* dengan komunikasi *WIFI* di PT. Jatim Autocomp Indonesia. Jadi diharapkan dengan adanya sistem ini dapat memantau waktu *downtime* 26 *conveyor* pada area *conveyor* tersebut. Dan juga memudahkan Supervisor dalam memantau *downtime* pada area *conveyor*, karena dapat dipantau dari kejauhan.

1.2 Permasalahan

Permasalahan yaitu dikarenakan alat pencatat *downtime* pada area *conveyor* masih menggunakan sistem manual yaitu dengan cara menggunakan *stopwatch* yang ditulis di papan laporan yang ada didekat *conveyor* masing - masing dan ditulis setiap 2 jam sekali.

1.3 Batasan Masalah

Adapun batasan masalah yang dibahas dalam Proyek Akhir ini meliputi:

- Tidak membahas secara mendetail mengenai pembuatan *web*
- *Microcontroller* menggunakan *Arduino Mega* 2560.

1.4 Tujuan

Tujuan yang akan dicapai dari Proyek Akhir ini adalah membuat sistem pencatat dan pelaporan *downtime* pada area *conveyor* di PT. Jatim Autocomp Indonesia.

1.5 Sistematika Laporan

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

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, batasan masalah, tujuan, sistematika laporan, dan relevansi.

Bab II Teori Penunjang

Dalam bab ini menjelaskan tentang tinjauan tentang konsep Sensor *Proximity* Induktif, *RTC* DS1037, *Ethernet Shield* W5100, *Arduino Mega*, Modul *Optocoupler* 8 *Channel* 24 V to 5 V.

Bab III Perancangan Alat

Dalam bab ini membahas perancangan sistem *hardware* maupun *software* pada sistem pencatat dan pelaporan *downtime* pada area *conveyor* pada Bab II

Bab IV Pengujian dan Analisa

Dalam bab ini membahas tentang, pengujian, serta analisa terhadap prinsip kerja dan proses dari suatu alat yang dibuat.

Bab V Penutup

Dalam bab ini membahas berisi tentang penutup yang menjelaskan tentang kesimpulan dari Proyek Akhir dan saran – saran untuk pengembangan alat ini lebih lanjut.

1.6 Relevansi

Dengan adanya sistem pencatat dan pelaporan *downtime* pada area *conveyor* ini diharapkan mampu menangani masalah dalam pencatatan waktu *downtime* di PT. Jatim Autocomp Indonesia yang menggunakan sistem manual.

BAB II

TEORI DASAR

Pada bab ini dibahas mengenai teori yang dapat menunjang dalam proses pembuatan Proyek Akhir. Adapun pada bab ini terdiri dari 8 subbab, yaitu Sensor Proxmity, *RTC DS1307*, *Arduino Mega*, *Software Arduino IDE*, *Software* XAMPP, *Ethernet Shield* W5100, Modul *Optocoupler* 8 channnel 24 V to 5 V, dan *Router*.

2.1 Sensor *Proximity* [1]

Sensor *Proximity* atau Sensor Jarak adalah sensor elektronik yang mampu mendeteksi keberadaan objek di sekitarnya tanpa adanya sentuhan fisik. Dapat juga dikatakan Sensor *Proximity* adalah perangkat yang dapat mengubah informasi tentang gerakan atau keberadaan objek menjadi sinyal listrik. Sensor *Proximity* tidak menggunakan bagian-bagian yang bergerak atau bagian mekanik untuk mendeteksi keberadaan objek disekitarnya, melainkan menggunakan medan elektromagnetik ataupun sinar radiasi elektromagnetik untuk mengetahui apakah ada objek tertentu disekitarnya. Jarak maksimum yang dapat dideteksi oleh sensor ini disebut dengan "nominal *range*" atau "kisaran nominal". Beberapa Sensor *Proximity* juga dilengkapi fitur pengaturan nominal *range* dan pelaporan jarak objek yang dideteksi.

Sensor *Proximity* ini adalah perangkat yang sangat berguna apabila digunakan di tempat yang berbahaya. Namun seiring dengan perkembangan teknologi, sensor *proximity* ini telah banyak digunakan untuk mempermudah pekerjaan manusia. Bahkan, Sensor Jarak ini sudah diaplikasikan pada hampir semua jenis ponsel pintar (*smartphone*) zaman ini. Sensor *Proximity* ini umumnya digunakan untuk mendeteksi keberadaan, kedekatan, posisi dan penghitungan pada mesin otomatis dan sistem manufaktur. Mesin-mesin yang menggunakan. Sensor *Proximity* ini diantaranya adalah mesin kemasan, mesin produksi, mesin percetakan, mesin pencetakan plastik, mesin pengerjaan logam, mesin pengolahan makanan dan masih banyak lagi.

2.1.1 Sensor Proximty Induktif

Sensor *Proximity* Induktif adalah sensor jarak yang digunakan untuk mendeteksi keberadaan logam baik logam jenis *Ferrous* maupun logam jenis *non-ferrous*. Sensor ini dapat digunakan untuk mendeteksi keberadaan (ada atau tidak adanya objek logam), menghitung objek logam. Sensor *Proximity* Induktif sering digunakan sebagai pengganti saklar mekanis karena kemampuannya yang dapat beroperasi pada kecepatan yang lebih tinggi dari sakelar mekanis biasa.

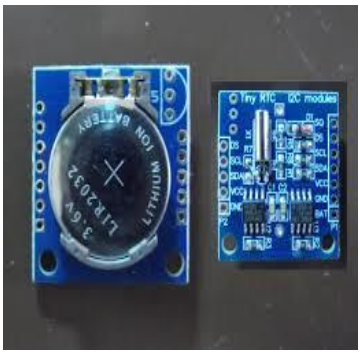
Sensor *Proximity* Induktif pada umumnya terbuat dari kumparan/koil dengan inti ferit sehingga dapat menghasilkan medan elektromagnetik frekuensi tinggi. *Output* dari sensor jarak jenis induktif ini dapat berupa analog maupun digital. Versi Analog dapat berupa tegangan (biasanya sekitar 0 – 10VDC) atau arus (4 – 20mA). Jarak pengukurannya bisa mencapai hingga 2 inci. Sedangkan versi Digital biasanya digunakan pada rangkaian DC saja ataupun rangkaian AC/DC. Sebagian besar Sensor Induktif Digital dikonfigurasi dengan *Output Normally – Open* namun ada juga yang dikonfigurasi dengan *Output Normally – Close*. Sensor *Proximity* Induktif ini sangat cocok untuk mendeteksi benda-benda logam di mesin dan di peralatan otomatis.

Sensor *Proximity* Induktif ini pada dasarnya terdiri dari sebuah osilator, sebuah koil dengan inti ferit, rangkaian detektor, rangkaian *output*, kabel dan konektor. Osilator pada Sensor Jarak ini akan membangkitkan gelombang sinus dengan frekuensi yang tetap. Sinyal ini digunakan untuk menggerakkan kumparan atau koil. Koil dengan Inti Ferit ini akan menginduksi medan elektromagnetik. Ketika garis-garis medan elektromagnetik ini ter-interupsi oleh objek logam, tegangan osilator akan berkurang sebanding dengan ukuran dan jarak objek dari kumparan/koil. Dengan demkian, Sensor *Proximity* Induktif ini dapat mendeteksi adanya objek yang sedang mendekatinya. Pengurangan tegangan osilator ini disebabkan oleh arus Eddy yang diinduksi pada logam yang menginterupsi garis-garis logam. Seperti yang ditunjukkan pada Gambar 2.1.

Gambar 2.1 Sensor *Proximity* Induktif

2.2 *RTC* DS1307 [2]

Realtime clock adalah komponen IC penghitung yang dapat difungsikan sebagai sumber data waktu baik berupa data jam, hari, bulan maupun tahun. Komponen DS1307 berupa IC yang perlu dilengkapi dengan komponen pendukung lainnya seperti crystal sebagai sumber *clock* dan *Battery External* 3,6 V sebagai sumber *energy* cadangan agar fungsi penghitung tidak berhenti. Bentuk komunikasi data dari IC *RTC* adalah I2C yang merupakan kepanjangan dari I*nter Integrated Circuit*. Komunikasi jenis ini hanya menggunakan 2 jalur komunikasi yaitu SCL dan SDA. Semua *Microcontroller* sudah dilengkapi dengan fitur komunikasi 2 jalur ini, termasuk diantaranya *Arduino Microcontroller*. Komponen *RTC* DS1307 memiliki ketelitian dengan *error* sebesar 1 menit per tahunnya. Dapat kita lihat pada Gambar 2.2.



Gambar 2.2 *RTC* DS 1307

Fungsi pin dari komponen *RTC* S1307 adalah sebagai berikut :

1. Pin Vcc (Nomor 8) berfungsi sebagai sumber energy listrik Utama. Tegangan kerja dari komponen ini adalah 5 V, dan ini sesuai dengan tegangan kerja dari *Microcontroller Arduino Board*
2. Pin GND (Nomor 4) Anda harus menghubungkan *ground* yang dimiliki oleh komponen *RTC* dengan *ground* dari *battery back-up*
3. SCL berfungsi sebagai saluran *clock* untuk komunikasi data antara *Microcontroller* dengan *RTC*
4. SDA berfungsi sebagai saluran Data untuk komunikasi data antara *Microcontroller* dengan *RTC*
5. X1 dan X2 berfungsi untuk saluran *clock* yang bersumber dari *crustal external*
6. Vbat Berfungsi sebagai saluran energy listrik dari *Battery external.*

2.3 *Arduino Mega* 2560 [2]

Arduino merupakan perangkat elektronik atau papan rangkaian elektronik *open – source* yang di dalamnya terdapat komponen utama, yaitu sebuah chip *microcontroller* dengan jenis AVR dari perusahaan Atmel. Tujuan menanamkan program pada *microcontroller* adalah agar rangkaian elektronik dapat membaca *input*, memproses *input* tersebut dan kemudian menghasilkan *output* sesuai yang diinginkan. Jadi mikrokontroler bertugas sebagai otak yang mengendalikan *input*, proses, dan *output* sebuah rangkaian elektronik.

Pada Gambar 2.3 tipe *Arduino* yang digunakan pada penelitian kali ini adalah *Arduino Mega* 2560. *Arduino Mega* 2560 adalah mikrokontroler berbasis *ATMega* 2560 dengan *Clock Speed* 16Mhz dan *Flash Memory* 256KB. Pada Tabel 2.1 dijelaskan tegangan operasi untuk *Arduino* jenis ini yaitu 5 V. Sedangkan tegangan *input* yang direkomendasikan yakni 7 – 12 V. *Arduino* ini memiliki 54 pin digital *input/output* pada pin 22-53 dengan 15 pin diantaranya merupakan pin PWM pada pin 0-13, 16 pin *analog input* pada pin A0 – A15, sambungan USB, sambungan catu daya tambahan dan tombol pengaturan ulang.

Gambar 2.3 *Arduino Mega* 2560

Spesifikasi dari *Arduino Mega* 2560 dapat ditunjukkan pada Tabel 2.1 berikut :

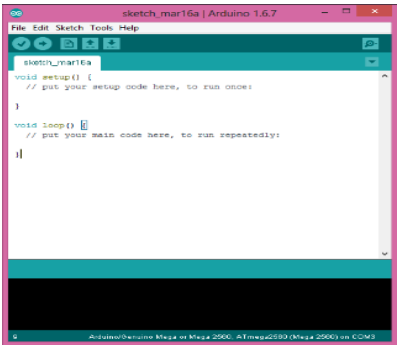
Tabel 2.1 Spesifikasi *Arduino Mega* 2560

Spesifikasi	Keterangan
Chip Mikrokontroler	*ATMega* 2560
Tegangan Operasi	5 V
Teganga *Input* (Rekomendasi)	7 V-12 V
Tegangan *Input* (Limit)	6 V- 20 V
Pin *Digital I/O*	54, (15 buah diantaranya dapat digunakan sebagai *Output* PWM)
Pin Analog *Input*	16 (A0 – A15)
Arus DC per Pin I/O	40 mA
Arus DC Pin 3,3V	50 mA
Memori Flash	256 KB, 8 KB telah digunakan untuk *Bootloader*
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

2.4 *Software Arduino IDE* [3]

Arduino IDE adalah perangkat lunak yang digunakan untuk memprogram, monitoring dan debugging *microcontroller*. *Arduino IDE* itu merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan

terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *software* inilah *Arduino* dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. *Arduino* menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman *Arduino* (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC *microcontroller Arduino* telah ditanamkan suatu program bernama *bootlader* yang berfungsi sebagai penengah antara compiler *Arduino* dengan *microcontroller*. Pada Gambar 2.4 merupakan tampilan awal untuk membuat program pada *software Arduino IDE*.

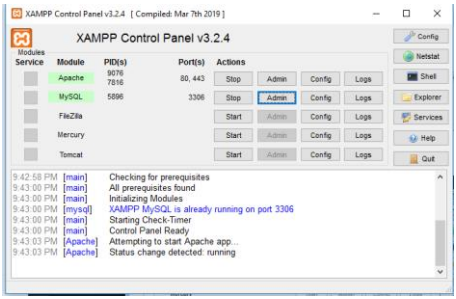


Gambar 2.4 Tampilan *Software Arduino IDE*

2.5 *Software* XAMPP [4]

XAAMP ialah perangkat lunak bebas yang mendukung XAMPP ialah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan campuran dari beberapa program dan tampilan XAMPP dapat dilihat pada Gambar 2.6. Yang mempunyai fungsi sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri dari program MySQL *database*, Apache HTTP *Server*, dan penerjemah ditulis dalam bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi), Apache, MySQL, PHP dan Perl. Program ini tersedia di bawah GNU (*General Public License*) dan bebas, adalah mudah untuk menggunakan *web server* yang dapat melayani tampilan halaman

web yang dinamis. Jika ingin mendapatkan xampp dapat mengunduh langsung dari situs resminya. Pada Gambar 2.6 merupakan tampilan awal untuk membuat program pada *software* XAAMP.



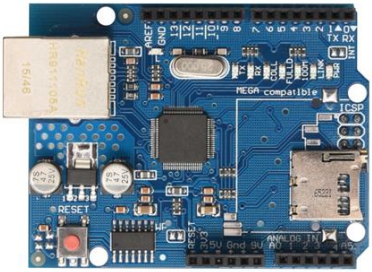
Gambar 2.5 Tampilan XAAMP

2.6 *Ethernet Shield* [5]

Ethernet Shield menambah kemampuan *Arduino Board* agar terhubung ke jaringan komputer. *Ethernet Shield* berbasiskan cip *Ethernet* Wiznet W5100. *Ethernet library* digunakan dalam menulis program agar *Arduino Board* dapat terhubung ke jaringan dengan menggunakan *Arduino Ethernet Shield*. Pada *Ethernet Shield* terdapat sebuah slot *micro-SD*, yang dapat digunakan untuk menyimpan file yang dapat diakses melalui jaringan. *On Board micro-SD card reader* diakses dengan menggunakan *SD library*. *Arduino Board* berkomunikasi dengan W5100 dan *SD card* mengunakan *bus SPI* (*Serial Peripheral Interface*). Komunikasi ini diatur oleh *library* SPI.h dan *Ethernet*.h. *Bus SPI* menggunakan pin digital 11, 12 dan 13 pada *Arduino* Uno, sedangkan pada *Arduino Mega* 2560 terdapat pada pin 50, 51, 52, dan 53.

Pin digital 10 digunakan untuk memilih W5100 dan pin digital 4 digunakan untuk memilih *SD card*. Pin-pin yang sudah disebutkan sebelumnya tidak dapat digunakan untuk *input/output* umum ketika kita menggunakan *Ethernet Shield* karena W5100 dan *SD card* berbagi *bus SPI*, hanya salah satu yang dapat aktif pada satu waktu. Jika kita menggunakan kedua perangkat dalam program kita, hal ini akan diatasi oleh *library* yang sesuai. Jika kita tidak menggunakan 15 salah satu perangkat dalam program kita, kiranya

kita perlu secara *eksplisit mendeselect* nya. Untuk melakukan hal ini pada SD card, set pin 4 sebagai *output* dan menuliskan logika tinggi padanya, sedangkan untuk W5100 yang digunakan adalah pin 10. Gambar *Ethernet Shield* dapat di lihat di Gambar 2.6



Gambar 2.6 *Ethernet Shield*

Spesifikasi dari *Ethernet Shield* dapat ditunjukkan pada Tabel 2.2 berikut :

Tabel 2.2 Spesifikasi *Ethernet Shield*

Spesifikasi	Keterangan
Chip Microcontroller	*Ethernet Shield* W5100
Internal Buffer	16 kb
Kecepatan Koneksi	10/100 Mb (Fast *Ethernet*)
Komunikasi	Serial dengan protokol SPI (*Serial Peripheral Interface*)
Dimensi	7 cm x 5,25 cm x 2,25 cm
Berat	26 gr

2.7 Modul *Optocoupler 8 Channel* 24V to 5V [6]

Optocoupler adalah komponen elektronika yang berfungsi sebagai penghubung berdasarkan cahaya optik. Pada dasarnya *Optocoupler* terdiri dari 2 bagian utama yaitu *transmitter* yang berfungsi sebagai pengirim cahaya optik dan *receiver* yang berfungsi sebagai pendeteksi sumber cahaya. Jenis-jenis *Optocoupler* yang sering ditemukan adalah *Optocoupler* yang terbuat dari bahan semikonduktor dan terdiri dari kombinasi LED (*light emitting diode*)

dan *Phototransistor*. Dalam kombinasi ini, LED berfungsi sebagai pengirim sinyal cahaya optik (*Transmitter*) sedangkan *Phototransistor* berfungsi sebagai penerima cahaya tersebut (*Receiver*). Modul *Optocoupler* dapat dilihat di Gambar 2.7.

Gambar 2.7 Modul *Optocoupler* 8 *Channel*

2.8 *Router* [7]

Router adalah suatu *hardware* jaringan komputer yang berfungsi untuk mengirimkan paket data melalui jaringan atau internet dari satu perangkat komputer ke perangkat lainnya, dimana proses tersebut disebut dengan *routing*.

Setiap *Router* mempunyai fasilitas DHCP (*Dynamic Host Configuration Procotol*) yang dapat di *setting* sedemikian rupa sehinga dapat membagi IP *address*. Selain itu, pada *Router* juga terdapat NAT (*Network Address Translator*) yaitu fasilitas yang memungkinkan suatu alamat IP atau koneksi internet dapat di *sharing* ke alamat IP lain. Gambar *Router* dapat dilihat pada Gambar 2.8.

Gambar 2.8 *Router*

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

BAB III
PERANCANGAN SISTEM DAN IMPLEMENTASI

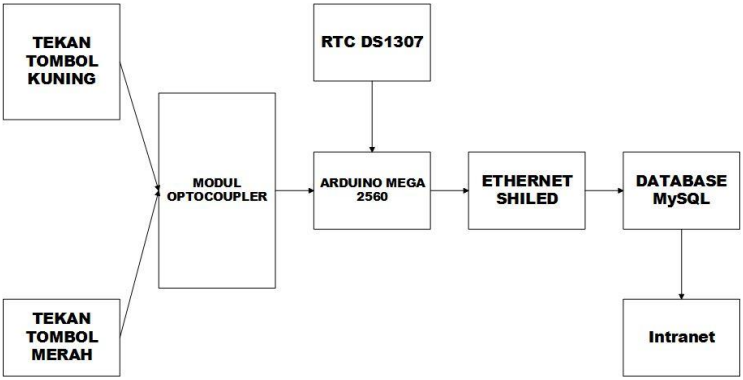
Pada bab ini dibahas mengenai perancangan perangkat keras (*hardware*) dan perangkat lunak (*software*). Hal tersebut guna mewujudkan Proyek Akhir ini.

Untuk perangkat keras meliputi:

1. Peracangan *Box* Panel
2. Perancangan *Port Arduino Mega* 2560
3. Perancangan Rangkaian *RTC DS1307*
4. Perancangan Rangkaian Modul *optocoupler* 24V to 5V
5. Peracangan Posisi Sensor *Proximity* Induktif
6. Perancangan *Prototype* Alat Monitoring *Downtime*

Sedangkan untuk perancangan perangkat lunak (*software*) yang digunakan sebagai pemogramman yaitu dengan menggunakan *Arduino IDE* yang dihubungkan ke mikrokontroler berupa *Arduino Mega* 2560 dan setelah menerima data dari *Arduino* dikirimkan ke *web* menggunakan *Ethernet Shield* sebagai jalur komunikasinya dibantu *Router* sebagai penguat sinyal.

3.1 Blok Fungsional Sistem



Gambar 3.1 Blok Fungsional

Dari Gambar 3.1 dapat dijelaskan bahwa sistem pencatat dan pelaporan *downtime* pada area *conveyor* di PT. Jatim Autocomp

Indonesia yakni bekerja ketika terdapat *downtime* yang terjadi pada saat proses pembuatan *wiring harness* (Pembuatan Kabel Bodi Mobil). Dan disetiap area *conveyor* terdapat 30 *conveyor* yang masing – masing *conveyor* memiliki 2 tombol sebagai penanda bahwa *conveyor* tersebut mengalami *downtime* dan juga untuk memberhentikan *conveyor* sementera. 2 tombol pada setiap *conveyor* ini memiliki definisi masing – masing yaitu untuk tombol kuning adalah untuk indikator kerusakan ringan, kerusakan ringan itu seperti kesalahan dalam merangkai kabel dan juga tombol kuning bukan hanya sebagai indikator kerusakan ringan tapi juga bisa indikator kewalahan dalam pembuatan kabel karena masing – masing *conveyor* memiliki tugas masing – masing dalam pembuatan wiring harness sehingga ketika salah satu *conveyor* kewalahan maka akan menghambat proses kerja *conveyor* yang lainnya. Selain sebagain indikator kerusakan ringan dan indikator kewalahan dalam produksi, tombol kuning juga bisa sebagai indikator penanda terkahir produksi sebelum pekerja melakukan istirahat. Sedangkan tombol merah digunakan sebagai penunjuk indikator kerusakan parah, kerusakan parah itu seperti, rantai pada *conveyor* putus, atau mesin utama tidak bekerja seperti biasanya. Dan yang menekan kedua tombol itu hanya operator karena operator yang mengetahui jenis kerusakan pada *conveyor*.

Setelah saklar dinyalakan mengrimkan sinyal langsung ke *relay* 24 VDC karena *Arduino Mega* 2560 memiliki tegangan kerja sebesar 5 VDC maka dari itu dari *Relay* 24 VDC harus diturunkan menjadi 5VDC agar sinyal dari saklar dapat terbaca di *Arduino Mega* 2560, untuk menurunkan tegangan tersebut disini dibantu dengan modul *optocoupler* 24 VDC to 5VDC. Setelah diturunkan menjadi 5 VDC sinyal masuk ke *Arduino Mega* 2560 dan mulai pengambilan data. Cara pengambilan datanya yaitu menggunakan rumus persamaan 3.1 yaitu selisih waktu antara waktu pertama dengan waktu kedua.

Durasi = Waktu Kedua – Waktu Pertama……. (3.1)

Waktu Pertama di peroleh ketika saklar di tekan langsung mengirim sinyal ke *Arduino Mega* 2560 untuk mengambil data tanggal dan waktu pada saat saklar ditekan, Sedangkan untuk waktu kedua diperoleh ketika saklar di matikan langsung mengirim sinyal

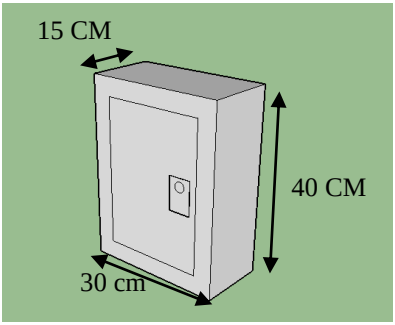
ke *Arduino Mega* 2560 mengambil data tanggal dan waktu pada saat saklar dimatikan. Dalam pengambilan data tanggal dan waktu ini dibantu dengan komponen *RTC* DS1307. Setelah data waktu kedua dan waktu pertama di peroleh kemudian data di kirim ke *database*. Untuk mengirm data tersebut dibantu dengan *Ethernet Shield* dan *Router* sebagai penguat sinyalnya untuk jalur komunikasinya. Sedangkan untuk *Database* sendiri menggunakan MySQL. Setelah data dikirim ke MySQL lalu ditampilkan melalui *Web*.

3.2 Perancangan *Hardware*

Perancangan penunjang *hardware* terdiri dari dari bebrapa subab yang akan dijelaskan per subabnya antara lain : Peracangan *Box* Panel**,** Perancangan *Port Arduino Mega* 2560**,** Perancangan Rangkaian *RTC DS1307***,** Perancangan Rangkaian Modul *optocoupler* 24V to 5V**.**

3.2.1. Perancangan *Box* Panel

Bentuk *box* panel dengan ukuran 30 CM x 40 CM x 15 CM seperti pada Gambar 3.2 yang berisi rangkaian elektronik meliputi rangkaian modul *optocoupler Channel* 8 24V to 5V, *Arduino Mega* 2560, *RTC* DS1307, dan *Ethernet Shield*. Perancangan *box* panel digunakan sebagai tempat untuk meletakkan semua rangkaian kelistrikan yang diperlukan agar lebih praktis karena berada pada satu tempat yang sama, sehingga apabila terjadi kerusakan akan lebih mudah dalam pengecekannya. Dan Bentuk *box* panel dapat dilihat di Gambar 3.2



Gambar 3.2 Perancangan *Box* Panel

3.2.2 Perancangan *Port Arduino Mega* 2560

Pada perancangan rangkaian kontroler ini menggunakan *Arduino Mega* 2560 untuk menerima masukan data dari saklar, membaca data dari *RTC* (*Real-Time Clock*), serta mengirim data dengan *Ethernet Shield* untuk komunikasi alat dengan *Web*. *Arduino Mega* 2560 diprogram agar dapat menghitung berapa lama *downtime* terjadi serta data *downtime* tersebut dijadikan *database*. Rangkaian *Shield* ini dapat mempermudah dalam koneksi antar komponen sistem dan koneksi terhadap *Arduino Mega* 2560. Selain itu, dengan adanya modul *Shield* ini dapat meminimalisir jumlah *wiring* pada sistem ini.

Mapping pin Analog dan Digital dapat dilihat pada Tabel 3.1. Tabel tersebut terdapat informasi tentang *Shield Arduino* yang telah dibuat, sehingga dapat mempermudah pembacaan tiap - tiap pin. Penggunaan *mapping* pin ini dilakukan agar pembaca dapat dengan mudah mengetahui *wiring* antar komponen dan mengurangi terjadinya kesalahan dalam *wiring*, sehingga resiko terjadinya kerusakan antar komponen menjadi berkurang.

Tabel 3.1 *Mapping* Pin Analaog/Digital *Arduino Mega* 2560

No	Keterangan	Pin Analog / Digital
1	*RTC* DS1307	
	VCC	Pin 5 V
	GND	Pin GND
	SDA	Pin 20
	SCL	Pin 21
2	*Proximity* Sensor	
	VCC	Pin Vin
	GND	Pin GND
	Load	Pin A0
3	Tombol Pin	Pin A1
	Tombol Merah *Conveyor* 1	Pin A2
	Tombol Kuning *Conveyor* 1	Pin A3
	Tombol Merah *Conveyor* 2	Pin A4
	Tombol Kuning *Conveyor* 2	Pin A5
	Tombol Merah *Conveyor* 3	Pin A6
	Tombol Kuning *Conveyor* 3	Pin A7
	Tombol Merah *Conveyor* 4	Pin A8

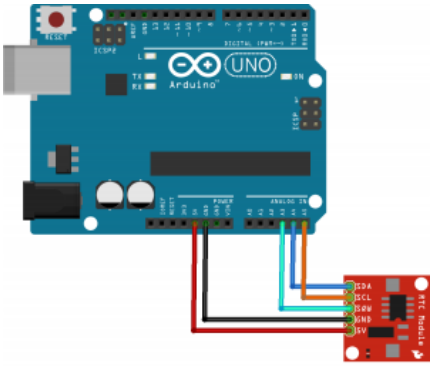
No	Keterangan	Pin Analog / Digital
	Tombol Kuning *Conveyor* 4	Pin A9
	Tombol Merah *Conveyor* 5	Pin A10
	Tombol Kuning *Conveyor* 5	Pin A11
	Tombol Merah *Conveyor* 6	Pin A12
	Tombol Kuning *Conveyor* 6	Pin A13
	Tombol Merah *Conveyor* 7	Pin A14
	Tombol Kuning *Conveyor* 7	Pin A15
	Tombol Merah *Conveyor* 8	Pin 49
	Tombol Kuning *Conveyor* 8	Pin 48
	Tombol Merah *Conveyor* 9	Pin 47
	Tombol Kuning *Conveyor* 9	Pin 46
	Tombol Merah *Conveyor* 10	Pin 45
	Tombol Kuning *Conveyor* 10	Pin 44
	Tombol Merah *Conveyor* A/B	Pin 43
	Tombol Kuning *Conveyor* A/B	Pin 42
	Tombol Merah *Conveyor* S1	Pin 41
	Tombol Kuning *Conveyor* S1	Pin 40
	Tombol Merah *Conveyor* S2	Pin 39
	Tombol Kuning *Conveyor* S2	Pin 38
	Tombol Merah *Conveyor* S4	Pin 37
	Tombol Kuning *Conveyor* S4	Pin 36
	Tombol Merah *Conveyor* S5	Pin 35
	Tombol Kuning *Conveyor* S5	Pin 34
	Tombol Merah *Conveyor* S6	Pin 32
	Tombol Kuning *Conveyor*	Pin 31

No	Keterangan	Pin Analog / Digital
	S6	
	Tombol Merah *Conveyor* S7	Pin 30
	Tombol Kuning *Conveyor* S7	Pin 29
	Tombol Merah *Conveyor* MS	Pin 28
	Tombol Kuning *Conveyor* MS	Pin 27
	Tombol Merah *Conveyor* GR	Pin 26
	Tombol Kuning *Conveyor* GR	Pin 25
	Tombol Merah *Conveyor* EC	Pin 24
	Tombol Kuning *Conveyor* EC	Pin 23
	Tombol Merah *Conveyor* OL	Pin 22
	Tombol Kuning *Conveyor* OL	Pin 2
	Tombol Merah *Conveyor* SG	Pin 3
	Tombol Kuning *Conveyor* SG	Pin 4
	Tombol Merah *Conveyor* V1	Pin 5
	Tombol Kuning *Conveyor* V1	Pin 6
	Tombol Merah *Conveyor* V2	Pin 7
	Tombol Kuning *Conveyor* V2	Pin 8
	Tombol Merah *Conveyor* R1	Pin 9
	Tombol Kuning *Conveyor* R1	Pin 10

No	Keterangan	Pin Analog / Digital
	Tombol Merah *Conveyor* TR	Pin 11
	Tombol Kuning *Conveyor* TR	Pin 12

3.2.3. Perancangan Rangkaian *RTC DS1307*

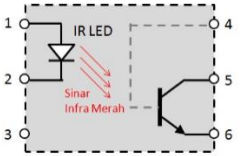
RTC (*Real-Time Clock*) digunakan untuk menyimpan dan menampilkan waktu dan tanggal secara *Real Time*. Pada *RTC* terdapat baterai 3V yang digunakan sumber daya cadangan saat *Power* utama dari rangkaian *RTC* mati, sehingga data waktu dan tanggal yang sedang berjalan tidak akan tereset kembali ke awal. Rangkaian *RTC* dapat dilihat pada Gambar 3.3.



Gambar 3.3 Perancangan Rangkaian *RTC DS1307*

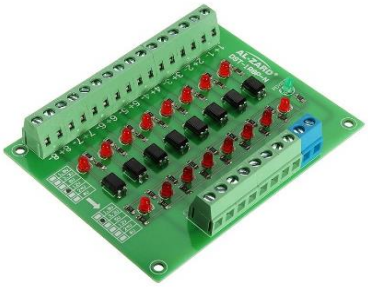
3.2.4 Perancangan Modul *Optocoupler*

Rangkain modul *Optocoupler* 24 V to 5 V 8 *Channel* di gunakan untuk swtich tegangan dari 24 V ke 5 V. Dapat kita tahu bahwa *optocoupler* yaitu komponen elektronika yang berfungsi sebagai penghubung berdasarkan cahaya optik. *Optocoupler* sendiri terdiri dari 2 bagian LED berfungsi sebagai pengirim sinyal cahaya optik (*Transmitter*) sedangkan *Phototransistor* berfungsi sebagai penerima cahaya tersebut (*Receiver*). Untuk cara kerjanya dapat dilihat di Gambar 3.4



Gambar 3.4 Rangkain Kerja Modul *Optocoupler*

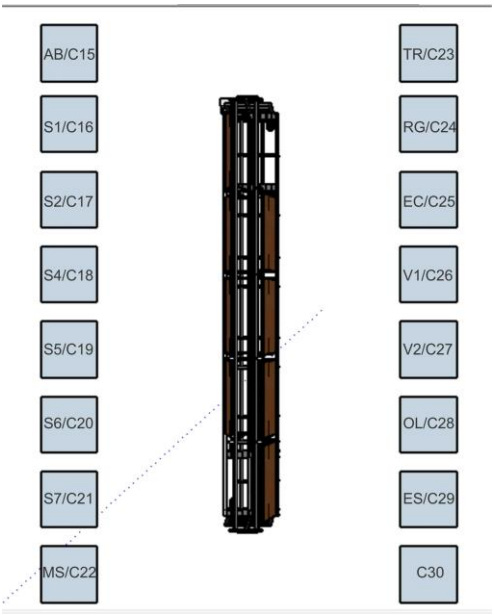
Dari Gambar 3.4 dapat dijelaskan bahwa Arus listrik yang mengalir melalui IR LED akan menyebabkan IR LED memancarkan sinyal cahaya Infra merahnya. Selanjutnya cahaya infra merah yang dipancarkan tersebut akan dideteksi oleh *Phototransistor* dan menyebabkan terjadinya hubungan atau *Switch ON* pada *Phototransistor*. Gambar Modul *Optocoupler* dapat kita lihat di Gambar 3.5.



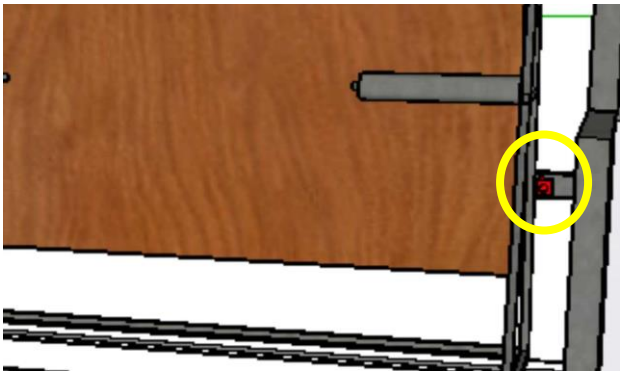
Gambar 3.5 Modul *Optocoupler* 8 *Channel*

3.2.5 Perancangan Posisi Sensor *Proximity* Induktif

Pada Perancangan ini akan diberitahukan tempat atau posisi Sensor *Proximity* Induktif pada *conveyor* yang berada di PT. Jatim Autocomp Indonesia. Perancangan ini berguna untuk memberitahu tempat atau posisi Sensor *Proximity* Induktif yang juga sebagai *home base* pada saat *conveyor* berhenti. Sensor *Proximity* Induktif sendiri disini berfungsi sebagai penanda untuk posisi awal pada saat mesin akan jalan atau juga sebagai penanda untuk berhentinya *conveyor*. Letak Sensor *Proximity* Induktif dapat di lihat di Gambar 3.6 dan Gambar 3.7



Gambar 3.6 *Conveyor* Tampak Atas

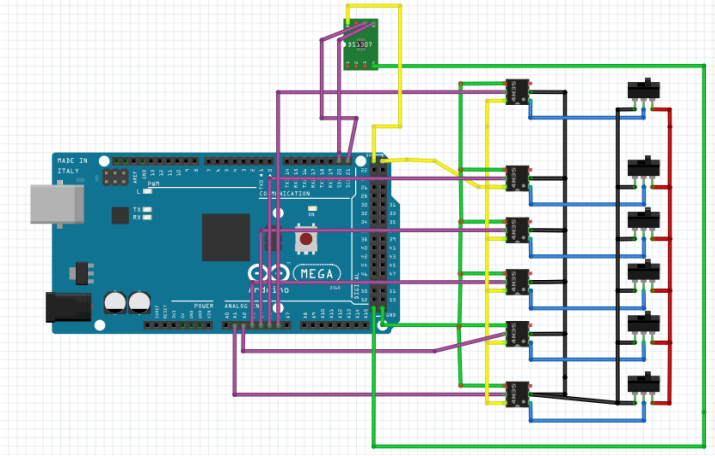


Gambar 3.7 *Conveyor* Tampak Sisi lain

Dapat dilihat pada Gambar 3.6 dan Gambar 3.7 yang berada dalam lingkaran kuning itu adalah Sensor *Proximity* Induktif. Untuk Gambar 3-6 adalah posisi ketika *conveyor* berhenti.

3.2.6 Perancangan *Prototype* Alat Monitoring *Downtime*

Pembuatan *prototype* disini dikarenakan untuk berjaga – jaga jika alat yang di pabrik tidak dapat untuk melakukan pengambilan video. Dari kebijakan pihak perusahaan sendiri sangat melarang untuk pengambilan gambar ataupun video karena untuk menjaga kepercayaan konsumen kepada pihak pabrik sendiri dalam pembuatan *wire harness.* Perancangan dibuat dengan sederhana mungkin dengan 6 *push button* , 1 Modul *Optocoupler* 8 *Channel*, 1 *Arduino Mega* 2560, 1 *Ethernet Shield*, 1 Sensor *Proximity* Induktif, 1 *RTC* DS 1307. Untuk sumber tegangannya sendiri kita beri sama dengan seperti di alat yang ada pabrik yaitu 24 VDC. Perancangan Rangkaian dapat dilihat di Gambar 3.8.



: Tegangan Positif dari *power supply*

: Tegangan Negatif dari *power supply*

: Tegangan Masuk dari *Push*

: Tegangan 5 VDC dari *Arduino*

: *Ground* dari *Arduino*

: *Input* ke *Arduino*

Gambar 3.8 Rancangan *Prototype*

Gambar 3.8 di atas adalah merupakan rancangan *prototype* yang telah dibuat, teradapat 6 *input* atau 6 *push button*. 6 *push button* tersebut mewakili 3 buah *station/conveyor*, yang masing – masing *conveyor* memiliki 2 tombol yakni 1 tombol kuning dan 1 tombol merah. *Push button* tersebut mendapatkan tegangan *input* dari *power supply* 24VDC yang di beri keterangan garis warna merah untuk tegangan positif dan keterangan garis warna hitam untuk tegangan negatif. Untuk pin *output* dari *push button* disambung ke pin anoda pada pin modul *Optocoupler* sedangkan untuk pin katoda nya pada modul *Optocoupler* di sambungkan dengan tegangan minus.

Untuk *outputnya* modul *Optocoupler* terdiri dari dari base, emitter dan collector. Karna *output* yang diinginkan sebesar 5VDC maka Base pada modul *Optocoupler* diberi tegangan sebesar 5VDC yang didapatkan dari pin 5V di *Arduino Mega* 2560. Rangkaian 5V tersebut diberi warna kuning pada Gambar 3.8. Untuk Pin Emitter sendiri disambungkan dengan *ground* pada *Arduino Mega* 2560 yang diberi tanda warna garis hijau. Sedangkan Collector disambungkan dengan pin Analog / Digital pada *Arduino Mega* 2560.

Selanjutnya ada *RTC DS1307* yang disambungkan sesuai dengan pin yang berada di *Arduino Mega* 2560, Untuk Pin SDA disambungkan dengan pin 20, untuk SCL disambungkan dengan pin 21, dan yang terakhir pin VCC dan *Ground* disambungkan dengan pin 5V dan *Ground* pada *Arduino Mega* 2560.

3.3 Perancangan Perangkat Lunak (*Software*)

Pada perancangan perangkat lunak ini bertujuan untuk menunjang perangkat keras (*hardware*) agar terbentuk menjadi satu kesatuan sistem. Perangkat lunak yang digunakan pada sistem ini yaitu program *Arduino Mega* 2560 untuk membuat dan merencanakan program menggunakan *Arduino IDE* dan program Adobe Dreamwever 2018 sebagai tampilan di *web* yang menampilkan data yang sebelumnya sudah disimpan di *database*. Dalam perancangan *software Arduino* dengan fungsi terkait diperlukan beberapa tahapan yang harus dilakukan terlebih dahulu.

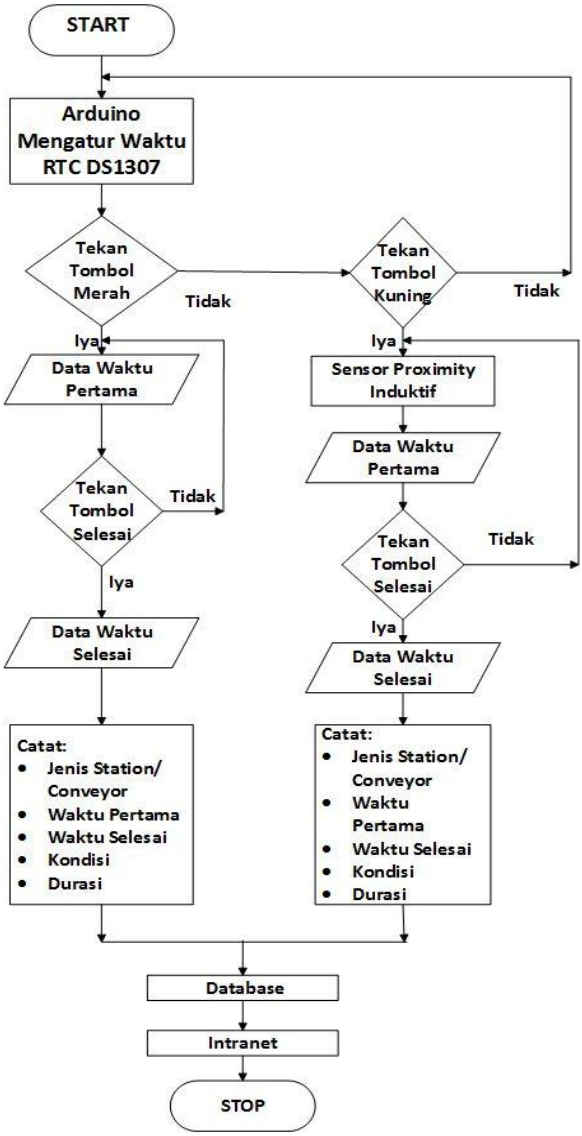
Dengan membuat algoritma setelah membuat *flowchart* sistem ini agar program lebih sederhana. Pemograman *Ardunio ID*E ini juga meliputi program Tombol, program *RTC* (*Real Time Clock*) dan program *Ethernet Shield*. Setelah itu barulah memprogram menggunakan bahasa C. Perancangan *software* ini juga dilakukan

tiap komponen agar mengetahui hasil dari setiap komponen. Hal tersebut diharapkan supaya penggunaan antar komponen saling terkoneksi dan tidak merusak program komponen lain.Perancangan program *Arduino IDE* diberikan tahapan – tahapan pada pembuatan program tiap komponen. Setiap program pada tiap komponen dilakukan percobaan dan hasil program diharapkan sesuai dengan keinginan pengguna. Penggunaan antar komponen sendiri harus mengetahui karakteristik agar dapat membantu dalam pengerjaan program komponen tersebut, karena setiap komponen memiliki karakteristik yang berbeda-beda dan dapat mempengaruhi kinerja dari komponen lainnya.

Perancangan program Dreamwever 2018 dilakukan setelah perancangan program *Arduino IDE* sudah memenuhi kriteria data yang dihasilkan. Hal ini mempengaruhi pengiriman data dari *hardware* yang akan ditampilkan pada *web*. Perancangan program Dreamwever 2018 ini dibuat dengan 2 program utama, yaitu program pengiriman data dan program pada tampilan yang akan dibuat. Setiap pilihan menu pada tampilan Dreamwever 2018 terdapat program yang akan melakukan tindakan apabila menu itu digunakan. Program pada Dreamwever 2018 juga menggunakan program yang terkoneksi pada *software* XAMPP agar dapat di kirim secara online dan ditampilkan *Web.* Untuk pengiriman datanya di bantu dengan menggunakan *Router* sebagai jalur komunikasinya.

3.3.1 *Arduino* IDE

Arduino IDE adalah *software* yang diperlukan oleh board *Arduino*. *Software* ini *Arduino* dilakukan pemrograman untuk melakukan fungsi-fungsi yang kita perintahkan melalui *sintaks*. *Arduino* menggunakan bahasa pemrograman sendiri yang menyerupai Bahasa C. *Arduino IDE* juga telah dilengkapi dengan *library* C/C++ yang biasa disebut *Wiring* yang membuat operasi *input* dan *output* menjadi lebih mudah. Alur program sistem ini dapat dilihat pada Gambar – gambar dibawah ini yang dinama program *Arduino* ini memperlihatkan proses awal hingga akhir program *Arduino*



Gambar 3.9 Program *Arduino*

Penjelasan *Flowchart* berdasarkan Gambar 3.9 adalah sebagai berikut :

1. *Start* adalah ketika program dimulai.
2. Mengatur waktu yang ada di *RTC DS1307* dengan waktu yang *real time*.
3. Tekan tombol ketika *conveyor* mengalami *downtime*.
4. Jika tidak maka kembali mengatur waktu *RTC DS1307*
5. Tekan tombol merah ketika *conveyor* mengalami kerusakan yang parah. Dan membuat *conveyor* langsung berhenti Ketika *conveyor* berhenti, *Arduino* mengambil data waktu pertama ketika *conveyor* itu berhenti. Ketika *conveyor* sudah di dibenarkan dan bisa beroperasi lagi. *Conveyor* bergerak dan *Arduino* menampilkan data di serial monitor *Conveyor* yang mengalami *downtime*, Waktu pertama ketika *conveyor* mengalami *downtime*, Waktu kedua ketika *conveyor* telah selesai di perbaiki, Jenis kerusakan, dan Durasi lamanya *downtime*. Untuk jenis kerusakan sendiri akan menampilkan tulisan "Merah", "Merah" yang dimaksud jenis kerusakan parah.
6. Mengirim data ke *database*
7. Lalu ditampilkan ke dalam *web*
8. Atau Tekan tombol kuning ketika *conveyor* ketika *conveyor* mengalami kerusakan yang ringan. Dan membuat *conveyor* berhenti. *Conveyor* tidak langsung berhenti, tetapi berhentinya pada *home base* yang diberi Sensor *Proximity* Induktif. Sehingga *Arduino* mengambil data waktu pertama ketika telah membaca Sensor *Proximity* Induktif dan *conveyor*nya berhenti. Ketika *conveyor* sudah di dibenarkan dan bisa beroperasi lagi. *Conveyor* bergerak dan *Arduino* menampilkan data di serial monitor *conveyor* yang mengalami *downtime*, Waktu pertama ketika *conveyor* mengalami *downtime*, Waktu kedua ketika *conveyor* telah selesai di perbaiki, Jenis kerusakan, dan Durasi lamanya *downtime*. Untuk jenis kerusakan sendiri akan

menampilkan tulisan “Kuning”, “Kuning“ yang dimaksud jenis kerusakan ringan.

9. Mengirim data ke *database*
10. Lalu ditampilkan ke dalam *web*
11. Dan selesai

3.3.2. Perancangan Pengirim Dengan *Ethernet*

Dalam mengirim dari *Arduino* ke *Ethernet Shield* kemudian dibantu dengan *Router* dan modem lalu dikirim ke *database* dan ditampilkan di*web* pasti memerlukan alamat dalam pengirimnya. untuk *Byte* IP adalah alamat untuk *Ethernet Shield* sendiri. *Byte* DNS adalah IP alamat *web*, *Byte* serv adalah IP dari *Ethernet Shield* itu sendiri, *servername* yaitu alamat *web* mu sendiri yang menampilkan data monitoring. Alamat tadi dapat dilihat di Gambar 3.10.

```
byte ip[] = {192, 168, 1, 18 }; //Enter the IP of ethernet
byte serv[] = {192,168,1,5 } ; //Enter the IPv4 address;
EthernetClient cliente;
RTC_DS1307 rtc;
```

Gambar 3.10 Program Alamat *Ip Ethernet*

Pada Gambar 3.11 ini adalah perintah *Ethernet* untuk mengirimkan ke data ke *database* perintah tersebut yaitu *Cliente print*. data tersebut dikirm dan sambungkan ke *database* dengan perintah yaitu /yoi/data/.php. Apabila terkirm akan menampilkan “*Done Send*” pada serial monitor dan apabila data tidak terkirim akan menampilkan “*Connection Failed*”. Program dapat dilihat pada Gambar 3.11.

```
//////////////////KIRIM DATA//////////////////////////////////////////////////
void kirim (int stationKe, String warnaTombol ){
     if (cliente.connect(serv, 80)) { //Connecting at the IP address and port we saved before
    cliente.print("GET /yoi/data.php?"); //Connecting and Sending values to database
    cliente.print("Station=");
    String stationCetak ="Station"+String(stationKe);
    cliente.print(stationCetak);
    cliente.print("&WaktuPertama=");
    String waktu1Print=String(jam1)+':'+String(menit1)+':'+String(detik1);
    cliente.print(waktu1Print);
    cliente.print("&WaktuSelesai=");
    String waktu2Print=String(jam2)+':'+String(menit2)+':'+String(detik2);
    cliente.print(waktu2Print);
    cliente.print("&Kondisi=");
    cliente.println(warnaTombol);
    cliente.print("&Durasi=");
    cliente.println(hasil);
    Serial.println("Done Send");
    cliente.stop(); //Closing the connection
    }
    else {
    // if you didn't get a connection to the server:
    Serial.println("connection failed");
    }
```

Gambar 3.11 Program Mengirim Data Dengan *Ethernet*

3.3.3. Perancangan *Database*

Basis data (*database*) adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. *Database* pada sistem ini berisi Tanggal, Jenis Station, waktu Pertama, waktu Selesai. Waktu kedua, Kondisi, dan yang terkahir hasil atau durasi. Dalam membuat *database* ini, menggunakan *Software* yang dinamakan phpMyAdmin. Bentuk tabel dapat dilihat di Gambar 3.12.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	date	date			No	None			Change Drop More
3	Station	text	latin1_swedish_ci		No	None			Change Drop More
4	WaktuPertama	text	latin1_swedish_ci		No	None			Change Drop More
5	WaktuSelesai	text	latin1_swedish_ci		No	None			Change Drop More
6	Kondisi	text	latin1_swedish_ci		No	None			Change Drop More
7	Durasi	int(50)			No	None			Change Drop More

Gambar 3.12 Tabel *Database* Pada PhpMyadmin

3.3.4. Perancangan *Web*

Pada perancangan *web* ini dilakukan beberapa pemograman, diantaranya pemograman mySQL dengan koneksi *Arduino Mega* 2560, program pengirim data *Input* dari *database* ke *web*.data Pemograman diberikan tahapan dalam mengelola data yang akan dijadikan *database*. Perancangan program ini dapat membantu untuk mengoneksikan antara *hardware* dan *web*. Penggunaan program – program ini akan memnyesuaikan hasil dari data yang dikeluarkan oleh *hardware* atau *output Arduino*. Dengan hasil data yang dapat dilihat pada serial monitor *Arduino*, pemograman dapat menyesuaikan dengan hasil tersebut agar kolom *database* yang telah dibuat akan sesuai.

Pada Gambar 3.13 dapat dilihat program koneksi antara *database* dengan *Arduino*. Program ini digunakan agar data yang dihasilkan *Arduino* dapat dikirim ke *database*.

connection - Notepad

File Edit Format View Help

```
<?php
$username = "root";
$pass = "";
$host = "localhost";
$db_name = "yoi";
$con = mysqli_connect ($host, $username, $pass);
$db = mysqli_select_db ( $con, $db_name );
?>
```

Gambar 3.13 Program Koneksi Data Ke MySQL

Hasil dari data yang telah dikirim dapat dilihat pada tampilan *web* yang telah dibuat pada *software* Dreamwever 2015. Pada Gambar 3.14 dapat dilihat program mengirim data ke tabel *database* yang telah disesuaikan dengan *output Arduino*. Dari koneksi *hardware* data dikirim dan akan disimpan pada *database*.

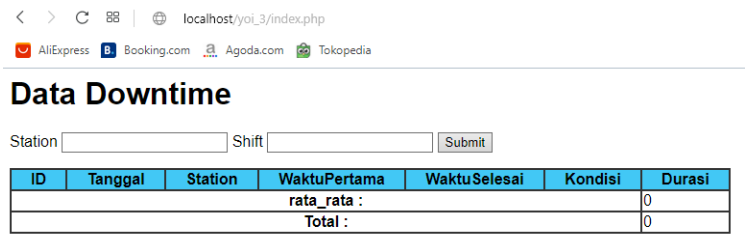
data - Notepad

File Edit Format View Help

```
<?php
include ('connection.php');
$sql_insert = "INSERT INTO data (Station, WaktuPertama, WaktuSelesai, Kondisi, Durasi)
VALUES ('".$_GET["Station"]."', '".$_GET["WaktuPertama"]."', '".$_GET["WaktuSelesai"]."', '".$_GET["Kondisi"]."', '".$_GET["Durasi"]."')";
if(mysqli_query($con,$sql_insert))
{
echo "Done";
mysqli_close($con);
}
else
{
echo "error is ".mysqli_error($con );
}
?>
```

Gambar 3.14 Program Mengirim Data Ke Tabel *Database*

Setelah itu akan dipanggil atau ditampilkan pada *web* yang datanya sebelumnya disimpan di *database*. Yang ditampilkan pada Gambar 3.15.



Gambar 3.15 Tampilan *Web*

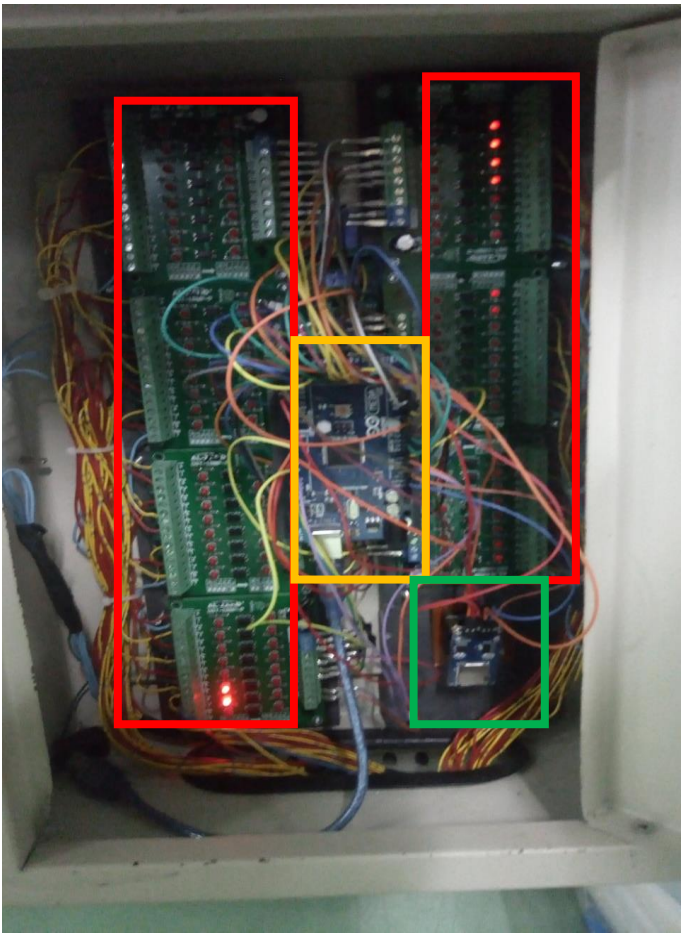
BAB IV
PENGUJIAN DAN ANALISA

Pada bab ini membahas tentang pengujian alat "Sistem Pencatat Dan Pelaporan *Downtime* Pada Area *Conveyor* Menggunakan *Arduino* Dengan Komunikasi *WIFI* Di PT. Jatim Autocomp Indonesia". Pengujian alat yang dilakukan merupakan pengujian terhadap perangkat keras (*Hardware*) dan pengujian *Web* (*Software*) dari sistem secara keseluruhan yang telah selesai dibuat untuk memastikan agar komponen - komponen sistem yang akan digunakan dapat berfungsi dengan baik. Selain pengujian masing masing komponen juga terdapat tampilan realisasi alat juga. Pengujian ini dilakukan dari sebagian komponen utama yang ada pada alat. Pengujian dan analisis alat terbagi menjadi beberapa bagian, diantaranya pengujian *RTC* (*Real Time Clock*) DS 1307, pengujian Modul *Optocoupler*, pengujian Koneksi *Router* Terhadap Jarak, pengujian Sensor *Proximity* Induktif, pengujian *Database* dan *Web*, yang terkahir pengujian keseluruan sistem alat.

1. Realisasi Alat
2. Pengujian *RTC* (*Real Time Clock*) DS1307
3. Pengujian Modul *Optocoupler*
4. Pengujian Jarak Pengirim Dengan *Router*
5. Pengujian Sensor *Proximity* Induktif
6. Pengujian *Database* dan *Web*
7. Pengujian Keseluruhan

4.1 Realisasi Alat

Setelah melakukan berbagai proses perancangan alat, yang telah di bahas di bab 3, tahap selanjutnya yaitu tahap terakhir pada Proyek Akhir yaitu merealisasikan alat monitoring *downtime*. Yang terdiri dari *Arduino Mega* 2560, Modul *Optocoupler* 8V, *RTC* DS 1307, *Sensor Proximity Induktif* dan *Ethernet* Shiled W5100. Untuk lebih jelasnya lagi dapat dilihat di Gambar 4.1



: Modul *Optocoupler*
: *Arduino Mega* 2560
: *RTC DS1307*

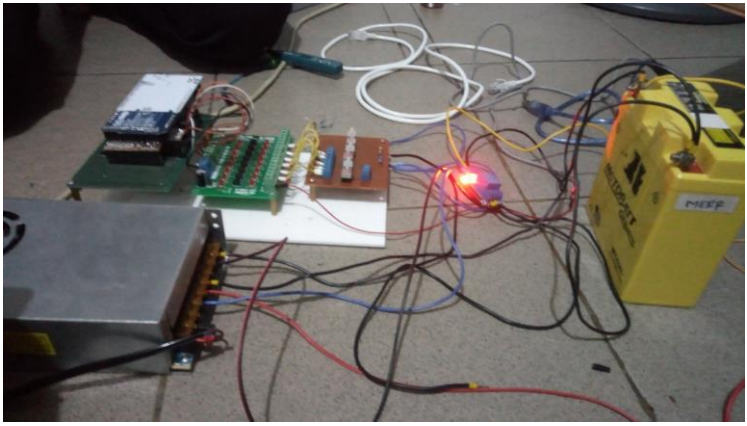
Gambar 4.1 Realisasi Alat

Sedangkan untuk realisasi alat sendiri disini terdapat komponen yang berguna untuk menunjang alat ini Pencatatan dan Pelaporan *Downtime* ini. Disini Terdapat 7 Modul *Optocoupler*

digunakan sebagai me *switch* tegangan dari 24VDC ke tegangan kerja *Arduino* yaitu sebesar 5VDC. Karena 1 Area *conveyor* terdapat 26 *Station* / *Conveyor* dan masing masing *Station* / *Conveyor* memiliki 2 tombol masing – masing yakni tombol merah dan tombol kuning sehingga 1 area *conveyor* memiliki 52 *input*.

Maka dari itu mengguanakan 7 Modul *Optocoupler* ini untuk menampung *input* sebanyak 52 *input*. Satu *input* lagi untuk *input* sensor *proximity* induktif. Gambar 7 Modul *Optocoupler* ini dapat dilihat di Gambar 4.1 yang diberi kotak warna merah.

Selanjutnya di Gambar 4.1 yang diberi kotak warna hijau yaitu adalah komponen *RTC* DS1307, komponen ini digunakan untuk menampilkan waktu dan tanggal. Selanjutnya untuk kompenen yang ditandi dengan kotak berwarna *orange* yaitu *Arduino Mega* 2560.



Gambar 4.2 Realisasi *Prototype* Alat Monitoring *Downtime*

Pada Gambar 4.2 merupakan *prototype* dari alat sitem pencatat dan pelaporan *downtime* yang ada pada Gambar 4.1. Di Gambar 4-2 hanya menggunakan 6 tombol saja yang masing- masing tombol memiliki 2 tombol untuk 1 *station* / *conveyor*. Untuk tombolnya sendiri terdapat 1 tombol merah dan 1 tombol kuning. Untuk tegangan *input* sendiri menggunakan *Power Supply* sebesar 24VDC. Selanjutnya juga terdapat Modul *Optocoupler* sebagai me

switch tegangan dari 24 VDC ke tegangan kerja *Arduino* yaitu sebesar 5VDC. Selanjutnya ada *Arduino* dan *Ethernet*.

4.2 Pengujian *RTC* DS1307

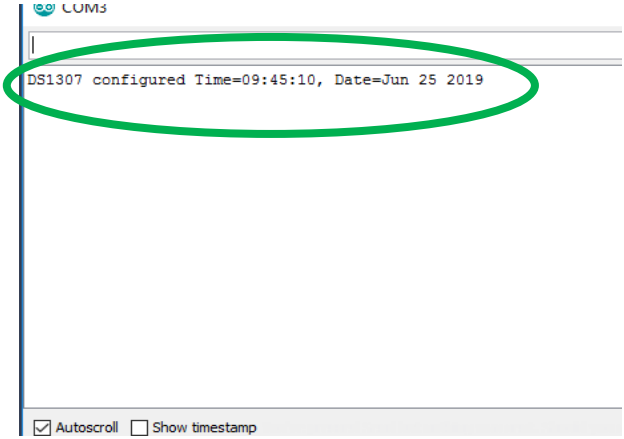
Pengujian *RTC* DS1307 ini dilakukan untuk mengetahui format tanggal dan waktu sesuai dengan kondisi *Real-Time*. Pengujian ini dilakukan dengan menggunakan *Arduino Mega* 2560. *RTC* memerlukan *Supply* +5V yang didapatkan dari 5 V *Arduino Mega* 2560, pin SDA / pin 20 *Arduino Mega* 2560 ke SDA *RTC* dan pin SCL / pin 21 *Arduino Mega* 2560 ke SCL *Arduino Mega* 2560. Format tanggal dan waktu yang dikirim oleh *RTC* akan di tampilkan pada serial monitor.

```
bool parse=false;
bool config=false;

// get the date and time the compiler was run
if (getDate(__DATE__) && getTime(__TIME__)) {
  parse = true;
  // and configure the RTC with this info
  if (RTC.write(tm)) {
    config = true;
  }
}

// Serial.begin(9600);
while (!Serial) ; // wait for Arduino Serial Monitor
delay(200);
if (parse && config) {
  Serial.print("DS1307 configured Time=");
  Serial.print(__TIME__);
  Serial.print(", Date=");
  Serial.println(__DATE__);
} else if (parse) {
  Serial.println("DS1307 Communication Error :-{");
  Serial.println("Please check your circuitry");
} else {
  Serial.print("Could not parse info from the compiler, Time=\"");
  Serial.print(__TIME__);
  Serial.print("\", Date=\"");
  Serial.print(__DATE__);
  Serial.println("\"");
```

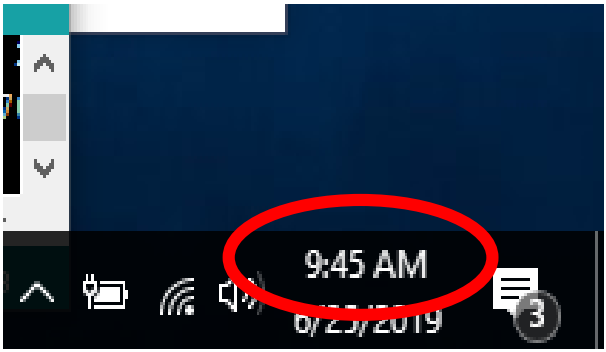
Gambar 4.3 Program *RTC*

Dapat dilihat di Gambar 4.3, ini adalah program untuk mengatur waktu pada *RTC* DS 1307 dengan waktu *real time*. Untuk program pengaturan waktu sendiri berasal dari *library RTC* DS 1307

sendiri. Pengujian Pengaturan waktu ini mencocokan waktu yang sudah di atur dengan program yang berada di Gmbar 4.1 dengan waktu *real time* yang berada pada laptop. Gambar dapat dilihat di Gambar 4.4



Gambar 4.4 Pengujian *RTC* Pada Serial Monitor



Gambar 4.5 Waktu Pada Laptop

Dapat dilihat pada Gambar 4.4 yang dilingkari warna hijau bahwa pengujian *RTC* DS1307 sesuai dengan waktu yang ada atau *real time* yang hasilnya dapat dilihat pada Gambar 4.5 yang dilingkari warna merah. Pada Gambar 4.4 dapat dilihat penulisan

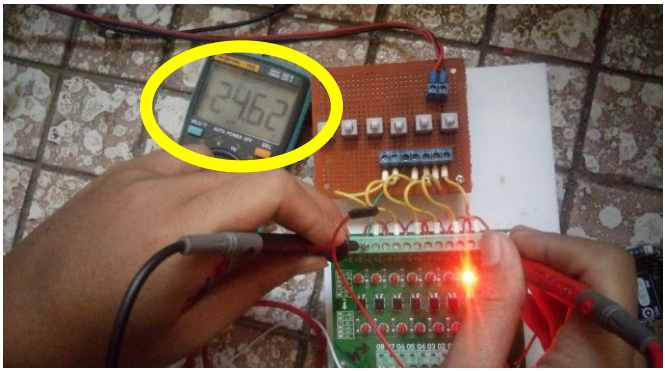
angka telah sesuai. Hasil pengujian *RTC* pada gambar tersebut dapat dilihat bahwa format penulisan sebagai berikut :
09 : Jam
45 : Menit
10 : Detik

Juni : Bulan
25 : Tanggal
2019 : Tahun

Dari pengujian tersebut dapat dibandingkan dengan jam yang tertera pada laptop. Hasilnya akan sama tetapi biasanya terdapat perbedaan sedikit pada "Detik".

4.3 Pengujian Modul *Optocoupler*

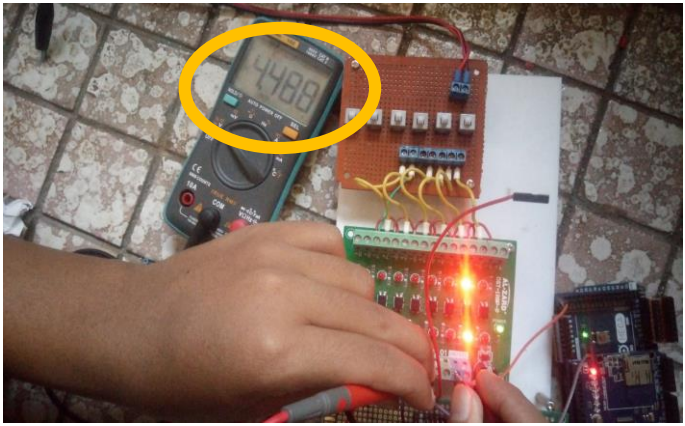
Pengujian Modul *Optocoupler* dilakukan dengan mengukur tegangan *input* yang diberikan tegangan sebesar 24 V oleh *power supply* dan mengukur tegangan *output* sebesar 5 V. Tegangan *output* tersebut digunakan sebagai *input* ke *port Arduino Mega* 2560. Karena *port Arduino* bekerja ketika ada *input* sebesar 5 V. Dapat dilihat pengukuran pada *port input* alat terbaca 24 V. Dapat dilihat pada Gambar 4.6



Gambar 4.6 Pengujian Modul *Optocoupler* Diberi *Input*

Dapat dilihat bahwa LED yang menyala di dekat *port input* memancarkan sinyal cahaya Infra merahnya dengan tegangan 24VDC yang pengukurannya dapat dilihat di lingkaran warna

kuning. Selanjutnya cahaya infra merah yang dipancarkan tersebut akan dideteksi oleh *Phototransistor* yang berada di dekat *port output* dan menyebabkan terjadinya hubungan atau *Switch ON* pada *Phototransistor*. Dan akhirnya *port output* pada modul *Optocoupler* tegangan sama besar dengan tegangan *Arduino* yaitu 5 V. Pengukuran dapat dilihat pada Gambar 4.7



Gambar 4.7 Pengujian Modul *Optocoupler* Ketika Meneruskan Ke *Port Output*

Dapat dilihat Tegangan *port Output* menjadi 5 V yang diberi lingkaran berwana *orange*. *Port* tersebut menjadi 5 V dikarenakan mendapat tegangan sebesar 5 V *Arduino*. Jadi Modul *Optocoupler* me*switch* tegangan dari 24 V yang di dapatkan dari *power supply* ke 5 V yang di dapatkan dari tegangan *Arduino*. Dari percobaan pengukuran di atas dapat di simpulkan modul tersebut tidak dalam keadaan rusak.

4.4 Pengujian Koneksi Router Terhadap Jarak

Pengujian jarak ini memungkinkan pengukuran jarak jauh dan pelaporan informasi kepada PC yang berada di *office*. Untuk mengetahui stabil koneksi terhadap jarak antara PC dan *Router* maka perlu dilakukan pengujian jarak.

4.4.1 Pengujian Koneksi Tanpa Halangan

Untuk mengetahui jarak maksimal yang dapat dicapai dari koneksi *router* ke dalam PC *office* dengan tanpa penghalang, maka dilakukan langkah pengujian dengan meletakkan *Router* dengan tanpa penghalang. Dan selanjutnya melakukan cek koneksi di *command prompt* dengan melakukan ping koneksi dengan nomor IP *Address*. Untuk nomor IP *Address* disini menggunakan nomor IP 192.168.1.3. Hasil pengujian Koneksi tanpa penghalang dapat dilihat di Tabel 4.1

Tabel 4.1 Pengujian *Router* Tanpa Halangan

NO	Jarak (M)	*Loss* (%)
1	5	0%
2	10	0%
3	15	0%
4	20	0%
5	25	0%
6	30	0%
7	35	0%
8	40	25%
9	45	25%
10	50	50%

Di jarak 50 M ini ,sudah sulit untuk menangkap jaringan *Ethernet Shield* sendiri meskipun *loss* nya hanya baru 50% saja . tapi masih memungkinkan untuk mendapatkan sinyal tapi sulit. Jadi untuk tanpa halangan jarak maksimal untuk mendapatkan sinyal yang stabil yaitu sebesar 50 M.

4.4.2 Pengujian Koneksi Dengan Penghalang

Selanjutnya merupakan hasil pengujian dengan halangan. Dalam langkah pengujian yang dilakukan sama dengan pengujian sebelumnya yang tanpa halangan. Yang membedakan yaitu untuk *Router* sendiri di masukan dalam ruangan untuk pengujian dengan penghalang. Data hasil pengujian dengan halangan dapat dilihat di Tabel 4.2

Tabel 4.2 Pengujian *Router* Dengan Penghalang

NO	Jarak (M)	*Loss* (%)
1	5	0%
2	10	0%
3	15	0%
4	20	0%
5	25	0%
6	30	25%
7	35	25%
8	40	25%
9	45	25%
10	50	25%

Dapat dilihat dari table diatas, ketika jarak sudah mencapai 30 M ini *loss* sudah mencapai 25%. Tetapi ketika mencapai 40 M *loss* nya sebesar 25% dan di jarak 40 M sudah sulit untuk menangkap jaringan *Ethernet Shield* sendiri meskipun *loss* nya hanya baru 25% saja . tapi masih memungkinkan untuk mendapatkan sinyal tapi sulit. Jadi untuk tanpa halangan jarak maksimal untuk mendapatkan sinyal yang stabil yaitu sebesar 50 M. Dari pengujian diatas dapat disimpulkan bahwa sinyal *Ethernet Shield* jika dengan penghalang 40 M. Jadi makin sedikit penghalang maka sinyal makin luas juga jangkauan sinyalnya, tapi semakin banyak penghalang maka semakin dikit juga jangkaunnya jaringannya.

4.5 Pengujain Sensor *Proximity* Induktif

Pengujian pertama kali yang dilakukan adalah pengukuran tegangan terhadap *output* sensor *proximity* induktif ketika terkena benda logam atau tidak terkena benda logam dan juga bisa sekaligus diklasifikasikan termasuk Sensor *Proximity* Induktif NC (*Normally Close*) atau Sensor *Proximity* Induktif NO (*Normally Open*). Pengukuran menggunakan *power supply* sebagai sumber sensor. Untuk *power supply* sendiri memakai tegangan sebesar 24 VDC. Pengukuran dapat dilihat pada Gambar 4.8.

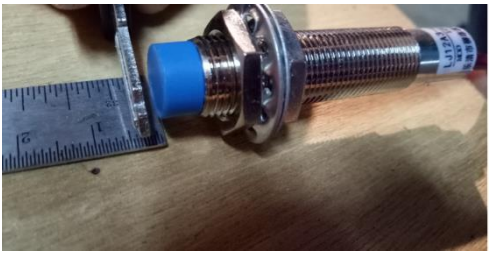
Gambar 4.8 Pengujian Sensor *Proximity* Induktif Ketika Tidak Terkena Benda Logam

Dapat dilihat ketika tidak ada benda logam yang mendekati Sensor *Proximity* Induktif Tegangan sebesar 24 VDC. Pengukuran dapat dilihat di lingakaran warna merah. Dapat di simpulkan bahwa ketika benda logam mengenai Sensor *Proximity* Induktif indikator lampu akan menyala pada Sensors *Proximity* Induktif dan tegangannya sebesar 5 VDC.

Gambar 4.9 Pengujian Sensor *Proximity* Induktif Terkena Benda Logam

Pada Gambar 4.9 dapat kita lihat tegangnnya menjadi 0 V ketika Sensor *Proximity* Induktif mengenai benda logam.

Pengukuran dapat dilihat di lingkaran warna merah Sehingga Sensor *Proximity* Induktif ini termasuk jenis NC (*Normally Close*).



Gambar 4.10 Pengujian Jarak Sensor *Proximity* Induktif

Pada Gambar 4.10 adalah pengujian jarak terhadap *inductive proximity sensor type* LJ12A3-4-Z/BX. *Proximity sensor* tersebut dapat mendeteksi jarak maksimal sebesar 0,4 CM. Sensor tersebut digunakan untuk pendeteksi jika *conveyor* mengalami *downtime* dan jenis *downtime* ringan maka *conveyor* akan berhenti di *home base* yang diberi tanda sensor *proximity* induktif tersebut.

```
1  void setup() {
2    // put your setup code here, to run once:
3  Serial.begin(9600);
4  pinMode (A0, INPUT_PULLUP);
5  }
6
7  void loop() {
8    // put your main code here, to run repeatedly:
9  Serial.println(digitalRead(A0));
.0 delay(400);
.1 }
```

Gambar 4.11 Program Pengambilan Data Jarak *Proximity Sensor*

Pada uji coba ini di Gambar 4.11 adalah program pengujian di *Arduino IDE* untuk mengambil data kemampun jarak *proximity* sensor. Pada pengujian Sensor *Proximity* ini menggunakan *range* maksimal yaitu sebesar 1 cm. Berikut pada Tabel 4.1 adalah data jarak *proximity* sensor *type* LJ12A3-4-Z/BX

Tabel 4.3 Nilai Jarak Sensor *Proximity*

Type Sensor	Jarak (CM)	Keterangan
Proximity sensor Type LJ12A3-4-Z/BX	0,1	Terdeteksi
	0,2	Terdeteksi
	0,3	Terdeteksi
	0,4	Terdeteksi
	0,5	Tidak Terdeteksi
	0,6	Tidak Terdeteksi
	0,7	Tidak Terdeteksi
	0,8	Tidak Terdeteksi
	0,9	Tidak Terdeteksi
	1,0	Tidak Terdeteksi

Pada Tabel 4.3 dapat disimpulkan bahwa *proximity* sensor *type* 18-8DN mampu mendeteksi program pada jarak maksimal 1 cm sementara *proximity* sensor LJ12A3-4-Z/BX mampu mendeteksi program pada jarak maksimal 0,4 CM.

4.6 Pengujian *Web*

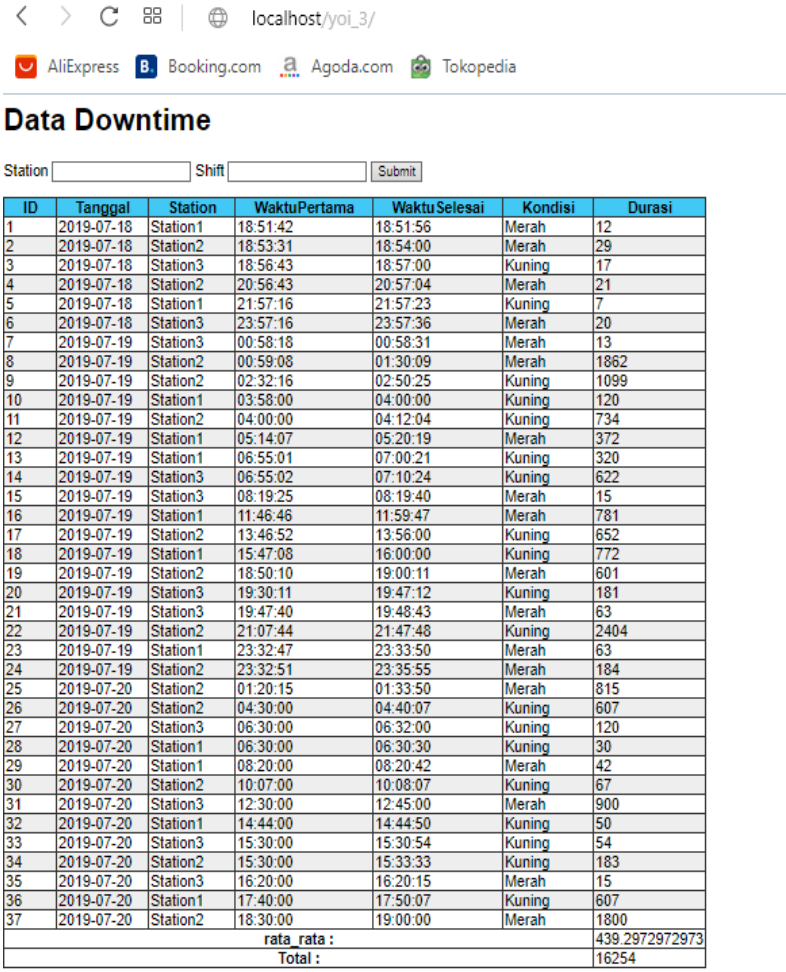
Pengujian *Web* merupakan sebuah transformasi dari perancangan sistem program ke dalam bahasa pemrograman. Dalam Proyek Akhir ini terdapat pengujian tampilan dalam *web*.

4.6.1 Pengujian Tampilan *Web*

Implementasi tampilan *web* dilakukan dengan membuka tiap tiap halaman pada *web* untuk memastikan apakah *web* dapat berjalan sesuai dengan perancangan atau tidak. Berikut merupakan proses tampilan antar muka untuk *web*.

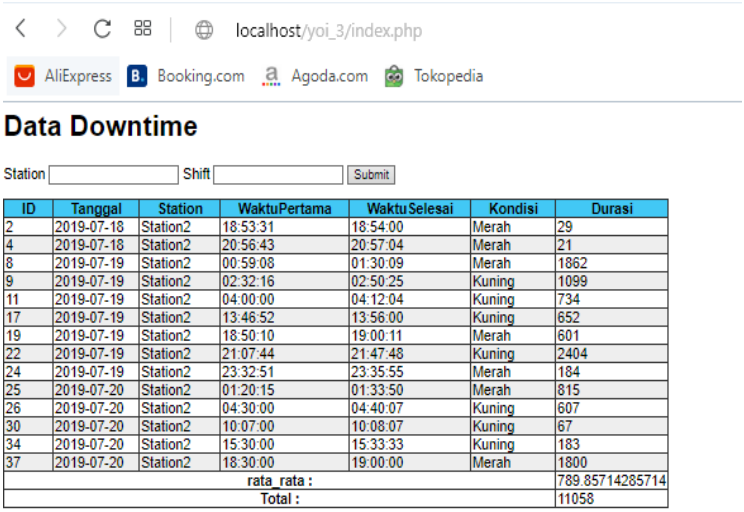
1. Halaman *Logger*

   Halaman *logger* pada *web* system pencatat dan pelaporan *downtime* di area *conveyor* ditunjukkan oleh Gambar 4.11, pada halaman ini *logger* yang sekaligus sebagai halaman home juga, terdapat tabel yang berisikan station yang berfungsi sebagai menu sortir tiap *Station* yang di inginkan. Nantinya akan muncul data total durasi *downtime* yang terjadi baik keseluruhan area *conveyor* maupun tiap *conveyor*.



ID	Tanggal	Station	WaktuPertama	WaktuSelesai	Kondisi	Durasi
1	2019-07-18	Station1	18:51:42	18:51:56	Merah	12
2	2019-07-18	Station2	18:53:31	18:54:00	Merah	29
3	2019-07-18	Station3	18:56:43	18:57:00	Kuning	17
4	2019-07-18	Station2	20:56:43	20:57:04	Merah	21
5	2019-07-18	Station1	21:57:16	21:57:23	Kuning	7
6	2019-07-18	Station3	23:57:16	23:57:36	Merah	20
7	2019-07-19	Station3	00:58:18	00:58:31	Merah	13
8	2019-07-19	Station2	00:59:08	01:30:09	Merah	1862
9	2019-07-19	Station2	02:32:16	02:50:25	Kuning	1099
10	2019-07-19	Station1	03:58:00	04:00:00	Kuning	120
11	2019-07-19	Station2	04:00:00	04:12:04	Kuning	734
12	2019-07-19	Station1	05:14:07	05:20:19	Merah	372
13	2019-07-19	Station1	06:55:01	07:00:21	Kuning	320
14	2019-07-19	Station3	06:55:02	07:10:24	Kuning	622
15	2019-07-19	Station3	08:19:25	08:19:40	Merah	15
16	2019-07-19	Station1	11:46:46	11:59:47	Merah	781
17	2019-07-19	Station2	13:46:52	13:56:00	Kuning	652
18	2019-07-19	Station1	15:47:08	16:00:00	Kuning	772
19	2019-07-19	Station2	18:50:10	19:00:11	Merah	601
20	2019-07-19	Station3	19:30:11	19:47:12	Kuning	181
21	2019-07-19	Station3	19:47:40	19:48:43	Merah	63
22	2019-07-19	Station2	21:07:44	21:47:48	Kuning	2404
23	2019-07-19	Station1	23:32:47	23:33:50	Merah	63
24	2019-07-19	Station2	23:32:51	23:35:55	Merah	184
25	2019-07-20	Station2	01:20:15	01:33:50	Merah	815
26	2019-07-20	Station2	04:30:00	04:40:07	Kuning	607
27	2019-07-20	Station3	06:30:00	06:32:00	Kuning	120
28	2019-07-20	Station1	06:30:00	06:30:30	Kuning	30
29	2019-07-20	Station1	08:20:00	08:20:42	Merah	42
30	2019-07-20	Station2	10:07:00	10:08:07	Kuning	67
31	2019-07-20	Station3	12:30:00	12:45:00	Merah	900
32	2019-07-20	Station1	14:44:00	14:44:50	Kuning	50
33	2019-07-20	Station3	15:30:00	15:30:54	Kuning	54
34	2019-07-20	Station2	15:30:00	15:33:33	Kuning	183
35	2019-07-20	Station3	16:20:00	16:20:15	Merah	15
36	2019-07-20	Station1	17:40:00	17:50:07	Kuning	607
37	2019-07-20	Station2	18:30:00	19:00:00	Merah	1800
rata_rata :						439.2972972973
Total :						16254

Gambar 4.12 Tampilan *Web* Data Keseluruhan



ID	Tanggal	Station	WaktuPertama	WaktuSelesai	Kondisi	Durasi
2	2019-07-18	Station2	18:53:31	18:54:00	Merah	29
4	2019-07-18	Station2	20:56:43	20:57:04	Merah	21
8	2019-07-19	Station2	00:59:08	01:30:09	Merah	1862
9	2019-07-19	Station2	02:32:16	02:50:25	Kuning	1099
11	2019-07-19	Station2	04:00:00	04:12:04	Kuning	734
17	2019-07-19	Station2	13:46:52	13:56:00	Kuning	652
19	2019-07-19	Station2	18:50:10	19:00:11	Merah	601
22	2019-07-19	Station2	21:07:44	21:47:48	Kuning	2404
24	2019-07-19	Station2	23:32:51	23:35:55	Merah	184
25	2019-07-20	Station2	01:20:15	01:33:50	Merah	815
26	2019-07-20	Station2	04:30:00	04:40:07	Kuning	607
30	2019-07-20	Station2	10:07:00	10:08:07	Kuning	67
34	2019-07-20	Station2	15:30:00	15:33:33	Kuning	183
37	2019-07-20	Station2	18:30:00	19:00:00	Merah	1800
rata_rata :						789.85714285714
Total :						11058

Gambar 4.13 Tampilan *Web* Data Station 2

Untuk Gambar 4.12 adalah merupakan contoh tampilan dari fungsi dari mensortir data sesuai dengan jenis *station*, Jadi ketika membuka kotak *station* nantinya akan banyak jenis *station* yang dapat dimonitoring. Selain kotak jenis station, disebelahnya terdapat kotak *Shift* untuk melihat data *downtime shift* pagi dan data *downtime shift* malam, untuk *shift* malam pagi sendiri dimulai pukul 07.30 WIB sampai 17.00 WIB sedangkan untuk *shift* malam dimulai pukul 19.30 WIB sampai 05.00 WIB. Sedangkan di Gambar 4.13 yaitu mencotohkan hanya menampikan data Station 2 dengan total durasi *downtime* sebesar 11058 Detik dan memiliki rata – rata sebesar 789,85 Detik.

4.6.2 Pengujian Aspek F*ungsionalty* DanAspek *Usability*

Pada tahap pengujian terdapat 2 (dua) jenis pengujian yaitu pengujian aspek *fungsionalty* dan aspek *usability.*

1. Pengujian Aspek *Fungsionality*

Pengujian aspek *functionalty* dilakukan kepada 2 pengembang menggunakan kuisioner yang berisi fungsi pada *Web* apakah ada dapat berjalan lancar atau tidak. Hasil pengujian

functionality pada rancangan *interface* sistem monitoring *downtime* ditunjukan pada Tabel 4.4 dan Tabel 4.5 seperti berikut:

Tabel 4.4 Pengujian Aspek *Fungsionality*

No	Kasus Uji	Lolos	
		Ya	Tidak
1	Melihat *Home* dan *Logger*	V	
2	Menyortir Data *Downtime* tiap Station	V	
3	Menjumlahkan Data Total *Downtime*	V	
4	Menjumlahkan Data Total *Downtime* tiap *Station*	V	

Tabel 4.5 Pengujian Aspek Fungsionality

No	Kasus Uji	Lolos	
		Ya	Tidak
1	Melihat *Home* dan *Logger*	V	
2	Menyortir Data *Downtime* tiap Station	V	
3	Menjumlahkan Data Total *Downtime*	V	
4	Menjumlahkan Data Total *Downtime* tiap *Station*	V	

Berdasarkan hasil pengujian aspek *functionalty* dari dua pengembang menyatakan semuanya lolos sehingga dapat diketahui persentase untuk masing masing penilaian adalah sebagai berikut:

Total Kasus uji = 4 x 2 = 8

Ya $= \frac{8}{8}$ x 100% = 100%

Tidak $= \frac{0}{8}$ x 100% = 0%

2. Pengujian Aspek *Usability*

Pengujian Aspek *Ussability* dilakukan kepada 4 karyawan PT. Jatim Autocomp Indonsia dan 1 mahasiswa dengan menggunakan kuisioner yang berisi lima pertanyaan. Hasil pengujian *Ussability* rancangan *interface* sistem monitoring ditunjukan pada Tabel 4.6 – Tabel 4.10, hasil tabel dapat dlihat tabel dibawah ini sebagai berikut:

Tabel 4.6 Pengujian Aspek *Usability*

No	Pernyataan	Setuju	
		Ya	Tidak
1	Secara keseluruhan saya merasa puas dengan *Web* ini		V
2	Cara menggunakan *Web* ini mudah	V	
3	Mudah menemukan Data *Downtime* tiap *Station*	V	
4	Saya suka menggunakan *Web* ini		V
5	Informasi pada *Web* cukup jelas	V	

Tabel 4.7 Pengujian Aspek *Usability*

No	Pernyataan	Setuju	
		Ya	Tidak
1	Secara keseluruhan saya merasa puas dengan *Web* ini		V
2	Cara menggunakan *Web* ini mudah	V	
3	Mudah menemukan Data *Downtime* tiap *Station*	V	
4	Saya suka menggunakan *Web* ini	V	
5	Informasi pada *Web* cukup jelas	V	

Tabel 4.8 Pengujian Aspek *Usability*

No	Pernyataan	Setuju	
		Ya	Tidak
1	Secara keseluruhan saya merasa puas dengan *Web* ini		V
2	Cara menggunakan *Web* ini mudah	V	
3	Mudah menemukan Data *Downtime* tiap *Station*	V	
4	Saya suka menggunakan *Web* ini		V
5	Informasi pada *Web* cukup jelas	V	

Tabel 4.9 Pengujian Aspek *Usability*

No	Pernyataan	Setuju	
		Ya	Tidak
1	Secara keseluruhan saya merasa puas dengan *Web* ini	V	
2	Cara menggunakan *Web* ini mudah	V	
3	Mudah menemukan Data *Downtime* tiap *Station*	V	
4	Saya suka menggunakan *Web* ini	V	
5	Informasi pada *Web* cukup jelas	V	

Tabel 4.10 Pengujian Aspek *Usability*

No	Pernyataan	Setuju	
		Ya	Tidak
1	Secara keseluruhan saya merasa puas dengan *Web* ini		V
2	Cara menggunakan *Web* ini mudah	V	

No	Pernyataan	Setuju	
		Ya	Tidak
3	Mudah menemukan Data *Downtime* tiap *Station*	V	
4	Saya suka menggunakan *Web* ini	V	
5	Informasi pada *Web* cukup jelas	V	

Berdasarkan hasil pengujian aspek *usability* dari 4 karyawan PT. Jatim Autocomp Indonesia dan 1 mahasiswa, hanya 1 mahasiswa yang menyatakan bahwa menjawab setuju semua. 2 karyawana lainnya menjawab dengan hanya 4 setuju semua, dan 2 lainnya yang terkahir menjawab dengan 3 setuju semua. dengan itu dapat kita ketahui persentase persentase untuk masing - masing penilaian adalah sebagai berikut:

Total Kasus uji = 5 x 5 = 25

Ya = $\frac{19}{25}$ x 100% = 76%

Tidak = $\frac{6}{25}$ x 100% = 24%

4.7 Pengujian Keseluruhan

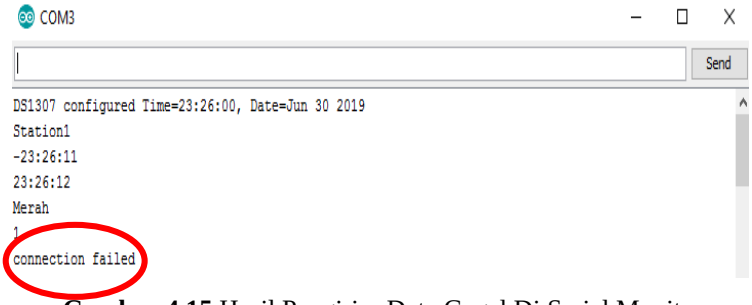
Pengujian keseluruhan bertujuan untuk mengetahui *downtime* pada area *conveyor* di PT. Jatim Autocomp Indonesia. Pengujian dilakukan menggunakan sumber dari panel *relay* dan disambungkan ke panel alat *downtime* sebagai *input* tiap tombol per *conveyor*. Dan tiap *relay* tersambung hanya 1 *port Arduino*. Jadi untuk membuat sistem monitoring pada area tersebut dengan terdapat 26 *Conveyor*. Membutuhkan 52 kabel yang terdiri dari 13 kabel merah dan 13 kabel kuning.

Di tiap *conveyor* disediakan 2 tombol untuk sebagai penanda *downtime*, yakni tombol merah dan tombol kuning. Nanti hasilnya akan di tampilkan di serial monitor, tampilannya terdiri dari tanggal, Jenis *station*, waktu pertama pada saat *downtime*, waktu selesai *downtime*, kondisi dan durasi *downtime*. Selanjutnya pengujian keseluruhan ini menggunakan alat *prototype* ini dengan 6

tombol. 6 tombol tersebut mewakili 3 *station*. Jadi terdapat 1 tombol merah *Station* 1 dan 1 tombol kuning *Station* 1, 1 Tombol Merah *Station* 2 dan Tombol kuning *Station* 2, yang terakhir yaitu 1 Tombol Merah *Station* 3 dan Tombol kuning *Station* 3.

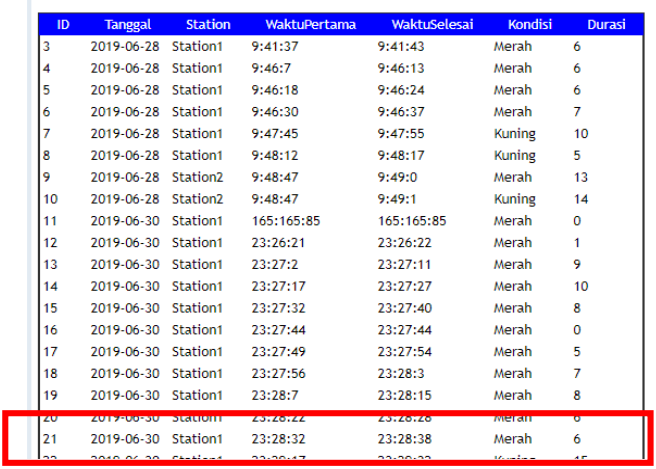
Gambar 4.14 Hasil Pengirim Data Di Serial Monitor Gambar

Dapat kita lihat pada Gambar 4.14 adalah hasil pengirim data yang di tampilkan di serial monitor *software Arduino IDE*. Kemudian di Gambar 4.14 ada tulisan "*Done Send*" yang dilingkari dengan garis hitam, itu menandakan sebagai indikasi bahwa data tersebut telah terkirim ke *Web*. Tetapi jika terdapat tulisan "*connection failed*" pada serial monitor yang ditunjukkan dengan garis merah, menandakan bahwa data tersebut tidak terkirim ke *Web*. Biasanya data tidak terkirim dikarenakan tidak tersambung dengan jaringan internet. Untuk tampilannnya dapat dilihat di Gambar 4.15 di bawah ini.



Gambar 4.15 Hasil Pengirim Data Gagal Di Serial Monitor

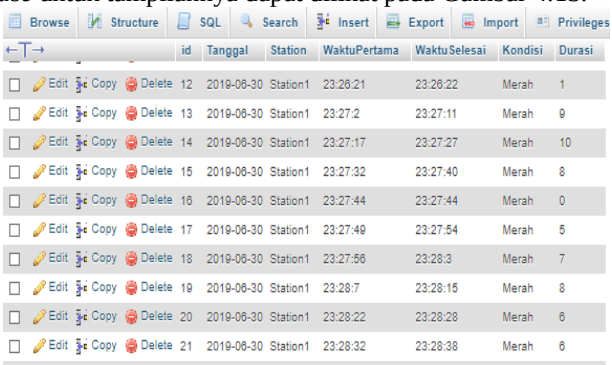
Setelah data terkirim di serial monitor pada *software Arduino IDE* dan ada indikator jika data terkirim "*Done Send*". Kemudian data ditampilkan di *Web*. Tampilan dapat dilihat di Gambar 4.16.



ID	Tanggal	Station	WaktuPertama	WaktuSelesai	Kondisi	Durasi
3	2019-06-28	Station1	9:41:37	9:41:43	Merah	6
4	2019-06-28	Station1	9:46:7	9:46:13	Merah	6
5	2019-06-28	Station1	9:46:18	9:46:24	Merah	6
6	2019-06-28	Station1	9:46:30	9:46:37	Merah	7
7	2019-06-28	Station1	9:47:45	9:47:55	Kuning	10
8	2019-06-28	Station1	9:48:12	9:48:17	Kuning	5
9	2019-06-28	Station2	9:48:47	9:49:0	Merah	13
10	2019-06-28	Station2	9:48:47	9:49:1	Kuning	14
11	2019-06-30	Station1	165:165:85	165:165:85	Merah	0
12	2019-06-30	Station1	23:26:21	23:26:22	Merah	1
13	2019-06-30	Station1	23:27:2	23:27:11	Merah	9
14	2019-06-30	Station1	23:27:17	23:27:27	Merah	10
15	2019-06-30	Station1	23:27:32	23:27:40	Merah	8
16	2019-06-30	Station1	23:27:44	23:27:44	Merah	0
17	2019-06-30	Station1	23:27:49	23:27:54	Merah	5
18	2019-06-30	Station1	23:27:56	23:28:3	Merah	7
19	2019-06-30	Station1	23:28:7	23:28:15	Merah	8
20	2019-06-30	Station1	23:28:22	23:28:28	Merah	6
21	2019-06-30	Station1	23:28:32	23:28:38	Merah	6

Gambar 4.16 Tampilan Data Terkirim Di *Web*

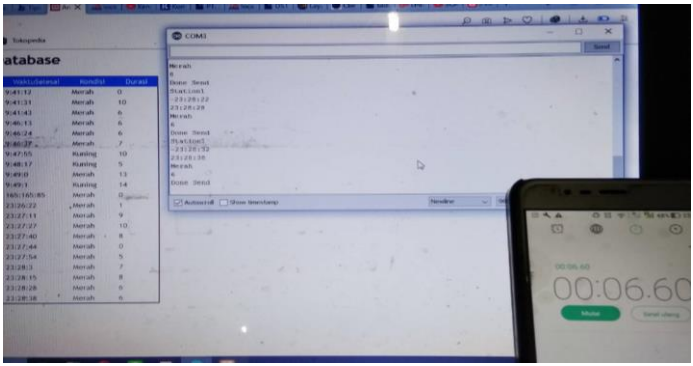
Dapat dilihat data yang sebelumnya ditampilkan di Serial Monitor pada *Sofwatre Arduino IDE* di Gambar 4.14. Selanjutnya ditampilkan di halaman *Web*. Untuk tampilannya seperti di Gambar 4.16. Selain ditampilkan di halaman 4.17 data juga tersimpan di *database* untuk tampilannya dapat dilihat pada Gambar 4.13.



Browse | Structure | SQL | Search | Insert | Export | Import | Privileges

	id	Tanggal	Station	WaktuPertama	WaktuSelesai	Kondisi	Durasi
Edit Copy Delete	12	2019-06-30	Station1	23:26:21	23:26:22	Merah	1
Edit Copy Delete	13	2019-06-30	Station1	23:27:2	23:27:11	Merah	9
Edit Copy Delete	14	2019-06-30	Station1	23:27:17	23:27:27	Merah	10
Edit Copy Delete	15	2019-06-30	Station1	23:27:32	23:27:40	Merah	8
Edit Copy Delete	16	2019-06-30	Station1	23:27:44	23:27:44	Merah	0
Edit Copy Delete	17	2019-06-30	Station1	23:27:49	23:27:54	Merah	5
Edit Copy Delete	18	2019-06-30	Station1	23:27:56	23:28:3	Merah	7
Edit Copy Delete	19	2019-06-30	Station1	23:28:7	23:28:15	Merah	8
Edit Copy Delete	20	2019-06-30	Station1	23:28:22	23:28:28	Merah	6
Edit Copy Delete	21	2019-06-30	Station1	23:28:32	23:28:38	Merah	6

Gambar 4.17 Tampilan Di *Database*



Gambar 4.18 Tampilan Pengujian Keseluruhan

Pada Gambar 4.14 dapat kilta lihat tampilan pengujian keseluruhan menggunakan *stopwatch* dan data yang di tampilkan sama dengan data yang ditampilkan pada Sistem Pencatat dan Pelaporan *Downtime* yaitu *Downtime* pada *Station* / *Conveyor* 1 selama 6 Detik pada jam 23:38:32 sampai 23:28:38 dengan kondisi merah atau kondisi kerusakan parah. Untuk data serial monitor, tampilan pada *web*, dan tampilan pada *database* sendiri bias dilihat di Gambar sebelumnya yatiu Gambar 4.11, 4.12, dan 4.13.

Selanjutnya untuk Untuk Pengujan Alat Pengujan Pencatatan dan Pelaporan *Downtime* sendiri yang berada pada pabrik. Ini adalah hasil data dari beberapa *downtime* yang di ambil pada tanggal 23 – 26 Januari 2019. Pada tabel 4.1 dan Table 4.2

Tabel 4.11 Monitoring *Downtime Conveyor* 7

No	Tanggal	Station	Waktu Pertama	Waktu Selesai	Kondisi	Durasi (Detik)
1	23/1/2019	Station 7	23:0:10	23:3:52	Merah	222
2	24/1/2019	Station 7	4:6:32	4:9:4	Merah	152
3	24/1/2019	Station 7	4:28:37	4:29:8	Kuning	31
4	24/1/2019	Station 7	4:29:9	4:31:18	Merah	129
No	**Tanggal**	**Station**	**Waktu**	**Waktu**	**Kondisi**	**Durasi**

			Pertama	Selesai		(Detik)
5	24/1/2019	Station 7	22:38:29	22:39:42	Kuning	54
TOTAL						**588**

Tabel 4.12 Monitoring *Downtime Conveyor* 20

No	Tanggal	Station	Waktu Pertama	Waktu Selesai	Kondisi	Durasi (Detik)
1	23/1/2019	Station 7	17:17:0	17:50:21	Merah	201
2	23/1/2019	Station 7	18:32:0	18:42:29	Merah	629
3	24/1/2019	Station 7	2:21:0	2:52:25	Merah	1885
4	24/1/2019	Station 7	6:8:0	6:29:26	Merah	1286
5	24/1/2019	Station 7	6:48:0	6:49:23	Merah	83
TOTAL						**4084**

Dapat dilihat dari data tabel 4.1 dan tabel 4.2 di atas, tabel diatas merupakan data monitoring *downtime* yang terjadi pada 2 *conveyor* yaitu *conveyor* 22 dan *conveyor* 25. Di tabel itu ada 6 kolom tabel yang memberikan informasi, kolom pertama dimulai dari tanggal untuk menunjukkan kapan *conveyor* mengalami *downtime*, kolom kedua yaitu karena terlalu banyak *conveyor* yang dipakai pada area *conveyor* tersebut maka perlunya Indikator atau nama pada tiap *conveyor*, jadi kolom nama yaitu nama *conveyor* yang sedang mengalami *downtime*, untuk kolom yang ketiga yaitu kolom waktu pertama yaitu kolom waktu dimana *conveyor* mengalami *downtime* , kolom keempat yaitu kolom waktu dimana *conveyor* telah selesai mengalami *downtime* dan siap untuk bekerja kembali, kolom kelima yaitu kolom Kondisi yaitu dikatakan "MERAH" jika :

- Ditekan Tombol Merah
- Mengalami *Downtime* yang parah
- *Downtime* yang parah itu seperti :
    a. Kesalahan dalam memasukan *connector* ke kabel

b. Kesalahan dalam merangkai *wire harness*
c. Putusnya rantai pada *conveyor*
d. Kerusakan pada PLC utama.

Lalu dikatakan dalam kondisi "KUNING" jika:

- Ditekan Tombol Kuning
- Mengalami *Downtime* yang ringan
- *Downtime* yang ringan itu seperti :
    a. Kewalahan pada saaat proses produksi
    b. Seabagai penanda terkahir produksi sebelum pekerja melakukan istirahat

Untuk Kolom terkahir yaitu Durasi yang menampilkan waktu berapa lama *conveyor* itu mengalami *downtime*. Untuk penunjukkan waktu sendiri dalam satuan Detik atau *second*. Dapat disimpulkan dari data tabel di atas *conveyor* 25 lebih membutuhkan perhatian khusus dikarenakan data sampel hasil *downtime* pada tanggal 24 -25 Januari 2019 sebanyak 1400 Detik. lebih banyak dari pada *downtime* yang dimiliki *conveyor* 22 yang hanya 800 Detik.

Tabel 4.13 Pengujian Lama Pengiriman Data

NO	Pengujian	Lama Pengiriman
1	Pengujian 1	5 Detik
2	Pengujian 2	4.9 Detik
3	Pengujian 3	4,8 Detik
4	Pengujian 4	4,8 Detik
5	Pengujian 5	5 Detik
6	Pengujian 6	5 Detik
7	Pengujian 7	4,9 Detik
8	Pengujian 8	4,8 Detik
9	Pengujian 9	5,1 Detik
10	Pengujian 10	4, 9 Detik

Dari tabel 4.13 merupakan percobaan lama pengiriman data waktu *downtime*, dari 10 pengujian terdapat *delay* dalam pengiriman dari *hardware* ke *web* untuk rata – rata waktu dalam pengiriman data *downtime* yaitu sekitar 4,92 Detik dalam mengirim data wakt*u downtime*. Hal itu disebabkan karena di program web diberi *delay* sebanyak 5 Detik dan juga jika jarak terlalu jauh dengan *router* bisa lebih dari 5 Detik. Dengan adanya monitoring seperti ini sangat

membantu untuk mengetahui *conveyor* mana yang perlu perhatian dan perawatan khusus dan mengetahui untung ruginya selama proses produksi bagi PT. Jatim Autocomp Indonesia. Juga bisa lebih cepat mencatat waktu *dowmtime* yang biasanya dicatat tiap 2 jam sekali tapi dengan adanya alat ini bisa mencatat dengan kecepatan 5 Detik.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari tahapan secara keseluruhan yang sudah dilaksanakan pada penyusunan Proyek Akhir ini mulai dari studi literatur, perancangan dan pembuatan sampai dengan pengujian alat, maka dapat diperoleh kesimpulan bahwa:

1. Sistem Monitoring dan Pelaporan *Downtime* Pada Area *Conveyor* di PT. Jatim Autocomp Indonesia bekerja sesuai dengan fungsinya yaitu dapat menampilkan waktu awal terjadinya *downtime* dan durasi waktu lamanya *downtime* pada masing – masing *conveyor*.
2. Dari hasil pengujian data jarak maksimal *Ethernet Shield* jika tanpa penghalang 50 meter dan jika dengan penghalang 40 meter
3. Dangan adanya Sistem Monitoring dan Pelaporan *Downtime* Pada Area *Conveyor* di PT. Jatim Autocomp Indonesia dalam mencatat waktu *downtime* lebih cepat dari biasanya, yang biasanya mencatat tiap 2 jam sekali dalam laporan. Tetapi dengan sistem ini mampu mengirim data dengan lama waktu pengiriman yaitu 5 detik.

5.2 Saran

Untuk lebih memperbaiki dan menyempurnakan kinerja dari alat ini, maka perlu disarankan antara lain:

1. *Database* dapat dikembangkan lagi menjadi berbagai fitur agar mempermudah pekerja dengan menambahkan tombol print / cetak.
2. Perlu membuat sistem login lebih mudah dan data agar lebih aman.
3. Perlu dikembangkan di bagian *web* menunya agar dapat dilihat harian / maupun bulanan.

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

DAFTAR PUSTAKA

[1] Desiani, "Aplikasi Sensor Proximity Pada Lengan Robot Sebagai Penyortir Kotak Berdasarkan Ukuran Berbasis Arduino Uno", **Tugas Akhir**, Teknik Elektronika, Politeknik Negeri Sriwijaya,Palembang, 2015

[2] A. Muhyidin, Monitoring Dan Pelaporan Kehadiran Siswa, **Tugas Akhir,** Departemen Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya, 2018

[3] A. Haris, " Pemantau Isi Kulkas Menggunakan Ethernet Shield R3 Berbasis Arduino Uno R3", **Tugas Akhir,** Teknik Elektro Universitas Negeri Yogyakarta, Yogyakarta, 16

[4] F. D. KPetruzella, "**Elektronika Industri**", Yogyakarta: Andi, 1996

[5] H. R. &. W. Y. Hendy H., "Analisis Dan Perancangan Jaringan Dan Perhitungan Pemakaian Akses Internet Pada PT. BONET UTAMA",**Tugas Akhir**, Teknik Informatika, Universitas Bina Nusantara, Jakarta, 2006

[6] A. Kadir, "**Belajar Database Menggunakan MySQL**", Yogyakarta: Andi, 2008

[7] Hartono, "Perancangan Sistem Data Logger Temperatur Baterai Berbasis Arduino Deumilanove", **Tugas Akhir**, Teknik Elektro, Universitas Jember, Jember, 2013

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

LAMPIRAN A

PROGRAM

A.1 Program Arduino

```
#include <SPI.h>
#include <Ethernet.h>
#include <Wire.h>
#include <TimeLib.h>
#include <DS1307RTC.h>
#include "RTClib.h"


byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; //
disetiap ipdaress itu komponen memiliki ip sndri untuk pengenalan
byte ip[] = {192, 168, 1, 18 }; //Enter the IP of ethernet shield
byte serv[] = {192,168,1,3 } ; //Enter the IPv4 address;
EthernetClient cliente;
RTC_DS1307 rtc;

char daysOfTheWeek[7][12] = {"Sunday", "Monday",
"Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};
int jam1, jam2, menit1, menit2, detik1, detik2, hasiljam ,
hasilmenit , hasildetik, mem, hasil;


const int jmlTombol=6;
byte arrayPin[jmlTombol]={A1,A2,A3,A4, A8,A9};
int state[jmlTombol];

 //28=4 A2 =2 A1=3 26=1
 //26=0 28=3 A1=2 A2=1
 //s1  merah 26 array 3
// s4 h 28 2
 //s2 k a2 0
 //s3 abu A1 1

//////////////////////////Fungsi  SET TIME RTC//////////////////////////
```

```
const char *monthName[12] = {
  "Jan", "Feb", "Mar", "Apr", "May", "Jun",
  "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"
};

tmElements_t tm;

bool getTime(const char *str)
{
  int Hour, Min, Sec;

  if (sscanf(str, "%d:%d:%d", &Hour, &Min, &Sec) != 3) return
false;
  tm.Hour = Hour;
  tm.Minute = Min;
  tm.Second = Sec;
  return true;
}

bool getDate(const char *str)
{
  char Month[12];
  int Day, Year;
  uint8_t monthIndex;

  if (sscanf(str, "%s %d %d", Month, &Day, &Year) != 3) return
false;
  for (monthIndex = 0; monthIndex < 12; monthIndex++) {
    if (strcmp(Month, monthName[monthIndex]) == 0) break;
  }
  if (monthIndex >= 12) return false;
  tm.Day = Day;
  tm.Month = monthIndex + 1;
  tm.Year = CalendarYrToTm(Year);
  return true;
}

//////////////////////////////////////////////////
```

```
          void utama(int tombolKe, String warnaTombol, String
stationKe){
            DateTime now = rtc.now();
        // TOMBOL MERAH STATION 1
          if ((digitalRead(arrayPin[tombolKe]) == HIGH) &&
(state[tombolKe] == 0))
          {
            jam1  = now.hour();
            menit1 = now.minute();
            detik1 = now.second();
            state[tombolKe] = 1 ;
          }

          if ((digitalRead(arrayPin[tombolKe]) == LOW) &&
(state[tombolKe] == 1))
           {
           Serial.println(stationKe);
           Serial.print("-");
           Serial.print(jam1);
           Serial.print(':');
           Serial.print(menit1);
           Serial.print(':');
           Serial.println(detik1);
           jam2  = now.hour();
           Serial.print(jam2);
           Serial.print(':');
           menit2 = now.minute();
           Serial.print(menit2);
           Serial.print(':');
           detik2 = now.second();
           Serial.println(detik2);
           Serial.println(warnaTombol);

           if ( menit2 >= menit1 && detik2 >= detik1 )
           {
            hasiljam = jam2 - jam1;
            hasilmenit = menit2 - menit1;
            hasildetik = detik2 - detik1;
            hasil = ((hasiljam*60) + (hasilmenit*60) + hasildetik );
```

```
  Serial.print(hasil);
  Serial.println();
 }

 if (menit2 < menit1 && detik2 >= detik1)
 {
  hasiljam = jam2 - jam1 - 1 ;
  hasilmenit = ( menit2 + 60 ) - menit1;
  hasildetik = detik2 - detik1;
  hasil = ((hasiljam*60) + (hasilmenit*60) + hasildetik );
  Serial.print(hasil);
  Serial.println();
 }
 if (menit2 > menit1 && detik2 < detik1)
 {
  hasiljam = jam2 - jam1  ;
  hasilmenit = menit2 - menit1 - 1;
  hasildetik = (detik2 + 60 ) - detik1;
  hasil = ((hasiljam*60) + (hasilmenit*60) + hasildetik );
  Serial.print(hasil);
  Serial.println();
 }
 if (menit2 < menit1 && detik2 < detik1 )
 {
  hasiljam = jam2 - jam1 - 1 ;
  hasilmenit = ( menit2 + 59 ) - menit1;
  hasildetik = (detik2 + 60 ) - detik1 ;
  hasil = ((hasiljam*60) + (hasilmenit*60) + hasildetik );
  Serial.print(hasil);
  Serial.println();
 }
 state[tombolKe] = 0 ;
 kirim(warnaTombol,stationKe);
 }
 }
////////////////////////////////////////////////////////////////////////////

  void utama2(int tombolKe, String warnaTombol, String
stationKe){
```

```
            DateTime now = rtc.now();
          // TOMBOL KUNING STATION 2
            if      (digitalRead(arrayPin[tombolKe])      ==      HIGH      &&
state[tombolKe] == 0 && digitalRead (A5)== LOW)
            {
              jam1  = now.hour();
              menit1 = now.minute();
              detik1 = now.second();
              state[tombolKe] = 1 ;
             }


            if    (digitalRead(arrayPin[tombolKe])    ==    LOW    &&    state
[tombolKe] == 1)
             {

             Serial.println(stationKe);
             Serial.print("-");
             Serial.print(jam1);
             Serial.print(':');
             Serial.print(menit1);
             Serial.print(':');
             Serial.println(detik1);
             jam2  = now.hour();
             Serial.print(jam2);
             Serial.print(':');
             menit2 = now.minute();
             Serial.print(menit2);
             Serial.print(':');
             detik2 = now.second();
             Serial.println(detik2);
             Serial.println(warnaTombol);

             if ( menit2 >= menit1 && detik2 >= detik1 )
             {
              hasiljam = jam2 - jam1;
              hasilmenit = menit2 - menit1;
              hasildetik = detik2 - detik1;
              hasil = ((hasiljam*60) + (hasilmenit*60) + hasildetik );
```

```
  Serial.print(hasil);
  Serial.println();
 }

 if (menit2 < menit1 && detik2 >= detik1)
 {
  hasiljam = jam2 - jam1 - 1 ;
  hasilmenit = ( menit2 + 60 ) - menit1;
  hasildetik = detik2 - detik1;
  hasil = ((hasiljam*60) + (hasilmenit*60) + hasildetik );
  Serial.print(hasil);
  Serial.println();
 }
 if (menit2 > menit1 && detik2 < detik1)
 {
  hasiljam = jam2 - jam1  ;
  hasilmenit = menit2 - menit1 - 1;
  hasildetik = (detik2 + 60 ) - detik1;
  hasil = ((hasiljam*60) + (hasilmenit*60) + hasildetik );
  Serial.print(hasil);
  Serial.println();
 }
 if (menit2 < menit1 && detik2 < detik1 )
 {
  hasiljam = jam2 - jam1 - 1 ;
  hasilmenit = ( menit2 + 59 ) - menit1;
  hasildetik = (detik2 + 60 ) - detik1 ;
  hasil = ((hasiljam*60) + (hasilmenit*60) + hasildetik );
  Serial.print(hasil);
  Serial.println();
 }
 state[tombolKe] = 0 ;
 kirim(warnaTombol,stationKe);
 }
 }

////////////////////////////////////////////////////////////////////////////
//////////////////MENGIRIM DATA///////////
void kirim (String warnaTombol, String stationKe){
```

```
            while(!cliente.connect(serv, 80)){
             Serial.print('.');}
             Serial.println(' ');
             if (cliente.connected()) { //
             cliente.print("GET /yoi/data.php?"); //
             cliente.print("Station=");
             cliente.print(stationKe);
             cliente.print("&WaktuPertama=");
             String
waktu1Print=String(jam1)+':'+String(menit1)+':'+String(detik1);
             cliente.print(waktu1Print);
             cliente.print("&WaktuSelesai=");
             String
waktu2Print=String(jam2)+':'+String(menit2)+':'+String(detik2);
             cliente.print(waktu2Print);
             cliente.print("&Kondisi=");
             cliente.print(warnaTombol);
             cliente.print("&Durasi=");
             String cetakHasil = String(hasil);
             cliente.println(cetakHasil);
             Serial.println("Done Send");
             cliente.stop(); //Closing the connection

             }
             else {
             // if you didn't get a connection to the server:
             Serial.println("connection failed");
             }

             }

             void setup() {
             Serial.begin(9600); //setting the baud rate at 9600
             Ethernet.begin(mac, ip);
             for (int deklarasi=0; deklarasi<jmlTombol; deklarasi++){
             pinMode(arrayPin[deklarasi], INPUT_PULLUP);}
             pinMode(A5,INPUT);

          //////////////////////RTC Setup//////////////////////////
```

```
  bool parse=false;
  bool config=false;

  // get the date and time the compiler was run
  if (getDate(__DATE__) && getTime(__TIME__)) {
    parse = true;
    // and configure the RTC with this info
    if (RTC.write(tm)) {
      config = true;
    }
  }

 // Serial.begin(9600);
  while (!Serial) ; // wait for Arduino Serial Monitor
  delay(200);
  if (parse && config) {
    Serial.print("DS1307 configured Time=");
    Serial.print(__TIME__);
    Serial.print(", Date=");
    Serial.println(__DATE__);
  } else if (parse) {
    Serial.println("DS1307 Communication Error :-{");
    Serial.println("Please check your circuitry");
  } else {
    Serial.print("Could not parse info from the compiler, Time=\"");
    Serial.print(__TIME__);
    Serial.print("\", Date=\"");
    Serial.print(__DATE__);
    Serial.println("\"");
  }
 ////////////////////////////////////EndRTCSetup//////////////////////

    }
  //kebaca Arry 0 tercetak 1
    void loop(){
  utama(0,String("Merah"),String("Station1"));
  utama2(1, String("Kuning"), String("Station1"));
  utama(2, String("Merah"), String("Station2"));
```

```
  utama2(3, String("Kuning"), String("Station2"));
  utama(4, String("Merah"), String("Station3"));
  utama2(5, String("Kuning"), String("Station3"));

}
```

A.2 Program Web

```
<?php
  $url=$_SERVER['REQUEST_URI'];
  header("Refresh: 5; URL=$url"); // Refresh the webpage every
5 seconds
  ?>
<html>
<head>
<title>Arduino Ethernet Database</title>
<style type="text/css">
  .table_titles {
    /* position: absolute; */
    padding-right: 20px;
    padding-left: 20px;
    color: #000;
    background-color: #42c8f5;
  }
  table, th, td {
    border: 2px solid #333;
    border-collapse: collapse;
  }
  body {
    font-family: 'Nunito', sans-serif;
  }
  table tbody tr:nth-child(even) {
    background-color: #eee;
  }
  table tbody tr:nth-child(odd) {
    background-color: #fff;
  }
</style>
</head>
<body>
<h1>Data Downtime</h1>
<form action="index.php" method="get" id="main-form">
  <label for="station">Station</label>
  <input list="station" name="station">
```

```
            <datalist id="station">
            <option value="All Station">
            <?php
               for ($i=1; $i <=26 ; $i++) {
                  echo "<option value='Station".$i."'>";
               }
            ?>
            </datalist>

            <label for="shift">Shift</label>
            <input list="shift" name="shift">
            <datalist id="shift">
               <option value="Pagi"></option>
               <option value="Malam"></option>
            </datalist>
            <input type="submit" value="Submit">
         </form>
         <table>
         <thead>
            <tr>
               <th class="table_titles">ID</th>
               <th class="table_titles">Tanggal</th>
               <th class="table_titles">Station</th>
               <th class="table_titles">WaktuPertama</th>
               <th class="table_titles">WaktuSelesai</th>
               <th class="table_titles">Kondisi</th>
               <th class="table_titles">Durasi</th>
            </tr>
         </thead>
         <tbody>
         <?php
            include('connection.php');
            $query = "SELECT * FROM data";
            if (isset($_GET['shift']) && $_GET['shift'] != "" &&
isset($_GET['station']) && $_GET['station'] != ""){
               // echo "SET ALL";
               $shift = $_GET['shift'];
               $station = $_GET['station'];
               if ($station == "All Station"){
```

```
            $query = "SELECT * FROM data WHERE ";
        } else {
            $query = "SELECT * FROM data WHERE
Station='".$station."' AND ";
        }

        if ($shift == "Pagi"){
            $query .= "WaktuPertama >= '07:30:00' AND
WaktuPertama < '19:30:00'";
        } else {
            $query .= "(WaktuPertama < '07:30:00' OR
WaktuPertama >= '19:30:00')";
        }
    } else if (isset($_GET['shift']) && $_GET['shift'] != ""){
        // echo "SET SHIFT";
        $shift = $_GET['shift'];
        $query = "SELECT * FROM data WHERE ";
        if ($shift == "Pagi"){
            $query .= "WaktuPertama >= '07:30:00' AND
WaktuPertama < '19:30:00'";
        } else {
            $query .= "WaktuPertama < '07:30:00' OR WaktuPertama
>= '19:30:00'";
        }
    } else if (isset($_GET['station']) && $_GET['station'] != ""){
        // echo "SET STATION";
        $station = $_GET['station'];
        if ($station != "All Station"){
            $query = "SELECT * FROM data WHERE
Station='".$station."'";
        }
        // $query = "SELECT * FROM data WHERE
Station='".$station."'";
    }
    $result = mysqli_query($con,$query);
    // Process every record
    $total = 0;
    while($row = mysqli_fetch_array($result)){
        echo "<tr>";
```

```
        echo "<td>" . $row['id'] . "</td>";
        echo "<td>" . $row['Tanggal'] . "</td>";
        echo "<td>" . $row['Station'] . "</td>";
        echo "<td>" . $row['WaktuPertama'] . "</td>";
        echo "<td>" . $row['WaktuSelesai'] . "</td>";
        echo "<td>" . $row['Kondisi'] . "</td>";
        echo "<td>" . $row['Durasi'] . "</td>";
        echo "</tr>";
        $total += $row['Durasi'];
    }
    // Close the connection
    mysqli_close($con);
?>
</tbody>
<tfoot>
    <tr>
      <th id="total" colspan="6">Total :</th>
      <td><?php echo $total;?></td>
    </tr>
</tfoot>
</table>
</body>
</html>
```

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

LAMPIRAN B

DATASHEET

1. *Datasheet Arduino Mega*

CE

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

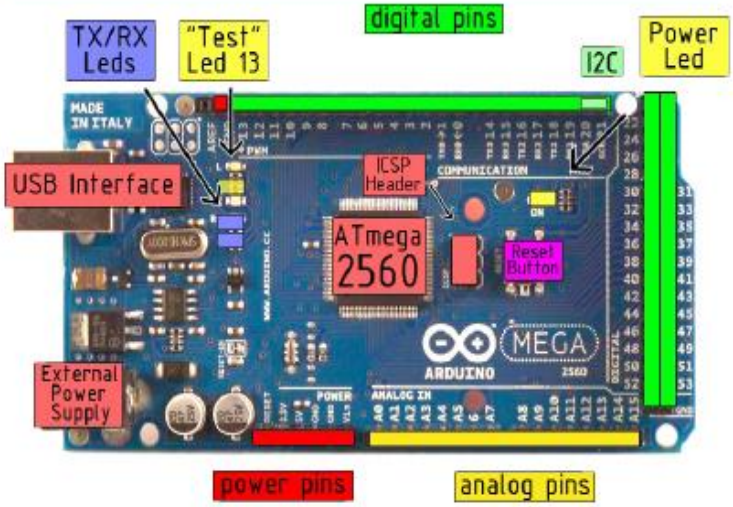
Technical Specification

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.
- **I²C: 20 (SDA) and 21 (SCL).** Support I²C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I²C 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 I²C 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

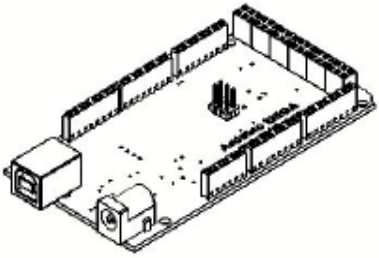
Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

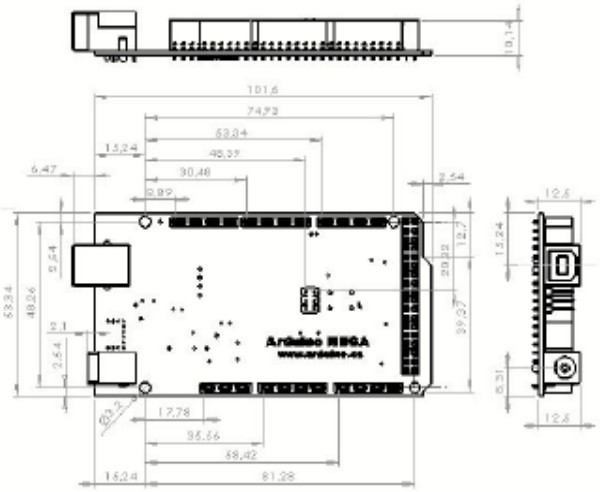
File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink

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.




101,6
74,93
63,84
15,24
48,37
30,48
6,47
2,89
2,54
12,7
20,32
39,37
53,34
48,26
2,54
5,1
2,54
5,31
12,8
12,8
17,78
35,56
58,42
15,24
81,28
Ø3,2
Arduino MEGA
www.arduino.cc

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 warranty period.

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.

2. Datasheet Optocoupler TLP250

TOSHIBA TLP250

TOSHIBA Photocoupler GaAlAs Ired & Photo-IC

TLP250

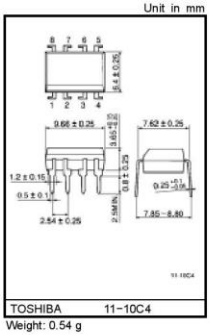
Transistor Inverter
Inverter For Air Conditionor
IGBT Gate Drive
Power MOS FET Gate Drive

The TOSHIBA TLP250 consists of a GaAlAs light emitting diode and a integrated photodetector.
This unit is 8-lead DIP package.
TLP250 is suitable for gate driving circuit of IGBT or power MOS FET.

- Input threshold current: I_F=5mA(max.)
- Supply current (I_{CC}): 11mA(max.)
- Supply voltage (V_{CC}): 10~35V
- Output current (I_O): ±1.5A (max.)
- Switching time (t_{pLH}/t_{pHL}): 1.5μs(max.)
- Isolation voltage: 2500V_{rms}(min.)
- UL recognized: UL1577, file No.E67349
- Option (D4) type
  VDE approved: DIN VDE0884/06.92,certificate No.76823
  Maximum operating insulation voltage: 630V_{PK}
  Highest permissible over voltage: 4000V_{PK}

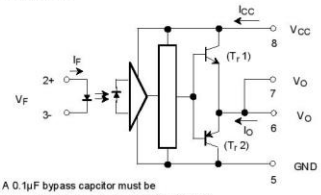
  (Note) When a VDE0884 approved type is needed, please designate the "option (D4)"
- Creepage distance: 6.4mm(min.)
  Clearance: 6.4mm(min.)

Unit in mm



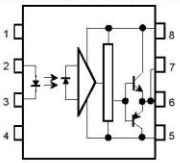
TOSHIBA 11-10C4
Weight: 0.54 g

Schmatic



A 0.1μF bypass capcitor must be connected between pin 8 and 5 (See Note 5).

Pin Configuration (top view)



1 : N.C.
2 : Anode
3 : Cathode
4 : N.C.
5 : GND
6 : V_O (Output)
7 : V_O
8 : V_{CC}

Truth Table

		Tr1	Tr2
Input LED	On	On	Off
	Off	Off	On

Absolute Maximum Ratings (Ta = 25°C)

Characteristic				Symbol	Rating	Unit
LED	Forward current			I_F	20	mA
	Forward current derating (Ta ≥ 70°C)			ΔI_F / ΔTa	−0.36	mA / °C
	Peak transient forward curent		(Note 1)	I_{FPT}	1	A
	Reverse voltage			V_R	5	V
	Junction temperature			Tj	125	°C
Detector	"H"peak output current (P_W ≤ 2.5μs,f ≤ 15kHz)		(Note 2)	I_{OPH}	−1.5	A
	"L"peak output current (P_W ≤ 2.5μs,f ≤ 15kHz)		(Note 2)	I_{OPL}	+1.5	A
	Output voltage	(Ta ≤ 70°C)		V_O	35	V
		(Ta = 85°C)			24	
	Supply voltage	(Ta ≤ 70°C)		V_{CC}	35	V
		(Ta = 85°C)			24	
	Output voltage derating (Ta ≥ 70°C)			ΔV_O / ΔTa	−0.73	V / °C
	Supply voltage derating (Ta ≥ 70°C)			ΔV_{CC} / ΔTa	−0.73	V / °C
	Junction temperature			Tj	125	°C
Operating frequency			(Note 3)	f	25	kHz
Operating temperature range				T_{opr}	−20~85	°C
Storage temperature range				T_{stg}	−55~125	°C
Lead soldering temperature (10 s)			(Note 4)	T_{sol}	260	°C
Isolation voltage (AC, 1 min., R.H.≤ 60%)			(Note 5)	BV_S	2500	Vrms

Note 1: Pulse width P_W ≤ 1μs, 300pps

Note 2: Exporenential wavefom

Note 3: Exporenential wavefom, I_{OPH} ≤ −1.0A(≤ 2.5μs), I_{OPL} ≤ +1.0A(≤ 2.5μs)

Note 4: It is 2 mm or more from a lead root.

Note 5: Device considerd a two terminal device: Pins 1, 2, 3 and 4 shorted together, and pins 5, 6, 7 and 8 shorted together.

Note 6: A ceramic capacitor(0.1μF) should be connected from pin 8 to pin 5 to stabilize the operation of the high gain linear amplifier. Failure to provide the bypassing may impair the switching proparty. The total lead length between capacitor and coupler should not exceed 1cm.

Recommended Operating Conditions

Characteristic		Symbol	Min.	Typ.	Max.		Unit
Input current, on	(Note 7)	$I_{F(ON)}$	7	8	10		mA
Input voltage, off		$V_{F(OFF)}$	0	—	0.8		V
Supply voltage		V_{CC}	15	—	30	20	V
Peak output current		I_{OPH}/I_{OPL}	—	—	±0.5		A
Operating temperature		T_{opr}	−20	25	70	85	°C

Note 7: Input signal rise time (fall time) < 0.5 μs.

MUR1520

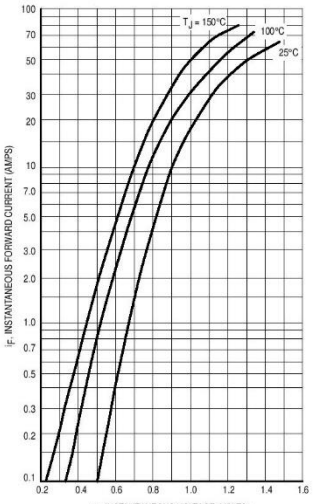


Figure 1. Typical Forward Voltage

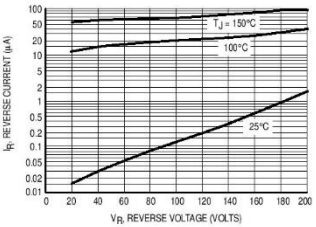


Figure 2. Typical Reverse Current

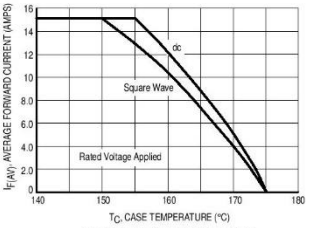


Figure 3. Current Derating, Case

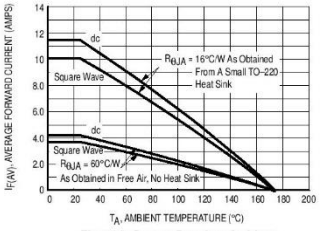


Figure 4. Current Derating, Ambient

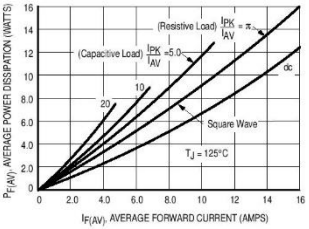


Figure 5. Power Dissipation

MUR1540

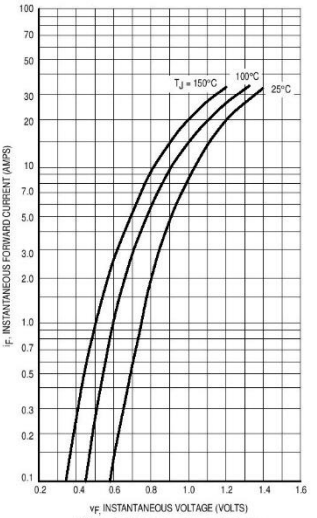


Figure 6. Typical Forward Voltage

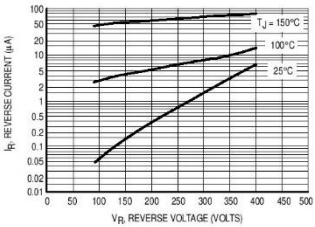


Figure 7. Typical Reverse Current

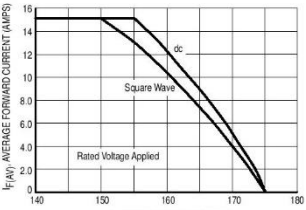


Figure 8. Current Derating, Case

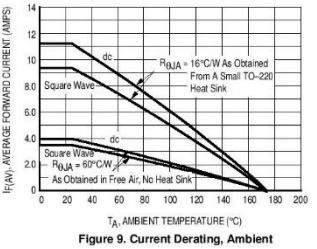


Figure 9. Current Derating, Ambient

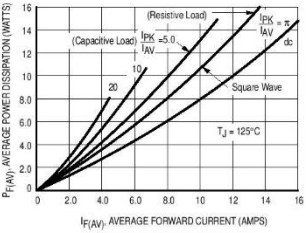


Figure 10. Power Dissipation

MUR1520, MUR1540, MUR1560

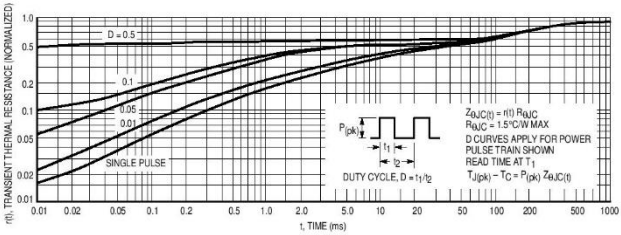


Figure 16. Thermal Response

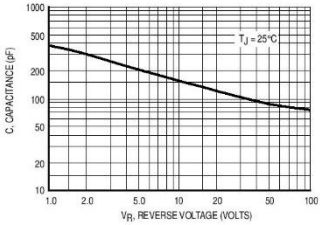


Figure 17. Typical Capacitance

MUR1520, MUR1540, MUR1560

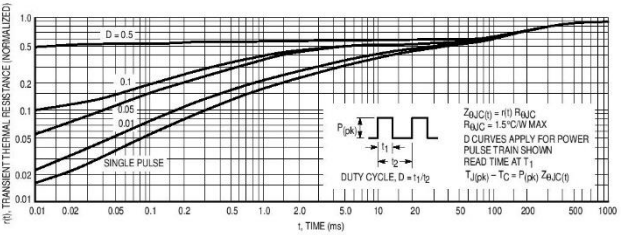


Figure 16. Thermal Response

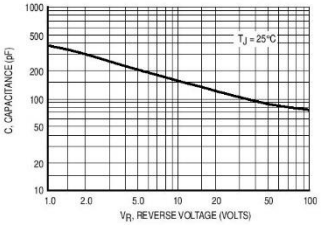


Figure 17. Typical Capacitance

3. Datasheet *RTC DS1307*

maxim integrated™

DS1307
64 x 8, Serial, I²C Real-Time Clock

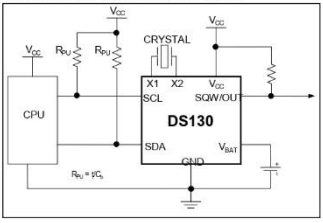
GENERAL DESCRIPTION

The DS1307 serial real-time clock (RTC) is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially through an I²C, bidirectional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power-sense circuit that detects power failures and automatically switches to the backup supply. Timekeeping operation continues while the part operates from the backup supply.

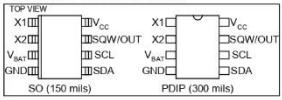
FEATURES

- Real-Time Clock (RTC) Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the week, and Year with Leap-Year Compensation Valid Up to 2100
- 56-Byte, Battery-Backed, General-Purpose RAM with Unlimited Writes
- I²C Serial Interface
- Programmable Square-Wave Output Signal
- Automatic Power-Fail Detect and Switch Circuitry
- Consumes Less than 500nA in Battery-Backup Mode with Oscillator Running
- Optional Industrial Temperature Range: -40°C to +85°C
- Available in 8-Pin Plastic DIP or SO
- Underwriters Laboratories (UL) Recognized

TYPICAL OPERATING CIRCUIT



PIN CONFIGURATIONS



ORDERING INFORMATION

PART	TEMP RANGE	VOLTAGE (V)	PIN-PACKAGE	TOP MARK*
DS1307+	0°C to +70°C	5.0	8 PDIP (300 mils)	DS1307
DS1307N+	-40°C to +85°C	5.0	8 PDIP (300 mils)	DS1307N
DS1307Z+	0°C to +70°C	5.0	8 SO (150 mils)	DS1307
DS1307ZN+	-40°C to +85°C	5.0	8 SO (150 mils)	DS1307N
DS1307Z+T&R	0°C to +70°C	5.0	8 SO (150 mils) Tape and Reel	DS1307
DS1307ZN+T&R	-40°C to +85°C	5.0	8 SO (150 mils) Tape and Reel	DS1307N

+Denotes a lead-free/RoHS-compliant package.
**A "+" anywhere on the top mark indicates a lead-free package. An "N" anywhere on the top mark indicates an industrial temperature range device.*

For pricing, delivery, and ordering information, please contact Maxim Direct at 1-888-629-4642, or visit Maxim's website at www.maximintegrated.com.

REV: 100208

ABSOLUTE MAXIMUM RATINGS

Voltage Range on Any Pin Relative to Ground ..... -0.5V to +7.0V
Operating Temperature Range (Noncondensing)
Commercial ..... 0°C to +70°C
Industrial ..... -40°C to +85°C
Storage Temperature Range ..... -55°C to +125°C
Soldering Temperature (DIP, leads) ..... +260°C for 10 seconds
Soldering Temperature (surface mount) ..... Refer to the JPC/JEDEC J-STD-020 Specification.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to the absolute maximum rating conditions for extended periods may affect device reliability.

RECOMMENDED DC OPERATING CONDITIONS

(T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V_{CC}		4.5	5.0	5.5	V
Logic 1 Input	V_{IH}		2.2		V_{CC} + 0.3	V
Logic 0 Input	V_{IL}		-0.3		+0.8	V
V_{BAT} Battery Voltage	V_{BAT}		2.0	3	3.5	V

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 4.5V to 5.5V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Input Leakage (SCL)	I_{LI}		-1		1	µA
I/O Leakage (SDA, SQW/OUT)	I_{LO}		-1		1	µA
Logic 0 Output (I_{OL} = 5mA)	V_{OL}				0.4	V
Active Supply Current (f_{SCL} = 100kHz)	I_{CCA}				1.5	mA
Standby Current	I_{CCS}	(Note 3)			200	µA
V_{BAT} Leakage Current	I_{BATLKG}			5	50	nA
Power-Fail Voltage (V_{BAT} = 3.0V)	V_{PF}		1.216 x V_{BAT}	1.25 x V_{BAT}	1.284 x V_{BAT}	V

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 0V, V_{BAT} = 3.0V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
V_{BAT} Current (OSC ON); SQW/OUT OFF	I_{BAT1}			300	500	nA
V_{BAT} Current (OSC ON); SQW/OUT ON (32kHz)	I_{BAT2}			480	800	nA
V_{BAT} Data-Retention Current (Oscillator Off)	I_{BATDR}			10	100	nA

WARNING: Negative undershoots below -0.3V while the part is in battery-backed mode may cause loss of data.

AC ELECTRICAL CHARACTERISTICS

(V_{CC} = 4.5V to 5.5V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCL Clock Frequency	f_{SCL}		0		100	kHz
Bus Free Time Between a STOP and START Condition	t_{BUF}		4.7			µs
Hold Time (Repeated) START Condition	$t_{HD:STA}$	(Note 4)	4.0			µs
LOW Period of SCL Clock	t_{LOW}		4.7			µs
HIGH Period of SCL Clock	t_{HIGH}		4.0			µs
Setup Time for a Repeated START Condition	$t_{SU:STA}$		4.7			µs
Data Hold Time	$t_{HD:DAT}$		0			µs
Data Setup Time	$t_{SU:DAT}$	(Notes 5, 6)	250			ns
Rise Time of Both SDA and SCL Signals	t_R				1000	ns
Fall Time of Both SDA and SCL Signals	t_F				300	ns
Setup Time for STOP Condition	$t_{SU:STO}$		4.7			µs

CAPACITANCE

(T_A = +25°C)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Pin Capacitance (SDA, SCL)	$C_{I/O}$				10	pF
Capacitance Load for Each Bus Line	C_B	(Note 7)			400	pF

Note 1: All voltages are referenced to ground.

Note 2: Limits at -40°C are guaranteed by design and are not production tested.

Note 3: I_{CCS} specified with V_{CC} = 5.0V and SDA, SCL = 5.0V.

Note 4: After this period, the first clock pulse is generated.

Note 5: A device must internally provide a hold time of at least 300ns for the SDA signal (referred to the $V_{IH(MIN)}$ of the SCL signal) to bridge the undefined region of the falling edge of SCL.

Note 6: The maximum $t_{HD:DAT}$ only has to be met if the device does not stretch the LOW period (t_{LOW}) of the SCL signal.

Note 7: C_B—total capacitance of one bus line in pF.

TIMING DIAGRAM

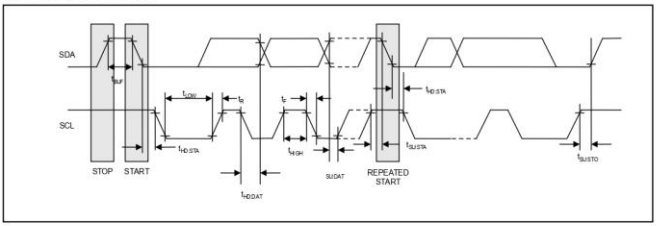
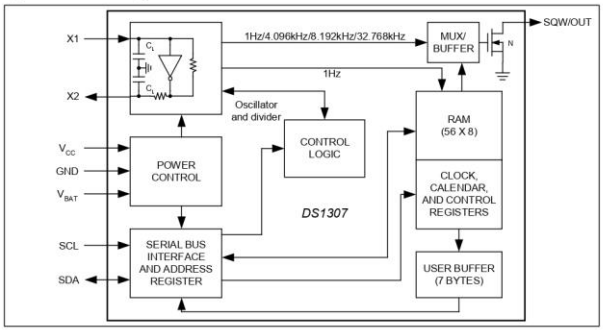
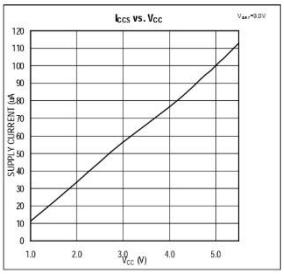


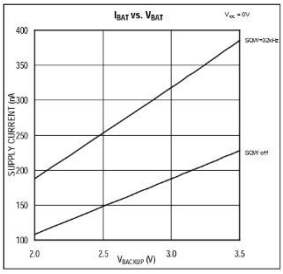
Figure 1. Block Diagram

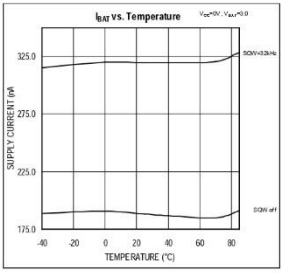


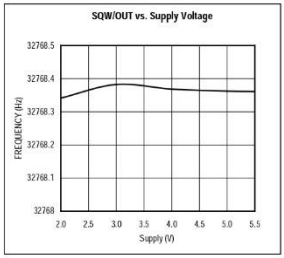
TYPICAL OPERATING CHARACTERISTICS

(V_{CC} = 5.0V, T_A = +25°C, unless otherwise noted.)









PIN DESCRIPTION

PIN	NAME	FUNCTION
1	X1	Connections for Standard 32.768kHz Quartz Crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance (C_L) of 12.5pF. X1 is the input to the oscillator and can optionally be connected to an external 32.768kHz oscillator. The output of the internal oscillator, X2, is floated if an external oscillator is connected to X1.
2	X2	**Note:** For more information on crystal selection and crystal layout considerations, refer to *Application Note 58: Crystal Considerations with Dallas Real-Time Clocks.*
3	V_{BAT}	Backup Supply Input for Any Standard 3V Lithium Cell or Other Energy Source. Battery voltage must be held between the minimum and maximum limits for proper operation. Diodes in series between the battery and the V_{BAT} pin may prevent proper operation. If a backup supply is not required, V_{BAT} must be grounded. The nominal power-fail trip point (V_{PF}) voltage at which access to the RTC and user RAM is denied is set by the internal circuitry as 1.25 x V_{BAT} nominal. A lithium battery with 48mAh or greater will back up the DS1307 for more than 10 years in the absence of power at +25°C. UL recognized to ensure against reverse charging current when used with a lithium battery. Go to: www.maxim-ic.com/qa/info/ul/.
4	GND	Ground
5	SDA	Serial Data Input/Output. SDA is the data input/output for the I²C serial interface. The SDA pin is open drain and requires an external pullup resistor. The pullup voltage can be up to 5.5V regardless of the voltage on V_{CC}.
6	SCL	Serial Clock Input. SCL is the clock input for the I²C interface and is used to synchronize data movement on the serial interface. The pullup voltage can be up to 5.5V regardless of the voltage on V_{CC}.
7	SQW/OUT	Square Wave/Output Driver. When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square-wave frequencies (1Hz, 4kHz, 8kHz, 32kHz). The SQW/OUT pin is open drain and requires an external pullup resistor. SQW/OUT operates with either V_{CC} or V_{BAT} applied. The pullup voltage can be up to 5.5V regardless of the voltage on V_{CC}. If not used, this pin can be left floating.
8	V_{CC}	Primary Power Supply. When voltage is applied within normal limits, the device is fully accessible and data can be written and read. When a backup supply is connected to the device and V_{CC} is below V_{TP}, read and writes are inhibited. However, the timekeeping function continues unaffected by the lower input voltage.

DETAILED DESCRIPTION

The DS1307 is a low-power clock/calendar with 56 bytes of battery-backed SRAM. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The DS1307 operates as a slave device on the I²C bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When V_{CC} falls below 1.25 x V_{BAT}, the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out-of-tolerance system. When V_{CC} falls below V_{BAT}, the device switches into a low-current battery-backup mode. Upon power-up, the device switches from battery to V_{CC} when V_{CC} is greater than V_{BAT} +0.2V and recognizes inputs when V_{CC} is greater than 1.25 x V_{BAT}. The block diagram in Figure 1 shows the main elements of the serial RTC.

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

LAMPIRAN C

DATA SAMPEL DOWNTIME

1. Data kesulurhan downtime pada tanggal 23 – 24 Januari 2019 pada *conveyor* 7 / Station 7 Shift Malam

No	Tanggal	Station	Waktu Pertama	Waktu Selesai	Konndisi	Durasi (Detik)
1	23/1/2019	Station 7	23:0:10	23:3:52	Merah	222
2	24/1/2019	Station 7	4:6:32	4:9:4	Merah	152
3	24/1/2019	Station 7	4:28:37	4:29:8	Kuning	31
4	24/1/2019	Station 7	4:29:9	4:31:18	Merah	129
5	24/1/2019	Station 7	22:38:29	22:39:42	Kuning	54
TOTAL						**588**

2. Data kesulurhan downtime pada tanggal 24 – 25 Januari 2019 pada *conveyor* 7 / Station 7 Shift Malam

No	Tanggal	Station	Waktu Pertama	Waktu Selesai	Konndisi	Durasi (Detik)
1	24/1/2019	Station 7	22:38:29	22:39:42	Merah	73
2	24/1/2019	Station 7	22:52:48	22:53:3	Merah	15
3	24/1/2019	Station 7	23:8:50	23:9:10	Merah	20
4	24/1/2019	Station 7	23:19:48	23:20:18	Kuning	30
5	24/1/2019	Station 7	23:20:20	23:21:41	Merah	81
6	25/1/2019	Station 7	2:32:43	2:33:7	Kuning	24

No	Tanggal	Station	Waktu Pertama	Waktu Selesai	Konndisi	Durasi (Detik)
7	25/1/2019	Station 7	6:28:57	6:29:40	Kuning	42
8	25/1/2019	Station 7	6:32:46	6:33:9	Kuning	23
TOTAL						**308**

3. Data kesulurhan downtime pada tanggal 25 – 26 Januari 2019 pada *Conveyor* 7 / Station 7 Shift Malam

No	Tanggal	Station	Waktu Pertama	Waktu Selesai	Konndisi	Durasi (Detik)
1	25/1/2019	Station 7	20:28:12	20:29:1	Kuning	49
2	25/1/2019	Station 7	21:40:15	21:41:21	Merah	66
3	25/1/2019	Station 7	21:47:36	21:48:51	Merah	75
4	26/1/2019	Station 7	2:32:27	2:33:1	Kuning	34
5	26/1/2019	Station 7	2:39:3	2:40:18	Merah	75
6	26/1/2019	Station 7	2:50:36	2:52:1	Merah	85
7	26/1/2019	Station 7	3:45:25	3:46:5	Kuning	40
8	26/1/2019	Station 7	4:18:3	4:22:23	Merah	260
9	26/1/2019	Station 7	5:44:11	5:45:48	Merah	97
TOTAL						**781**

4. Data kesulurhan downtime pada tanggal 23 – 24 Januari 2019 pada *Conveyor* 20 / Station 20 Shift Malam

No	Tanggal	Station	Waktu Pertama	Waktu Selesai	Konndisi	Durasi (Detik)
1	23/1/2019	Station 7	17:17:0	17:50:21	Merah	201
2	23/1/2019	Station 7	18:32:0	18:42:29	Merah	629
3	24/1/2019	Station 7	2:21:0	2:52:25	Merah	1885
4	24/1/2019	Station 7	6:8:0	6:29:26	Merah	1286
5	24/1/2019	Station 7	6:48:0	6:49:23	Merah	83
TOTAL						**4084**

5. Data kesulurhan downtime pada tanggal 24 – 25 Januari 2019 pada *Conveyor* 20 / Station 20 Shift Pagi

No	Tanggal	Station	Waktu Pertama	Waktu Selesai	Konndisi	Durasi (Detik)
1	24/1/2019	Station 7	8:19:0	8:50:26	Merah	1886
2	24/1/2019	Station 7	12:6:0	12:32:22	Merah	1582
3	24/1/2019	Station 7	12:24:0	12:32:57	Merah	537
4	24/1/2019	Station 7	13:26:0	13:39:4	Merah	784
5	24/1/2019	Station 7	13:34:0	13:42:30	Merah	510
6	24/1/2019	Station 7	15:11:0	15:46:37	Merah	2137
7	25/1/2019	Station 7	18:55:0	18:55:4	Merah	4
TOTAL						**7440**

6. Pengujian *Router* Dengan Penghalang

NO	Jarak (m)	Loss (%)
1	2	0%
2	4	0%
3	6	0%
4	8	0%
5	10	0%
6	12	0%
7	14	0%
8	16	0%
9	18	0%
10	20	0%
11	22	0%
12	24	0%
13	26	0%
14	28	0%
15	30	0%
16	32	0%
17	34	25%
18	36	25%
19	38	25%
20	40	25%
21	42	25%
22	44	25%
23	46	25%
24	48	25%
25	50	50%

7. Pengujian *Router* Tanpa Penghalang

NO	Jarak (m)	Loss (%)
1	2	0%
2	4	0%
3	6	0%
4	8	0%
5	10	0%
6	12	0%

NO	Jarak (m)	Loss (%)
7	14	0%
8	16	0%
9	18	0%
10	20	0%
11	22	0%
12	24	0%
13	26	0%
14	28	0%
15	30	0%
16	32	0%
17	34	0%
18	36	0%
19	38	0%
20	40	0%
21	42	25%
22	44	25%
23	46	25%
24	48	25%
25	50	50%

8. Pengujia Alat 2 x 24 Jam dari tanggal 18 – 20 Juli 2019

No	Tanggal	Station	Waktu Pertama	Waktu Selesai	Kondisi	Durasi (Detik)
1	18/7/2019	Station 1	18:51:42	18:51:56	Merah	12
2	18/7/2019	Station 2	18:53:31	18:54:00	Merah	29
3	18/7/2019	Station 3	18:56:43	18:57:00	Kuning	17
4	18/7/2019	Station 2	20:56:43	20:57:04	Merah	21
5	18/7/2019	Station 1	21:57:16	21:57:23	Kuning	7
6	18/7/20	Station	23:57	23:57	Merah	20

No	Tanggal	Station	Waktu Pertama	Waktu Selesai	Konndisi	Durasi (Detik)
	19	3	:16	:36		
7	19/7/2019	Station 3	00:58:18	00:58:31	Merah	13
8	19/7/2019	Station 2	00:59:08	01:30:09	Merah	1862
9	19/7/2019	Station 2	02:32:16	02:50:25	Kuning	1099
10	19/7/2019	Station 1	03:58:00	04:00:00	Kuning	120
11	19/7/2019	Station 2	04:00:00	04:12:04	Kuning	734
12	19/7/2019	Station 1	05:14:07	05:20:19	Merah	372
13	19/7/2019	Station 1	06:55:01	07:00:21	Kuning	320
14	19/7/2019	Station 3	06:55:02	07:10:24	Kuning	622
15	19/7/2019	Station 3	08:19:25	08:19:40	Merah	15
16	19/7/2019	Station 1	11:46:46	11:59:47	Merah	781
17	19/7/2019	Station 2	13:46:52	13:56:00	Kuning	652
18	19/7/2019	Station 1	15:47:08	16:00:00	Kuning	772
19	19/7/2019	Station 2	18:50:10	19:00:11	Merah	601
20	19/7/2019	Station 3	19:30:11	19:47:12	Kuning	181
21	19/7/2019	Station 3	19:47:40	19:48:43	Merah	63
22	19/7/20	Station	21:07	21:47	Kunin	2404

No	Tanggal	Station	Waktu Pertama	Waktu Selesai	Konndisi	Durasi (Detik)
	19	2	:44	:48	g	
23	19/7/2019	Station 1	23:32:47	23:33:50	Merah	63
24	19/7/2019	Station 2	23:32:51	23:35:55	Merah	184
25	20/7/2019	Station 2	01:20:15	01:33:50	Merah	815
26	20/7/2019	Station 2	04:30:00	04:40:07	Kuning	607
27	20/7/2019	Station 3	06:30:00	06:32:00	Kuning	120
28	20/7/2019	Station 1	06:30:00	06:30:30	Kuning	30
29	20/7/2019	Station 1	08:20:00	08:20:42	Merah	42
30	20/7/2019	Station 2	10:07:00	10:08:07	Kuning	67
31	20/7/2019	Station 3	12:30:00	12:45:00	Merah	900
32	20/7/2019	Station 1	14:44:00	14:44:50	Kuning	50
33	20/7/2019	Station 3	15:30:00	15:30:54	Kuning	54
34	20/7/2019	Station 2	15:30:00	15:33:33	Kuning	183
35	20/7/2019	Station 3	16:20:00	16:20:15	Merah	15
36	20/7/2019	Station 1	17:40:00	17:50:07	Kuning	607
37	20/7/2019	Station 2	18:30:00	19:00:00	Merah	1800
TOTAL						**16254**
Rata – Rata						**439,2**

No	Tanggal	Station	Waktu Pertama	Waktu Selesai	Konndisi	Durasi (Detik)
						97

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

DAFTAR RIWAYAT HIDUP

Nama	: Lugas Jabar Waskito
TTL	: Probolinggo, 20 Desember 1997
Jenis Kelamin	: Laki - Laki
Agama	: Islam
Alamat Rumah	: Jl. Komak Indah No.11 RT.04 RW.10 Kec. Leces, Kab Probolinggo
Telp/HP	: 089668857297
E-mail	: lugasjabar15@Gmail.com

RIWAYAT PENDIDIKAN

1. 2004 - 2010 : SD Taruna Dra Zulaeha
2. 2010 - 2013 : SMP Taruna Dra Zulaeha
3. 2013 – 2016 : SMA Taruna Dra Zulaeha
4. 2016 – Sekarang : Departemen Teknik Elektro Otomasi-Fakultas Vokasi (FV) -Institut Teknologi Sepuluh Nopember (ITS)

PENGALAMAN KERJA

1. Magang di PT. Jatim Autocomp Indonesia Gempol, Pasuruan (Agustus 2018 – Januari 2019)

PENGALAMAN ORGANISASI

1. Pemberdayaan Sumber Daya Mahasiswa Himpunan Mahasiswa D3 Teknik Elektro ITS 2017- 2018
2. Pemberdayaan Sumber Daya Mahasiswa Himpunan Mahasiswa D3 Teknik Elektro ITS 2018- 2019
3. Staff Kementrian Perekonomian BEM ITS 2018 – 2019
4. Dirjen Kementrian Perekonomian BEM Vokasi 2018 – 2019

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