



**TUGAS AKHIR - TE 145561**

**MONITORING DAN PELAPORAN KEHADIRAN SISWA  
BERBASIS WEBSITE**

Amir Muhyidin  
NRP 10311500010012

Dosen Pembimbing  
Ir. Arif Musthofa, M.T.  
Dwi Lastomo, S.Si, M.T.

PROGRAM STUDI ELEKTRO INDUSTRI  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018





***FINAL PROJECT - TE 145561***

***MONITORING AND REPORTING WEB-BASED  
STUDENT ATTENDANCE***

Amir Muhyidin  
NRP 10311500010012

*Supervisor*  
Ir. Arif Musthofa, M.T.  
Dwi Lastomo, S.Si, M.T.

***ELECTRICAL INDUSTRY STUDY PROGRAM  
Electrical and Automation Engineering Department  
Vocational Faculty  
Sepuluh Nopember Institute Of Technology  
Surabaya 2018***



## PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **“Monitoring Dan Pelaporan Kehadiran Siswa Berbasis Website”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang kami 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, 22 Juni 2018



Amir Muhyidin  
NRP 10311500010012

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

**MONITORING DAN PELAPORAN KEHADIRAN SISWA  
BERBASIS WEBSITE**

**TUGAS AKHIR**

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

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

Menyetujui :

Dosen Pembimbing I

Dosen Pembimbing II

Ir. Arif Musthofa, M.T.  
NIP. 196608111992031004

Dwi Lastomo, S.Si, M.T.  
NIP. 1987201711059



**SURABAYA  
JULI, 2018**

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

## Monitoring Dan Pelaporan Kehadiran Siswa Berbasis Website

Nama Mahasiswa 1 : Amir Muhyidin  
NRP : 10311500010012  
Dosen Pembimbing 1 : Ir. Arif Musthofa, M.T.  
NIP : 196608111992031 004  
Dosen Pembimbing 2 : Dwi Lastomo, S.Si, M.T.  
NIP : 19872017111059

### ABSTRAK

Surabaya adalah salah satu kota besar di Indonesia. Pada tahun 2017, Surabaya dinobatkan sebagai kota cerdas (*Smart City*). *Smart City* sangat identik dengan kemajuan teknologi di segala bidang. Kemajuan teknologi juga dapat diterapkan dalam dunia pendidikan, seperti sistem monitoring kehadiran siswa berbasis IT. Meningkatnya pelanggaran siswa sekolah dalam hal membolos sangat memprihatinkan. Akan tetapi hal tersebut tidak ada penanganan secara langsung dari pihak sekolah. Hal ini mempengaruhi proses pembelajaran disekolah dan diperlukan sistem monitoring secara otomatis agar dapat mengetahui kehadiran siswa disekolah.

Maka dari itu pada Tugas Akhir ini dibuatlah sistem monitoring dan pelaporan kehadiran siswa disekolah dengan menggunakan akses *fingerprint*, sehingga dapat mengurangi pelanggaran membolos tersebut. Metode yang digunakan adalah pembacaan sidik jari atau yang lebih sering dikenal dengan metode *minutiea*. Metode ini sering digunakan sebagai metode perbandingan sensor *fingerprint* berdasarkan pencocokan pola *minutiea*. Data yang didapat akan dikelola oleh Arduino Mega dan akan ditampilkan dalam *database* pada Website.

Setelah dilakukan pengujian, hasil dari pengujian menunjukkan bahwa pembacaan sensor *fingerprint* dan kecocokan ID sebesar 96%. Hasil data dapat dilihat pada komputer dan Website sekolah, dimana proses ini membantu guru untuk mengetahui siswa yang tidak hadir sekolah dan membantu dalam merekap kehadiran siswa.

**Kata Kunci** : Arduino Mega, Database, Fingerprint.

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

## **MONITORING AND REPORTING WEB-BASED STUDENT ATTENDANCE**

**Name of Student 1** : Amir Muhyidin  
**Number of Registration** : 10311500010012  
**Supervisor 1** : Ir. Arif Musthofa, M.T.  
**ID Number** : 196608111992031 004  
**Supervisor 2** : Dwi Lastomo, S.Si, M.T.  
**ID Number** : 1987201711059

### **ABSTRACT**

*Surabaya is one of the big cities in Indonesia. In 2017, Surabaya was named the smart city (Smart City). Smart City is synonymous with technological advances in all areas. Technological advances can also be applied in education, such as IT-based attendance monitoring systems. Increased violation of school students in terms of ditching is very alarming. However, there is no direct treatment from the school. This affects the learning process in schools and monitoring system is required automatically in order to know the presence of students in school.*

*Therefore, in this final project, monitoring and reporting system of school attendance by fingerprint access can be made, thus reducing the trespass violation. The method used is fingerprint reading or more often known as minutia method. This method is often used as a fingerprint sensor comparison method based on minutia pattern matching. 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 test show that the fingerprint sensor readings and ID matches are 96%. The results of the data can be seen on the computer and the school Website, where this process helps the teacher to know the students who are not attending school and help in recording student attendance.

**Keywords** : Arduino Mega, Database, Fingerprint , Interface.

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

## **KATA PENGANTAR**

Segala puji syukur penulis panjatkan kepada Allah SWT yang selalu memberikan kemudahan, kelancaran dan kerahmatan sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik. Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma-3 pada Bidang Studi Teknik Elektro Industri, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

### **Monitoring Dan Pelaporan Kehadiran Siswa Berbasis Website**

Dalam mengerjakan Tugas Akhir ini tidak lepas dari bantuan, motivasi dan dukungan dari berbagai pihak. Dalam kesempatan ini penulis ingin menyampaikan terima kasih kepada semua pihak, khususnya:

1. Kedua orang tua dan keluarga yang senantiasa mendoakan dan memberikan dukungan dengan tulus tiada henti.
2. Ir. Arif Musthofa, M.T, sebagai dosen pembimbing 1, yang telah berkenan meluangkan waktu, tenaga, pikiran serta memberikan pengarahan dan bimbingan dari awal hingga terselesainya Tugas Akhir.
3. Dwi Lastomo, S.Si, M.T, sebagai dosen pembimbing 2, yang tidak pernah bosan mengingatkan dan memberikan bimbingan.
4. Rekan-rekan mahasiswa ITS yang telah banyak membantu dalam penyelesaian Tugas Akhir.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Kritik dan saran yang membangun sangat penulis harapkan. Semoga Tugas Akhir ini dapat bermanfaat bagi pembaca dalam pengembangan keilmuan

Surabaya, 22 Juni 2018

Amir Muhyidin

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

# DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN JUDUL .....	iii
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
HALAMAN PENGESAHAN .....	vii
ABSTRAK.....	ix
<i>ABSTRACT</i> .....	xi
KATA PENGANTAR .....	xiii
DAFTAR ISI .....	xv
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL .....	xix
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Permasalahan .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan.....	3
1.5 Metodologi Penelitian.....	3
1.6 Sistematika.....	3
1.7 Relevansi .....	4
BAB II TEORI PENUNJANG .....	5
2.1 <i>Fingerprint Scanner</i> .....	5
2.2 <i>Arduino Mega 2560</i> .....	7
2.3 <i>Graphic LCD</i> .....	8
2.4 <i>Keypad 4x4</i> .....	9
2.5 <i>Real Time Clock (RTC) DS1307</i> .....	10
2.6 <i>EEPROM</i> .....	11
2.7 <i>Visual Basic .Net</i> .....	12
2.9 <i>Software XAMPP</i> .....	13
BAB III PERANCANGAN SISTEM.....	14
3.1 Blok Fungsional Sistem.....	14
3.2 Perancangan <i>Hardware</i> .....	16
3.2.1 Perancangan Mekanik .....	16
3.2.2 Rangkaian Modul Shield Arduino Mega .....	17
3.2.3 Rangkaian GLCD 128x64 .....	19
3.2.4 Rangkaian <i>Fingerprint Scanner</i> .....	19

3.2.5 Rangkaian RTC .....	20
3.2.6 Perancangan Rangkaian <i>Power Supply</i> .....	20
3.2.7 Rangkaian Keypad 4x4 .....	21
3.3 Perancangan <i>Software</i> .....	21
3.3.1 Arduino IDE .....	22
3.3.2 Perancangan <i>Database</i> .....	30
BAB IV PENGUJIAN DAN ANALISIS .....	35
4.1 Pengujian Arduino Mega 2560 .....	35
4.2 Pengujian <i>Fingerprint Scanner</i> .....	38
4.3 Pengujian RTC DS1307 .....	40
4.4 Pengujian GLDC .....	41
4.5 Pengujian Keypad 4x4.....	42
4.6 Pengujian <i>Database</i> .....	43
4.7 Pengujian Sistem Keseluruhan .....	45
BAB V PENUTUP .....	55
5.1 Kesimpulan.....	55
5.2 Saran .....	55
DAFTAR PUSTAKA .....	57
LAMPIRAN A .....	A-1
LAMPIRAN B.....	B-1
RIWAYAT HIDUP .....	C-1

## DAFTAR GAMBAR

Gambar 2.1	Jenis Umum Pola pada Metode <i>Minutia</i> .....	5
Gambar 2.2	<i>Fingerprint Scanner</i> dan <i>Connector</i> .....	6
Gambar 2.3	Board Arduino MEGA .....	7
Gambar 2.4	Pin pada <i>GLCD</i> .....	8
Gambar 2.5	<i>Keypad Module</i> 4 x 4 .....	10
Gambar 2.6	Konfigurasi Pin RTC DS1307.....	11
Gambar 2.7	Modul EEPROM AT24C02 .....	12
Gambar 2.8	Logo Visual Basic .....	12
Gambar 2.9	Logo XAMPP.....	13
Gambar 3.1	Blok Fungsional Sistem.....	15
Gambar 3.2	Diagram <i>Hardware</i> dan <i>Software</i> .....	16
Gambar 3.3	Desain <i>Hardware</i> .....	18
Gambar 3.4	Rangkaian Shield Arduino Mega .....	19
Gambar 3.5	Rangkaian <i>GLCD</i> .....	20
Gambar 3.6	Rangkaian <i>Fingerprint Scanner</i> .....	21
Gambar 3.7	Rangkaian RTC .....	21
Gambar 3.8	Rangkaian <i>Power Supply</i> .....	22
Gambar 3.9	Rangkaian Keypad 4x4 .....	22
Gambar 3.10	Diagram Pendaftaran Data Siswa .....	24
Gambar 3.11	<i>Flowchart</i> pada Sistem Absensi Siswa.....	25
Gambar 3.12	Inisialisasi <i>Fingerprint Scanner</i> .....	26
Gambar 3.13	<i>Flowchart</i> Daftar Sidik Jari.....	27
Gambar 3.14	<i>Flowchart Log In</i> Sidik Jari Siswa .....	28
Gambar 3.15	Program <i>Enrollment</i> .....	29
Gambar 3.16	Program Kecocokan Sidik Jari .....	30
Gambar 3.17	Program <i>Delete Data</i> .....	31
Gambar 3.18	Program RTC .....	31
Gambar 3.19	Konstruksi Keypad 4x4 .....	32
Gambar 3.20	Inisialisasi Keypad 4x4 .....	33
Gambar 3.21	Inisialisasi <i>GLCD</i> .....	34
Gambar 3.22	Program EEPROM .....	35
Gambar 3.23	Program Koneksi Visual Basic dengan Arduino ...	37
Gambar 3.24	Program Koneksi <i>Database MS. Access</i> .....	38
Gambar 3.25	Program <i>Export</i> ke Localhost.....	38
Gambar 3.26	Tampilan <i>Software</i> Visual Basic .....	39
Gambar 3.27	Tampilan <i>Input</i> Data Siswa .....	39
Gambar 3.28	Tampilan <i>Database</i> pada Website.....	40

Gambar 3.29	Program <i>Database</i> pada Website .....	40
Gambar 4.1	Pengujian <i>Enrollment</i> pada <i>Fingerprint</i> .....	43
Gambar 4.2	Pengujian Kecocokan Sidik Jari.....	44
Gambar 4.3	Pengujian <i>Delete ID</i> .....	45
Gambar 4.4	Pengujian RTC .....	47
Gambar 4.5	Pengujian GLCD.....	48
Gambar 4.6	Pengujian Keypad Tombol 1 .....	49
Gambar 4.7	Pengujian Keypad Angka 22.....	49
Gambar 4.8	Pengujian <i>Database Visual Basic .Net</i> .....	50
Gambar 4.9	Pengujian <i>Database ID Siswa</i> .....	51
Gambar 4.10	<i>Hardware dan Software</i> Sistem Keseluruhan.....	51
Gambar 4.11	<i>Flowchart</i> Sistem Keseluruhan .....	52
Gambar 4.12	Tampilan Awal Sistem Absensi .....	53
Gambar 4.13	Tampilan Menu Sistem Absensi .....	53
Gambar 4.14	Pilih Tombol C pada Keypad.....	55
Gambar 4.15	Tampilan <i>Login</i> Sistem Absensi .....	55
Gambar 4.16	Tampilan <i>Log In</i> Berhasil .....	56
Gambar 4.17	Tampilan Visual Basic Sistem Keseluruhan .....	56
Gambar 4.18	Tampilan Laporan Absensi Siswa pada Website .....	57
Gambar 4.19	Tampilan Data Siswa Berbeda Status .....	57
Gambar 4.20	Tampilan Laporan Siswa Berbeda Status.....	58
Gambar 4.21	Tampilan Laporan Siswa dengan Tanggal Tertentu.....	59
Gambar 4.22	Pilihan <i>Search</i> Nama Siswa .....	59
Gambar 4.23	Tampilan Laporan Siswa berdasarkan Nama.....	60

## DAFTAR TABEL

Tabel 2.1	Deskripsi Spesifikasi pada Arduino MEGA .....	7
Tabel 2.2	Deskripsi Pin <i>GLCD</i> .....	8
Tabel 2.3	Pin RTC DS1307 .....	11
Tabel 3.1	Mapping Pin Analog/Digital Arduino Mega .....	19
Tabel 4.1	Data Akses <i>Fingerprint</i> .....	45
Tabel 4.2	Data ID Siswa .....	50

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

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Indonesia terdiri dari banyak kota yang tersebar di berbagai provinsi. Kota Surabaya adalah salah satu kota terbesar di Indonesia. Pada tahun 2017, kota Surabaya dinobatkan sebagai kota cerdas (*Smart City*). *Smart City* adalah konsep kota cerdas yang dirancang guna membantu berbagai hal kegiatan masyarakat, terutama dalam upaya mengelola sumber daya yang ada dengan efisien, serta memberikan kemudahan mengakses informasi kepada masyarakat. Salah satu aspek yang harus terpenuhi dalam kriteria *Smart City* adalah *smart technology*. *Smart technology* ini dapat diterapkan diberbagai bidang, salah satunya dapat diterapkan di dunia pendidikan. Seperti halnya penggunaan sistem *monitoring* kehadiran siswa berbasis IT, dimana alat ini dapat menjadi penunjang kemajuan IPTEK pada *Smart City*. (<http://jatimprov.go.id/read/berita-pengumuman/e-government-jadikan-surabaya-sebagai-smart-city>).

Ada beberapa pelajar yang tidak lagi mentaati peraturan sekolah. Terutama dalam hal masuk sekolah. Fakta di lapangan menunjukkan bahwa masih ada siswa yang sering terlambat, terutama di kota – kota besar seperti Surabaya. Dalam aturan sekolah mengharuskan siswa datang sebelum jam 07.00, akan tetapi kenyatannya masih ada siswa yang datang lewat jam tersebut. Banyaknya siswa yang terlambat mengakibatkan kurang lancarnya proses kegiatan belajar mengajar. Selain itu, untuk memantau siapa saja siswa yang terlambat tidak tercatat secara otomatis. Hal itu membuat sulit bagi para guru ketertiban dalam menindaklanjuti siswa dan sangat merugikan orang tua. Saat ini pihak sekolah tidak tahu ketika ada siswa yang membolos. Pihak orang tua pun ada yang mengetahui dan ada yang tidak mengetahui apabila anaknya tidak masuk sekolah atau membolos. Hal seperti ini masih kerap ditemui, tidak ada tindakan secara cepat dan tepat oleh pihak sekolah apabila ada siswanya yang tidak hadir sekolah. Oleh karena itu proses absensi secara konvensional

dengan mencatat siswa diatas kertas harus segera diganti. Pencatatan absensi siswa secara digital dapat membantu guru dan mempermudah mengetahui siswa yang tidak hadir. Absensi secara digital yang dilakukan pihak sekolah dapat membantu orang tua dengan melihat absensi anaknya.  
(<http://www.pendidikanekonomi.com/2013/04/perilaku-membolos-dan-faktor-yang.html>)

Sistem *monitoring* keberadaan siswa di sekolah dapat membantu orang tua dan pihak sekolah dalam memantau siswanya. Guru konseling selaku pihak yang berwenang menangani kehadiran siswa dapat memantau langsung secara *online* melalui komputer, sehingga mempermudah mengawasi siswanya dan dapat melakukan evaluasi. Hal ini juga dapat menunjang kemajuan teknologi di kota Surabaya, dan sebagai implementasi *Smart City*.

Berdasarkan beberapa hal yang telah dipaparkan di atas, maka penulis mengambil judul Proyek Akhir : “Monitoring dan Pelaporan Kehadiran Siswa Berbasis Website.”

## **1.2 Permasalahan**

Melihat dari latar belakang diatas, maka dapat dirumuskan beberapa permasalahan diantaranya :

1. Tidak adanya catatan otomatis mengenai kehadiran siswa di sekolah yang dapat dipantau oleh guru dan orang tua dapat menerima informasi anaknya.
2. Surabaya adalah *Smart City* yang selayaknya memanfaatkan kemajuan teknologi dalam bidang pendidikan.

## **1.3 Batasan Masalah**

Dalam menyusun Tugas Akhir ini, diperlukan suatu batasan masalah agar tidak terlalu luas pembahasannya. Adapun batasan masalahnya adalah sebagai berikut :

1. Membuat *prototpye* sistem absensi siswa dengan *Fingerprint Scanner* untuk melakukan pengujian program dengan Arduino Mega 2560.
2. Pengguna dari Website pada sistem ini adalah guru sekolah.

3. Pemrograman *software* menggunakan dengan aplikasi Arduino IDE .
4. Sistem ini diujikan dengan 5 data ID dan nama yang berbeda – beda.

#### **1.4 Tujuan**

Tujuan menuliskan Tugas Akhir ini adalah :

1. Mempermudah komunikasi orang tua dan pihak sekolah dalam memantau anaknya.
2. Mewujudkan Surabaya sebagai *Smart City* dalam bidang pendidikan.

#### **1.5 Metodologi Penelitian**

Penelitian ini dilakukan melalui beberapa tahapan metodologi, yaitu, studi literatur, perencanaan dan pembuatan alat, tahap pengujian dan analisa alat, dan yang terakhir adalah penyusunan laporan berupa buku Tugas Akhir.

Pada tahap studi literatur penulis mengumpulkan data dan informasi dengan membaca buku pustaka dan catatan kuliah, terutama mengenai materi yang berhubungan dengan Fingerprint Sensor dan pemrograman kontroler serta materi lain yang berhubungan dengan masalah yang dibahas. Tahap perancangan sistem meliputi perancangan perangkat keras dan perancangan perangkat lunak. Perancangan perangkat keras meliputi pengaturan koneksi *Fingerprint* dengan Arduino Mega 2560. Perancangan perangkat lunak meliputi pembuatan program kontroler dan program Visual Basic untuk proses berjalannya pada alat ini. Pengujian alat dilakukan setelah menggabungkan perangkat keras dan perangkat lunak menjadi satu sistem secara keseluruhan. Dalam tahapan ini, alat yang sudah dibuat akan diuji terlebih dahulu untuk mengetahui tingkat keberhasilan alat. Setelah pembuatan alat semuanya terselesaikan dengan baik dan benar, maka dilakukan penyusunan buku Tugas Akhir sebagai bukti dan hasil yang telah dicapai selama ini pada saat pengerjaan alat hingga selesai.

#### **1.6 Sistematika**

Sistematika pembahasan penyusunan Tugas Akhir ini direncanakan sebagai berikut:

## **BAB I : Pendahuluan**

Bab ini membahas pendahuluan yang terdiri dari latar belakang, permasalahan, batasan masalah, tujuan, sistematika, penjelasan secara singkat mengenai bab-bab dalam laporan Tugas Akhir dan relevansi atau manfaat.

## **BAB I : Teori Penunjang**

Bab ini akan membahas teori-teori yang menunjang dan berkaitan dengan penyelesaian Tugas Akhir, seperti Arduino Mega, *Fingerprint Scanner*, LCD Graphic, Arduino IDE, Visual Basic.Net 2010 dan lain-lain.

## **BAB III : Perancangan Sistem**

Bab ini membahas tentang tahapan yang dilakukan dalam perencanaan dan pembuatan Tugas Akhir sesuai dengan pembagian tugas dalam tim, yaitu perancangan *Software*.

## **BAB IV : Pengujian dan Analisis**

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

## **BAB V : Penutup**

Bab ini membahas kesimpulan dari pembahasan, analisa permasalahan, data, sintesis pemecahan masalah dan uji coba system. Untuk meningkatkan hasil akhir yang lebih baik, diberikan saran-saran terhadap apa yang dihasilkan dari Tugas Akhir ini.

### **1.7 Relevansi**

Relevansi dari Tugas Akhir ini adalah :

1. Untuk mempermudah dalam melakukan absensi.
2. Untuk me-*monitor* kehadiran siswa oleh guru.
3. Untuk me-*monitor* keberadaan siswa oleh orang tua.
4. Untuk memanfaatkan perkembangan teknologi yang menunjang *Smart City*.

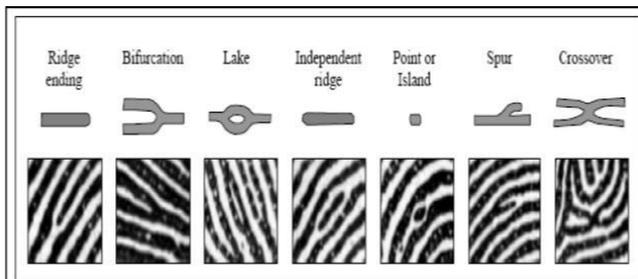
## BAB II TEORI PENUNJANG

Pada teori penunjang ini membahas tentang teori –teori yang menunjang dalam pembuatan sistem monitoring pada Tugas Akhir ini. Pada bab ini dijelaskan dasar teori berjumlah 8 poin. Penjelasan dari dasar teori tersebut terbagi menjadi poin yang sudah dijelaskan dibawah ini.

### 2.1 *Fingerprint Scanner* [1]

*Fingerprint scanner* atau alat pemindai sidik jari merupakan suatu komponen elektronik yang berfungsi membaca pola sidik jari dan mencocokkan dengan *database* yang telah tersimpan dalam memori. Sistem ini memiliki cara kerja yang terbagi menjadi tiga yaitu *enrollment*, *delete*, dan *fingerprint access*. Pada saat ini, sensor sidik jari telah banyak dipasarkan. Divisi forensik pada penegak hukum saat ini menggunakan sidik jari sebagai proses penyelidikan.

Hal ini dikarenakan setiap orang memiliki garis atau bentuk sidik jari yang berbeda – beda. Pola pembacaan sidik jari dapat dibandingkan dengan metode *minutia*. Metode ini sering digunakan sebagai metode perbandingan sensor *fingerprint* berdasarkan pencocokan pola *minutia*. *Minutia* yang berarti kecil, dalam konteks sidik jari hal tersebut berarti pola – pola pada sidik jari. Sebagian besar metode ini memerlukan gambar sidik jari untuk dikonversi menjadi gambar biner. Gambar 2.1 merupakan contoh pola metode *minutia*.



Gambar 2.1 Jenis Umum Pola pada Metode *Minutia*

Pada saat sistem memulai *scanning* sidik jari dengan alat pemindai, hasil akan disimpan dalam memori. Hasil *scanning* disimpan dalam format digital pada saat *enrollment* (pendaftaran sidik jari). Setelah itu, rekaman sidik jari tersebut diproses dan dibuatkan daftar pola fitur sidik jari yang unik. Pola fitur sidik jari yang unik tersebut kemudian disimpan dalam memori atau *database*. Pola sidik jari yang unik ini disebut dengan pola *minutia*. Pada saat identifikasi, pola *minutia* tersebut kemudian dicocokkan dengan hasil *scan* sidik jari.

Teknik pembacaan sidik jari oleh mesin absensi sidik jari dapat dikelompokkan menjadi empat tipe, yaitu teknik optik, ultrasonik, kapasitansi dan *thermal*. Kelemahan metode ini adalah hasil *scanning* sangat tergantung dari kualitas sidik jari. Jika kualitas sidik jari luka, kering, dan tidak jelas, maka kualitas hasil pembacaan tidak bagus. Kelemahan lain adalah teknik ini bisa diakali dengan jari palsu. Tapi teknik ini mempunyai keuntungan mudah dilakukan dan tidak membutuhkan biaya yang mahal.



Gambar 2.2 *Fingerprint Scanner dan Connector*

Pada Gambar 2.2 di atas merupakan contoh modul pemindai sidik jari yang dapat dikoneksikan dengan Arduino. Dengan DSP prosesor yang berkecepatan tinggi, modul *fingerprint scanner* ini juga dapat diaplikasikan dengan serial *device* yang lain, seperti MSP430, AVR, PIC, STM32, ARM dan FPGA *device*. Modul ini memiliki memori yang dapat menyimpan hingga 1000 data sidik jari. *Fingerprint scanner* memiliki kemampuan pembacaan sidik jari dengan tingkat sensitivitas yang tinggi baik dalam keadaan basah. Selain itu alat ini memiliki kecepatan tinggi saat melakukan sistem pemindaian, pencarian dan perbandingan pola sidik jari.

Dengan fitur - fitur dan segala keunggulan tersebut, *fingerprint scanner* ini dapat difungsikan dalam berbagai bidang, terutama yang bersangkutan dengan masalah keamanan. Dapat difungsikan sebagai piranti absensi modern, saklar elektronik maupun pengganti *password* dengan sensitivitas yang tinggi.

## 2.2 Arduino Mega 2560 [2]

Arduino Mega 2560 adalah board berbasis mikrokontroler pada ATmega 2560. Board ini memiliki pin I/O sejumlah 54 buah digital I/O pin (15 pin diantaranya adalah PWM), 16 pin analog input, 4 pin UART (serial port hardware). Arduino Mega 2560 dilengkapi dengan sebuah oscillator 16 Mhz, sebuah port USB, power jack DC, ICSP header, dan tombol reset. Spesifikasi tersebut dapat dilihat pada Tabel 2.1. Pin – pin ini berisi semua yang diperlukan untuk mendukung mikrokontroler, hanya terhubung ke komputer dengan kabel USB atau sumber tekanan bisa didapat dari adaptor AC – DC atau baterai untuk menggunakannya (Arduino, Inc., 2009). Gambar 2.3 seperti dibawah merupakan contoh board Arduino Mega 2560.



Gambar 2.3 Board Arduino Mega 2560

Tabel 2.1 Deskripsi Spesifikasi pada Arduino Mega 2560

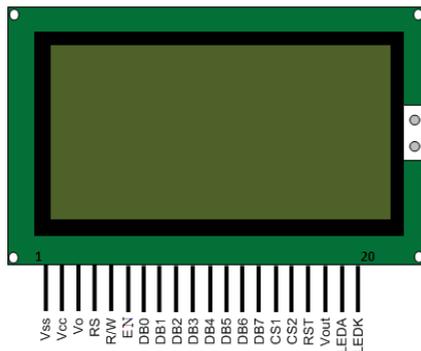
No	Deskripsi	Spesifikasi
1.	Mikrokontroler	ATmega 2560
2.	Tegangan Pengoperasian	5 V
3.	Batas tegangan yang disarankan	7 - 12 V
4.	Batas tegangan Input	6 - 20V
5.	Jumlah pin I/O digital	54 pin digital
6.	Jumlah pin input Analog	16 pin
7.	Arus DC tiap pin I/O	40 mA
8.	Arus DC untuk pin 3,3 V	50 mA

No	Deskripsi	Spesifikasi
9.	Memory Flash	256 KB, 8 KB digunakan oleh bootloader
10.	SRAM	8 KB
11.	EEPROM	4 KB
12.	ClocksPEED	16 MHz

### 2.3 *Graphic LCD* [3]

*GLCD* (*Graphic Liquid Crystal Display*) adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf, ataupun grafik. *GLCD* mempunyai pin data, kontrol catu daya, dan pengatur kontras tampilan. *GLCD* juga merupakan perangkat display yang paling umum dipasangkan di mikrokontroler, mengingat ukurannya yang kecil dan kemampuannya dalam menampilkan karakter atau grafik yang lebih dibandingkan dengan *seven segment*.

Salah satu jenis *GLCD* merupakan *GLCD*. Jenis *GLCD* ini masih berkembang saat ini. Resolusi *GLCD* ini bervariasi, diantaranya 128 x 64, 128 x 128. Bentuk *GLCD* beserta pin dapat dilihat pada Gambar 2.4. Pada table 2.1 menjelaskan tentang deskripsi *GLCD* sesuai *database*.



Gambar 2.4 Pin pada *GLCD*

Tabel 2.1 Deskripsi Pin *GLCD*

No.	Fungsi	Nama
1	<i>Ground</i> (0 V)	Vss

No.	Fungsi	Nama
2	<i>Supply voltage; 5V</i>	V <sub>cc</sub>
3	<i>Contrast adjustment</i>	V <sub>o</sub>
4	<i>High to display data; Low for instruction code</i>	<i>Register select (RS)</i>
5	<i>Low to write to the register; High to read from the register</i>	<i>Read/Write (R/W)</i>
6	<i>Reads data when High; Writes data at High to Low transition (falling edge)</i>	<i>Enable (EN)</i>
7		DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	<i>Chip selection for IC1; Active High</i>	CS1
16	<i>Chip selection for IC2; Active High</i>	CS2
17	<i>Reset signal; Active Low</i>	RST
18	<i>Output voltage for GLCD driving</i>	V <sub>out</sub>
19	<i>Backlight V<sub>CC</sub> (5V)</i>	LED A
20	<i>Backlight Ground (0V)</i>	LED K

#### 2.4 Keypad 4 x 4 [4]

Pada Gambar 2.5 merupakan contoh modul Keypad 4x4. Modul ini dapat difungsikan sebagai alat input dalam aplikasi-aplikasi seperti pengaman digital, datalogger, absensi, pengendali kecepatan motor, robotik, dan sebagainya. Mempunyai dimensi 9,3 cm (P) x 6,8 cm (L) x 1,5 cm (T). dan juga terdapat beberapa fitur sebagai berikut :

1. 16 tombol (dengan fungsi tergantung pada aplikasi),
2. Konfigurasi 4 baris (input scanning) dan 4 kolom (output scanning),
3. Kompatibel penuh dengan DT-51™ Low Cost Series dan DT-AVR Low Cost Series. Mendukung DT-51™ Minimum

System ver 3.0, DT-51™ PetraFuz, DT-BASIC Series, dan sistem lain.



Gambar 2.5 Keypad Module 4 x 4

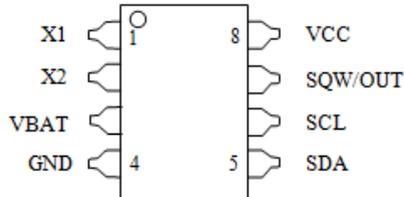
## 2.5 Real Time Clock (RTC) DS1307 [5]

*Real-Time Clock* disingkat RTC adalah jam di komputer yang umumnya berupa sirkuit terpadu yang berfungsi sebagai pemelihara waktu. RTC umumnya memiliki catu daya terpisah dari catu daya komputer (umumnya berupa baterai litium), sehingga dapat tetap berfungsi ketika catu daya komputer terputus. Kebanyakan RTC menggunakan *oscilator* kristal. Konfigurasi pin RTC dapat dilihat pada Gambar 2.6.

RTC tipe DS1307 merupakan jenis pewaktu yang menggunakan komunikasi serial untuk operasi tulis baca, dengan spesifikasi :

- *Real-Time Clock* (RTC) menyimpan data-data detik, menit, jam, tanggal, bulan, hari dalam seminggu, dan tahun valid hingga 2100.
- 56-byte, *battery-backed*, RAM *nonvolatile* (NC) RAM untuk penyimpanan
- Antarmuka serial *Two-wire* (12C)
- Sinyal luaran gelombang-kotak terprogram (*Programmable squarewave*)
- Deteksi otomatis kegagalan daya (*Power-fail*) dan rangkaian switch

- Konsumsi daya kurang dari 500nA menggunakan mode baterai cadangan dengan operasional osilator
- Tersedia fitus industri dengan ketahanan suhu : -40°C hingga +85°C
- Tersedia dalam kemasan 8-pin DIP atau SOI



Gambar 2.6 Konfigurasi Pin RTC DS1307

Tabel 2.2 Pin RTC DS1307

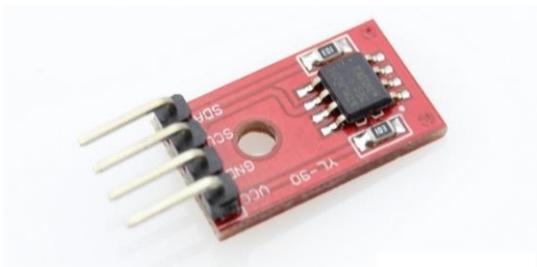
No.	Nama pin	Fungsi
1	X1	<i>Oscillator Crystal 32,768 KHz</i>
2	X2	
3	VBAT	<i>Battery Input (+3V)</i>
4	GND	<i>Ground</i>
5	SDA	<i>Serial Data</i>
6	SCL	<i>Serial Clock input</i>
7	SQW/OUT	<i>Square Wave/Output Driver</i>
8	VCC	<i>Supply Power</i>

## 2.6 EEPROM [6]

Serial EEPROM tipe 24xx adalah merupakan memori serial yang menggunakan teknologi I2C di mana dengan adanya penggunaan teknologi tersebut, jumlah I/O yang digunakan untuk mengakses memori tersebut semakin sedikit. Hal ini sangat bermanfaat bagi sebuah sistem yang memerlukan banyak I/O.

Penggunaan I/O yang semakin sedikit untuk mengakses memori, akan menyediakan lebih banyak I/O yang dapat digunakan untuk keperluan lain. I2C adalah teknologi komunikasi serial yang ditemukan oleh Philips pada tahun 1992 dan direvisi hingga versi 2.1 yang terbaru pada tahun 2000. Teknologi ini hanya menggunakan 2 buah jalur I/O

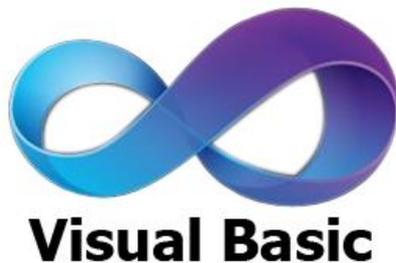
yaitu SDA dan SCL. Pada Tugas Akhir ini menggunakan modul EEPROM AT24C02 seperti pada Gambar 2.7.



Gambar 2.7 Modul EEPROM AT24C02

## 2.7 Visual Basic .Net 2010 [7]

Microsoft Visual Basic .NET adalah sebuah *software* untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem .NET *Framework*, dengan menggunakan bahasa BASIC. Pada Gambar 2.8 dapat dilihat logo dari Visual Basic. Dengan menggunakan alat ini para *programmer* dapat membangun aplikasi Windows Forms, aplikasi web berbasis ASP.NET, dan juga aplikasi *command-line*. Alat ini dapat diperoleh secara terpisah dari beberapa produk lainnya (seperti Microsoft Visual C++, Visual C#, atau Visual J#), atau juga dapat diperoleh secara terpadu dalam Microsoft Visual Studio .NET. Kelebihan dari Visual Basic.NET ialah mengatasi semua masalah yang sulit disekitar pengembangan aplikasi berbasis windows, windows Form designer memungkinka develover memperoleh aplikasi dekstop dalam waktu singkat dan menyediakan bagi Develover pemrograman data akses ActiveX Data Object(ADO).



Gambar 2.8 Logo Visual Basic

## 2.8 Software XAMPP [8]

XAMPP ialah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan campuran dari beberapa program dan logo XAMPP dapat dilihat pada Gambar 2.9. 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 mendownload langsung dari situs resminya.



Gambar 2.9 Logo XAMPP

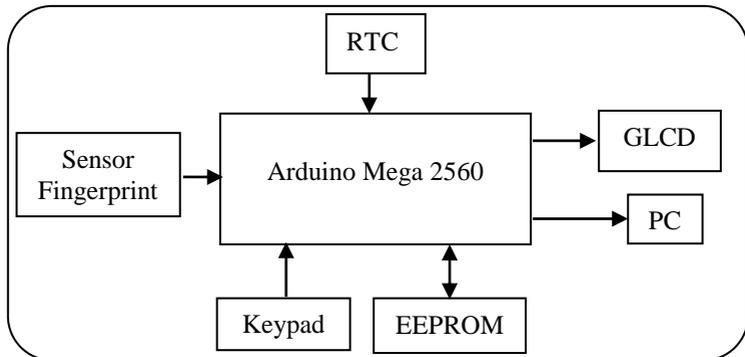
-----Halaman ini sengaja dikosongkan-----

## BAB III PERANCANGAN SISTEM

Pada bab ini membahas tentang tahapan perancangan dan pembuatan alat yang dilakukan terhadap Tugas Akhir yang berjudul “Monitoring dan Pelaporan Kehadiran Siswa Berbasis Website”. Perancangan dan pembuatan sistem meliputi perangkat keras dan perangkat lunak. Perangkat keras meliputi perancangan box *hardware*, rangkaian sensor *fingerprint*, rangkaian *GLCD*, rangkaian *RTC* (Real Time Clock), keypad dan pembuatan rangkaian seluruh komponen pada Shield Arduino Mega 2560. Perangkat lunak meliputi program *fingerprint*, tampilan *GLCD*, program *RTC*, *EEPROM* sebagai memori penyimpanan data *logger*, dan kontrol *keypad*.

### 3.1 Blok Fungsional Sistem

Berdasarkan tinjauan pustaka dan dasar teori pada bab II, maka dapat disimpulkan perancangan dan pembuatan sistem yang digunakan dengan menggunakan sebuah blok fungsional sistem yang menjelaskan hubungan fungsi antar komponen seperti pada Gambar 3.1.

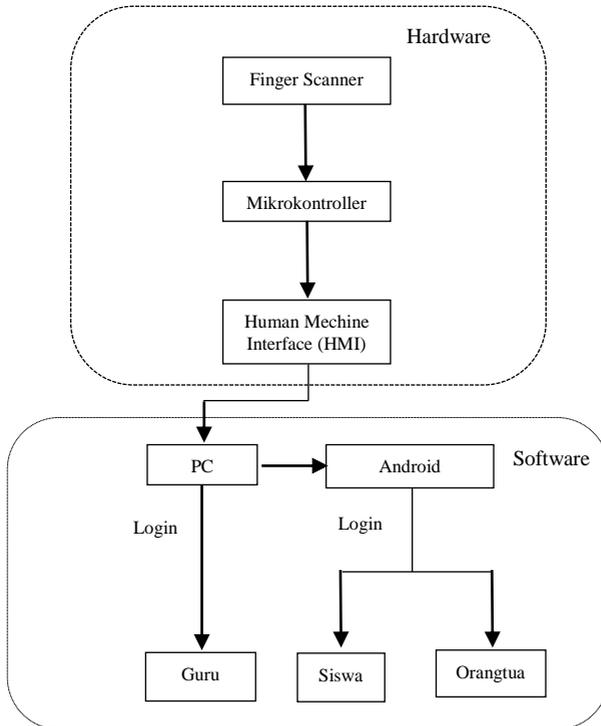


Gambar 3.1 Blok Fungsional Sistem

Berikut penjelasan mengenai blok diagram pada Gambar 3.1 yaitu:

1. Pemindai sidik jari (*fingerprint scanner*) digunakan untuk mengetahui kecocokan sidik jari pengguna dengan *database* yang ada di memori

2. Arduino Mega 2560 merupakan mikrokontroler yang berfungsi sebagai pengontrol perangkat elektronik dan dapat menyimpan program didalamnya.
3. *Keypad* digunakan sebagai *input* digit angka pada box.
4. EEPROM merupakan media penyimpanan data yang akan menyimpan data berupa ID, status sistem, jam dan tanggal.
5. RTC digunakan untuk mengatur waktu dan tanggal.
6. Tampilan *GLCD* berupa menu pilihan yang nantinya dapat membantu proses yang diinginkan menggunakan bantuan *keypad* 4x4.
7. Personal Computer (PC) digunakan sebagai alat bantu dalam menampilkan *database* yang telah dibuat.



Gambar 3.2 Diagram *Hardware* dan *Software*

Dari Diagram alur keseluruhan kerja sistem tersebut. Rencana Tugas Akhir ini penulis akan merancang serta membuat *hardware* berupa *prototype*, dan membuat dan menguji program *Fingerprint Scanner*, serta *interface* program *database* pada komputer. Sehingga pembuatan Android tidak dikerjakan karena telah dikerjakan oleh parter penulis.

Kontrol dari semua proses pada sistem Monitoring dan Pelaporan Kehadiran Siswa ini yaitu Mikrokontroler. Pada sistem ini dikendalikan secara otomatis dengan *Fingerprint scanner* dengan cara menempelkan pola jari yang telah terprogram pada Mikrokontroler. Jika pola jari yang dimasukkan melalui *Fingerprint scanner* benar, maka mikrokontroler akan mengelola data dan mengirimkannya ke *database* yang telah dibuat. EEPROM pada mikrokontroler juga merekap/menyimpan data berupa pola jari setiap siswa, status siswa, dan waktu. Waktu yang disimpan di dalam EEPROM itu karena adanya RTC (*Real Time Clock*) yang dapat menampilkan data jam dan tanggal secara *Real Time*. Namun, apabila pola jari yang dimasukkan melalui *Fingerprint scanner* salah, maka data diri siswa tidak akan terekap di *database*. Dan data berupa pola jari setiap siswa, status siswa, dan waktu, tetap akan terekap di dalam EEPROM. Data-data yang tersimpan di dalam EEPROM dapat dipindahkan ke *database* melalui program Visual Basic .Net yang terhubung dengan *Microsoft Access* yang tersimpan di dalam PC.

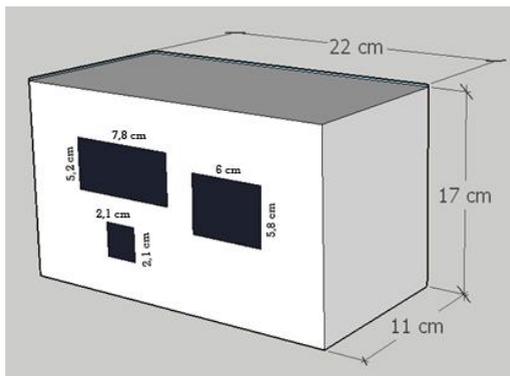
### **3.2 Perancangan Hardware**

Pada perancangan *hardware* dibagi menjadi beberapa sub bab yang akan dijelaskan per sub bab nya, antara lain : Perancangan Mekanik, Perancangan Rangkaian *Shield* Arduino Mega 2560, Rangkaian *GLCD 128 x 64*, Rangkaian *Power Supply*, Rangkaian RTC (*Real-Time Clock*), Rangkaian Keypad, Rangkaian EEPROM, dan Perancangan Rangkaian *Fingerprint Scanner*.

#### **3.2.1 Perancangan Mekanik**

Perangkat untuk mengetahui kehadiran siswa dengan *fingerprint scanner* tersebut merupakan alat untuk merekap data siswa yang telah hadir dan mengurangi resiko kecurangan membolos dengan tingkat kepraktisan yang tinggi. Secara umum perangkat keras ini terdiri dari *fingerprint scanner* sebagai masukan, *keypad* sebagai *input*

menu, penampil data berupa *Graphic LCD*, sistem mikrokontroler, box yang sudah didesain sedemikian rupa seperti Gambar 3.3 di bawah ini.

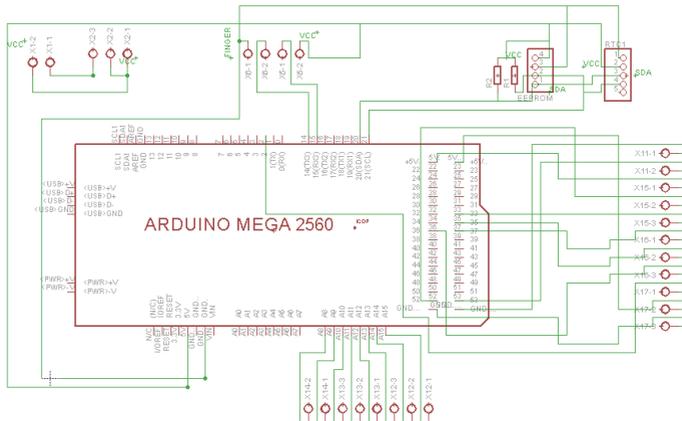


Gambar 3.3 Desain *Hardware*

### 3.2.2 Rangkaian Modul *Shield* Arduino Mega 2560

Pada perancangan rangkaian kontroler ini menggunakan Arduino Mega 2560 untuk menerima masukan data dari *Fingerprint scanner*, membaca data dari RTC (*Real-Time Clock*), membaca dan menulis data ke EEPROM, menampilkan tampilan ke *GLCD*, serta menjalankan program untuk komunikasi alat dengan PC. Arduino Mega 2560 diprogram agar dapat memproses data dari pola jari yang masuk oleh pengguna serta data ID dari pengguna untuk dijadikan *database*. Arduino Mega 2560 juga bertugas untuk menulis dan membaca data rekapan yang telah mengakses sistem tersebut. Rangkaian *Shield* Arduino Mega 2560 dapat dilihat pada Gambar 3.4. 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.



Gambar 3.4 Rangkaian *Shield* Arduino Mega 2560

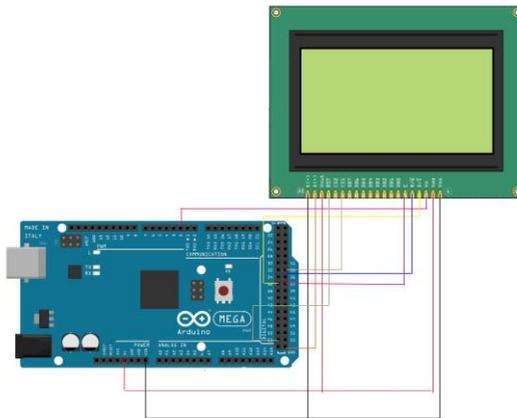
Tabel 3.1 Mapping Pin Analog/Digital Arduino Mega 2560

No	Modul	PIN ANALOG/DIGITAL
1	<i>FINGERPRINT SCANNER</i>	KABEL MERAH : VCC 5V KABEL HITAM : GND KABEL HIJAU : PIN 15 KABEL PUTIH : PIN 14
2	KEYPAD	PIN : A8 - A15 (Urut dari sebelah kiri ke kanan keypad 4x4)
3	<i>GRAPHIC LIQUID CRYSTAL DISPLAY 128X64</i>	VSS : GND VDD : VCC 5V VO: DIGITAL 2 RS: DIGITAL 36 RW: DIGITAL 35 E: DIGITAL 37 PSB : DIGITAL 33 RST : DIGITAL 52 VOUT : VCC 5V BLA : DIGITAL 53 BLK : GND

No	Modul	PIN ANALOG/DIGITAL
4	REAL TIME CLOCK	VCC : 5V GND : GND SDA : DIGITAL 20 SCL : DIGITAL 21
5	EEPROM	VCC : 5V GND : GND SDA : DIGITAL 20 SCL : DIGITAL 21

### 3.2.3 Perancangan Rangkaian *GLCD* 128x64

Rangkaian *GLCD* digunakan untuk membuat tampilan pada sistem. Seperti tampilan jam, tampilan data saat menginputkan pola jari dari *fingerprnt scanner*, serta tampilan data status siswa. Rangkaian *GLCD* dapat dilihat pada Gambar 3.5.

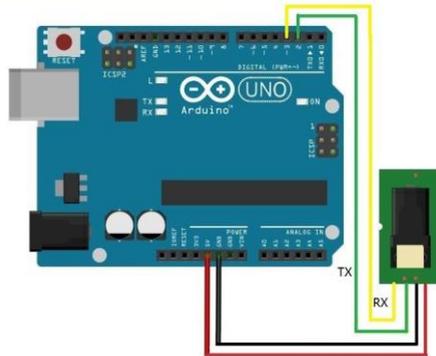


Gambar 3.5 Rangkaian *GLCD*

### 3.2.4 Perancangan Rangkaian *Fingerprint*

*Fingerprint scanner* digunakan untuk menginputkan data berupa pola jari atau sidik jari setiap individu siswa. Sidik jari dapat dibedakan menjadi beberapa tipe menurut Henry *Classification System*, yaitu *loop pattern*, *whorl pattern*, dan *arch pattern*. Pembacaan *fingerprint scanner* dilakukan dengan cara metode *scanning pattern*

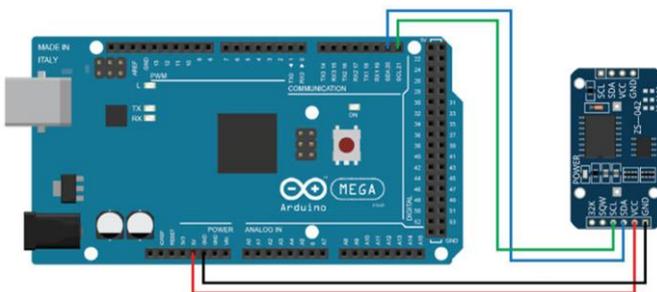
untuk mempermudah data yang tersimpan dan sesuai dengan biodata siswa. Fingerprint Scanner ini menggunakan komunikasi UART (TTL *Logic Level*), *Universal Asynchronous Receiver-Transmitter* adalah bagian perangkat keras komputer yang menerjemahkan antara bit-bit paralel data dan bit-bit serial. Rangkaian *Fingerprint scanner* dapat dilihat pada Gambar 3.6.



Gambar 3.6 Rangkaian *Fingerprint Scanner*

### 3.2.5 Perancangan Rangkaian RTC

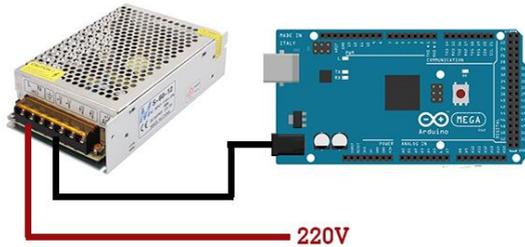
RTC (*Real-Time Clock*) digunakan untuk menyimpan dan menampilkan waktu dan tanggal secara *Real Time*. Pada RTC terdapat baterai 3V yang digunakan sumber daya cadangan saat *Power* utama dari rangkaian RTC mati, sehingga data waktu dan tanggal yang sedang berjalan tidak akan tereset kembali ke awal. Rangkaian RTC dapat dilihat pada Gambar 3.7.



Gambar 3.7 Rangkaian RTC

### 3.2.6 Perancangan Rangkaian *Power Supply*

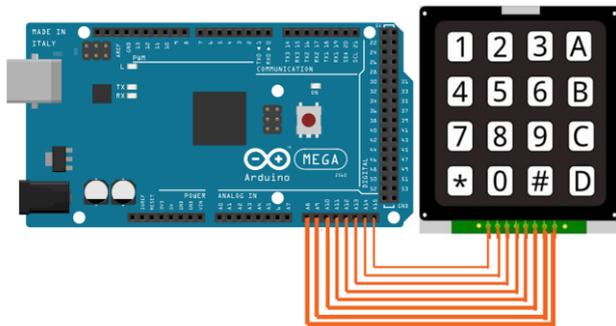
*Power Supply* digunakan untuk memberikan *input* tegangan pada keseluruhan komponen yang terdapat pada sistem alat ini, yaitu pada Arduino Mega 2860, *GLCD 128x64*, *RTC*, dan *EEPROM*. Rangkaian ini dapat menurunkan tegangan 220V menjadi 12V yang dapat mensuplai *power* Arduino Mega 2560.



Gambar 3.8 Rangkaian *Power Supply*

### 3.2.7 Rangkaian *Keypad*

*Keypad 4x4* digunakan sebagai *input* data berupa angka yang nantinya akan digunakan sebagai inisial setiap siswa. Penekanan *keypad* dapat disimulasikan dengan penyesuaian tombol *keypad* dan data yang muncul. Rangkaian *keypad 4x4* dapat dilihat pada Gambar 3.9.

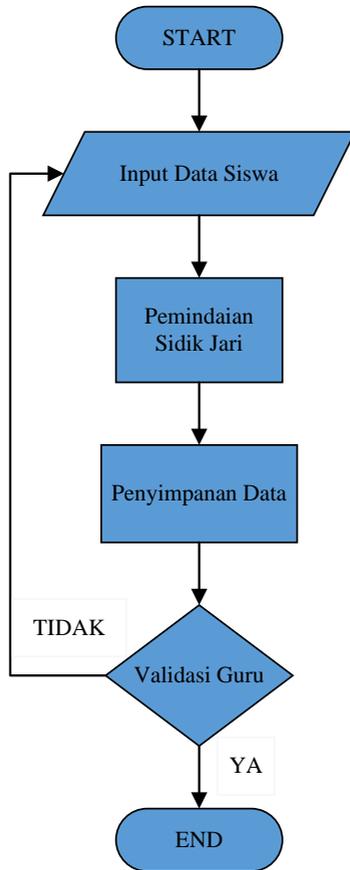


Gambar 3.9 Rangkaian *Keypad 4x4*

### 3.3 Perancangan 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 dalam bahasa C menggunakan IDE Arduino dan program Visual Basic .Net sebagai tampilan *database* yang telah direkap oleh sistem. 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. Pemrograman arduino IDE ini juga meliputi program *fingerpint*, program RTC (*Real Time Clock*), program keypad dan EEPROM 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 Visual Basic .Net dilakukan setelah perancangan program Arduino IDE sudah memenuhi kriteria data yang dihasilkan. Hal ini mempengaruhi pengiriman data dari *hardware* yang akan ditampilkan pada *software* Visual Basic .Net. Perancangan program Visual Basic .Net ini dibuat dengan 2 program utama, yaitu program pengiriman data dan program pada tampilan yang akan dibuat. Setiap pilihan menu atau tombol pada tampilan Visual Basic .Net terdapat program yang akan melakukan tindakan apabila menu atau tombol itu digunakan. Program pada Visual Basic .Net juga menggunakan program yang terkoneksi pada *software* XAMPP agar dapat ditampilkan di-*localhost* yang nantinya *database* dapat dilihat pada tampilan *Website*.

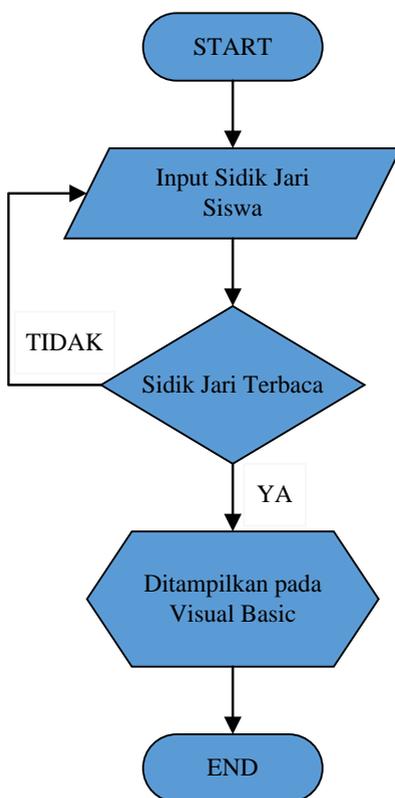


Gambar 3.10 Diagram Pendaftaran Data Siswa

Pada saat siswa akan melakukan proses absensi disekolah, pastikan siswa telah melakukan pendaftaran data diri yang prosesnya dapat dilihat pada Gambar 3.10. Siswa akan mengisi data diri sesuai dengan ID atau absen siswa. Setelah itu siswa didaftarkan oleh guru agar tidak terjadi kesalahan dalam pendaftaran data diri berupa nomer absen dan sidik jari. Ketika proses pembacaan sidik jari, data masukkan berupa ID (nomer absen siswa) dan sidik jari akan tersimpan secara otomatis. Proses pendfataran data siswa telah dilakukan sesuai prosedur dan divalidasi oleh guru.

### 3.3.1 Arduino IDE

Arduino IDE adalah software yang diperlukan oleh board arduino. Software ini arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan 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 3.11 dimana alur tersebut memperlihatkan proses awal hingga akhir program.



Gambar 3.11 *Flowchart* pada Sistem Absensi Siswa

Setelah melakukan pendaftaran data diri berupa ID (nomor absen) dan sidik jari, siswa telah dapat mengakses sistem ini. Ketika siswa akan masuk sekolah, siswa diwajibkan melakukan absensi secara *fingerprint*. Apabila dalam proses pembacaan sidik jari tidak bisa atau data yang telah dimasukkan salah, maka proses akan terhenti dan mengulang kembali seperti awal. Hal itu akan terjadi terus apabila ada kesalahan yang mengakibatkan sidik jari tidak terbaca. Ketika proses sidik jari berhasil, data akan dibaca oleh Arduino Mega 2560 dan akan dikirim ke komputer guru. Setelah dilakukan validasi oleh guru, maka siswa itu telah terbukti masuk sekolah dan tercantum namanya di *database* yang telah dibuat.

#### A. Program *Fingerprint*

Program *fingerprint* ini terdiri dari beberapa bagian. Setiap bagian memiliki program dan kegunaan masing masing sesuai dengan kerja sistem. Ada 3 bagian program yang digunakannya, diantaranya adalah *Fingerprint Acces*, *Enrollment*, dan *Delete Data*. Setiap bagian program akan dijelaskan tahapannya.

##### *Fingerprint Acces*

Inisialisasi penggunaan *library* dan pin serial pada Arduino Mega 2560 dapat dilihat pada Gambar 3.12. Berdasarkan gambar tersebut maka dapat diketahui bahwa pin untuk sensor fingerprint menggunakan komunikasi serial melalui pin 15 sebagai RX yang disambungkan pada kabel putih sensor (TD) dan pin 14 sebagai TX yang disambungkan pada kabel hijau sensor (RD).

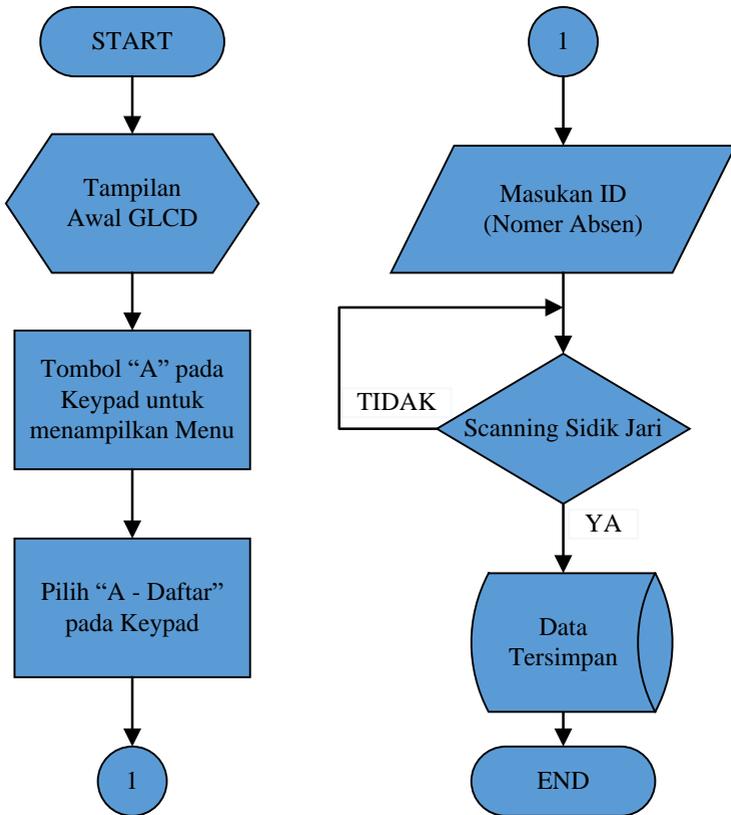
```
fingerprint_TA $
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>

int getFingerprintIDez();

// pin #14 is IN from sensor (GREEN wire)
// pin #15 is OUT from arduino (WHITE wire)
//SoftwareSerial mySerial(14, 15);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial3);

// On Leonardo/Micro or others with hardware serial, use those!
//Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial1);
```

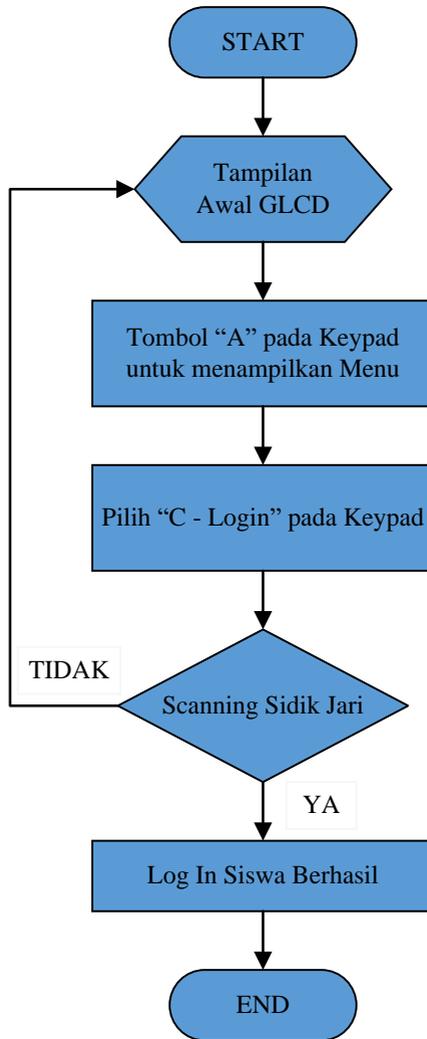
Gambar 3.12 Inisialisasi *Fingerprint Scanner*



Gambar 3.13 *Flowchart* Daftar Sidik Jari

Pada Gambar 3.13 Dapat dilihat *flowchart* tentang proses pendaftaran sidik jari. Dari gambar tersebut dapat dilihat langkah-langkah untuk mendaftarkan sidik jari siswa. Data itu akan tersimpan sesuai saat melakukan pendaftaran.

*Flowchart log in* sidik jari dapat dilihat pada Gambar 3.14. Dari gambar tersebut dapat dilihat langkah – langkah proses melakukan *log in* atau absen siswa sekolah. Berawal dari tampilan *GLCD* dan dilanjutkan ke pemilihan menu. Sampai melakukan pemindaian sidik jari dan data *log in* terekap.



Gambar 3.14 Flowchart Log In Sidik Jari Siswa

*Enrollment*

Sistem ini didesain agar hanya bisa diakses oleh orang yang telah didaftarkan oleh admin atau guru. Untuk melakukan pendaftaran

user baru, admin harus menjalankan program *enrollment* sidik jari menggunakan program seperti pada Gambar 3.15.

```
while (! getFingerprintEnroll() );
)

uint8_t getFingerprintEnroll() {
    int p = -1;
    Serial.print("Waiting for valid finger to enroll as #"); Serial.pr:
    while (p != FINGERPRINT_OK) {
        p = finger.getImage();
        switch (p) {
            case FINGERPRINT_OK:
                Serial.println("Image taken");
                break;
            case FINGERPRINT_NOFINGER:
                Serial.println(".");
                break:

```

Gambar 3.15 Program *Enrollment*

Pada *listing* program Gambar 3.15 dapat dilihat terdapat perintah `finger.getImage` yang digunakan untuk memindai sidik jari yang akan didaftarkan. Setelah proses tersebut dilakukan, apabila sidik jari telah terpindai maka perintah `FINGERPRINT_OK` dan apabila sidik jari tidak dapat terpindai maka `FINGERPRINT_NOFINGER` yang menyatakan sidik jari tidak terbaca.

#### Kecocokan Sidik Jari

Program digunakan untuk mengetahui kecocokan sidik jari dan ID yang telah didaftar. Program ini dapat mengetahui kecocokan sidik jari dengan ID menggunakan skala kecocokan 1 – 100, dimana semakin tinggi hasilnya maka persentase kecocokan sidik jari dengan ID semakin tinggi. Pada Gambar 3.16 dapat dilihat program kecocokan sidik jari, dimana terdapat perintah `fingerpacketreceiveerr` untuk menyambungkan antara *fingerpint sensor* dengan tampilan serial monitor. Perintah `serial print found ID` untuk menampilkan ID yang terpindai dan `serial print confidence` akan menampilkan hasil persentasi kococokan sidik jari dengan ID yang telah terdaftar dari skala 1 sampai 100.

```

fingerprint_TA$
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
    Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_NOTFOUND) {
    Serial.println("Did not find a match");
    return p;
} else {
    Serial.println("Unknown error");
    return p;
}

// found a match!
Serial.print("Found ID #"); Serial.print(finger.fingerID);
Serial.print(" with confidence of "); Serial.println(finger.confidence);

```

Gambar 3.16 Program Kecocokan Sidik Jari

### Delete Data

Program untuk penghapusan sidik jari untuk user atau siswa yang telah tidak terpakai atau sudah pindah kelas maupun sudah lulus, dapat dilihat pada Gambar 3.16 *listing* program *delete*. Pada saat penghapusan ID, admin atau guru akan memasukkan nomer ID yang akan dihapus. Setelah user ID dihapus dari memori, maka user tersebut tidak dapat mengakses sistem keamanan menggunakan *fingerprint scanner* ini. Pada Gambar 3.17 terdapat perintah `deleteFingerprint(uint8_t id)` yang digunakan menghapus ID yang telah terdaftar dan telah dipilih sebelumnya. Pemilihan ID yang dihapus sesuai dengan perintah `finger.deleteModel(id)`. Apabila proses menghapus ID telah berhasil dilkauan maka akan keluar perintah Deleted!. Program *Delete* ini hanya dapat dilakukan untuk menghapus ID yang terdaftar secara satu persatu. Hal ini agar dapat mengurangi kesalahan dalam proses penghapusan data sidik jari siswa. ID yang telah dihapus dapat digunakan kembali dengan menggunakan sidik jari yang sama ataupun diganti dengan sidik jari lainnya. Proses penggunaan ID yang telah dihapus akan merujuk pada prose *enrollment* atau pendaftaran sidik jari. Jadi proses tersebut sama seperti proses saat melakukan pendaftaran sidik jari.

```

        deleteFingerprint(id);
    }

    uint8_t deleteFingerprint(uint8_t id) {
        uint8_t p = -1;

        p = finger.deleteModel(id);

        if (p == FINGERPRINT_OK) {
            Serial.println("Deleted!");
        } else if (p == FINGERPRINT_PACKETRECEIVEERR) {
            Serial.println("Communication error");
            return p;
        } else if (p == FINGERPRINT_BADLOCATION) {
            Serial.println("Could not delete in that locati
            return p;

```

Gambar 3.17 Program *Delete Data*

#### B. Program RTC

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. Program RTC dapat dilihat pada Gambar 3.18. Gambar dibawah memperlihatkan perintah `DateTime(F(_DATE_),F(_TIME_))` yang digunakan untuk menampilkan waktu sesuai saat peng-*upload*-an program.

```

rtc.begin();
rtc.adjust(DateTime(F(__DATE__),F(__TIME__)));

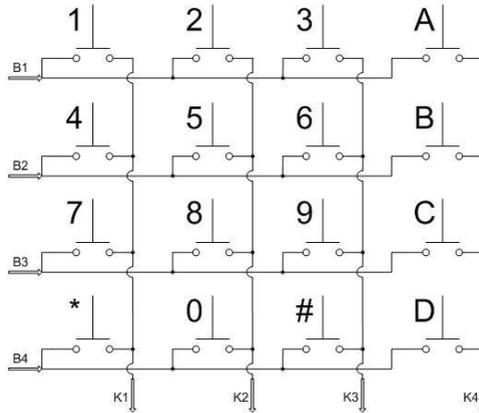
```

Gambar 3.18 Program RTC

#### C. Program *Keypad*

*Keypad* adalah bagian penting dari suatu perangkat elektronika yang membutuhkan interaksi manusia. *Keypad* berfungsi sebagai *interface* antara perangkat (mesin) elektronik dengan manusia atau dikenal dengan istilah HMI (*Human Machine Interface*). Konfigurasi keypad dengan susunan bentuk matrix 4×4 bertujuan

untuk penghematan port Arduino Mega 2560 karena jumlah *key* (tombol) yang dibutuhkan banyak pada suatu sistem dengan mikrokontroler. Konstruksi atau rangkaian matrix keypad 4×4 untuk mikrokontroler dapat dibuat seperti pada Gambar 3.19.



Gambar 3.19 Konstruksi Keypad 4x4

Konstruksi matrix keypad 4×4 diatas cukup sederhana, yaitu terdiri dari 4 baris dan 4 kolom dengan keypad berupa saklar *push button* yang diletakan disetiap persilangan kolom dan barisnya. Rangkaian matrix keypad diatas terdiri dari 16 saklar *push button* dengan konfigurasi 4 baris dan 4 kolom. 8 line yang terdiri dari 4 baris dan 4 kolom tersebut dihubungkan dengan port Arduino Mega 2560. Sisi baris dari matrix keypad ditandai dengan nama B1, B2, B3 dan B4 kemudian sisi kolom ditandai dengan nama K1, K2, K3 dan K4. Sisi input atau output dari matrix keypad 4×4 ini tidak mengikat, dapat dikonfigurasikan kolom sebagai input dan baris sebagai output atau sebaliknya tergantung programernya. Penggunaan keypad 4×4 pada sistem ini memiliki beberapa fungsi diantaranya pilihan Menu dan input data dalam bentuk angka. Inialisasi keypad 4×4 dapat dilihat pada Gambar 3.20. Inialisasi keypad 4×4 menggunakan pinMode A8 sampai A15 dengan dibagi antara pin input dan pin output pada Arduino Mega 2560. Pada perintah `digitalWrite` semua pin diberi logic

High agar dapat memberi proses ke Arduino Mega 2560 dan sesuai dengan konstruksi *keypad* 4×4.

```
void keypad_init() //inisialisasi pin input dan output untuk keypad
{
  pinMode(A8, INPUT);
  pinMode(A9, INPUT);
  pinMode(A10, INPUT);
  pinMode(A11, INPUT);

  pinMode(A12, OUTPUT);
  pinMode(A13, OUTPUT);
  pinMode(A14,
  OUTPUT);
  pinMode(A15, OUTPUT);

  digitalWrite(A8, HIGH);
  digitalWrite(A9, HIGH);
  digitalWrite(A10, HIGH);
  digitalWrite(A11, HIGH);

  digitalWrite(A12, HIGH);
  digitalWrite(A13, HIGH);
  digitalWrite(A14, HIGH);
  digitalWrite(A15, HIGH);
}
```

Gambar 3.20 Inisialisasi *Keypad* 4x4

#### D. Program *GLCD*

Suatu modul yang berfungsi sebagai *display* yang dapat menampilkan gambar yang memiliki resolusi 128 kolom dan 64 baris. *GLCD* ini memiliki warna dasar biru dan karakter berwarna putih dengan menggunakan backlight. Pada umumnya *GLCD* atau LCD Grafik ini terdapat dua tipe, ST7920 dan KS0108. Kali ini menggunakan LCD Grafik berbasis ST7920 dengan supply tegangan sebesar 5V DC. *LCD Graphic* ini memiliki 20 pin, diantaranya :

1. VSS
2. VDD
3. V0
4. RS
5. R/W
6. E
7. D0

8. D1
9. D2
10. D3
11. D4
12. D5
13. D6
14. D7
15. PSB
16. NC
17. RST
18. Vout
19. A
20. K

Hal yang paling umum dan mudah membedakan *GLCD* atau LCD Grafik ST7920 dan KS0108 yaitu terdapat pin NC dan PSB pada *GLCD* ST7920, pada *GLCD* KS0108 yaitu terdapat pin D/I, CS1 dan CS2. Program *GLCD* dapat dilihat pada Gambar 3.21. Pada contoh listing program dibawah dapat dilihat agar menghasilkan tampilan awal pada *GLCD*.

```

void drawMenu(int stat) //tampilkan menu ke glcd
{
    char c, buff[20], *temp;
    DateTime now = rtc.now();
    switch(stat)
    {
        case 0:
            u8g.drawRFrame(0,0,128, 64, 0);
            u8g.drawStr(7, 13, "SELAMAT DATANG");
            sprintf(buff, "%02d:%02d:%02d ", now.hour(), now.minute(), now.second());
            u8g.drawStr(7, 44, buff);
            sprintf(buff, "%02d:%02d:%02d ", now.day(), now.month(), now.year());
            u8g.drawStr(7, 59, buff);
            break;
    }
}

```

Gambar 3.21 Program *GLCD*

#### E. Program EEPROM

EEPROM disini digunakan untuk menyimpan data dari pembacaan ID pengguna, *password*, dan waktu dari pengaksesan sistem itu sendiri. Program EEPROM dapat dilihat pada Gambar 3.22.

```
void EEPROMWrite(byte wordaddress, byte data)
{
  Wire.beginTransmission(AT24C02);
  Wire.write(wordaddress);
  Wire.write(data);
  Wire.endTransmission();
  delay(5);
}

byte EEPROMRead(byte wordaddress)
{
  byte result;

  Wire.beginTransmission(AT24C02);
  Wire.write(wordaddress);
  Wire.endTransmission();
  Wire.requestFrom(AT24C02, 1);
  if(Wire.available()) result = Wire.read();
  Wire.endTransmission();
  return result;
}
```

Gambar 3.22 Program EEPROM

### 3.3.2 Perancangan Database

Program Visual Basic digunakan sebagai *database* siswa yang telah hadir sekolah. *Software* ini dapat menampilkan data yang telah diproses oleh Arduino Mega 2560 dan dikirim melalui kabel Serial. Hal ini mempermudah guru sekolah untuk memverifikasi siswanya yang telah hadir.

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 riwayat siswa yang melakukan absensi. Data ini menampilkan nomor ID, *password*, jam, tanggal, status berhasil atau tidak, dan keterangan. Dalam membuat *database* ini, menggunakan *Software* yang dinamakan Visual Basic .Net.

Hal hal yang harus disiapkan dalam membuat *database* atau basis data adalah *Hardware*, *Software*, data dan *user*. *Hardware* sebagai perangkat standar untuk pengolahan data dan menampilkan

data. *Software* sendiri dibagi menjadi 3 bagian. *Software* Sistem Operasi berfungsi untuk melakukan operasi dasar pada komputer dan mengendalikannya. *Software* Pengelola Basis Data untuk memasukkan data dan mengambil data dari database. Dan *Software* Pembuat *Database* yang dapat digunakan untuk menampilkan data sesuai keinginan pengguna. Data yang akan disimpan secara sistematis dan akan ditampilkan pada komputer atau perangkat yang lain. User atau pengguna adalah orang yang akan menggunakan database, dan user sendiri terbagi menjadi database manager, database administrator, dan database user.

Pada perancangan *database* ini dilakukan beberapa pemrograman, diantaranya pemrograman Visual Basic dengan koneksi Arduino Mega 2560, program koneksi data *input* dengan *database* MS. Access, dan program *export* untuk menampilkan data pada localhost. Pemrograman diberikan tahapan dalam mengelola data yang akan dijadikan *database*. Perancangan program ini dapat membantu untuk mengoneksikan antara *hardware* dan *software*, maupun membantu mengoneksikan antar *software* satu dengan *software* lainnya. Penggunaan program – program ini akan menyesuaikan hasil dari data yang dikeluarkan oleh *hardware* atau *output* Arduino. Dengan hasil data yang dapat dilihat pada serial monitor Arduino, pemrograman dapat menyesuaikan dengan hasil tersebut agar kolom *database* yang telah dibuat akan sesuai.

Pada Gambar 2.23 dapat dilihat program koneksi antara Visual Basic dengan Arduino. Program ini digunakan agar data yang dihasilkan Arduino dapat dikirim dan dikelola oleh software Visual Basic. Hasil dari data yang telah dikelola dapat dilihat pada tampilan awal aplikasi yang telah dibuat pada Visual Basic. Terdapat berbagai *icon* yang diprogram seperti tulisan *connected* atau *disconnected*.

Pada Gambar 2.24 dapat dilihat program koneksi database dari MS. Access. Dari koneksi hardware data dikirim dan akan disimpan pada MS. Access, setelah itu akan dipanggil atau ditampilkan pada Visual Basic. Tampilan akan menyesuaikan juga dengan data input siswa pada Visual Basic.

Pada Gambar 2.25 dapat dilihat program *Export* ke Localhost. Tampilan *database* yang telah dibuat dan data yang telah ditampilkan akan di-*export* ke localhost dengan menekan *icon Export* pada Visual Basic. Hasil dari export dapat dilihat pada localhost yang telah dibuat sebelumnya.

```

Public Sub konekserial(ByVal Com As String)
    Try
        With Me.serial_main
            .PortName = Com
            .BaudRate = 9660
            .ReadBufferSize = 500
            .Parity = IO.Ports.Parity.None
            .DataBits = 8
            .StopBits = 1
            .Handshake = IO.Ports.Handshake.None
        End With

        serial_main.Open()
        Timer1.Enabled = True

        System.Threading.Thread.Sleep("100")
        pic_connection.Visible = False
        pic_connection_on.Visible = True
        txt_connection.Text = "CONNECTED"
        txt_connection.ForeColor = Color.Green
        Me.Show()
        Connection.Hide()

    Catch ex As Exception
        MessageBox.Show("Serial Not Ready")
        pic_connection.Visible = True
        pic_connection_on.Visible = False
        Connection.Show()
        txt_connection.Text = "DISCONNECTED"
        txt_connection.ForeColor = Color.Red
        Timer1.Enabled = False
    End Try
End Sub

```

Gambar 2.23 Program Koneksi Visual Basic dengan Arduino

Pada koneksi antara Visual Basic dengan Arduino ini digunakan agar dapat mengirim data yang diperlukan dalam membuat *database* absensi kehadiran siswa. Hal ini dilakukan agar antara koneksi hardware dengan software dapat sesuai yang diinginkan, karena program ini menjadi awal koneksi dan pengolahan data pada sistem monitoring dan kehadiran siswa. Untuk mempermudah koneksi ini digunakan kabel serial agar dapat langsung terhubung antara komputer dan Arduino.

```

Sub koneksi_database_access()
    dirDB = mydocumentfolderDatabase & file
    provider = "Provider=Microsoft.ACE.OLEDB.12.0;"
    dbsource = "Data Source=" & dirDB
    con = New OleDbConnection(provider & dbsource)
    If con.State = ConnectionState.Closed Then
        con.Open()
    End If

    If con.State = ConnectionState.Open Then
        monitoring.lbl_koneksi_database.Text = "OK"
        update_database()
    Else
        monitoring.lbl_koneksi_database.Text = "FAIL"
    End If
End Sub

Private Sub PictureBox2_Click_1(ByVal sender As Object, ByVal e As EventArgs)
    koneksi_database_access()
    btExport.Enabled = True
End Sub

```

Gambar 2.24 Program Koneksi *Database* MS. Access

```

Public Sub Koneksi2()
    CONN = New OdbcConnection("dsn=conn_absen")
    CONN.Open()
    Str = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" & Application.StartupPath & "\Database"
    Conn2 = New OleDbConnection(Str)
    Conn2.Open()
End Sub

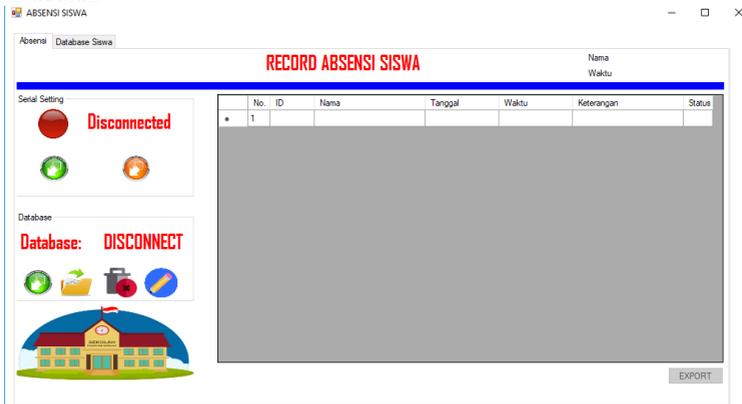
Private Sub btExport_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    With DataGridView1
        Call Koneksi2()
        For i As Integer = 0 To .Rows.Count - 2
            CMD = New OdbcCommand("Insert Into db_absen (WAKTU,TGL,ID_SISWA,
            '(' & DataGridView1.Rows(i).Cells(4).Value & _
            ',' & DataGridView1.Rows(i).Cells(3).Value & _
            ',' & DataGridView1.Rows(i).Cells(1).Value & _
            ',' & DataGridView1.Rows(i).Cells(6).Value & _
            ',' & DataGridView1.Rows(i).Cells(5).Value & _
            ')", CONN)
            DR = CMD.ExecuteReader
            DR.Read()
        Next
        MsgBox("Sukses!")
        btExport.Enabled = False
    End With
End Sub

```

Gambar 2.25 Program *Export* ke Localhost

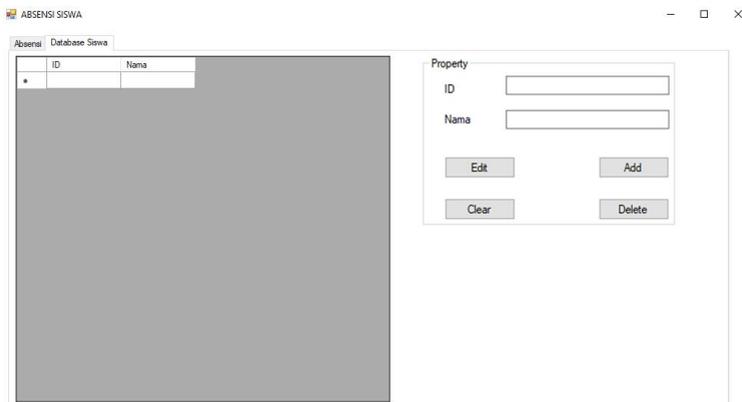
Setiap aplikasi biasanya memiliki jendela atau *Background interface*, yang dalam bahasa pemrograman yang berbasis visual, biasa

disebut dengan *Form*. *Form Designer* ini berfungsi sebagai tempat untuk mendesain *form* untuk aplikasi yang akan kami buat, dan juga sebagai tempat untuk meletakkan komponen-komponen yang kami ambil dari *component palette*. Tampilan *designer* pada *database* sistem ini adalah :



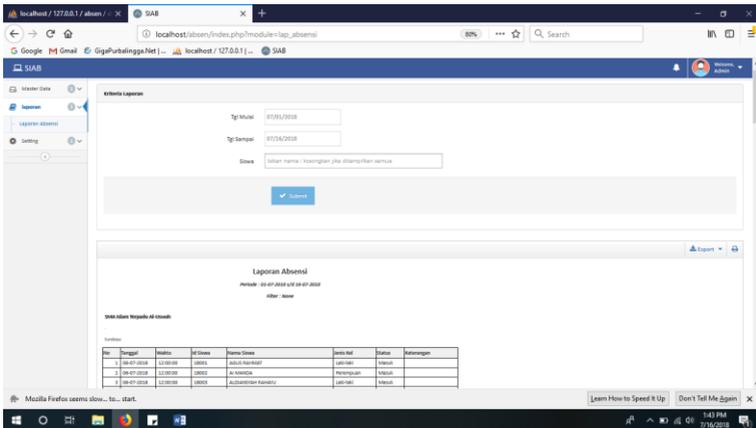
Gambar 3.26 Tampilan *Software* Visual Basic

Pada Gambar 3.26 dapat dilihat tampilan *database* yang ada didalam *software* Visual Basic. Terdapat beberapa kolom yang memiliki fungsi masing masing, seperti no urut, ID siswa, nama siswa, tanggal, waktu, keterangan dan status.



Gambar 3.27 Tampilan *Input* Data Siswa

Pada Gambar 3.27 dapat dilihat tampilan database siswa yang digunakan untuk menyesuaikan input data berupa ID dan nama siswa. Terdapat tombol *Edit* yang digunakan untuk mengubah data ID maupun nama, tombol *Add* yang digunakan untuk menambahkan *database* siswa, tombol *Clear* yang digunakan untuk menghapus semua *database* siswa dan tombol *Delete* yang digunakan untuk menghapus data yang dipilih.



Gambar 3.28 Tampilan *Database* pada Website

```
CREATE TABLE `db_absen` (
  `ID` int(9) NOT NULL,
  `TGL` date NOT NULL,
  `WAKTU` time NOT NULL,
  `ID_SISWA` int(5) NOT NULL,
  `STATUS` int(2) NOT NULL COMMENT '1:Masuk
2:Alpa 3:Izin',
  `KETERANGAN` text NOT NULL,
  `ID_USER` int(2) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Gambar 3.29 Program *Database* pada Website

*Database* pada Visual Basic akan di-*export* ke MySQL dan akan ditampilkan pada Website. Tampilan *database* pada Website

dapat dilihat pada Gambar 3.28 dimana database tersebut telah dilakukan verifikasi oleh guru. Sehingga tampilannya dibuat lebih mudah dibaca. Untuk pemograman MySQL dapat dilihat pada program utama Gambar 3.29. Gambar 3.29 program create table db\_absen untuk membuat tabel kolom pada database absen.

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

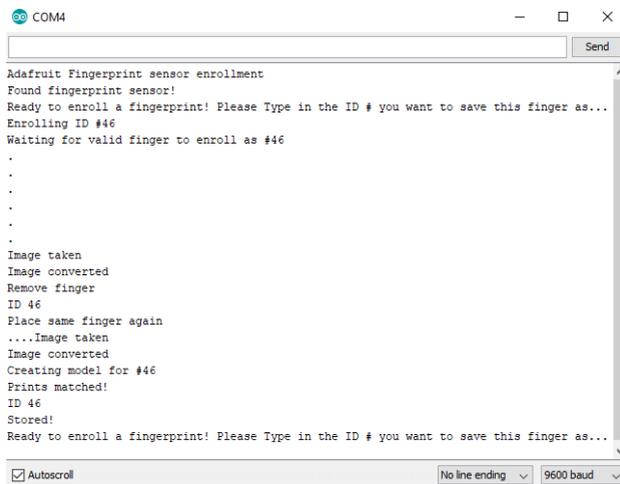
## BAB IV

### Pengujian dan Analisis

Pada bab ini membahas tentang pengujian alat “Monitoring dan Pelaporan Kehadiran Siswa Berbasis Website”. Pengujian ini dilakukan dari sebagian komponen utama yang ada pada alat. Pengujian dan analisis alat terbagi menjadi beberapa bagian, diantaranya pengujian *Fingerprint Scanner*, pengujian RTC (*Real Time Clock*), pengujian GLCD (*GRAPHIC LIQUID CRYSTAL DISPLAY*), pengujian keypad, pengujian *database* dan pengujian keseluruhan sistem alat.

#### 4.1 Pengujian *Fingerprint Scanner*

*Fingerprint scanner* diuji dengan menggunakan program yang diupload ke dalam Arduino. Ada beberapa macam program yang digunakan untuk menguji *fingerprint scanner*, diantaranya pendaftaran sidik jari (*enrollment*), pengujian kecocokan sidik jari, pengujian *delete* ID dan Data dari serial monitor.

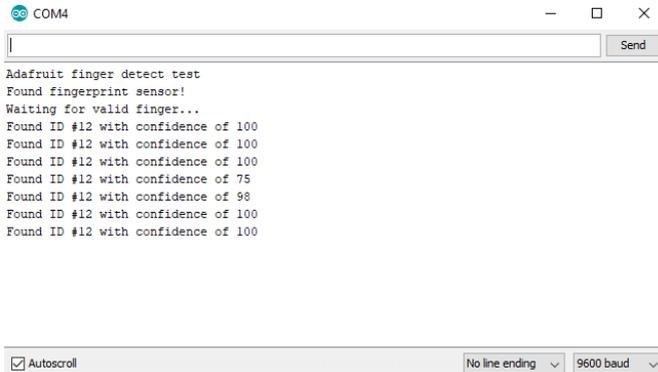


```
COM4
Adafruit Fingerprint sensor enrollment
Found fingerprint sensor!
Ready to enroll a fingerprint! Please Type in the ID # you want to save this finger as...
Enrolling ID #46
Waiting for valid finger to enroll as #46
.
.
.
.
.
.
Image taken
Image converted
Remove finger
ID 46
Place same finger again
...Image taken
Image converted
Creating model for #46
Prints matched!
ID 46
Stored!
Ready to enroll a fingerprint! Please Type in the ID # you want to save this finger as...
Autoscroll
No line ending
9600 baud
```

Gambar 4.1 Pengujian *Enrollment* pada *Fingerprint*

Pada pengujian Gambar 4.1 menunjukkan bahwa saat sidik jari telah berhasil mendapatkan image sidik jari maka sidik jari tersebut

telah terdaftar dan tersimpan sesuai dengan ID yang dimasukkan. Untuk mengetahui keakuratan pembacaan sensor fingerprint maka diperlukan pengujian *log in* dengan mencantumkan nilai *confidence* untuk mengetahui seberapa besar keakuratan posisi sidik jari.



Gambar 4.2 Pengujian Kecocokan Sidik Jari

Hasil pengujian pada Gambar 4.2 menunjukkan kecocokan sidik jari dengan *fingerprint scanner*. Semakin tinggi nilai *confidence* pada serial monitor, maka semakin tinggi pula tingkat kecocokan sidik jari tersebut. Tingkat kecocokan dapat ditentukan oleh posisi sidik jari, kondisi sidik jari dan lain sebagainya, tergantung jenis fingerprint scanner yang digunakan.

Pada Gambar 4.2 dapat dihitung persentase error yang didapat dari data pengujian kecocokan sidik jari, dengan menggunakan rumus sebagai berikut :

$$\text{Persentase Error} : \frac{(\text{Nilai Sebenarnya}) - (\text{Nilai Terbaca})}{(\text{Nilai Sebenarnya})} \times 100\%$$

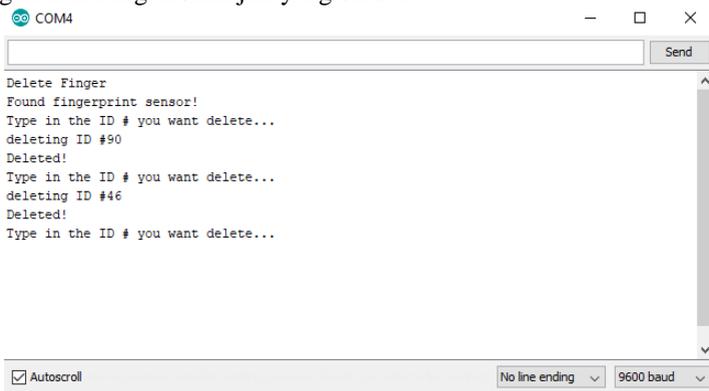
Dari Gambar 4.2 telah dilakukan pengujian sebanyak 7 kali atau total dari nilai sebenarnya adalah 700 dan nilai yang terbaca dari 7 kali pengujian adalah 671. Perhitungan persentase kesalahan dari pengujian sidik jari ID #12 adalah :

$$\text{Persentase Error} = \frac{(700) - (673)}{(700)} \times 100\%$$

$$\begin{aligned}
&= \frac{(27)}{(700)} \times 100\% \\
&= 0,038 \times 100\% \\
&= 3,8\%
\end{aligned}$$

Hasil dari perhitungan persentase error diatas menunjukkan bahwa fingerprint scanner terdapat persentase error sebesar 3,8%. Sehingga keakuratan pembacaan sensor fingerprint sebesar 96,2%.

Pengujian program delete dapat dilihat pada Gambar 4.3. Pengujian ini dilakukan dengan cara memilih ID yang telah terdaftar sebelumnya dan setelah itu dilakukan proses penghapusan data. Pemilihan ID dilakukan secara manual, sehingga admin dapat memilih ID mana yang tidak digunakan dan dapat digunakan lagi sesuai kebutuhan admin atau guru tersebut. ID yang telah dihapus juga dapat digunakan dengan sidik jari yang berbeda.



Gambar 4.3 Pengujian *Delete ID*

Dari data diatas dapat dilihat ID 90 dan ID 46 telah dilakukan proses hapus data. Dan data dengan ID itu tidak akan terbaca.

Tabel 4.1 Data Akses *Fingerprint*

No	Data	ID	Waktu	Tanggal	Status
1	12;01:39:11;25/06/18	12	1:39	25 - 06 - 2018	Berhasil

No	Data	ID	Waktu	Tanggal	Status
2	31;01:40:31;25/06/18	31	1:40	25 - 06 - 2018	Berhasil
3	21;01:40:29;25/06/18	21	1:40	25 - 06 - 2018	Berhasil
4	11;01:42:51;25/06/18	11	1:42	25 - 06 - 2018	Berhasil
5	9;01:42:09;25/06/18	9	1:42	25 - 06 - 2018	Berhasil
6	15;01:42:48;25/06/18	15	1:42	25 - 06 - 2018	Berhasil
7	32;01:43:38;25/06/18	32	1:43	25 - 06 - 2018	Berhasil
8	0;01:44:53;25/06/18	0	1:44	25 - 06 - 2018	Gagal

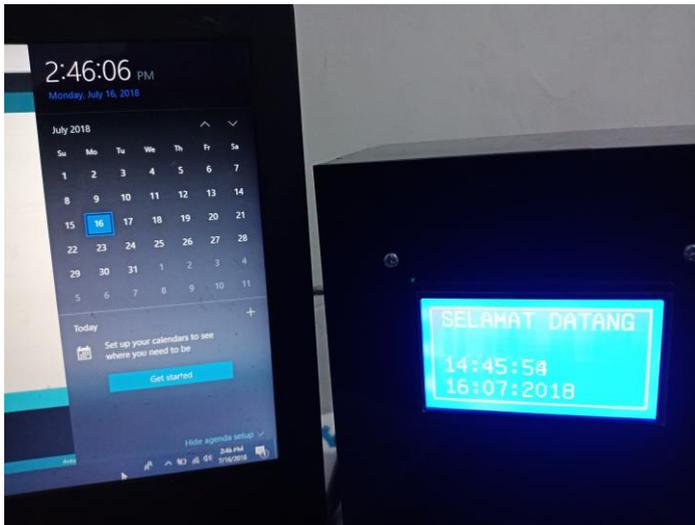
Pengujian error sidik jari dilakukan dengan melakukan *log in* pada ID yang sama secara berturut – turut. Hasilnya adalah seperti pada Tabel 4.1 dengan format sebagai berikut dengan contoh data pertama :

12 : ID  
01 : Hour  
39 : Minute  
11 : Second  
25 : Day  
06 : Month  
18 : Year

Berdasarkan pengujian tersebut terlihat bahwa *log in* ke – 8 terjadi kegagalan. Hal ini dikarenakan oleh posisi jari pada saat melakukan *log in* kurang lembab ataupun tidak sesuai dengan posisi saat didaftarkan. Biasanya posisi jari terlalu kebawah sehingga pembacaan sidik jari tidak sempurna. Ada pula hal yang mempengaruhi kesalahan dalam pembacaan, yaitu jari yang didaftarkan salah atau menggunakan jari yang lain. Apabila terjadi kegagalan diharapkan jari dibersihkan terlebih dahulu atau sensor *fingerprint* kotor sehingga mempengaruhi pemindaian.

## 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 mikrokontroler. RTC memerlukan *Supply +5V* yang didapatkan dari *Power Supply*, pin SDA ke SDA RTC dan pin SCL ke SCL Arduino Mega 2560. Format tanggal dan waktu yang dikirim oleh RTC akan di tampilkan pada *GLCD*.



Gambar 4.4 Pengujian RTC

Pada Gambar 4.4 dapat dilihat bahwa pengujian RTC DS1307 sesuai dengan keinginan, yaitu pengaturan waktu sesuai dengan waktu yang ada atau *real time*. Pada gambar diatas dapat dilihat penulisan angka telah sesuai. Hasil pengujian RTC pada gambar tersebut dapat dilihat bahwa format penulisan sebagai berikut :

00 : Jam  
53 : Menit  
04 : Detik  
25 : Tanggal  
06 : Bulan  
2018 : 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 *GLCD*

Pengujian *GLCD* atau *Graphic Liquid Crystal Display* dilakukan agar dapat mengetahui kelayakan *GLCD* tersebut untuk digunakan. Hal ini dapat dilihat dengan hasil yang ditampilkan oleh *GLCD*. Pada Gambar 4.5 dapat dilihat bahwa *GLCD* dapat menampilkan data sesuai dengan program. Pada program sebelumnya telah tertera kata “SELAMAT DATANG” dan data RTC.



Gambar 4.5 Pengujian *GLCD*

Dari hasil gambar diatas dapat dilihat tampilan awal pada *GLCD* terdapat kata Selamat datang dan dibawah tulisan terdapat waktu maupun tanggal. Ini adalah tampilan awal pada proses absensi siswa sekolah. Waktu yang tertera merupakan hasil dari data RTC yang telah deprogram sesuai keadaan.

### 4.4 Pengujian *Keypad 4x4*

Pengujian *Keypad 4x4* dilakukan untuk membandingkan apakah hasil dari penekanan *Keypad* sama dengan hasil yang keluar pada tampilan *GLCD*. Pada pengujian ini dilakukan dengan membandingkan data hasil penekanan dengan selang waktu penekanan beberapa detik kemudian dilepas dan hasilnya ditampilkan di *GLCD*, setelah dilakukan pengujian didapat hasil sebagai berikut :



Gambar 4.6 Pengujian *Keypad* Tombol 1

Dari Gambar 4.6 dapat disimpulkan bahwa lamanya waktu dalam penekanan *Keypad* tidak berpengaruh terhadap tampilan yang ditampilkan pada *GLCD*. Jika menekan “1” satu kali kemudian ditahan dalam waktu 3 detik yang muncul pada *GLCD* adalah “1” pada baris ke 1 kolom 1 saja, sedangkan baris dan kolom yang lain tetap kosong tidak ada karakter yang muncul pada *GLCD* tersebut.

Selanjutnya adalah pengujian *Keypad* dengan memasukkan ID atau absen siswa yang nantinya di tampilkan pada *GLCD*. Pengujian ini dilakukan agar dapat membuktikan bahwa keypad dapat memberikan *input* lebih dari satu angka. Pada Gambar 4.7 menunjukkan bahwa penulisan ID atau absen siswa yang ditampilkan pada *GLCD*. ID atau absen siswa yang di ujikan adalah “22”. Keypad akan menekan angka “22” dan hasil yang ditampilkan pada *GLCD* juga angka “22”.



Gambar 4.7 Pengujian *Keypad* Angka 22

#### 4.5 Pengujian Tampilan *Database*

Visual Basic .Net diuji dengan menampilkan hasil penyimpanan database pada EEPROM dengan format ID, Jam, Tanggal. Pengujian ini melalui PC yang telah memiliki aplikasi *database* Visual Basic .Net. Pengujian dilakukan dengan menggunakan 5 sidik jari yang berbeda, serta ID dan nama siswa yang telah didaftarkan sebelumnya. Data ID serta nama siswa dapat dilihat pada Tabel 4.3. Pengujian *database* ini digunakan agar kecocokan data berupa nama dan ID sesuai dan tidak ada kesalahan dalam pemindaian sidik jari siswa.

Tabel 4.2 Data ID siswa

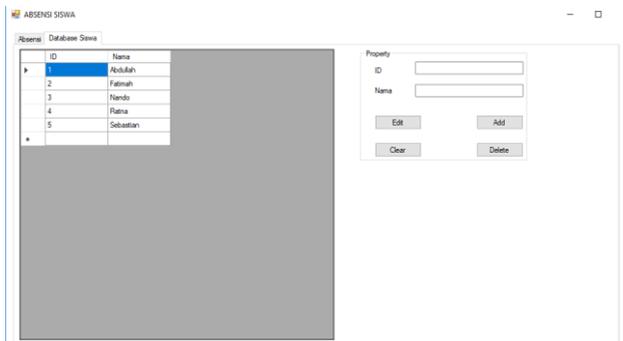
ID	Nama
1	Abdullah
2	Fatimah
3	Nando
4	Ratna
5	Sebastian

Data ID serta Nama akan tertera pada database absensi siswa dengan adanya keterangan waktu dan tanggal. Proses absensi akan terrekap sesuai dengan waktu absensi. Sehingga urutan *database* tidak tergantung oleh ID. Pengujian ini dilakukan untuk melihat kecocokan ID dan nama.



Gambar 4.8 Pengujian *Database* Visual Basic .Net

Pada Gambar 4.8 dapat dilihat hasil data yang ditampilkan oleh *database* Visual Basic .Net. Dimana ditampilkan tersebut terdapat status keberhasilan, waktu pengaksesan, serta kecocokan ID dan nama yang telah dibuat sebelumnya. ID dan sidik jari yang terdaftar akan terhubung dengan database siswa yang telah dibuat seperti Gambar 4.9. Sehingga *database* absensi dapat dilihat ID dan Nama yang sesuai.



Gambar 4.9 Tampilan *Database* ID Siswa

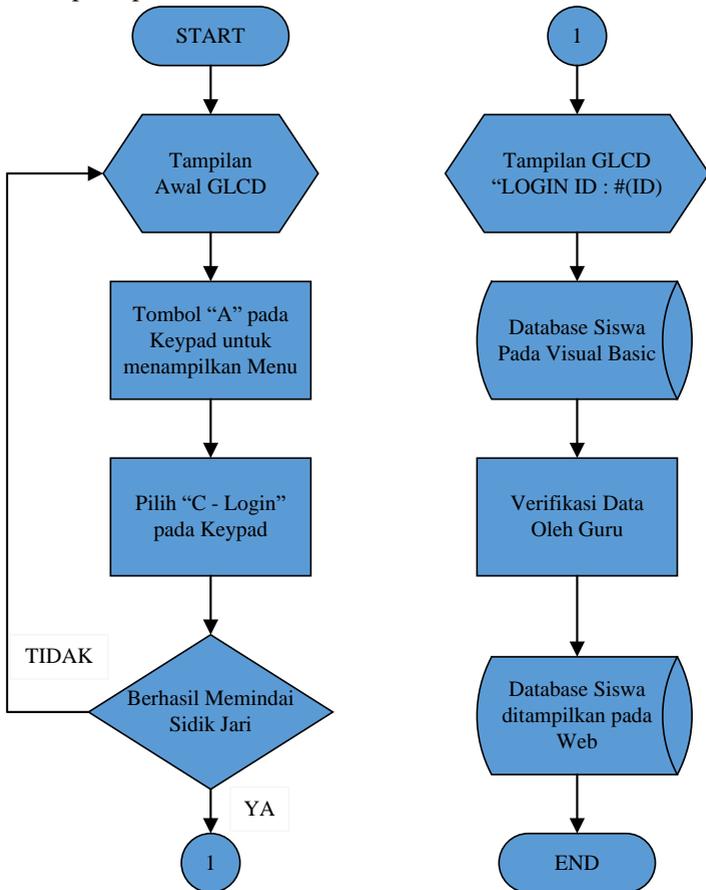
#### 4.6 Pengujian Sistem Keseluruhan

Pengujian keseluruhan pada sistem dilakukan setelah menggabungkan keseluruhan komponen. Dengan menguji kerja *fingerprint*, *keypad*, tampilan GLCD, Tampilan Visual Basic dan Tampilan pada Website. Sistem keseluruhan seperti pada Gambar 4.10.



Gambar 4.10 *Hardware* dan *Software* Sistem Keseluruhan

Pada pengujian sistem keseluruhan ini dapat dilihat juga alur dari sistem ini pada Gambar 4.11. dimana pada gambar tersebut terdapat *flowchart* dari siste keseluruhan mulai dari tampilan awal pada *GLCD* sampai tampilan pada Website.



Gambar 4.11 *Flowchart* Sistem Keseluruhan

Pada box, terdapat *GLCD* yang difungsikan sebagai penampil pada sistem absensi sekolah. Pada awal sistem menerima tegangan, *GLCD* akan menampilkan “SELAMAT DATANG” dengan mencantumkan jam dan pukul yang diletakkan dibawah tulisan seperti

pada Gambar 4.12. Setelah *keypad* ditekan huruf “A” maka sistem akan melanjutkan ketahap selanjutnya, maka akan menampilkan tampilan *GLCD* selanjutnya.



Gambar 4.12 Tampilan Awal Sistem Absensi



Gambar 4.13 Tampilan Menu Sistem Absensi

Pada Gambar 4.13 menunjukkan tampilan *GLCD* setelah ditekan huruf “A” melalui *keypad* yang digunakan untuk *input* data. Tampilan *GLCD* tersebut menunjukkan pilihan menu untuk melakukan eksekusi sistem dengan penjelasan sebagai berikut:

A – Daftar : Menu ini digunakan untuk melakukan proses *enrollment* atau pendaftaran sidik jari dengan memasukkan ID yang diinginkan dan mengikuti perintah meletakkan jari sesuai dengan yang diinginkan.

B – Hapus : Menu ini digunakan untuk melakukan proses penghapusan sidik jari yang telah didaftarkan dengan memasukkan ID yang ingin dihapus dan menekan “#” pada *keypad* sebagai fungsi enter. Jika berhasil terhapus, *GLCD* akan menampilkan tulisan “Berhasil Terhapus”.

C – Log In : Menu ini digunakan untuk melakukan proses *log in* dengan meletakkan jari pada scan area pada *fingerprint scanner* dengan delay 2 *second* untuk mencocokkan sidik jari. Jika sidik jari terdaftar dan berhasil *log in*, maka *GLCD* akan menampilkan tulisan “Berhasil LOGIN”. Jika sidik jari belum terdaftar dan tidak berhasil *log in*, maka *GLCD* akan menampilkan *page* awal untuk melakukan eksekusi kembali.

D – Hapus Data : Menu ini digunakan untuk menghapus data EEPROM. Jika berhasil, maka *GLCD* akan menampilkan tulisan “Menghapus EEPROM”.

Penggunaan pilihan menu dapat disesuaikan dengan keinginan atau prosedur yang telah dibuat. Langkah – langkah pada penggunaan meni Daftar dan *Log In* telah dijelaskan pada sebelumnya dan dapat juga dilihat pada *flowchart* yang telah dibuat. akan tetapi dalam proses ini dianjurkan memilih C - Log In agar dapat melihat proses keseluruhan yang datanya diperoleh saat melakukan *log in* atau saat siswa sekolah melakukan absensi.



Gambar 4.14 Pilih Tombol C pada Keypad

Setelah sistem memunculkan tampilan pilihan pada *GLCD* maka dipilihlah tombol “C” untuk melakukan *Log In* seperti pada Gambar 4.14. Selanjutnya *GLCD* menampilkan seperti pada Gambar 4.15 yaitu tertulis “Login: ” dan posisi lampu scan area pada *fingerprnt* menyala. Hal ini dikarenakan lampu tersebut berfungsi memperjelas proses pengambilan gambar sidik jari.



Gambar 4.15 Tampilan *Login* Sistem Absensi

Pada Gambar 4.16 menunjukkan kondisi *GLCD* yang menampilkan tulisan “Login ID: #1” yang berarti bahwa *Log In* berhasil dan ID yang terbaca yaitu 1. Setelah saat sidik jari berhasil melakukan *Log in*, maka tampilan pada *GLCD* akan kembali lagi ke tampilan awal seperti Gambar 4.11.



Gambar 4.16 Tampilan *Log In* Berhasil

Setelah *Log In* berhasil dan ID terbaca, sistem akan mengirim data melalui kabel serial yang terhubung antara Arduino dan PC. Data *Log In* tersebut akan diterima pada *database* yang telah dibuat pada aplikasi Visual Basic .Net seperti Gambar 4.17.



Gambar 4.17 Tampilan Visual Basic Sistem Keseluruhan

Tampilan database yang diterima pada aplikasi Visual Basic .Net akan dicek ulang oleh pihak sekolah dan akan dilakukan proses verifikasi siswa. Setelah proses verifikasi dilakukan maka data yang valid akan ditampilkan di Website dan dapat dilihat oleh setiap guru seperti Gambar 4.18.

**Laporan Absensi**  
 Periode : 10-07-2018  
 Filter : None

SMA Islam Terpadu Al-Uswah  
 -  
 Surabaya

No	Tanggal	Waktu	Id Siswa	Nama Siswa	Jenis Kel	Status	Keterangan
1	10-07-2018	12:02:21	1	ABDULLAH	Laki-Laki	Masuk	BERHASIL
2	10-07-2018	12:03:23	2	FATIMAH	Perempuan	Masuk	BERHASIL
3	10-07-2018	12:02:30	3	NANDO	Laki-Laki	Masuk	BERHASIL
4	10-07-2018	12:03:13	4	RATNA	Perempuan	Masuk	BERHASIL
5	10-07-2018	12:02:50	5	SEBASTIAN	Laki-Laki	Masuk	BERHASIL

Gambar 4.18 Tampilan Laporan Absensi Siswa pada Website

Pada tampilan *database* Visual basic dapat diedit secara manual agar mempermudah mengisi data keterangan sesuai dengan kebutuhan guru. Tampilan *database* ini juga dapat memberikan status masuk, izin dan alpha. Pengisian status dapat dilihat pada Gambar 4.19 dengan kode yang telah dibuat.

ABSENSI SISWA

Absensi Database Siswa

Sebastian  
13:20:9

### RECORD ABSENSI SISWA

No.	ID	Nama	Tanggal	Waktu	Keterangan	Status
1	5	Sebastian	18-7-10	13:20:9	BERHASIL	1
2	1	Abdullah	18-7-10	13:24:23	BERHASIL	1
3	3	Nando	18-7-10	13:24:41	BERHASIL	1
4	2	Fatmah	18-7-10	00:00:00	-	2
5	4	Ratna	18-7-10	13:25:18	DISPENSASI JAM 08.00-10	3
6						

Kode Status :  
 1 = Masuk 2 = Alpha 3 = Izin

EXPORT

Gambar 4.19 Tampilan Data Siswa dengan Berbeda Status

Kode status pada kolom status dapat dilihat pada penjelasan dibawah ini :

- 1 : Masuk
- 2 : Alpha
- 3 : Izin

Setelah status diisi oleh guru dengan kode yang telah dibuat, maka tampilan database pada Website akan berubah sesuai dengan arti kode tersebut. Hal tersebut dapat dilihat pada Gambar 4.20. Pada kolom Keterangan admin atau guru dapat mengisi dengan cara manual atau mengetik biasa. Pengisian kolom keterangan secara manual dapat mempermudah guru dalam mengubah status siswa secara mendadak. Tulisan pada tiap kolom memiliki warna yang berbeda dan memiliki arti. Seperti contohnya warna merah pada kolom yang menandakan bahwa siswa tersebut memiliki status Alpha atau tidak masuk tanpa keterangan. Apabila warna biru pada kolom menandakan bahwa siswa tersebut memiliki status Ijin dan terdapat penjelasan pada kolom keterangan siswa yang ijin. Apabila warna hitam menandakan bahwa siswa tersebut telah hadir atau masuk sekolah dengan terdapat tulisan berhasil pada kolom keterangan.

**Laporan Absensi**

Periode : 10-07-2018

Filter : None

SMA Islam Terpadu Al-Uswah

-

Surabaya

No	Tanggal	Waktu	Id Siswa	Nama Siswa	Jenis Kel	Status	Keterangan
1	10-07-2018	01:24:23	1	ABDULLAH	Laki-laki	Masuk	BERHASIL
2	10-07-2018	12:00:00	2	FATIMAH	Perempuan	Alpha	-
3	10-07-2018	01:24:41	3	NANDO	Laki-laki	Masuk	BERHASIL
4	10-07-2018	01:25:18	4	RATNA	Perempuan	Ijin	DISPENSASI JAM 08.00-10.00
5	10-07-2018	01:20:09	5	SEBASTIAN	Laki-laki	Masuk	BERHASIL

Gambar 4.20 Tampilan Laporan Siswa Berbeda Status

Pada tampilan Website terdapat pilihan tanggal akses dari tanggal berapa hingga tanggal berapa. Setelah itu dapat di-submit dan hasilnya akan tertera di Website tersebut. Tampilan laporan database dengan tanggal tertentu dapat dilihat pada Gambar 4.21 yang berisi dari tanggal 08 Juli 2018 sampai 10 Juli 2018.

Pada tampilan Website juga terdapat pilihan *search* nama siswa yang berada dibawah pilihan tanggal. Nama yang dituliskan harus sesuai dengan nama yang telah didaftarkan, dapat dilihat pada Gambar 4.22. Dan hasil dari pilihan *search* akan ditampilkan ke laporan *database* Website seperti Gambar 4.23.

**Laporan Absensi**  
 Periode : 08-07-2018 s/d 10-07-2018  
 Filter : None

SMA Islam Terpadu Al-Uswah  
 Surabaya

No	Tanggal	Waktu	Id Siswa	Nama Siswa	Jenis Kel	Status	Keterangan
1	08-07-2018	01:21:35	1	ABDULLAH	Laki-laki	Masuk	BERHASIL
2	08-07-2018	01:19:49	2	FATIMAH	Perempuan	Masuk	BERHASIL
3	08-07-2018	01:23:38	3	NANDO	Laki-laki	Masuk	BERHASIL
4	08-07-2018	01:20:26	4	RATNA	Perempuan	Masuk	BERHASIL
5	08-07-2018	01:20:09	5	SEBASTIAN	Laki-laki	Masuk	BERHASIL
6	09-07-2018	01:22:45	1	ABDULLAH	Laki-laki	Masuk	BERHASIL
7	09-07-2018	01:22:25	2	FATIMAH	Perempuan	Masuk	BERHASIL
8	09-07-2018	01:22:35	3	NANDO	Laki-laki	Masuk	BERHASIL
9	09-07-2018	01:22:12	4	RATNA	Perempuan	Masuk	BERHASIL
10	09-07-2018	01:20:55	5	SEBASTIAN	Laki-laki	Masuk	BERHASIL
11	10-07-2018	01:24:23	1	ABDULLAH	Laki-laki	Masuk	BERHASIL
12	10-07-2018	01:25:03	2	FATIMAH	Perempuan	Masuk	BERHASIL
13	10-07-2018	01:24:41	3	NANDO	Laki-laki	Masuk	BERHASIL
14	10-07-2018	01:25:18	4	RATNA	Perempuan	Masuk	BERHASIL

Gambar 4.21 Tampilan Laporan Siswa dengan Tanggal Tertentu

Tgl Mulai

Tgl Sampai

Siswa 

- ABDULLAH
- FATIMAH
- NANDO
- RATNA
- SEBASTIAN

Gambar 4.22 Pilihan *Search* Nama Siswa

### Laporan Absensi

Periode : 08-07-2018 s/d 10-07-2018

Filter : By Nama Siswa

SMA Islam Terpadu Al-Uswah

-

Surabaya

No	Tanggal	Waktu	Id Siswa	Nama Siswa	Jenis Kel	Status	Keterangan
1	08-07-2018	01:21:35	1	ABDULLAH	Laki-laki	Masuk	BERHASIL
2	09-07-2018	01:22:45	1	ABDULLAH	Laki-laki	Masuk	BERHASIL
3	10-07-2018	01:24:23	1	ABDULLAH	Laki-laki	Masuk	BERHASIL

Gambar 4.23 Tampilan Laporan Siswa berdasarkan Nama

## **BAB V PENUTUP**

### **5.1 Kesimpulan**

Setelah dilakukan pengujian dari masing masing komponen dan sistem secara keseluruhan oleh peneliti, maka dapat disimpulkan sebagai berikut :

1. Sistem Monitoring dan Pelaporan Kehadiran Siswa Berbasis Website ini dapat bekerja sesuai dengan fungsinya sebagai rekapitulasi data siswa masuk.
2. Keakuratan pembacaan sensor 96,2% berdasarkan pengujian kecocokan sidik jari dengan ID yang sama. Data dari sensor *fingerprint* dan ID yang masuk sesuai dengan data siswa yang telah direkap.
3. Tampilan pada Visual Basic adalah tampilan *database* yang waktu penampilannya sesuai dengan saat melakukan pemindaian sidik jari.
4. Tampilan pada Website adalah tampilan *database* yang waktu penampilannya setelah dilakukan proses verifikasi oleh guru.

### **5.2 Saran**

Sistem monitoring dan pelaporan kehadiran siswa disekolah ini telah berjalan sesuai dengan rancangannya, tetapi untuk menyempurnakan hasilnya perlu dipertimbangkan hal – hal sebagai berikut :

1. Perlu ditingkatkan proses pengiriman data menggunakan *Wifi* dan *software* penampil data agar lebih mudah diakses.
2. *Database* dapat dikembangkan lagi menjadi berbagai fitur agar mempermudah guru dengan menambahkan kolom nilai dan pemberitahuan apabila ada ujian.
3. Perlu membuat sistem login lebih mudah tanpa melalui pilihan menu. Alat dapat memindai sidik jari pada tampilan awal tanpa memilih menu terlebih dahulu.
4. Perlu dikembangkan sistem notifikasi melalui surat elektronik atau sejenisnya. Agar setiap orang tua tahu apabila anaknya telah berada disekolah.

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

## DAFTAR PUSTAKA

- [1] Dwirani, E. I., "**Implementasi Sidik Jari untuk Sistem Absensi pada SDN II Cineam Tasikmalaya**", Skripsi, Universitas Komputer Indonesia, Bandung, 2004.
- [2] Kadir, Abdul. **Buku Pintar Pemrograman Arduino**. MediaKom. Yogyakarta. 2015.
- [3] ....., **Datasheet 128x64Dots Serial/Parallel LCD**. Digole, 2007.
- [4] ....., "**Matrix Keypad 4x4 untuk Mikrokontroler**". <http://elektronika-dasar.web.id/matrix-keypad-4x4-untuk-mikrokontroler/>
- [5] ....., **Datasheet DS1307 64 x 8, Serial, I 2 C Real-Time Clock**. Maxim Integrated Products, 2015.
- [6] Atmel. 2007. **Datasheet Two-wire Serial EEPROM**. San Jose : Atmel Corporation.
- [7] ....., "**Apa itu Visual Basic? Visual Basic adalah**". <http://www.carawebs.info/2013/04/apa-itu-visual-basic-visual-basic-adalah.html>
- [8] ....., "**Pengertian, Komponen dan Fungsi XAMPP**". <http://www.dosenpendidikan.com/pengertian-komponen-dan-fungsi-xampp-lengkap-dengan-penjelasan/>
- [9] Budiharto, Widodo. 2007. **12 Proyek Sistem Akuisi Data**. Jakarta : PT Elex Media Komputindo.
- [10] Aswindha Nasrullah, "Sistem Lock Door Panel Listrik Menggunakan Fingerprint Scanner Berbasis Android", **Tugas Akhir**, Program D3 Teknik Elektro. 2016.

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

## LAMPIRAN A PROGRAM

### A.1 Program Arduino

```
#include <SoftwareSerial.h>
#include <Adafruit_Fingerprint.h>
#include <Wire.h>
#include "U8glib.h"
#include "RTClib.h"

#define AT24C02 0x50 // Address with three address pins grounded.
U8GLIB_ST7920_128X64 u8g(37, 35, 36, U8G_PIN_NONE);
RTC_DS1307 rtc;

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial3);

int jam, menit, detik, tanggal, bulan, tahun;
byte logdata[9];
char str[100], X, v;
int str_index, readaddr=0;// Mengirim maksimal 100 karakter
int i;

void setup()
{
  Wire.begin();
  finger.begin(57600);
  Serial.begin(9600);
  Serial2.begin(9600);

  rtc.begin();
  rtc.adjust(DateTime(F(__DATE__),F(__TIME__)));

  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(7, OUTPUT);
```

```

pinMode(52, OUTPUT);
pinMode(53, OUTPUT);
pinMode(33, OUTPUT);

analogWrite(2, 220);
digitalWrite(3, HIGH);
digitalWrite(4, HIGH);
digitalWrite(5, HIGH);
digitalWrite(7, HIGH);
digitalWrite(52, HIGH);
digitalWrite(53, HIGH);
digitalWrite(33, LOW);

delay(500);

keypad_init();

logdata[7] = 10; //KARAKTER NEWLINE
logdata[8] = 0;

// assign default color value
if ( u8g.getMode() == U8G_MODE_R3G3B2 ) {
    u8g.setColorIndex(255);    // white
}
else if ( u8g.getMode() == U8G_MODE_GRAY2BIT ) {
    u8g.setColorIndex(3);     // max intensity
}
else if ( u8g.getMode() == U8G_MODE_BW ) {
    u8g.setColorIndex(1);     // pixel on
}
else if ( u8g.getMode() == U8G_MODE_HICOLOR ) {
    u8g.setHiColorByRGB(255,255,255);
}
u8g.setFont(u8g_font_unifont);
}

void loop()
{
    awal:

```

```

int id, caddr, result;
char c, buff[20], *temp;
c = 0;

showMenu(0);

str_index = 0; //reset index penerimaan karakter serial
X = keypad();
if (X=='A')//jika tidak ada karakter masuk lewat serial
{ goto menu;}
if (X=='B'){ goto awal;}//program interface
goto awal;

menu:
{
X=Serial.read();
if (X == 'R'){ goto awal;}
DateTime now = rtc.now();
c = 0;
c = keypad();
showMenu(1);
switch(c) //c = karakter yang diinput melalui keypad
{
case 'A' : enrollID(atoi(waitInput("ID Baru :")));
break;
case 'B' : deleteID(atoi(waitInput("Hapus ID :")));
break;
case 'C' : showMenu(2);
while((id=getID()) == -1) if(keypad() == 'D') break;
//menunggu sidik jari atau cancel menekan keypad D

//memasukkan parameter rtc ke dalam format penyimpanan
logdata[2] = now.hour();
logdata[3] = now.minute();
logdata[7] = now.second();
logdata[4] = now.day();
logdata[5] = now.month();
logdata[6] = now.year()%2000;

```

```

if(id >= 0) //jika sidik jari terdaftar
{
    logdata[0] = 0;
    logdata[1] = id;
    itoa(id, buff, 10);

    //tampilan glcd
    u8g.firstPage();
    do {
        u8g.drawRFrame(0, 16, 128, 35, 0);
        u8g.drawStr(3, 29, "LOGIN ID :");
        u8g.drawStr(3, 44, "#");
        u8g.drawStr(11, 44, buff);
    } while( u8g.nextPage() );

    //nyalakan reLay dan buzzer
    digitalWrite(3, LOW);
    delay(1000);
    digitalWrite(3, HIGH);
    digitalWrite(4, LOW);
    digitalWrite(5, LOW); //Indikator Hijau
    delay(3000);
    digitalWrite(4, HIGH);
    digitalWrite(5, HIGH); //Indikator hijau
}
else //Jika sidik jari tak terdaftar atau error
{
    u8g.firstPage();
    do {
        u8g.drawRFrame(0,16,128, 35, 0);
        u8g.drawStr(3, 29, "LOGIN GAGAL !");
    } while( u8g.nextPage() );

    logdata[0] = 0;
    logdata[1] = 0;
}
//String jamx=;
Serial.print(String(logdata[1]));

```

```

Serial.print (');

for(i=0; i<=1; i++)
{EEPROMWrite(i, logdata[i]);

//Serial.print(logdata[i]);
} //menulis data ke eeprom*/

Serial.print(String(logdata[2])+"."+String(logdata[3])+"."+S
tring(logdata[7]));
Serial.print (');
Serial.print(String(logdata[6])+"-"+String(logdata[5])+"-
"+String(logdata[4]));
Serial.println (');
goto awal;
break;
case 'D': showMenu(3);
for(int i=0; i<2047; i++) EEPROMWrite(i, 0); //menulis
karakter 0 (null) di semua alamat eeprom = menghapus data
break;
}
}
goto menu;

eeprom:
readaddr = 0; v = -1;
for(int i=readaddr; i<=readaddr+7; i++) //membaca 8 data eeprom
{
//membaca eeprom dengan alamat i
Serial.println(EEPROMRead(i));
}
readaddr += 8; //untuk membaca 8 alamat berikutnya
goto awal;
}

char * waitInput(char * header) //proses input angka dari keypad
{
boolean del = false;
int index = 0;

```

```

char buff[10];
memset(buff, 0, 10); //reset char array buff

drawInput(header, buff);
while(1)
{
    char c = keypad(); //input karakter keypad

    if(c >= '0' && c <= '9') //untuk karakter 0-9
    {
        if(del == true) { index++; del = false; } //algoritma untuk
menghapus karakter dalam array
        buff[index++] = c; //menyimpan karakter c ke dalam char array
buff
        if(index >= 8) index = 8; //input tidak bisa melebihi 8 karakter

        drawInput(header, buff); //menampilkan hasil input ke glcd
    }
    else if(c == '*') //karakter * adalah untuk menghapus input
    {
        if(del == false) { index--; del = true; } //algoritma untuk
menghapus karakter dalam array
        if(index <= 0) { index = 0; memset(buff, 0, 10); } //jika tidak ada
karakter lagi untuk dihapus
        buff[index--] = 0; //karakter terakhir diganti dengan karakter 0
(null), dengan kata lain sama dengan dihapus

        drawInput(header, buff); //tampilkan ke glcd
    }
    else if(c == '#') break; //enter, keluar dari perulangan
}
return buff; //nilai balik buff
}

void drawInput(char * header, char str[]) //menampilkan data ke glcd
{
    u8g.firstPage();
    do {
        u8g.drawStr(0, 11, header);

```

```

    u8g.drawStr(0, 26, str);
  } while( u8g.nextPage() );
}

```

```

void showMenu(int menu) //tampilkan data ke gLCD
{
  u8g.firstPage();
  do {
    drawMenu(menu);
    DateTime now = rtc.now();
  } while( u8g.nextPage() );
}

```

```

void drawMenu(int stat) //tampilkan menu ke gLCD
{
  char c, buff[20], *temp;
  DateTime now = rtc.now();
  switch(stat)
  {
    case 0:
      u8g.drawRFrame(0,0,128, 64, 0);
      u8g.drawStr(7, 13, "SELAMAT DATANG");
      sprintf(buff, "%02d:%02d:%02d ", now.hour(), now.minute(),
now.second());
      u8g.drawStr(7, 44, buff);
      sprintf(buff, "%02d:%02d:%02d ", now.day(), now.month(),
now.year());
      u8g.drawStr(7, 59, buff);
      break;
    case 1:
      u8g.drawRFrame(0,0,128, 64, 0);
      u8g.drawStr(3, 13, "A - Daftar");
      u8g.drawStr(3, 29, "B - Hapus");
      u8g.drawStr(3, 44, "C - Login");
      u8g.drawStr(3, 59, "D - Hapus Data");
      break;
    case 2:
      u8g.drawRFrame(0,16,128, 35, 0);
      u8g.drawStr(3, 29, "Login :");

```

```
        break;
    case 3:
        u8g.drawRFrame(0,16,128, 35, 0);
        u8g.drawStr(3, 29, "Menghapus EEPROM...");
        break;
    }
}
```

## A.2 Program Visual Basic .Net

- Program Koneksi dengan Arduino Mega2560

```
Public Sub konekserial(ByVal Com As String)
```

```
Try
```

```
With Me.serial_main
```

```
.PortName = Com
```

```
.BaudRate = 9660
```

```
.ReadBufferSize = 500
```

```
.Parity = IO.Ports.Parity.None
```

```
.DataBits = 8
```

```
.StopBits = 1
```

```
.Handshake = IO.Ports.Handshake.None
```

```
End With
```

```
serial_main.Open()
```

```
Timer1.Enabled = True
```

```
System.Threading.Thread.Sleep("100")
```

```
pic_connection.Visible = False
```

```
pic_connection_on.Visible = True
```

```
txt_connection.Text = "CONNECTED"
```

```
txt_connection.ForeColor = Color.Green
```

```
Me.Show()
```

```
Connection.Hide()
```

```
Catch ex As Exception
```

```
MessageBox.Show("Serial Not Ready")
```

```
pic_connection.Visible = True
```

```
pic_connection_on.Visible = False
```

```
Connection.Show()
```

```
txt_connection.Text = "DISCONNECTED"
```

```
txt_connection.ForeColor = Color.Red
```

```
Timer1.Enabled = False
```

```
End Try
```

```
End Sub
```

```
Private Sub btn_connect_Click_1(ByVal sender As Object, ByVal e  
As EventArgs) Handles btn_connect.Click
```

```
cektxt(alamat_com)
```

```
Dim com As String() = readtxt(alamat_com, 1)
konekserial(com(0))
```

```
End Sub
```

- Program Koneksi Database MS. Acces

```
Sub koneksi_database_access()
    dirDB = mydocumentfolderDatabase & file
    provider = "Provider=Microsoft.ACE.OLEDB.12.0;"
    dbsource = "Data Source=" & dirDB
    con = New OleDbConnection(provider & dbsource)
    If con.State = ConnectionState.Closed Then
        con.Open()
    End If
```

```
    If con.State = ConnectionState.Open Then
        monitoring.lbl_koneksi_database.Text = "OK"
        update_database()
    Else
        monitoring.lbl_koneksi_database.Text = "FAIL"
    End If
```

```
End Sub
```

```
Private Sub PictureBox2_Click_1(ByVal sender As Object, ByVal e
As EventArgs) Handles PictureBox2.Click
```

```
    koneksi_database_access()
    btExport.Enabled = True
```

```
End Sub
```

- Program Export

```
Public Sub Koneksi2()
    CONN = New OleDbConnection("dsn=conn_absen")
    CONN.Open()
    Str = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" &
Application.StartupPath & "\database\master.accdb"
    Conn2 = New OleDbConnection(Str)
    Conn2.Open()
End Sub
```

```

Private Sub btExport_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btExport.Click
    With DataGridView1
        Call Koneksi2()
        For i As Integer = 0 To .Rows.Count - 2
            CMD = New OdbcCommand("Insert Into db_absen
(WAKTU,TGL,ID_SISWA,STATUS,KETERANGAN) values" & _
            "(" & DataGridView1.Rows(i).Cells(4).Value & _
            "," & DataGridView1.Rows(i).Cells(3).Value & _
            "," & DataGridView1.Rows(i).Cells(1).Value & _
            "," & DataGridView1.Rows(i).Cells(6).Value & _
            "," & DataGridView1.Rows(i).Cells(5).Value & _
            ")", CONN)
            DR = CMD.ExecuteReader
            DR.Read()
        Next
        MsgBox("Sukses!")
        btExport.Enabled = False
    End With
End Sub

```

### A.3 Program XAMPP

```
CREATE TABLE `db_absen` (  
  `ID` int(9) NOT NULL,  
  `TGL` date NOT NULL,  
  `WAKTU` time NOT NULL,  
  `ID_SISWA` int(5) NOT NULL,  
  `STATUS` int(2) NOT NULL COMMENT '1:Masuk  
2:Alpa 3:Izin',  
  `KETERANGAN` text NOT NULL,  
  `ID_USER` int(2) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
CREATE TABLE `db_siswa` (  
  `ID_SISWA` int(5) NOT NULL,  
  `NAMA` varchar(50) NOT NULL,  
  `JENIS_KELAMIN` char(2) NOT NULL,  
  `TEMPAT_LAHIR` varchar(50) NOT NULL,  
  `TGL_LAHIR` date NOT NULL,  
  `ALAMAT` text NOT NULL,  
  `ID_USER` int(2) NOT NULL,  
  `TIMESTAMP` timestamp NOT NULL DEFAULT  
CURRENT_TIMESTAMP  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

# LAMPIRAN B DATASHEET

## B.1 Datasheet Real Time Clock

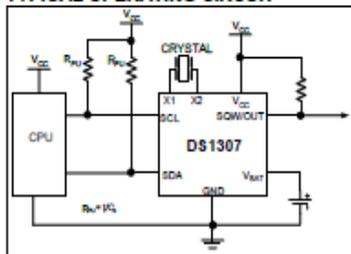


### DS1307 64 x 8, Serial, I<sup>2</sup>C Real-Time Clock

#### 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 I<sup>2</sup>C, 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.

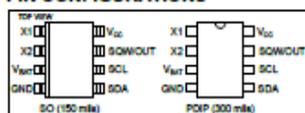
#### TYPICAL OPERATING CIRCUIT



#### BENEFITS AND FEATURES

- Completely Manages All Timekeeping Functions
  - Real-Time Clock 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
  - Programmable Square-Wave Output Signal
- Simple Serial Port Interfaces to Most Microcontrollers
  - I<sup>2</sup>C Serial Interface
- Low Power Operation Extends Battery Backup Run Time
  - Consumes Less than 500nA in Battery-Backup Mode with Oscillator Running
  - Automatic Power-Fail Detect and Switch Circuitry
- 8-Pin DIP and 8-Pin SO Minimizes Required Space
- Optional Industrial Temperature Range: -40°C to +85°C Supports Operation in a Wide Range of Applications
- Underwriters Laboratories® (UL) Recognized

#### 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/PbHS-compliant package.

\*"N" anywhere on the top mark indicates a lead-free package. An "N" anywhere on the top mark indicates an industrial temperature range device. Underwriters Laboratories, Inc. is a registered certification mark of Underwriters Laboratories, Inc.

**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 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<sub>A</sub> = 0°C to +70°C, T<sub>A</sub> = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V <sub>CC</sub>		4.5	5.0	5.5	V
Logic 1 Input	V <sub>HI</sub>		2.2		V <sub>CC</sub> + 0.3	V
Logic 0 Input	V <sub>LI</sub>		-0.3		+0.8	V
V <sub>BAT</sub> Battery Voltage	V <sub>BAT</sub>		2.0	3	3.5	V

**DC ELECTRICAL CHARACTERISTICS**

(V<sub>CC</sub> = 4.5V to 5.5V; T<sub>A</sub> = 0°C to +70°C, T<sub>A</sub> = -40°C to +85°C.) (Notes 1, 2)

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

**DC ELECTRICAL CHARACTERISTICS**

(V<sub>CC</sub> = 0V, V<sub>BAT</sub> = 3.0V; T<sub>A</sub> = 0°C to +70°C, T<sub>A</sub> = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
V <sub>BAT</sub> Current (OSC ON); SQW/OUT OFF	I <sub>BAT1</sub>			300	500	nA
V <sub>BAT</sub> Current (OSC ON); SQW/OUT ON (32kHz)	I <sub>BAT2</sub>			480	800	nA
V <sub>BAT</sub> Data-Retention Current (Oscillator Off)	I <sub>BATDR</sub>			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<sub>CC</sub> = 4.5V to 5.5V; T<sub>A</sub> = 0°C to +70°C, T<sub>A</sub> = -40°C to +85°C.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCL Clock Frequency	f <sub>SCL</sub>		0		100	kHz
Bus Free Time Between a STOP and START Condition	t <sub>SUF</sub>		4.7			μs
Hold Time (Repeated) START Condition	t <sub>HD,STA</sub>	(Note 4)	4.0			μs
LOW Period of SCL Clock	t <sub>LOW</sub>		4.7			μs
HIGH Period of SCL Clock	t <sub>HIGH</sub>		4.0			μs
Setup Time for a Repeated START Condition	t <sub>SU,STA</sub>		4.7			μs
Data Hold Time	t <sub>HD,DAT</sub>		0			μs
Data Setup Time	t <sub>SU,DAT</sub>	(Notes 5, 6)	250			ns
Rise Time of Both SDA and SCL Signals	t <sub>r</sub>				1000	ns
Fall Time of Both SDA and SCL Signals	t <sub>f</sub>				300	ns
Setup Time for STOP Condition	t <sub>SU,STO</sub>		4.7			μs

**CAPACITANCE**(T<sub>A</sub> = +25°C)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Pin Capacitance (SDA, SCL)	C <sub>IO</sub>				10	pF
Capacitance Load for Each Bus Line	C <sub>0</sub>	(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<sub>CC3</sub> specified with V<sub>CC</sub> = 5.0V and SDA, SCL = 5.0V.**Note 4:** After this period, the first clock pulse is generated.**Note 6:** A device must internally provide a hold time of at least 300ns for the SDA signal (referred to the V<sub>I(HMIN)</sub> of the SCL signal) to bridge the undefined region of the falling edge of SCL.**Note 8:** The maximum t<sub>HD,STO</sub> only has to be met if the device does not stretch the LOW period (t<sub>LOW</sub>) of the SCL signal.**Note 7:** C<sub>0</sub>—total capacitance of one bus line in pF.

## TIMING DIAGRAM

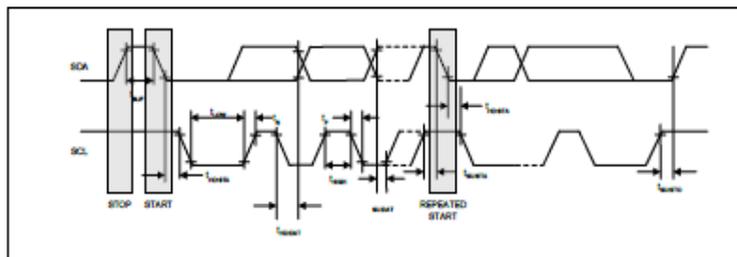
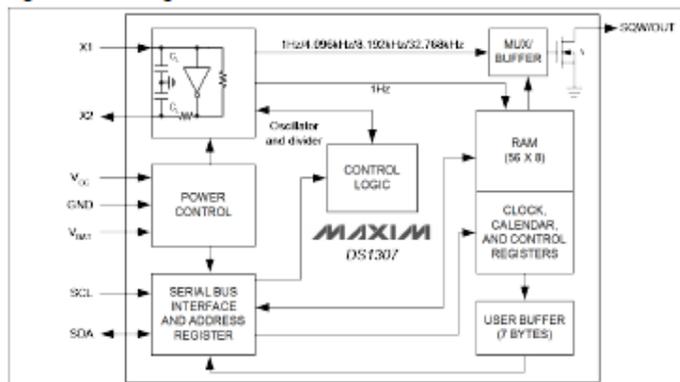
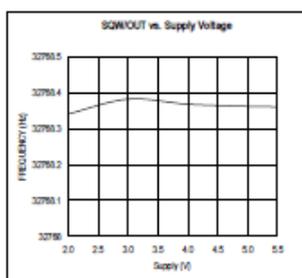
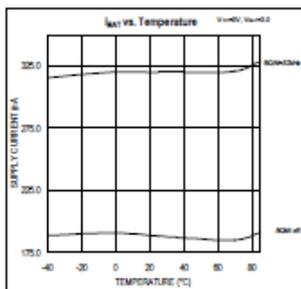
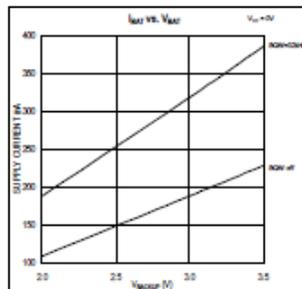
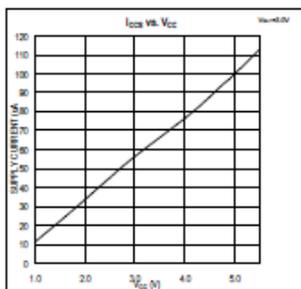


Figure 1. Block Diagram



**TYPICAL OPERATING CHARACTERISTICS**(V<sub>CC</sub> = 5.0V, T<sub>A</sub> = +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	<b>Note:</b> For more information on crystal selection and crystal layout considerations, refer to Application Note 58: <i>Crystal Considerations with Dallas Real-Time Clocks</i> .
3	V <sub>BAT</sub>	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 <sub>BAT</sub> pin may prevent proper operation. If a backup supply is not required, V <sub>BAT</sub> must be grounded. The nominal power-fail trip point (V <sub>PF</sub> ) voltage at which access to the RTC and user RAM is denied is set by the internal circuitry as 1.25 x V <sub>BAT</sub> 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: <a href="http://www.maxim-ic.com/ga/info/ul/">www.maxim-ic.com/ga/info/ul/</a> .
4	GND	Ground
5	SDA	Serial Data Input/Output. SDA is the data input/output for the I <sup>2</sup> C 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 <sub>CC</sub> .
6	SCL	Serial Clock Input. SCL is the clock input for the I <sup>2</sup> C 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 <sub>CC</sub> .
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 <sub>CC</sub> or V <sub>BAT</sub> applied. The pullup voltage can be up to 5.5V regardless of the voltage on V <sub>CC</sub> . If not used, this pin can be left floating.
8	V <sub>CC</sub>	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 <sub>CC</sub> is below V <sub>PF</sub> , 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 I<sup>2</sup>C 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<sub>CC</sub> falls below 1.25 x V<sub>BAT</sub>, 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<sub>CC</sub> falls below V<sub>BAT</sub>, the device switches into a low-current battery-backup mode. Upon power-up, the device switches from battery to V<sub>CC</sub> when V<sub>CC</sub> is greater than V<sub>BAT</sub> + 0.2V and recognizes inputs when V<sub>CC</sub> is greater than 1.25 x V<sub>BAT</sub>. The block diagram in Figure 1 shows the main elements of the serial RTC.

### OSCILLATOR CIRCUIT

The DS1307 uses an external 32.768kHz crystal. The oscillator circuit does not require any external resistors or capacitors to operate. Table 1 specifies several crystal parameters for the external crystal. Figure 1 shows a functional schematic of the oscillator circuit. If using a crystal with the specified characteristics, the startup time is usually less than one second.

### CLOCK ACCURACY

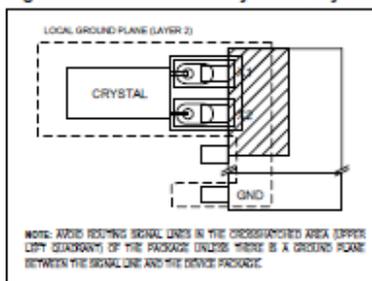
The accuracy of the clock is dependent upon the accuracy of the crystal and the accuracy of the match between the capacitive load of the oscillator circuit and the capacitive load for which the crystal was trimmed. Additional error will be added by crystal frequency drift caused by temperature shifts. External circuit noise coupled into the oscillator circuit may result in the clock running fast. Refer to Application Note 58: *Crystal Considerations with Dallas Real-Time Clocks* for detailed information.

Table 1. Crystal Specifications\*

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS
Nominal Frequency	$f_o$		32.768		kHz
Series Resistance	ESR			45	k $\Omega$
Load Capacitance	$C_L$		12.5		pF

\*The crystal, traces, and crystal input pins should be isolated from RF generating signals. Refer to Application Note 58: *Crystal Considerations for Dallas Real-Time Clocks* for additional specifications.

Figure 2. Recommended Layout for Crystal



### RTC AND RAM ADDRESS MAP

Table 2 shows the address map for the DS1307 RTC and RAM registers. The RTC registers are located in address locations 00h to 07h. The RAM registers are located in address locations 08h to 3Fh. During a multibyte access, when the address pointer reaches 3Fh, the end of RAM space, it wraps around to location 00h, the beginning of the clock space.

## CLOCK AND CALENDAR

The time and calendar information is obtained by reading the appropriate register bytes. Table 2 shows the RTC registers. The time and calendar are set or initialized by writing the appropriate register bytes. The contents of the time and calendar registers are in the BCD format. The day-of-week register increments at midnight. Values that correspond to the day of week are user-defined but must be sequential (i.e., if 1 equals Sunday, then 2 equals Monday, and so on.) Illogical time and date entries result in undefined operation. Bit 7 of Register 0 is the clock halt (CH) bit. When this bit is set to 1, the oscillator is disabled. When cleared to 0, the oscillator is enabled. On first application of power to the device the time and date registers are typically reset to 01/01/00 01 00:00:00 (MM/DD/YY DOW HH:MM:SS). The CH bit in the seconds register will be set to a 1. The clock can be halted whenever the timekeeping functions are not required, which minimizes current ( $I_{DA(TOT)}$ ).

The DS1307 can be run in either 12-hour or 24-hour mode. Bit 5 of the hours register is defined as the 12-hour or 24-hour mode-select bit. When high, the 12-hour mode is selected. In the 12-hour mode, bit 5 is the AM/PM bit with logic high being PM. In the 24-hour mode, bit 5 is the second 10-hour bit (20 to 23 hours). The hours value must be re-entered whenever the 12/24-hour mode bit is changed.

When reading or writing the time and date registers, secondary (user) buffers are used to prevent errors when the internal registers update. When reading the time and date registers, the user buffers are synchronized to the internal registers on any I<sup>2</sup>C START. The time information is read from these secondary registers while the clock continues to run. This eliminates the need to re-read the registers in case the internal registers update during a read. The divider chain is reset whenever the seconds register is written. Write transfers occur on the I<sup>2</sup>C acknowledge from the DS1307. Once the divider chain is reset, to avoid rollover issues, the remaining time and date registers must be written within one second.

Table 2. Timekeeper Registers

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00h	CH	10 Seconds			Seconds			Seconds		00–59
01h	0	10 Minutes			Minutes			Minutes		00–59
02h	0	12	10 Hour	10 Hour	Hours			Hours	1–12 +AM/PM 00–23	
		24	PM/ AM							
03h	0	0	0	0	DAY			Day	01–07	
04h	0	0	10 Date		Date			Date	01–31	
05h	0	0	0	10 Month	Month			Month	01–12	
06h	10 Year				Year			Year	00–99	
07h	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08h–3Fh									RAM 56 x 8	00h–FFh

0 = Always reads back as 0.

**CONTROL REGISTER**

The DS1307 control register is used to control the operation of the SQW/OUT pin.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OUT	0	0	SQWE	0	0	RS1	RS0

**Bit 7: Output Control (OUT).** This bit controls the output level of the SQW/OUT pin when the square-wave output is disabled. If SQWE = 0, the logic level on the SQW/OUT pin is 1 if OUT = 1 and is 0 if OUT = 0. On initial application of power to the device, this bit is typically set to a 0.

**Bit 4: Square-Wave Enable (SQWE).** This bit, when set to logic 1, enables the oscillator output. The frequency of the square-wave output depends upon the value of the RS0 and RS1 bits. With the square-wave output set to 1Hz, the clock registers update on the falling edge of the square wave. On initial application of power to the device, this bit is typically set to a 0.

**Bits 1 and 0: Rate Select (RS[1:0]).** These bits control the frequency of the square-wave output when the square-wave output has been enabled. The following table lists the square-wave frequencies that can be selected with the RS bits. On initial application of power to the device, these bits are typically set to a 1.

RS1	RS0	SQW/OUT OUTPUT	SQWE	OUT
0	0	1Hz	1	X
0	1	4.096KHz	1	X
1	0	8.192KHz	1	X
1	1	32.768KHz	1	X
X	X	0	0	0
X	X	1	0	1

## I<sup>2</sup>C DATA BUS

The DS1307 supports the I<sup>2</sup>C protocol. A device that sends data onto the bus is defined as a transmitter and a device receiving data as a receiver. The device that controls the message is called a master. The devices that are controlled by the master are referred to as slaves. The bus must be controlled by a master device that generates the serial clock (SCL), controls the bus access, and generates the START and STOP conditions. The DS1307 operates as a slave on the I<sup>2</sup>C bus.

Figures 3, 4, and 5 detail how data is transferred on the I<sup>2</sup>C bus.

- Data transfer can be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is HIGH. Changes in the data line while the clock line is HIGH will be interpreted as control signals.

Accordingly, the following bus conditions have been defined:

**Bus not busy:** Both data and clock lines remain HIGH.

**START data transfer:** A change in the state of the data line, from HIGH to LOW, while the clock is HIGH, defines a START condition.

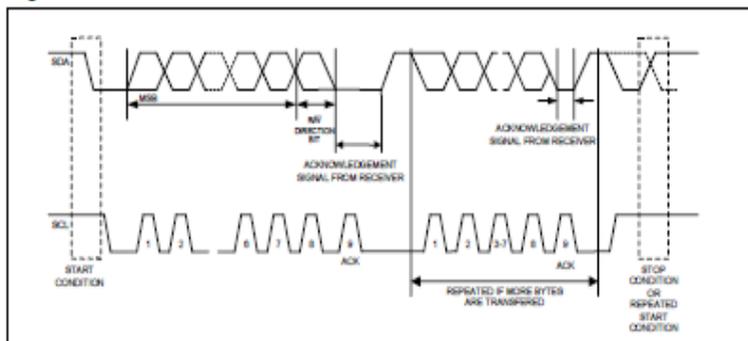
**STOP data transfer:** A change in the state of the data line, from LOW to HIGH, while the clock line is HIGH, defines the STOP condition.

**Data valid:** The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the HIGH period of the clock signal. The data on the line must be changed during the LOW period of the clock signal. There is one clock pulse per bit of data.

Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between START and STOP conditions is not limited, and is determined by the master device. The information is transferred byte-wise and each receiver acknowledges with a ninth bit. Within the I<sup>2</sup>C bus specifications a standard mode (100kHz clock rate) and a fast mode (400kHz clock rate) are defined. The DS1307 operates in the standard mode (100kHz) only.

**Acknowledge:** Each receiving device, when addressed, is obliged to generate an acknowledge after the reception of each byte. The master device must generate an extra clock pulse which is associated with this acknowledge bit.

A device that acknowledges must pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse. Of course, setup and hold times must be taken into account. A master must signal an end of data to the slave by not generating an acknowledge bit on the last byte that has been clocked out of the slave. In this case, the slave must leave the data line HIGH to enable the master to generate the STOP condition.

Figure 3. Data Transfer on I<sup>2</sup>C Serial Bus

Depending upon the state of the R/w bit, two types of data transfer are possible:

1. **Data transfer from a master transmitter to a slave receiver.** The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte. Data is transferred with the most significant bit (MSB) first.
2. **Data transfer from a slave transmitter to a master receiver.** The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. This is followed by the slave transmitting a number of data bytes. The master returns an acknowledge bit after all received bytes other than the last byte. At the end of the last received byte, a "not acknowledge" is returned.

The master device generates all the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the bus will not be released. Data is transferred with the most significant bit (MSB) first.

The DS1307 can operate in the following two modes:

1. **Slave Receiver Mode (Write Mode):** Serial data and clock are received through SDA and SCL. After each byte is received an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Hardware performs address recognition after reception of the slave address and direction bit (see Figure 4). The slave address byte is the first byte received after the master generates the START condition. The slave address byte contains the 7-bit DS1307 address, which is 1101000, followed by the direction bit (R/w), which for a write is 0. After receiving and decoding the slave address byte, the DS1307 outputs an acknowledge on SDA. After the DS1307 acknowledges the slave address + write bit, the master transmits a word address to the DS1307. This sets the register pointer on the DS1307, with the DS1307 acknowledging the transfer. The master can then transmit zero or more bytes of data with the DS1307 acknowledging each byte received. The register pointer automatically increments after each data byte are written. The master will generate a STOP condition to terminate the data write.
2. **Slave Transmitter Mode (Read Mode):** The first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit will indicate that the transfer direction is reversed. The DS1307 transmits serial data on SDA while the serial clock is input on SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer (see Figure 5). The slave address byte is the first byte received after the START condition is generated by the master. The slave address byte contains the 7-bit DS1307 address, which is 1101000, followed by the direction bit (R/w), which is 1 for a read. After receiving and decoding the slave address the DS1307 outputs an acknowledge on SDA. The DS1307 then begins to transmit data starting with the register address pointed to by the register pointer. If the register pointer is not written to before the initiation of a read mode the first address that is read is the last one stored in the register pointer. The register pointer automatically increments after each byte are read. The DS1307 must receive a Not Acknowledge to end a read.

Figure 4. Data Write—Slave Receiver Mode

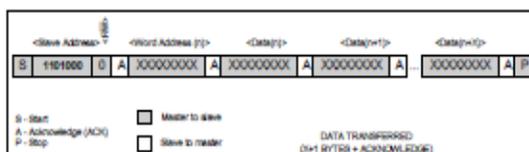


Figure 5. Data Read—Slave Transmitter Mode

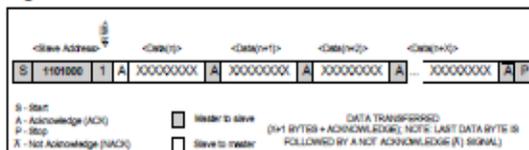
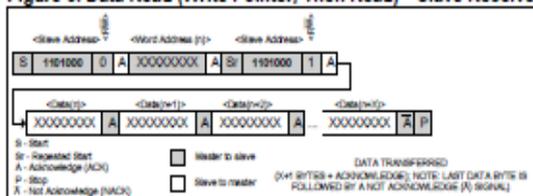


Figure 6. Data Read (Write Pointer, Then Read)—Slave Receive and Transmit



### PACKAGE INFORMATION

For the latest package outline information and land patterns, go to [www.maxim-ic.com/packages](http://www.maxim-ic.com/packages).

PACKAGE TYPE	PACKAGE CODE	DOCUMENT NO.
8 PDIP	—	<a href="#">21-0043</a>
8 SO	—	<a href="#">21-0041</a>

## REVISION HISTORY

REVISION DATE	DESCRIPTION	PAGES CHANGED
100208	Moved the <i>Typical Operating Circuit</i> and <i>Pin Configurations</i> to first page.	1
	Removed the leaded part numbers from the <i>Ordering information</i> table.	1
	Added an open-drain transistor to SQW/OUT in the block diagram (Figure 1).	4
	Added the pullup voltage range for SDA, SCL, and SQW/OUT to the <i>Pin Description</i> table and noted that SQW/OUT can be left open if not used.	6
	Added default time and date values on first application of power to the <i>Clock and Calendar</i> section and deleted the note that initial power-on state is not defined.	8
	Added default on initial application of power to bit info in the <i>Control Register</i> section.	9
	Updated the <i>Package information</i> section to reflect new package outline drawing numbers.	13
3/15	Updated <i>Benefits and Features</i> section	1

## B.2 Datasheet Arduino Mega 2560

### Arduino MEGA 2560



#### Product Overview

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

#### Index

Technical Specifications

Page 2

How to use Arduino  
Programming Environment, Basic Tutorials

Page 6

Terms & Conditions

Page 7

Environmental Policies  
half sqm of green via Impatto Zero®

Page 7



radiospares

RADIONICS



# Technical Specification

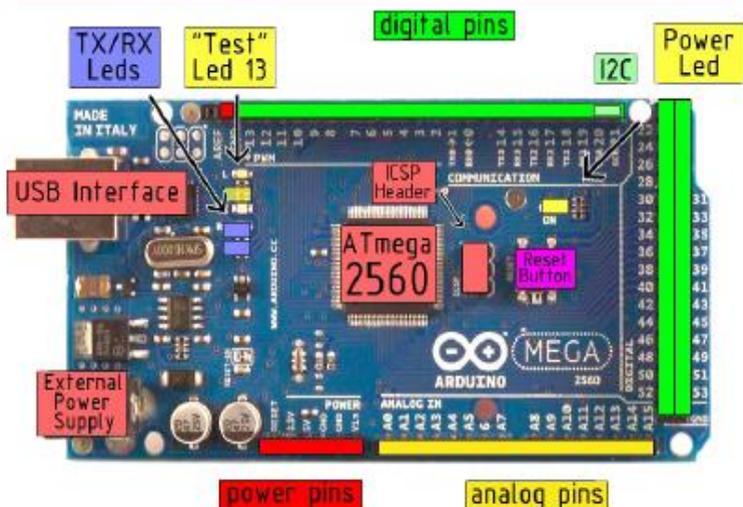


EAGLE files: [arduino-mega2560-reference-design.zip](#) Schematic: [arduino-mega2560-schematic.pdf](#)

## 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



radiospares

RADIONICS



## 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.
- **PC: 20 (SDA) and 21 (SCL).** Support I<sup>2</sup>C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I<sup>2</sup>C 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 it is 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.



radiospares

RADIONICS



## 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.



*radiospares*

**RADIONICS**



## Automatic (Software) Reset

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

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

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

## USB Overcurrent Protection

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

## Physical Characteristics and Shield Compatibility

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

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



radiospares

RADIONICS



# 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-platform 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 skeeth 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.

```
File Edit Sketch Tools Help
Sketch
int ledPin = 13; // LED connected to digital pin 13
// This setup() method runs once, when the boards starts
void setup() {
  // Initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}
// the loop() method runs over and over again,
// as long as the arduino has power.
void loop()
{
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000); // wait for a second
  digitalWrite(ledPin, LOW); // set the LED off
  delay(1000); // wait for a second
}
```



**radiospares RADIONICS**





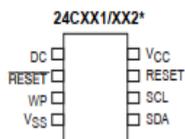
## B.3 Datasheet EEPROM AT24C02

### DESCRIPTION

The CAT24CXXX is a single chip solution to three popular functions of EEPROM memory, precision reset controller and watchdog timer. The 24C161/162(16K), 24C081/082(8K), 24C041/042(4K) and 24C021/022(2K) feature a I<sup>2</sup>C Serial CMOS EEPROM. Catalyst's advanced CMOS technology substantially reduces device power requirements. The 24CXXX features a 16-byte page and is available in 8-pin DIP or 8-pin SOIC packages.

The reset function of the 24CXXX protects the system during brown out and power up/down conditions. During system failure the watchdog timer feature protects the microcontroller with a reset signal. 24CXXX features active low reset on pin 2 and active high reset on pin 7. 24CXX1 features watchdog timer on the SDA line. 24CXX2 does not feature the watchdog timer function.

### PIN CONFIGURATION

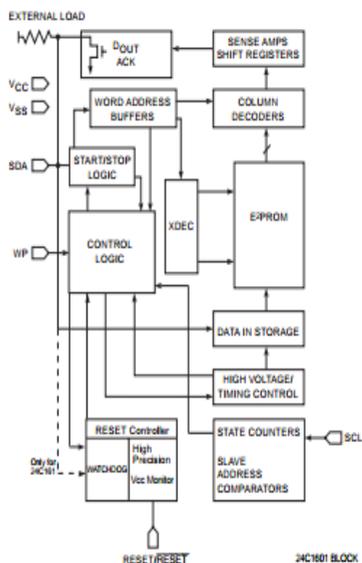


\*All products offered in P and J packages

### PIN FUNCTIONS

Pin Name	Function
SDA	Serial Data/Address
RESET/RESET <sup>̄</sup>	Reset I/O
SCL	Clock Input
Vcc	Power Supply
DC	Do Not Connect
VSS	Ground
WP	Write Protect

### BLOCK DIAGRAM



## RELIABILITY CHARACTERISTICS

Symbol	Parameter	Min.	Max.	Units	Reference Test Method
NEND <sup>(3)</sup>	Endurance	1,000,000		Cycles/Byte	MIL-STD-883, Test Method 1033
TDR <sup>(3)</sup>	Data Retention	100		Years	MIL-STD-883, Test Method 1008
VZAP <sup>(3)</sup>	ESD Susceptibility	2000		Volts	MIL-STD-883, Test Method 3015
I <sub>LTH</sub> <sup>(3)(4)</sup>	Latch-up	100		mA	JEDEC Standard 17

## D.C. OPERATING CHARACTERISTICS

V<sub>CC</sub> = +2.7V to +6.0V, unless otherwise specified.

Symbol	Parameter	Limits			Units	Test Conditions
		Min.	Typ.	Max.		
I <sub>CC</sub>	Power Supply Current			3	mA	f <sub>SCL</sub> = 100 KHz
I <sub>sb</sub>	Standby Current			40	μA	V <sub>CC</sub> =3.3V
				50	μA	V <sub>CC</sub> =5
I <sub>LI</sub>	Input Leakage Current			2	μA	V <sub>IN</sub> =GND or V <sub>CC</sub>
I <sub>LO</sub>	Output Leakage Current			10	μA	V <sub>IN</sub> =GND or V <sub>CC</sub>
V <sub>IL</sub>	Input Low Voltage	-1		V <sub>CC</sub> x 0.3	V	
V <sub>IH</sub>	Input High Voltage	V <sub>CC</sub> x 0.7		V <sub>CC</sub> + 0.5	V	
V <sub>OL</sub>	Output Low Voltage (SDA)			0.4	V	I <sub>OL</sub> = 3 mA, V <sub>CC</sub> =3.0V

CAPACITANCE T<sub>A</sub> = 25°C, f = 1.0 MHz, V<sub>CC</sub> = 5V

Symbol	Test	Max.	Units	Conditions
C <sub>IO</sub> <sup>(3)</sup>	Input/Output Capacitance (SDA)	8	pF	V <sub>IO</sub> = 0V
C <sub>IN</sub> <sup>(3)</sup>	Input Capacitance (SCL)	6	pF	V <sub>IN</sub> = 0V

Note:

- (1) The minimum DC Input voltage is -0.5V. During transitions, Inputs may undershoot to -2.0V for periods of less than 20 ns. Maximum DC voltage on output pins is V<sub>CC</sub> +0.5V, which may overshoot to V<sub>CC</sub> + 2.0V for periods of less than 20ns.
- (2) Output shorted for no more than one second. No more than one output shorted at a time.
- (3) This parameter is tested initially and after a design or process change that affects the parameter.
- (4) Latch-up protection is provided for stresses up to 100 mA on address and data pins from -1V to V<sub>CC</sub> +1V.

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

## RIWAYAT HIDUP



Nama : Amir Muhyidin  
TTL : Surabaya, 15 April  
1997  
Jenis Kelamin : Laki-Laki  
Agama : Islam  
Alamat : Perum Giri Asri C-13  
Kebomas, Gresik  
Telp/HP : 081332850470  
E-mail : amrmhydn@gmail.com

### RIWAYAT PENDIDIKAN

1. 2003 – 2009 : MINU Tratee Putra Gresik
2. 2009 – 2012 : SMP Negeri 2 Gresik
3. 2012 – 2015 : SMA Negeri 1 Manyar
4. 2015 – 2018 : D3 Teknik Elektro, Program Studi Teknik Elektro Industri - FV Institut Teknologi Sepuluh Nopember (ITS)

### PENGALAMAN KERJA

1. Kerja Praktek di PT. PLN (Persero), Gresik
2. Kerja Praktek di PT. PJB UP Gresik

### PENGALAMAN ORGANISASI

1. Staff Departemen Dalam Negeri Periode 2016/2017 HIMA D3TEKTRO, FV – ITS
2. Ketua Departemen Minat Bakat Periode 2017/2018 HIMA D3TEKTRO, FV - ITS

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