

TUGAS AKHIR

**PERANCANGAN RANGKAIAN INFORMASI GEMPA DITAMPILKAN PADA
*RUNNING TEXT***

Diajukan Sebagai Salah Satu Syarat Untuk Menyelesaikan Program Starata I Pada Jurusan
Teknik Elektro Fakultas Teknik Universitas Andalas

Oleh : Zuraida

Hannum NIM.

1110951005

Pembimbing :

Darwison. MT

NIP. 196409141995121001



JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS ANDALAS
PADANG

2015

LEMBAR PENGESAHAN

PERANCANGAN RANGKAIAN INFORMASI GEMPA
DITAMPILKAN PADA *RUNNING TEXT*

OLEH :

ZURAIDA HANNUM

NIM : 1110951005

LULUS SIDANG TUGAS AKHIR

27 JULI 2015

Mengetahui,

Ketua Jurusan Teknik Elektro



Dr.Eng. Ariadi Hazmi

NIP. 197503141999031003

Disetujui,

Pembimbing

Darwison, MT

NIP.196409141995121001

ABSTRAK

Indonesia merupakan wilayah yang rawan terhadap gempa, sehingga memiliki resiko gempa yang tinggi. Oleh karena itu dibutuhkan informasi yang cepat mengenai gempa tersebut. Dengan adanya informasi gempa, bahaya yang terjadi ketika terjadi gempa dapat diminimalisir sekecil mungkin. Skala *Modified Mercalli Intensity* (MMI) merupakan skala gempa yang digunakan untuk melihat intensitas gempa dan skala ini mudah dipahami, karena skala MMI telah dibagi menjadi 12 pecahan berdasarkan informasi dari orang-orang yang mengalami gempa tersebut dan melihat serta membandingkan tingkat kerusakan yang terjadi akibat gempa. Informasi gempa akan ditampilkan pada *running text* berdasarkan skala MMI yang dikelompokkan kedalam 4 bagian yaitu gempa kecil, gempa sedang, gempa berat dan gempa dasyat.

Kata Kunci : *Gempa, Modified Marcalli Intensity, Running text*

ABSTRACT

Indonesia is a region vulnerable to earthquakes, so it has a high seismic risk. Therefore, it needs quick information about the quake. With the earthquake information, the danger occurs when an earthquake can be minimized. Scale Modified Mercalli Intensity (MMI) scale earthquake is used to see the intensity of the earthquake and the scale is easy to understand, because the MMI scale has been divided into 12 fractions based on information from people - people who experienced the earthquake and see and compare the level of impairment that occurs quake. Earthquake information will be displayed on the running text is based on a scale of MMI, which are grouped into four sections, namely a small earthquake, earthquake moderate, severe earthquake and a catastrophic earthquake.

Keywords: Earthquake, Mercalli Modified Intensity, Running text

DAFTAR ISI

ABSTRAK	i
KATA PENGANTAR	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR	vii
DAFTAR TABEL	viii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Tujuan Penelitian	2
1.3 Manfaat Penelitian	3
1.4 Batasan Masalah	3
1.5 Metodologi Penelitian	3
1.6. Sistematika Penulisan	4
BAB II LANDASAN TEORI	5
2.1 Pengenalan Gempa Bumi	5
2.1.1 Penyebab Terjadinya Gempa Bumi	6
2.1.2 Intensitas Gempa Bumi	8
2.1.2.1 Peak Ground Acceleration (PGA).....	9
2.2 Sensor Gerak	11
2.3 Sensor Suhu dan Kelembaban	14
2.4 Arduino Mega 2560	16
2.4.1 Daya (Power) Arduino Mega 2560	18

2.4.2 Memory Arduino Mega 2560	20
2.4.3 Input dan Output Arduino Mega 2560	20
2.4.4 <i>Integrated Development Environment (IDE) Arduino</i>	22
2.5 <i>Dot Matrix</i>	23
BAB III METODOLOGI PENELITIAN	26
3.1 Studi Literatur.....	27
3.2 Perancangan Sistem	27
3.3 Pengujian Sistem Informasi Gempa.....	31
BAB IV HASIL DAN PEMBAHASAN	33
4.1 Hasil dan Pengujian Sistem.....	33
4.1.1 Sensor <i>Accelerometer</i> BMA220	33
4.1.2 Sensor SHT11	34
4.1.3 Tampilan pada <i>Dot Matrix</i>	35
4.2 Analisa Pengujian Sistem	36
4.2.1 Analisa Pengujian Sensor <i>Accelerometer</i> BMA220	36
4.2.2 Analisa Pengujian Sensor SHT11	39
4.2.3 Analisa Tampilan pada <i>Dot Matrix</i>	39
BAB V PENUTUP.....	40
5.1 Kesimpulan	40
5.2 Saran	40
DAFTAR PUSTAKA	41
LAMPIRAN	

DAFTAR GAMBAR

Gambar 2.1 Penampang Kerak Bumi.....	7
Gambar 2.2 Sensor <i>accelerometer</i> BMA220	12
Gambar 2.3 Modul SHT 11	15
Gambar 2.4 Arduino Mega 2560	17
Gambar 2.5 Arduino IDE.....	22
Gambar 2.6 Susunan LED pada <i>Dot Matrix Display</i> HS-2058A/B.....	23
Gambar 2.7 Susunan data biner untuk menampilkan karakter A	24
Gambar 3.1 Diagram Alir Penelitian	26
Gambar 3.2 Diagram Alir Cara Kerja Sistem Informasi Gempa	28
Gambar 3.3 Konfigurasi Pin pada Sistem Informasi Gempa	29
Gambar 3.4 Simulator Gempa.....	32
Gambar 4.1 Tampilan Gempa pada <i>Dot Matrix</i>	36
Gambar 4.2 Tampilan Suhu pada <i>Dot Matrix</i>	36
Gambar 4.3 Tampilan Kelembaban pada <i>Dot Matrix</i>	36

DAFTAR TABEL

Tabel 2.1 Satuan Ukuran Skala <i>Intensitas Mercalli</i>.....	9
Tabel 2.2 Persamaan MMI yang digunakan dan Penelitian Sebelumnya.....	11
Tabel 2.3 Akselerasi dan Resolusi <i>accelerometer</i> BMA220	13
Tabel 3.1 Konfigurasi Pin pada Sistem Informasi Gempa.....	30
Tabel 4.1 Nilai MMI dan tampilan pada LED <i>Dot Matrix</i>	34
Tabel 4.2 Hasil Pengukuran Nilai Suhu dan Kelembaban.....	35
Tabel 4.3 Pengelompokan Kekuatan Gempa	37

KATA PENGANTAR

Alhamdulillah ‘*aalamiin*. Segala puji hanya untuk Allah SWT, yang telah melimpahkan rahmat dan karunia-Nya kepada segenap makhluk di alam ini. Sehingga dengan rahmat Allah SWT tersebut penulis dapat menyelesaikan tugas akhir yang berjudul “**Perancangan Rangkaian Informasi Gempa Ditampilkan pada *Running Text*”**”.

Tugas akhir ini merupakan salah satu syarat untuk menyelesaikan pendidikan strata-1 (sarjana) sesuai dengan kurikulum yang berlaku di lingkungan Jurusan Teknik Elektro Universitas Andalas. Laporan ini penulis susun berdasarkan hasil penelitian dan referensi yang didapatkan dari berbagai sumber.

Penulis menyadari tugas akhir ini dapat terselesaikan atas bantuan dari berbagai pihak yang telah menyumbangkan pikiran, memberikan bimbingan, arahan, diskusi, koreksi serta dukungan moral yang tak berhingga nilainya bagi penulis. Untuk itu penulis menyampaikan rasa hormat dan mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak Ariadi Hazmi, Dr. Eng. Selaku Ketua Jurusan Teknik Elektro Universitas Andalas.
2. Bapak Darwison.MT, sebagai pembimbing penulis dalam melaksanakan tugas akhir ini, yang telah memberikan segala bantuan bimbingan, arahan, wawasan, semangat dan didikan yang sangat berharga selama penulis menyusun tugas akhir ini.

3. Uda Wahyu, Aulia abdi, asisten Laboratorium Elektronika Industri (LEI), dan rekan-rekan bimbingan yang memberikan masukan dan waktu untuk berdiskusi bersama penulis dalam penelitian ini.
4. Rekan-rekan mahasiswa Jurusan Teknik Elektro UNAND yang tidak dapat penulis sebutkan satu persatu yang selalu memberikan semangat kepada penulis.

Penulis menyadari bahwa tugas akhir ini masih sangat jauh dari kesempurnaan, cara penyajian maupun teknik penulisan. Oleh karena itu penulis mengharapkan kritik dan saran yang konstruktif demi menyempurnakan tugas akhir ini, kritik dan saran dapat disampaikan melalui email zuraidahannum@yahoo.com. Semoga tugas akhir ini dapat bermanfaat bagi kita semua dan bagi penulis sendiri tentunya.

Padang, Juli 2015

Penulis

BAB I

PENDAHULUAN

1.1 Latar Belakang

Indonesia termasuk dalam wilayah yang sangat rawan bencana gempa bumi seperti halnya Jepang dan California karena posisi geografisnya menempati zona tektonik yang sangat aktif. Hal ini dikarenakan tiga lempeng besar dunia dan sembilan lempeng kecil lainnya saling bertemu di wilayah Indonesia serta membentuk jalur-jalur pertemuan lempeng yang kompleks. Keberadaan interaksi antar lempeng-lempeng ini menempatkan wilayah Indonesia sebagai wilayah yang sangat rawan terhadap gempa bumi.^[1]

Gempa merupakan suatu bencana alam yang bisa terjadi kapan saja, tanpa memberikan peringatan kedatangannya. Meskipun demikian, konsentrasi gempa cenderung terjadi di tempat – tempat tertentu saja, seperti pada batas lempeng Plat Pasifik. Untuk mengetahui kecepatan pergerakan tanah dan untuk mengukur besarnya suatu gempa dapat digunakan suatu alat yaitu seismograf. Seismograf akan memantau gerakan – gerakan bumi. Skala intensitas *Marcalli* akan menunjukkan intensitas gempa atau pengaruh gempa terhadap tanah, gedung dan manusia. Berdasarkan penjelasan tersebut maka diperlukan suatu sistem yang dapat memberikan informasi kepada masyarakat supaya mampu membedakan gempa yang berbahaya dan gempa yang tidak berbahaya.

Dengan berkembangnya teknologi serta tingkat mobilitas manusia yang semakin meningkat maka memudahkan untuk penyampaian informasi. *LED Dot matrix* merupakan salah satu contoh peralatan elektronik yang digunakan untuk

penyampaian informasi yang berupa *display* dengan resolusi kecil untuk menampilkan informasi berupa teks dan karakter. *LED Dot matrix* sangat diminati untuk penyampaian informasi berupa iklan layanan masyarakat, media kampanye, iklan wisata, dan lainnya.

Berdasarkan pemikiran di atas, penulis tertarik untuk merancang sebuah rangkaian informasi gempa yang ditampilkan pada *running text* sebagai tugas akhir. Judul tugas akhir yang penulis angkat adalah “**Perancangan Rangkaian Informasi Gempa Ditampilkan Pada *Running Text***”.

Adapun penelitian yang berkaitan dengan sistem, diantaranya :

1. **Ahmad Fulki** dalam penelitiannya yang berjudul “Analisis Parameter *b value* dan PGA di Daerah Papua”. Dalam penelitian ini dilakukan pencarian nilai *b value* dengan menggunakan rumus likelihood maksimum, dan mencari nilai PGA dengan cara melakukan perhitungan menggunakan rumus empiris percepatan tanah dengan perhitungan rumus empiris Estave untuk mencari rumus empiris yang baru.

1.2 Tujuan

Adapun tujuan penulisan tugas akhir ini adalah :

1. Merancang sebuah sistem penyampaian informasi gempa dengan menampilkannya pada *running text*.
2. Merancang sistem penyampaian informasi suhu dan kelembaban dengan menampilkannya pada *running text*

1.3 Manfaat Penelitian

Dari tugas akhir ini, diharapkan dapat memberikan informasi gempa bumi kepada masyarakat sehingga mampu membedakan gempa yang berbahaya dan gempa yang tidak berbahaya.

1.4 Batasan Masalah

Adapun batasan masalah dari tugas akhir ini adalah :

1. Pengendali LED *Dot Matrix* yang digunakan adalah Mikrokontroler Arduino MEGA 2560.
2. Sensor yang digunakan adalah sensor gerak *accelerometer* BMA 220 .
3. Sensor Suhu dan Kelembapan SHT 11.

1.5 Metodologi Penelitian

Metodelogi penelitian yang digunakan pada tugas akhir ini adalah :

1. Studi literatur sebagai gambaran awal mengenai konsep dari masalah yang akan dibahas pada tugas akhir ini.
2. Perancangan sistem informasi gempa, suhu dan kelembapan pada LED *Dot Matrix*.
3. Pengujian sistem informasi gempa, suhu dan kelembapan pada LED *Dot Matrix*.
4. Penyusunan laporan akhir.

1.6 Sistematika Penulisan

BAB I : Pendahuluan, berisi latar belakang, tujuan penelitian, manfaat penelitian, batasan masalah, metode penelitian, dan sistematika penulisan.

BAB II : Landasan Teori, penjelasan teori dasar yang berhubungan dengan gempa dan perangkat keras yang digunakan.

BAB III : Metodologi Penelitian, berisikan tentang langkah-langkah beserta penjelasan mengenai penelitian yang dilakukan

BAB IV : Hasil dan Pembahasan, ini berisikan analisis hasil penelitian.

BAB V : Penutup, berisikan beberapa kesimpulan dan saran yang bisa ditarik dan disampaikan yang didasari dari hasil dan pembahasan penelitian ini.

BAB II

Dasar Teori

2.1 Pengenalan Gempa Bumi

Gempa bumi adalah peristiwa bergetarnya bumi akibat pelepasan energi di dalam bumi secara tiba-tiba yang ditandai dengan patahnya lapisan batuan pada kerak bumi. Akumulasi energi penyebab terjadinya gempa bumi dihasilkan dari pergerakan lempeng-lempeng tektonik. Energi yang dihasilkan dipancarkan kesegala arah berupa gelombang gempab umi sehingga efeknya dapat dirasakan sampai ke permukaan bumi^[2].

Di bawah kerak bumi terdapat lapisan lunak terbentuk dari batuan panas yang lumer. Kerak bumi yang terbentuk dari nikel dan besi dengan bahagian yang padat ditengahnya. Kerak tersebut bisa mencapai ketebalan 70 km di bawah barisan pengunungan terbesar di dunia. Kebanyakan gempa bumi berasal dari kerak bumi. Kadang-kadang gempa bumi juga bisa terjadi pada kedalaman 700 km di bawah permukaan bumi. Berdasarkan kedalaman dari posisi gempa, gempa dapat dikategorikan atas 3 kategori:

1. Gempa dangkal, (hiposenternya terletak pada kedalaman 0 – 65 km)
2. Gempa sedang, (hiposenternya terletak pada kedalaman 65 – 200 km)
3. Gempa dalam, (hiposenternya terletak pada kedalaman > 200 km)

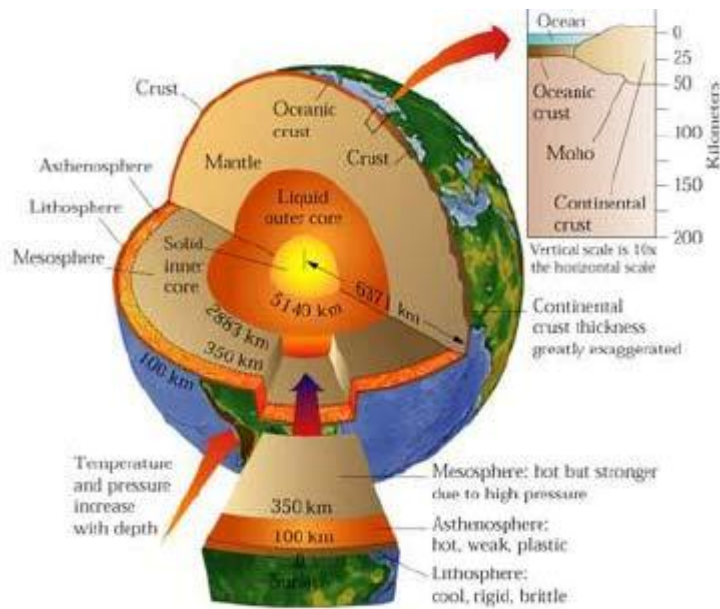
Sedangkan menurut sumber terjadinya gempa, gempa dikelompokkan menjadi^[3]:

1. Gempa tektonik yaitu gempa bumi yang berasal dari pergeseran lapisan – lapisan batuan sepanjang bidang patahan di dalam bumi.

2. Gempa vulkanik yaitu gempa bumi yang berasal dari pergerakan magma karena aktifitas gunung api.
3. Gempa longsor atau runtuh yaitu gempa bumi yang terjadi karena aktifitas runtuh pada daerah pertambangan atau daerah tanah longsor.
4. Gempa buatan yaitu getaran gempa bumi yang terjadi karena adanya aktivitas manusia di kulit bumi yang menyebabkan getaran yang cukup kuat.

2.1.1 Penyebab Terjadinya Gempa Bumi

Menurut teori lempeng tektonik, permukaan bumi terpecah menjadi beberapa lempeng tektonik besar. Lempeng tektonik adalah segmen keras kerak bumi yang mengapung diatas astenosfer yang cair dan panas. Oleh karena itu, maka lempeng tektonik ini bebas untuk bergerak dan saling berinteraksi satu sama lain. Daerah perbatasan lempeng-lempeng tektonik, merupakan tempat-tempat yang memiliki kondisi tektonik yang aktif, yang menyebabkan gempa bumi, gunung berapi dan pembentukan dataran tinggi. Teori lempeng tektonik merupakan kombinasi dari teori sebelumnya yaitu: Teori Pergerakan Benua (*Continental Drift*) dan Pemekaran Dasar Samudra (*Sea Floor Spreading*).



Gambar 2.1 Penampang Kerak Bumi^[2]

Gambar 2.1 merupakan penampang kerak bumi. Lapisan paling atas bumi, yaitu litosfir, merupakan batuan yang relatif dingin dan bagian paling atas berada pada kondisi padat dan kaku. Di bawah lapisan ini terdapat batuan yang jauh lebih panas yang disebut mantel. Lapisan ini sedemikian panasnya sehingga senantiasa dalam keadaan tidak kaku, sehingga dapat bergerak sesuai dengan proses pendistribusian panas yang kita kenal sebagai aliran konveksi. Lempeng tektonik yang merupakan bagian dari litosfir padat dan terapung di atas mantel ikut bergerak satu sama lainnya. Ada tiga kemungkinan pergerakan satu lempeng tektonik relatif terhadap lempeng lainnya, yaitu apabila kedua lempeng saling menjauhi (*spreading*), saling mendekati (*collision*) dan saling geser (*transform*).

Jika dua lempeng bertemu pada suatu sesar/patahan bumi, keduanya dapat bergerak saling menjauhi, saling mendekati atau saling bergeser. Umumnya, gerakan

ini berlangsung lambat dan tidak dapat dirasakan oleh manusia namun terukur sebesar 0-15cm pertahun. Kadang-kadang, gerakan lempeng ini macet dan saling mengunci, sehingga terjadi pengumpulan energi yang berlangsung terus sampai pada suatu saat batuan pada lempeng tektonik tersebut tidak lagi kuat menahan gerakan tersebut sehingga terjadi pelepasan mendadak yang kita kenal sebagai gempa bumi.

2.1.2 Intensitas Gempa

Intensitas dapat dipersepsikan pada pengukuran non-instrumental dari suatu kerusakan struktur, efek permukaan tanah, retakan dan longsor, dan reaksi manusia terhadap guncangan gempa bumi. Secara sederhana dideskripsikan sebagai ukuran gempa bumi. Skala Intensitas pertama kali diperkenalkan di Eropa oleh Rossi-Forel (1883), yang terdiri dari 10 skala. Selanjutnya Skala Sieberg (1912,1923) menjadi pondasi dari semua skala intensitas modern 12 tingkat.^[4]

Versi terbaru skala intensitas dikenal sebagai Skala Mercalli Cancani-Sieberg atau skala MCS (Sieberg 1932) yang hingga kini masih digunakan di Eropa bagian selatan. Lalu pada Wood dan Neuman (1931) melakukan modifikasi pada skala tersebut dan menerjemahkannya dalam bahasa Inggris dan dikenal dengan nama Skala *Modified Mercalli* (Skala MM). Pada tahun 1956, Richter (1956) menyempurnakan beberapa pernyataan dalam skala bentuk ringkasan dan menambahkan penjelasan mengenai tipe bangunannya, yang dikenal sebagai Skala *Modified Mercalli* 1956 (MM56) yang digunakan Australia dan Selandia Baru. Adapun Satuan ukuran skala *Modified Intensitas Mercalli* dapat dilihat pada tabel 2.1.

Tabel 2.1 Satuan Ukuran Skala *Modified Intensitas Mercalli*^[5]

MMI	Keterangan
I	Tidak terasa
II	Terasa oleh orang yang berada di bangunan tinggi
III	Getaran dirasakan seperti ada kereta yang berat melintas.
IV	Getaran dirasakan seperti ada benda berat yang menabrak dinding rumah, benda tergantung bergoyang.
V	Dapat dirasakan di luar rumah, hiasan dinding bergerak, benda kecil di atas rak mampu jatuh.
VI	Terasa oleh hampir semua orang, dinding rumah rusak.
VII	Dinding pagar yang tidak kuat pecah, orang tidak dapat berjalan/berdiri.
VIII	Bangunan yang tidak kuat akan mengalami kerusakan.
IX	Bangunan yang tidak kuat akan mengalami kerusakan tekuk.
X	Jembatan dan tangga rusak, terjadi tanah longsor. Rel kereta api bengkok.
XI	Rel kereta api rusak. Bendungan dan tanggul hancur. Seluruh bangunan hampir hancur dan terjadi longsor besar. Efek bencana yang lain seperti tsunami, dan kebakaran.
XII	Seluruh bangunan hancur lebur. Batu dan barang-barang terlempar ke udara. Tanah bergerak seperti gelombang. Kadang- kadang aliran sungai berubah. Pasir dan lumpur bergeser secara horizontal. Air dapat terlempar dari danau, sungai dan kanal. Diikuti dengan suara gemuruh yang besar. Biasanya bisa menyebabkan longsor besar, kebakaran, banjir, tsunami di daerah pantai, dan aktivitas gunung berapi. Pasir dan tanah halus terlihat meledak.

2.1.2.1 Peak Ground Acceleration (PGA)

Peak ground acceleration (PGA) adalah ukuran bagaimana permukaan bumi bergetar (*accelerated*) disuatu daerah tertentu. *Acceleration* pada istilah *peak ground acceleration* secara umum bisa dirasakan sebagai perubahan dari kecepatan dalam satu waktu. Nilai terbesar dari beberapa perubahan kecepatan dalam satuan waktu,

inilah yang dikenal sebagai nilai peak. Secara umum *peak ground acceleration* ini dapat diartikan sebagai akselerasi maksimum yang dirasakan suatu lapisan /partikel ketika terjadi gerakan gempa.

Hubungan antara skala intensitas dan percepatan puncak, intensitas guncangan pada skala rendah berhubungan secara wajar dengan kedua PGA (*Peak Ground Acceleration*) dan PGV (*Peak Ground Velocity*), sedangkan pada intensitas tinggi berkorelasi baik dengan kecepatan puncak. Secara mendasar, percepatan puncak melampaui level intensitas tinggi ketika kecepatan puncak mengalami kenaikan (Wald, *et al.*, 1999).^[6]

Estimasi PGA sangat bergantung pada magnitude. Menurut USGS (2012), magnitudo merupakan angka yang menunjukkan karakteristik ukuran relatif dari suatu gempa bumi. Magnitudo didasarkan pada pengukuran gerakan maksimum pada rekaman seismograf. Penentuan nilai PGA dapat dilakukan dengan menggunakan alat ukur maupun perasamaan prediksi getaran tanah. Banyak sekali persamaan yang dapat digunakan untuk menghitung prediksi getaran tanah. Hasil nilai PGA ini berguna untuk merepresentasikan distribusi tingkat resiko gempa. Nilai PGA tinggi menandakan bahwa wilayah tersebut sering terjadi gempa (probabilitas kejadian gempa sangat besar), sedangkan nilai PGA rendah berarti jarang terjadi gempa atau memiliki bahaya gempa yang lebih intensif.

Pada tugas akhir ini digunakan rumus MMI dan nilai PGA untuk wilayah Turki. Persamaan MMI dan nilai PGA yang digunakan pada penelitian ini merupakan perbandingan dengan penelitian sebelumnya. Tabel 2.2 merupakan persamaan MMI yang digunakan dan penelitian sebelumnya.

Tabel 2.2 Persamaan MMI yang digunakan dan Penelitian Sebelumnya^[7]

No	Nama	Persamaan
1	Persamaan yang digunakan	$MMI=0.287+3.625*\log(PGA)$
2	Arjo lu et al. (2001)	$MMI=1.748*\ln(PGA)-1.078$
3	Tselentis and Danciu (2008)	$MMI=-0.946+3.563*\log(PGA)$
4	Faenza and Michelini (2010)	$MMI=1.68+2.58*\log(PGA)$
5	Murphy and O'Brien (1977)	$MMI=(\log(PGA)-0.25)/0.25$
6	Trifunac and Brady (1975)	$MMI=(\log(PGA)-0.14)/0.30$
7	Wald et al. (1999)	$MMI=3.66*\log(PGA)-1.66$

Persamaan MMI yang digunakan dapat dilihat pada persamaan 2.1 :

$$MMI = 0.287 + 3.625 \log(PGA) \quad (2.1)$$

2.2 Sensor Gerak

Sensor adalah sebuah alat yang mampu merubah besaran fisik seperti gaya, kecepatan perputaran dan penerangan menjadi besaran listrik yang sebanding, disebut juga alat elektronik yang bisa mengubah fenomena alam sekitar menjadi sinyal elektronik. Sensor dapat digunakan sebagai sumber *input* bagi keseluruhan sistem.^[8]

Accelerometer adalah alat yang digunakan untuk mengukur percepatan, mendeteksi dan mengukur getaran (*vibrasi*), dan mengukur percepatan akibat gravitasi (*inklinasi*).^[9] Sensor *accelerometer* mengukur percepatan akibat gerakan benda yang melekat padanya. *Accelerometer* dapat digunakan untuk mengukur getaran pada mobil, mesin, bangunan, dan instalasi pengamanan.

Pada tugas akhir ini sensor *accelerometer* yang digunakan adalah sensor BMA220. Adapun gambar *accelerometer* ditunjukkan pada gambar 2.2.



Gambar 2.2 Sensor *accelerometer* BMA220^[10]

Fitur Sensor *accelerometer* BMA220 yaitu :

1. *Power supply* : 2.0-3.6V
2. *Interface* : I2C
3. *Acceleration range* : $\pm 2g/\pm 4g/\pm 8g/\pm 16g$
4. Konsumsi daya yang sangat rendah
5. LED sebagai indikator daya
6. *Design* ukuran yang mungil sehingga mudah digunakan
7. *Compatible* dengan *Arduino controllers*
8. *Size* : 13x22mm

Sensor *accelerometer* BMA220 memiliki 4 pengaturan jangkauan yang berbeda untuk berbagai akselerasi skala penuh, penggunaannya selalu kisaran skala

penyut terbawah dengan pemilihah resolusi maksimal. Tabel 2.3 merupakan tabel Resolusi dan akselerasi dari *accelerometer* BMA220 :

Tabel 2.3 Akselerasi dan Resolusi *accelerometer* BMA220^[11]

<i>Range</i>	<i>Full Scale</i>	<i>Sensitivity</i>	<i>Resolution</i>	<i>Exampel to Use Case</i>
00	±2g	16 LSB/g	62.5 mg/LSB	<i>Orientation</i>
01	±4g	8 LSB/g	125 mg/LSB	<i>Orientation</i>
10	±8g	4 LSB/g	0.25 g/LSB	<i>Gaming</i>
11	±16g	2 LSB/g	0.5 g/LSB	<i>SShock Vibratoin Detection</i>

Accelerometer BMA220 memiliki 3 *variable* yaitu x, y dan z. *Variable* x, y dan z merupakan keluaran percepatan pada sumbu x, y dan z dalam satuan m/s². Pada tugas akhir ini digunakan resolusi 125 mg/LSB pada *accelerometer*. Untuk mengkonversi keluaran *accelerometer* pada sumbu x, y dan z kedalam satuan m/s² maka digunakan rumus pada persamaan 2.2 sampai 2.4 :

$$a_x = \frac{V_{out}}{S} \cdot \frac{g}{LSB} \quad \dots\dots (2.2)$$

$$a_y = \frac{V_{out}}{S} \cdot \frac{g}{LSB} \quad \dots\dots (2.3)$$

$$a_z = \frac{V_{out}}{S} \cdot \frac{g}{LSB} \quad \dots\dots (2.4)$$

Keterangan : x = sumbu x pada *accelerometer*

y = sumbu y pada *accelerometer*

z = sumbu z pada *accelerometer*

Setelah di dapat nilai x, y dan z kemudian dihitung nilai resultannya (R) dengan menggunakan rumus pada 2.5 :

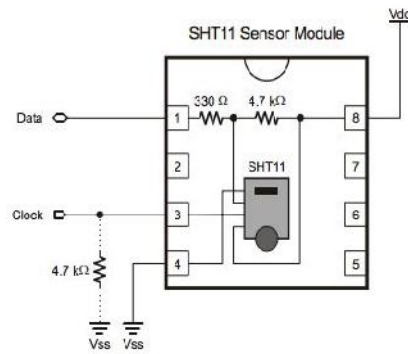
$$R = \sqrt{x^2 + y^2 + z^2} \quad (2.5)$$

Accelerometer BMA220 menghitung semua nilai percepatan termasuk percepatan gravitasi, jadi pada saat sensor diam tetap memiliki nilai percepatan yaitu 9.8 m/s^2 . Sehingga untuk menghitung keluaran sensor *accelerometer* digunakan rumus pada persamaan 2.6 :

$$a = \frac{R}{9.8} \quad (2.6)$$

2.3 Sensor Suhu dan Kelembaban

Sensor yang digunakan pada alat yang dibuat ini adalah sensor SHT11. SHT11 merupakan modul sensor suhu dan kelembaban relatif dari *Sensirion*. Modul ini dapat digunakan sebagai alat pengindra suhu dan kelembaban dalam aplikasi pengendali suhu dan kelembaban ruangan maupun aplikasi pemantau suhu dan kelembaban relatif ruangan. Adapun gambar modul SHT11 dapat dilihat pada gambar 2.3



Gambar 2.3 Modul SHT11^[12]

Spesifikasi dari SHT11 ini adalah sebagai berikut:

1. Berbasis sensor suhu dan kelembaban relatif Sensirion SHT11.
2. Mengukur suhu dari -40C hingga +123,8C, atau dari -40F hingga +254,9F dan kelembaban relatif dari 0%RH hingga 100%RH.
3. Memiliki ketetapan pengukuran suhu hingga 0,5C pada suhu 25C dan ketepatan pengukuran kelembaban relatif hingga 3,5%RH.
4. Memiliki atarmuka serial *synchronous 2-wire*, bukan I2C.
5. Jalur antarmuka telah dilengkapi dengan rangkaian pencegah kondisi sensor *lock-up*.
6. Membutuhkan catu daya +5V DC dengan konsumsi daya rendah 30 μ W.
7. Modul ini memiliki faktor bentuk 8 pin DIP 0,6 sehingga memudahkan pemasangannya.

SHT11 adalah sebuah *single chip* sensor suhu dan kelembaban relatif dengan multi modul sensor yang outputnya telah dikalibrasi secara digital. Dibagian dalamnya terdapat kapasitas polimer sebagai elemen untuk sensor kelembaban relatif

dan sebuah pita regangan yang digunakan sebagai sensor temperatur. Output kedua sensor digabungkan dan dihubungkan pada ADC 14 bit dan sebuah interface serial pada satu chip yang sama. Sensor ini menghasilkan sinyal keluaran yang baik dengan waktu respon yang cepat. SHT11 ini dikalibrasi pada ruangan dengan kelembaban yang teliti menggunakan hygrometer sebagai referensinya. Koefisien kalibrasinya telah diprogramkan kedalam OTP memory. Koefisien tersebut akan digunakan untuk mengkalibrasi keluaran dari sensor selama proses pengukuran.

2.4 Arduino Mega 2560

Arduino merupakan sebuah *platform* dari *physical computing* yang bersifat *open source*. Arduino adalah kombinasi dari *hardware*, bahasa pemrograman, dan *Integrated Development Environment* (IDE) yang canggih^[13]. IDE adalah sebuah software yang sangat berperan untuk menulis program, meng-*compile* menjadi kode biner dan meng-*upload* kedalam memori mikrokontroler.

Arduino Board terdiri dari sebuah mikroprosesor Atmel AVR, Kristal atau osilator (Pengukuran waktu dasar yang mengirimkan *timepulses* pada frekuensi tertentu agar dapat beroperasi pada kecepatan yang benar) dan regulator 5V. Pada beberapa jenis arduino juga ada yang memiliki soket USB untuk memungkinkannya dihubungkan ke PC atau Mac untuk meng-*upload* atau mengambil data. Dengan meng-*expose* I/O (*input/output*) pin mikrokontroler dapat dihubungkan dengan sensor atau perangkat elektronika lain yang berhubungan dengan pekerjaan tertentu.

Yang membuat Arduino dengan cepat diterima oleh orang-orang adalah karena:

- Murah, dibandingkan platform yang lain.

- Lintas platform, software Arduino dapat dijalankan pada system operasi Windows, Macintosh OSX dan Linux, sementara platform lain umumnya terbatas hanya pada Windows.
- Sangat mudah dipelajari dan digunakan. *Processing* adalah bahasa pemrograman yang digunakan untuk menulis program di dalam Arduino. *Processing* adalah bahasa pemrograman tingkat tinggi yang dialeknnya sangat mirip dengan C++ dan Java, sehingga pengguna yang sudah terbiasa dengan kedua bahasa tersebut tidak akan menemui kesulitan dengan *Processing*. Bahasa pemrograman *processing* sungguh-sungguh sangat memudahkan dan mempercepat pembuatan sebuah program karena bahasa ini sangat mudah dipelajari dan diaplikasikan dibandingkan bahasa pemrograman tingkat rendah seperti Assembler yang umum digunakan pada platformlain namun cukup sulit.
- Sistem yang terbuka, baik dari sisi *hardware* maupun *software*-nya.

Gambar 2.4 merupakan gambar arduino mega 2560 :



Gambar 2.4 Arduino Mega 2560^[13]

Arduino Mega 2560 adalah *board* berbasis mikrokontroler pada ATmega2560. Board ini memiliki 54 digital *input / output* pin (dimana 15 pin dapat digunakan sebagai *output* PWM), 16 *input analog*, koneksi USB, dan *jack* listrik tombol *reset*.

Untuk menghubungkan Arduino ke komputer diperlukan kabel USB. Kabel USB dapat digunakan untuk beberapa tujuan^[14]:

- Meng-*upload software* baru pada papan.
- Berkomunikasi dengan papan Arduino dari komputer Anda.
- Menyalurkan daya ke dalam papan Arduino.

2.4.1 Daya (*Power*) Arduino Mega 2560

Arduino Mega dapat diaktifkan melalui koneksi USB atau dengan catu daya *eksternal*. Sumber daya dipilih secara otomatis. Sumber daya *eksternal (non-USB)* dapat berasal baik dari adaptor AC-DC atau baterai. Adaptor dapat dihubungkan dengan mencolokkan steker 2,1 mm yang bagian tengahnya terminal positif ke *jack* sumber tegangan pada papan. Jika tegangan berasal dari baterai dapat langsung dihubungkan melalui *header* pin Gnd dan pin Vin dari konektor *power*^[15].

Papan Arduino ATmega2560 dapat beroperasi dengan pasokan daya *eksternal* 6 Volt sampai 20 volt. Jika diberi tegangan kurang dari 7 Volt, maka, pin 5 Volt mungkin akan menghasilkan tegangan kurang dari 5 Volt dan ini akan membuat papan menjadi tidak stabil. Jika sumber tegangan menggunakan lebih dari 12 Volt,

regulator tegangan akan mengalami panas berlebihan dan bisa merusak papan.

Rentang sumber tegangan yang dianjurkan adalah 7 Volt sampai 12 Volt.

Pin tegangan yang tersedia pada papan Arduino adalah sebagai berikut :

- VIN : Adalah input tegangan untuk papan Arduino ketika menggunakan sumber daya eksternal (sebagai 'saingan' tegangan 5 Volt dari koneksi USB atau sumber daya ter-regulator lainnya). Tegangan dapat diberikan melalui pin ini, atau jika memasok tegangan untuk papan melalui *jack power*, maka bisa mengakses/mengambil tegangan melalui pin ini.
- 5 Volt : Sebuah pin yang mengeluarkan tegangan ter-regulator 5 Volt, dari pin ini tegangan sudah diatur (*ter-regulator*) dari regulator yang tersedia (*built-in*) pada papan. Arduino dapat diaktifkan dengan sumber daya baik berasal dari jack power DC (7-12 Volt), konektor USB (5 Volt), atau pin VIN pada board (7-12 Volt). Memberikan tegangan melalui pin 5 Volt atau 3.3 Volt secara langsung tanpa melewati *regulator* dapat merusak papan Arduino.
- 3.3 Volt : Sebuah pin yang menghasilkan tegangan 3,3 Volt. Tegangan ini dihasilkan oleh *regulator* yang terdapat pada papan (*on-board*). Arus maksimum yang dihasilkan adalah 50 mA.
- GND : Pin *Ground* atau Massa.
- IOREF : Pin ini pada papan Arduino berfungsi untuk memberikan referensi tegangan yang beroperasi pada mikrokontroler. Sebuah perisai (*shield*) dikonfigurasi dengan benar untuk dapat membaca pin tegangan IOREF dan memilih sumber daya yang tepat atau mengaktifkan penerjemah tegangan

(*voltage translator*) pada output untuk bekerja pada tegangan 5 Volt atau 3,3 Volt.

2.4.2 Memori Arduino Mega 2560

Arduino ATmega2560 memiliki 256 KB flash memory untuk menyimpan kode (yang 8 KB digunakan untuk bootloader), 8 KB SRAM dan 4 KB EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM).

2.4.3 Input dan Output

Masing-masing dari 54 digital pin pada Arduino Mega dapat digunakan sebagai input atau output, menggunakan fungsi `pinMode()` , `digitalWrite()` , dan `digitalRead()`. Arduino Mega beroperasi pada tegangan 5 volt. Setiap pin dapat memberikan atau menerima arus maksimum 40 mA dan memiliki resistor pull-up internal (yang terputus secara *default*) sebesar 20-50 kOhms. Selain itu, beberapa pin memiliki fungsi khusus, antara lain:

- Serial : 0 (RX) dan 1 (TX); Serial 1 : 19 (RX) dan 18 (TX); Serial 2 : 17 (RX) dan 16 (TX); Serial 3 : 15 (RX) dan 14 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL. Pins 0 dan 1 juga terhubung ke pin chip ATmega16U2 Serial USB-to-TTL.
- Eksternal Interupsi : Pin 2 (interrupt 0), pin 3 (interrupt 1), pin 18 (interrupt 5), pin 19 (interrupt 4), pin 20 (interrupt 3), dan pin 21 (interrupt 2). Pin ini dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun, atau perubah nilai.

- SPI : Pin 50 (MISO), pin 51 (MOSI), pin 52 (SCK), pin 53 (SS). Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI. Pin SPI juga terhubung dengan *header* ICSP, yang secara fisik kompatibel dengan Arduino Uno, Arduino Duemilanove dan Arduino Diecimila.
- LED : Pin 13. Tersedia secara built-in pada papan Arduino ATmega2560. LED terhubung ke pin digital 13. Ketika pin diset bernilai *HIGH*, maka LED menyala (ON), dan ketika pin diset bernilai *LOW*, maka LED padam (*OFF*).
- TWI : Pin 20 (SDA) dan pin 21 (SCL). Yang mendukung komunikasi TWI menggunakan perpustakaan Wire. Perhatikan bahwa pin ini tidak di lokasi yang sama dengan pin TWI pada Arduino Duemilanove atau Arduino Diecimila.

Arduino Mega2560 memiliki 16 pin sebagai *analog input*, yang masing-masing menyediakan resolusi 10 bit (yaitu 1024 nilai yang berbeda). Secara default pin ini dapat diukur/diatur dari mulai *ground* sampai 5 Volt, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan pin AREF dan fungsi *analog Reference()*.

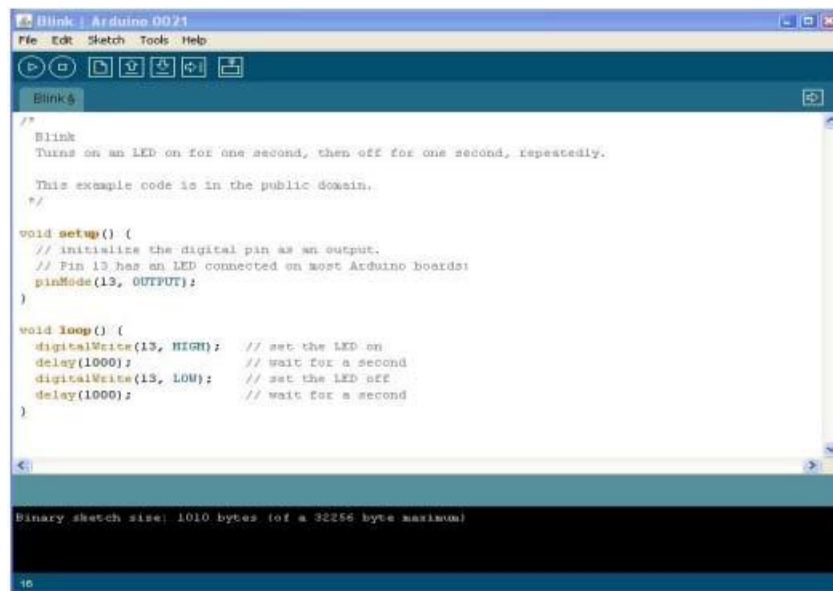
Ada beberapa pin lainnya yang tersedia, antara lain:

- AREF : Referensi tegangan untuk input analog. Digunakan dengan fungsi *analogReference()*.
- RESET : Jalur *LOW* ini digunakan untuk *me-reset* (menghidupkan ulang) mikrokontroler. Jalur ini biasanya digunakan untuk menambahkan tombol *reset* pada *shield* yang menghalangi papan utama Arduino.

2.4.4 *Integrated Development Environment (IDE) Arduino*

Arduino pada umumnya bekerja menggunakan pemrograman dengan bahasa C dan java yang dituliskan pada *software* Arduino IDE. IDE Arduino adalah *software* yang sangat canggih ditulis dengan menggunakan Java. IDE Arduino terdiri dari^[13]:

- Editor program, sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *Processing*.
- Compiler, sebuah modul yang mengubah kode program (bahasa *Processing*) menjadi kode biner. Bagaimanapun sebuah microcontroller tidak akan bisa memahami bahasa *Processing*. Yang bisa dipahami oleh mikrokontroler adalah kode biner. Itulah sebabnya *compiler* diperlukan dalam hal ini.
- *Uploader*, sebuah modul yang memuat kode biner dari computer ke dalam memori di dalam papan Arduino.

The image shows a screenshot of the Arduino IDE interface. The window title is "Blink : Arduino 0021". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for running, stopping, and other functions. The main text area contains the following code:

```
/**
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000);           // wait for a second
}
```

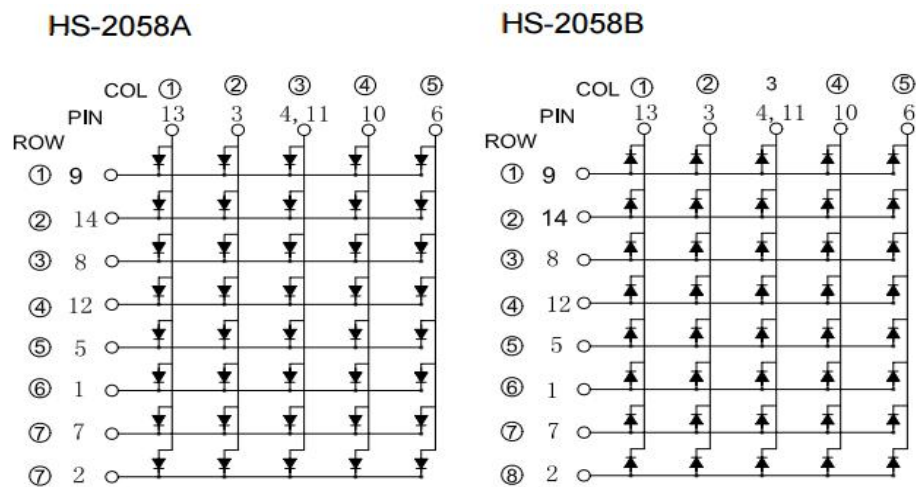
At the bottom of the window, a status bar indicates "Binary sketch size: 1010 bytes (of a 32256 byte maximum)".

Gambar 2.5 Arduino IDE^[14]

2.5 Dot matrix

Dot Matrix Display merupakan sebuah display *biner* dengan resolusi yang sangat kecil. Penggunaan dari *Dot Matrix Display* ini kebanyakan adalah media periklanan seperti untuk iklan layanan masyarakat, iklan objek wisata, dan media kampanye.

Dot Matrix Display memiliki resolusi 8X20, 8X32, 8X40, 7X32, 16X32, dan 16X64 dengan ukuran pixel 1 cm X 1 cm atau kurang. *Dot Matrix Display* biasanya menggunakan LED tipe *outdoor* agar *display* dapat terlihat jelas pada siang hari. Terdapat dua jenis susunan LED pada *Dot Matrix Display* dapat dilihat pada gambar 2.6.



Gambar 2.6 Susunan LED pada *Dot Matrix Display* HS-2058A/B^[17]

Berdasarkan pengawatannya antar LED, LED dibagi menjadi dua kategori, yaitu *common anode* dan *common cathode*^[16].

- Pada *common anode*, anode – anode LED saling dihubungkan. Untuk menyalakan LED, diperlukan tegangan level *low* pada kolom dan *high* pada baris.
- Pada *common cathode*, katode – katode LED saling dihubungkan. Untuk menyalakan LED, diperlukan tegangan level *high* pada kolom dan *low* pada baris.

Tujuan pengawatan bersama adalah untuk meminimalkan jumlah pin yang diperlukan.

Susunan LED pada *Dot Matrix Display* HS-2058A disebut *Dot Matrix Display common cathode*, sedangkan susunan LED pada *Dot Matrix Display* HS-2058B disebut *Dot Matrix Display common anode*. Dalam pengoperasiannya, data biner yang akan ditampilkan akan dikirim secara bergantian per kolom *dot matrix*. Saat data kolom 1 dikirim, maka data biner yang akan dikirim adalah 8 bit untuk baris-baris pada kolom 1. Pada saat data biner tersebut dikirim (untuk *Dot Matrix Display common anode*) pin kolom 1 diberi logika 0, sehingga LED pada kolom 1 akan menyala sesuai bentuk karakter yang diinginkan.

0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	1	1	1	1
1	0	0	0	1
1	0	0	0	1

Gambar 2.7 Susunan data biner untuk menampilkan karakter A^[18]

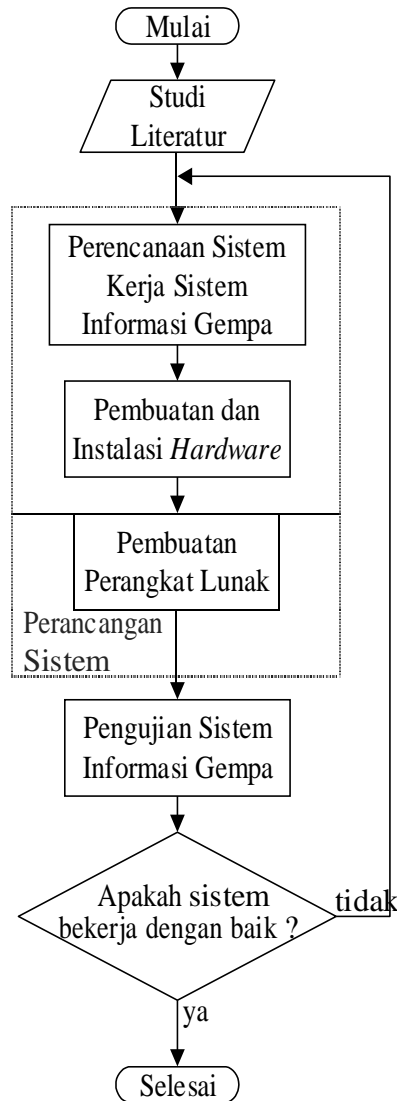
Sebagai contoh pada karakter A, saat data kolom 1 dikirim maka akan dikirim data 00111111 pada pin baris 1-8 dan pin kolom 1 diberi logika 0, dengan demikian LED pada kolom tersebut akan aktif. Kemudian data kolom 2 dikirim, pin kolom 2 diberi logika 0, kemudian dikirim data 01000100 pada pin baris 1-8, sehingga LED pada kolom 2 akan menyala. Pergantian hidup mati dari LED *Dot Matrix Display* yang sangat cepat menyebabkan mata tidak mampu mendeteksi perubahannya, sehingga mata hanya melihat seluruh LED tersebut hidup secara bersamaan.

Prinsip kerja *dot matrix* adalah menggunakan sistem *scanning* kolom dimana pada satu waktu dari sekian banyak kolom hanya satu kolom yang menyala^[19]. Karena dalam proses pengulangan penyalaan kolom dari kolom 1 sampai kolom terakhir begitu cepat dan berulang – ulang maka huruf yang ditampilkan tampak menyala bersamaan. Tetapi apabila *scanning* kolom diperlambat maka pergeseran penyalaan kolom akan terlihat satu persatu.

BAB III

METODOLOGI PENELITIAN

Penelitian yang dilakukan pada penelitian ini memiliki 3 tahapan, yaitu studi literatur, perancangan sistem, dan pengujian sistem. Adapun tahapan tersebut dapat digambarkan pada gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

3.1 Studi Literatur

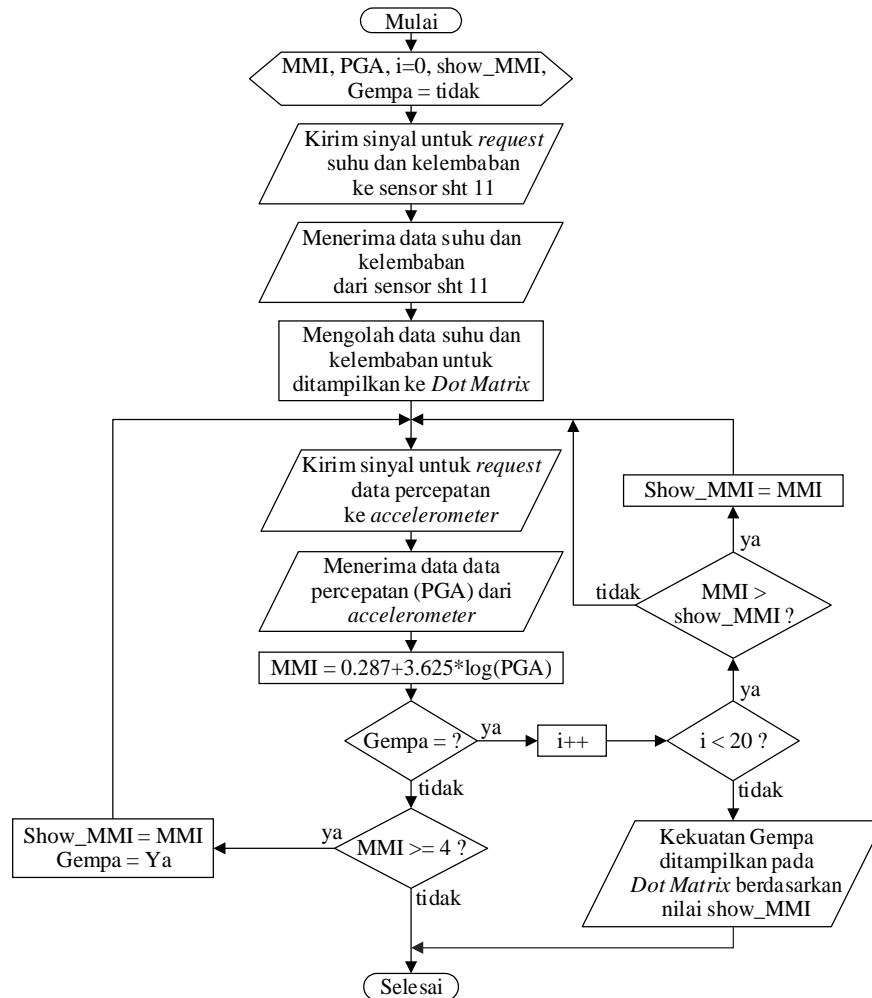
Pada tahapan ini dilakukan pengkajian literatur yang telah ada yang berhubungan dengan penelitian yang dilakukan. Pada tahap ini akan dipelajari mengenai teori – teori yang ada atau yang telah dilakukan oleh peneliti sebelumnya yang berhubungan dengan rancangan sistem.

3.2 Perancangan Sistem

Pada penelitian ini, proses pembuatan rancangan Sistem Informasi Gempa dibagi menjadi 3 tahapan, yaitu

a. Perencanaan Sistem Kerja Sistem Informasi Kerja

Pada tahapan ini dilakukan perencanaan cara kerja Sistem Informasi Gempa. Adapun cara kerja dari Sistem informasi digambarkan pada gambar 3.2.

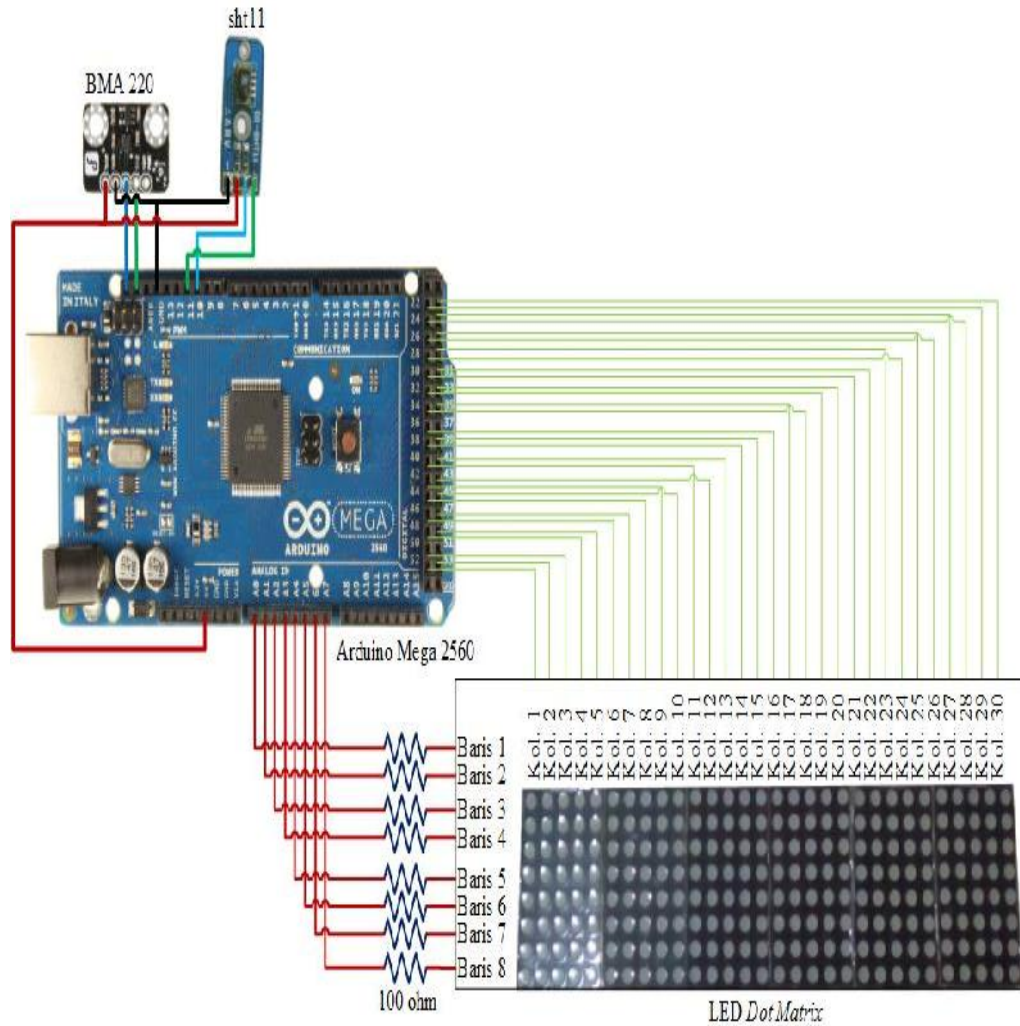


Gambar 3.2 Diagram Alir Cara Kerja Sistem Informasi Gempa

b. Pembuatan dan Instalasi *Hardware*

Pada tahapan ini dilakukan pembuatan *hardware socket LED Dot Matrix* dan mengkoneksikan (*instalasi*) semua *hardware* ke Arduino Mega 2560. *Hardware* yang digunakan pada Sistem Informasi Gempa ini adalah 1 unit Arduino Mega 2560, 6 unit 5X8 *common anode LED Dot Matrix* (HS-2058A), 1 unit sht11 *humidity and temperature sensor*, dan 1 unit *accelerometer* BMA220. Adapun

konfigurasi pin dari masing-masing *hardware* ke Arduino Mega 2560 dapat dilihat pada gambar 3.3 dan tabel 3.1.



Gambar 3.3 Konfigurasi Pin pada Sistem Informasi Gempa

Tabel 3.1 Konfigurasi Pin pada Sistem Informasi Gempa

No.	Hardware	Koneksi dengan Arduino
1.	sht11	<ul style="list-style-type: none"> • Pin Data sensor dihubungkan pada pin 10 Arduino • Pin Clock sensor dihubungkan pada pin 11 Arduino • Pin VCC sensor dihubungkan pada 5V Arduino • Pin Ground sensor dihubungkan pada GND Arduino
2.	BMA 220	<ul style="list-style-type: none"> • Pin SCL dihubungkan pada SCL Arduino • Pin SDA dihubungkan pada SDA Arduino • Pin VCC sensor dihubungkan pada 5V Arduino • Pin Ground sensor dihubungkan pada GND Arduino
3.	LED <i>Dot Matrix</i>	<ul style="list-style-type: none"> • Pin baris 1 dihubungkan pada pin A0 Arduino • Pin baris 2 dihubungkan pada pin A1 Arduino • Pin baris 3 dihubungkan pada pin A2 Arduino • Pin baris 4 dihubungkan pada pin A3 Arduino • Pin baris 5 dihubungkan pada pin A4 Arduino • Pin baris 6 dihubungkan pada pin A5 Arduino • Pin baris 7 dihubungkan pada pin A6 Arduino • Pin baris 8 dihubungkan pada pin A7 Arduino • Pin kolom 1 dihubungkan pada pin 52 Arduino • Pin kolom 2 dihubungkan pada pin 53 Arduino • Pin kolom 3 dihubungkan pada pin 50 Arduino • Pin kolom 4 dihubungkan pada pin 48 Arduino • Pin kolom 5 dihubungkan pada pin 49 Arduino • Pin kolom 6 dihubungkan pada pin 46 Arduino • Pin kolom 7 dihubungkan pada pin 47 Arduino • Pin kolom 8 dihubungkan pada pin 44 Arduino • Pin kolom 9 dihubungkan pada pin 42 Arduino • Pin kolom 10 dihubungkan pada pin45 Arduino • Pin kolom 11 dihubungkan pada pin 40 Arduino • Pin kolom 12 dihubungkan pada pin 43 Arduino • Pin kolom 13 dihubungkan pada pin 41 Arduino • Pin kolom 14 dihubungkan pada pin 38 Arduino • Pin kolom 15 dihubungkan pada pin 39 Arduino • Pin kolom 16 dihubungkan pada pin 36 Arduino • Pin kolom 17 dihubungkan pada pin 35 Arduino • Pin kolom 18 dihubungkan pada pin 34 Arduino • Pin kolom 19 dihubungkan pada pin 32 Arduino • Pin kolom 20 dihubungkan pada pin 33 Arduino • Pin kolom 21 dihubungkan pada pin 30 Arduino • Pin kolom 22 dihubungkan pada pin 31 Arduino • Pin kolom 23 dihubungkan pada pin 29 Arduino • Pin kolom 24 dihubungkan pada pin 28 Arduino • Pin kolom 25 dihubungkan pada pin 27 Arduino

No.	Hardware	Koneksi dengan Arduino
		<ul style="list-style-type: none"> • Pin kolom 26 dihubungkan pada pin 26 Arduino • Pin kolom 27 dihubungkan pada pin 25 Arduino • Pin kolom 28 dihubungkan pada pin 24 Arduino • Pin kolom 29 dihubungkan pada pin 22 Arduino • Pin kolom 30 dihubungkan pada pin 23 Arduino

c. Pembuatan Perangkat Lunak

Perangkat Lunak dibuat menggunakan *software* Arduino. Adapun perangkat lunak yang akan dibuat mencakup pembuatan perangkat lunak untuk LED *Dot Matrix* dan menggabungkannya dengan perangkat lunak untuk sht11 dan BMA 220 yang telah disediakan *developer hardware* tersebut. Penggabungan program dilakukan sesuai dengan diagram alir cara kerja Sistem Informasi Gempa pada gambar 2.2.

3.3. Pengujian Sistem Informasi Gempa

Pengujian Sistem Informasi Gempa dilakukan menggunakan Simulator Gempa. Simulator gempa dibuat menggunakan papan yang terdiri atas dua bagian, yaitu bagian atas dan bagian bawah yang dihubungkan dengan pegas. Bagian atas simulator gempa terdiri atas dua buah papan yang terpisah guna mendapatkan efek pergeseran lempeng pada gempa tiruan saat simulator gempa diberikan goncangan atau getaran. Simulator gempa dapat dilihat pada gambar 3.4.



Gambar 3.4 Simulator Gempa

Terdapat dua poin yang akan di uji dalam pengujian Sistem Informasi Gempa. Sistem Informasi Gempa dikatakan dapat bekerja dengan baik apabila kedua poin tersebut dipenuhi. Poin-poin tersebut adalah :

1. Sistem mampu menampilkan informasi suhu dan kelembaban saat tidak terjadi gempa.
2. Sistem mampu menampilkan informasi kekuatan gempa saat terjadi gempa.

BAB IV

HASIL DAN PEMBAHASAN

Informasi gempa, suhu dan kelembaban pada tugas akhir ini ditampilkan pada *dot matrix*, dimana pengukuran gempa dilakukan menggunakan sensor *accelerometer* BMA220. Sensor *accelerometer* akan mengukur kekuatan gempa yang berikan simulator yaitu berupa getaran. Kemudian getaran yang terukur sensor *accelerometer* akan ditampilkan pada *dot matrix*. Sedangkan pembacaan suhu dan kelembaban menggunakan sensor SHT11. Suhu dan kelembaban akan terbaca ketika simulator tidak diberikan getaran.

4.1 Hasil Pengujian Sistem

4.1.1 Sensor Accelerometer BMA220

Sensor *accelerometer* akan membaca nilai akselerasi pada saat simulasi yang dilakukan, nilai akselerasi ini yang menjadi acuan untuk perhitungan nilai skala MMI. Data yang terukur oleh sensor akan dihitung menggunakan rumus pada persamaan 2.1 halaman 11, sehingga akan di dapatkan data yang sesuai dengan skala MMI. Nilai skala MMI akan ditampilkan pada serial monitor, sedangkan informasi gempanya akan di tampilkan pada *dot matrix*. Tabel 4.1 merupakan hubungan nilai MMI dan informasi yang ditampilkan pada dot matrix berdasarkan hasil percobaan.

Tabel 4.1 Data Hasil Percobaan Hubungan MMI dan tampilan LED Dot Matrix

No.	Nilai MMI pada serial Monitor	Tampilan pada dot matrix pada dot Matrix
1	Tidak Ditampilkan	Menampilkan suhu dan kelembaban
2	Tidak Ditampilkan	Menampilkan suhu dan kelembaban
3	Tidak Ditampilkan	Menampilkan suhu dan kelembaban
4	4	Gempa Kecil
5	5	Gempa Kecil
6	6	Gempa Sedang
7	7	Gempa Sedang
8	9	Gempa Berat
9	9	Gempa Berat
10	11	Gempa Dasyat
11	14	Gempa Dasyat
12	14	Gempa Dasyat

4.1.2 Sensor SHT11

Sensor SHT11 digunakan untuk mengukur suhu dan kelembaban. Sensor ini akan membaca nilai suhu dan kelembapan ketika nilai yang terbaca pada skala MMI adalah skala I, II dan III dan saat tidak ada gempa. Berdasarkan spesifikasi SHT11 pada 2.3 suhu yang dapat diukur yaitu dari -40C hingga +123,8C, dan kelembaban relatif dari 0%RH hingga 100%RH, tetapi sistem ini mampu menampilkan suhu dan kelembaban dari 0°C hingga 99⁰C dan kelembaban dari 0% hingga 99% dalam bentuk

bilangan bulat karena tampilan *dot matrix* dirancang untuk menampilkan 2 digit bilangan positif yang tidak berkoma. Pengukuran suhu dan kelembaban dapat dilihat pada tabel berikut :

Tabel 4.2 Hasil Pengukuran Nilai Suhu dan Kelembaban

Waktu	Tempat	Suhu	Kelembaban
Pukul 12:25, 26/06/2015	Ruangan TA LEI	31 ⁰ C	69%
Pukul 10:52 01/07/2015	Ruangan TA LEI	30 ⁰ C	59%

4.1.3 Tampilan pada *Dot Matrix*

Hasil dari pengukuran nilai skala *intensitas marcalli* (MMI) oleh *accelerometer* BMA220 yang sudah dikelompokkan akan di tampilkan pada *dot matrix*, ketika tidak terjadi gempa dan skala MMI adalah skala I, II dan III maka informasi yang akan ditampilkan pada *dot matrix* adalah informasi suhu dan kelembaban. Tampilan informasi gempa, suhu dan kelembaban dapat dilihat pada gambar 4.1 sampai 4.3.



Gambar 4.1 Tampilan Gempa pada *Dot Matrix*



Gambar 4.2 Tampilan Suhu pada *Dot Matrix*



Gambar 4.3 Tampilan Kelembaban pada *Dot Matrix*

4.2 Analisa Pengujian Sistem

4.2.1 Analisa Pengujian Sensor *Accelerometer* BMA220

Sensor *accelerometer* digunakan sebagai masukan untuk perhitungan kekuatan gempa skala MMI. Nilai Skala MMI yang dikelompokkan akan di tampilkan

pada *dot matrix*. Tampilan *dot matrix* tersebut yang menjadi informasi gempa yang akan diterima atau dibaca masyarakat. Dengan adanya pengelompokan skala MMI, informasi gempa lebih mudah dipahami, sehingga lebih efisien untuk melakukan tindakan selanjutnya jika terjadi gempa yang kuat. Adapun pengelompokan skala MMI tersebut dapat dilihat pada tabel 4.3

Tabel 4.3 Pengelompokan Kekuatan Gempa

MMI	Keterangan	Kelompok Kekuatan Gempa
I	Tidak terasa	Di anggap tidak terjadi gempa
II	Terasa oleh orang yang berada di bangunan tinggi	
III	Getaran dirasakan seperti ada kereta yang berat melintas.	
IV	Getaran dirasakan seperti ada benda berat yang menabrak dinding rumah, benda tergantung bergoyang.	Gempa Kecil
V	Dapat dirasakan di luar rumah, hiasan dinding bergerak, benda kecil di atas rak mampu jatuh.	
VI	Terasa oleh hampir semua orang, dinding rumah rusak.	Gempa Sedang
VII	Dinding pagar yang tidak kuat pecah, orang tidak dapat berjalan/berdiri.	
VIII	Bangunan yang tidak kuat akan mengalami kerusakan.	Gempa Berat
IX	Bangunan yang tidak kuat akan mengalami kerusakan tekuk.	
X	Jembatan dan tangga rusak, terjadi tanah longsor. Rel kereta api bengkok.	Gempa Dasyat
XI	Rel kereta api rusak. Bendungan dan tanggul hancur. Seluruh bangunan hampir hancur dan terjadi longsor besar. Efek bencana yang lain	

MMI	Keterangan	Kelompok Kekuatan Gempa
	seperti tsunami, dan kebakaran.	
XII	Seluruh bangunan hancur lebur. Batu dan barang-barang terlempar ke udara. Tanah bergerak seperti gelombang. Kadang-kadang aliran sungai berubah. Pasir dan lumpur bergeser secara horizontal. Air dapat terlempar dari danau, sungai dan kanal. Diikuti dengan suara gemuruh yang besar. Biasanya bisa menyebabkan longsor besar, kebakaran, banjir, tsunami di daerah pantai, dan aktivitas gunung berapi. Pasir dan tanah halus terlihat meledak.	

Berdasarkan data hasil percobaan pada tabel 4.1, sistem telah mampu menampilkan informasi berdasarkan data kekuatan gempa berdasarkan pengukuran sensor *accelerometer* BMA 220. Berdasarkan data tersebut, dapat dilihat bahwa sistem yang dibuat mampu menampilkan kekuatan gempa berdasarkan pengukuran sensor *accelerometer* BMA 220. Namun, pada data ke 11 dan 12 besar skala MMI berdasarkan pengukuran sensor *accelerometer* BMA 220 mencapai nilai 14 dimana skala MMI hanya memiliki skala I hingga XII.

Hal ini karena sistem yang dibuat menggunakan perhitungan matematis dalam menentukan besar skala MMI. Getaran yang diberikan pada simulator gempa terlalu kuat dapat menyebabkan besar percepatan permukaan tanah / *Peak Ground Acceleration* (PGA) pada simulator permukaan juga sangat besar. Karena sistem yang dirancang menggunakan besar percepatan permukaan tanah / *Peak Ground Acceleration* (PGA) sebagai parameter dalam menghitung kekuatan gempa skala MMI, maka nilai MMI hasil perhitungan sistem dapat melebihi skala XII.

4.2.2 Analisa Pengujian Sensor SHT11

Sensor SHT11 akan bekerja ketika nilai skala *intensitas* MMI dibawah IV yakni pada saat simulator tidak di berikan getaran atau diberikan getaran kecil. Sehingga, saat tidak terjadi gempa maka masyarakat akan mengetahui informasi suhu dan kelembapan daerahnya. Informasi suhu dan kelembapan ini akan ditampilkan pada *dot matrix*.

4.2.3 Analisa Tampilan pada *Dot Matrix*

Informasi yang ditampilkan pada *dot matrix* mudah dimengerti, dengan informasi tersebut masyarakat akan mudah memahami situasi yang terjadi jika terjadi gempa.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan tugas akhir yang telah dilakukan dapat kita tarik beberapa kesimpulan berikut ini :

1. Sistem Informasi Gempa yang dirancang mampu menampilkan kekuatan gempa pada simulator gempa berdasarkan pengelompokan nilai skala *intensitas marcalli* (MMI) yaitu dari skala IV sampai skala XII .
2. Sistem Informasi Gempa yang dirancang mampu menampilkan informasi suhu dan kelembaban, suhu yang mampu dibaca 0°C hingga 99°C dan kelembaban dari 0 % hingga 99% .

5.2. Saran

Pada penelitian yang dilakukan, penulis menyarankan untuk adanya penelitian selanjutnya yaitu penyampaian informasi gempa yang diaplikasikan pada *handphone* android sehingga informasi gempa lebih mudah di dapatkan masyarakat.

DAFTAR PUSTAKA

- [1]. Tim Revisi Gempa Indonesia. 2010. *Peta Hazard Gempa Indonesia 2010 Sebagai Acuan Dasar Perencanaan dan Perancangan Infrastruktur Tahan Gempa*. Jakarta
- [2]. Badan Metereologi, Klimatologi dan geofisika. 2015. *Gempa Bumi*. (www.bmkg.go.id)
- [3]. Fulki, Ahmad 2011. *Analisis Parameter Gempa, b Value dan PGA di Daerah Papua*. Tugas Akhir: Universitas Islam Negeri Syarif Hidayatullah Jakarta.
- [4]. Sukanta, I Nyoman. Dkk. 2010. *Accelerograph BMKG dalam Penentuan Peta Intensitas Gempa Kuat*. Jakarta : Badan Mereorologi Klimatologi dan Geofisika.
- [5]. Rotlink. 2014. *Skala Mercalli*. (https://id.wikipedia.org/wiki/Skala_Mercalli)
- [6]. Wald, J. David, *et al.*, 1999. Relationships between Peak Ground Acceleration, Peak Ground Velocity, and Modified Mercalli Intensity in California. *Jurnal Earthquake Spectra, Vol 15 No. 3, 561-563, Agustus 1999*. California: Earthquake Spectra.

- [7]. Bilal, Mustafa. 2013. *Relationships Between Felt Intensity And Recorded Ground Motion Parameters For Turkey*. Tesis.: Middle East Technical University Turki. (<http://etd.lib.metu.edu.tr/upload/12615426/index.pdf>)
- [8]. Nalwan, Andi. 2013. *Teknik Rancang Bangun Robot Tingkat Dasar*. Yogyakarta : CV ANDI Offset,p.34, 38, 39, 45, 59.
- [9]. Alma'i, Vidi rahman. *Aplikasi Sensor Accelerometer pada pendeteksi Posisi*. Tugas Akhir. Universitas Diponegoro. (core.ac.uk/download/pdf/1172490.pdf)
- [10]. DFROBOT- 3- AXIS ACCELEROMETER BMA220 "Digitalmeans" (<http://digitalmeans.co.uk>)
- [11]. BOSCH Invented For Live. "Data Sheet Bma220 Digital Triaxial Acceleration Sensor . (<http://ae-bst.resource.bosch.com>)
- [12]. Gabriellalevine. "Visualize Humadity with the SHT11 Sensor".(m.instructable.com)
- [13]. Djuandi, Feri. 2011. *Pengenalan Arduino*. (www.tobuku.com).
- [14]. Agung.M. Bangun. 2014. *Arduino For Beginners*. (SuryaUniv-Arduino-Muhammad-Bangun-_Agung-202136575862733)

- [15]. ALLIED ELECTRONICS AN ELECTROCOMPONENTS COMPANY.
RADIO SPARES RADIONICS . *Arduino Mega 2560*.
(www.mantech.co.za).
- [16]. Kadir, Abdul. 2013. *Panduan Prktis Mempelajari Aplikasi Mikrokontroler dan Pemrograman Menggunakan Arduino*. Yogyakarta: CV ANDI.
- [17]. XINDALI ELECTRONICS(H.K.) CO., LIMITED, “Data Sheet HS-2058A/BX”. (<http://www.szxindali.com/>)
- [18]. Sekawan Servis. “Dasar – dasar Led Dot Matrix Layar”.(sekawan-servis-electronic.blogspot.com)
- [19]. Hamdani, Ari Zona. 2013 *Perancangan Sistem Update Informasi pada Papan Informasi Elektronik Menggunakan SMS Berbasis Mikrokontroler*. Tugas Akhir: Universitas Andalas.

LAMPIRAN A
LISTING PROGRAM SISTEM
INFORMASI GEMPA

SistemInfformasiGempa.ino

```
//Deklarasi Variabel untuk SHT
#include <SHT1x.h>
#define dataPin 10
#define clockPin 11
SHT1x sht1x(dataPin, clockPin);
float temp_c;
float temp_f;
float humidity;
int show_temp;
int show_humidity;
int suhu_satuan, suhu_puluhan, humidity_puluhan, humidity_satuan;

//Deklarasi Variabel untuk Accelerometer
#include <Wire.h>
byte Version[3];
int8_t x_data;
int8_t y_data;
int8_t z_data;
float x,y,z,PGA,R;

//Deklarasi Variabel untuk Gempa
int cek_gempa = 0;
float cek_MMI=0;
float MMI=0;
float show_MMI=0;
float m_PGA=0.833350281020331;

//Deklarasi Variabel untuk Dot Matriks
Int shift_speed=25; // Atur Kecepatan Dengan Merubah Nilai shift_speed
const int kolom[] =
{23,22,24,25,26,27,28,29,31,30,33,32,34,35,36,39,38,41,43,40,45,42,44,47,46,49,48,50,53,52};
byte show_sht[] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x61, 0x91, 0x91, 0x91, 0x8E, 0x00, //S
0xFE, 0x01, 0x01, 0x01, 0xFE, 0x00, //U
0xFF, 0x10, 0x10, 0x10, 0xFF, 0x00, //H
0xFE, 0x01, 0x01, 0x01, 0xFE, //U
0x00, 0x66, 0x66, 0x00, //:
0x43, 0x85, 0x89, 0x91, 0x61, 0x00, //angka puluhan suhu(awal = 2)
0xF2, 0x91, 0x91, 0x91, 0x8E, 0x00, //angka satuan suhu (awal = 5)
0xE0, 0xA0, 0xE0, 0x00, //derajat
0x7E, 0x81, 0x81, 0x81, 0x42, //C
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, //spasi
0xFF, 0x18, 0x24, 0x42, 0x81, 0x00, //K
0xFF, 0x91, 0x91, 0x91, 0x81, 0x00, //E
0xFF, 0x01, 0x01, 0x01, 0x01, 0x00, //L
0xFF, 0x91, 0x91, 0x91, 0x81, 0x00, //E
0xFF, 0x40, 0x30, 0x40, 0xFF, 0x00, //M
0xFF, 0x91, 0x91, 0x91, 0x6E, 0x00, //B
0x7F, 0x88, 0x88, 0x88, 0x7F, 0x00, //A
0xFF, 0x91, 0x91, 0x91, 0x6E, 0x00, //B
0x7F, 0x88, 0x88, 0x88, 0x7F, 0x00, //A
0xFF, 0x40, 0x30, 0x08, 0xFF, //N
0x00, 0x66, 0x66, 0x00, //:
0x6E, 0x91, 0x91, 0x91, 0x6E, 0x00, //angka puluhan kelembaban (awal = 8 )
0x7E, 0x89, 0x91, 0xA1, 0x7E, 0x00, //angka satuan kelembaban (awal = 0)
0xC3, 0xC6, 0x18, 0x63, 0xC3, //%
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };

int n_sht=134;

//Deklarasi Dot Matriks Gempa
//Gempa Kecil (IV - V Skala MMI)
byte gempa_kecil[] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x7E, 0x81, 0x89, 0x89, 0x4E, 0x00, //G
0xFF, 0x91, 0x91, 0x91, 0x81, 0x00, //E
0xFF, 0x40, 0x30, 0x40, 0xFF, 0x00, //M
0xFF, 0x88, 0x88, 0x88, 0x70, 0x00, //P
0x7F, 0x88, 0x88, 0x88, 0x7F, //A
0x00, 0x00, 0x00, 0x00, 0x00, //spasi
```

SistemInfformasiGempa.ino

```
0xFF, 0x18, 0x24, 0x42, 0x81, 0x00, //K
0xFF, 0x91, 0x91, 0x91, 0x81, 0x00, //E
0x7E, 0x81, 0x81, 0x81, 0x42, 0x00, //C
0x81, 0xFF, 0x81, 0x00, //I
0xFF, 0x01, 0x01, 0x01, 0x01, //L
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };
int n_gempa_kecil = 61;

//Gempa Sedang (VI - VII Skala MMI)
byte gempa_sedang[] = {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x7E, 0x81, 0x89, 0x89, 0x4E, 0x00, //G
0xFF, 0x91, 0x91, 0x91, 0x81, 0x00, //E
0xFF, 0x40, 0x30, 0x40, 0xFF, 0x00, //M
0xFF, 0x88, 0x88, 0x88, 0x70, 0x00, //P
0x7F, 0x88, 0x88, 0x88, 0x7F, //A
0x00, 0x00, 0x00, 0x00, 0x00, //spasi
0x61, 0x91, 0x91, 0x91, 0x8E, 0x00, //S
0xFF, 0x91, 0x91, 0x91, 0x81, 0x00, //E
0xFF, 0x81, 0x81, 0x42, 0x3C, 0x00, //D
0x7F, 0x88, 0x88, 0x88, 0x7F, 0x00, //A
0xFF, 0x40, 0x30, 0x08, 0xFF, 0x00, //N
0x7E, 0x81, 0x89, 0x89, 0x4E, //G
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };
int n_gempa_sedang = 69;

//Gempa Berat (VIII - IX Skala MMI)
byte gempa_berat[] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x7E, 0x81, 0x89, 0x89, 0x4E, 0x00, //G
0xFF, 0x91, 0x91, 0x91, 0x81, 0x00, //E
0xFF, 0x40, 0x30, 0x40, 0xFF, 0x00, //M
0xFF, 0x88, 0x88, 0x88, 0x70, 0x00, //P
0x7F, 0x88, 0x88, 0x88, 0x7F, //A
0x00, 0x00, 0x00, 0x00, 0x00, //spasi
0xFF, 0x91, 0x91, 0x91, 0x6E, 0x00, //B
0xFF, 0x91, 0x91, 0x91, 0x81, 0x00, //E
0xFF, 0x88, 0x8C, 0x8A, 0x71, 0x00, //R
0x7F, 0x88, 0x88, 0x88, 0x7F, 0x00, //A
0x80, 0x80, 0xFF, 0x80, 0x80, //T
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };
int n_gempa_berat = 63;

//Gempa Dasyat (X - XII Skala MMI)
byte gempa_dasyat[] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x7E, 0x81, 0x89, 0x89, 0x4E, 0x00, //G
0xFF, 0x91, 0x91, 0x91, 0x81, 0x00, //E
0xFF, 0x40, 0x30, 0x40, 0xFF, 0x00, //M
0xFF, 0x88, 0x88, 0x88, 0x70, 0x00, //P
0x7F, 0x88, 0x88, 0x88, 0x7F, //A
0x00, 0x00, 0x00, 0x00, 0x00, //spasi
0xFF, 0x81, 0x81, 0x42, 0x3C, 0x00, //D
0x7F, 0x88, 0x88, 0x88, 0x7F, 0x00, //A
0x61, 0x91, 0x91, 0x91, 0x8E, 0x00, //S
0xE0, 0x10, 0x0F, 0x10, 0xE0, 0x00, //Y
0x7F, 0x88, 0x88, 0x88, 0x7F, 0x00, //A
0x80, 0x80, 0xFF, 0x80, 0x80, //T
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };
int n_gempa_dasyat = 69;

//Fungsi SHT
void run_sht()
```



```

{

temp_c = sht1x.readTemperatureC();
humidity = sht1x.readHumidity();

show_temp = (int)temp_c;
show_humidity = (int)humidity;

suhu_satuan = show_temp%10;
suhu_puluhan = (int)((show_temp%100)/10);
humidity_satuan = show_humidity%10;
humidity_puluhan = (int)((show_humidity%100)/10);

switch(suhu_puluhan){
case 0:
show_sht[57]=0x00; show_sht[58]=0x00; show_sht[59]=0x00;
show_sht[60]=0x00; show_sht[61]=0x00; break;
case 1:
show_sht[57]=0x00; show_sht[58]=0x41; show_sht[59]=0xFF;
show_sht[60]=0x01; show_sht[61]=0x00; break;
case 2:
show_sht[57]=0x43; show_sht[58]=0x85; show_sht[59]=0x89;
show_sht[60]=0x91; show_sht[61]=0x61; break;
case 3:
show_sht[57]=0x42; show_sht[58]=0x81; show_sht[59]=0x91;
show_sht[60]=0x91; show_sht[61]=0x6E; break;
case 4:
show_sht[57]=0x18; show_sht[58]=0x28; show_sht[59]=0x48;
show_sht[60]=0xFF; show_sht[61]=0x08; break;
case 5:
show_sht[57]=0xF2; show_sht[58]=0x91; show_sht[59]=0x91;
show_sht[60]=0x91; show_sht[61]=0x8E; break;
case 6:
show_sht[57]=0x1E; show_sht[58]=0x29; show_sht[59]=0x49;
show_sht[60]=0x89; show_sht[61]=0x86; break;
case 7:
show_sht[57]=0x80; show_sht[58]=0x8F; show_sht[59]=0x90;
show_sht[60]=0xA0; show_sht[61]=0xC0; break;
case 8:
show_sht[57]=0x6E; show_sht[58]=0x91; show_sht[59]=0x91;
show_sht[60]=0x91; show_sht[61]=0x6E; break;
case 9:
show_sht[57]=0x70; show_sht[58]=0x89; show_sht[59]=0x89;
show_sht[60]=0x8A; show_sht[61]=0x7C; break;
default:
show_sht[57]=0xFF; show_sht[58]=0xFF; show_sht[59]=0xFF;
show_sht[60]=0xFF; show_sht[61]=0xFF; break;
}

switch(suhu_satuan){
case 0:
show_sht[63]=0x7E; show_sht[64]=0x89; show_sht[65]=0x91;
show_sht[66]=0xA1; show_sht[67]=0x7E; break;
case 1:
show_sht[63]=0x00; show_sht[64]=0x41; show_sht[65]=0xFF;
show_sht[66]=0x01; show_sht[67]=0x00; break;
case 2:
show_sht[63]=0x43; show_sht[64]=0x85; show_sht[65]=0x89;
show_sht[66]=0x91; show_sht[67]=0x61; break;
case 3:
show_sht[63]=0x42; show_sht[64]=0x81; show_sht[65]=0x91;
show_sht[66]=0x91; show_sht[67]=0x6E; break;
case 4:
show_sht[63]=0x18; show_sht[64]=0x28; show_sht[65]=0x48;
show_sht[66]=0xFF; show_sht[67]=0x08; break;
case 5:
show_sht[63]=0xF2; show_sht[64]=0x91; show_sht[65]=0x91;
show_sht[66]=0x91; show_sht[67]=0x8E; break;
case 6:
show_sht[63]=0x1E; show_sht[64]=0x29; show_sht[65]=0x49;
show_sht[66]=0x89; show_sht[67]=0x86; break;
case 7:
show_sht[63]=0x80; show_sht[64]=0x8F; show_sht[65]=0x90;
show_sht[66]=0xA0; show_sht[67]=0xC0; break;
}
}

```

SistemInformasiGempa.ino

```
case 8:
show_sht[63]=0x6E; show_sht[64]=0x91; show_sht[65]=0x91;
show_sht[66]=0x91; show_sht[67]=0x6E; break;
case 9:
show_sht[63]=0x70; show_sht[64]=0x89; show_sht[65]=0x89;
show_sht[66]=0x8A; show_sht[67]=0x7C; break;
default:
show_sht[63]=0xFF; show_sht[64]=0xFF; show_sht[65]=0xFF;
show_sht[66]=0xFF; show_sht[67]=0xFF; break;
}

switch(humidity_puluhan){
case 0:
show_sht[147]=0x00; show_sht[148]=0x00; show_sht[149]=0x00;
show_sht[150]=0x00; show_sht[151]=0x00; break;
case 1:
show_sht[147]=0x00; show_sht[148]=0x41; show_sht[149]=0xFF;
show_sht[150]=0x01; show_sht[151]=0x00; break;
case 2:
show_sht[147]=0x43; show_sht[148]=0x85; show_sht[149]=0x89;
show_sht[150]=0x91; show_sht[151]=0x61; break;
case 3:
show_sht[147]=0x42; show_sht[148]=0x81; show_sht[149]=0x91;
show_sht[150]=0x91; show_sht[151]=0x6E; break;
case 4:
show_sht[147]=0x18; show_sht[148]=0x28; show_sht[149]=0x48;
show_sht[150]=0xFF; show_sht[151]=0x08; break;
case 5:
show_sht[147]=0xF2; show_sht[148]=0x91; show_sht[149]=0x91;
show_sht[150]=0x91; show_sht[151]=0x8E; break;
case 6:
show_sht[147]=0x1E; show_sht[148]=0x29; show_sht[149]=0x49;
show_sht[150]=0x89; show_sht[151]=0x86; break;
case 7:
show_sht[147]=0x80; show_sht[148]=0x8F; show_sht[149]=0x90;
show_sht[150]=0xA0; show_sht[151]=0xC0; break;
case 8:
show_sht[147]=0x6E; show_sht[148]=0x91; show_sht[149]=0x91;
show_sht[150]=0x91; show_sht[151]=0x6E; break;
case 9:
show_sht[147]=0x70; show_sht[148]=0x89; show_sht[149]=0x89;
show_sht[150]=0x8A; show_sht[151]=0x7C; break;
default:
show_sht[147]=0xFF; show_sht[148]=0xFF; show_sht[149]=0xFF;
show_sht[150]=0xFF; show_sht[151]=0xFF; break;
}

switch(humidity_satuan){
case 0:
show_sht[153]=0x7E; show_sht[154]=0x89; show_sht[155]=0x91;
show_sht[156]=0xA1; show_sht[157]=0x7E; break;
case 1:
show_sht[153]=0x00; show_sht[154]=0x41; show_sht[155]=0xFF;
show_sht[156]=0x01; show_sht[157]=0x00; break;
case 2:
show_sht[153]=0x43; show_sht[154]=0x85; show_sht[155]=0x89;
show_sht[156]=0x91; show_sht[157]=0x61; break;
case 3:
show_sht[153]=0x42; show_sht[154]=0x81; show_sht[155]=0x91;
show_sht[156]=0x91; show_sht[157]=0x6E; break;
case 4:
show_sht[153]=0x18; show_sht[154]=0x28; show_sht[155]=0x48;
show_sht[156]=0xFF; show_sht[157]=0x08; break;
case 5:
show_sht[153]=0xF2; show_sht[154]=0x91; show_sht[155]=0x91;
show_sht[156]=0x91; show_sht[157]=0x8E; break;
case 6:
show_sht[153]=0x1E; show_sht[154]=0x29; show_sht[155]=0x49;
show_sht[156]=0x89; show_sht[157]=0x86; break;
case 7:
show_sht[153]=0x80; show_sht[154]=0x8F; show_sht[155]=0x90;
show_sht[156]=0xA0; show_sht[157]=0xC0; break;
case 8:
show_sht[153]=0x6E; show_sht[154]=0x91; show_sht[155]=0x91;
```

SistemInformasiGempa.ino

```
show_sht[156]=0x91; show_sht[157]=0x6E; break;
case 9:
show_sht[153]=0x70; show_sht[154]=0x89; show_sht[155]=0x89;
show_sht[156]=0x8A; show_sht[157]=0x7C; break;
default:
show_sht[153]=0xFF; show_sht[154]=0xFF; show_sht[155]=0xFF;
show_sht[156]=0xFF; show_sht[157]=0xFF; break;
}
}

//Fungsi Accelerometer
void AccelerometerInit()
{
  Wire.beginTransmission(0x0A); // address of the accelerometer
  // reset the accelerometer
  Wire.write(0x04); // Y data
  Wire.endTransmission();
  Wire.requestFrom(0x0A,1); // request 6 bytes from slave device #2
  while(Wire.available()) // slave may send less than requested
  {
    Version[0] = Wire.read(); // receive a byte as characte
  }
  x_data=(int8_t)Version[0]>>2;

  Wire.beginTransmission(0x0A); // address of the accelerometer
  // reset the accelerometer
  Wire.write(0x06); // Y data
  Wire.endTransmission();
  Wire.requestFrom(0x0A,1); // request 6 bytes from slave device #2
  while(Wire.available()) // slave may send less than requested
  {
    Version[1] = Wire.read(); // receive a byte as characte
  }
  y_data=(int8_t)Version[1]>>2;

  Wire.beginTransmission(0x0A); // address of the accelerometer
  // reset the accelerometer
  Wire.write(0x08); // Y data
  Wire.endTransmission();
  Wire.requestFrom(0x0A,1); // request 6 bytes from slave device #2
  while(Wire.available()) // slave may send less than requested
  {
    Version[2] = Wire.read(); // receive a byte as characte
  }
  z_data=(int8_t)Version[2]>>2;

  x = (float)x_data*125*9.8/1000;
  y = (float)y_data*125*9.8/1000;
  z = (float)z_data*125*9.8/1000;
  R = sqrt((x*x)+(y*y)+(z*z));
  PGA = abs(R-9.8);
}

//Fungsi Gempa
void gempa(){
  m_PGA=PGA;
  PGA=0;
  for (int index=0;index<20;index++)
  {
    AccelerometerInit();
    delay(100);
    if (PGA>m_PGA){
    m_PGA=PGA;}
    PGA = 0;
  }
  MMI=0.287+3.625*log(m_PGA);
  m_PGA= 0.833350281020331;

  if (MMI<4){
  Serial.print("MMI sekarang = ");Serial.println(MMI);
  }
}
```


SistemInfformasiGempa.ino

```
for(int k=1;k<31+n_gempa_sedang;k++){
for(int i=0;i<=shift_speed;i++){ //Atur Kecepatan Dengar Merubah Nilai i
for(int j=0;j<30;j++){
digitalWrite(kolom[j],LOW);
PORTF = gempa_sedang[j+k];
//delay(1);
delayMicroseconds(500);
digitalWrite(kolom[j],HIGH);

//Mengecek Gempa Setiap 250 mili detik
cek_gempa++;
if (cek_gempa>=500){
cek_gempa=0;
AccelerometerInit();
if(PGA>= 0.833350281020331){
cek_MMI=0.287+3.625*log(PGA);
if (cek_MMI>MMI){
gempa();
}
}
}

}
}
}
}
MMI = 0;
}
else if(MMI>=4){
show_MMI=MMI;
Serial.print("MMI pada LED Dot Matrix = ");Serial.println(show_MMI);
show_MMI=0;
for(int kali=0;kali<1000;kali++){
for(int k=1;k<31+n_gempa_kecil;k++){
for(int i=0;i<=shift_speed;i++){ //Atur Kecepatan Dengar Merubah Nilai i
for(int j=0;j<30;j++){
digitalWrite(kolom[j],LOW);
PORTF = gempa_kecil[j+k];
//delay(1);
delayMicroseconds(500);
digitalWrite(kolom[j],HIGH);

//Mengecek Gempa Setiap 250 mili detik
cek_gempa++;
if (cek_gempa>=500){
cek_gempa=0;
AccelerometerInit();
if(PGA>= 0.833350281020331){
cek_MMI=0.287+3.625*log(PGA);
if (cek_MMI>MMI){
gempa();
}
}
}

}
}
}
}
}
MMI = 0;
}
}

void setup() {
//inisialisasi komunikasi serial
Serial.begin(9600);

//Inisialisasi LED Dot Matriks
for (int i = 0;i<30;i++){
pinMode(kolom[i],OUTPUT);
digitalWrite(kolom[i],HIGH);
}
}
```

SistemInfformasiGempa.ino

```
DDRF = B11111111;

// Inisialisasi Accelerometer BMA220
Wire.begin();
Wire.beginTransmission(0x0A); // address of the accelerometer

// range settings
Wire.write(0x22); //register address
Wire.write(0x01); //can be set at"0x00""0x01""0x02""0x03", refer to Datashhet on wiki

// low pass filter, range settings
Wire.write(0x20);
Wire.write(0x02);
Wire.endTransmission();

}

void loop() {

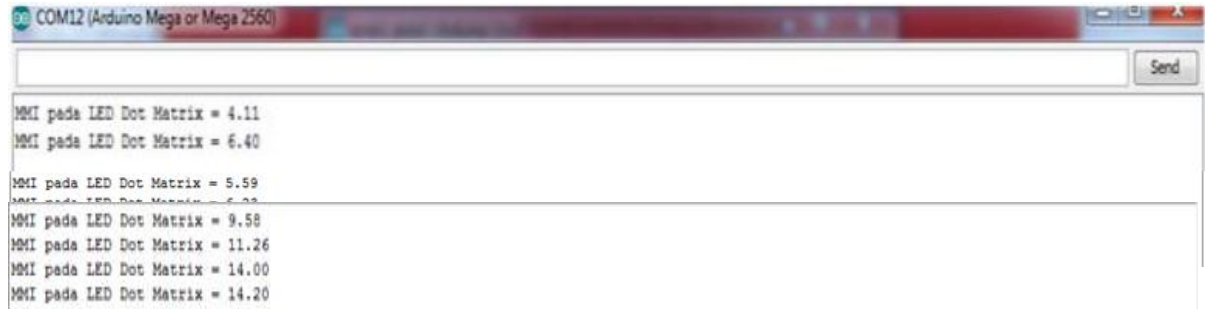
  //Menjalankan Fungsi SHT
  run_sht();

  //Menampilkan suhu dan humidity ke Dot Matriks
  for(int kali=0;kali<3;kali++){
    for(int k=1;k<31+n_sht;k++){
      for(int i=0;i<=25;i++){ //Atur Kecepatan Dengar Merubah Nilai i
        for(int j=0;j<30;j++){
          digitalWrite(kolom[j],LOW);
          PORTF = show_sht[j+k];
          //delay(1);
          delayMicroseconds(500);
          digitalWrite(kolom[j],HIGH);

          //Mengecek Gempa Setiap 250 mili detik
          cek_gempa++;
          if (cek_gempa>=500){
            cek_gempa=0;
            AccelerometerInit();
            if(PGA>= 0.833350281020331){
              cek_MMI=0.287+3.625*log(PGA);
              if (cek_MMI>=4){
                gempa();
              }
            }
          }
        }
      }
    }
  }
}
```

LAMPIRAN B
NILAI MMI YANG DITAMPILKAN PADA SERIAL
MONITOR

Nilai MMI Gempa yang Ditampilkan pada Serial Monitor



The screenshot shows a serial monitor window titled "COM12 (Arduino Mega or Mega 2560)". The window contains a text area with the following text:

```
MMI pada LED Dot Matrix = 4.11  
MMI pada LED Dot Matrix = 6.40  
MMI pada LED Dot Matrix = 5.59  
MMI pada LED Dot Matrix = 6.20  
MMI pada LED Dot Matrix = 9.58  
MMI pada LED Dot Matrix = 11.26  
MMI pada LED Dot Matrix = 14.00  
MMI pada LED Dot Matrix = 14.20
```

A "Send" button is visible in the top right corner of the window.