

**RANCANG BANGUN PAPAN INFORMASI BERBASIS *DOT*
MATRIX DI BAAKPK POLMAN BABEL YANG
TERINTEGRASI DENGAN ANDROID**

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan
Diploma III Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh :

DINA FORSA ANGGRAINI NIRM : 0031536

RESTU PANGESTU NIRM : 0031557

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG**

2018

LEMBAR PENGESAHAN

**RANCANG BANGUN PAPAN INFORMASI BERBASIS *DOT*
MATRIX DI BAAKPK POLMAN BABEL YANG
TERINTEGRASI DENGAN ANDROID**

Oleh :

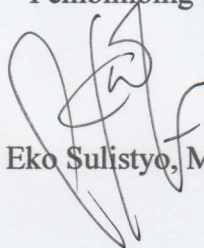
Dina Forsa Anggraini / 003 15 36

Restu Pangestu / 003 15 57

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Diploma III Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,

Pembimbing 1




Eko Sulistyono, M.T

Pembimbing 2



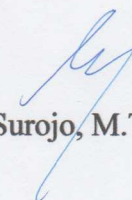
Aan Febriansyah, M.T

Penguji 1




Indra Dwisaputra, M.T

Penguji 2



Surojo, M.T

Penguji 3



Dr. Parulian Silalahi, M.Pd

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini :

Nama Mahasiswa 1 : Dina Forsa Anggraini NIRM : 0031536

Nama Mahasiswa 2 : Restu Pangestu NIRM : 0031557

Dengan Judul : RANCANG BANGUN PAPAN INFORMASI BERBASIS *DOT MATRIX* DI BAAKPK POLMAN BABEL YANG TERINTEGRASI DENGAN ANDROID

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia diberikan sanksi yang berlaku.

Sungailiat,

Nama Mahasiswa

Tanda Tangan

1. Dina Forsa Anggraini



2. Restu Pangestu



ABSTRAK

Tugas Akhir (PA) Rancang Bangun Papan Informasi Berbasis Dot Matrix Di BAAKPK POLMAN BABEL Yang Terintegrasi Dengan Android ini, menggunakan komunikasi Bluetooth dalam pengiriman data dari Smartphone ke Mikrokontroler, sedangkan pengiriman data dari Komputer menggunakan media kabel USB. Aplikasi Android pada Smartphone dibuat menggunakan software App Inventor, sedangkan untuk aplikasi Windows pada Komputer menggunakan software Microsoft Visual Basic. Mikrokontroler jenis Arduino Mega 2560 dan Arduino UNO digunakan sebagai media pengolah data. Tujuan PA ini adalah untuk memudahkan BAAKPK POLMAN BABEL dalam melakukan proses penggantian dan penyampaian informasi. Metodologi pada PA ini yaitu informasi ditulis pada aplikasi yang ada pada Android atau Windows, pengiriman data dari Android, dikirim melalui Bluetooth Transmitter yang terdapat pada Smartphone ke Bluetooth Receiver yang ada pada perangkat papan informasi. Sedangkan pengiriman data dari Windows menggunakan media kabel USB yang terpasang pada Port USB Komputer dan Port yang ada pada Mikrokontroler perangkat papan informasi. Hasil PA yang diperoleh antara lain papan informasi berbasis dot matrix menampilkan text informasi sesuai dengan text yang dituliskan pada aplikasi di Android atau Komputer dengan ukuran tinggi font yang dapat diubah-ubah.

Kata kunci : Dot Matrix, App Inventor, Microsoft Visual Basic 2010 Express, Bluetooth, Arduino, Smartphone, Android, Komputer, Windows

ABSTRACT

Final Project (PA) Design of Dot Matrix Based Information Board in BABEL POLMAN BAAKPK Integrated With this Android, using Bluetooth communication in sending data from Smartphone to Microcontroller, while sending data from Computer using USB cable media. Android applications on smartphones are made using App Inventor software, while for Windows applications on a Computer using Microsoft Visual Basic software. Arduino Mega 2560 and Arduino UNO microcontrollers are used as data processing media. The purpose of this PA is to facilitate BAAKPK POLMAN BABEL in carrying out the process of replacing and delivering information. The methodology in this PA is that information is written on applications that are on Android or Windows, sending data from Android, sent via Bluetooth Transmitter that is on the Smartphone to the Bluetooth Receiver on the information board device. While sending data from Windows using a USB cable media that is attached to the USB Computer Port and the Port on the Microcontroller information board device. PA results obtained include a dot matrix-based information board displaying text information in accordance with the text written on an application on Android or a computer with a font size that can be changed.

Keywords: Dot Matrix, App Inventor, Microsoft Visual Basic 2010 Express, Bluetooth, Arduino, Smartphone, Android, Computer, Windows

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT atas berkat rahmat dan hidayah-nya, sehingga penulis dapat menyelesaikan karya tulis proyek akhir ini dengan baik dan tepat pada waktunya.

Karya tulis Proyek Akhir ini disusun sebagai salah satu persyaratan dan kewajiban mahasiswa untuk menyelesaikan kurikulum program Diploma III di Politeknik Manufaktur Negeri Bangka Belitung.

Penulis mencoba untuk menerapkan ilmu pengetahuan yang telah ditetapkan selama 3 tahun mengecap pendidikan di Politeknik Manufaktur Negeri Bangka Belitung dan pengalaman yang penulis dapatkan selama melaksanakan Program Kerja Lapangan pada pembuatan alat dan makalah Proyek Akhir ini.

Pada kesempatan ini penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada orang-orang yang telah berperan sehingga dapat terselesaikannya Proyek Akhir, sebagai berikut :

1. Keluarga besar (Ayah, Ibu, Adik-adik penulis, Nenek, Kakek, dll) yang selalu senantiasa memberikan kasih sayang, doa, dukungan moril maupun materi dan semangat.
2. Bapak Sugeng A, M.Eng, Ph.D selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
3. Bapak Eko Sulistyono, M.T, selaku Kepala Prodi DIII Teknik Elektronika Politeknik Manufaktur Negeri Bangka Belitung sekaligus pembimbing 1 dalam Proyek Akhir ini.
4. Bapak Aan Febriansyah, M.T, selaku Kepala Jurusan Teknik Elektro dan Informatika Politeknik Manufaktur Negeri Bangka Belitung sekaligus pembimbing 2 dalam Proyek Akhir ini.
5. Bapak Dr. Parulian Silalahi, M.Pd, selaku wali kelas III EB.
6. Seluruh staf pengajar dan karyawan Politeknik Manufaktur Negeri Bangka Belitung.
7. Rekan-rekan mahasiswa Politeknik Manufaktur Negeri Bangka Belitung yang telah banyak membantu selama menyelesaikan Proyek Akhir.

8. Pihak-pihak lain yang telah memberikan bantuan secara langsung maupun tidak langsung dalam pembuatan Proyek Akhir ini yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa penulisan Proyek Akhir ini masih jauh dari sempurna dikarenakan penulis adalah manusia biasa yang tidak luput dari kesalahan. Karena yang benar hanya datang dari Allah SWT dan yang salah datang dari penulis sendiri. Oleh karena itu, sangat diharapkan segala petunjuk, kritik dan saran yang membangun dari pembaca agar dapat menunjang pengembangan dan perbaikan penulisan selanjutnya.

Besar harapan penulis semoga makalah tugas akhir dan alat yang dibuat dapat memberikan manfaat bagi pihak yang berkepentingan pada khususnya dan baik bagi perkembangan ilmu teknologi pada umumnya.

Sungailiat,

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN	ii
PERNYATAAN BUKAN PLAGIAT	iii
ABSTRAK.....	iv
<i>ABSTRACT</i>	v
KATA PENGANTAR	vi
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN.....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan dan Batasan Masalah	3
1.2.1 Rumusan Masalah.....	3
1.2.2 Batasan Masalah	3
1.3 Tujuan Proyek Akhir.....	3
BAB II DASAR TEORI	4
2.1 <i>Dot Matrix Display P10</i>	4
2.2 Android	5
2.3 Windows	6
2.4 App Inventor	7
2.5 Microsoft Visual Basic	9
2.6 Arduino Uno	10
2.7 Arduino Mega 2560	13
2.8 <i>Bluetooth</i>	15
2.9 RTC.....	17
2.10 <i>Power Supply</i>	18

BAB III METODE PELAKSANAAN	19
3.1 <i>Flow Chart</i> Perancangan dn Pembuatan Alat.....	19
3.2 Pengumpulan Data	20
3.2.1 Pengumpulan Data Secara Tidak Langsung	20
3.2.2 Pengumpulan Data Secara Langsung.....	20
3.3 Pengolahan Data	20
3.4 Perancangan <i>Hardware</i> dan <i>Software</i>	21
3.5 Pembuatan <i>Hardware</i> dan <i>Software</i>	21
3.6 Uji Coba	22
3.7 Perbaikan.....	22
3.8 Pembuatan Laporan	22
BAB IV PEMBAHASAN.....	23
4.1 Diagram Blok.....	23
4.2 <i>Hardware</i> Mekanik Papan Informasi Berbasis <i>Dot Matrix</i>	24
4.2.1 Perancangan <i>Hardware</i> Mekanik	24
4.2.2 Pembuatan <i>Hardware</i> Mekanik Papan Informasi Berbasis <i>Dot Matrix</i>	24
4.3 <i>Hardware</i> Elektrik Papan Informasi Berbasis <i>Dot Matrix</i>	26
4.3.1 Catu Daya.....	26
4.3.2 Arduino Mega 2560	28
4.3.3 Arduino UNO.....	29
4.3.4 Modul RTC DS1307.....	30
4.3.5 <i>Bluetooth</i> HC-05	33
4.3.6 <i>Dot Matrix Display</i> P10	36
4.4 <i>Software</i> Papan Informasi Berbasis <i>Dot Matrix</i>	41
4.4.1 Aplikasi SiPETRIX.....	41
4.4.2 Program Arduino Mega 2560	46
4.4.3 Program Arduino UNO	46
4.5 Pengujian Papan Informasi Berbasis <i>Dot Matrix</i>	46
4.5.1 Pengujian Konsumsi Arus <i>Dot Matrix</i>	48
4.5.2 Pengujian Menampilkan Waktu Pada <i>Dot Matrix</i> 32x32	48

4.5.3	Pengujian Menampilkan Logo Pada <i>Dot Matrix</i> 32x32	50
4.5.4	Pengujian Menampilkan Font Pada <i>Dot Matrix</i> 32x64	50
4.5.5	Pengujian Pengiriman dan Menampilkan Teks ke <i>Dot Matrix</i> 32x64.....	55
BAB V KESIMPULAN DAN SARAN.....		57
5.1	Kesimpulan	57
5.2	Saran	57
DAFTAR PUSTAKA		58
LAMPIRAN.....		60

DAFTAR TABEL

Tabel	Halaman
2.1 Spesifikasi Arduino UNO.....	11
2.2 Spesifikasi Arduino Mega 2560	13
2.3 Konfigurasi Pin Arduino Mega 2560	14
2.4 Spesifikasi <i>Bluetooth</i> HC-05	16
2.5 Konfigurasi pin RTC dan Arduino	18
4.1 kriteria material yang digunakan	25
4.2 Pengujian rangkaian catu daya	27
4.3 Konfigurasi pin RTC DS1307 dengan Arduino Mega 2560	31
4.4 Koneksi pin <i>Bluetooth</i> HC-05 dengan Arduino Mega 2560	34
4.5 Hasil Pengujian Jarak <i>Bluetooth</i>	35
4.6 Koneksi pin pada <i>Dot Matrix Display</i> P10 dengan Ardunio.....	37
4.7 Hasil perhitungan daya	48
4.8 Hasil Pengujian Menampilkan Waktu.....	49
4.9 Hasil Pengujian Font.....	55

DAFTAR GAMBAR

Gambar	Halaman
2.1 Skema <i>Dot Matrix Display</i> P10	4
2.2 Bentuk Fisik <i>Dot Matrix Display</i> P10	5
2.3 Beberapa Logo Versi Android.....	6
2.4 Beberapa Logo Versi Windows.....	7
2.5 Tampilan <i>Workbench</i> App Inventor	8
2.6 Tampilan <i>Workbench</i> Visual Basic	10
2.7 Arduino UNO	11
2.8 Pemetaan Pin Arduino UNO	12
2.9 Bentuk Fisik Arduino Mega 2560	13
2.10 Bentuk Fisik Modul <i>Bluetooth</i> HC-05.....	16
2.11 Pengiriman Data Lewat Komunikasi UART <i>Bluetooth</i>	17
2.12 RTC DS1307	18
2.13 <i>Power Supply</i>	18
3.1 <i>Flow chart</i> Proses Pembuatan Proyek Akhir.....	19
4.1 Diagram Blok Papan Informasi Berbasis <i>Dot Matrix</i>	23
4.2.a Hasil Desain 2D.....	24
4.2.b Hasil Desain 3D.....	24
4.3 Pembuatan frame <i>dot matrix</i>	26
4.4 Pembuatan <i>cover</i> belakang <i>dot matrix</i>	26
4.5 <i>Power Supply Switching</i> yang digunakan sebagai catu daya.....	27
4.6 Pemasangan Arduino Mega 2560.....	28
4.7 Pengujian Arduino Mega 2560.....	29
4.8 Pemasangan Arduino UNO	29
4.9 Pengujian Arduino UNO	30
4.10 Modul RTC DS1307.....	31
4.11 Pemasangan Pin Modul RTC dengan Arduino Mega 2560	32

4.12	Pengujian modul RTC DS1307	33
4.13	<i>Bluetooth</i> HC-05 yang telah terpasang pada alat Proyek akhir	33
4.14	Koneksi pin <i>Bluetooth</i> HC-05 dengan Arduino Mega 2560	34
4.15	Hasil Pengujian <i>Bluetooth</i> HC-05	35
4.16	<i>Dot Matrix Display</i> P10 yang telah terpasang pada alat Proyek akhir.....	36
4.17	Soket dari <i>Dot Matrix Display</i> P10.	37
4.18	Rangkaian Elektrik <i>Dot Matrix Display</i> P10 32 x 32 cm.....	38
4.19	Rangkaian Elektrik <i>Dot Matrix Display</i> P10 64 x 32 cm.....	39
4.20	Hasil Pengujian DMD P10 32x32	40
4.21	Hasil Pengujian DMD P10 32x64	41
4.22	Hasil Pembuatan Aplikasi SiPETRIX pada Android	42
4.23.a	Tampilan 1.....	43
4.23.b	Tampilan 2	43
4.23.c	Tampilan 3a.....	43
4.23.d	Tampilan 3b	43
4.23.e	Tampilan 3c.....	43
4.23.f	Tampilan 3d.....	43
4.24.a	Mengirim teks dari Android.....	45
4.24.b	Data yang diterima Arduino.....	45
4.25.a	Mengirim teks dari Windows	45
4.25.b	Data yang diterima Arduino.....	45
4.26	Hasil Akhir Papan Informasi Berbasis <i>Dot Matrix</i>	47
4.27.a	Hasil Pengujian Arus <i>Dot Matrix</i>	48
4.27.b	Arus Terukur pada <i>Dot Matrix</i>	48
4.28.a	Pengujian ke-1 Menampilkan Waktu.....	49
4.28.b	Pengujian ke-2 Menampilkan Waktu.....	49
4.28.c	Pengujian ke-3 Menampilkan Waktu.....	49
4.29	Hasil Pengujian Menampilkan Logo	50
4.30	Hasil Pengukuran Font Sangat Besar	51
4.31	Hasil Pengukuran Font Besar.....	52

4.32	Hasil Pengukuran Font Kecil.....	53
4.33	Hasil Pengukuran Font Sangat Kecil.....	54
4.34	Hasil Pengiriman 1 baris teks	55
4.35	Hasil Pengiriman 2 baris teks	55
4.36	Hasil Pengiriman 3 baris teks	56
4.37	Hasil Pengiriman 4 baris teks	56

DAFTAR LAMPIRAN

Lampiran 1 : Daftar Riwayat Hidup

Lampiran 2 : *Datasheet BLUETOOTH HC-05, Datasheet RTC DS1307,
Datasheet modul LED DOT MATRIX P10*

Lampiran 3 : Gambar Konstruksi Papan Informasi Berbasis *Dot Matrix*

Lampiran 4 : Program Pada Arduino MEGA 2560 dan Arduino UNO

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Bagian Administrasi Akademik, Kemahasiswaan, Perencanaan dan Kerja sama (BAAKPK) merupakan salah satu unit kerja yang ada pada Politeknik Manufaktur Negeri Bangka Belitung. Sistem penyampaian informasi berupa nama unit kerja dan jenis layanan yang disediakan oleh BAAKPK masih terbilang manual yaitu dengan menggunakan kertas yang ditempel di depan pintu masuk, sehingga memungkinkan informasi tersebut cepat rusak dan kurang diperhatikan oleh pembaca. Selain itu jika terjadi kerusakan atau perubahan informasi tentu akan memperumit petugas untuk proses penggantian seperti harus membuat, mencetak, kemudian menempel ulang informasi tersebut. Sistem penyampaian informasi seperti ini tentu kurang sesuai dengan Politeknik Manufaktur Negeri Bangka Belitung yang merupakan kampus berbasis teknologi. Oleh karena itu BAAKPK memerlukan sebuah media penyampai informasi yang lebih modern yaitu berupa papan informasi berbasis *Dot Matrix*, dengan menggunakan papan informasi berbasis *Dot Matrix* ini tentu akan lebih menarik perhatian pembaca serta dapat memudahkan petugas dalam proses penggantian informasi.

Papan informasi berbasis *dot matrix* atau yang lebih dikenal *Running Text* adalah sebuah perangkat yang digunakan sebagai media penyampai informasi yang terdiri dari LED (*Light Emitting Diode*) yang disusun dengan pola tertentu sehingga dapat membentuk karakter maupun gambar. Perkembangan ilmu dan teknologi saat ini pun membuat sistem pengontrolan pada perangkat ini menjadi semakin berkembang, salah satunya adalah penggunaan komunikasi nirkabel. Komunikasi nirkabel banyak digunakan sebagai salah satu *interface* pada peralatan elektronika sebagai sarana kontrol jarak jauh. Salah satu jenis komunikasi nirkabel yang paling banyak diaplikasikan saat ini adalah *Bluetooth*. Komunikasi nirkabel *Bluetooth* ini nantinya akan menghubungkan perangkat *Dot*

Matrik dengan *Smartphone* berbasis OS Android, sehingga akan lebih mempermudah petugas untuk mengganti informasi yang akan disampaikan.

Penelitian yang berkaitan dengan judul proyek akhir kami yaitu “Kontrol Sistem Informasi *Running Text* Menggunakan Arduino” [1]. Kekurangan dari proyek akhir ini adalah pada jumlah baris yang dapat ditampilkan yaitu berjumlah tiga baris, dimana dua baris bersifat tetap yang digunakan sebagai penunjuk waktu dan satu baris lainnya digunakan sebagai penampil informasi yang dapat diubah-ubah. Penggunaan satu baris membuat alat ini hanya bisa menampilkan 62 karakter. Penelitian selanjutnya yaitu “*Running Text* Menggunakan Keyboard AT PS/2 Berbasis Mikrokontroler Atmega8535” [2]. Hasil dari proyek akhir ini yaitu *running text* dengan pergantian informasi yang masih menggunakan proses download IC mikrokontroler dari downloader yang terhubung pada keyboard.

Pada penelitian atau proyek akhir sebelumnya, dirasakan kurang efektif untuk diletakan pada BAAKPK yang menampilkan informasi lebih dari 200 karakter dan berubah-ubah. Untuk itu perlu perbaikan agar perangkat tersebut lebih efektif dan efisien, sehingga kami mengembangkan perangkat ini dengan jumlah baris dan tinggi font yang bisa diubah, sehingga saat ingin menampilkan informasi yang sedikit, cukup menggunakan satu baris karakter, namun saat ingin menampilkan informasi dengan jumlah yang banyak maka menggunakan empat baris karakter, sehingga jumlah karakter yang ditampilkan menjadi lebih banyak. Selain itu pada perangkat sebelumnya teks yang ditampilkan hanya pergerakan dari kanan ke kiri sedangkan pada perangkat ini kami buat lebih bervariasi agar lebih menarik untuk dibaca.

Berdasarkan latar belakang tersebut dibuatlah sebuah alat pengembangan yang dapat memudahkan BAAKPK dalam penyampaian informasi berupa Papan Informasi Berbasis *Dot Matrix* dengan menggunakan aplikasi pada *Smartphone* Android dan Komputer berbasis Windows untuk mengubah informasi, serta dilengkapi dengan penunjuk waktu secara *real time*, jumlah baris karakter sebanyak 4 baris serta ukuran atau baris font yang bisa diubah-ubah.

1.2. Rumusan dan Batasan Masalah

1.2.1. Rumusan masalah

Adapun rumusan masalah yang diangkat berdasarkan latar belakang proyek akhir ini antara lain :

1. Bagaimana merancang dan mengembangkan *hardware* dan *software* agar *Dot Matrix* dapat terintegrasi dengan *Smartphone* OS Android dan *Laptop* OS Windows.
2. Bagaimana hasil pengujian peralatan yang dikembangkan.

1.2.2. Batasan masalah

Adapun batasan masalah pada proyek akhir ini adalah sebagai berikut:

1. Jumlah maksimal baris yang dapat ditampilkan pada *Dot Matrix* sebanyak 4 baris karakter.
2. Ukuran tinggi font yang dapat diubah sebanyak 4 jenis ukuran.
3. Jumlah maksimal karakter yang dapat dikirim pada setiap kotak teks sebanyak 55 karakter.
4. Jumlah *screen* yang bisa diubah sebanyak 3 *screen*.
5. Jarak Android ke *receiver Bluetooth* pada perangkat *dot matrix* tidak lebih dari 20 m.

1.3. Tujuan Proyek Akhir

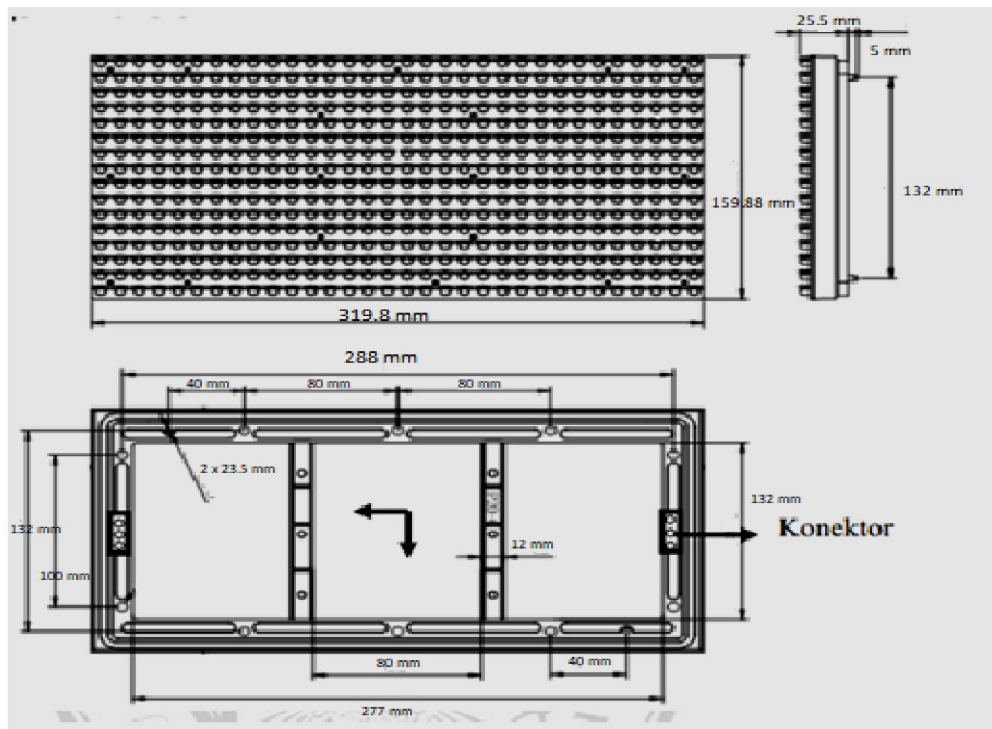
Adapun tujuan penulisan dalam penyusunan proyek akhir ini diantaranya adalah :

1. Merancang dan mengembangkan papan informasi berbasis *Dot Matrix* yang terintegrasi dengan *Smartphone* OS Android dan Komputer OS Windows.
2. Mendeskripsikan hasil uji coba peralatan yang dikembangkan.

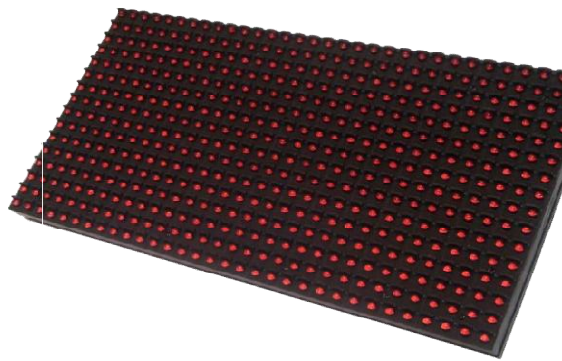
BAB II DASAR TEORI

2.1 *Dot Matrix Display P10*

Dot Matrix Display yaitu berupa LED (*Light Emitting Diode*) yang disambung dan dirangkai menjadi deretan LED ataupun dapat berupa *Dot Matrix*. *Dot Matrix* merupakan deretan LED yang membentuk *array* dengan jumlah kolom dan baris tertentu, sehingga titik-titik yang menyala dapat membentuk suatu karakter angka, huruf, tanda baca, dan sebagainya. Panel *Dot Matrix display P10* ukuran 16 x 32 merupakan modul display dot matrix yang sudah tersusun *register* untuk mengendalikan nyala *array* LED dan *input* teks. Jika *Dot Matrix* tidak menggunakan *shift register*, maka LED bisa menyala bersamaan satu kolom atau satu baris. Gambar 2.1 menunjukkan skema *Dot Matrix display P10* :



Gambar 2.1 Skema *Dot Matrix Display P10* [3]



Gambar 2.2 Bentuk Fisik *Dot Matrix Display* P10

2.2 Android

Android adalah sistem operasi perangkat *Mobile* berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet yang mencakup sistem operasi, *middleware*, dan aplikasi. Keunggulan utama Android adalah gratis dan *open source* bagi pengembang (*programmer*) untuk membuat aplikasi. Struktur aplikasi Android ditulis dalam bahasa pemrograman *Java*. Kode *Java* dikompilasi bersama dengan *file resource* yang dibutuhkan oleh aplikasi, prosesnya *dipackage* oleh *tools* yang dinamakan *apt tools* kedalam paket Android, sehingga menghasilkan *file* dengan ekstensi *apk*. *File* *apk* ini yang disebut dengan aplikasi dan nantinya dapat dijalankan pada *device/peralatan Mobile* [4].

Android, Inc. didirikan di Palo Alto, California, pada Oktober 2003 oleh Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Tujuan awal pengembangan Android adalah untuk mengembangkan sebuah sistem operasi canggih yang diperuntukkan bagi kamera digital, namun kemudian disadari bahwa pasar untuk perangkat tersebut tidak cukup besar, dan pengembangan Android dialihkan bagi pasar telepon pintar untuk menyaingi *Symbian* dan *Windows Mobile*. Google mengakuisisi Android Inc. pada 17 Agustus 2005, menjadikannya sebagai anak perusahaan yang sepenuhnya dimiliki oleh Google. Versi Android yang dirilis kini telah bermacam-macam. Adapun versi Android yang digunakan pada Proyek Akhir ini yaitu Android 4.4 (*KitKat*) yang dirilis pada 31 Oktober 2013, versi ini memiliki antar muka terbaru dengan status bar dan navigasi

transparan pada layar depan, *webviews* berbasis *chromium*, mendukung media komunikasi inframerah yang memungkinkan *devices* bisa menjadi *remote* untuk *smart TV*. Beberapa logo versi Android dapat dilihat pada Gambar 2.3 Berikut :



Gambar 2.3 Beberapa Logo Versi Android [5]

2.3 Windows

Microsoft Windows atau sering disebut dengan Windows merupakan sistem operasi yang dibuat dan dikembangkan oleh Microsoft Corporation, perusahaan perangkat lunak yang didirikan oleh Bill Gates pada tahun 1975 dan berkantor pusat di Redmond, Washington, Amerika Serikat [6].

Windows merupakan sebuah sistem operasi yang diciptakan oleh Microsoft, dimana sistem operasi ini menyediakan antarmuka grafis (GUI / Graphical User Interface) agar lebih mudah dioperasikan. Dengan adanya Windows, pengguna tidak perlu lagi mengetikkan perintah melalui command line layaknya pada MS-DOS. Cukup dengan menggunakan mouse atau keyboard, pengguna dapat memberikan perintah untuk membuka menu, kotak dialog, menjalankan aplikasi, menghapus file dan lain sebagainya. Berikut merupakan beberapa fungsi dari sistem operasi Windows:

1. Menghubungkan antara aplikasi dan perangkat keras, sehingga dapat terintegrasi bekerja secara konsisten dan stabil.
2. Mengendalikan dan mengelola sumber daya yang sedang dijalankan, termasuk perangkat lunak dan perangkat keras pada komputer.
3. Mengelola proses yang terdiri dari persiapan, penjadwalan, serta pemantauan program yang sedang dijalankan.

4. Mengelola data input dan output serta mengendalikannya.

Versi Windows yang dirilis kini telah bermacam-macam. Adapun versi Windows yang digunakan pada Proyek Akhir ini yaitu Windows 7 yang diluncurkan dengan misi memperbaiki berbagai macam masalah dan kritik yang diterima oleh Windows Vista. Microsoft menambah kemudahan pengguna dengan desain dari Windows 7 yang lebih baik. Windows 7 dapat dijalankan dengan lebih cepat, stabil, dan mudah, sehingga banyak pengguna yang akhirnya beralih ke Windows 7 dari sebelumnya XP atau Vista. Gambar 2.4 menunjukkan beberapa logo versi Windows.



Gambar 2.4 Beberapa Logo Versi Windows

2.4 App Inventor

App Inventor adalah sebuah aplikasi *builder* untuk membuat aplikasi yang berjalan di sistem operasi Android yang disediakan oleh Google *labs*. Jadi *user* harus mempunyai *account* Google dulu untuk bisa masuk ke *home* App Inventor. App Inventor ini menggunakan teknik *visual programming*, berbentuk seperti susunan *puzzle-puzzle* yang memiliki logika tertentu.

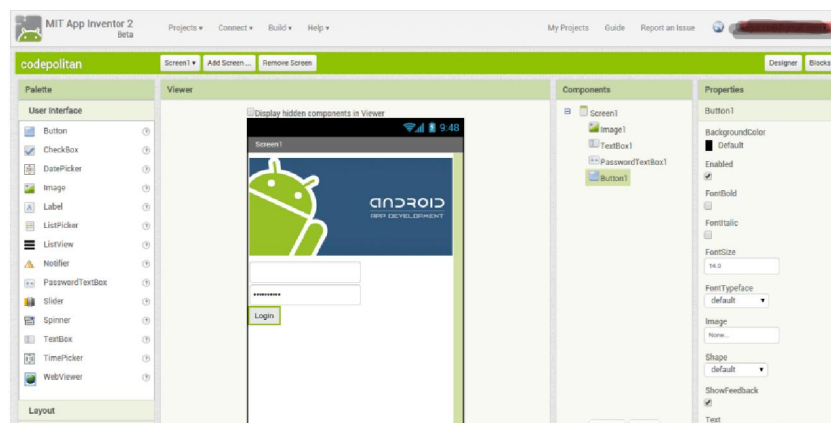
App Inventor kini dikembangkan oleh MIT, universitas yang bergerak di bidang teknologi. Semula App Inventor dikembangkan oleh Google, namun sekarang MIT yang memegang kendali terhadap pengembangan *tools* App Inventor. App inventor memungkinkan untuk mengembangkan aplikasi untuk ponsel Android menggunakan *browser web* dan baik telepon yang terhubung atau

emulator. *Server* App Inventor menyimpan pekerjaan dan proyek yang sudah dikerjakan sebelumnya.

Pada lingkungan kerja App Inventor ini terdapat beberapa komponen yang terdiri dari:

1. Komponen *desainer*, komponen *desainer* berjalan pada *browser* yang digunakan untuk memilih komponen yang dibutuhkan dan mengatur *property*-nya. Pada komponen *desainer* sendiri terdapat 5 bagian, yaitu *palette*, *viewer*, *component*, *media* dan *properties*.
 - *Palette*: list komponen yang bisa digunakan
 - *Viewer*: untuk menempatkan komponen dan mengaturnya sesuai tampilan yang diinginkan
 - *Component*: tempat list komponen yang dipakai pada *project*
 - *Media*: mengambil media audio atau gambar untuk *project*
 - *Properties*: mengatur *properties* komponen yang digunakan, seperti *width*, *height*, *name*, dan lain-lain
2. *Block Editor*, *Block Editor* berjalan di luar *browser* dan digunakan untuk membuat dan mengatur *behaviour* dari komponen-komponen yang dipilih dari komponen *desainer*.
3. *Emulator*, *Emulator* digunakan untuk menjalankan dan menguji *project* yang telah dibuat.

Tampilan *software* App Inventor dapat dilihat pada gambar 2.5 berikut.



Gambar 2.5 Tampilan *Workbench* App Inventor [7]

2.5 Microsoft Visual basic

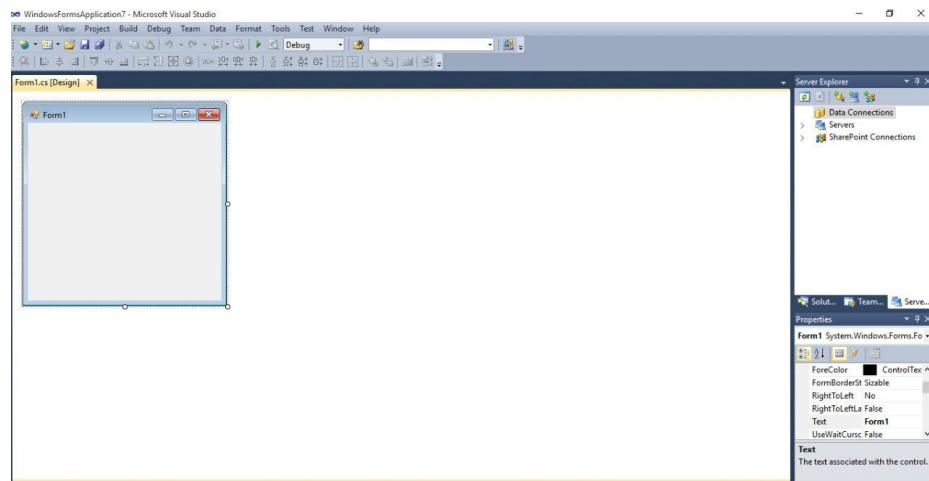
Microsoft Visual Basic atau lebih sering disebut VB adalah bahasa pemrograman berbasis windows yang merupakan *Object Oriented Programming* (OOP), yaitu pemrograman yang diorientasikan pada objek-objek yang disediakan di dalam visual basic itu sendiri. Bahasa BASIC pada dasarnya merupakan bahasa yang mudah untuk dimengerti sehingga pemrograman menggunakan bahasa BASIC dapat dilakukan dengan mudah sekalipun yang melakukannya adalah seorang pemula. Pada pemrograman visual ini, pengembangan aplikasi diawali dengan membentuk User Interface kemudian mengatur properti dari berbagai objek yang akan digunakan dalam User Interface, dan setelah itu baru dilakukan penulisan kode pada program di setiap event-event yang ada. Tahapan pengembangan aplikasi seperti ini lebih dikenal dengan pendekatan Bottom Up. Adapun Komponen yang digunakan pada Visual Basic antara lain:

1. Properti merupakan atribut-atribut yang terdapat pada kontrol (objek) yang biasanya berupa karakteristik penampilan seperti jenis huruf, ukuran, warna, dan sebagainya.
2. Event merupakan segala sesuatu atau kejadian yang dapat dialami oleh sebuah objek. Suatu kontrol (objek) bisa saja memiliki banyak event.
3. Method (Metode) adalah perbuatan atau aksi yang dimiliki oleh kontrol (objek) sehingga programmer dapat menggunakannya untuk memanipulasi sesuatu, method ini berasal dari penulisan kode yang diberikan oleh programmer.

Microsoft Visual Basic memiliki beberapa kelebihan diantaranya adalah sebagai berikut:

1. Bahasa pemrograman dengan teknik pemrograman visual yang dapat menjadikan penggunanya berekspresi lebih baik dalam menghasilkan suatu program aplikasi, sekalipun digunakan oleh pemula.
2. Memiliki fitur database dan multimedia yang semakin baik.
3. Memiliki pemrograman internet dengan DHTML (Dynamic Hyper Text Markup Language)

4. Dapat berinteraksi dengan aplikasi lain di dalam sistem operasi windows dengan komponen ActiveX Control.
5. Tidak memakan banyak memori komputer dan lebih ringan dari bahasa pemrograman yang lain.
6. File executable yang dihasilkan oleh Visual Basic berukuran kecil dan menggunakan resource yang tidak banyak.
7. Proses dalam belajar Visual Basic lebih singkat jika dibandingkan dengan bahasa pemrograman yang lain [8]. Tampilan *Workbench* Visual Basic dapat dilihat pada Gambar 2.6 berikut.



Gambar 2.6 Tampilan *Workbench* Visual Basic

2.6 Arduino Uno

Arduino UNO adalah sebuah board mikrokontroler yang didasarkan pada ATmega328. Arduino UNO memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkannya ke sebuah computer dengan sebuah kabel USB atau mensuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai untuk memulainya. Gambar *Hardware* arduino UNO ditunjukkan pada gambar 2.6.



Gambar 2.7 Arduino UNO [9]

Arduino Uno berbeda dari semua board Arduino sebelumnya, Arduino UNO tidak menggunakan chip driver FTDI USB-to-serial. Sebaliknya, fitur-fitur Atmega16U2 (Atmega8U2 sampai ke versi R2) diprogram sebagai sebuah pengubah USB ke serial. Revisi 2 dari board Arduino Uno mempunyai sebuah resistor yang menarik garis 8U2 HWB ke ground, yang membuatnya lebih mudah untuk diletakkan ke dalam DFU mode.

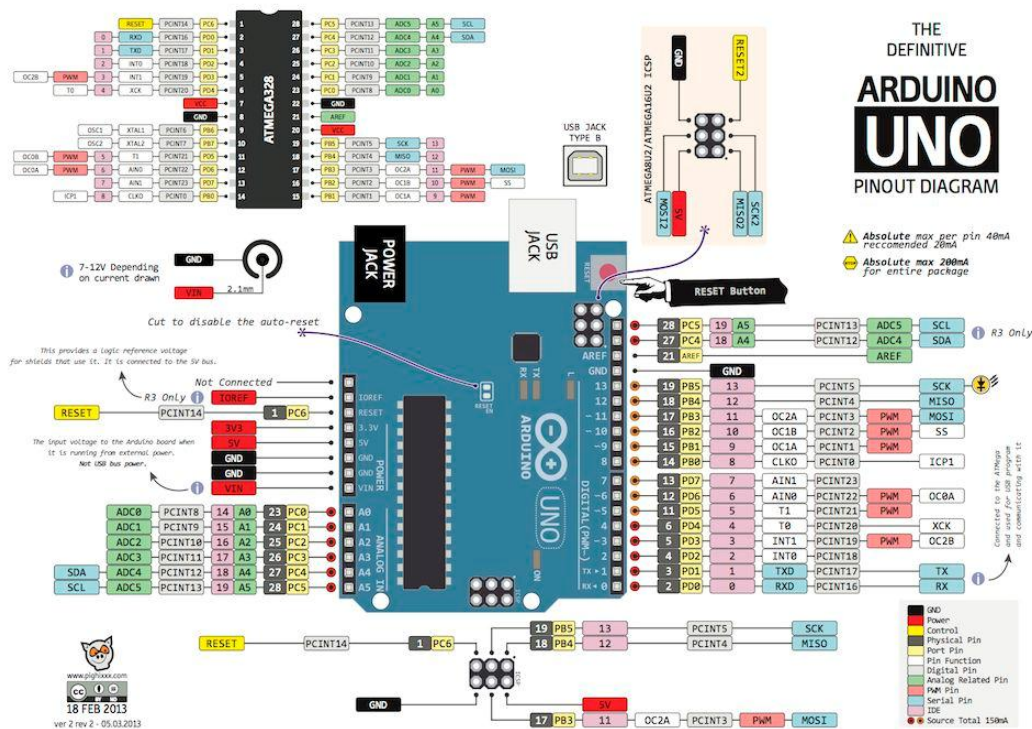
Spesifikasi dari Arduino UNO dapat dilihat pada tabel 2.1 berikut ini.

Tabel 2.1 Tabel Spesifikasi Arduino UNO

No	Spesifikasi	Keterangan
1	Mikrokontroler	ATmega328
2	Tegangan Operasi	5V
3	Tegangan <i>input</i> yang disarankan	7-12V
4	Batas tegangan <i>input</i>	6-20V
5	Jumlah pin I/O digital	14 (6 di antaranya menyediakan keluaran PWM)
6	Jumlah pin <i>input</i> analog	6
7	Jumlah pin I/O PWM	6
8	Arus DC tiap pin I/O	20 Ma
9	Arus DC untuk pin 3.3V	50 Ma
10	Memori Flash	32 KB (ATmega328), sekitar 0.5 KB

		digunakan oleh bootloader
11	SRAM	2 KB (ATmega328)
12	EEPROM	1 KB (ATmega328)
13	Clock Speed	16 MHz
14	LED_BUILTIN	13
15	Panjang	68.6mm
16	Lebar	53.4mm

Pemetaan dari pin Atmega 168/328 dengan Arduino UNO ditunjukkan pada gambar 2.8 berikut.



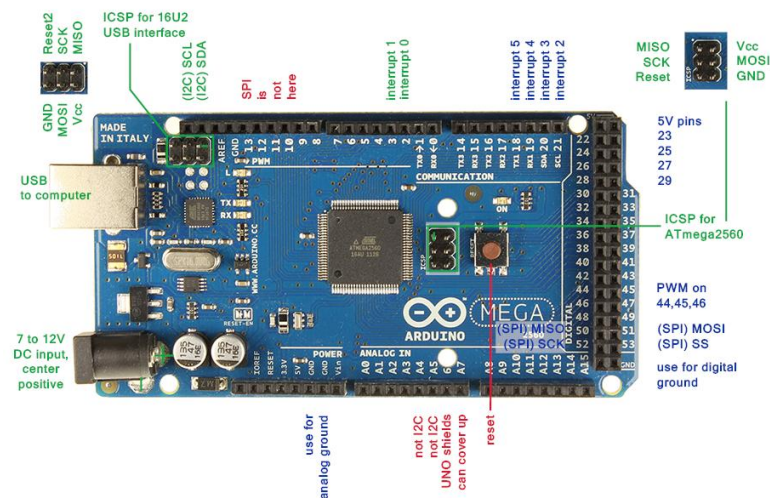
Gambar 2.8 Pemetaan Pin Arduino UNO

Pada gambar 2.8 di atas, Arduino UNO mempunyai 14 pin digital input/output (6 di antaranya dapat digunakan sebagai output PWM), 6 input analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah power jack, sebuah ICSP header, dan sebuah tombol reset.

2.7 Arduino Mega 2560

Arduino Mega 2560 adalah papan Mikrokontroler berdasarkan Atmega 2560 seperti yang ditunjukkan pada gambar 2.9. Mikrokontroler itu sendiri adalah *chip* atau IC (*integrated circuit*) yang bisa diprogram menggunakan komputer. Tujuan menanamkan program pada Mikrokontroler adalah agar rangkaian elektronik dapat membaca *input*, memproses *input* dan menghasilkan *output* sesuai yang diinginkan. Jadi Mikrokontroler bertugas sebagai otak yang mengendalikan *input*, proses dan *output* sebuah rangkaian elektronik. Adapun *pinout* diagram dari Arduino Mega 2560 ini dapat dilihat pada gambar 2.9.

Arduino tidak perlu perangkat *chip programmer* karena didalamnya sudah ada *bootloader* yang akan menangani *upload* program dari komputer. Bahasa pemrograman relatif mudah karena *software* Arduino dilengkapi dengan kumpulan *library* yang cukup lengkap. Bahasa pemrograman yang digunakan adalah bahasa C.



Gambar 2.9 Bentuk Fisik Arduino Mega 2560 [10]

Spesifikasi dari Arduino Mega 2560 dapat dilihat pada tabel 22 berikut.

Tabel 2.2 Spesifikasi Arduino Mega 2560

No.	Spesifikasi	Keterangan
1.	Mikrokontroler	Atmega 2560
2.	Tegangan Operasi	5V

3.	<i>Input Voltage</i> (disarankan)	7-12V
4.	<i>Input Voltage</i> (batas)	6-20V
5.	Digital I/O Pin	54 (15 <i>output</i> PWM)
6.	Pin Masukan Analog	16
7.	DC <i>Current</i> per I/O Pin	20mA
8.	DC saat ini untuk 3.3V Pin	50mA
9.	<i>Flash Memory</i>	256 KB , 8 KB digunakan oleh <i>bootloader</i>
10.	SRAM	8 KB
11.	EEPROM	4 KB
12.	Kecepatan <i>Clock</i>	16 Mhz
13.	LED_BUILTIN	13
14.	Panjang	10,52 mm
15.	Lebar	53,3 mm
16.	Berat	37 g

Konfigurasi pin Arduino Mega 2560 dapat dilihat pada tabel 2.3 berikut.

Tabel 2.3 Konfigurasi Pin Arduino Mega 2560

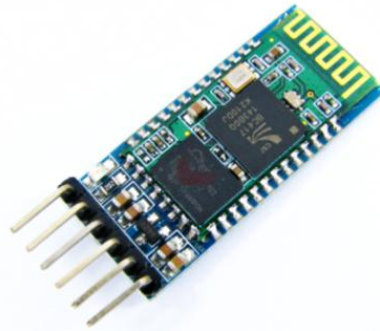
No.	Pin	Fungsi
1.	Vin	Tegangan <i>input</i> ke papan menggunakan sumber daya eksternal.
2.	5V	Memberikan <i>output</i> 5V yang diatur dari <i>regulator on-board</i> .
3.	3V3	Memberikan <i>output</i> 3,3V dengan arus maksimum 50 mA.
4.	GND	Pin ini berhubungan dngan tegangan 0 volt (<i>Ground</i>).
5.	IOREF	Pin ini menyediakan referensi tegangan operasi Mikrokontroler.
6.	<i>Serial</i>	<i>Serial 0: 0(RX) dan 1(TX); Serial 1: 19(RX) dan 18 (TX); Serial 2: 17(RX) dan 16(TX) dan Serial 3: 15(RX) dan 14(TX): Digunakan untuk menerima (RX) dan mengirimkan data serial (TX) TTL.</i>
7.	Interupsi	Eksternal 2 (<i>interrupt</i> 0), Eksternal 3 (<i>interrupt</i> 1), Eksternal 18 (<i>interrupt</i> 5), Eksternal 19 (<i>interrupt</i> 4), Eksternal 20 (<i>interrupt</i>

	3) dan Eksternal 21 (<i>interrupt 2</i>) dapat dikonfigurasi untuk memicu interupsi <i>rising up</i> atau <i>falling edge</i> .
8. PWM	Memberikan output PWM 8-bit.
9. SPI	50 (MISO), 51 (MOSI), 52 (SCK) dan 53 (SS) mendukung komunikasi SPI menggunakan <i>library</i> SPI.
10. LED 13	Terdapat built-in LED terhubung ke pin digital 13.
11. TWI	20 (SDA) dan 21 (SCL).
12. AREF	Tegangan referensi untuk <i>input</i> analog.
13. Reset	Untuk <i>me-reset</i> Mikrokontroler.

Pada tabel 2.3 di atas, pin-pin yang kami gunakan pada proyek akhir yaitu pin 4 sebagai input, RX dan TX 1 untuk *Bluetooth*, RX dan TX 2 untuk komunikasi serial ke Arduino Uno, serta pin SDA dan SCL untuk dihubungkan ke RTC.

2.8 *Bluetooth*

Bluetooth adalah sebuah teknologi komunikasi *wireless* (tanpa kabel) yang beroperasi dalam pita frekuensi 2,4 GHz *unlicensed ISM (Industrial, Scientific and Medical)* dengan menggunakan sebuah *frequency hopping tranceiver* yang mampu menyediakan layanan komunikasi data dan suara secara *real-time* antara *host-host Bluetooth* dengan jarak jangkauan layanan yang terbatas. *Bluetooth* HC-05 adalah modul siap pakai yang memiliki kemampuan berkomunikasi secara serial dengan protokol standar *Bluetooth* versi 2.0. Papan inti HC-05 (menggunakan *chipset* CSR BC417) sudah dipasangkan dengan adapter koneksi (*black-plane break-out board*) sehingga mudah untuk digunakan. Bentuk fisik modul HC-05 dapat dilihat pada gambar 2.10 berikut.



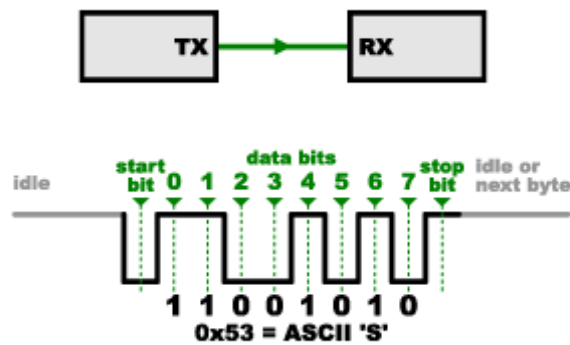
Gambar 2.10 Bentuk Fisik Modul *Bluetooth* HC-05

Modul inti HC-05 memiliki dua modus kerja yaitu modus eksekusi manual (memproses saat diperintahkan) dan modus koneksi otomatis. Pada mode eksekusi manual modul ini dioperasikan lewat perintah AT (*AT comands*) yang dikirimkan secara serial. Koneksi secara *default* diatur pada kecepatan 9.600 bps. HC-05 dapat berperan sebagai *master* dan *slave* pada *device*. Catu daya untuk modul ini sebesar 3,3V-5V. Spesifikasi dari *Bluetooth* HC-05 ditunjukkan pada tabel 2.4.

Tabel 2.4 Spesifikasi *Bluetooth* HC-05 [11]

No	Spesifikasi	Keterangan
1.	<i>Bluetooth</i>	v2.0+EDR
2.	Frekuensi	2.4Ghz ISM band
3.	Modulasi	GFSK (<i>Gaussian Frequency Shift Keying</i>)
4.	Kecepatan Asinkron	2.1 Mbps (Max)/160 bps
5.	Kecepatan Sinkron	1Mbps
6.	Keamanan	<i>Authentication and encryption</i>
7.	Profil	<i>Bluetooth serial port</i>
8.	Catu Daya	+3.3-5 VDC 50mA
9.	Rentang Suhu	-20°C hingga +75°C

Komunikasi serial *via Bluetooth* HC-05 menggunakan *device Bluetooth* HC-05 sebagai *converter* komunikasi serial level TTL (UART) kedalam bentuk komunikasi *wireless* yaitu *Bluetooth*. Komunikasi serial adalah komunikasi yang pengiriman datanya per-bit secara berurutan dan bergantian. *Bluetooth* HC-05 memiliki format pengiriman dan penerimaan data serial yang sama dengan peralatan yang menggunakan perangkat keras UART (*Universal Asynchronous Receiver-Transmitter*) seperti yang ditunjukkan pada gambar 2.11 Pengiriman data lewat komunikasi UART *Bluetooth* berikut.



Gambar 2.11 Pengiriman Data Lewat Komunikasi UART *Bluetooth* [12]

2.9 RTC

RTC merupakan komponen yang diperlukan untuk memberikan informasi mengenai waktu. Waktu disini dapat berupa detik, menit, hari, bulan dan tahun. Arduino (misalnya UNO) tidak dilengkapi secara internal dengan RTC. Dengan demikian, untuk aplikasi yang memerlukan pewaktuan, kita harus menyertakannya secara tersendiri. Agar tetap dapat bekerja, sebuah RTC dilengkapi dengan baterai, yang umumnya orang-orang menyebutkannya sebagai baterai "CMOS". Untuk konfigurasi anatara pin RTC dengan pin Arduino dapat dilihat pada tabel 2.5, untuk bentuk fisik dari RTC ditunjukkan oleh gambar 2.12.

Tabel 2.5 Konfigurasi pin RTC dan Arduino[13]

No	Pin RTC	Pin Arduino
1.	VCC	5V
2.	GND	GND
3.	SDA	Analog
4.	SCL	Analog



Gambar 2.12 RTC DS1307

2.10 Power Supply

Power supply ini berfungsi sebagai konversi dari tegangan AC 200-240VAC 2.5A ke tegangan DC 5V 40A. Tujuan dari penurunan tegangan ini adalah untuk memberi tegangan terhadap *Dot Matrix* dimana tegangan yang digunakan untuk mengaktifkan *Dot Matrix* adalah tegangan DC. Tegangan DC yang dipakai untuk mengaktifkan *Dot Matrix* sebesar 5V dan memakai arus 4A untuk masing-masing *Dot Matrix* [14]. Biasanya *power supply* diletakan dibagian belakang atas casing. Gambar 2.13 menunjukkan bentuk fisik dari *power supply* .

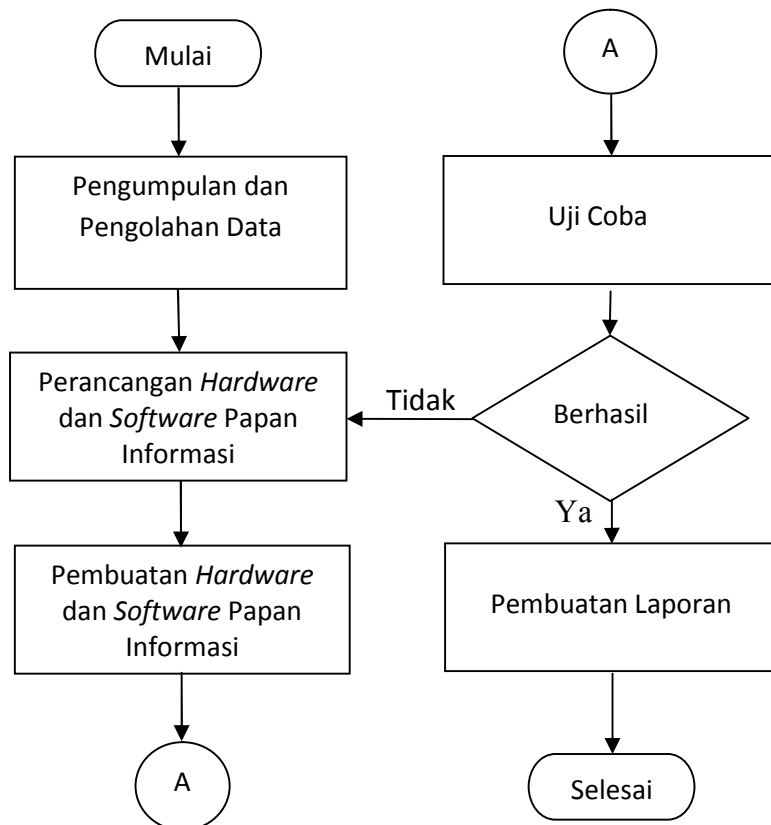


Gambar 2.13 Power Supply

BAB III METODE PELAKSANAAN

Dalam pembuatan Proyek Akhir dengan judul “Rancang Bangun Papan Informasi Berbasis *Dot Matrix* di BAAKPK POLMAN BABEL yang Terintegrasi dengan Android” ini dilakukan beberapa tahap dalam proses pengerjaan yang bertujuan untuk mempermudah dalam pencapaian target penyelesaian. Metode pelaksanaan yang diterapkan yaitu seperti yang ditunjukkan pada gambar 3.1 berikut.

3.1 Flow Chart Perancangan dan Pembuatan Alat



Gambar 3.1 *Flow Chart* proses pembuatan proyek akhir

Berdasarkan gambar *Flow Chart*, proses pengerjaan proyek akhir dilakukan melalui beberapa metode yaitu sebagai berikut.

3.2 Pengumpulan Data

Data-data yang diperlukan penulis dalam menyelesaikan proyek akhir ini terdiri dari data primer (langsung) dan data sekunder (tidak langsung). Maksud dari data primer adalah data yang didapatkan langsung dari hasil konsultasi dengan para dosen, sedangkan data sekunder adalah data yang didapatkan secara tidak langsung atau didapat dari internet, buku-buku, dll.

3.2.1 Pengumpulan Data Secara Tidak Langsung

Dalam metode pengumpulan data secara tidak langsung yaitu berupa pengumpulan data yang diperoleh dari *searching* di google, makalah PA pada tahun sebelumnya yang berjudul “Kontrol Sistem Informasi *Running Text* Menggunakan Arduino”, serta referensi-referensi dari buku yang masih berhubungan dengan proyek akhir penulis.

3.2.2 Pengumpulan Data Secara Langsung

Pada metode pengumpulan data ini, penulis memperoleh data dari konsultasi bersama Bapak Eko Sulisty, M.T. selaku dosen pembimbing 1 dan Bapak Aan Febriansyah, M.T. selaku dosen pembimbing 2 serta pihak BAAKPK POLMAN BABEL, selain itu data juga diperoleh dari teman-teman yang mengetahui dan memiliki pengetahuan tentang proyek akhir penulis.

3.3 Pengolahan Data

Pada tahap pengolahan data, data-data yang telah didapatkan akan dipilih dan dikumpulkan untuk menjadi referensi dan acuan dalam pembuatan proyek akhir.

3.4 Perancangan *Hardware* dan *Software*

Perancangan *hardware* dan *software* meliputi perancangan alat dan kontrol yang akan diterapkan pada proyek akhir.

Berikut rencana rancangan *hardware*:

1. Perancangan ukuran Papan Informasi.
2. Bahan dan ukuran frame serta *cover* belakang yang akan digunakan.
3. Rangkaian kontrol yang akan digunakan.

Berikut rencana rancangan *software*:

1. Rancangan tampilan aplikasi pada *Smartphone* dan *Laptop*
2. Program Arduino

3.5 Pembuatan *Hardware* dan *Software*

Pembuatan *hardware* meliputi pemilihan tipe dan warna *Dot Matrix*, pembuatan frame dan pembuatan bagian-bagian mekanik lainnya serta pembuatan rangkaian kontrol yang digunakan. Seluruh bahan, komponen, maupun alat yang digunakan dalam pembuatan Proyek Akhir ini menggunakan bahan, komponen dan peralatan yang mudah didapatkan serta memiliki harga yang terjangkau bagi mahasiswa pada umumnya.

Pembuatan *software* yang dilakukan meliputi pembuatan aplikasi serta program untuk menjalankan *Dot Matrix*. Pembuatan aplikasi merupakan proses perancangan serta pembuatan aplikasi yang akan digunakan pada Android dan Windows. Setelah rangkaian kontrol serta aplikasi Android dan Windows selesai dibuat, dilakukan proses pemrograman. Pemrograman berfungsi untuk mengontrol *Dot Matrix* agar dapat menjalankan fungsinya sesuai dengan yang diinginkan. Proses pemrograman meliputi metode pemrograman yang digunakan serta bagaimana sistem kontrol yang diterapkan. Pemrograman dilakukan dengan menggunakan *software* Arduino IDE dan bahasa yang digunakan dalam pemrograman adalah bahasa C.

3.6 Uji Coba

Setelah melakukan pembuatan hardware dan *software*, hal yang harus dilakukan adalah melakukan uji coba. Tahap ini dilakukan untuk mengetahui berhasil atau tidak proses pembuatan proyek akhir dengan cara mengetes proyek akhir sesuai dengan syarat-syarat yang telah ditentukan.

Uji coba yang dilakukan antara lain :

1. Uji coba pengiriman karakter dari Aplikasi Android atau Aplikasi Windows ke Arduino
2. Uji coba menampilkan Karakter yang dikirim ke *Dot Matrix*
3. Uji coba menampilkan waktu secara *real time* pada *Dot Matrix*
4. Uji coba memilih ukuran karakter melalui Android dan Windows
5. Uji coba menampilkan 4 baris karakter ke *Dot Matrix*

3.7 Perbaikan

Tahap ini akan dilakukan apabila pada saat melakukan uji coba mengalami kegagalan atau masih belum sesuai dengan apa yang diinginkan sehingga harus melakukan perbaikan baik secara *software* atau secara *hardware* tergantung hasil analisa dari penyebab kegagalan. Setelah perbaikan selesai kemudian dilakukan uji coba ulang.

3.8 Pembuatan Laporan

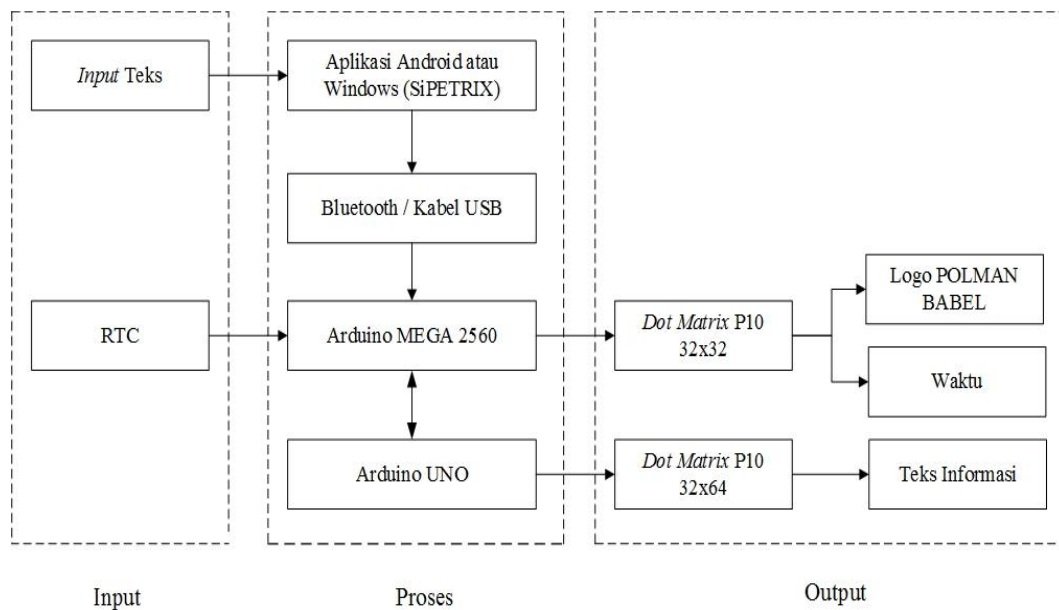
Tahap ini merupakan tahap terakhir dalam pembuatan proyek akhir yang bertujuan untuk merangkum keseluruhan detail mengenai alat yang dibuat, yang meliputi latar belakang, tujuan, rumusan masalah, batasan masalah, landasan teori, metode pelaksanaan, pembahasan, serta kesimpulan dan saran.

BAB IV PEMBAHASAN

Pada bab ini akan dibahas mengenai proses serta pengujian alat yang dikembangkan dalam pembuatan Proyek Akhir dengan judul “Rancang Bangun Papan Informasi Berbasis *Dot Matrix* yang Terintegrasi dengan Android”, yaitu sebagai berikut:

4.1 Diagram Blok

Diagram blok dari Rancang Bangun Papan Informasi Berbasis *Dot Matrix* yang Terintegrasi dengan Android ditunjukkan pada gambar 4.1 berikut.



Gambar 4.1 Diagram Blok Papan Informasi Berbasis *Dot Matrix*

Teks yang ingin ditampilkan pada perangkat *Dot Matrix* 32x64 diketik pada aplikasi bernama SiPETRIX yang ada pada Android atau Windows. Kemudian data teks berupa string tersebut dikirim ke Arduino Mega 2560. Proses pengiriman data dari Android ke Arduino Mega 2560 menggunakan media *Bluetooth*, sedangkan proses pengiriman data dari Windows menggunakan kabel

USB. Setelah data teks diterima oleh Arduino Mega 2560 data dikirim ke Arduino UNO menggunakan komunikasi Serial untuk diolah agar data teks dapat ditampilkan pada *Dot Matrix* 32x64. Untuk menampilkan waktu pada *Dot Matrix* 16x32 memanfaatkan modul RTC (*Real Time Clock*) sebagai sumber data pewaktu, data tersebut dikirim ke Arduino Mega 2560 menggunakan komunikasi I2C (*Inter Integrated Circuit*) untuk diolah agar data pewaktu tersebut dapat ditampilkan pada *Dot Matrix* 16x32. Pada *Dot Matrix* 16x32, selain menampilkan data waktu juga menampilkan data berupa logo POLMAN BABEL, dimana penampilannya dilakukan secara bergantian sesuai dengan intruksi yang diberikan oleh Arduino UNO.

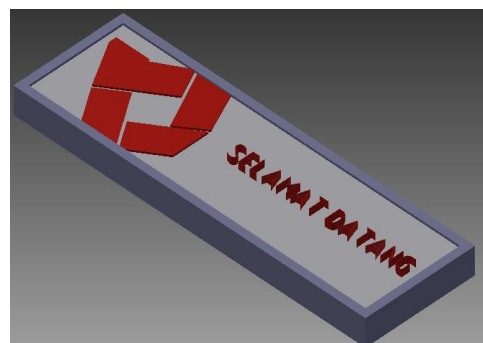
4.2 *Hardware* Mekanik Papan Informasi Berbasis *Dot Matrix*

4.2.1 Perancangan *Hardware* Mekanik

Perancangan *hardware* mekanik adalah proses pembuatan desain Papan Informasi Berbasis *Dot Matrix*. Pembuatan desain ini mencakup bagaimana konstruksi yang akan dibuat. Pembuatan desain dilakukan melalui PC/Komputer menggunakan *software* Autodesk Inventor. Proses awal yang dilakukan adalah menentukan ukuran/dimensi dari Papan Informasi yang akan dibuat yaitu 120x8,6x37,6 cm. Desain papan Informasi Berbasis *Dot Matrix* yang dibuat dapat dilihat pada gambar 4.2 dibawah ini.



Gambar 4.2a Hasil Desain 2D



Gambar 4.2b Hasil Desain 3D

4.2.2 Pembuatan *Hardware* Mekanik Papan Informasi Berbasis *Dot Matrix*

Pembuatan *hardware* mekanik papan informasi berbasis *dot matrix* dilakukan dalam beberapa tahapan antara lain:

A. Pemilihan Bahan

Konstruksi *frame* dan *cover* belakang *Dot Matrix* dibuat dengan memperhitungkan beberapa kriteria dalam pemilihan materialnya. Tidak hanya itu, waktu pengerjaan dan harga juga menjadi pertimbangan. Dari pertimbangan tersebut didapat kriteria material yang akan digunakan dalam pembuatan konstruksi seperti pada table 4.1.

Tabel 4.1 Tabel kriteria material yang digunakan

No.	Kriteria	Alasan
1.	Ringan	Agar beban tidak terlalu berat saat alat digantung
2.	Mudah dikerjakan	Agar tidak memakan waktu dalam proses pengerjaan
3.	Harga terjangkau	Menghemat dana tanpa mengurangi fungsi dari alat
4.	Mudah didapat	Agar proses pengerjaan tidak terhambat karena ketidaktersediaan bahan baku
5.	Perawatan mudah	Agar tidak perlu perawatan berkala pada <i>frame</i> dan <i>cover</i> belakang

Berdasarkan beberapa pertimbangan diatas kami memutuskan untuk menggunakan bahan berjenis Alumunium dikarenakan bahan Alumunium mudah didapat, ringan, mudah dikerjakan, harga terjangkau, serta perawatan yang mudah.

B. Pembuatan *Frame*

Proses pembuatan *Frame* berukuran 102x7,6x37,6 cm ini diperlukan sebagai penutup bagian samping, bagian atas dan bagian bawah dari *dot matrix*. Selain itu *frame* juga digunakan sebagai dudukan untuk *dot matrix* dan beberapa komponen lainnya seperti *Power Supply*. Pada *frame* yang telah dibuat terdapat empat buah plat yang telah dibentuk sedemikian rupa guna memperkuat *frame* dan sebagai penghubung antar *dot matrix*. Hasil dari *frame* yang telah dibuat dapat dilihat pada gambar 4.3 berikut.



Gambar 4.3 Pembuatan frame *dot matrix*

C. Pembuatan *Cover* Belakang

Pembuatan *cover* belakang diperlukan sebagai penutup bagian belakang dari *dot matrix*, serta menutupi komponen-komponen elektrik dan kontrol yang ada didalam *frame* dari beberapa faktor seperti debu, percikan air, dan sebagainya. Ukuran dari *cover* belakang yang telah dibuat adalah 102x1x37,6 cm. Pada gambar 4.4 menunjukkan gambar hasil *cover* belakang yang telah dibuat.

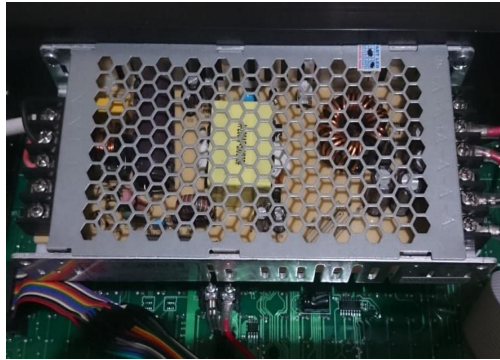


Gambar 4.4 Pembuatan *cover* belakang *dot matrix*

4.3 *Hardware* Elektrik Papan Informasi Berbasis *Dot Matrix*

4.3.1 *Catu Daya*







Pada proyek akhir ini *catu daya* yang digunakan adalah *power supply switching* dengan tegangan masukan sebesar 200-240VAC dan arus sebesar 2,5A. Sedangkan untuk tegangan keluaran dari *power supply switching* ini sebesar +5VDC dengan arus 40A. Gambar 4.5 menunjukkan gambar *power supply switching* yang digunakan dalam proyek akhir.



Gambar 4.5 *Power Supply Switching* yang digunakan sebagai catu daya

Pada saat uji coba rangkaian catu daya, dilakukan pengujian apakah rangkaian catu daya yang digunakan sesuai dengan yang diinginkan. Berikut adalah tabel hasil pengujian rangkaian catu daya yang ditunjukkan pada tabel 4.2 berikut.

Tabel 4.2 Pengujian rangkaian catu daya

Tegangan	Arus	Keterangan
		Tegangan yang dihasilkan : 5 V, Arus per LED : 0.0212A
		Tegangan yang dihasilkan : 5 V, Arus per blok : 3.63 A
		Tegangan yang dihasilkan : 5 V, Arus 6 blok : 6.94 A

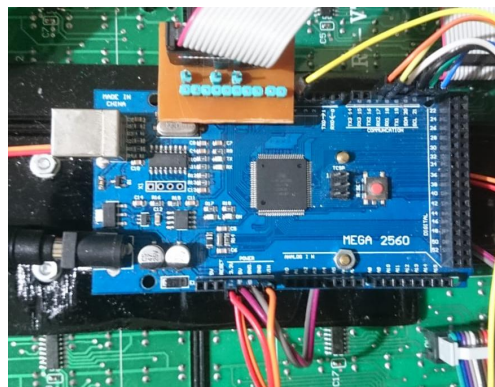
Dari tabel pengujian di atas, didapat hasil persentase error dari pengujian tegangan yaitu 0%. Hasil persentase error dari pengujian arus 6 blok yaitu 0.8%.

4.3.2 Arduino Mega 2560

Arduino Mega 2560 merupakan salah satu Mikrokontroler yang digunakan untuk mengontrol sebagian sistem dari Proyek Akhir ini. Arduino Mega 2560 dipilih karena memiliki pin RX dan TX lebih banyak dibanding Arduino tipe Nano ataupun UNO. Pin RX dan TX ini digunakan untuk komunikasi serial.

A. Pembuatan Arduino Mega 2560

Pembuatan Arduino Mega 2560 diputuskan dengan membeli modul Arduino Mega 2560 yang telah jadi dan mudah didapat dipasaran. Pemasangan Arduino Mega 2560 pada Papan Informasi berbasis *Dot Matrix* dapat dilihat pada gambar 4.6 berikut.



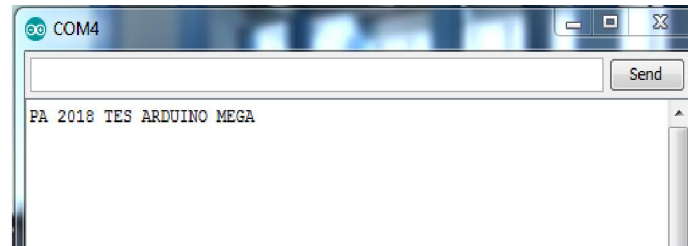
Gambar 4.6 Pemasangan Arduino Mega 2560

B. Pengujian Arduino Mega 2560

Pengujian Arduino Mega 2560 dilakukan dengan mengecek Arduino Mega 2560 pada PC/Komputer menggunakan *software* Arduino IDE 1.8.4. Pengujian yang dilakukan antara lain pengujian PORT Arduino Mega 2560 pada COM yang ada pada *device manager* di PC/Komputer. Berikut adalah *list* program pada pengujian Arduino Mega 2560 beserta hasil dari pengujian yang ditunjukkan pada gambar 4.7.

```
void setup() {  
  Serial.begin(9600);  
  Serial.println("PA 2018 TES ARDUINO MEGA");  
}
```

```
void loop() {};
```



Gambar 4.7 Pengujian Arduino Mega 2560

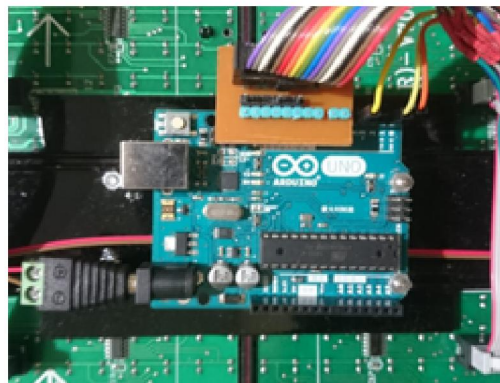
Dari hasil pengujian yang dilakukan dapat dianalisa bahwa Arduino Mega 2560 yang digunakan sebagai Mikrokontroler pada Proyek Akhir ini dapat berfungsi dengan baik dan sesuai dengan program yang telah dibuat.

4.3.3 Arduino UNO

Selain menggunakan Arduino Mega 2560, pembuatan Proyek Akhir ini juga menggunakan Arduino UNO sebagai Mikrokontroler yang digunakan untuk mengontrol sebagian dari sistem Proyek Akhir. Arduino UNO dipilih karena Arduino jenis ini kompatibel dengan pustaka DMD dimana pustaka tipe ini lebih stabil dibandingkan pustaka DMD2 yang digunakan pada *Dot Matrix* P10.

A. Pembuatan Arduino UNO

Pembuatan Arduino UNO diputuskan dengan membeli modul Arduino UNO yang telah jadi dan mudah didapat dipasaran. Pemasangan Arduino UNO pada Papan Informasi berbasis *Dot Matrix* dapat dilihat pada gambar 4.8 berikut.



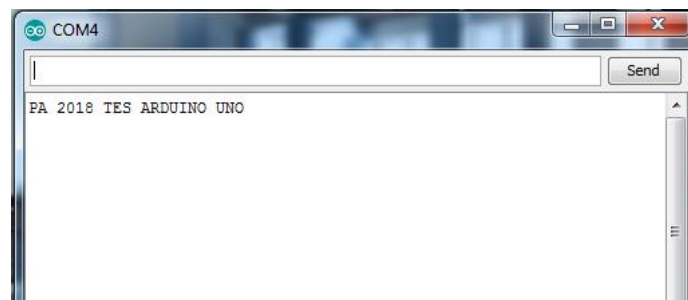
Gambar 4.8 Pemasangan Arduino UNO

B. Pengujian Arduino UNO

Proses pengujian Arduino UNO sama dengan proses pengujian yang dilakukan pada Arduino Mega 2560 yaitu dengan cara mengecek Arduino UNO pada PC/Komputer menggunakan *software* Arduino IDE 1.8.4. Pengujian yang dilakukan antara lain pengujian PORT Arduino UNO pada COM yang ada pada *device manager* di PC/Komputer.

Berikut adalah *list* program pada pengujian Arduino UNO. Hasil dari pengujian Arduino UNO dapat dilihat pada gambar 4.9 berikut.

```
void setup() {  
  Serial.begin(9600);  
  Serial.println("PA 2018 TES ARDUINO UNO");  
}  
void loop() {};
```



Gambar 4.9 Pengujian Arduino UNO

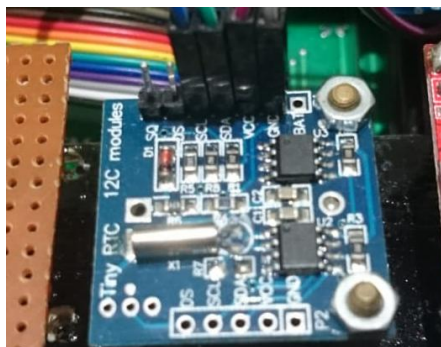
Dari hasil pengujian yang dilakukan dapat dianalisa bahwa Arduino UNO yang digunakan sebagai Mikrokontroler pada Proyek Akhir dapat berfungsi dengan baik dan sesuai dengan program yang telah dibuat.

4.3.4 Modul RTC DS1307

Modul RTC (*Real Time Clock*) digunakan sebagai sumber data pewaktu. Data tersebut dikirim ke Arduino Mega 2560 menggunakan komunikasi I2C (*Inter Integrated Circuit*) untuk diolah agar data pewaktu tersebut dapat ditampilkan pada *dot matrix* 32x32.

A. Pembuatan Modul RTC DS1307

Pembuatan Modul RTC DS1307 diputuskan dengan membeli modul yang telah jadi dan yang mudah didapatkan di pasaran. Hal ini dikarenakan modul yang dijual secara umum dapat langsung digunakan tanpa harus membuat modul RTC tersebut, hal ini tentu dapat meminimalisir waktu dalam proses pengerjaan proyek akhir. Gambar 4.10 menunjukkan modul RTC DS1307 yang sudah terpasang pada Papan Informasi berbasis *Dot Matrix*.



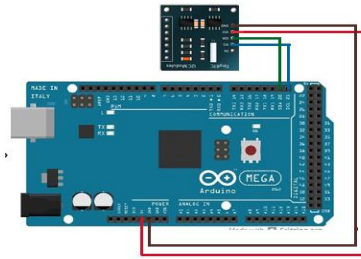
Gambar 4.10 Modul RTC DS1307

Konfigurasi pin modul RTC DS1307 dengan Arduino Mega 2560 dapat dilihat pada tabel 4.3 berikut.

Tabel 4.3 Konfigurasi pin RTC DS1307 dengan Arduino Mega 2560

No.	Pin RTC DS1307	Pin Arduino MEGA 2560
1	VCC	5V
2	GND	GND
3	SDA	20
4	SCL	21

Dari tabel 4.3, pemasangan pin modul RTC DS1307 dengan Arduino Mega 2560 ditunjukkan oleh gambar 4.11 berikut.



Gambar 4.11 Pemasangan Pin modul RTC dengan Arduino Mega 2560

B. Pengujian Modul RTC DS1307

Pengujian modul RTC DS1307 dilakukan dengan membuat program menampilkan waktu secara *real time* pada serial monitor menggunakan *software* Arduino IDE. Berikut adalah *list* program Arduino untuk pengujian modul RTC DS1307.

```
#include <Wire.h>
#include "RTClib.h"
RTC_DS1307 rtc;
void setup () {
  Serial.begin(9600);
  rtc.begin();
  rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
}
void loop () {
  DateTime now = rtc.now();
  Serial.print(now.day(), DEC);
  Serial.print('/');
  Serial.print(now.month(), DEC);
  Serial.print('/');
  Serial.print(now.year(), DEC);
  Serial.print(" ");
  Serial.print(now.hour(), DEC);
  Serial.print(':');
  Serial.print(now.minute(), DEC);
  Serial.print(':');
  Serial.print(now.second(), DEC);
  Serial.println();
  delay(1000);
}
```

Menampilkan
waktu pada
serial monitor

Gambar 4.12 menunjukkan hasil pengujian modul RTC DS1307 berdasarkan *list* program diatas.

```
6/8/2018 22:20:14
6/8/2018 22:20:15
6/8/2018 22:20:16
6/8/2018 22:20:17
6/8/2018 22:20:18
6/8/2018 22:20:19
6/8/2018 22:20:20
```

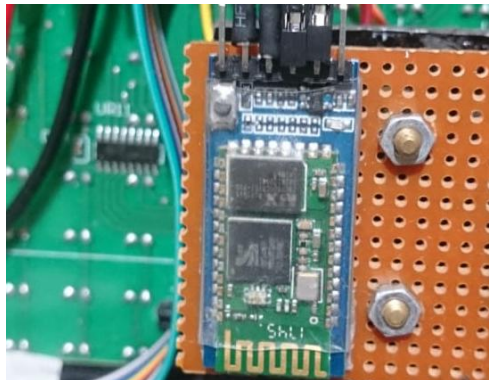
Gambar 4.12 Pengujian modul RTC DS1307

4.3.5 *Bluetooth* HC-05

Bluetooth HC-05 digunakan untuk menerima data yang dikirimkan oleh aplikasi SiPETRIX pada Android melalui *bluetooth* yang ada pada Smartphone. Setelah *bluetooth* HC-05 menerima data, kemudian data tersebut dikirim ke Arduino Mega 2560 melalui komunikasi serial.

A. Pembuatan *Bluetooth* HC-05

Pembuatan *Bluetooth* HC-05 diputuskan dengan membeli modul yang telah jadi dan mudah didapat dipasaran. Adapun *Bluetooth* HC-05 yang digunakan pada proyek akhir dapat dilihat pada gambar 4.13 berikut.

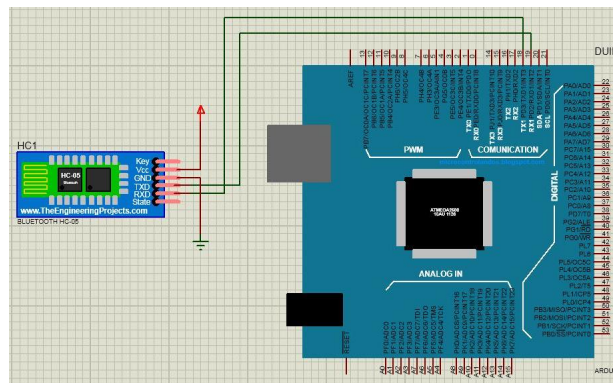


Gambar 4.13 *Bluetooth* HC-05 yang telah terpasang pada alat Proyek akhir

Koneksi pin *Bluetooth* HC-05 dengan Ardunio Mega 2560 dapat dilihat pada tabel 4.4 dan pada gambar 4.14 berikut.

Tabel 4.4 Koneksi pin *Bluetooth* HC-05 dengan Arduino Mega 2560

No.	<i>Bluetooth</i> HC-05	Pin Arduino MEGA 2560
1	VCC	3,3V
2	GND	GND
3	RX	18 (TX1)
4	TX	19 (RX1)



Gambar 4.14 Koneksi pin *Bluetooth* HC-05 dengan Arduino Mega 2560

B. Pengujian *Bluetooth* HC-05

Langkah awal dalam pengujian modul *Bluetooth* HC-05 yaitu *Bluetooth* disetting terlebih dahulu menggunakan perintah *AT Command* lewat *Serial Monitor* menggunakan Arduino Mega 2560. Langkah yang dilakukan yaitu hubungkan pin *Bluetooth* seperti pada pengkoneksian yang dibuat, hubungkan pin *Key* ke VCC/+5V, tekan tombol yang ada pada *Bluetooth* kemudian pasang catu daya tunggu selama 3 detik hingga LED pada *Bluetooth* HC-05 berkedip pelan. Selanjutnya masuk ke *Serial Monitor* pada Arduino, perintah yang ditulis yaitu :

- AT+NAME: Untuk memberikan nama *Bluetooth*.
- AT+UART=9600,0: Untuk mengatur *baudrate Bluetooth* ke 9600.
- AT+PSWD=1234: Untuk mengatur *password*.

Berikut adalah *list* program Arduino Mega 2560 untuk membaca data yang dikirim oleh aplikasi SiPETRIX pada Android.

```
void setup() {
  Serial.begin(9600);
```

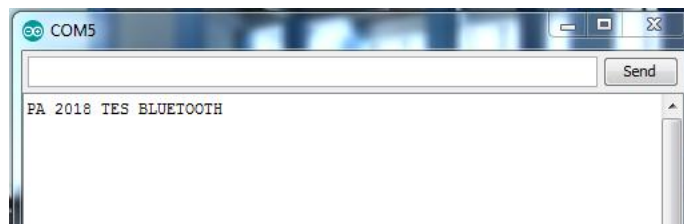
```

Serial1.begin(9600);
}
void loop() {
  if (Serial1.available()) {
    Serial.write(Serial1.read());
  }
}

```

Pembacaan Data
Bluetooth

Adapun hasil pengujian yang dilakukan terhadap *Bluetooth* HC-05 dapat dilihat pada gambar 4.15 berikut.



Gambar 4.15 Hasil Pengujian *Bluetooth* HC-05

Sedangkan hasil pengujian jarak penerimaan data oleh *bluetooth* HC-05 yang telah terpasang pada alat dapat dilihat pada tabel 4.5 berikut.

Tabel 4.5 Hasil Pengujian Jarak *Bluetooth*

No.	Jarak	Data Diterima
1.	1 Meter	Ya
2.	3 Meter	Ya
3.	5 Meter	Ya
4.	7 Meter	Ya
5.	9 Meter	Ya
6.	11 Meter	Ya
7.	13 Meter	Ya
8.	15 Meter	Ya
9.	17 Meter	Ya
10.	20 Meter	Ya
11.	21 Meter	Tidak

Dari pengujian yang dilakukan dapat dianalisa bahwa *Bluetooth* HC-05 yang digunakan dapat menerima data yang dikirim oleh aplikasi pada Android hingga jarak 20 meter. Pada pengujian jarak 21 meter *Bluetooth* HC-05 tidak lagi dapat menerima data yang dikirim oleh aplikasi pada Android. Android dapat terhubung dan mengirim data dengan baik pada modul *Bluetooth* HC-05 yang digunakan pada Papan Informasi Berbasis *Dot Matrix*. Dari hasil pengujian dan analisa dapat disimpulkan bahwa *Bluetooth* yang digunakan sesuai dengan perancangan yang diinginkan.

4.3.6 *Dot Matrix Display* P10

Dot Matrix Display P10 digunakan sebagai *Outputan* atau hasil akhir dari proyek akhir, dimana nantinya *Dot Matrix Display* P10 akan menampilkan berbagai jenis informasi.

A. Pembuatan *Dot Matrix Display* P10

Pembuatan *Dot Matrix Display* P10 diputuskan dengan membeli modul yang telah jadi dan biasa dijual secara umum dipasaran. Adapun *Dot Matrix Display* P10 yang digunakan dan telah terpasang pada alat proyek akhir ini dapat dilihat pada gambar 4.16 berikut.



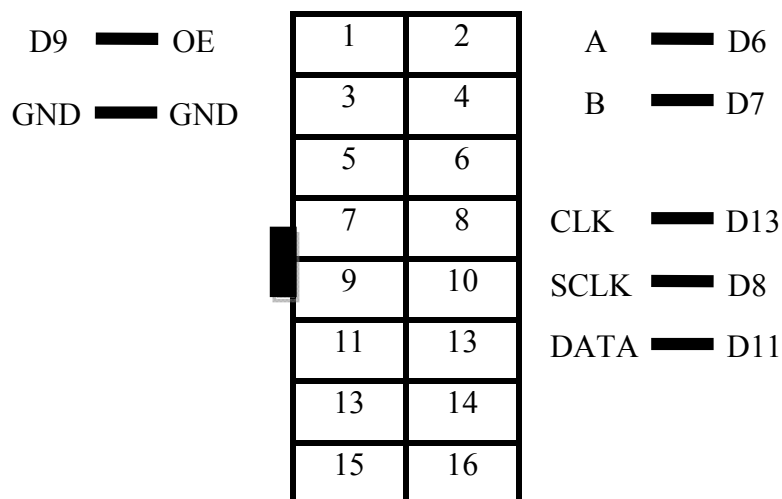
Gambar 4.16 *Dot Matrix Display* P10 yang telah terpasang pada alat Proyek akhir

Koneksi pin pada *Dot Matrix Display* P10 dengan Arduino dapat dilihat pada tabel 4.6 berikut.

Tabel 4.6 Koneksi pin pada *Dot Matrix Display* P10 dengan Arduino

No.	Soket Pin DMD P10	Pin Arduino
1	1	9 (D9)
2	2	6 (D6)
3	3	GND
4	4	7 (D7)
5	8	13 (D13)
6	10	8 (D8)
7	12	11 (D11)

Gambar 4.17 merupakan gambar soket dari *Dot Matrix Display* P10.



Dari gambar 4.17 di atas dapat dilihat bahwa :

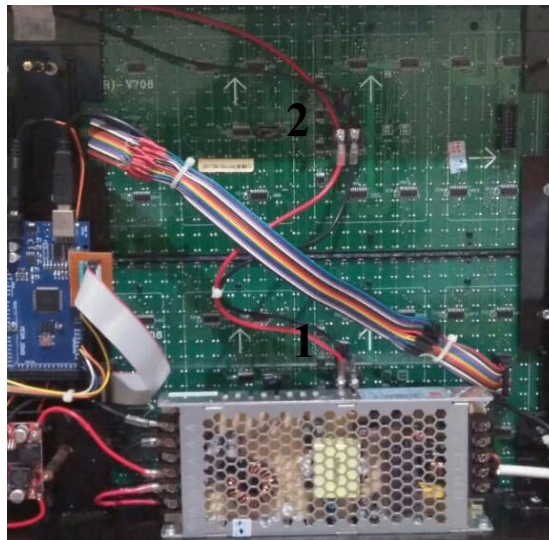
- OE : Output Enable untuk on/off semua LED
- A dan B : memilih kolom yang aktif
- CLK : SPI clock
- SCLK : Latch data register
- Data : Serial data SPI

Dot Matrix Display P10 yang digunakan pada proyek akhir ini sebanyak 6 buah modul, dimana sistem pengoperasiannya dibagi menjadi dua yaitu, 2 buah

Dot Matrix Display P10 yang disusun 1 buah kesamping dan 2 buah disusun kebawah (*Dot matrix* 32x32) dan dioperasikan oleh Arduino Mega 2560 serta 2 buah *Dot Matrix Display* P10 yang disusun kesamping dan 2 buah modul yang disusun kebawah (*Dot matrix* 32x64) yang dioperasikan oleh Arduino UNO.

B. Rangkaian Elektrik *Dot Matrix Display* P10

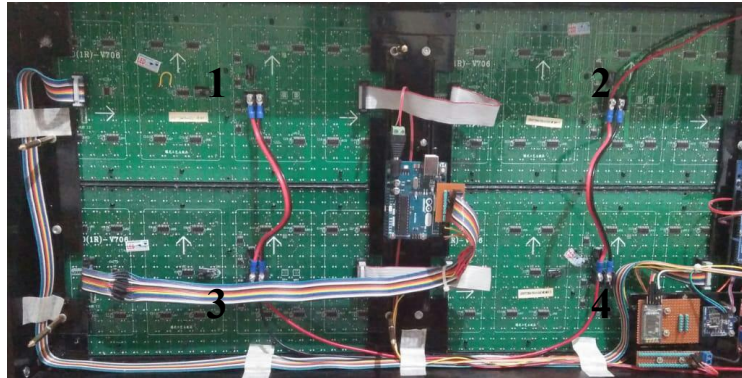
Gambar 4.18 di bawah ini merupakan rangkaian elektrik dari *Dot Matrix Display* P10 32 x 32 cm.



Gambar 4.18 Rangkaian Elektrik *Dot Matrix Display* P10 32 x 32 cm

Pada gambar 4.18 di atas, dapat dilihat bahwa *Dot Matrix Display* P10 yang diberi nomor 1 dihubungkan dengan *Dot Matrix Display* P10 yang diberi nomor 2. Kedua *Dot Matrix Display* tersebut dikontrol dengan Arduino Uno. Arduino Uno juga dihubungkan ke RTC untuk menampilkan waktu dan tanggal.

Gambar 4.19 di bawah ini merupakan rangkaian elektrik dari *Dot Matrix Display* P10 64 x 32 cm.



Gambar 4.19 Rangkaian Elektrik *Dot Matrix Display P10 64 x 32 cm*

Pada gambar 4.19 di atas, dapat dilihat bahwa *Dot Matrix Display P10* yang diberi nomor 4 dihubungkan dengan *Dot Matrix Display P10* yang diberi nomor 1. Selanjutnya, *Dot Matrix Display P10* yang diberi nomor 1 dihubungkan ke nomor 2 dan *Dot Matrix Display P10* yang diberi nomor 3 dihubungkan ke Arduino Mega. Keempat *Dot Matrix Display* tersebut dikontrol oleh Arduino Mega. Arduino Mega juga dihubungkan ke *Bluetooth HC 05* untuk menampilkan menerima informasi yang dikirim oleh *Smartphone OS Android* dan dihubungkan ke kabel USB untuk menerima informasi dari Komputer OS Windows.

C. Pengujian *Dot Matrix Display P10*

Pengujian yang dilakukan pada *Dot Matrix Display P10* dilakukan dengan cara menampilkan teks yang telah dibuat pada program Arduino. Pengujian *Dot Matrix Display P10* dibagi menjadi 2, yaitu pengujian *Dot Matrix Display P10 32x32* yang terkoneksi dengan Arduino Mega 2560 dan pengujian *Dot Matrix Display P10 32x64* yang terkoneksi dengan Arduino UNO. Adapun *list* program yang digunakan dalam pengujian *Dot Matrix Display P10 32x32* adalah sebagai berikut.

```
#include <SPI.h>
#include <DMD2.h>
#include <DejaVuSansBold9.h>

SoftDMD dmd(1,2);
DMD_TextBox box(dmd);
```

```

void setup() {
  Serial.begin(9600);
  dmd.selectFont(DejaVuSansBold9);
  dmd.begin();
  dmd.drawString(9,6,"PA");
  dmd.drawString(4,18,"2018");
}
void loop() {}

```

} Menampilkan String

Hasil dari pengujian *Dot Matrix Display P10 32x32* dapat dilihat pada gambar 4.20 berikut.



Gambar 4.20 Hasil Pengujian DMD P10 32x32

Sedangkan untuk *list* program yang digunakan dalam pengujian *Dot Matrix Display P10 32x64* adalah sebagai berikut.

```

#include <SPI.h>
#include <DMD.h>
#include <TimerOne.h>
#include "Arial_Black_16_ISO_8859_1.h"

DMD dmd(2, 2);
void ScanDMD(){dmd.scanDisplayBySPI();}
void setup(void){
  Timer1.initialize( 5000 );
  Timer1.attachInterrupt( ScanDMD );
  dmd.selectFont(Arial_Black_16_ISO_8859_1);
  dmd.drawString(4,7,"PA 2018",8,GRAPHICS_NORMAL,true);
}
void loop(){}

```

} Menampilkan String

Hasil dari pengujian *Dot Matrix Display* P10 32x64 dapat dilihat pada gambar 4.21 berikut.



Gambar 4.21 Hasil Pengujian DMD P10 32x64

4.4 Software Papan Informasi Berbasis *Dot Matrix*

Software yang akan diterapkan pada Papan Informasi Berbasis *Dot Matrix* meliputi aplikasi yang akan digunakan, pemrograman Mikrokontroler Arduino Mega 2560 dan Arduino UNO.

4.4.1 Aplikasi SiPETRIX

Aplikasi yang dibuat pada Android dan Komputer dilengkapi dengan *user name* dan *password* sehingga tidak semua orang dapat mengakses Papan Informasi Berbasis *Dot Matrix* yang telah dibuat. Selain itu aplikasi juga dirancang agar *user* dapat memilih ukuran font yang telah disediakan. Aplikasi SiPETRIX pada Android dibuat menggunakan Software App Inventor sedangkan Aplikasi SiPETRIX pada komputer dibuat menggunakan software Microsoft Visual Basic.

Untuk mengoperasikan aplikasi yang telah dibuat, *user* diminta untuk memasukan *user name* dan *password* kemudian memilih jenis font yang telah disediakan, memilih *screen* yang digunakan, setelah itu baru *user* dapat menulis teks yang akan dikirim ke Arduino melalui koneksi *bluetooth* untuk Android dan kabel USB untuk Komputer.

A. Pembuatan Aplikasi SiPETRIX

Pembuatan aplikasi SiPETRIX pada Android dilakukan menggunakan *software* App Inventor secara *online* menggunakan Komputer. Pembuatan aplikasi menggunakan App Inventor ini dibagi menjadi 2 komponen utama yaitu *Designer* dan *Block Editor*. Adapun hasil dari aplikasi yang telah dibuat pada Android dapat dilihat pada gambar 4.22 berikut ini.

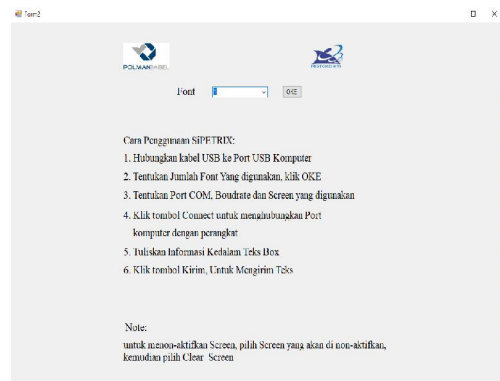


Gambar 4.22 Hasil Pembuatan Aplikasi SiPETRIX pada Android

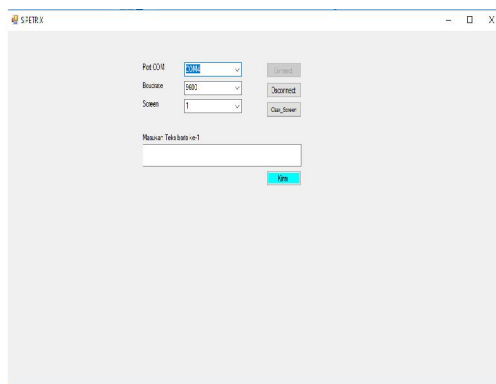
Sedangkan pembuatan aplikasi SiPETRIX pada Komputer dilakukan menggunakan *software* Microsoft Visual Basic. Pembuatan aplikasi menggunakan Microsoft Visual Basic ini dibagi menjadi 2 komponen utama yaitu *Desin* dan *code* program. Adapun hasil dari aplikasi SiPETRIX yang telah dibuat menggunakan Microsoft Visual Basic dapat dilihat pada gambar 4.23 berikut.



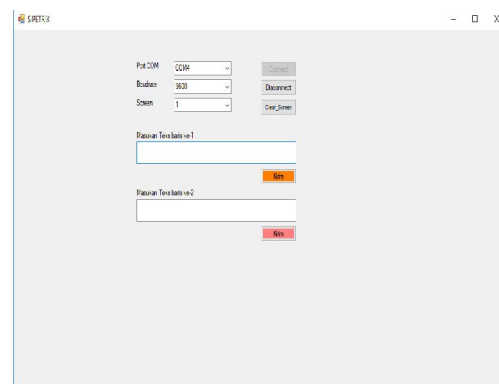
Gambar 4.23a Tampilan 1



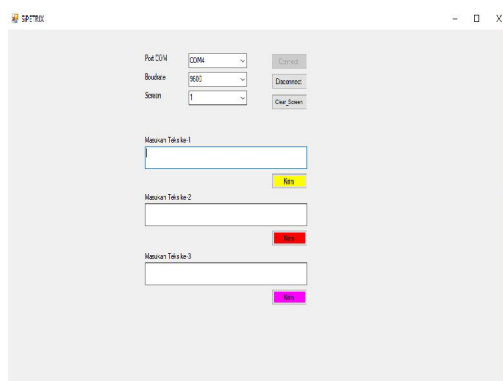
Gambar 4.23b Tampilan 2



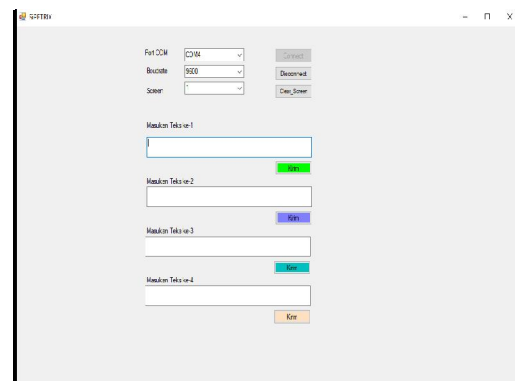
Gambar 4.23c Tampilan 3a



Gambar 4.23d Tampilan 3b



Gambar 4.23e Tampilan 3c



Gambar 4.23f Tampilan 3d

B. Pengujian Aplikasi SiPETRIX

Pengujian aplikasi SiPETRIX bertujuan mengetahui apakah aplikasi dapat mengirim teks informasi sesuai dengan format yang diinginkan. Pengujian dilakukan dalam dua tahapan. Tahap pertama dilakukan pada aplikasi SiPETRIX yang ada pada Android dan tahap kedua dilakukan pada aplikasi SiPETRIX yang

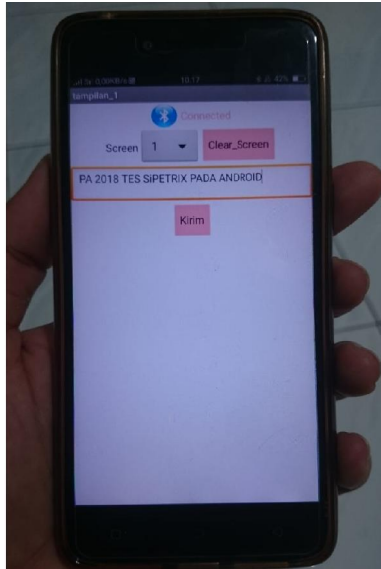
terdapat pada Komputer. Adapun langkah-langkah yang dilakukan dalam pengujian Aplikasi yang ada pada Android yaitu sebagai berikut.

1. Hidupkan *Bluetooth* pada perangkat Android.
2. Buka aplikasi SiPETRIX.
3. Masukkan *username* dan *password*. Tekan tombol Oke.
4. Tentukan jenis font yang digunakan.
5. Hubungkan *Bluetooth* Android dengan *Bluetooth* yang ada pada perangkat *Dot Matrix*.
6. Tunggu hingga *Bluetooth* sudah terhubung (*connected*).
7. Tentukan *screen* yang akan menampilkan teks informasi yang dikirim.
8. Tulis teks kedalam kotak teks yang telah disediakan, kemudian tekan tombol Kirim.

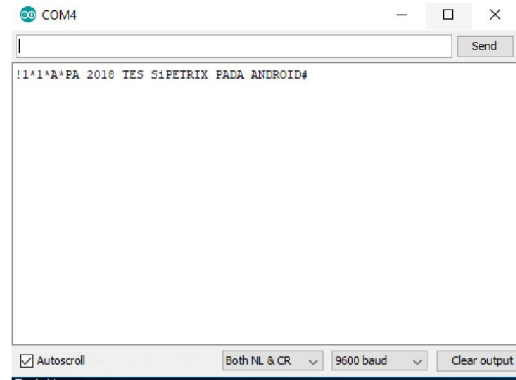
Sedangkan langkah-langkah yang dilakukan dalam pengujian Aplikasi yang ada pada Komputer yaitu sebagai berikut.

1. Hubungkan kabel USB ke Port USB komputer.
2. Buka aplikasi SiPETRIX.
3. Masukkan *username* dan *password*. Tekan tombol Oke.
4. Tentukan jenis font yang digunakan.
5. Tentukan Port COM, boudrate, dan *screen*.
6. Klik tombol Connect untuk menghubungkan Port Komputer dengan perangkat.
7. Tuliskan teks kedalam kotak teks, kemudian klik tombol Kirim.

Aplikasi nantinya tidak hanya mengirim teks yang ditulis pada kotak teks, tetapi juga mengirimkan data berupa jenis font, *screen*, dan kotak teks yang digunakan, sehingga format pengiriman data dari aplikasi adalah “!*screen**JenisFont*KotakTeks*TeksInformasi# “. Berikut adalah hasil pengujian terhadap Aplikasi SiPETRIX pada Android yang ditampilkan pada serial monitor yang dapat dilihat pada gambar 4.24 berikut.

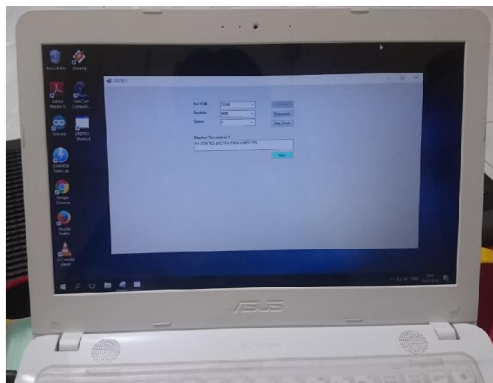


Gambar 4.24a Mengirim teks dari Android

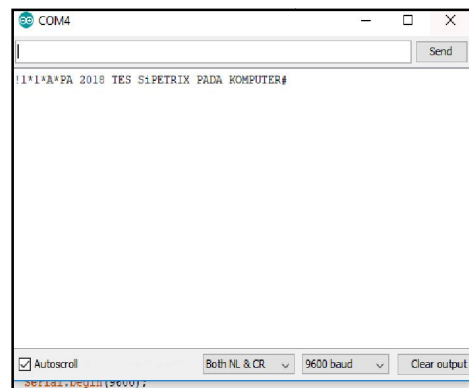


Gambar 4.24b Data yang diterima Arduino

Hasil pengujian Aplikasi SiPETRIX pada Komputer ditunjukkan pada gambar 4.25 berikut.



Gambar 4.25a Mengirim teks dari Windows



Gambar 4.25b Data yang diterima Windows

Dari hasil pengujian yang dilakukan, Aplikasi dapat mengirim data sesuai dengan yang diharapkan. Data yang dikirim ke Arduino nantinya akan dilolah agar dapat ditampilkan pada *Dot Matrix*.

4.4.2 Program Arduino Mega 2560

Pembuatan program dilakukan menggunakan *software* Arduino IDE 1.8.4. Pemrograman *dot matrix* 32x32 menggunakan Arduino MEGA 2560 dibagi menjadi beberapa tahapan yaitu sebagai berikut:

- Pemrograman menerima data serial dari *bluetooth* HC-05 dan kabel USB.
- Pemrograman pengiriman data serial ke Arduino UNO.
- Pemrograman modul RTC DS1307 untuk sumber data pewaktu
- Pemrograman *Dot Matrix* P10 untuk menampilkan informasi.
- Pemrograman pembuatan Logo POLMAN BABEL.
- Pemrograman Papan Informasi Berbasis *Dot Matrix* secara keseluruhan.

4.4.3 Program Arduino UNO

Pembuatan program dilakukan menggunakan *software* Arduino IDE 1.8.4. Pemrograman *dot matrix* 32x64 menggunakan Arduino UNO dibagi menjadi beberapa tahapan sesuai dengan perancangan yang telah dibuat yaitu sebagai berikut:

- Pemrograman menerima data serial dari Arduino MEGA 2560.
- Pemrograman parsing data.
- Pemrograman *Dot Matrix* 32x64 untuk menampilkan informasi.
- Pemrograman animasi teks.
- Pemrograman menyimpan data pada EEPROM.

4.5 Pengujian Papan Informasi Berbasis *Dot Matrix*

Setelah bagian *hardware* dan *software* dari Papan Informasi Berbasis *Dot Matrix* telah dibuat, proses selanjutnya adalah pengujian terhadap keseluruhan sistem. Pengujian sistem ini dilakukan untuk menguji serta melihat bagaimana hasil akhir yang didapat setelah melakukan perancangan, pembuatan serta pengujian masing-masing *hardware* maupun *software*. Adapun hasil Papan Informasi Berbasis *Dot Matrix* yang telah dibuat dapat dilihat pada gambar 4.26 berikut.



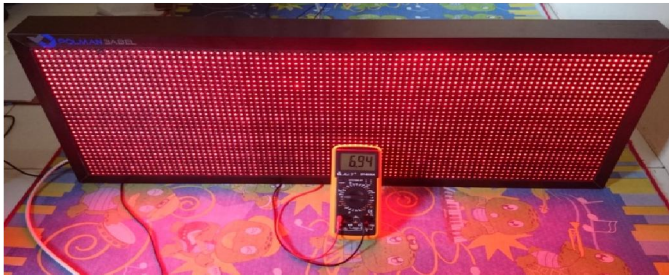
Gambar 4.26 Hasil Akhir Papan Informasi Berbasis *Dot Matrix*

Percobaan dilakukan dengan tahapan sebagai berikut :

1. Hidupkan Power Supply.
2. *Input* teks informasi yang ingin ditampilkan melalui Android atau Komputer
3. Amati teks yang ditampilkan pada *Dot Matrix*. Text yang ditampilkan harus sama dengan teks yang *diinputkan*.
4. Ganti tipe ukuran font pada aplikasi dan amati perubahan yang terjadi pada *Dot Matrix*, tinggi font yang ditampilkan harus sesuai dengan tipe font yang telah dipilih pada aplikasi.
5. Gunakan tipe ukuran font sangat kecil untuk dapat mengetahui apakah *Dot Matrix* dapat menampung 4 baris teks.
6. Matikan dan dihidupkan kembali Papan Informasi Berbasis *Dot Matrix*, amati apakah tampilan teks informasi masih ada atau tidak. *Dot Matrix* harus tetap menampilkan teks informasi yang sama dengan sebelum perangkat dimatikan.
7. Kemudian, amati jam ketika dihidupkan kembali apakah masih tetap melanjutkan hitungan sebelumnya atau tidak. Tampilan jam harus tetap menghitung meskipun tidak ada sumber listrik.
8. Kemudian tes jarak *bluetooth* yang dapat mengirim informasi ke *LED Dot Matrix*.
9. Lakukan pemeriksaan kondisi *frame LED Dot Matrix*. untuk mengetahui kerapian dari *frame* tersebut.

4.5.1 Pengujian Konsumsi Arus Pada *Dot Matrix*

Pengujian konsumsi arus pada *Dot Matrix* berfungsi untuk menentukan berapa daya yang dibutuhkan ketika semua LED pada *Dot Matrix* sedang aktif. Tegangan yang digunakan yaitu 5V, arus yang didapat dari hasil pengukuran yaitu 6.94 A. Adapun hasil pengujian ditunjukkan pada gambar 4.27 berikut.



Gambar 4.27a Hasil Pengujian Arus *Dot Matrix*



Gambar 4.27b Arus Terukur pada *Dot Matrix*

Untuk mengetahui daya dari *Dot Matrix* tersebut menggunakan perhitungan sebagai berikut:

$$P = V \cdot I = 5 \text{ V} \cdot 6,94 \text{ A} = 34,7 \text{ VA}$$

Hasil perhitungan daya dapat dilihat pada tabel 4.7 berikut.

Tegangan (V)	Arus (A)	Daya (VA)
5	6,94	34,7

Pada 6 blok *Dot Matrix Display* sendiri, daya yang dibutuhkan yaitu 34.7 VA. Hal ini didapat dari perhitungan dengan mengalikan tegangan dan arus.

4.5.2 Pengujian Menampilkan Waktu Pada *Dot Matrix 32x32*

Dot Matrix 32x32 akan menampilkan waktu apabila pin 4 pada Arduino Mega 2560 berlogika LOW. Adapun hasil dari pengujian waktu pada *Dot Matrix 32x32* dapat dilihat pada gambar 4.28 dan tabel 4.8 berikut.



Gambar 4.28a Pengujian ke-1 Menampilkan Waktu



Gambar 4.28b Pengujian ke-2 Menampilkan Waktu



Gambar 4.28c Pengujian ke-3 Menampilkan Waktu

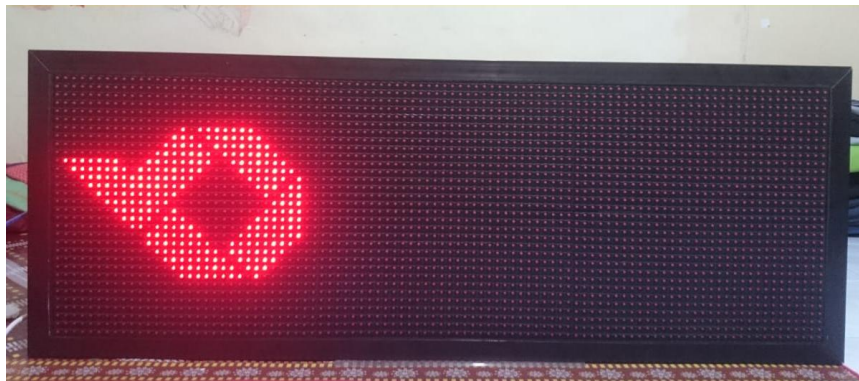
Tabel 4.8 Hasil Pengujian Menampilkan Waktu

No	Dot Matrix	HP	Error (menit)
1.	19:53, 07-08-2018	19:53, 07-08-2018	0
2.	20:33, 07-08-2018	20:33, 07-08-2018	0
3.	21:14, 07-08-2018	21:14, 07-08-2018	0

4.5.3 Pengujian Menampilkan Logo Pada *Dot Matrix* 32x32

Dot Matrix 32x32 akan menampilkan logo apabila pin 4 pada Arduino Mega 2560 berlogika HIGH. Logo POLMAN BABEL pada *Dot Matrix* dibuat dengan cara memanfaatkan intruksi “ `dmd.setPixel(x , y, GRAPHICS_ON);` ”. Intruksi ini digunakan untuk menhidupkan satu buah pixel pada *Dot Matrix*. Adapun program yang digunakan untuk membuat logo POLMAN BABEL dilampirkan.

Adapun hasil dari logo yang telah dibuat berdasarkan *list* program diatas dapat lihat pada gambar 4.29 berikut.



Gambar 4.29 Hasil Pengujian Menampilkan Logo

4.5.4 Pengujian Menampilkan Font Pada *Dot Matrix* 32x64

Dot Matrix 32x64 akan menampilkan teks dengan ukuran tinggi font sesuai dengan jenis font yang dipilih oleh *user* berdasarkan pilihan yang telah disediakan pada aplikasi SiPETRIX. Jenis ukuran font yang disediakan sebanyak empat jenis.

Pembuatan pilihan font yang telah disediakan dilakukan dengan cara mencari dan mencoba berbagai jenis pustaka font yang telah disediakan oleh para pengembang *Dot matrix* P10. Setelah mencari dan mencoba berbagai jenis pustaka font, diambil 4 jenis font dengan ukuran tinggi yang berbeda dan dikategorikan sebagai ukuran sangat besar, besar, kecil, dan sangat kecil.

Pengujian dilakukan dengan cara menampilkan huruf “ j “, kemudian huruf tersebut diukur ketinggiannya menggunakan penggaris. Huruf “ j “ dipilih karena merupakan huruf tertinggi dalam setiap jenis font yang digunakan.

A. Font Sangat Besar

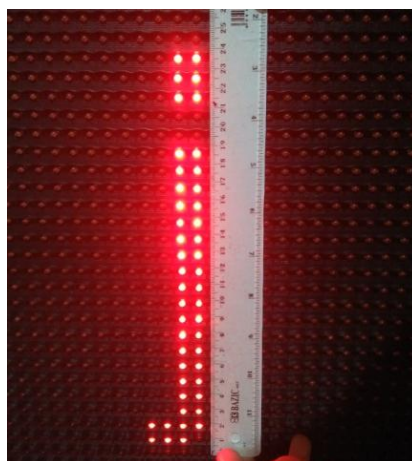
Ukuran font sangat besar menggunakan jenis pustaka font “Droid_Sans_24.h” dimana tinggi ukuran keseluruhan font ini setinggi 24cm, sehingga menyebabkan Papan Informasi Berbasis *Dot Matrix* yang telah dibuat hanya mampu menampung satu baris teks. Pengujian dilakukan dengan menggunakan *list* program berikut.

```
#include <SPI.h>
#include <DMD.h>
#include <TimerOne.h>
#include <Droid_Sans_24.h>
DMD dmd(2, 2);
void ScanDMD() {dmd.scanDisplayBySPI();}
void setup(void)
{
  Timer1.initialize( 5000 );
  Timer1.attachInterrupt( ScanDMD );
  dmd.selectFont(Droid_Sans_24);
  dmd.drawString(0,3,"j",1,GRAPHICS_NORMAL);
}
void loop(void){}
```

} Pustaka Font yang digunakan

} Memilih Font yang digunakan

Hasil pengukuran jenis font sangat besar dapat dilihat pada gambar 4.30 berikut.



Gambar 4.30 Hasil Pengukuran Font Sangat Besar

B. Font Besar

Ukuran font besar menggunakan jenis pustaka font “Arial_black_16.h” dimana tinggi ukuran font ini adalah 15cm, sehingga menyebabkan Papan Informasi Berbasis *Dot Matrix* yang telah dibuat hanya mampu menampung dua baris teks. Pengujian dilakukan dengan menggunakan *list* program berikut.

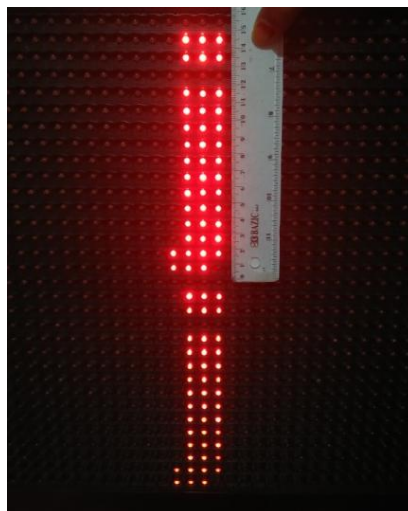
```
#include <SPI.h>
#include <DMD.h>
#include <TimerOne.h>
#include <Arial_black_16.h>

DMD dmd(2, 2);
void ScanDMD() {dmd.scanDisplayBySPI();}
void setup(void)
{
  Timer1.initialize( 5000 );
  Timer1.attachInterrupt( ScanDMD );
  dmd.selectFont(Arial_black_16);
  dmd.drawString(0,0,"j",1,GRAPHICS_NORMAL);
  dmd.drawString(0,16,"j",1,GRAPHICS_NORMAL);
}
void loop(void){}
```

} Pustaka Font
yang digunakan

} Memilih Font
yang digunakan

Hasil pengujian font besar dapat dilihat pada gambar 4.31 berikut.



Gambar 4.31 Hasil Pengukuran Font Besar

C. Font Kecil

Ukuran font kecil menggunakan jenis pustaka font “DejaVuSansBold9.h” dimana tinggi ukuran font ini adalah 10cm, sehingga Papan Informasi Berbasis *Dot Matrix* yang telah dibuat dapat menampung tiga baris teks. Pengujian dilakukan dengan menggunakan *list* program berikut.

```
#include <SPI.h>
#include <DMD.h>
#include <TimerOne.h>
#include <DejaVuSansBold9.h>
```

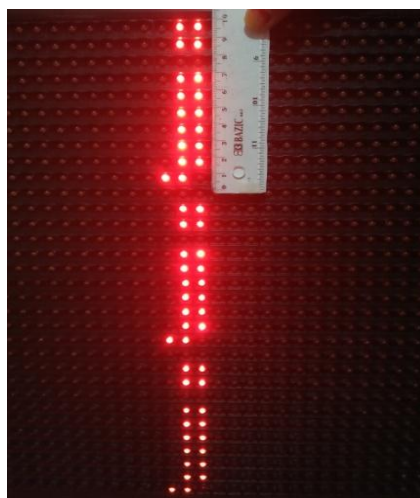
} Pustaka Font yang digunakan

```
DMD dmd(2, 2);
void ScanDMD(){dmd.scanDisplayBySPI();}
void setup(void)
{
  Timer1.initialize( 5000 );
  Timer1.attachInterrupt( ScanDMD );

  dmd.selectFont(DejaVuSansBold9);
  dmd.drawString(0,0,"j",1,GRAPHICS_NORMAL );
  dmd.drawString(0,11,"j",1,GRAPHICS_NORMAL );
  dmd.drawString(0,22,"j",1,GRAPHICS_NORMAL );
}
void loop(void){}
```

} Memilih Font yang digunakan

Hasil pengujian font besar dapat dilihat pada gambar 4.32 berikut.



Gambar 4.32 Hasil Pengukuran Font Kecil

D. Font Sangat Kecil

Ukuran font sangat kecil menggunakan jenis pustaka font “SystemFont5x7.h” dimana tinggi ukuran font ini adalah 7cm, sehingga Papan Informasi Berbasis *Dot Matrix* yang telah dibuat dapat menampung empat baris teks. Pengujian dilakukan dengan menggunakan *list* program berikut.

```
#include <SPI.h>
#include <DMD.h>
#include <TimerOne.h>
#include <SystemFont5x7.h>
```

} Pustaka Font
yang digunakan

```
DMD dmd(2, 2);
void ScanDMD() {dmd.scanDisplayBySPI();}
void setup(void)
{
  Timer1.initialize( 5000 );
  Timer1.attachInterrupt( ScanDMD );
  dmd.selectFont(SystemFont5x7);
  dmd.drawString(0,0,"j",1,GRAPHICS_NORMAL );
  dmd.drawString(0,8,"j",1,GRAPHICS_NORMAL );
  dmd.drawString(0,16,"j",1,GRAPHICS_NORMAL );
  dmd.drawString(0,24,"j",1,GRAPHICS_NORMAL );
}
void loop(void){}
```

} Memilih Font
yang digunakan

Hasil pengujian font besar dapat dilihat pada gambar 4.33 berikut.



Gambar 4.33 Hasil Pengukuran Font Sangat Kecil

Dari Pengujian font yang telah dilakukan, dapat diketahui hasil dari pengujian tersebut yang terdapat pada tabel 4.9 berikut.

Tabel 4.9 Hasil Pengujian Font

Jenis Font	Ukuran (cm)	Jumlah Maksimal Baris
Sangat Besar	24	1
Besar	15	2
Kecil	10	3
Sangat Kecil	7	4

4.5.5 Pengujian Pengiriman dan Menampilkan Teks ke *Dot Matrix* 32x64

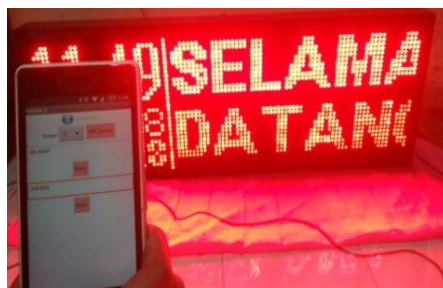
Pengujian ini adalah pengujian terakhir pada alat, dimana pengujian ini dilakukan dengan cara mengirimkan teks dari aplikasi SiPETRIX yang ada pada Android maupun Laptop ke perangkat *Dot Matrix*. Hasil dari pengujian dapat dilihat pada gambar dibawah ini.

- Hasil pengujian mengirim satu baris teks menggunakan Windows dapat dilihat pada gambar 4.34 berikut.



Gambar 4.34 Hasil Pengiriman 1 baris Teks

- Hasil pengujian mengirim dua baris teks menggunakan Android dapat dilihat pada gambar 4.35 berikut.



Gambar 4.35 Hasil Pengiriman 2 baris teks

- Hasil pengujian mengirim tiga baris teks menggunakan Windows dapat dilihat pada gambar 4.36 berikut.



Gambar 4.36 Hasil Pengiriman 3 baris Teks

- Hasil pengujian mengirim empat baris teks menggunakan Android dapat dilihat pada gambar 4.37 berikut.



Gambar 4.37 Hasil Pengiriman 4 baris teks

Dari hasil pengujian yang telah dilakukan, yaitu menginput teks dari aplikasi SiPETRIX pada Android maupun Windows hingga teks tersebut dapat ditampilkan pada *Dot Matrix*, dapat diketahui bahwa alat yang dibuat pada Proyek Akhir ini dapat berfungsi dengan baik dan sesuai dengan yang diharapkan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisa terhadap fungsi alat pada Proyek Akhir dengan judul “Rancang Bangun Papan Informasi Berbasis *Dot Matrix* di BAAKPK POLMAN BABEL yang Terintegrasi dengan Android ” ini maka dapat ditarik kesimpulan sebagai berikut:

- a. Android, Windows dan Arduino terhubung dan berfungsi dengan baik.
- b. Teks yang ditulis pada Android / Windows sesuai dengan teks yang tampil pada LED *Dot Matrix*.
- c. Jumlah karakter yang ditampilkan pada *Dot Matrix* sebanyak 660 karakter yang terbagi kedalam beberapa *screen* dan kotak teks.
- d. Papan Informasi berbasis *Dot Matrix* yang telah dirakit berfungsi dengan baik dan sesuai dengan yang diharapkan oleh Mitra.

5.2 Saran

Apabila alat ini akan dikembangkan lebih lanjut, fungsi yang perlu diperbaiki dan ditambahkan antara lain :

- a. Proses penulisan *dot matrix* dapat diakses melalui *Website*.
- b. Lakukan perawatan alat secara berkala agar alat dapat bertahan lama.

DAFTAR PUSTAKA

- [1] Azhari Rahman dan Yudiansyah, “Kontrol Sistem Informasi *Running Text* Menggunakan Arduino”, Proyek Akhir, Politeknik Manufaktur Negeri Bangka Belitung, 2017.
- [2] Sabri Rusli, “*Running Text* Menggunakan Keyboard AT PS/2 Berbasis Mikrokontroler Atmega8535”, Skripsi, Universitas Muhammadiyah Purwokerto, 2011.
- [3] Khamdani, “Rancang Bangun *Running Text* Led Display Berbasis Mikrokontroler sebagai Media Informasi di Universitas Muhammadiyah Purwokerto”, Skripsi, Universitas Muhammadiyah Purwokerto, 2014.
- [4] Evan Taruna Setiawan, “Pengendalian Lampu Rumah Berbasis Mikrokontroler Arduino Menggunakan Smartphone Android”, *Jurnal TI-Atma STMIK Atma Luhur Pangkalpinang*, vol. 2, no. 1, pp. 1-8, 2014.
- [5] Ardilas, Sejarah dan Versi Android [Online], diakses pada 09 Mei 2018, Available : <http://www.ardilas.com/2014/09/apakah-pengertian-arti-maksud -dari-tentang-sejarah-versi-kata-android-itu-adalah.html>.
- [6] Gufron Rajo Kaciak, Pengenalan Sistem Operasi Microsoft Windows[Online], diakses pada 10 Mei 2018, Available : <http://dosen.gufron.com/artikel/pengenalan-sistem-operasi-microsoft-windows/12/>.
- [7] Toto Haryanto, App Inventor [Online], diakses pada 10 Juni 2018, Availbale : <https://www.totoharyanto.com/apa-sih-app-inventor-itu/>.
- [8] Teknisi Muda Handal, Manfaat Kelebihan dan Perkembangan Microsoft Visual Basic [Online], diakses pada 10 juni 2018, Avaibale : <https://www.teknisimudahandal.com/2017/10/pengertian-manfaat-dan-kelebihan.html>.
- [9] Arduino, Arduino UNO Rev3 SMD [Online], diakses pada tanggal 11 Mei 2018, Available : <https://store.arduino.cc/usa/arduino-uno-smd-rev3>.
- [10] Arduino Info, Arduino Mega 2560 [Online], diakses pada tanggal 18 April 2017, Available : <https://arduino-info.wikispaces.com/MegaQuickRef>.
- [11] Rajguruelectronics, Bluetooth Module [Online], diakses pada 11 Mei 2018, Available : <http://www.rajguruelectronics.com/bluetoothmodule.html>.
- [12] Ignorae, Serial UART [Online], diakses pada 11 Mei 2018, Available : <http://ingorae.tistory.com/1427/serial-uart/>.
- [13] Tutor Keren, Menggunakan Real Time Clock (RTC) pada Arduino [Online], diakses pada 11 Mei 2018, Available : <https://tutorkeren.com/artikel/tutorial-menggunakan-real-time-clock-rtc-pada-arduino.htm>.

- [14] Anwar Mujadin, "Sistem Proteksi *Power Supply* Modul Praktikum Teknik Digital," *Jurnal AL-AZHAR INDONESIA SERI SAINS DAN TEKNOLOGI*, Vol . 2, No. 3, pp. 156 – 163, 2014.

LAMPIRAN

LAMPIRAN 1

DAFTAR RIWAYAT HIDUP

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama lengkap : Dina Forsa Anggraini
Tempat dan tanggal lahir : Tempilang, 15 Agustus 1997
Alamat rumah : Tempilang Utara II RT 006/RW 002
Kecamatan Tempilang
Telp : -
Hp : 0813 1062 7216
Email : dinaocha.forsa@gmail.com
Jenis kelamin : Perempuan
Agama : Islam



2. Riwayat Pendidikan

SDN 2	Tempilang	2003-2009
SMPN 1	Tempilang	2009-2012
SMAN 1	Tempilang	2012-2015

3. Pendidikan Non Formal

-

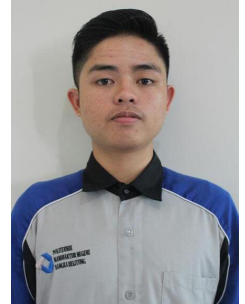
Sungailiat, 2 Agustus 2018

Dina Forsa Anggraini

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama lengkap : Restu Pangestu
Tempat dan tanggal lahir : Air Mesu, 18 September 1997
Alamat rumah : Desa Air Mesu RT 007/RW 000
Kecamatan Pangkalan Baru
Telp : -
Hp : 0857 5836 1762
Email : pangestu_bangka@gmail.com
Jenis kelamin : Laki-laki
Agama : Islam



2. Riwayat Pendidikan

SDN 12	Lubuk Besar	2003-2009
SMPN 2	Namang	2009-2012
SMK 2	Pangkal Pinang	2012-2015

3. Pendidikan Non Formal

-

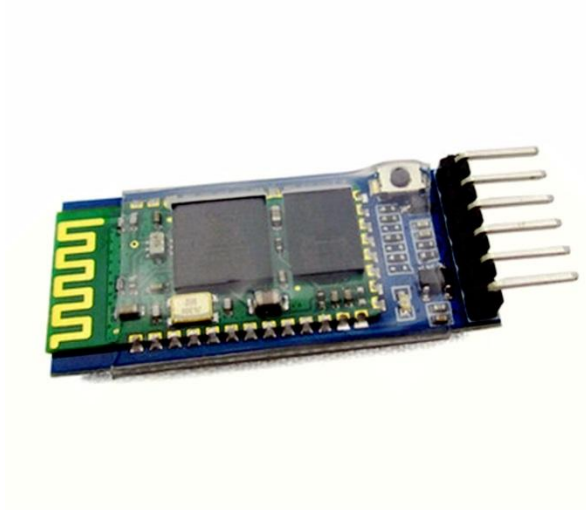
Sungailiat, 2 Agustus 2018

Restu Pangestu

LAMPIRAN 2

***DATASHEET BLUETOOTH HC-05, DATASHEET
RTC DS1307, DATASHEET MODUL LED DOT
MATRIX P10***

BLUETOOTH HC-05 SPECIFICATIONS



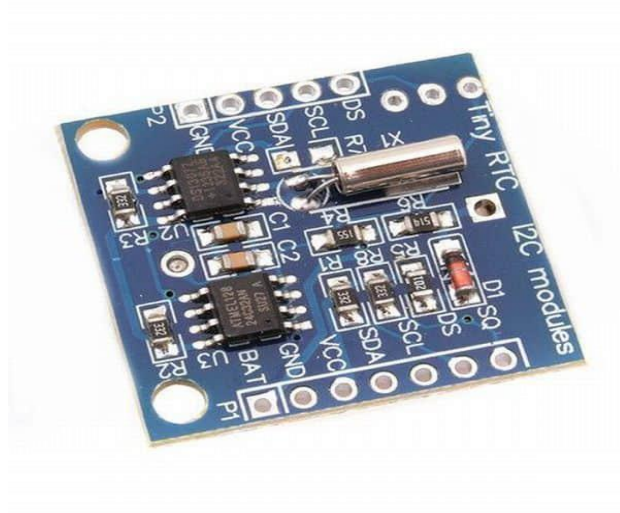
Specifications:

- Bluetooth protocol: Bluetooth Specification v2.0+EDR
- Frequency: 2.4GHz ISM band
- Modulation: GFSK(Gaussian Frequency Shift Keying)
- Emission power: =4dBm, Class 2
- Sensitivity: =-84dBm at 0.1% BER
- Speed: Asynchronous: 2.1Mbps(Max) / 160 kbps, Synchronous: 1Mbps/1Mbps
- Security: Authentication and encryption
- Profiles: Bluetooth serial port
- Power supply: +3.3VDC 50mA
- Working temperature: -20 ~ +75 Centigrade
- Dimension: 26.9mm x 13mm x 2.2 mm

Application:

- Computer and peripheral devices
- GPS receiver
- Industrial control
- MCU projects

RTC DS1307 SPECIFICATIONS



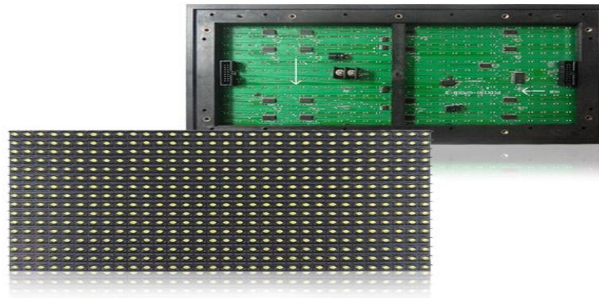
Specification

- Working voltage: 5v
- Two wire I2C interface
- Hour : Minutes : Seconds AM/PM
- Day Month, Date - Year
- Leap year compensation
- Accurate calendar up to year 2100
- Consumes Less than 500nA in Battery-Backup
- Battery included
- 1Hz output pin
- 56 Bytes of Non-volatile memory available to user
- provides 32,768 bits(4KB) of serial electrically erasable and programmable
- read only memory (EEPROM)
- Embed DS18B20 temperature sensor interface with the pull-up resistor
- Dimensions: 28mm x 25mm x 8mm

Application

- Real time monitoring system
- Timer

P10 LED RED MODULE



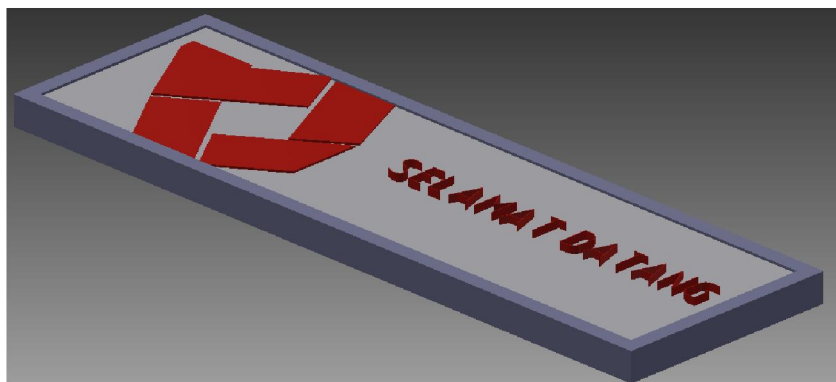
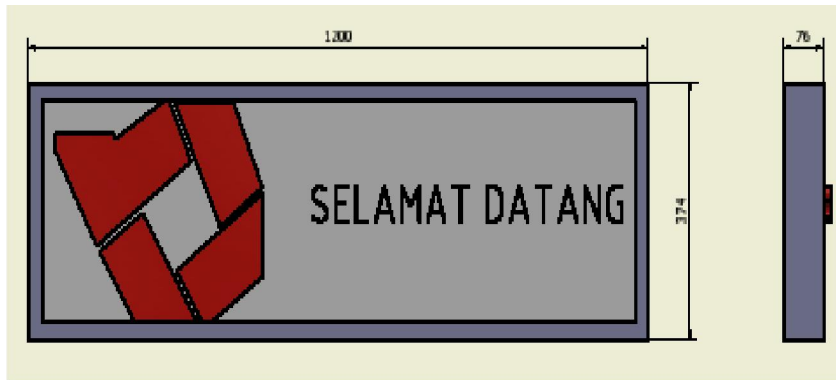
P10 DIP RED LED Module Parameters	
Model	P10
Pixel pitch	10 mm
Pixel configuration	1R
LED lamp	346 DIP
Pixel matrix per panel	32*16pixels
Module resolution rate	32*16=512dots
Module dimensions (W×H)	320*160mm
Module dimensions (W×H)	12.6*6.3 Inch
Weight	0.7kg±0.05kg
Package material	Polycarbonate
Construction	combined driver board
Input votage	4.8-5.5V
MAX. input current	3.4A±0.1A
Driving mode	1/4 constant votage scan
Module power	≤17.0 W
cabinet parameters	
Brightness	≥3500 cd/m2
Cabinet dimensions (W×H×D)	960*960*70 mm
Cabinet dimensions (W×H×D)	37.8*37.8*2.76 Inch
Resolution ratio	96*96=9216dots
Weight	33.5±0.05 kg
Weight	99.78±0.023 lbs
Pixel density	3906 dots/m2
Viewing angle (H/V)	80±15 deg
Best viewing distance	≥10m
MAX. Power	≤400W/m2
AVE. Power	200W/m2
control parameters	
Gray scale per color	0-128 to optional
Colors	256
Life time	≥100,000 hours

MTBF	≥10,000 hours
Refresh rate	≥ 75Hz
MAX. Power	1000 W/m ²
Input voltage (AC)	110V ~ 240V
Operating temperature	-20°C ~+ 50°C
Operating humidity	10% ~ 90%

LAMPIRAN 3

GAMBAR KONSTRUKSI PAPAN

INFORMASI BERBASIS *DOT MATRIX*



LAMPIRAN 4

PROGRAM PADA ARDUINO MEGA 2560

DAN ARDUINO UNO

PROGRAM MEMBUAT LOGO POLMAN

```
int x,y;
dmd.setPixel(15+a, 1+b, GRAPHICS_ON);
for(x=14; x<17; x++){dmd.setPixel(x+a, 2+b, GRAPHICS_ON);}
for(x=13; x<18; x++){dmd.setPixel(x+a, 3+b, GRAPHICS_ON);}
for(x=12; x<19; x++){dmd.setPixel(x+a, 4+b, GRAPHICS_ON);}
for(x=1; x<8; x++) {dmd.setPixel(x+a, 5+b, GRAPHICS_ON);}
for(x=11; x<19; x++){dmd.setPixel(x+a, 5+b, GRAPHICS_ON);}
for(x=2; x<9; x++) {dmd.setPixel(x+a, 6+b, GRAPHICS_ON);}
for(x=10; x<18; x++){dmd.setPixel(x+a, 6+b, GRAPHICS_ON);}
for(x=3; x<17; x++) {dmd.setPixel(x+a, 7+b, GRAPHICS_ON);}
for(x=4; x<16; x++) {dmd.setPixel(x+a, 8+b, GRAPHICS_ON);}
for(x=5; x<15; x++) {dmd.setPixel(x+a, 9+b, GRAPHICS_ON);}
for(x=6; x<14; x++) {dmd.setPixel(x+a, 10+b, GRAPHICS_ON);}
for(x=7; x<13; x++) {dmd.setPixel(x+a, 11+b, GRAPHICS_ON);}
for(x=8; x<12; x++) {dmd.setPixel(x+a, 12+b, GRAPHICS_ON);}
for(x=9; x<11; x++) {dmd.setPixel(x+a, 13+b, GRAPHICS_ON);}

for(x=17; x<24; x++){dmd.setPixel(x+a, 1+b, GRAPHICS_ON);}
for(x=18; x<25; x++){dmd.setPixel(x+a, 2+b, GRAPHICS_ON);}
for(x=19; x<26; x++){dmd.setPixel(x+a, 3+b, GRAPHICS_ON);}
for(x=20; x<27; x++){dmd.setPixel(x+a, 4+b, GRAPHICS_ON);}
for(x=21; x<28; x++){dmd.setPixel(x+a, 5+b, GRAPHICS_ON);}
for(x=22; x<29; x++){dmd.setPixel(x+a, 6+b, GRAPHICS_ON);}
for(x=23; x<28; x++){dmd.setPixel(x+a, 7+b, GRAPHICS_ON);}
for(x=24; x<27; x++){dmd.setPixel(x+a, 8+b, GRAPHICS_ON);}
dmd.setPixel(25+a, 9+b, GRAPHICS_ON);

for(y=7; y<15; y++) {dmd.setPixel(29+a, y+b, GRAPHICS_ON);}
for(y=8; y<16; y++) {dmd.setPixel(28+a, y+b, GRAPHICS_ON);}
for(y=9; y<17; y++) {dmd.setPixel(27+a, y+b, GRAPHICS_ON);}
for(y=10; y<18; y++){dmd.setPixel(26+a, y+b, GRAPHICS_ON);}
for(y=11; y<19; y++){dmd.setPixel(25+a, y+b, GRAPHICS_ON);}
for(y=12; y<20; y++){dmd.setPixel(24+a, y+b, GRAPHICS_ON);}
for(y=13; y<19; y++){dmd.setPixel(23+a, y+b, GRAPHICS_ON);}
for(y=14; y<18; y++){dmd.setPixel(22+a, y+b, GRAPHICS_ON);}
for(y=15; y<17; y++){dmd.setPixel(21+a, y+b, GRAPHICS_ON);}

dmd.setPixel(14+a, 11+b, GRAPHICS_ON);
for(x=13; x<16; x++){dmd.setPixel(x+a, 12+b, GRAPHICS_ON);}
```

```
for(x=12; x<17; x++){dmd.setPixel(x+a, 13+b, GRAPHICS_ON);}
for(x=11; x<18; x++){dmd.setPixel(x+a, 14+b, GRAPHICS_ON);}
for(x=11; x<19; x++){dmd.setPixel(x+a, 15+b, GRAPHICS_ON);}
for(x=12; x<20; x++){dmd.setPixel(x+a, 16+b, GRAPHICS_ON);}
for(x=13; x<21; x++){dmd.setPixel(x+a, 17+b, GRAPHICS_ON);}
for(x=14; x<22; x++){dmd.setPixel(x+a, 18+b, GRAPHICS_ON);}
for(x=15; x<23; x++){dmd.setPixel(x+a, 19+b, GRAPHICS_ON);}
```


PROGRAM ARDUINO UNO

```
/*-----  
 * Project   : PAPAN INFORMASI BERBASIS DOTMATRIX DI BAAKPK  
POLMAN BABEL  
 * Author    : RESTU PANGESTU DAN DINA FORSA ANGGRAINI  
 * Ket       : PROYEK AKHIR 2018  
 *  
-----*/  
  
///library EEPROM  
#include <EEPROMex.h>  
#include <EEPROMVar.h>  
  
///library DMD P10  
#include <Arial_black_16.h>  
#include <DejaVuSansBold9.h>  
#include <Droid_Sans_24.h>  
#include <SystemFont5x7.h>  
#include <DMD.h>  
#include <SPI.h>    //SPI.h must be included as DMD is written by SPI (the  
IDE complains otherwise)  
#include <TimerOne.h> //  
  
//Total Panjang dan Lebar  
#define DISPLAYS_ACROSS 2  
#define DISPLAYS_DOWN 2  
DMD dmd(DISPLAYS_ACROSS, DISPLAYS_DOWN);  
  
//Deklarasi variable parsing data  
String dataIn;  
String dt[6];
```

```

int i;
boolean parsing=false;
byte ScreenIn;

//Deklarasi PIN logo POLMAN
const int logo = 4;

/*-----
Interrupt handler for Timer1 (TimerOne) driven DMD refresh scanning, this gets
called at the period set in Timer1.initialize();
-----*/
void ScanDMD()
{
  dmd.scanDisplayBySPI();
}

void setup()
{
  pinMode(logo,OUTPUT);

  //initialize TimerOne's interrupt/CPU usage used to scan and refresh the display
  Timer1.initialize( 5000 ); //period in microseconds to call ScanDMD.
Anything longer than 5000 (5ms) and you can see flicker.
  Timer1.attachInterrupt( ScanDMD ); //attach the Timer1 interrupt to
ScanDMD which goes to dmd.scanDisplayBySPI()

  //bersihkan(off) semua pixel
  dmd.clearScreen( true ); //true is normal (all pixels off), false is negative (all
pixels on)

  while (!Serial) {

```

```
    ; // wait for serial port to connect. Needed for native USB port only
  }
  Serial.begin(9600);
  dataIn="";
}
```

```
void loop()
```

```
{
  awal:
  if(Serial.available()>0)
  {
    char inChar = (char)Serial.read();
    dataIn += inChar;
    if (inChar == '#')
    {
      parsing = true;
    }
  }
}
```

```
if(Serial.available()>0){goto awal;}
```

```
if(parsing)
```

```
{
  parsingData();
  parsing=false;
  dataIn="";
}
```

```
tampilan_utama();
```

```
if(Serial.available()>0){goto awal;}
```

```
tampilan1();
```

```

if(Serial.available()>0){goto awal;}
tampilan2();
if(Serial.available()>0){goto awal;}
tampilan3();
}

```

```

////////////////////////////////////
////////////////////////////////////
//////////////////////////////////// < END OF MAIN
PROGRAM> //////////////////////////////////////
////////////////////////////////////
////////////////////////////////////

```

```

void parsingData()
{
    int j = 0;

    //kirim data yang telah diterima sebelumnya
    Serial.print("data masuk : ");
    delay(100);Serial.print(dataIn);
    Serial.print("\n\n");

    dt[j]=""; //inisialisasi variabel, (reset isi variabel)

    //proses parsing data
    for(i=1;i<dataIn.length();i++)
    {
        if ((dataIn[i] == '#') || (dataIn[i] == '*')) //pengecekan tiap karakter dengan
        karakter (#) dan (*)
        {

```

```

    j++; //increment variabel j, digunakan untuk
merubah index array penampung
    dt[j]=""; //inisialisasi variabel array dt[j]
}
else
{
    dt[j] += dataIn[i]; //proses tampung data saat pengecekan
karakter selesai.
}
delay(50);
}
ScreenIn=dt[0].toInt();
}

```

```

void tampilan_utama()
{
    int v,h;

    //TAMPILAN SELAMAT DATANG
    delay(200);Serial.println("tampilan utama");
    digitalWrite(logo,HIGH); //Logo polman ON
    dmd.clearScreen( true );
    dmd.selectFont(Droid_Sans_24);
    dmd.drawMarquee("SELAMAT DATANG", 14,(32*DISPLAYS_ACROSS)-1
,5);
    long start=millis();
    long timer=start;
    boolean ret=false;
    while(!ret)
    {

```

```

if ((timer+30) < millis())
{
    ret=dmd.stepMarquee(-1,0);
    timer=millis();
}
}

//TAMPILAN DIBAAKPK POLMAN BABEL
dmd.clearScreen( true );
dmd.selectFont(Arial_Black_16);
for( v=(DISPLAYS_DOWN*16); v>-1; v--)//masuk text 1
{
    dmd.drawString(1, v, "DI BAAKPK", 9, GRAPHICS_NORMAL );
    delay(50);
}
for( v=(DISPLAYS_DOWN*16); v>15; v--)//masuk text2
{
    dmd.drawString(1, v, "POLMAN BABEL", 12, GRAPHICS_NORMAL );
    delay(50);
}
delay(700);

for( h=0; h>-96; h--)//keluar text 1
{
    dmd.drawString(h, 0, "DI BAAKPK", 9, GRAPHICS_NORMAL );
    delay(30);
}
delay(300);
for( h=0; h>-138; h--)//keluar text2
{
    dmd.drawString(h, 16, "POLMAN BABEL", 12, GRAPHICS_NORMAL );
    delay(30);
}

```

```

    }

    digitalWrite(logo,LOW);    // logo polman OFF
}

void tampilan1()
{
    byte  Screan;
    byte  Rows;
    byte  jumlah1;
    byte  jumlah2;
    byte  jumlah3;
    byte  jumlah4;
    String kolom;
    char  Text1[56];
    char  Text2[56];
    char  Text3[56];
    char  Text4[56];

    Serial.println("tampilan1");

    if(ScreanIn==1)
    {
        Screan = ScreanIn;
        Rows  = dt[1].toInt();
        kolom = dt[2];
        EEPROM.write(0, Screan);
        EEPROM.write(1, Rows);
    }

    // 1 rows

```

```
if(Screen==1 && Rows==1 && kolom=="A")
{jumlah1=dt[3].length();dt[3].toCharArray(Text1,56);EEPROM.write(2,
jumlah1);EEPROM.writeBlock<char>( 6, Text1, jumlah1);}
```

```
//2 rows
```

```
else if(Screen==1 && Rows==2 && kolom=="B")
{jumlah1=dt[3].length();dt[3].toCharArray(Text1,56);EEPROM.write(2,
jumlah1);EEPROM.writeBlock<char>( 6, Text1, jumlah1);}
```

```
else if(Screen==1 && Rows==2 && kolom=="C")
{jumlah2=dt[3].length();dt[3].toCharArray(Text2,56);EEPROM.write(3,
jumlah2);EEPROM.writeBlock<char>( 62, Text2, jumlah2);}
```

```
//3 rows
```

```
else if(Screen==1 && Rows==3 && kolom=="D")
{jumlah1=dt[3].length();dt[3].toCharArray(Text1,56);EEPROM.write(2,
jumlah1);EEPROM.writeBlock<char>( 6, Text1, jumlah1);}
```

```
else if(Screen==1 && Rows==3 && kolom=="E")
{jumlah2=dt[3].length();dt[3].toCharArray(Text2,56);EEPROM.write(3,
jumlah2);EEPROM.writeBlock<char>( 62, Text2, jumlah2);}
```

```
else if(Screen==1 && Rows==3 && kolom=="F")
{jumlah3=dt[3].length();dt[3].toCharArray(Text3,56);EEPROM.write(4,
jumlah3);EEPROM.writeBlock<char>(118, Text3, jumlah3);}
```

```
//4 rows
```

```
else if(Screen==1 && Rows==4 && kolom=="G")
{jumlah1=dt[3].length();dt[3].toCharArray(Text1,56);EEPROM.write(2,
jumlah1);EEPROM.writeBlock<char>( 6, Text1, jumlah1);}
```

```
else if(Screen==1 && Rows==4 && kolom=="H")
{jumlah2=dt[3].length();dt[3].toCharArray(Text2,56);EEPROM.write(3,
jumlah2);EEPROM.writeBlock<char>( 62, Text2, jumlah2);}
```



```

else if(Screan==1 && Rows==4 && kolom=="I")
{jumlah3=dt[3].length();dt[3].toCharArray(Text3,56);EEPROM.write(4,
jumlah3);EEPROM.writeBlock<char>(118, Text3, jumlah3);}
else if(Screan==1 && Rows==4 && kolom=="J")
{jumlah4=dt[3].length();dt[3].toCharArray(Text4,56);EEPROM.write(5,
jumlah4);EEPROM.writeBlock<char>(174, Text4, jumlah4);}

```

```
//baca data dari EEPROM
```

```

Screan = EEPROM.read(0);
Rows   = EEPROM.read(1);
jumlah1 = EEPROM.read(2);
jumlah2 = EEPROM.read(3);
jumlah3 = EEPROM.read(4);
jumlah4 = EEPROM.read(5);
EEPROM.readBlock<char>(6, Text1, jumlah1);
EEPROM.readBlock<char>(62, Text2, jumlah2);
EEPROM.readBlock<char>(118, Text3, jumlah3);
EEPROM.readBlock<char>(174, Text4, jumlah4);

```

```
//penampil
```

```

if(Screan==1 && Rows==1)                                     //Tampil 1 Rows
{
  dmd.clearScreen( true );
  dmd.selectFont(Droid_Sans_24);
  dmd.drawMarquee(Text1, jumlah1,(32*DISPLAYS_ACROSS)-1 ,3);
  long start=millis();
  long timer=start;
  boolean ret=false;
  while(!ret)
  {
    if ((timer+30) < millis())

```

```

    {
        ret=dmd.stepMarquee(-1,0);
        timer=millis();
    }
}
}
}

```

```

if(Screen==1 && Rows==2) //Tampil 2 Rows

```

```

{
    int v,h;
    dmd.clearScreen( true );
    dmd.selectFont(Arial_Black_16);
    for( v=(DISPLAYS_DOWN*16); v>-1; v--)//masuk text 1
    {
        dmd.drawString(1, v, Text1, jumlah1, GRAPHICS_NORMAL );
        dmd.drawFilledBox( 0, (v+16), 96, (v+16), GRAPHICS_INVERSE );
        delay(50);
    }
    for( v=(DISPLAYS_DOWN*16); v>15; v--)//masuk text2
    {
        dmd.drawString(1, v, Text2, jumlah2, GRAPHICS_NORMAL );
        dmd.drawFilledBox( 0, (v-1), 96, (v-1), GRAPHICS_INVERSE );
        delay(50);
    }
    delay(1000);

```

```

for( h=0; h>(-1*jumlah1*12); h--)//keluar text1

```

```

{
    dmd.drawString(h, 0, Text1, jumlah1, GRAPHICS_NORMAL );
    dmd.drawFilledBox( 0, (v+16), 96, (v+16), GRAPHICS_INVERSE );

```

```

    delay(30);
}
delay(700);
for( h=0; h>(-1*jumlah2*12); h--)//keluar text2
{
    dmd.drawString(h, 16, Text2, jumlah2, GRAPHICS_NORMAL );
    dmd.drawFilledBox( 0, (v-1), 96, (v-1), GRAPHICS_INVERSE );
    delay(30);
}
delay(700);
}

if(Screen==1 && Rows==3)                                //Tampil 3 Rows
{
    int h;
    dmd.clearScreen( true );
    dmd.selectFont(DejaVuSansBold9);
    for(h=(-1*jumlah1*7); h<2; h++)//masuk text1
    {
        dmd.drawString(h, 1, Text1, jumlah1, GRAPHICS_NORMAL );
    }
    for(h=(-1*jumlah2*7); h<2; h++)//masuk text2
    {
        dmd.drawString(h, 11, Text2, jumlah2, GRAPHICS_NORMAL );
    }
    for(h=(-1*jumlah3*7); h<2; h++)//masuk text3
    {
        dmd.drawString(h, 21, Text3, jumlah3, GRAPHICS_NORMAL );
    }
    delay(1000);
}

```

```

for( h=0; h>(-1*jumlah1*7); h--)//keluar text1
{
    dmd.drawString(h, 1, Text1, jumlah1, GRAPHICS_NORMAL );
    delay(30);
}
delay(700);
for( h=0; h>(-1*jumlah2*7); h--)//keluar text2
{
    dmd.drawString(h, 11, Text2, jumlah2, GRAPHICS_NORMAL );
    delay(30);
}
delay(700);
for( h=0; h>(-1*jumlah3*7); h--)//keluar text3
{
    dmd.drawString(h, 21, Text3, jumlah3, GRAPHICS_NORMAL );
    delay(30);
}
delay(700);
}

if(Screen==1 && Rows==4)                                //Tampil 4 Rows
{
    int v,h;
    dmd.clearScreen( true );
    dmd.selectFont(SystemFont5x7);

    for( v=0; v<26; v++)//masuk text 4
    {
        dmd.drawString(1, v, Text4, jumlah4, GRAPHICS_NORMAL );
        dmd.drawFilledBox( 0, (v-1), 96, (v-1), GRAPHICS_INVERSE );
        delay(50);
    }
}

```

```

}
for( v=0; v<18; v++)//masuk text 3
{
    dmd.drawString(1, v, Text3, jumlah3, GRAPHICS_NORMAL );
    dmd.drawFilledBox( 0, (v-1), 96, (v-1), GRAPHICS_INVERSE );
    delay(50);
}
for( v=0; v<10; v++)//masuk text 2
{
    dmd.drawString(1, v, Text2, jumlah2, GRAPHICS_NORMAL );
    dmd.drawFilledBox( 0, (v-1), 96, (v-1), GRAPHICS_INVERSE );
    delay(50);
}
for( v=0; v<2; v++)//masuk text 1
{
    dmd.drawString(1, v, Text1, jumlah1, GRAPHICS_NORMAL );
    dmd.drawFilledBox( 0, (v-1), 96, (v-1), GRAPHICS_INVERSE );
    delay(50);
}
delay(1000);

for(h=0; h>(-1*jumlah1*6); h--)//keluar text1
{
    dmd.drawString(h, 1, Text1, jumlah1, GRAPHICS_NORMAL );
    delay(25);
}
delay(700);
for(h=0; h>(-1*jumlah2*6); h--)//keluar text2
{
    dmd.drawString(h, 9, Text2, jumlah2, GRAPHICS_NORMAL );
    delay(25);
}

```

```

    }
    delay(700);
    for(h=0; h>(-1*jumlah3*6); h--)//keluar text3
    {
        dmd.drawString(h, 17, Text3, jumlah3, GRAPHICS_NORMAL );
        delay(25);
    }
    delay(700);
    for(h=0; h>(-1*jumlah4*6); h--)//keluar text4
    {
        dmd.drawString(h, 25, Text4, jumlah4, GRAPHICS_NORMAL );
        delay(25);
    }
    delay(700);
}
}

```

```

void tampilan2()

```

```

{
    byte  Scream;
    byte  Rows;
    byte  jumlah1;
    byte  jumlah2;
    byte  jumlah3;
    byte  jumlah4;
    String kolom;
    char  Text1[56];
    char  Text2[56];
    char  Text3[56];
    char  Text4[56];

```

```

Serial.println("tampilan2");

if(ScreanIn==2)
{
    Screan = ScreanIn;
    Rows = dt[1].toInt();
    kolom = dt[2];
    EEPROM.write(231, Screan);
    EEPROM.write(232, Rows);
}

// 1 rows
if(Screan==2 && Rows==1 && kolom=="A")
{jumlah1=dt[3].length();dt[3].toCharArray(Text1,56);EEPROM.write(233,
jumlah1);EEPROM.writeBlock<char>(237, Text1, jumlah1);}

//2 rows
else if(Screan==2 && Rows==2 && kolom=="B")
{jumlah1=dt[3].length();dt[3].toCharArray(Text1,56);EEPROM.write(233,
jumlah1);EEPROM.writeBlock<char>(237, Text1, jumlah1);}
else if(Screan==2 && Rows==2 && kolom=="C")
{jumlah2=dt[3].length();dt[3].toCharArray(Text2,56);EEPROM.write(234,
jumlah2);EEPROM.writeBlock<char>(293, Text2, jumlah2);}

//3 rows
else if(Screan==2 && Rows==3 && kolom=="D")
{jumlah1=dt[3].length();dt[3].toCharArray(Text1,56);EEPROM.write(233,
jumlah1);EEPROM.writeBlock<char>(237, Text1, jumlah1);}
else if(Screan==2 && Rows==3 && kolom=="E")
{jumlah2=dt[3].length();dt[3].toCharArray(Text2,56);EEPROM.write(234,
jumlah2);EEPROM.writeBlock<char>(293, Text2, jumlah2);}

```

```
else if(Screen==2 && Rows==3 && kolom=="F")
{jumlah3=dt[3].length();dt[3].toCharArray(Text3,56);EEPROM.write(235,
jumlah3);EEPROM.writeBlock<char>(349, Text3, jumlah3);}

```

```
//4 rows

```

```
else if(Screen==2 && Rows==4 && kolom=="G")
{jumlah1=dt[3].length();dt[3].toCharArray(Text1,56);EEPROM.write(233,
jumlah1);EEPROM.writeBlock<char>(237, Text1, jumlah1);}

```

```
else if(Screen==2 && Rows==4 && kolom=="H")
{jumlah2=dt[3].length();dt[3].toCharArray(Text2,56);EEPROM.write(234,
jumlah2);EEPROM.writeBlock<char>(293, Text2, jumlah2);}

```

```
else if(Screen==2 && Rows==4 && kolom=="I")
{jumlah3=dt[3].length();dt[3].toCharArray(Text3,56);EEPROM.write(235,
jumlah3);EEPROM.writeBlock<char>(349, Text3, jumlah3);}

```

```
else if(Screen==2 && Rows==4 && kolom=="J")
{jumlah4=dt[3].length();dt[3].toCharArray(Text4,56);EEPROM.write(236,
jumlah4);EEPROM.writeBlock<char>(405, Text4, jumlah4);}

```

```
//baca data dari EEPROM

```

```
Screen = EEPROM.read(231);

```

```
Rows = EEPROM.read(232);

```

```
jumlah1 = EEPROM.read(233);

```

```
jumlah2 = EEPROM.read(234);

```

```
jumlah3 = EEPROM.read(235);

```

```
jumlah4 = EEPROM.read(236);

```

```
EEPROM.readBlock<char>(237, Text1, jumlah1);

```

```
EEPROM.readBlock<char>(293, Text2, jumlah2);

```

```
EEPROM.readBlock<char>(349, Text3, jumlah3);

```

```
EEPROM.readBlock<char>(405, Text4, jumlah4);

```

```
//penampil

```



```

if(Screen==2 && Rows==1)                                //Tampil 1 Rows
{
    dmd.clearScreen( true );
    dmd.selectFont(Droid_Sans_24);
    dmd.drawMarquee(Text1, jumlah1, (32*DISPLAYS_ACROSS)-1, 3);
    long start=millis();
    long timer=start;
    boolean ret=false;
    while(!ret)
    {
        if ((timer+30) < millis())
        {
            ret=dmd.stepMarquee(-1,0);
            timer=millis();
        }
    }
}

```

```

if(Screen==2 && Rows==2)                                //Tampil 2 Rows
{
    int v,h;
    dmd.clearScreen( true );
    dmd.selectFont(Arial_Black_16);
    for( v=(DISPLAYS_DOWN*16); v>-1; v--)//masuk text 1
    {
        dmd.drawString(1, v, Text1, jumlah1, GRAPHICS_NORMAL );
        dmd.drawFilledBox( 0, (v-1), 96, (v-1), GRAPHICS_INVERSE );
        delay(50);
    }
    for( v=(DISPLAYS_DOWN*16); v>15; v--)//masuk text2
    {

```

```

    dmd.drawString(1, v, Text2, jumlah2, GRAPHICS_NORMAL );
    delay(50);
}
delay(1000);

for( h=0; h>(-1*jumlah1*12); h--)//keluar text1
{
    dmd.drawString(h, 0, Text1, jumlah1, GRAPHICS_NORMAL );
    delay(50);
}
delay(700);
for( h=0; h>(-1*jumlah2*12); h--)//keluar text2
{
    dmd.drawString(h, 16, Text2, jumlah2, GRAPHICS_NORMAL );
    delay(50);
}
delay(700);
}

if(Screen==2 && Rows==3)                                //Tampil 3 Rows
{
    int h;
    dmd.clearScreen( true );
    dmd.selectFont(DejaVuSansBold9);
    for(h=(-1*jumlah1*7); h<2; h++)//masuk text1
    {
        dmd.drawString(h, 1, Text1, jumlah1, GRAPHICS_NORMAL );
    }
    for(h=(-1*jumlah2*7); h<2; h++)//masuk text2
    {
        dmd.drawString(h, 11, Text2, jumlah2, GRAPHICS_NORMAL );
    }
}

```



```

dmd.clearScreen( true );
dmd.selectFont(SystemFont5x7);

for( v=0; v<26; v++)//masuk text 4
{
    dmd.drawString(1, v, Text4, jumlah4, GRAPHICS_NORMAL );
    dmd.drawFilledBox( 0, (v-1), 96, (v-1), GRAPHICS_INVERSE );
    delay(50);
}
for( v=0; v<18; v++)//masuk text 3
{
    dmd.drawString(1, v, Text3, jumlah3, GRAPHICS_NORMAL );
    dmd.drawFilledBox( 0, (v-1), 96, (v-1), GRAPHICS_INVERSE );
    delay(50);
}
for( v=0; v<10; v++)//masuk text 2
{
    dmd.drawString(1, v, Text2, jumlah2, GRAPHICS_NORMAL );
    dmd.drawFilledBox( 0, (v-1), 96, (v-1), GRAPHICS_INVERSE );
    delay(50);
}
for( v=0; v<2; v++)//masuk text 1
{
    dmd.drawString(1, v, Text1, jumlah1, GRAPHICS_NORMAL );
    dmd.drawFilledBox( 0, (v-1), 96, (v-1), GRAPHICS_INVERSE );
    delay(50);
}
delay(1000);

for(h=0; h>(-1*jumlah1*6); h--)//keluar text1
{

```

```

    dmd.drawString(h, 1, Text1, jumlah1, GRAPHICS_NORMAL );
    delay(25);
}
delay(700);
for(h=0; h>(-1*jumlah2*6); h--)//keluar text2
{
    dmd.drawString(h, 9, Text2, jumlah2, GRAPHICS_NORMAL );
    delay(25);
}
delay(700);
for(h=0; h>(-1*jumlah3*6); h--)//keluar text3
{
    dmd.drawString(h, 17, Text3, jumlah3, GRAPHICS_NORMAL );
    delay(25);
}
delay(700);
for(h=0; h>(-1*jumlah4*6); h--)//keluar text4
{
    dmd.drawString(h, 25, Text4, jumlah4, GRAPHICS_NORMAL );
    delay(25);
}
delay(700);
}
}

```

```

void tampilan3()

```

```

{
    byte Screan;
    byte Rows;
    byte jumlah1;
    byte jumlah2;

```

```

byte jumlah3;
byte jumlah4;
String kolom;
char Text1[56];
char Text2[56];
char Text3[56];
char Text4[56];

Serial.println("tampilan3");

if(ScreanIn==3)
{
    Screan = ScreanIn;
    Rows = dt[1].toInt();
    kolom = dt[2];
    EEPROM.write(462, Screan);
    EEPROM.write(463, Rows);
}

// 1 rows
if(Screan==3 && Rows==1 && kolom=="A")
{jumlah1=dt[3].length();dt[3].toCharArray(Text1,56);EEPROM.write(464,
jumlah1);EEPROM.writeBlock<char>(468, Text1, jumlah1);}

//2 rows
else if(Screan==3 && Rows==2 && kolom=="B")
{jumlah1=dt[3].length();dt[3].toCharArray(Text1,56);EEPROM.write(464,
jumlah1);EEPROM.writeBlock<char>(468, Text1, jumlah1);}
else if(Screan==3 && Rows==2 && kolom=="C")
{jumlah2=dt[3].length();dt[3].toCharArray(Text2,56);EEPROM.write(465,
jumlah2);EEPROM.writeBlock<char>(524, Text2, jumlah2);}

```

```

//3 rows
else if(Screan==3 && Rows==3 && kolom=="D")
{jumlah1=dt[3].length();dt[3].toCharArray(Text1,56);EEPROM.write(464,
jumlah1);EEPROM.writeBlock<char>(468, Text1, jumlah1);}
else if(Screan==3 && Rows==3 && kolom=="E")
{jumlah2=dt[3].length();dt[3].toCharArray(Text2,56);EEPROM.write(465,
jumlah2);EEPROM.writeBlock<char>(524, Text2, jumlah2);}
else if(Screan==3 && Rows==3 && kolom=="F")
{jumlah3=dt[3].length();dt[3].toCharArray(Text3,56);EEPROM.write(466,
jumlah3);EEPROM.writeBlock<char>(580, Text3, jumlah3);}

//4 rows
else if(Screan==3 && Rows==4 && kolom=="G")
{jumlah1=dt[3].length();dt[3].toCharArray(Text1,56);EEPROM.write(464,
jumlah1);EEPROM.writeBlock<char>(468, Text1, jumlah1);}
else if(Screan==3 && Rows==4 && kolom=="H")
{jumlah2=dt[3].length();dt[3].toCharArray(Text2,56);EEPROM.write(465,
jumlah2);EEPROM.writeBlock<char>(524, Text2, jumlah2);}
else if(Screan==3 && Rows==4 && kolom=="I")
{jumlah3=dt[3].length();dt[3].toCharArray(Text3,56);EEPROM.write(466,
jumlah3);EEPROM.writeBlock<char>(580, Text3, jumlah3);}
else if(Screan==3 && Rows==4 && kolom=="J")
{jumlah4=dt[3].length();dt[3].toCharArray(Text4,56);EEPROM.write(467,
jumlah4);EEPROM.writeBlock<char>(636, Text4, jumlah4);}

//baca data dari EEPROM
Screan = EEPROM.read(462);
Rows = EEPROM.read(463);
jumlah1 = EEPROM.read(464);
jumlah2 = EEPROM.read(465);

```

```

jumlah3 = EEPROM.read(466);
jumlah4 = EEPROM.read(467);
EEPROM.readBlock<char>(468, Text1, jumlah1);
EEPROM.readBlock<char>(524, Text2, jumlah2);
EEPROM.readBlock<char>(580, Text3, jumlah3);
EEPROM.readBlock<char>(636, Text4, jumlah4);

//penampil
if(Screen==3 && Rows==1)                                //Tampil 1 Rows
{
  dmd.clearScreen( true );
  dmd.selectFont(Droid_Sans_24);
  dmd.drawMarquee(Text1, jumlah1,(32*DISPLAYS_ACROSS)-1 ,3);
  long start=millis();
  long timer=start;
  boolean ret=false;
  while(!ret)
  {
    if ((timer+30) < millis())
    {
      ret=dmd.stepMarquee(-1,0);
      timer=millis();
    }
  }
}

if(Screen==3 && Rows==2)                                //Tampil 2 Rows
{
  int v,h;
  dmd.clearScreen( true );
  dmd.selectFont(Arial_Black_16);

```



```

for( v=(DISPLAYS_DOWN*16); v>-1; v--)//masuk text 1
{
    dmd.drawString(1, v, Text1, jumlah1, GRAPHICS_NORMAL );
    dmd.drawFilledBox( 0, (v-1), 96, (v-1), GRAPHICS_INVERSE );
    delay(50);
}
for( v=(DISPLAYS_DOWN*16); v>15; v--)//masuk text2
{
    dmd.drawString(1, v, Text2, jumlah2, GRAPHICS_NORMAL );
    delay(50);
}
delay(1000);

for( h=0; h>(-1*jumlah1*12); h--)//keluar text1
{
    dmd.drawString(h, 0, Text1, jumlah1, GRAPHICS_NORMAL );
    delay(50);
}
delay(700);
for( h=0; h>(-1*jumlah2*12); h--)//keluar text2
{
    dmd.drawString(h, 16, Text2, jumlah2, GRAPHICS_NORMAL );
    delay(50);
}
delay(700);
}

if(Screen==3 && Rows==3) //Tampil 3 Rows
{
    int h;
    dmd.clearScreen( true );
}

```

```

dmd.selectFont(DejaVuSansBold9);
for(h=(-1*jumlah1*7); h<2; h++)//masuk text1
{
    dmd.drawString(h, 1, Text1, jumlah1, GRAPHICS_NORMAL );
}
for(h=(-1*jumlah2*7); h<2; h++)//masuk text2
{
    dmd.drawString(h, 11, Text2, jumlah2, GRAPHICS_NORMAL );
}
for(h=(-1*jumlah3*7); h<2; h++)//masuk text3
{
    dmd.drawString(h, 21, Text3, jumlah3, GRAPHICS_NORMAL );
}
delay(1000);

for( h=0; h>(-1*jumlah1*7); h--)//keluar text1
{
    dmd.drawString(h, 1, Text1, jumlah1, GRAPHICS_NORMAL );
    delay(30);
}
delay(700);
for( h=0; h>(-1*jumlah2*7); h--)//keluar text2
{
    dmd.drawString(h, 11, Text2, jumlah2, GRAPHICS_NORMAL );
    delay(30);
}
delay(700);
for( h=0; h>(-1*jumlah3*7); h--)//keluar text3
{
    dmd.drawString(h, 21, Text3, jumlah3, GRAPHICS_NORMAL );
    delay(30);
}

```

```

    }
    delay(700);
}

if(Screen==3 && Rows==4)                                //Tampil 4 Rows
{
    int v,h;
    dmd.clearScreen( true );
    dmd.selectFont(SystemFont5x7);

    for( v=0; v<26; v++)//masuk text 4
    {
        dmd.drawString(1, v, Text4, jumlah4, GRAPHICS_NORMAL );
        dmd.drawFilledBox( 0, (v-1), 96, (v-1), GRAPHICS_INVERSE );
        delay(50);
    }
    for( v=0; v<18; v++)//masuk text 3
    {
        dmd.drawString(1, v, Text3, jumlah3, GRAPHICS_NORMAL );
        dmd.drawFilledBox( 0, (v-1), 96, (v-1), GRAPHICS_INVERSE );
        delay(50);
    }
    for( v=0; v<10; v++)//masuk text 2
    {
        dmd.drawString(1, v, Text2, jumlah2, GRAPHICS_NORMAL );
        dmd.drawFilledBox( 0, (v-1), 96, (v-1), GRAPHICS_INVERSE );
        delay(50);
    }
    for( v=0; v<2; v++)//masuk text 1
    {
        dmd.drawString(1, v, Text1, jumlah1, GRAPHICS_NORMAL );

```

```

    dmd.drawFilledBox( 0, (v-1), 96, (v-1), GRAPHICS_INVERSE );
    delay(50);
}
delay(1500);

for(h=0; h>(-1*jumlah1*6); h--)//keluar text1
{
    dmd.drawString(h, 1, Text1, jumlah1, GRAPHICS_NORMAL );
    delay(25);
}
delay(700);
for(h=0; h>(-1*jumlah2*6); h--)//keluar text2
{
    dmd.drawString(h, 9, Text2, jumlah2, GRAPHICS_NORMAL );
    delay(25);
}
delay(700);
for(h=0; h>(-1*jumlah3*6); h--)//keluar text3
{
    dmd.drawString(h, 17, Text3, jumlah3, GRAPHICS_NORMAL );
    delay(25);
}
delay(700);
for(h=0; h>(-1*jumlah4*6); h--)//keluar text4
{
    dmd.drawString(h, 25, Text4, jumlah4, GRAPHICS_NORMAL );
    delay(25);
}
delay(700);
}
}

```

PROGRAM ARDUINO MEGA 2560

```
/*-----  
* Project   : PAPAN INFORMASI BERBASIS DOTMATRIX DI BAAKPK  
POLMAN BABEL  
* Author    : RESTU PANGESTU DAN DINA FORSA ANGGRAINI  
* Ket       : PROYEK AKHIR 2018  
*  
-----*/  
  
// Menyisipkan File Library  
#include <DMD2.h>  
#include <fonts/SystemFont5x7.h>  
#include <fonts/DejaVuSansBold9.h>  
#include <BigNumber.h>  
#include <RTCLib.h>  
  
//#include <SoftwareSerial.h>  
//SoftwareSerial mySerial(2, 3); // RX, TX  
  
// Deklarasi Fungsi  
SoftDMD dmd(1, 2); // Panjang x Lebar  
RTC_DS1307 rtc; // Chip RTC yang digunakan  
  
// Deklarasi Variable Waktu  
String Tahun;  
String Bulan;  
String Tanggal;  
String Jam;  
String Menit;  
String Detik;  
String message;
```

```

//Deklarasi Variable logo POLMAN
const int logo = 4;

void setup() {
  pinMode(logo,INPUT);

  Serial3.begin(9600);
  Serial2.begin(9600);
  Serial1.begin(9600);
  Serial.begin(9600);
  // Setup DMD
  dmd.setBrightness(200);
  dmd.begin();
  dmd.clearScreen();
  // Setup RTC
  rtc.begin();
  // Cara Setting Jam gunakan 2 fungsi dibawah, secara manual atau otomatis
  singcron PC, kemudian upload program
  // Setelah terupload, cek jam sudah sesuai, hapus code setting jam dibawah, dan
  upload ulang.

  // Set RTC sesuai dengan Laptop / PC
  // rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  // January 21, 2014 at 3am you would call:
  // rtc.adjust(DateTime(2014, 1, 21, 3, 0, 0));
}

void loop()
{
  awal:

```

```

if (Serial.available())
{
    message=Serial.readString();
    Serial2.print(message);
}
if (Serial1.available())
{
    message=Serial1.readString();
    Serial2.print(message);
}

int dataLogo = digitalRead(logo);
if (dataLogo == HIGH)
{
    dmd.clearScreen();
    while (dataLogo == HIGH)
    {
        int dataLogo = digitalRead(logo);
        if (dataLogo == LOW){break;}
        logoPolman(0,5);
        if (Serial.available() || Serial1.available()){goto awal;}
    }
}

if (dataLogo == LOW)
{
    dmd.clearScreen();
    while (dataLogo == LOW)
    {
        int dataLogo = digitalRead(logo);
        if (dataLogo == HIGH){break;}
    }
}

```

```

    showTime();
    if (Serial.available() || Serial1.available()){goto awal;}
  }
}
}

```

```

void showTime()

```

```

{
  for(int y=0; y<32; y++){dmd.setPixel(31, y, GRAPHICS_ON);} //garis
  pembatas

```

```

// Program menampilkan waktu

```

```

DateTime now = rtc.now();

```

```

Tahun = now.year();

```

```

TwoDigit(now.month(),Bulan);

```

```

TwoDigit(now.day(),Tanggal);

```

```

TwoDigit(now.hour(),Jam);

```

```

TwoDigit(now.minute(),Menit);

```

```

TwoDigit(now.second(),Detik);

```

```

dmd.selectFont(BigNumber);

```

```

dmd.drawString(0,0,Jam);

```

```

// dmd.drawString(14,0,".");

```

```

dmd.drawString(17,0,Menit);

```

```

dmd.selectFont(SystemFont5x7);

```

```

dmd.drawString(1,16,Tanggal);

```

```

dmd.drawString(13,16,"-");

```

```

dmd.drawString(19,16,Bulan);

```

```

dmd.selectFont(DejaVuSansBold9);

```



```
for(int x=1; x<7;x++){dmd.drawString(x,23,Tahun);delay(65);}
for(int x=7; x>1;x--){dmd.drawString(x,23,Tahun);delay(65);}
```

```
//titik dua
```

```
int det=now.second()%2;
```

```
if(det!=0)
```

```
{
    dmd.setPixel(14, 4, GRAPHICS_ON);
    dmd.setPixel(14, 5, GRAPHICS_ON);
    dmd.setPixel(15, 4, GRAPHICS_ON);
    dmd.setPixel(15, 5, GRAPHICS_ON);
    dmd.setPixel(14, 9, GRAPHICS_ON);
    dmd.setPixel(14,10, GRAPHICS_ON);
    dmd.setPixel(15, 9, GRAPHICS_ON);
    dmd.setPixel(15,10, GRAPHICS_ON);
}
```

```
else
```

```
{
    dmd.setPixel(14, 4, GRAPHICS_OFF);
    dmd.setPixel(14, 5, GRAPHICS_OFF);
    dmd.setPixel(15, 4, GRAPHICS_OFF);
    dmd.setPixel(15, 5, GRAPHICS_OFF);
    dmd.setPixel(14, 9, GRAPHICS_OFF);
    dmd.setPixel(14,10, GRAPHICS_OFF);
    dmd.setPixel(15, 9, GRAPHICS_OFF);
    dmd.setPixel(15,10, GRAPHICS_OFF);
}
}
```

```
void TwoDigit(int digit, String &hasilDigit) //tampil 2 digit( jika kurang dari 10
maka di awali angka 0)
```

```

{
    hasilDigit = "";
    if (digit < 10)
    {
        hasilDigit += "0";
        hasilDigit += digit;
    }
    else {hasilDigit += digit;}
}

```

```

void logoPolman(int a, int b)

```

```

{
    int x,y;

    dmd.setPixel(15+a, 1+b, GRAPHICS_ON);
    for(x=14; x<17; x++){dmd.setPixel(x+a, 2+b, GRAPHICS_ON);}
    for(x=13; x<18; x++){dmd.setPixel(x+a, 3+b, GRAPHICS_ON);}
    for(x=12; x<19; x++){dmd.setPixel(x+a, 4+b, GRAPHICS_ON);}
    for(x=1; x<8; x++) {dmd.setPixel(x+a, 5+b, GRAPHICS_ON);}
    for(x=11; x<19; x++){dmd.setPixel(x+a, 5+b, GRAPHICS_ON);}
    for(x=2; x<9; x++) {dmd.setPixel(x+a, 6+b, GRAPHICS_ON);}
    for(x=10; x<18; x++){dmd.setPixel(x+a, 6+b, GRAPHICS_ON);}
    for(x=3; x<17; x++) {dmd.setPixel(x+a, 7+b, GRAPHICS_ON);}
    for(x=4; x<16; x++) {dmd.setPixel(x+a, 8+b, GRAPHICS_ON);}
    for(x=5; x<15; x++) {dmd.setPixel(x+a, 9+b, GRAPHICS_ON);}
    for(x=6; x<14; x++) {dmd.setPixel(x+a, 10+b, GRAPHICS_ON);}
    for(x=7; x<13; x++) {dmd.setPixel(x+a, 11+b, GRAPHICS_ON);}
    for(x=8; x<12; x++) {dmd.setPixel(x+a, 12+b, GRAPHICS_ON);}
    for(x=9; x<11; x++) {dmd.setPixel(x+a, 13+b, GRAPHICS_ON);}

    for(x=17; x<24; x++){dmd.setPixel(x+a, 1+b, GRAPHICS_ON);}
}

```

```
for(x=18; x<25; x++){dmd.setPixel(x+a, 2+b, GRAPHICS_ON);}
for(x=19; x<26; x++){dmd.setPixel(x+a, 3+b, GRAPHICS_ON);}
for(x=20; x<27; x++){dmd.setPixel(x+a, 4+b, GRAPHICS_ON);}
for(x=21; x<28; x++){dmd.setPixel(x+a, 5+b, GRAPHICS_ON);}
for(x=22; x<29; x++){dmd.setPixel(x+a, 6+b, GRAPHICS_ON);}
for(x=23; x<28; x++){dmd.setPixel(x+a, 7+b, GRAPHICS_ON);}
for(x=24; x<27; x++){dmd.setPixel(x+a, 8+b, GRAPHICS_ON);}
dmd.setPixel(25+a, 9+b, GRAPHICS_ON);
```

```
for(y=7; y<15; y++) {dmd.setPixel(29+a, y+b, GRAPHICS_ON);}
for(y=8; y<16; y++) {dmd.setPixel(28+a, y+b, GRAPHICS_ON);}
for(y=9; y<17; y++) {dmd.setPixel(27+a, y+b, GRAPHICS_ON);}
for(y=10; y<18; y++){dmd.setPixel(26+a, y+b, GRAPHICS_ON);}
for(y=11; y<19; y++){dmd.setPixel(25+a, y+b, GRAPHICS_ON);}
for(y=12; y<20; y++){dmd.setPixel(24+a, y+b, GRAPHICS_ON);}
for(y=13; y<19; y++){dmd.setPixel(23+a, y+b, GRAPHICS_ON);}
for(y=14; y<18; y++){dmd.setPixel(22+a, y+b, GRAPHICS_ON);}
for(y=15; y<17; y++){dmd.setPixel(21+a, y+b, GRAPHICS_ON);}
```

```
dmd.setPixel(14+a, 11+b, GRAPHICS_ON);
for(x=13; x<16; x++){dmd.setPixel(x+a, 12+b, GRAPHICS_ON);}
for(x=12; x<17; x++){dmd.setPixel(x+a, 13+b, GRAPHICS_ON);}
for(x=11; x<18; x++){dmd.setPixel(x+a, 14+b, GRAPHICS_ON);}
for(x=11; x<19; x++){dmd.setPixel(x+a, 15+b, GRAPHICS_ON);}
for(x=12; x<20; x++){dmd.setPixel(x+a, 16+b, GRAPHICS_ON);}
for(x=13; x<21; x++){dmd.setPixel(x+a, 17+b, GRAPHICS_ON);}
for(x=14; x<22; x++){dmd.setPixel(x+a, 18+b, GRAPHICS_ON);}
for(x=15; x<23; x++){dmd.setPixel(x+a, 19+b, GRAPHICS_ON);}
}
```