



**SISTEM KENDALI PAPAN PENGUMUMAN DIGITAL
MENGUNAKAN MIKROKONTROLER ARDUINO DAN
ANDRCID**

Disusun dan Diajukan untuk Memenuhi Persyaratan Ujian Akhir Memperoleh
Gelar Sarjana Komputer pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH

NAMA : HARDI PERMANA
NPM : 1414370556
PROGRAM STUDI : SISTEM KOMPUTER

**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2019**

LEMBAR PENGESAHAN

SISTEM KENDALI PAPAN PENGUMUMAN DIGITAL MENGUNAKAN *MIKROKONTROLER ARDUINO* DAN ANDROID

Disusun Oleh :

NAMA : HARDI PERMANA
NPM : 1414370556
PROGRAM STUDI : SISTEM KOMPUTER

Skripsi Telah Disetujui Oleh Dosen Pembimbing Skripsi

Pada Tanggal: 24 Agustus 2019

Dosen Pembimbing I


Herdianto, S.Kom., M.T

Dosen Pembimbing II


Amani Darma Tarigan, ST., M.T

Mengetahui,

Dekan Fakultas Sains Dan Teknologi

Ketua Program Studi Sistem Komputer



Sri Shanti Indera, S.T., M.Sc


Dr. Muhammad Iqbal, S.Kom., M.Kom

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : HARDI PERMANA
NPM : 1414370556
Prodi : Sistem Komputer
Konsentrasi : Keamanan Jaringan Komputer
Judul Skripsi : “SISTEM KENDALI PAPAN
PENGUMUMAN DIGITAL
MENGUNAKAN *MIKROKONTROLER*
ARDUINO DAN ANDROID”

Dengan ini menyatakan bahwa :

1. Tugas Akhir/Skripsi saya bukan hasil Plagiat
2. Saya tidak akan menuntut perbaikan nilai indeks Prestasi Kumulatif (IPK) setelah ujian Sidang Meja Hijau
3. Skripsi saya dapat dipublikasikan oleh pihak lembaga, dan saya tidak akan menuntut akibat publikasi tersebut

Demikian pernyataan ini saya perbuat dengan sebenar-benarnya,
terimakasih

Medan, 24 Agustus 2019



buat pernyataan

HARDI PERMANA

1414370556

ABSTRAK

Perkembangan yang terjadi di era globalisasi sekarang banyaknya perubahan yang semakin pesat untuk di bidang teknologi informasi dan digital, kita bisa melihat bagaimana contoh teknologi digital yang ada di sekeliling kita seperti alat informasi atau pengumuman yang berbentuk digital contohnya seperti *Running text* yang merupakan sebuah alat yang berfungsi menampilkan suatu informasi, iklan, peringatan dan lain-lain yang bersifat formal ataupun non formal. Oleh karena itu *running text* sangatlah dibutuhkan di era globalisasi ini dengan tujuan tertentu dan memberikan kesan menarik saat dilihat oleh masyarakat. Rancang bangun ini bertujuan untuk membuat papan informasi atau pengumuman yang menyajikan fitur-fitur berupa ucapan selamat datang dan lain-lain, menggunakan sebuah mikrokontroler Arduino Uno sebagai pengendalinya. Tampilan ucapan selamat datang disajikan di *LED Dot Matrix* dan dikendalikan oleh android lalu di transfer data melalui via *Bluetooth* dari jarak tertentu, Dari hasil analisa perancangan ini dapat disimpulkan bahwa di dalam *mikrokontroler, arduino uno* dan dikendalikan oleh android melalui *Bluetooth*, berbagai macam karakter kata tampilan ucapan selamat datang yang dapat di kontrol oleh android melalui transfer data menggunakan via *bluetooth* dan menampilkan karakter kata di *Led Dot Matrix*.

Kata Kunci : *Arduino uno, Android, Bluetooth, LED Dot Matrix,*

KATA PENGANTAR

Puji syukur saya ucapkan terhadap kehadiran ALLAH SWT, yang telah melimpahkan berkat dan karunianya sehingga saya dapat melaksanakan penelitian dan menyelesaikan laporan yang dimana laporan skripsi merupakan salah satu syarat kelulusan di Program Studi Sistem Komputer S1 pada Fakultas *Sains And Technology* Universitas Pembangunan Pancabudi Medan. Adapun judul dari laporan skripsi ini “**Sistem Kendali Papan Pengumuman Digital Menggunakan Mikrokontroler Arduino & Android**” pengalaman penulis yang terbatas, untuk itu saya mengharapkan kritik dan sarannya yang membangun.

Selesainya laporan ini tidak terlepas dari bantuan dan bimbingan dari berbagai pihak, untuk itu pada kesempatan ini saya dengan tulus dan ikhlas menyampaikan ucapan terima kasih sebesar-besarnya kepada :

1. Bapak Dr. H.Muhammad Isa Indrawan, SE., M.M., selaku Rektor Universitas Pembangunan Panca Budi Medan.
2. Sri Shindi Indira, ST.,M.Sc., selaku Dekan Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.
3. Bapak Muhammad Iqbal, S.Kom.M.Kom., selaku Ketua Program Studi Sistem Komputer Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan..
4. **Herdianto, S.Kom.,MT**, Selaku Dosen Pembimbing I dan **Amani Dharma Tarigan, S.T.,M.T** Selaku Dosen Pembimbing II yang juga telah banyak

memberikan bimbingan kepada saya.

5. Terima Kasih kepada kedua orang tua saya yang telah banyak memberikan dukungan kepada saya, memberikan motivasi dan doa sehingga saya dapat menyelesaikan tugas akhir laporan skripsi ini.
6. Dan tidak lupa juga saya mengucapkan banyak terima kasih kepada Bapak dan Ibu Dosen selaku Pengajar pada Fakultas Sains And Technology Universitas Pembangunan Panca Budi Medan.

Akhir kata saya mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah banyak membantu, semoga semua bantuan yang telah diberikan kepada saya mendapat balasan dari ALLAH SWT dan laporan ini dapat bermanfaat bagi seluruh pembaca.

Medan, 24 AGUSTUS 2019

Penulis

HARDI PERMANA

NPM : 1414370556

DAFTAR ISI

ABSTRAK	i
KATA PENGANTAR	ii
DAFTAR ISI	iv
DAFTAR TABEL	vii
DAFTAR GAMBAR	viii
BAB I PENDAHULUAN	1
1. Latar Belakang	1
2. Rumusan Masalah.....	3
3. Batasan Masalah.....	3
4. Tujuan Penelitian	3
5. Manfaat Penelitian	4
6. Metodologi Penelitian	4
7. Sistematika Penulisan	5
BAB II LANDASAN TEORI	7
1. Pengertian Running Text.....	7
2. LED Dot Matrix	7
3. IC MAX7219	10
4. Shift Register.....	11
5. Arduino Uno R3.....	13
6. Arduino IDE.....	18
7. Bluetooth HC-05	23

8. B4A(<i>Basic4Android</i>)	26
9. B4A Bridge	28
10. Keamanan dan Privasi	29
11. Comfiler.....	30
12. Memulai Program Basic4Android.....	30
13. Rilis dan Rilis (obfuscated) mode	33
14. Smartphon.....	34
15. Android.....	34
BAB III GAMBARAN UMUM.....	35
1. Analisis Sistem.....	35
2. Analisis Arsitektur Perancangan Sistem.....	33
3. Analisis Kebutuhan Non Fungsional	35
4. Analisis Perancangan Alat Keras.....	35
5. Perancangan Perangkat Keras	37
a. Arduino Uno.....	37
b. Rangkaian Bloethooth.....	38
c. Rangkaian Modul Dot Matrix 8x8	39
6. Analisis Kebutuhan Fungsional	40
a. Use Case Diagram.....	40
b. Use Case Scenario.....	41
7. Perancangan Sistem	49
8. Perancangan Antarmuka	55

BAB IV ANALISA DAN PEMBAHASAN.....	57
1. Implementasi Sistem.....	57
2. Implementasi Perangkat Keras.....	57
3. Implementasi Perangkat Lunak.....	58
4. Petunjuk Pengoprasian Aplikasi	58
5. Tampilan Aplikasi	58
6. Tampilan Hardware.....	59
7. Pengujian Sistem.....	60
8. Pengujian dan Analisis Sistem	60
a. Pengujian Koneksi Awal	61
b. Pengujian Karakter Masukan	61
c. Pengujian Kecepatan Tampilan Karakter	63
d. Pengujian koneksi Ulang.....	64
e. Pengujian Karakter Berupa Simbol.....	65
f. Pengujian Bluetooth.....	67
BAB V PENUTUP	69
B. Kesimpulan.....	69
C. Saran	70
DAFTAR PUSTAKA	

DAFTAR GAMBAR

No.	Judul	Halaman
1.	Display LED Dot Matrix.....	8
2.	LED Matrix	9
3.	Konfigurasi Pin IC MAX7219.....	10
4.	driver MAX7219.....	11
5.	<i>Shift Register</i>	13
6.	<i>Arduino Uno R3</i>	18
7.	<i>Arduino IDE</i>	22
8.	Modul Bluetooth HC-05	24
9.	Konfigurasi Modul Bluetooth HC-05.....	25
10.	Basic4Android	28
11.	Interface B4A Bridge.....	30
12.	Confile.....	31
13.	Keamanan & Privasi	30
14.	Basic4Android	33
15.	Rancangan Sistem.....	36
16.	Metode Transfer Data Bluetooth.....	38
17.	Rangkaian Perangkat Keras <i>Arduino</i>	40
18.	Rangkaian Bluetooth	41
19.	Rangkaian <i>Modul Dot Matrix 8x8</i>	42

20. USE Case diagram	43
21. Activity Diagram Input	47
22. Activity Diagram Kecepatan Teks	48
23. Activity Diagram Kecerahan Teks.....	47
24. Diagram Input Teks	50
25. Diagram Kecepatan Teks.....	50
26. Diagram Kecerahan Teks.....	51
27. Flowchart	52
28. Flowchart Input Text	53
29. Flowchat Kecerahan	54
30. Flowchat Kecepatan	56
31. Perancangan Antarmuka	57
32. Tampilan Aplikasi Running Text.....	59
33. Tampilan Hardware.....	60

DAFTAR TABEL

No.	Judul	halaman
1.	Tabel 1. Spesifikasi <i>Arduino Uno R3</i>	14
2.	Tabel 2. Use <i>Case</i> Skenario <i>input</i> teks	42
3.	Tabel 3. Use <i>Case</i> skenario Kecepatan teks	42
4.	Tabel 4. Use <i>Case</i> skenario Kecerahan teks	43
5.	Tabel 5. Implementasi Perangkat Keras	57
6.	Tabel 6. Perangkat Lunak	58
7.	Tabel 7. Pengujian Koneksi	61
8.	Tabel 8. Pengujian Karakter Masukan.....	62
9.	Tabel 9. Pengujian Kecepatan Tampilan Karakter Input.....	63
10.	Tabel 10. Pengujian Koneksi Ulang Setelah Terputus	64
11.	Tabel 11. Pengujian Karakter Simbol-Simbol Google Keyboard.....	65
12.	Tabel 12. Pengujian Jarak Koneksi <i>Bluetooth</i>	67

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Running text merupakan sebuah alat yang berfungsi menampilkan suatu teks informasi, iklan, peringatan dan lain-lain yang bersifat formal ataupun non formal. Oleh karena itu *running text* sangatlah dibutuhkan di era globalisasi ini dengan tujuan tertentu dan memberikan kesan menarik saat dilihat oleh masyarakat. Rancang bangun ini bertujuan untuk membuat papan informasi yang menyajikan fitur-fitur berupa ucapan selamat datang, ataupun berbagai text ucapan lainnya *Running text* saat ini semakin sering di gunakan di lingkungan masyarakat terutama di tempat-tempat umum baik itu di pusat perbelanjaan, pertokoan, SPBU, kantor-kantor pemerintahan, hotel-hotel dan sebagainya yang digunakan untuk berbagai kepentingan informasi, iklan, promosi dan yang lainnya yang disajikan semenarik mungkin untuk di lihat masyarakat. (lilik dan Rita, 2017).

Seiring berjalanya waktu penelitian-penelitian yang berkaitan dengan teknologi terkhusus penelitian yang dilakukan dengan *running text*, antara lain penelitian *running text* yang dilakukan oleh Royhasian 2015; dengan judul Implementasi *Running Text* berbasis *Mikrokontroler AT89S51*. Pada penelitian ini menggunakan mikrokontroler AT89S51 sebagai pengolah data sekaligus menyimpan data-data biner yang diperlukan untuk mengendalikan *dotmatrix* sebagai output data. (Royhasian, 2015).

Pada Penelitian selanjutnya yang dilakukan oleh Catur Budi Waluyo Volume I, Nomor 1, Juli 2018 . dengan judul pelatihan pembuatan Running Text berbasis Matrix Displau Led dengan Menggunakan HD-U6A. Pada penelitian ini mendesain dan merakit *running text* secara *software* dan *hardware* dan secara *software* dilaksanakan dengan menggunakan aplikasi *HD2016A* dan *Power* Mendesain tulisan dalam program.

Pada penelitian kali ini penulis mengembangkan *running text* dengan menggunakan *arduino uno r3* yang dapat menampilkan informasi pada modul *display dotmatrix LED 8x32* yang diproses pada *arduino*. Dimana penulis menggunakan modul *bluetooth hc-05* untuk memudahkan menyunting dan menampilkan informasi secara real – time yang dapat dilakukan android dan diproses melalui *arduino* dan ditampilkan pada modul *display dotmatrix LED 8x32*. Pada kesempatan ini saya memberikan judul “**SISTEM KENDALI PAPAN PENGUMUMAN DIGITAL MENGGUNAKAN MIKROKONTROLER ARDUINO DAN ANDROID**”.

2.1 Rumusan Masalah

Berdasarkan latar belakang permasalahan di atas, maka dalam skripsi ini masalah pada peneliti ini adalah :

- a. Bagaimana merancang dan merubah tampilan pada *running text* antara *smartphone android* ke *arduino* dengan menggunakan koneksi *bluetooth* ?
- b. Bagaimana tingkat akurasi daripada running teks yang telah dibuat dari

smartphone android ke mikrokontroller *arduino* yang akan ditampilkan pada modul *display dotmatrix LED 8x32* ?

3.1 Batasan Masalah

Berdasarkan latar belakang permasalahan di atas, maka dalam tugas akhir ini batasan masalahnya adalah sebagai berikut :

- a. Aplikasi yang dipakai menggunakan *Basic4Android*.
- b. Menggunakan mikrokontroler *arduino uno R3*.
- c. Pengiriman informasi menggunakan modul *bluetooth hc-05*.
- d. Aplikasi yang dibuat dapat mengirim teks, mengatur kecepatan teks, dan mengatur ketebalan cahaya.

4.1 Tujuan Penelitian

- a. Untuk mengetahui cara merancang aplikasi *running text* menggunakan *mikrokontroler arduino* yang dapat di kontrol melalui *smartphone android* dengan media *bluetooth* sebagai media transfer data.
- b. Untuk mengetahui tingkat akurasi dari *running teks* yang telah di buat melalui *smartphone android* menggunakan *mikrokontroler arduino* dengan media *bluetooth* pada modul *display dotmatrix LED 8x32*.

5.1 Manfaat Penelitian

Adapun manfaat dari pada dibuatnya skripsi ini adalah sebagai berikut :

- a. Dapat mengatur informasi yang akan ditampilkan pada *running text* melalui *smartphone android*.

- b. Dapat menampilkan mengatur informasi *running text* tanpa harus membuka *chip* pada *running text* secara manual.

6.1 Metodologi Penelitian

Dalam pengerjaan skripsi ini dilakukan beberapa tahapan yaitu:

- a. Studi pustaka

Tahap untuk mempelajari, mengumpulkan serta menyeleksi penjelasan dari data – data tentang penggunaan mikrokontroler *arduino* dan penjelasan tentang pemrograman *android*.

- b. Analisis Sistem

Tahap untuk bertujuan menganalisis dan mengimplementasi terhadap program serta komponen – komponen yang digunakan pada *running text*.

- c. Perancangan Sistem

Merancang aplikasi *running text* berbasis *android* dengan mikrokontroler *arduino*. Mencakup koneksi aplikasi *android* dan penyusunan komponen–komponen elektronika.

- d. *Testing*

Menguji sistem yang sudah dibangun agar dapat melakukan koreksi apabila terdapat kesalahan pada sistem.

- e. Dokumentasi Sistem

Melakukan dokumentasi terhadap sistem serta analisis yang telah diperoleh.

7.1 Sistematika Penulisan

Penulisan skripsi ini tersusun dalam 5 (lima) bab dengan sistematika penulisan

sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi latar belakang penelitian, perumusan masalah penelitian, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan Sistem Kendali Papan Pengumuman Digital Menggunakan *Mikrokontroler Arduino Dan Android*.

BAB II LANDASAN TEORI

Bab ini menjelaskan tentang teori – teori yang berkaitan dengan *running text, android*, dan koneksi *bluetooth*.

BAB III ANALISA DAN PERANCANGAN

Bab ini menguraikan tentang tahapan–tahapan dalam sistem yang dibangun, termasuk aplikasi, rancangan dan alur program, serta komponen yang digunakan.

BAB IV IMPLEMENTASI DAN PEMBAHASAN

Bab ini menjelaskan tentang tata cara kerja dari komponen - komponen yang dibangun pada sistem serta menganalisis hasil pengujian terhadap aplikasi berbasis *android* di *smartphone* menggunakan *mikrokontroler arduino*.

BAB V PENUTUP

Bab ini berisi kesimpulan dari hasil penulisan dan saran – saran yang dapat digunakan sebagai bahan untuk perbaikan dan pengembangan sistem *running text* dengan aplikasi berbasis *android* menggunakan mikrokontroler *arduino*.

BAB II

LANDASAN TEORI

2.1 Pengertian Running Text

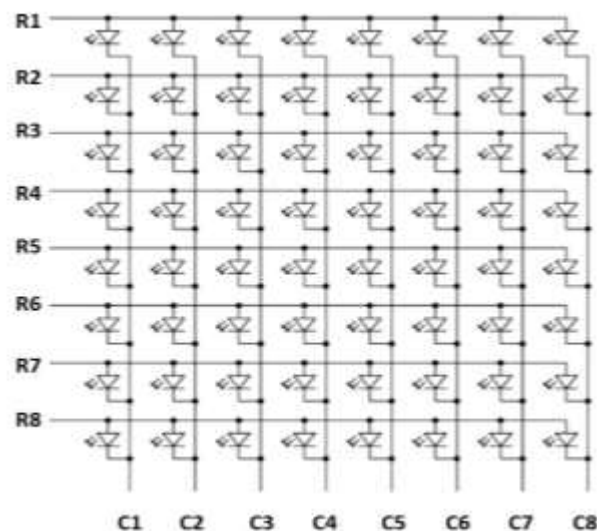
Running Text ialah LED Display adalah alat yang terdiri dari susunan lampu-lampu LED dan menghasilkan bentuk tulisan dalam rangkaiannya. Untuk menampilkan tulisan didalam LED ini semua telah terkomputerisasi, sehingga tidak perlu merubah susunan dari lampu tersebut. (Agus Mariyadi 2015) . Alat ini menjadi sebuah sarana penyampai informasi yang menarik dan fleksibel karena letaknya dapat disesuaikan dengan keinginan. Penyebaran informasi dilakukan oleh organisasi-organisasi, perusahaan swasta dan instansi pemerintahan. LED Display Saat ini informasi yang disampaikan jika hanya disampaikan menggunakan koran, spanduk, banner, dan media cetak lainnya hanya dapat dipakai untuk satu kali saja, namun apabila kita sampaikan melalui papan pengumuman yang berupa *running text* dapat kita pergunakan berulang – ulang dan menyajikan tampilan yang lebih menarik. *Running text* yang banyak dipakai saat ini untuk dapat mengubah informasi yang di tampilkan *running text* harus dilakukan secara manual dari dalam *chip* yang tertanam pada komponen elektronik pembangun *running text*.

2.2 LED Dot Matrix

LED dot matrix adalah sejumlah *LED* yang disusun dalam kolom dan baris. *LED-LED* ini kemudian digunakan untuk menampilkan gambar-gambar atau tulisan yang biasanya ditampilkan dengan efek animasi tertentu *Led dot matrix display* pada dasarnya adalah serangkaian *LED* yang dihubungkan dalam baris dan kolom. Hal ini dilakukan untuk menghemat jumlah pin yang diperlukan untuk

menyalakan tiap *LED*. Sebagai contoh pada *LED dot matrix* 8 x 32 (Gambar 1) memiliki 64 pin I/O untuk setiap piksel *LED*. Dengan menghubungkan semua anode bersama di dalam beberapa baris (R1 hingga R8) katoda dalam satu kolom – kolom (C1 samapi C8), maka jumlah pin I/O dapat direduksi menjadi hanya 16 pin. Setiap piksel *LED* dialamatkan oleh kardinat baris dan kolomnya.

Display LED Dot Matrix pada umumnya terbentuk oleh beberapa *LED* (berbentuk “Dot”) yang disusun membentuk matriks kolom dan 8 baris (8x8). Kolom berfungsi sebagai katoda (*Common Chatode*) dan baris sebagai anoda (*Common anode*) atau sebaliknya. *Display Dot Matrix* 8 kolom dan 8 baris (8x8) bisa menampilkan angka dan huruf atau bahkan gambar (grafik). Gambar 1 memperlihatkan *display* dot matrix dengan kolom sebagai katoda dan baris sebagai anoda. pada dasarnya dot matrix adalah *LED display* yang disusun sedemikian rupa sehingga untuk menghidupkan led *display* tersebut dibutuhkan kombinasi *logic* antara pin baris dan kolom.



Gambar 1. *Display LED Dot Matrix*

Sumber : (Muslihin, 2016:29)

Ada banyak cara untuk mengendalikan led dot matriks, masing–masing cara memiliki kelebihan dan kekurangannya, cara sederhana menegendalikan langsung dot matrix dengan menggunakan *ic Multiplexer* (dapatmenyalurkan data dari data keluarannya) . Namun pada tulisan ini penulis menggunakan komunikasi SPI dengan menggunakan *ic MAX 7219* sehingga jumlah pin *input* yang diperlukan dan bisa dipangkas menjadi hanya 5 pin (*VCC*, *GND*, *DIN*, *CS*, *CLK*). Berikut penampakanya dari modul *LED dot matriks display* yang penulis gunakan pada tutorial ini. Berdasarkan gambar rangkaian diatas maka *LED* akan menyala apabila baris (R1-R8) diberi *Vcc* dan Kolom (C1-C8) diberi *Ground*. Untuk dapat menampilkan karakter yang diinginkan maka kolom atau baris harus diberi logika secara bergantian. Karena jika tidak maka *LED matrix* tidak dapat menampilkan karakter tertentu. Sebagai contoh untuk menampilkan angka 0 maka pada *LED matrix* akan seperti gambar 2 berikut:

	A0	A1	A2	A3	A4	A5	A6	A7
R0	0	0	0	0	0	0	0	0
R1	0	0	0	1	1	1	0	0
R2	0	0	1	0	0	0	1	0
R3	0	0	1	0	0	0	1	0
R4	0	0	1	0	0	0	1	0
R5	0	0	1	0	0	0	1	0
R6	0	0	1	0	0	0	1	0
R7	0	0	0	1	1	1	0	0

Gambar 2. *LED matrix*

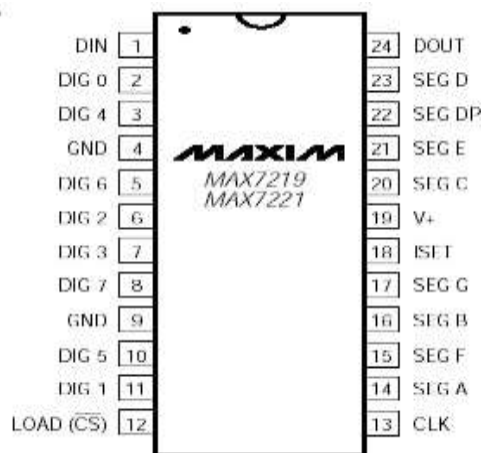
Sumber. ((Muslihin, 2016:31)

Angka 0 pada *LED matrix* misalkan kolom adalah *anoda* dan baris adalah *katoda* pada *LED*. Pertama-tama, kolom A0 diberi logika 1, dan baris R0-R7 diberi logika

1, sehingga semua *LED* pada kolom A0 mati. Selanjutnya kolom A1, sama seperti kolom A0. Lalu saat giliran kolom A2 diberi logika 1, maka baris R0-R7 diberikan logika sebagai berikut berurutan dari R0-R7, 00111110. Lalu A3 diberi logic 1, maka baris R0-R7 diberi logic, 01000001, dan seterusnya hingga A7.

2.3 IC MAX7219

Driver LED MAX7219 merupakan suatu penghemat pin prosesor dan waktu pemrosesan. Menggunakan MAX 7219 Anda dapat menggerakkan 64 LED saat Anda hanya membutuhkan 3 kabel untuk menghubungkannya ke *mikrokontroler* (Ini tidak termasuk V CC dan GND yang dianggap tersedia). Selain itu Anda dapat memadukan beberapa chip 7219 untuk tampilan yang lebih besar.

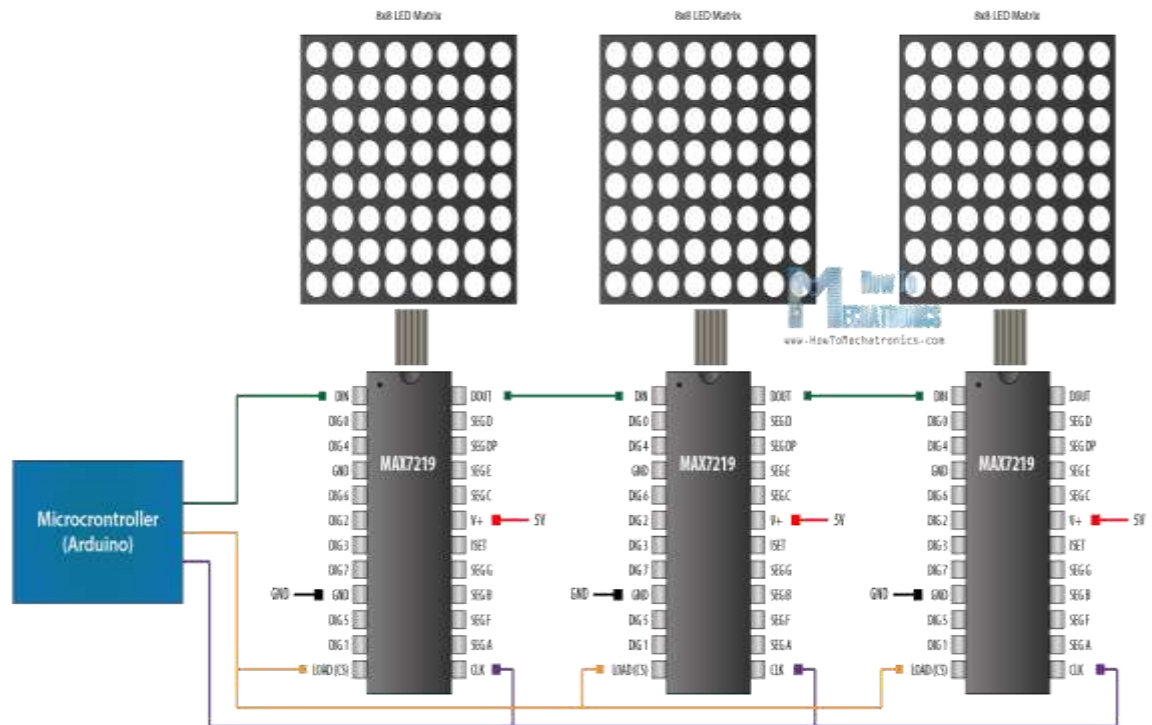


Gambar 3. Konfigurasi Pin IC MAX7219

Sumber : (Muslihin, 2016:27)

2.4 Driver IC MAX7219

Sekarang mari kita lihat lebih dekat pada driver MAX7219. IC mampu mengemudikan 64 LED individu saat hanya menggunakan 3 kabel untuk komunikasi dengan Arduino, dan terlebih lagi kita dapat daisy rantai beberapa driver dan matriks dan masih menggunakan 3 kabel yang sama.



Gambar 4. driver MAX7219

Sumber : (Dejan Nedelkovski 2017)

64 LED didorong oleh 16 pin output IC. Pertanyaannya sekarang adalah bagaimana itu mungkin. Jumlah maksimum LED menyala pada saat yang sama sebenarnya adalah delapan. LED diatur sebagai 8×8 set baris dan kolom. Jadi MAX7219 mengaktifkan setiap kolom untuk periode waktu yang sangat singkat dan pada saat yang sama juga menggerakkan setiap baris. Jadi dengan cepat beralih melalui kolom dan baris mata manusia hanya akan melihat cahaya terus menerus.

2.5 Shift Register

Register geser adalah suatu rangkaian yang menggunakan flip-flop yang saling disambung secara seri sehingga setiap bit yang disimpan di keluaran Q digeser ke flip-flop berikutnya. Pergeseran bit ini terjadi pada setiap pulsa clock.

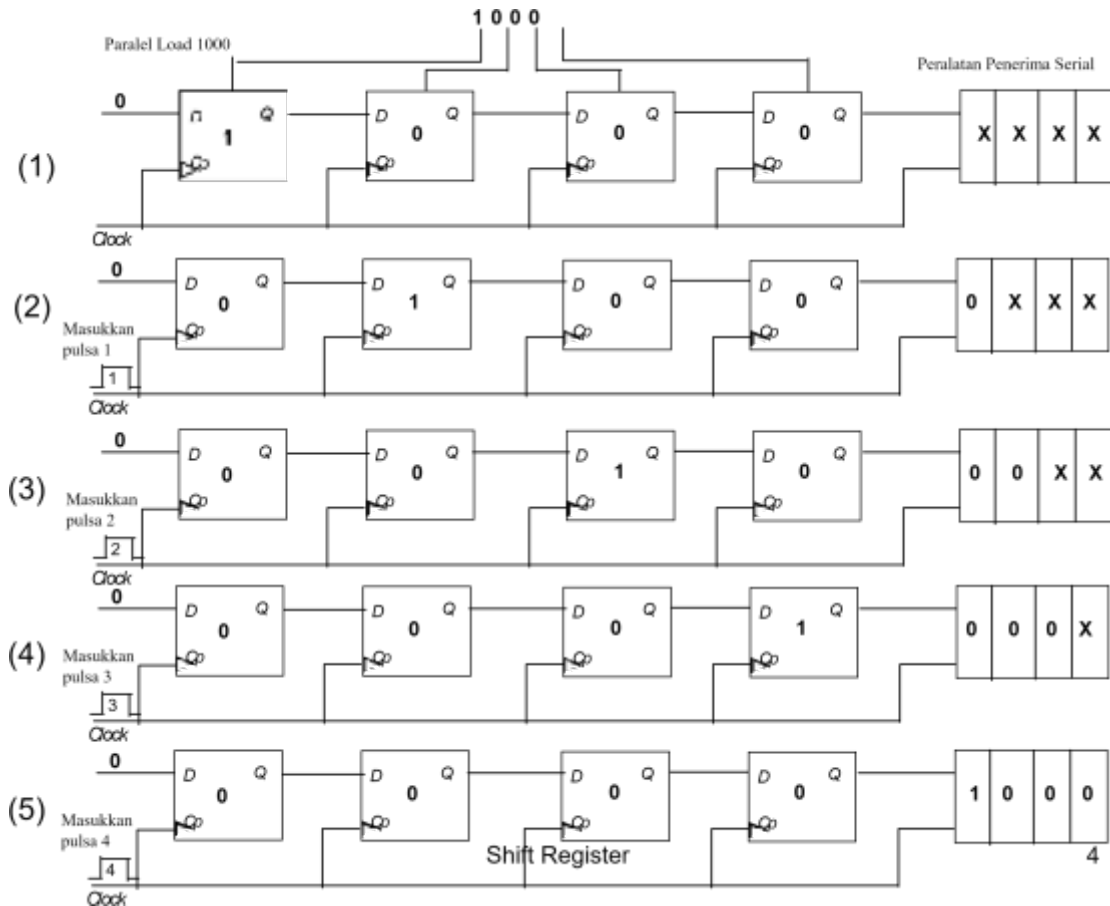
Pulsa-pulsaclock tersebut dikirim kesemua flip-flop dalam register, sehingga operasinya berjalan secara sinkron. Flip-flop jenis apapun yang operasinya sesuai (terpicu tepian) dapat dipakai.

Register merupakan blok logika yang sangat penting dalam kebanyakan sistem digital. Register sering digunakan untuk menyimpan (sementara) informasi biner yang muncul pada keluaran sebuah matrik pengkodean. Disamping itu, register sering digunakan untuk menyimpan (sementara) data biner yang sedang dikodekan

Maka register membentuk suatu kaitan yang sangat penting antara sistem digital utama dan kanal-kanal keluaran.

Register yang paling sederhana terdiri dari satu flip-flop saja, yang berarti hanya dapat menyimpan data terdiri suatu bit bilangan biner saja yaitu 0 atau 1 oleh sebab itu untuk menyimpan data yang terdiri empat bit bilangan biner maka diperlukan empat buah flip-flop.

Register geser merupakan kelas komponen yang sangat penting dalam semua tipe rangkaian digital. Karena keluaran flip-flop diubah hanya oleh pulsa clock yang datang sesudah masukan berubah, maka penghilangan pulsa clock (tegangan catu tetap ada) tidak mengubah keluaran flip-flop selama kondisi ini terjaga. Karena itu, setiap flip-flop dapat dipakai untuk menyimpan digit biner (bit) selama daya masih dikenakan dan pulsa-pulsa clock ditahan. Seperangkat bit dapat disimpan dalam register, dengan satu flip-flop untuk setiap bit.



Gambar 4. Shift Register

Sumber : (Prima Lecture,2016:5)

2.6 Arduino Uno R3

Arduino Uno R3 adalah papan pengembangan mikrokontroler yang berbasis chip ATmega328P. Arduino Uno memiliki 14 *digital pin input / output* (atau biasa ditulis I/O, dimana 14 pin diantaranya dapat digunakan sebagai *output PWM* antara lain pin 0 sampai 13), 6 pin input *analog*, menggunakan *crystal* 16 MHz antara lain pin A0 sampai A5, koneksi *USB*, jack listrik, *header ICSP* dan tombol *reset*. Hal tersebut adalah semua yang diperlukan untuk mendukung sebuah rangkaian mikrokontroler (Sigit Nugrorho,2017:12). Berikut pada tabel 2.1 diterangkan tentang spesifikasi *arduino uno r3*.

Tabel 1. Speifikasi *Arduino Uno R3*

<i>Chip mikrokontroller</i>	<i>ATmega328P</i>
Tegangan operasi	5V
Tegangan <i>input</i>	7V - 12V
Tegangan <i>input (limit, via jack DC)</i>	6V - 20V
Digital <i>I/O</i> pin	14 buah, 6 diantaranya menyediakan PWM
<i>Analog Input</i> pin	6 buah
Arus DC per pin <i>I/O</i>	20 mA
Arus DC pin 3.3V	50 mA
<i>SRAM</i>	2 KB
<i>Clock speed</i>	16 Mhz
Dimensi	68.6 mm x 53.4 mm

Sumber : (Penulis)

Dibawah ini merupakan bagian – bagian dari *arduino uno R3* :

A. Power (USB/Barrel Jack)

Untuk dapat mengaktifkan papan *arduino* diperlukan sumber listrik. *Arduino Uno* dapat diaktifkan dari kabel *USB* yang bersumber dari komputer atau dari *power supply* yang berdiri sendiri, atau baterai. Pada gambar 2 port *USB* dan *jack power supply*/baterai. Koneksi *USB* juga digunakan untuk mengkomunikasikan kode-kode dari komputer ke papan

arduino. Sedangkan rekomendasi tegangan untuk *arduino* berkisar antara 6V sampai dengan 12V.

B. Pin (5V, 3.3V, GND, *Analog*, *Digital*, PWM, AREF)

Pin pada *arduino* adalah tempat dimana kita menghubungkan kabel-kabel untuk membangun sebuah sirkuit (rangkaian elektronik) yang terhubung dengan "*breadboard*". Biasanya berbentuk plastik hitam tempat menancapkan kabel langsung ke papan *arduino*. *Arduino* mempunyai beberapa jenis pin, yang masing-masing diberi label di papannya dan digunakan untuk fungsi yang berbeda.

- 1 GND: disebut juga "*Ground*". Ada beberapa pin ground pada papan *arduino* yang dapat digunakan untuk rangkaian elektronik yang akan dibuat.
- 2 5V dan 3,3V: Sumber tegangan 5 VDC dan 3,3 VDC yang dapat digunakan untuk rangkaian elektronik yang kita buat.
- 3 *Analog: Input analog* (pin A0 sampai dengan A5). Pin ini dapat membaca sinyal *analog* (seperti sensor temperatur dan lain sebagainya) dan mengkonversikannya menjadi sinyal *digital*.
- 4 *Digital*: disebelah *analog* pin ada pin *digital* (pin 0 sampai dengan 13 pada papan *andruino uno R3*). Pin ini dapat digunakan untuk *input digital* seperti tombol tekan atau difungsikan sebagai keluaran (*output*) *digital* seperti menhidupkan LED.
- 5 PWM: kita dapat melihat tanda (~) disamping tanda pin *digital* (3, 5, 6, 9, 10, dan 11 pada papan *Arduino Uno R3*). Pin ini bertindak sebagai pin *digital* normal, tetapi juga dapat digunakan untuk *Pulse-*

Width Modulation (PWM), yaitu mampu mensimulasikan keluaran *analog* (seperti mengatur cahaya *LED*).

6 AREF: Singkatan *Analog Referensi*. Digunakan untuk mengatur tegangan referensi eksternal (antara 0 dan 5 *Volt*) sebagai batas atas untuk pin *input analog*.

C. Reset Button: menekan tombol ini akan menghubungkan rangkaian *arduino* sementara ke *ground*, dan menstart ulang kode-kode yang ada untuk dijalankan kembali. Biasanya digunakan untuk mengetes program yang dibuat.

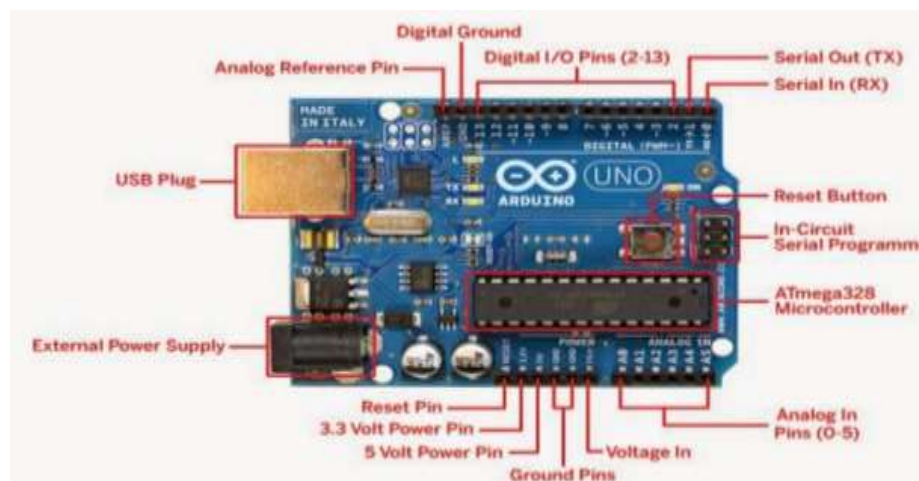
D. Power LED Indicator: Tepat di bawah dan di sebelah kanan kata “*UNO*” pada papan sirkuit, ada *LED* kecil di samping kata “*ON*”. *LED* ini harus menyala setiap kali *Arduino* dipasang ke sumber listrik. Jika lampu ini tidak menyala, ada sesuatu yang salah/rusak.

E. TX, RX, dan LED: *TX* adalah singkatan *Transmitter* (mengirimkan), *RX* adalah singkatan *Receiver* (menerima). *LED* ini akan memberi beberapa indikasi visual yang bagus setiap kali *Arduino* menerima atau mengirimkan data (seperti ketika sedang loading program baru ke papan).

F. Main IC: adalah *IC*, atau *Integrated Circuit*. Merupakan *central* dari *Arduino*. *IC* utama pada *Arduino* berbeda pada setiap jenis modul *arduino*, tetapi biasanya dari jenis *ATmega*, *IC* dari perusahaan *ATMEL*. Informasi ini biasanya dapat ditemukan di sisi atas *IC*. Untuk keterangan lebih detail dapat dilihat pada *datasheet* (lembar data) nya.

G. Voltage Regulator: *Regulator* tegangan, untuk mengatur dan menjaga tegangan *input arduino* tetap stabil. Jangan memberi catu tegangan lebih besar dari 20 *Volt*.

Papan *Arduino Uno R3* dapat mengambil daya dari *USB port* pada komputer dengan menggunakan *USB charger* atau dapat pula mengambil daya dengan menggunakan suatu *AC adapter* dengan tegangan 9V. Jika tidak terdapat *power supply* yang melalui *AC adapter*, maka papan *arduino* akan mengambil daya dari *USB port*. Tetapi apabila diberikan daya melalui *AC adapter* secara bersamaan dengan *USB port* maka papan *arduino* akan mengambil daya melalui *AC adapter* secara otomatis.



Gambar 5. Arduino Uno R3

Sumber : (Hadi Santosa,2012)

2.7 Arduino IDE

Software arduino yang digunakan adalah *driver* dan *IDE*, walaupun masih ada beberapa *software* lain yang sangat berguna selama pengembangan *arduino*. *IDE* atau *Integrated Development Environment* suatu program khusus untuk suatu komputer agar dapat membuat suatu rancangan atau sketsa program untuk papan

arduino.IDE arduino merupakan *software* yang sangat canggih ditulis dengan menggunakan *java*. *IDE arduino* terdiri dari:

A. Soket USB

Soket *USB* adalah soket untuk kabel *USB* yang disambungkan ke komputer atau laptop. Berfungsi untuk mengirimkan program ke *Arduino* dan juga sebagai port komunikasi serial. (Hary Santosa,2012)

B. Input / Oustput Digital

Input/Output Digital atau *digital* pin adalah pin-pin untuk menghubungkan *arduino* dengan komponen atau rangkaian *digital*. Misalnya kalau ingin membuat *LED* berkedip, *LED* tersebut bisa dipasang pada salah satu pin *I/O digital* dan *ground*. Komponen lain yang menghasilkan *output digital* atau menerima input *digital* bisa disambungkan ke pin-pin ini.

C. Input Analog

Input Analog atau *analog* pin adalah pin-pin yang berfungsi untuk menerima sinyal dari komponen atau rangkaian *analog*. Misalnya dari potensiometer, sensor suhu, sensor cahaya, dan sebagainya.

D. Catu Daya

Pin-pin catu daya adalah pin yang memberikan tegangan untuk komponen atau rangkaian yang dihubungkan dengan *Arduino*. Pada bagian catu daya ini terdapat juga pin *Vin* dan *Reset*. *Vin* digunakan untuk memberikan tegangan langsung kepada *arduino* tanpa melalui tegangan *USB* atau *adaptor*. *Reset* adalah pin untuk memberikan sinyal reset melauai tombol atau rangkaian eksternal. Pada saat terkoneksi dengan komputer dari *USB*, *Arduino* mendapatkan supply power sebesar 5V dari *port USB* komputer. Hal ini

sangat berguna ketika kita melakukan pemrograman dan pengujian terhadap percobaan yang sedang kita buat, karena tidak perlu menambahkan *power supply* eksternal pada *arduino* yang kita pakai sebagai percobaan.

Namun perlu diperhatikan bahwa kapasitas *power supply* dari *USB* sangatlah terbatas, hanya 500 mA saja, sehingga jika proyek kita memerlukan daya yang lebih besar misalnya untuk menggerakkan motor maka kita tetap membutuhkan *power supply* eksternal.

E. Baterai / Adaptor

Soket baterai atau adaptor digunakan untuk menyuplai *arduino* dengan tegangan dari baterai/adaptor 9V pada saat *arduino* sedang tidak disambungkan ke komputer. Kalau *arduino* sedang disambungkan ke komputer melalui *USB*, *arduino* mendapatkan *supply* tegangan dari *USB*, jadi tidak perlu memasang baterai/adaptor saat memprogram *arduino*.

Dalam bahasa pemrograman *arduino* ada tiga bagian utama yaitu struktur, variabel dan fungsi:

1. Kerangka Program

Kerangka program *arduino* sangat sederhana, yaitu terdiri atas dua blok.

Blok pertama adalah *void setup()* dan blok kedua adalah *void loop*.

a. Blok Void setup ()

Berisi kode program yang hanya dijalankan sekali sesaat setelah *arduino* dihidupkan atau di-reset. Merupakan bagian persiapan atau instalasi program.

b. Blok *Void loop()*

Berisi kode program yang akan dijalankan terus menerus. Merupakan tempat untuk program utama.

c. Sintaks Program

Baik blok void *setup loop ()* maupun blok function harus diberi tanda kurung kurawal buka “{“ sebagai tanda awal program di blok itu dan kurung kurawal tutup “}” sebagai tanda akhir program.

2. Variabel

Sebuah program secara garis besar dapat didefinisikan sebagai instruksi untuk memindahkan angka dengan cara yang cerdas dengan menggunakan sebuah variabel.

3. Fungsi

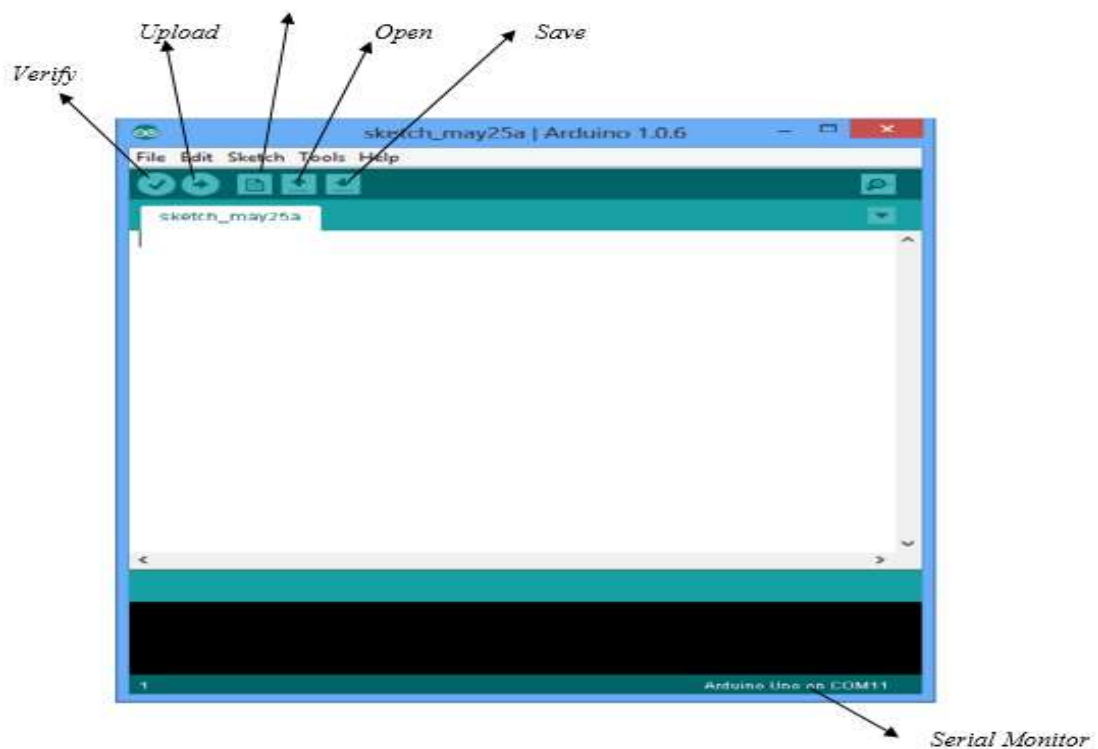
Pada bagian ini meliputi fungsi input output *digital*, input output *analog*, *advanced I/O*, fungsi waktu, fungsi matematika serta fungsi komunikasi.

Pada proses *uploader* dimana pada proses ini mengubah bahasa pemrograman yang nantinya *dcompile* oleh *avr-gcc (avr-gcc compiler)* yang hasilnya akan disimpan kedalam papan *arduino*.

Avr-gcc compiler merupakan suatu bagian penting untuk *software* bersifat *open source*. Dengan adanya *avr-gcc compiler*, maka akan membuat bahasa pemrograman dapat dimengerti oleh mikrokontroler. Proses terakhir ini sangat penting, karena dengan adanya proses ini maka akan membuat proses pemrograman mikrokontroler menjadi sangat mudah. Berikut ini merupakan gambaran siklus yang terjadi dalam melakukan pemrograman *arduino*:

- a. Koneksikan papan *arduino* dengan komputer melalui *USB port*.

- b. Tuliskan sketsa rancangan suatu program yang akan dimasukkan ke dalam papan *arduino*.
- c. *Upload* sketsa program ke dalam papan *arduino* melalui kabel *USB* dan kemudian tunggu beberapa saat untuk melakukan *restart* pada papan *arduino*. Papan *Arduino* akan mengeksekusi rancangan sketsa program yang telah dibuat dan di-*upload* ke papan *Arduino* (Wike Febriani, 2017:13). *IDE arduino* dapat dilihat pada gambar dibawah ini.



Gambar 6. Arduino IDE

Sumber : (Sigit Nugroho, 2017:4)

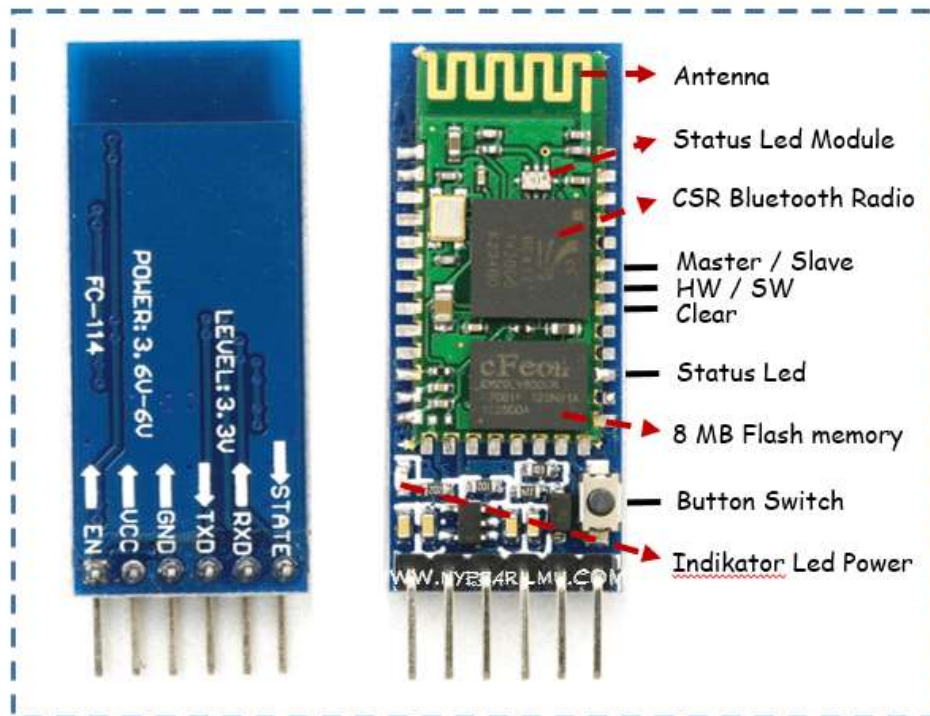
- a. Icon menu *verify* yang bergambar ceklis berfungsi untuk mengecek program yang ditulis apakah ada yang salah atau *error*.

- b. Icon menu **upload** yang bergambar panah ke arah kanan berfungsi untuk memuat / *transfer* program yang dibuat di *software arduino* ke *hardware arduino*.
- c. Icon menu **New** yang bergambar sehelai kertas berfungsi untuk membuat halaman baru dalam pemrograman.
- d. Icon menu **Open** yang bergambar panah ke arah atas berfungsi untuk membuka program yang disimpan atau membuka program yang sudah dibuat dari pabrikan *software arduino*.
- e. Icon menu **Save** yang bergambar panah ke arah bawah berfungsi untuk menyimpan program yang telah dibuat atau dimodifikasi.
- f. Icon menu **serial monitor** yang bergambar kaca pembesar berfungsi untuk mengirim atau menampilkan serial komunikasi data saat dikirim dari *hardware arduino* (Sigit Nugroho, 2017:5).

2.8 Bluetooth HC-05

Bluetooth adalah protokol komunikasi *wireless* yang bekerja pada frekuensi radio 2.4 GHz untuk pertukaran data pada perangkat bergerak seperti PDA, laptop, HP, dan lain-lain . Salah satu hasil contoh modul *Bluetooth* yang paling banyak digunakan adalah tipe *HC-05*. modul *bluetooth HC-05* merupakan salah satu modul *bluetooth* yang dapat ditemukan dipasaran dengan harga yang relatif murah. Modul *bluetooth HC-05* terdiri dari 6 pin konektor, yang setiap pin konektor memiliki fungsi yang berbeda – beda (Linarti, 2014:5). Untuk gambar modul *bluetooth* dapat dilihat pada gambar 3 dibawah ini.

MODULE BLUETOOTH HC-05



Gambar 7. Modul Bluetooth HC 05

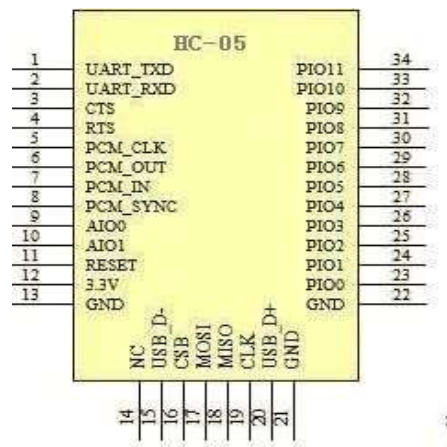
Sumber : (Linarti, 2014:5)

Deskripsi modul HC-05 :

1. Level tegangan kerja 3.3V.
2. Kebutuhan Arus : Pairing 20-30mA. Setelah Pair: 8mA
3. Frekuensi yang digunakan : 2.5 GHz
4. Pin KEY : Jika dibawa TINGGI sebelum PERINTAH diterapkan, pasukan AT Command Pengaturan Mode. LED berkedip perlahan (2 detik) (Biasanya jarang ditemui Pin KEY tersebut).
5. Pin VCC: +5 Power.
6. Pin GND: *Ground*/Tanah.
7. Pin TXD: Transmit data Serial dari HC-05 untuk Serial Penerima Arduino. . CATATAN: 3.3V Tegangan TINGGI: OK untuk Arduino.

8. Pin RXD: Menerima data Serial dari Arduino Serial Transmit.
9. STATE : Memberitahu jika terhubung atau tidak.
10. Antenna : Pengirim sinyal *bluetooth*.
11. *Led Module* : Memberikan informasi HC-05 hidup (indikator *led power*) dan keadaan bluetooth terkoneksi atau tidak (*Status led module*).

Modul *bluetooth HC-05* dengan *supply* tegangan sebesar 3,3 V ke pin 12 modul *bluetooth* sebagai *VCC*. Pin 1 pada modul *bluetooth* sebagai *transmitter*. kemudian pin 2 pada *bluetooth* sebagai receiver. Berikut merupakan konfigurasi pin *bluetooth HC-05* ditunjukkan pada gambar 4 dibawah ini:



Gambar 8. Konfigurasi Modul Bluetooth HC 05

Sumber : (Linarti, 2014:6)

Module Bluetooth HC-05 merupakan *module Bluetooth* yang bisa menjadi *slave* ataupun *master* hal ini dibuktikan dengan bisa memberikan notifikasi untuk melakukan *pairing* perangkat lain, maupun perangkat lain tersebut yang melakukan *pairing* ke *module Bluetooth CH-05*. Untuk mengeset perangkat *Bluetooth* dibutuhkan perintah-perintah *AT Command* yang mana perintah *AT*

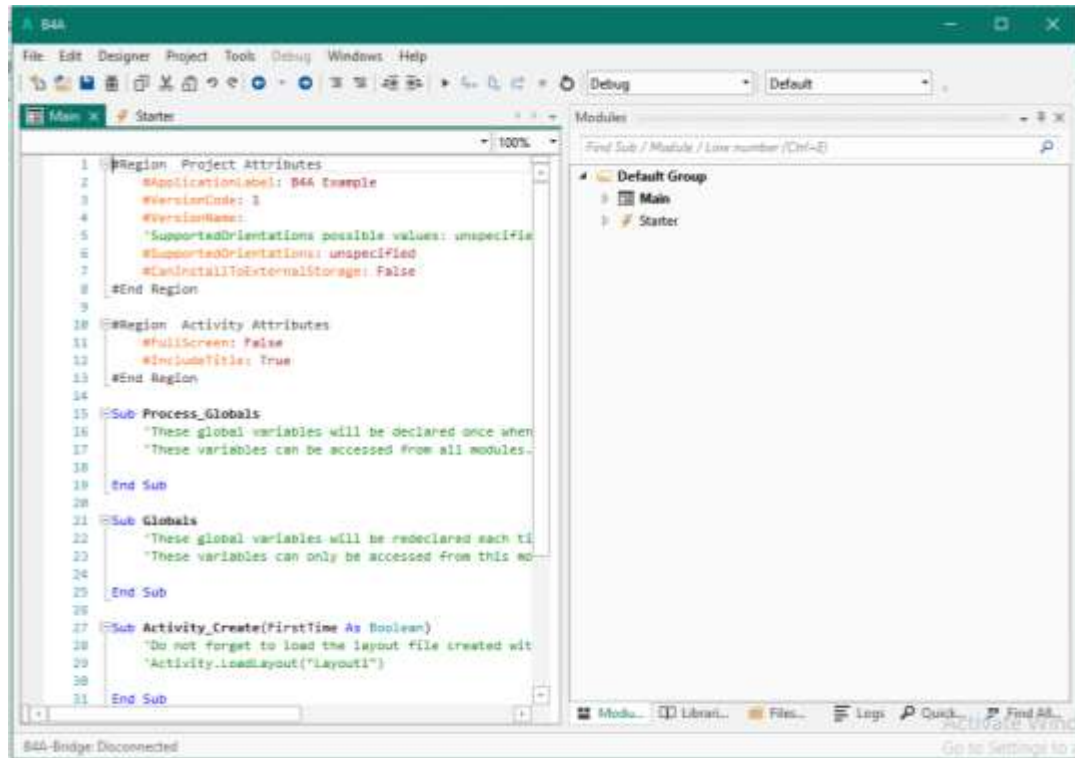
Command tersebut akan di respon oleh perangkat *Bluetooth* jika modul *Bluetooth* tidak dalam keadaan terkoneksi dengan perangkat lain.

2.9 B4A

Definisi *Basic4Android* adalah *development tool* sederhana yang *powerfull* untuk membangun aplikasi *Android*. Bahasa *Basic4Android* mirip dengan bahasa *Visual Basic* dengan tambahan dukungan untuk objek. Aplikasi *Android* (*APK*) yang *dicompile* oleh *Basic4Android* adalah aplikasi *Android native/asli* dan tidak ada *extra runtime* seperti di *Visual Basic* yang ketergantungan *filemsvbm60.dll*, yang pasti aplikasi yang *dicompile* oleh *Basic4Android* adalah *No Dependencies* tidak ketergantungan file oleh lain). *IDE Basic4Android* hanya fokus pada *development Android* (Maimunah, 2017:52). Adapun beberapa kelebihan dan fitur yang dimiliki oleh *Basic4Android* adalah :

- A. *Simple dan Powerfull RAD (Rapid Application Development) tools* untuk mengembangkan aplikasi *native android* tanpa harus menghabiskan waktu mempelajari pemrograman *Eclipse/Java*.
- B. *IDE (Integrated Development Environment)* lengkap yang fokus 100% pada pengembangan aplikasi *android*.
- C. Dikompil ke dalam *native bytecode*, tidak ada tambahan *runtime library* yang dibutuhkan. *APK file* yang dihasilkan sama dengan *APK* yang dihasilkan ketika anda membuat aplikasi menggunakan *Eclipse/Java*.
- D. Performa dari aplikasi yang dihasilkan sama dengan aplikasi yang dibuat dengan *Java*.
- E. *Object Oriented Programming Language* (sintaks sama dengan *Visual Basic*).

- F. Tidak perlu menggunakan *XML* untuk *layout*.
- G. Sangat ekstensibel dengan dukungan *custom java library*.
- H. Satu-satunya visual editor untuk android yang benar-benar *WYSIWYG*.
Visual editor mendukung banyak *screen* dan *resolusi*.
- I. Mempunyai fitur *designer script*, membuat anda dapat dengan mudah membuat *layout* yang *sophisticated*.
- J. Fitur *Basic4Android UI Cloud Services* dapat digunakan untuk mengetes aplikasi pada smartphone atau tablet asli lewat *cloud*.
- K. Mendukung semua *android phone* dan tablet dari versi 1.6 sampai dengan 6x.
- L. *IDE* yang modern dengan dukungan *autocomplete*, *built-in* dokumentasi, *internal index*, dan fitur *advance* yang lain.
- M. Fitur *step by step debugger* yang *powerfull*.
- N. Fitur *built-in code obfuscation*
Mendukung semua *android core* (*GPS*, *SQL Database*, *Widgets*, *Live*



Gambar 8. Basic4Android

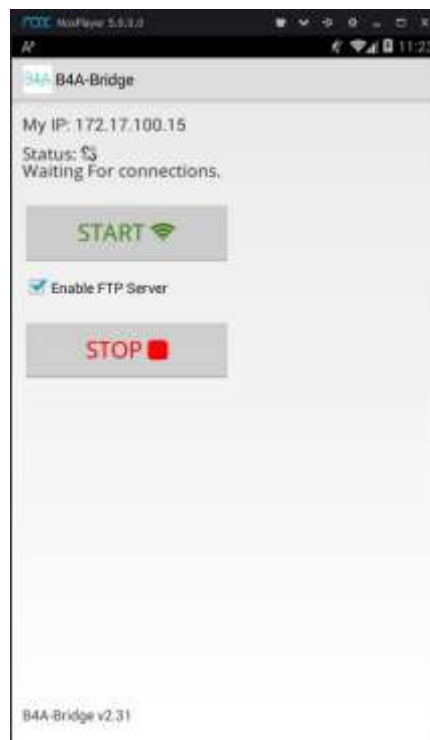
Sumber : (Penulis)

2.10 B4A Bridge

B4A Bridge adalah aplikasi aplikasi pada *android* yang menghubungkan B4A dengan perangkat *smartphone*. *B4A Bridge* terbuat dari dua komponen. Satu komponen berjalan pada perangkat *smartphone* komponen kedua yang merupakan bagian dari *IDE* untuk terhubung dan berkomunikasi dengan perangkat *smartphone*. Koneksi dilakukan melalui jaringan lokal atau dengan koneksi Bluetooth melalui IP. Setelah terhubung, *B4A-Bridge* mendukung semua fitur *IDE* yang meliputi menginstal aplikasi, melihat *log*, *debugging* dan perancang visual (mengambil screenshot tidak didukung).

Android tidak mengizinkan aplikasi untuk menginstal aplikasi lain secara diam-diam, oleh karena itu ketika menjalankan aplikasi menggunakan *B4A-Bridge*, akan melihat dialog yang meminta untuk menyetujui penginstalan.

Pengguna selalu disarankan untuk menggunakan perangkat nyata bukan *emulator Android* yang sangat lambat dibandingkan dengan perangkat nyata (terutama dengan instalasi aplikasi). Namun tidak semua perangkat mendukung *debug ADB* ini adalah alasan untuk alat *B4A-Bridge*.



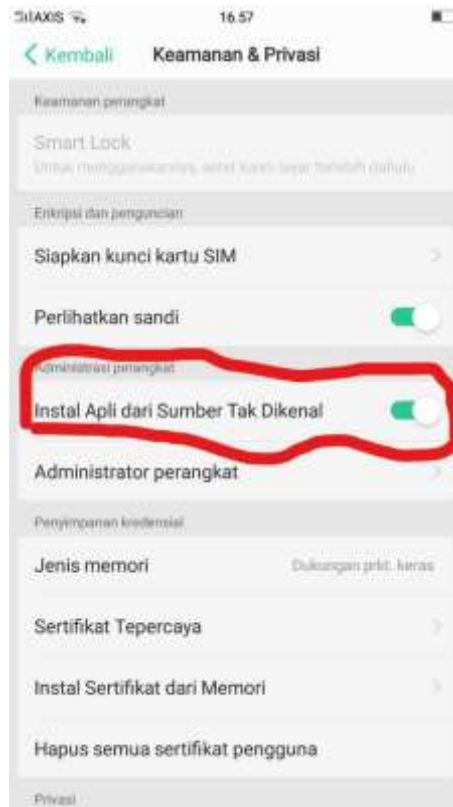
Gambar 9. Interface B4A Bridge

Sumber : (playstore)

2.11 Keamanan & privasi

Pengguna di sarankan untuk mengecek pengaturan keamanan & privasi dari hanpone android terlebih dahulu dari pengaturan, pengaturan tambahan lalu keamanan & privasi, dan sewaktu penginstalan aplikasi pastikan kita harus

mengecek untuk instal aplikasi sumber tidak dikenal pastikan harus aktif, jika tidak aktif maka aplikasi tidak bisa terinstal, contoh seperti yang ada di gambar sebagai berikut:



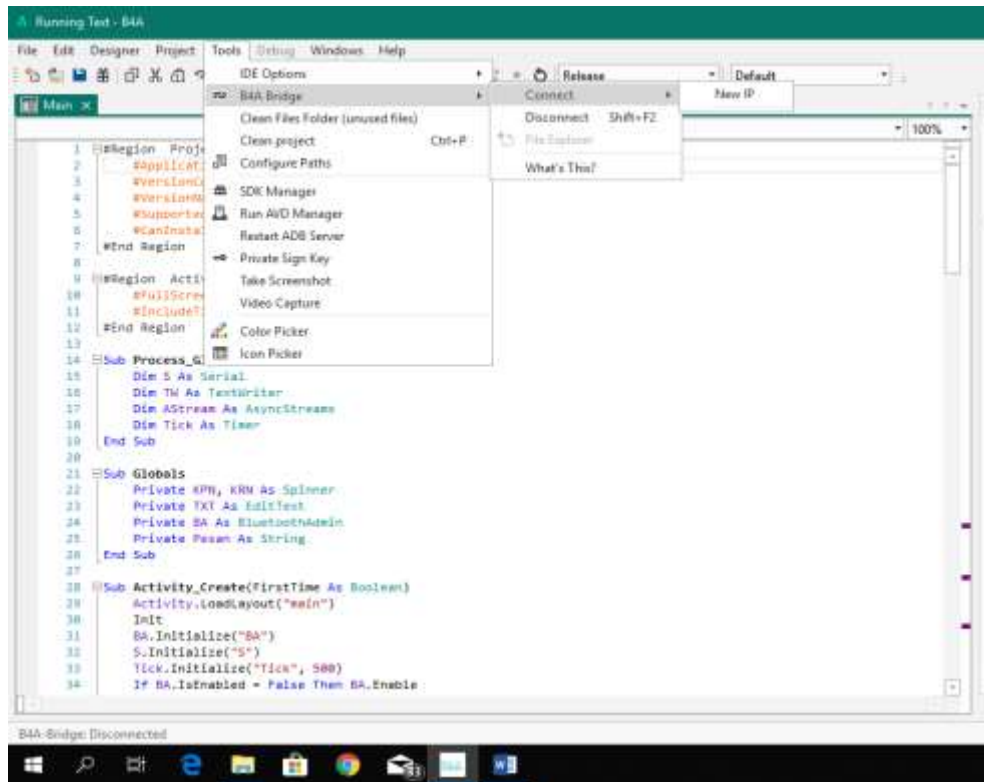
Gambar 10. Keamanan dan privasi

Sumber : (ANDROID)

2.12 COMFILE

COMFILE merupakan salah satu cara metode untuk menginstal aplikasi *running text* ke android melalui *B4A Bridge*, yaitu ada beberapa tahap penginstalan, terlebih dahulu kita membuka program *Running texts* yang sudah terisi program yang ada di *B4A(Basic4android)* lalu mengkoneksikan jaringan dari android ke PC dan jangan lupa membuka aplikasi *B4A Bridge* untuk koneksi IP yang ada pada

android ,jika semua sudah selesai lalu ke menu *tools - B4A Bridge – connect – New IP* setelah itu *RUN* ,dan akan otomatis akan terinstal ke *android*.



Gambar 11. CONFILE B4A ke Android

Sumber : (Penulis)

2.13 Memulai Program Basic4Android

Awal memulai program Basic4Android akan menampilkan *template coding* seperti pada gambar 10. Pada bagian kiri atas ada dua modul Tab yaitu *main* dan *starter* yang berguna untuk menyatakan variabel *ProcessGlobal* dan variabel ini dapat diakses dari semua modul dalam *project*, *main* diartikan sebagai aktivitas awal dan tidak dapat dihapus.

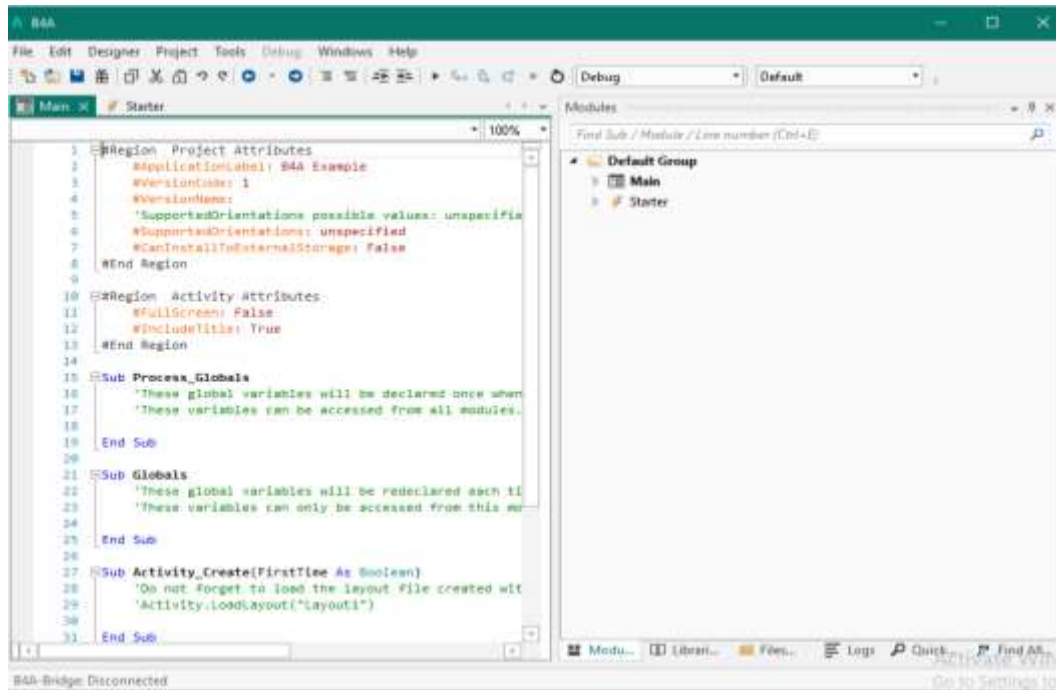
Variabel dapat berupa global atau lokal. Variabel lokal adalah variabel yang dideklarasikan di dalam sub selain *Process_Globals* atau *Globals*. Variabel lokal bersifat lokal ke sub atau modul yang berisi. Setelah sub berakhir, variabel-variabel ini tidak ada lagi. Variabel global dapat diakses dari semua subs dalam modul yang mengandung variabel global.

Ada dua jenis variabel global. Variabel proses (dapat diakses dari semua modul) dan variabel aktivitas (dapat diakses dari satu modul). Memproses variabel global Variabel-variabel ini hidup selama proses berlangsung. pengguna harus menyatakan variabel-variabel ini sebagai *Public* di dalam *Sub Process_Globals* dari *Starter Service* seperti. *Sub Process_Globals* Variabel global ini akan dideklarasikan satu kali ketika aplikasi dimulai. Variabel-variabel ini dapat diakses dari semua modul. Publik *MyVariable = "Test" As String* Sub ini disebut satu kali saat proses dimulai. Variabel-variabel ini adalah satu-satunya variabel "publik". Yang berarti bahwa mereka dapat diakses dari modul lain juga.

Ada juga rutinitas *Process_Globals* di setiap modul Aktivitas. Jika pengguna memerlukan variabel, hanya berlaku di Aktivitas, yang hanya diinisialisasi satu kali saat program diluncurkan, pengguna harus menempatkannya dalam rutinitas *Activity_Globals Activity* (ini berlaku untuk semua aktivitas, bukan hanya aktivitas pertama).

Namun, tidak semua jenis objek dapat dinyatakan sebagai variabel proses alasannya adalah bahwa pengguna tidak ingin memegang referensi ke objek yang harus dihancurkan bersama dengan aktivitas. Dengan kata lain, ketika aktivitas tersebut dihancurkan, semua proses yang terkandung dalam aktivitas tersebut juga dihancurkan. Jika pengguna tidak melakukan ini, dan terus referensi ke tampilan

setelah Kegiatan dihancurkan, pengumpul sampah tidak akan dapat membebaskan sumber daya dan akan memiliki kebocoran memori. *Compiler* akan memberlakukan persyaratan ini.



Gambar 12. Basic4Android

Sumber : (Penulis)

2.14 Rilis dan Rilis (obfuscated) mode

Untuk mendistribusikan aplikasi yang dibuat maka pengguna harus mengkompilasinya dengan Rilis Kode *debugger* tidak akan ditambahkan ke *file apk*. Release (*obfuscated*) Kode *debugger* tidak akan ditambahkan ke *file apk*, tetapi *file* program akan dimodifikasi.

Selama kompilasi *B4A* menghasilkan kode *Java* yang kemudian dikompilasi dengan *compiler Java* dan dikonversi ke *Dalvik* (format kode *byte Android*). Ada alat yang memungkinkan dekompilasi kode *byte Dalvik* ke dalam kode *Java*. Tujuan dari kebingungan adalah untuk membuat kode dekompilasi

menjadi kurang terbaca, lebih sulit untuk dipahami dan membuatnya lebih sulit untuk mengekstrak *string* seperti kunci akun pengembang.

2.15 Smartphone

Telepon pintar (*smartphone*) adalah telepon genggam yang mempunyai kemampuan tingkat tinggi, kadang-kadang dengan fungsi yang menyerupai komputer. Belum ada standar pabrik yang menentukan arti telepon pintar. (Elcom, 2011)

Bagi beberapa orang, telepon pintar merupakan telepon yang bekerja menggunakan seluruh perangkat lunak sistem operasi yang menyediakan hubungan standar dan mendasar bagi pengembang aplikasi. Bagi yang lainnya, telepon cerdas hanyalah merupakan sebuah telepon yang menyajikan fitur canggih seperti surel (surat elektronik), internet dan kemampuan membaca buku elektronik (*e-book*) atau terdapat papan ketik (baik sebagaimana jadi maupun dihubung keluar) dan penyambung VGA. Dengan kata lain, telepon cerdas merupakan komputer kecil yang mempunyai kemampuan sebuah telepon.

2.15 Android

Android adalah sistem operasi yang berbasis *Linux* untuk telepon seluler seperti telepon pintar dan komputer tablet. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak.

Fitur yang tersedia di *Android* adalah:

1. Kerangka aplikasi itu memungkinkan penggunaan dan penghapusan komponen yang tersedia.
2. Mesin virtual dioptimalkan untuk perangkat *mobile*.

3. Grafik di 2D dan grafis 3D berdasarkan pustaka *OpenGL*.

BAB III

ANALISIS DAN PERANCANGAN SISITEM

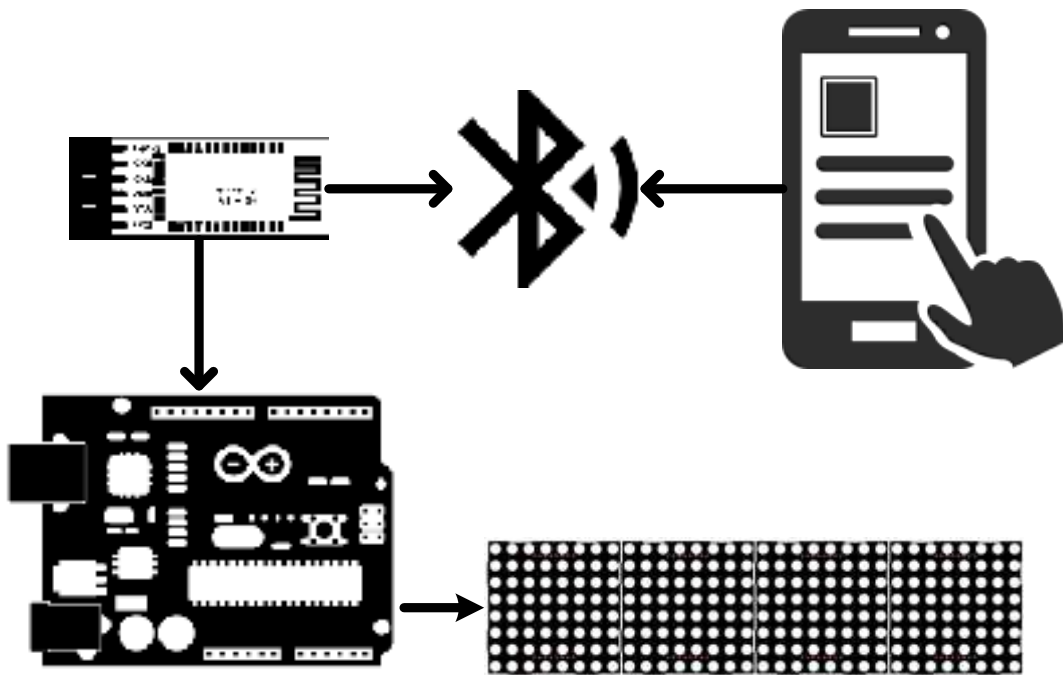
3.1 Analisis Sistem

Pembangunan sistem papan pengumuman digunakan untuk mengendalikan teks pada modul *dot matrix* 8x32, sistem ini digabungkan dengan sebuah *mikrokontroler arduino* yang di program dengan bahasa pemograman C. Dengan pembangunan sistem ini pengguna tidak harus mengontrol papan pengumuman dengan datang ke tempat pusat pengontrolan papan pengumuman, pengguna dapat mengontrol, memonitor serta dapat mengatur kecerahan dan kecepatan teks sesuai dengan kebutuhan, sistem ini di kendalikan dengan sebuah aplikasi yang terkoneksi pada perangkat android untuk setiap pengaturan serta pengontrolan papan, sehingga sistem dapat dikendalikan dengan jarak jauh. Berikut Analisis sistem yang ada pada sistem papan pengumuman elektronik:

1. Menginput Text yang ada Pada perangkat *Android* yang akan di Kirim Ke *Mikrokontroler arduino* dan di tampilkan pada *Led dot matrix* 8x32.
2. Sistem yang dibuat dapat Mengatur kecepatan teks sesuai dengan kebutuhan.
3. Sistem yang dibuat dapat mengatur kecerahan text pada *dot matrix* 8x32.
4. Sisitem papan Pengumuman Dikontrol pada aplikasi Berbasis *android* yang dihubungkan dengan metode *Bluethooth*.

3.2. Analisis Arsitektur Perancangan Sistem

Analisis arsitektur sistem merupakan gambaran sistem yang akan dibangun, aplikasi papan Pengumuman digital ini akan berkomunikasi dengan alat *Mikrokontroler* yang di jembatani oleh koneksi *bluetooth* sebagai sarana pengiriman data. Adapun rancangan sistem yang akan dibangun dapat dilihat pada gambar 13 berikut.



Gambar. 13 Rancangan Sistem

Sumber : (Penulis)

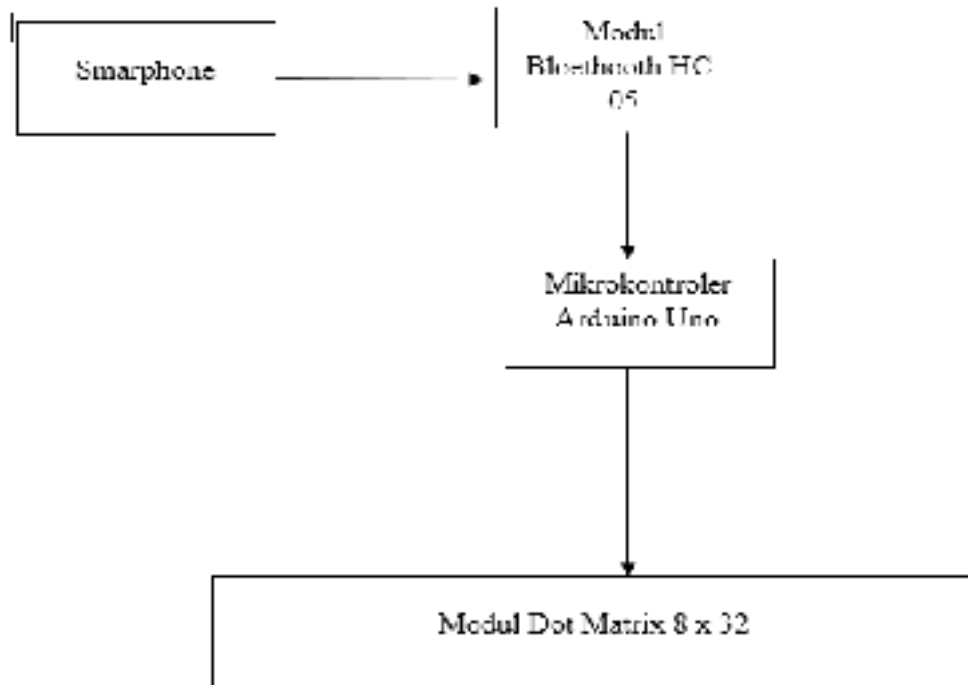
Gambar diatas menggambarkan sistem yang akan dibangun, dimana perangkat android akan mengirim data ke *mikrokontroler arduino uno* dengan koneksi bluetooth kemudian data akan diproses di *mikrokontroler* dan selanjutnya akan di tampilkan pada *led dot matrix 8x32*.

3.3. Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan. Kebutuhan non fungsional yang dibutuhkan untuk membangun ulang sistem ini terdiri dari dua hal, yaitu : kebutuhan perangkat keras dan kebutuhan perangkat lunak.

3.4 Analisis Perancangan Perangkat Keras

Dibawah ini adalah diagram blok dari alat penampil papan pengumuman digital berbasis *mikrokontroler arduino uno* dengan metode *transfer data bluethooth* dapat dilihat pada ambar 14.



Gambar. 14 Metode Transfer data *Bluetooth*

Sumber : (Penulis)

Keterangan dari diagram blok diatas adalah:

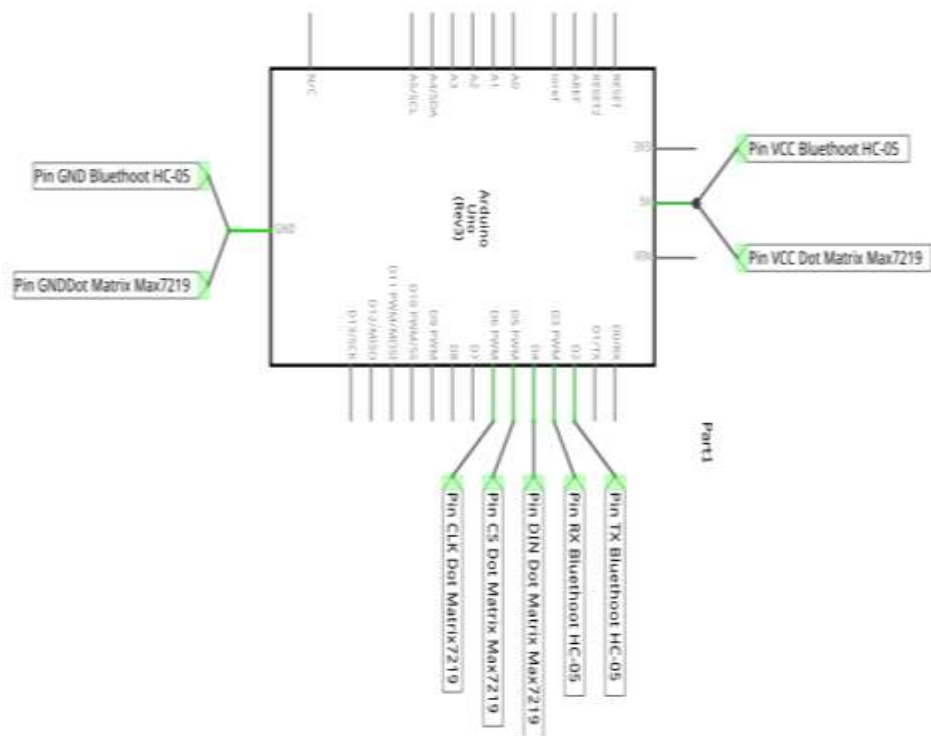
1. *Smartphone android* berfungsi sebagai *input* data yang akan di tampilkan pada modul *dot matrix* 8x32.
2. Modul *bluetooth HC-05* berfungsi sebagai penghubung antara *mikrokontoler* dengan *smartphone android*.
3. *Arduino uno* berfungsi sebagai pengolah data yang masuk dari *smartphone*, selain itu *mikrokontoler* memberi perintah kepada *modul led* 8x8 untuk menampilkan data input yang masuk melalui *smartphone* dan mikrokontoler memberi perintah terima data kepada modul *bluetooth HC-05*.

4. Modul *Led* 8x32 berfungsi untuk menampilkan data yang masuk melalui *smartphone*.

3.5. Perancangan Perangkat Keras

3.5.1. Arduino Uno

Untuk perancangan perangkat keras *arduino uno* penulis menggunakan 9 pin yaitu pin 2 pada *arduino* dihubungkan dengan pin rx pada modul *bluetooth HC-05*, pin 3 pada *arduino* dihubungkan dengan pin tx pada modul *bluetooth HC-05*. Pin 3 pada *arduino* dihubungkan dengan pin DIN pada Modul *Dot matrix 8x32*, pin 4 pada *arduino* dihubungkan dengan pin cs pada modul *Dot matrix 8x32* dan pin 5 pada *arduino* dihubungkan ke pin CLK pada modul *Dot matrix 8x32*. Pin 5v pada *arduino* dihubungkan pada pin vcc pada modul *dotmatrix 8x32* dan pin vcc modul *bluetooth HC-05*. Pin GND pada *arduino* dihubungkan pada pin GND modul *dot matrix 8x32* dan pin GND modul *bluetooth HC-05*. Berikut Gambar 15 Rangkaian perangkat keras *Arduino*.

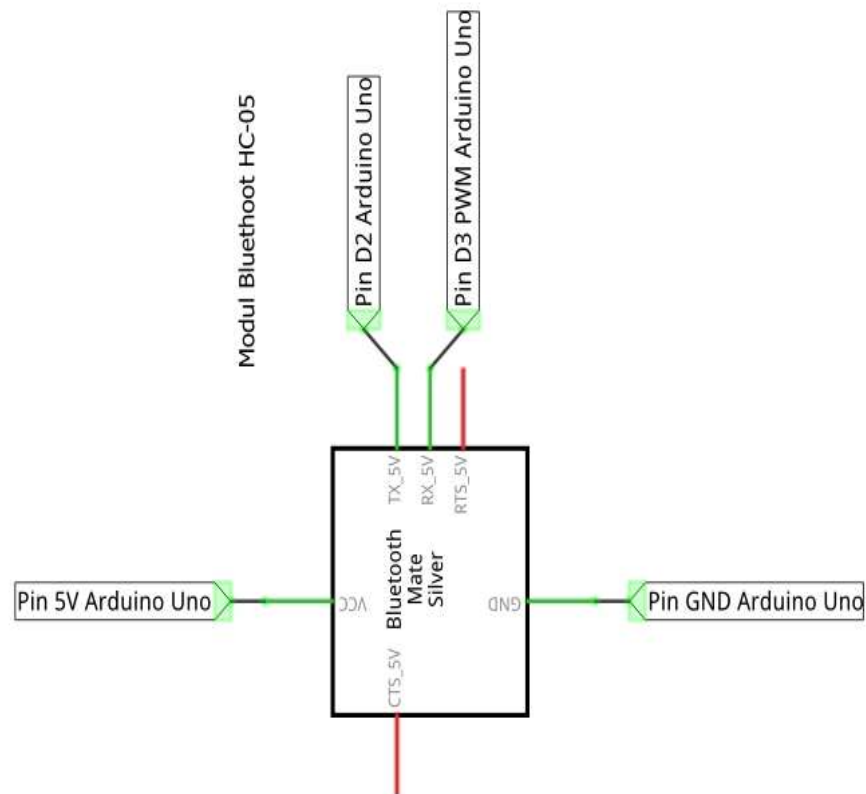


Gambar. `15 Rangkaian perangkat keras *Arduino*

Sumber : (Penulis)

3.5.2. Rangkaian Bluetooth

Untuk rangkaian *Modul Bluetooth HC-05* menggunakan 4 pin dimana pin rx pada modul *bluetooth HC-05* dihubungkan dengan pin 2 arduino dan pin tx pada modul *bluetooth hc-05* dihubungkan dengan pin 3 pada arduino . Pin vcc Pada modul *bluetooth HC-05* dihubungkan dengan pin 5v pada arduino dan pin GND pada Modul *bluetooth HC-05* dihubungkan dengan PIN GND pada *Arduino*. Berikut gambar 16 rangkaian Perangkat Keras Modul *Bluetooth HC-05*.

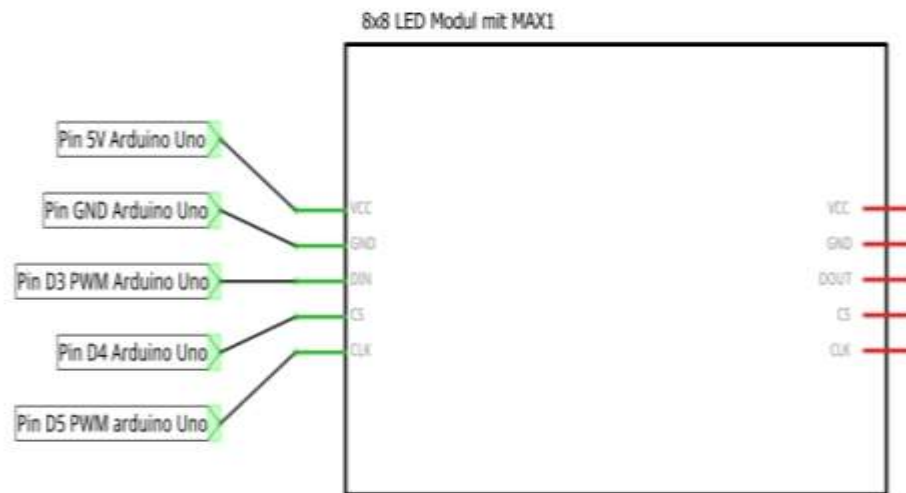


Gambar. 16 Rangkaian Bluetooth

Sumber : (Penulis)

3.5.3. Rangkaian Modul Dot Matrix 8x8

Pada Rangkaian Modul *Dot matrix 8x8* terdapat 5 pin dimana masing-masing pin dihubungkan dengan *mikrokontroler arduino*. Pin Vcc dihubungkan dengan pin 5V pada arduino dan pin GND dihubungkan dengan pin GND pada arduino. Pin DIN dihubungkan dengan pin 4 pada arduino, pin CS dihubungkan pada pin 5 arduino dan pin CLK dihubungkan pada pin 6 *arduino*. Berikut Gambar 17 Rangkaian perangkat keras modul *dot matrix 8x8*.

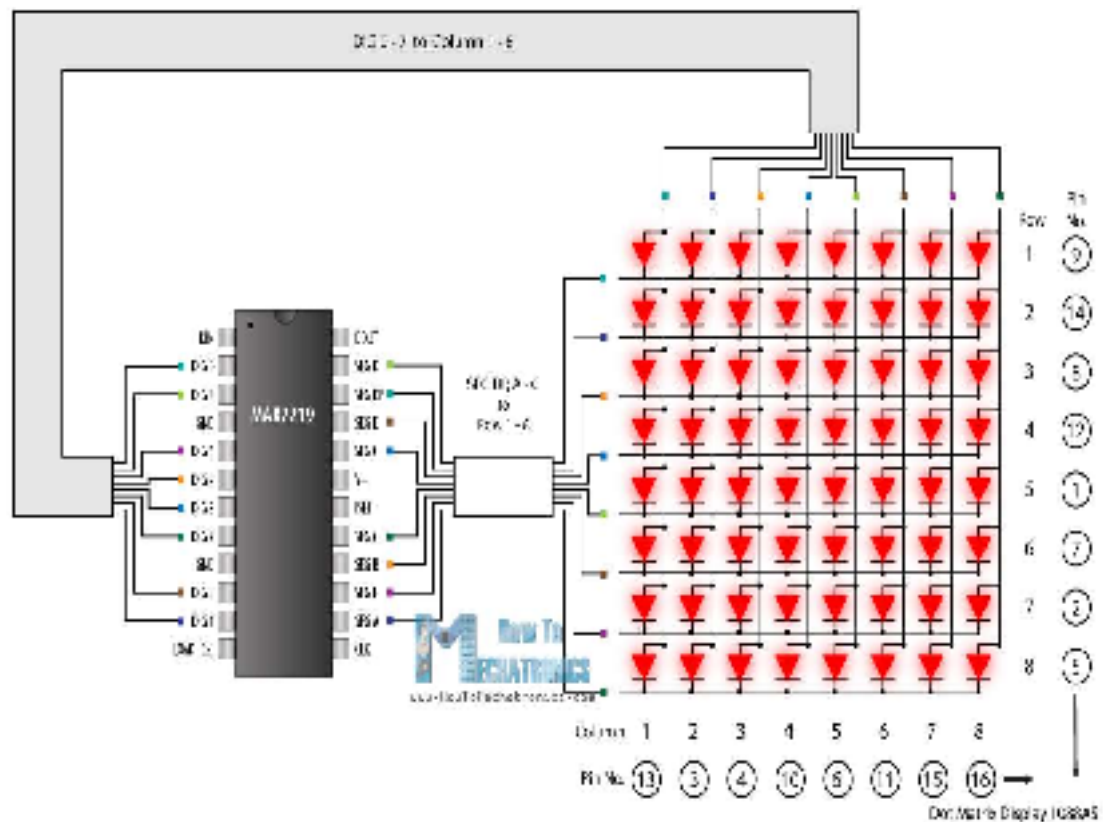


Gambar.17 Rangkaian Modul Dot Matrix 8x8

Sumber : (Penulis)

3.5.4. Pin 8 x 8 LED Matrix

Perhatikan bagaimana pin dari Matriks LED 8×8 umum diatur secara internal, jadi jika ingin membuat matriks sendiri, harus mempertimbangkannya. Perhatikan juga bahwa papan breakout umum untuk MAX7219 dilengkapi dengan resistor antara 5V dan nomor pin IC 18. Resistor digunakan untuk mengatur kecerahan atau aliran arus ke LED.



Gambar. 19 Pin 8 x 8 LED Matrix

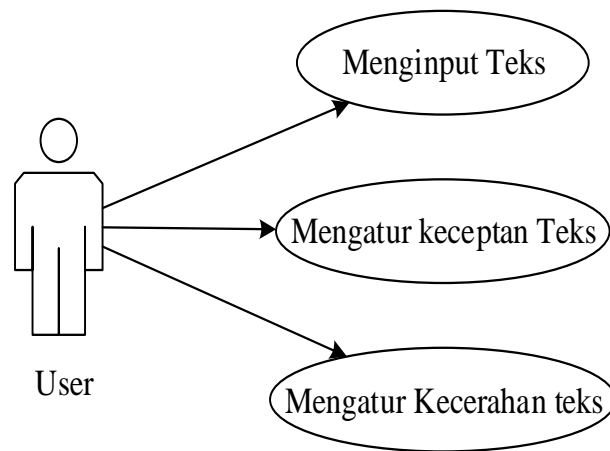
Sumber : (Dejan Nedelkovski 2017)

3.6. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sistem dan menjelaskan kebutuhan yang diperlukan agar sistem dapat belajar dengan baik serta sesuai dengan kebutuhan sistem. Analisis yang dilakukan dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Tahapan pemodelan dalam analisis tersebut antara lain mengidentifikasi aktor, pembuatan *Use Case Diagram*, *Use Case Scenario*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*.

3.6.1. Use Case Diagram

Use case diagram adalah diagram yang menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar dan menjelaskan sistem secara fungsional yang terlihat pengguna. Dari identifikasi aktor yang terlibat di atas maka *Use Case* Diagram dapat dilihat pada Gambar 18.



Gambar. 18 Use case diagram

Sumber : (Penulis)

3.6.2. Use Case Scenario

Use case scenario bertujuan untuk mendeskripsikan urutan langkah-langkah dalam proses bisnis baik yang dilakukan aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor. Berdasarkan *Use Case* Diagram pada Gambar maka *use case* scenario dijelaskan sebagai berikut :

1. Use case scenario input teks

Interaksi antara aktor pengguna dengan *use case* di jelaskan dalam *use case* skenario *input* teks pada tabel Berikut;

Tabel 2. use case skenario input teks

Scenario	
Nama Use case	Input Teks
Nama Aktor	<i>User</i>
Deskripsi	Sistem akan Menampilkan Teks yang Telah di Input
Kondisi awal	<i>User</i> Telah Menginput Teks
Aksi Aktor	Respon Sistem
1. Memasukkan Teks	
2. Menekan Tombol Kirim	
	3. sistem Menampilkan Pesan sukses
Kondidi Akhir	Teks yang Dikirim Ditampilkan

Sumber : (Penulis)

2. Use case scenario Kecepatan Teks

Interaksi antara aktor pengguna dengan *use case* di jelaskan dalam *use case* skenario Kecepatan teks pada tabel Berikut;

Tabel 3. use case skenario Kecepatan teks

Scenario	
Nama Use case	Kecepatan Teks
Nama Aktor	<i>User</i>
Deskripsi	Sistem akan Mempercepat Teks Sesuai dengan Input
Kondisi awal	Teks Berjalan Normal
Aksi Aktor	Respon Sistem
1. M emilih Kecepatan Teks	
	2. sistem Menampilkan Pesan sukses
Kondisi Akhir	Teks yang ditampilkan semakin cepat

Sumber : (Penulis)

3. Use case scenario Kecerahan Teks

Interaksi antara aktor pengguna dengan use case di jelaskan dalam use case skenario Kecerahan teks pada tabel 4:

Tabel 4. dalam use case skenario Kecerahan teks

Scenario	
Nama Use case	Kecerahan Teks
Nama Aktor	<i>User</i>

Deskripsi	Sistem akan Mencerahkan Teks Sesuai dengan Input
Kondisi awal	Teks Berjalan Normal
Aksi Aktor	Respon Sistem
1. Memilih Kecerahan Teks Teks	
	2. sistem Menampilkan Pesan sukses
Kondisi Akhir	Teks yang ditampilkan semakin cerah

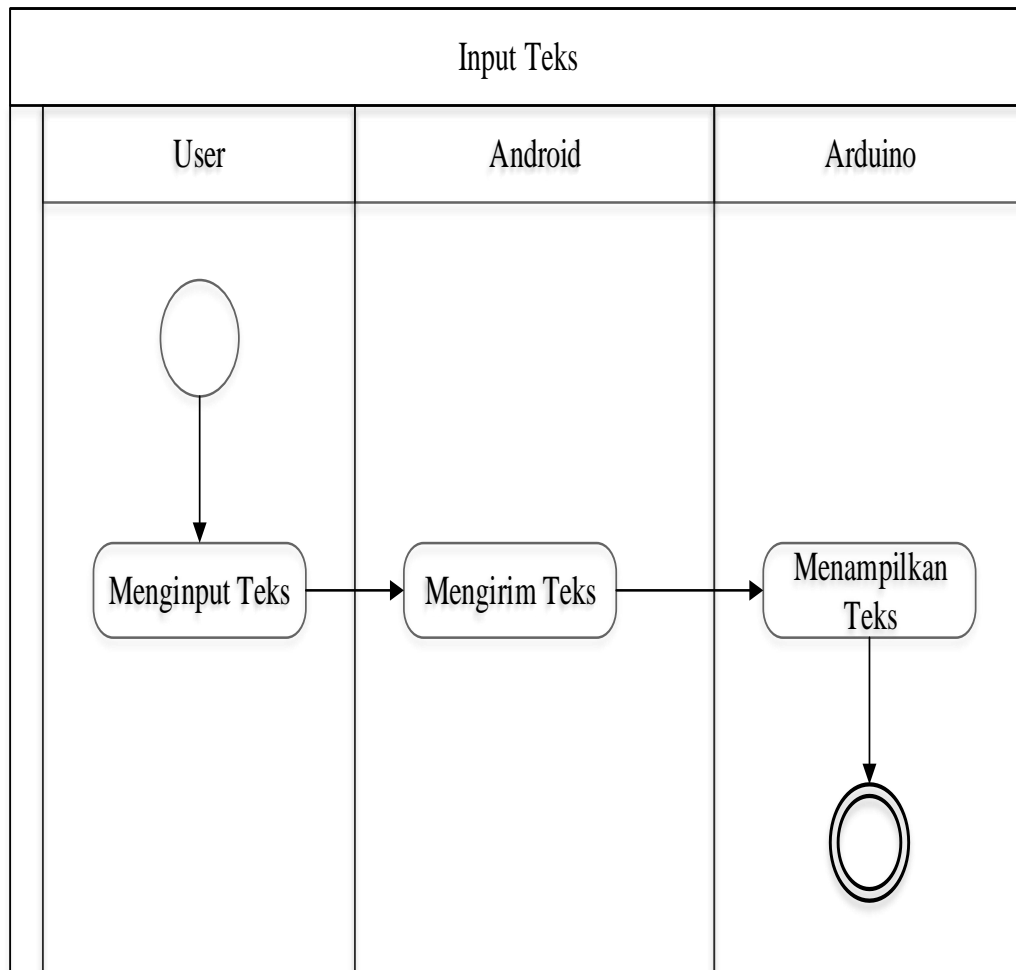
Sumber : (Penulis)

3.6.3. Activity Diagram

Activity diagram memodelkan aliran kerja atau *workflow* dari urutan aktifitas dalam suatu proses yang mengacu pada *Use case* diagram yang ada.

1. Activity diagram Input Teks

Activity diagram input teks dapat dilihat dari Gambar 19:

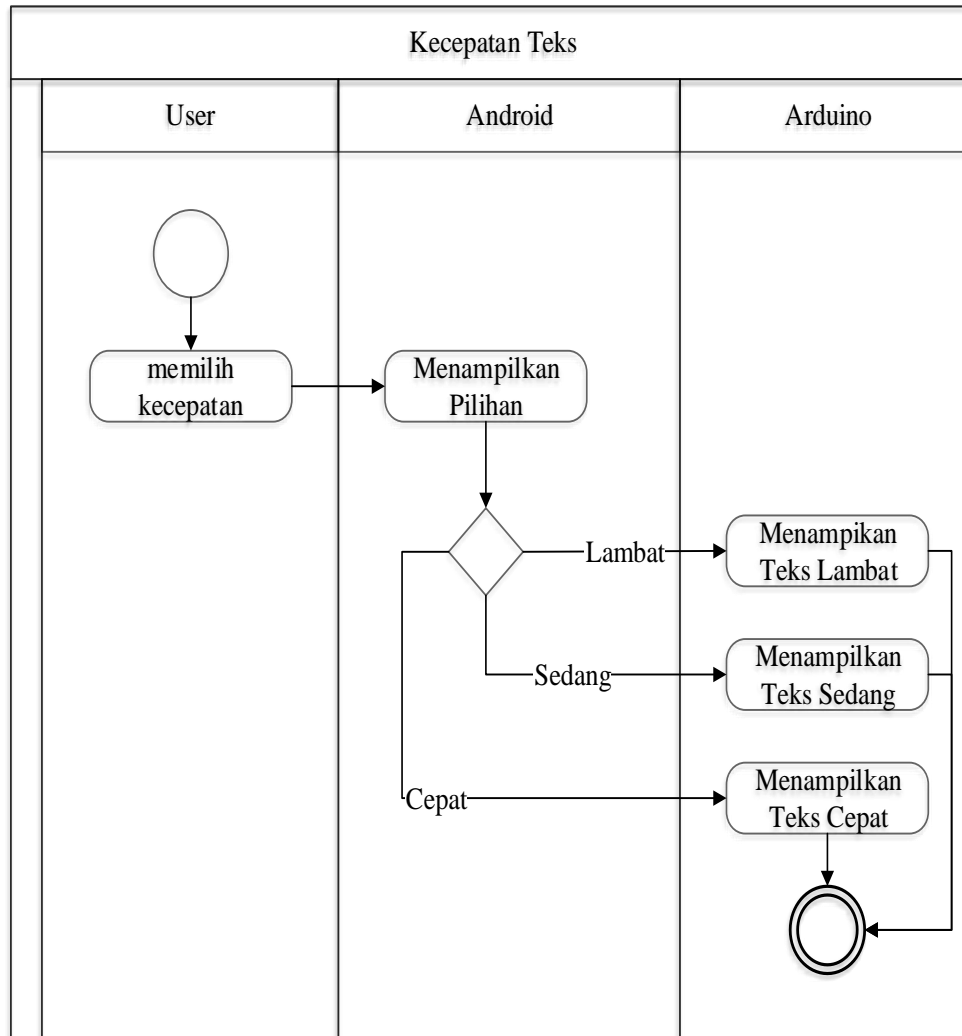


Gambar. 19 Activity diagram input Teks

Sumber : (Penulis)

2. Activity diagram Kecepatan Teks

Activity diagram Kecepatan teks dapat dilihat dari Gambar 21:

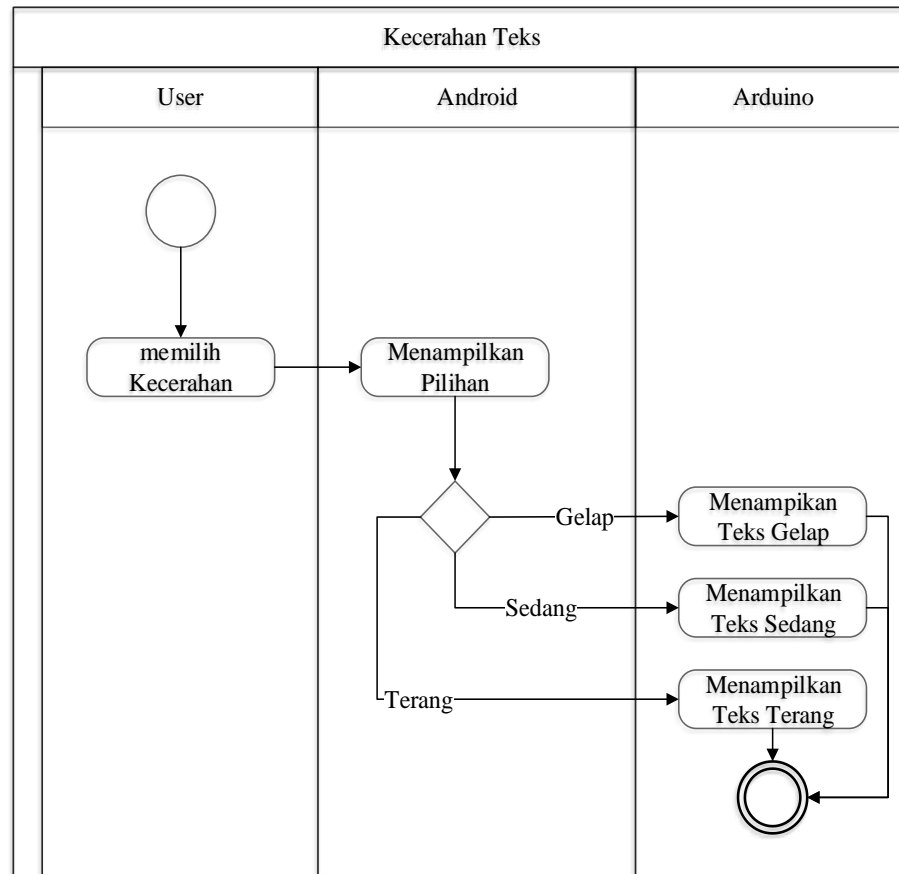


Gamabar. 20 Activity diagram Kecepatan Teks

Sumber : (Penulis)

3. Activity diagram Kecerahan Teks

Activity diagram Kecerahan teks dapat dilihat dari Gambar 20.



Gambar. 21 Activity diagram Kecerahan teks

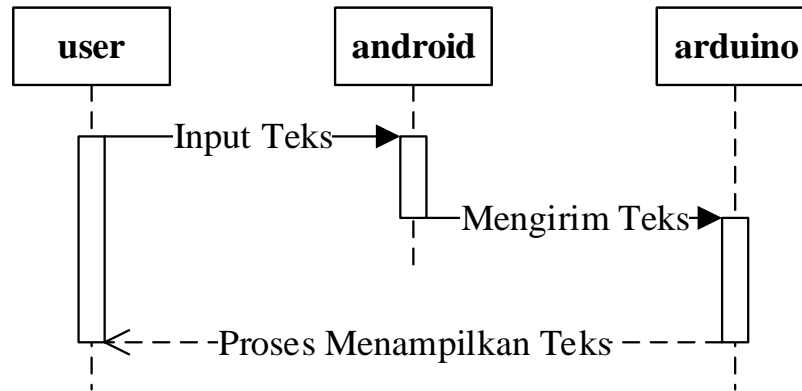
Sumber : (Penulis)

d. Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh objek dan pesan yang diletakkan diantara objek-objek ini didalam *Use case*. Oleh karena itu, untuk menggambarkan sequence diagram maka harus diketahui objek-objek yang terlibat dalam sebuah *Use case* beserta metode-metode yang dimiliki oleh kelas yang diinstansiasi menjadi objek itu.

1. Sequence Diagram Input Teks

Sequence diagram Input Teks dapat dilihat dari Gambar 22.

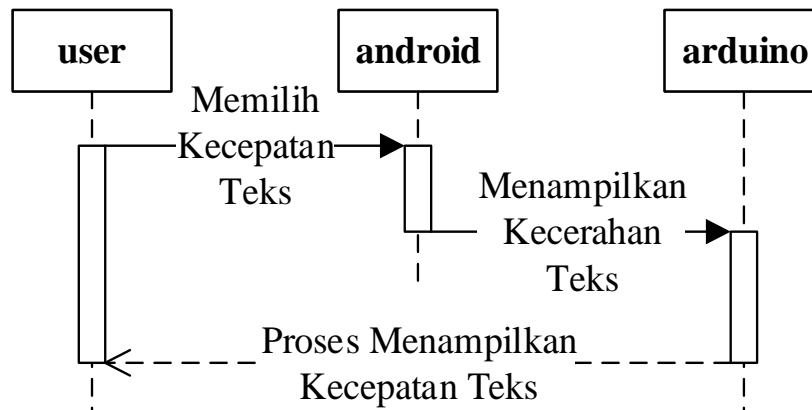


Gambar. 22 Diagram Input Teks

Sumber : (Penulis)

2. Sequence Diagram Kecepatan Teks

Sequence diagram Kecepatan Teks dapat dilihat dari gambar 23.

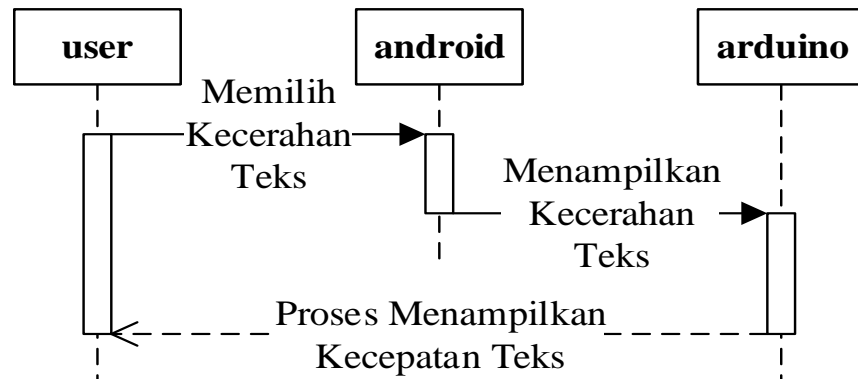


Gambar. 23 Diagram Kecepatan Teks

Sumber : (Penulis)

3. Sequence Diagram Kecerahan Teks

Sequence diagram Kecepatan Teks dapat dilihat dari gambar 24.

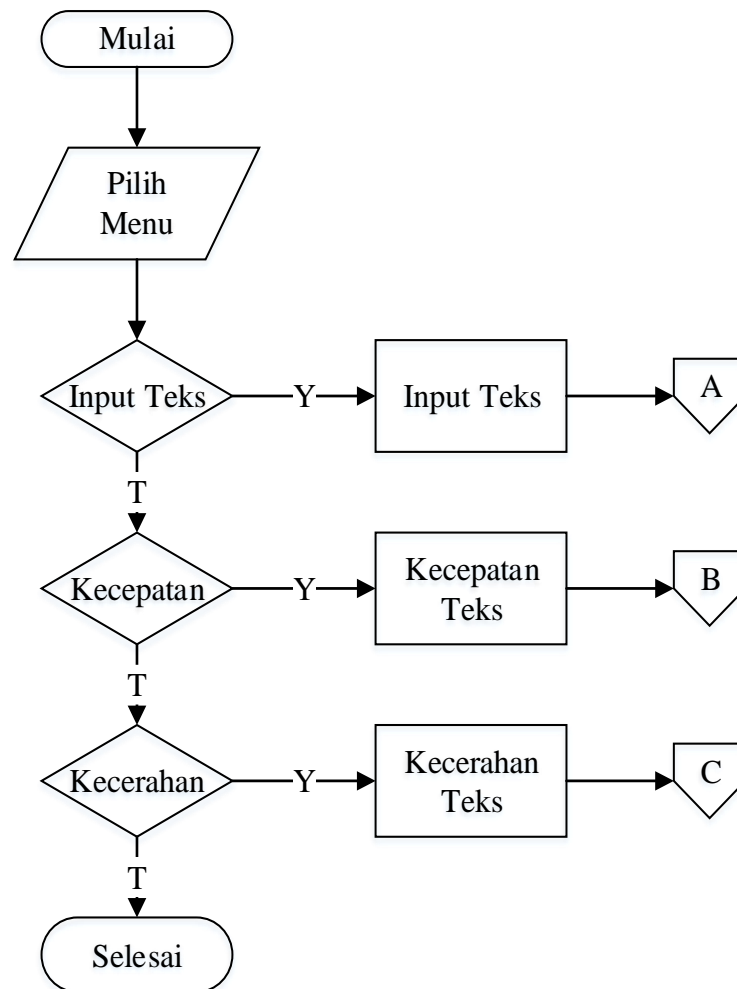


Gambar. 24 Diagram Kecerahan Teks

Sumber : (Penulis)

3.7. Perancangan Sistem

Perancangan merupakan penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh. Berikut Gambar 25 .*Fowchart*.



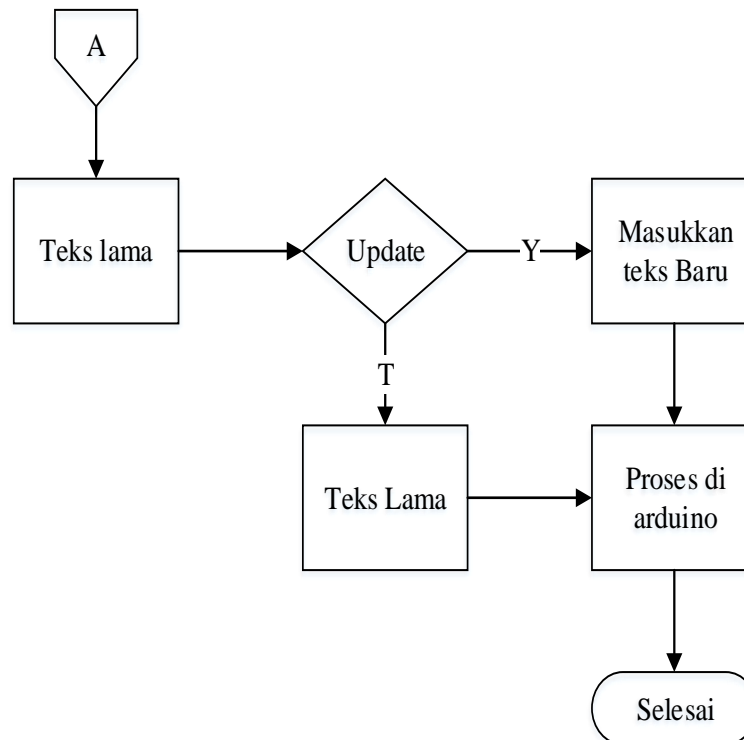
Gambar. 25 Flowchart

Sumber : (Penulis)

Pada gambar 25 menjelaskan alur diagram program sistem kendali papan pengumuman dimana pada halaman awal terdapat tiga pilihan menu yaitu menu *input text*, *kecerahan text* dan *kecepatan text*. Pada menu *input text* user dapat memasukkan *text* yang akan ditampilkan pada papan pengumuman berupa *running text*. Pada menu *kecerahan text* user dapat memilih tingkatan *kecerahan running text* yang ditampilkan. Pada menu *kecepatan text* user dapat mengatur kecepatan tampilan pada *running text*.

1. Flowchart Input Text

Pada menu *input text* terdapat form isian yang akan ditampilkan pada *running text*. Berikut ini adalah alur proses pada menu *input text* pada Gambar 26:



Gambar. 26 Flowchart Input Text

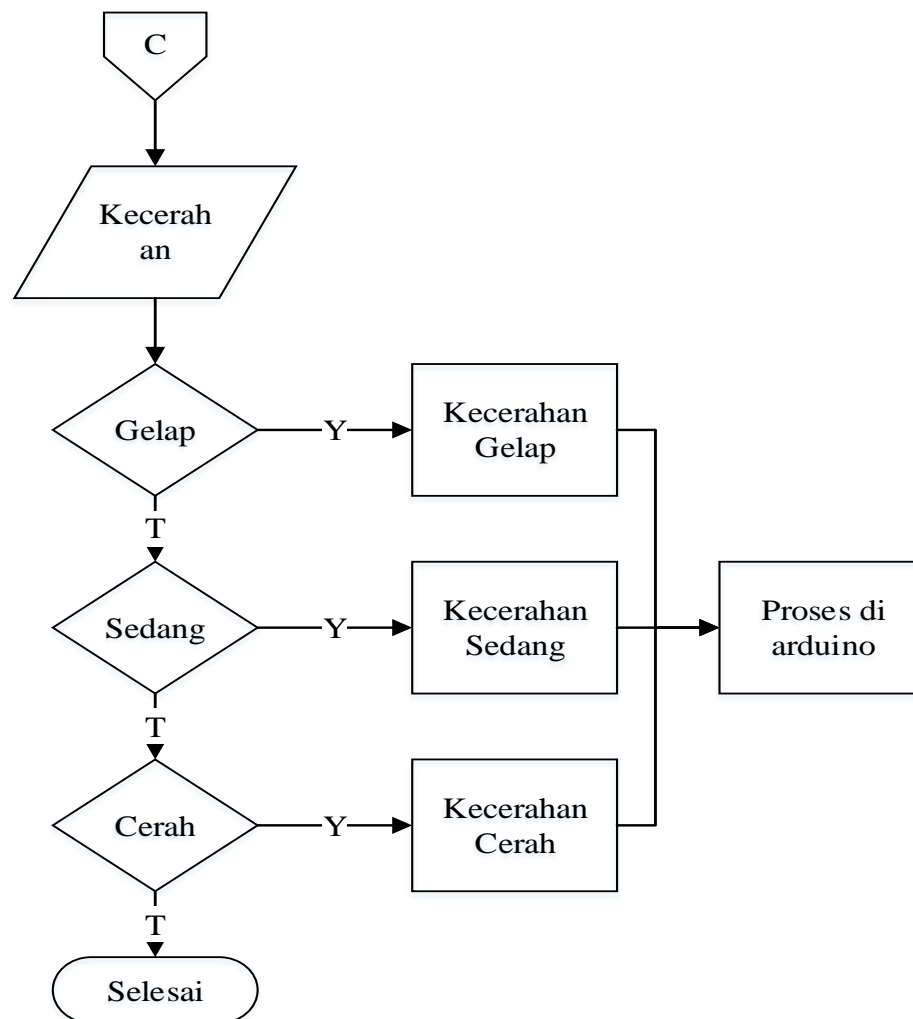
Sumber : (Penulis)

Gambar 25 diatas menjelaskan alur dari proses yang ada di dalam menu *input text*. Pertama sistem akan menampilkan teks lama kemudian akan memeriksa ada atau tidaknya *update* yang dilakukan *user*. Jika *user* melakukan update maka teks baru akan dimasukkan ke arduino dan ditampilkan pada *running text*, namun apabila tidak ada

update teks maka teks lama akan diproses langsung ke arduino dan ditampilkan pada *running text*.

2. Flowchart Kecerahan

Pada menu kecerahan akan ditampilkan tiga pilihan tingkat kecerahan yang dapat dipilih *user* yaitu gelap, sedang, dan cerah Berikut gambaran alur proses pada menu kecerahan pada Gambar 27:



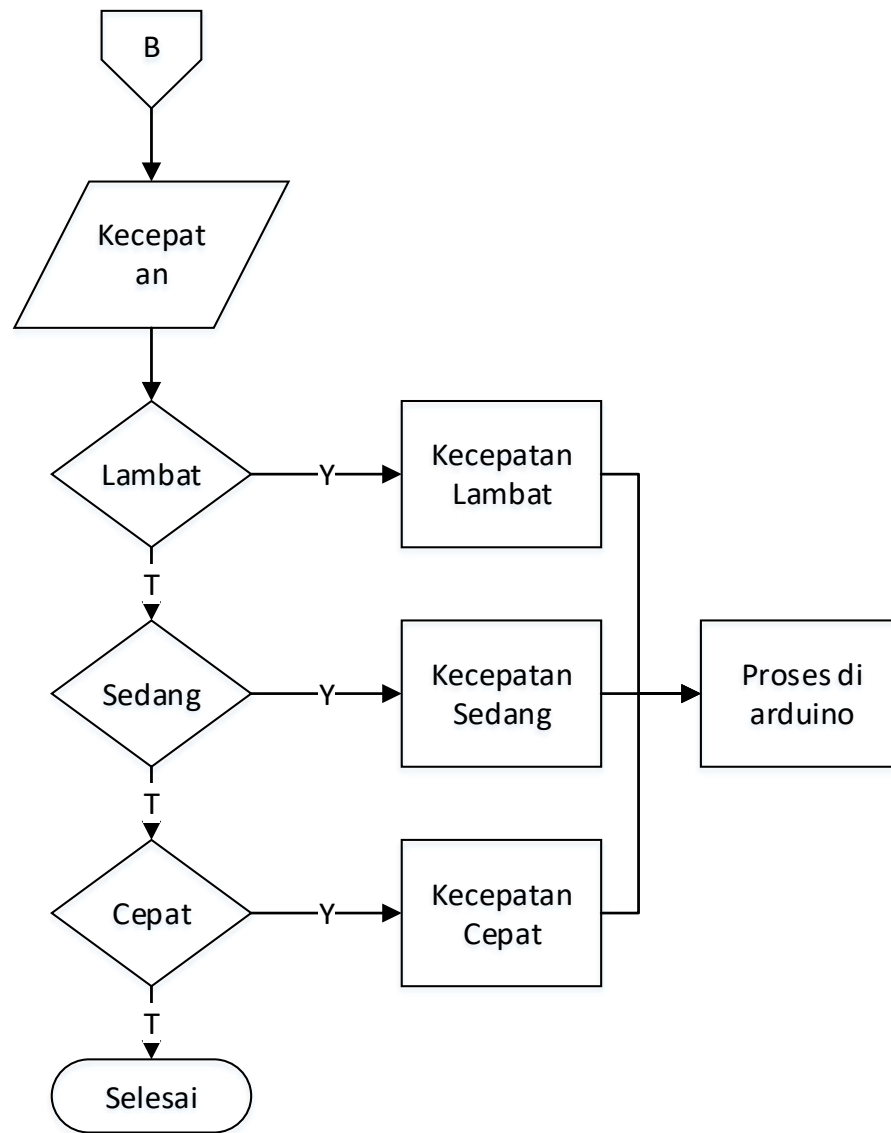
Gambar. 27 Flowchart Kecerahan

Sumber : (Penulis)

Gambar 27 ini menjelaskan proses yang terjadi pada menu kecerahan. Jika *user* memilih kecerahan gelap maka sistem akan mengirimkan data dengan tingkat kecerahan lemah pada proses yang terjadi didalam *arduino* dan ditampilkan pada *running text*, apabila *user* memilih kecerahan sedang maka sistem akan mengirimkan data dengan tingkat kecerahan sedang pada proses yang terjadi di *arduino* dan ditampilkan pada *running text*, apabila *user* memilih kecerahan cerah maka sistem akan mengirimkan data dengan tingkat kecerahan tinggi pada proses yang terjadi di *arduino* dan ditampilkan pada *running text*.

3. Flowchart Kecepatan

Pada menu kecepatan akan ditampilkan tiga pilihan tingkat kecepatan yang dapat dipilih *user* yaitu lambat, sedang, dan cepat Berikut gambaran alur proses pada menu kecepatan pada Gambar 28 :



Gambar. 28 Flowchart Kecepatan

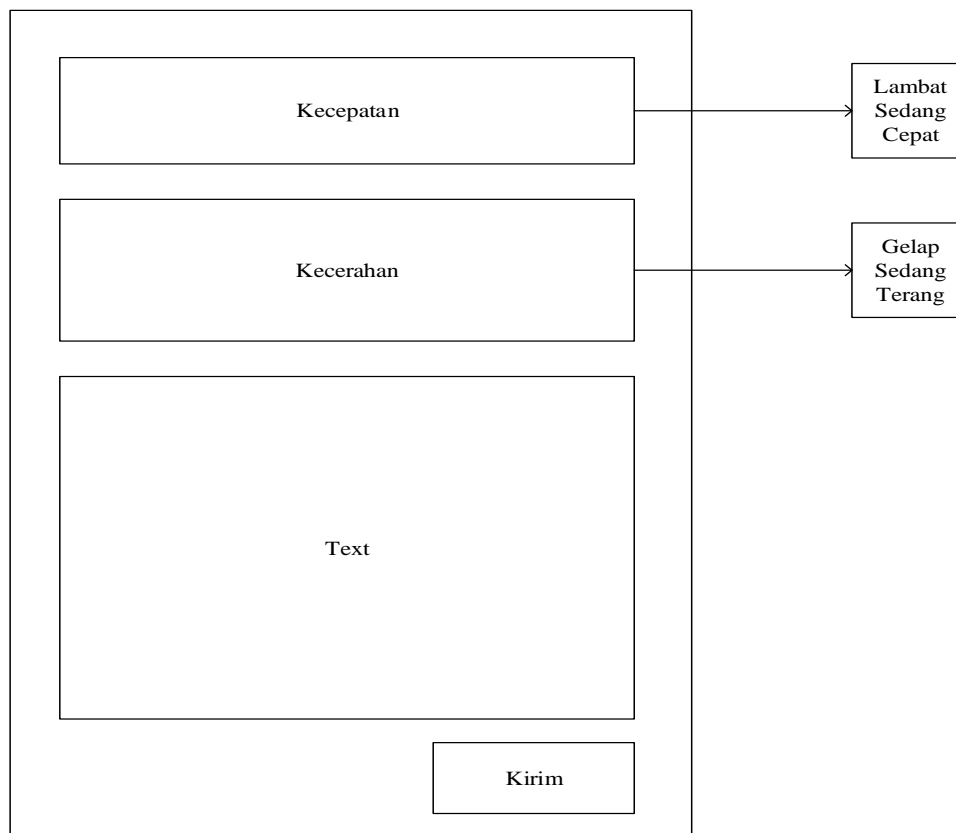
Sumber : (Penulis)

Gambar 28 ini menampilkan alur diagram proses pada menu pilihan kecepatan. Apabila *user* memilih kecepatan lambat maka data yang dikirimkan ke proses yang terjadi di arduino dan ditampilkan pada *running text* adalah tingkat kecepatan lambat. Apabila user memilih kecepatan sedang maka data yang dikirimkan ke proses yang

terjadi di arduino dan ditampilkan pada *running text* adalah tingkat kecepatan sedang. Apabila *user* memilih kecepatan cepat maka data yang dikirimkan ke proses yang terjadi di *arduino* dan ditampilkan pada *running text* adalah tingkat kecepatan cepat.

3.8. Perancangan Antarmuka

Antarmuka (*interface*) merupakan bentuk tampilan dari program yang tampil pada perangkat android yang bertujuan memberikan gambaran tentang aplikasi yang akan dibangun, sehingga akan memudahkan mengimplementasikan aplikasi sesuai ukuran layar dan memudahkan dalam pembuatan aplikasi pada Gambar 29.



Gambar 29. Perancangan Antarmuka

Sumber : (Penulis)

Komponen yang dipakai untuk membangun antar muka *layout lock* pada Gambar 29 adalah sebagai berikut :

1. *Spinner* "Kecepatan" : *Spinner* yang terdiri dari 3 pilihan yaitu Lambat, Sedang dan Cepat berfungsi untuk mengatur Kecepatan teks.
2. *Spinner* "Kecerahan" : *Spinner* yang terdiri dari 3 pilihan yaitu Gelap, Sedang dan Terang berfungsi untuk mengatur kecerahan teks.
3. *Edit Text* "teks" : *Edit Text* Yang berfungsi untuk memasukkan teks yang akan di kirim ke *arduino*.
4. *Button* "Kirim" : Tombol yang Berfungsi untuk mengirim teks ke *arduino*.

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Implementasi Sistem

Pada bab ini akan dilakukan tahapan implementasi dan pengujian terhadap perangkat lunak sistem kendali papan pengumuman digital yang berbasis android. Tahapan ini dilakukan setelah analisis perancangan selesai dilakukan dan selanjutnya akan diimplementasikan kedalam bahasa pemrograman, setelah tahapan implementasi dilakukan maka selanjutnya akan dilakukan tahapan pengujian dan akan dilihat kekurangan-kekurangan untuk pengembangan sistem selanjutnya.

Setelah pembangunan aplikasi dianalisis dan dirancang secara rinci, tahap selanjutnya adalah implementasi. tujuan dari tahapan implementasi ini adalah agar dapat memberikan masukan terhadap para pengembang sistem untuk pengembangan lebih baik lagi.

4.2. Implementasi Perangkat Keras

Perangkat keras yang digunakan untuk mengimplentasikan sistem dapat dilihat pada tabel 5 di bawah ini.

Tabel 5. Implementasi Perangkat Keras

no	Nama	Fungsi
1	Arduino uno R3	Mikrokontroler
2	Modul Bluetooth Hc-05	Komunikasi serial
3	Modul Dot Matrix 8x32	Menampilkan Teks

Sumber : (Penulis)

4	SmartPhone Android	Kontrol papan pengumuman
---	--------------------	--------------------------

4.3. Impelementasi Perangkat Lunak

Impelementasi Perangkat lunak yang ingin digunakan untuk membangun dan mengimplemntasikan sistem dapat dilihat pada tabel 6 berikut:

Tabel 6. Perangkat Lunak

no	Nama	Fungsi
1	Basic 4 Android	Merancang Aplikasi Android
2	Arduino Ide	Memprogram Arduino Uno
3	Fritzing	Menmbuat Rancangan Sketsa Hardware
4	SmartPhone Android	Kontrol papan pengumuman

Sumber : (Penulis)

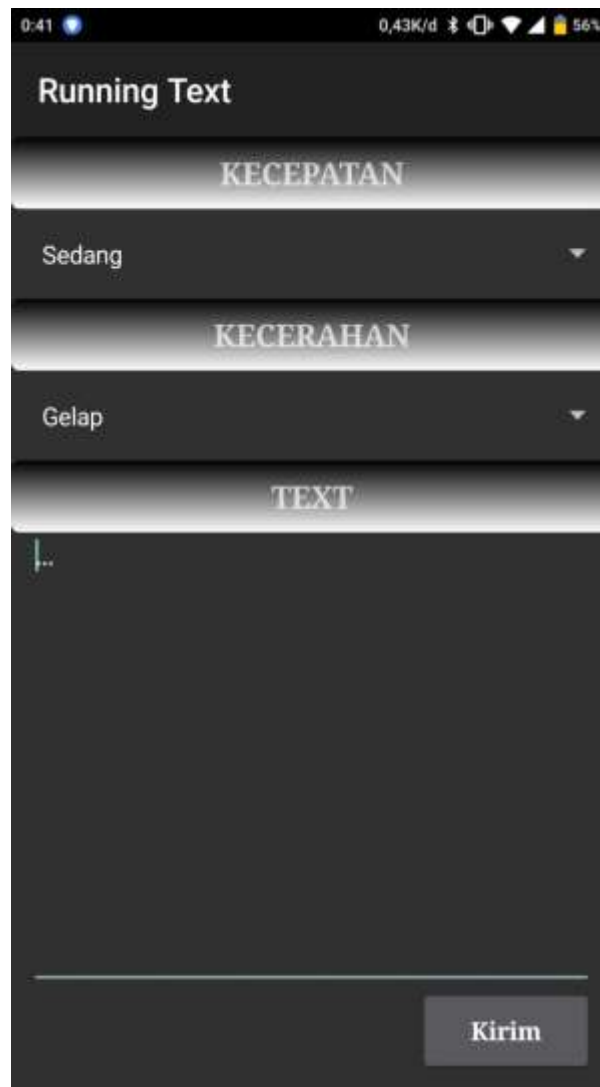
4.4. Petunjuk pengoperisan aplikasi

Pada saat akan memasang aplikasi di perangkat android terlebih dahulu melakukan penginstalasian master aplikasi Papan pengumuman. Setelah terinstal diperangkat android maka perangkat harus terkoneksi dengan Bluethoot untuk menjalankan aplikasi *papan pengumuman* ini, kemudian setelah terkoneksi maka aplikasi siap untuk digunakan.

4.5. Tampilan aplikasi

Pada halaman aplikasi kita dapat menemukan 3 menu yaitu menu Kecepatan, menu Kecerahan dan menu teks. Didalam menu kecepatan terdapat 3 pilihan yaitu lambat, sedan dan cepat yang berfungsi untuk mengatur kecepatan teks pada papan : pengumuman. Pada menu kecerahan terdapat juga 3 pilihan yaitu gelap sedan dan terang yang berfungsi untuk mengatur kecerahan teks. Pada menu teks

terdapat satu teksbox yang berfungsi untuk menginput teks yang akan dikirimkan ke papan pengumuman dan terdapat juga satu tombol kirim yang berfungsi untuk mengirim teks yang telah diketik di teksbox ke papan pengumuman berikut tampilan aplikasi papan pengumuman.



Gambar 30. Tampilan Aplikasi Running Text

Sumber : (Penulis)

4.6. Tampilan Hardware

Sistem kendali papan pengumuman digital dapat memberikan informasi dengan cara mengirimkan pesan dari smartphone android. Berikut gambar 31 dari perangkat keras sistem kendali papan pengumuman digital



Gambar 31. Tampilan Hardware

Sumber : (Penulis)

Pada gambar 28 menunjukkan kontruksi utama sistem kendali papan pengumuman digital. Perangkat ini terdiri dari 3 komponen yaitu *arduino uno*, *modul bluetooth hc-05* dan modul *dotmatrix 8x32*. *Arduino uno* berfungsi sebagai *mikrokontroler* yang akan memproses data yang dikirimkan dari smartphone android. Modul *Bluetooth HC-05* Berfungsi sebagai media transfer data antara mikrokontroler arduino uno dengan *smartphone android*, sedangkan modul *dot matrix 8x32* berfungsi sebagai penampil data yang dikirim dari *smartphone android*.

4.7. Pengujian Sistem

Pengujian sistem dilakukan untuk mengetahui jarak maksimal antara *Smartphone android* dengan papan pengumuman digital,

4.8. Pengujian dan Analisis Sistem

Pada BAB ini penulis melakukan beberapa pengujian baik untuk *hardware* dan *software* yang saling terintegrasi dalam sistem kendali papan pengumuman digital menggunakan *mikrokontroler arduino* dan *android*. Pengujian ini penulis lakukan dapat mengetahui perbaikan dan peningkatan terhadap sistem.

Berikut beberapa pengujian yang penulis lakukan :

4.1.1. Pengujian Koneksi Awal

Pada saat pertama kali membuka aplikasi *running text* kita diharuskan melakukan koneksi melalui *smartphone* ke *mikrokontroler arduino* melalui media transfer *bluetooth*. Berikut tabel hasil pengujian koneksi :

Tabel 7. Pengujian Koneksi

Pengujian Koneksi		
NO	Waktu (/detik)	Status
1	1,55	Berhasil
2	1,58	Berhasil
3	2,66	Berhasil
4	1,83	Berhasil
5	2,45	Berhasil
Rata-rata	2,04	

Sumber : (Penulis)

Dari hasil pengujian didapatkan rata – rata waktu koneksi sebesar 2,04 detik. Maka waktu yang diperlukan untuk menyambungkan pada awal memulai sistem sekitaran 2,04 detik agar terkoneksi dengan sempurna.

4.1.2. Pengujian Karakter Masukan

Pengujian ini dilakukan untuk melihat perbandingan antara karakter masukan dengan karakter yang tampil di *running text*, agar dapat mengetahui karakter minimum dan karakter maksimum yang dapat ditampilkan dengan sempurna di papan *running text*. Berikut tabel 8 hasil pengujian :

Tabel 8. Pengujian Karakter Masukan

Percoobaan Pengujian Karakter Masukan			
No	Jumlah karakter Masukan	Jumlah Karakter yang Tampil	Keterangan
1	10	10	Berhasil
2	20	20	Berhasil
3	30	30	Berhasil
4	40	40	Berhasil
5	50	50	Berhasil
6	60	60	Berhasil
7	70	70	Berhasil

8	80	80	Berhasil
9	90	90	Berhasil
10	100	100	Berhasil
11	110	100	Gagal
12	120	1 (-> / tanda panah kanan)	Gagal
13	130	130	Berhasil
14	140	0	Gagal
15	150	0	Gagal
16	160	0	Gagal

Sumber : (Penulis)

Dari hasil pengujian yang didapatkan karakter yang tampil dengan sempurna hanya sampai 100 karakter, seterusnya hanya menampilkan beberapa karakter dan ketika masukan ditambah menjadi 140 tidak ada karakter yang tampil.

4.1.3. Pengujian Kecepatan Tampilan Karakter

Pengujian ini dimaksudkan untuk mengetahui kecepatan karakter yang ditampilkan pada papan *running text* yang dihitung sesuai kecepatan yang diatur dari aplikasi *smarthome* yaitu dengan tingkat kecepatan lambat, sedang dan cepat. Berikut tabel hasil pengujian terhadap kecepatan tampilan karakter :

Tabel 9. Pengujian Kecepatan Tampilan Karakter input

Kecepatan Tampilan Karakter input				
No	Jumlah Karakter Inputan	Tingkat Kecepatan		
		Lambat (/detik)	Sedang (/detik)	Cepat (/detik)
1	5	18,63	6,31	3,50
2	10	27,52	9,67	8,13
3	20	46,30	16,14	8,56
4	30	64,28	22,57	12,14
Rata – rata		39,18	13,67	8,08

Sumber : (Penulis)

Hasil dari pengujian yang dilakukan didapatkan data yang menampilkan rata – rata kecepatan karakter yang tampil pada tingkat kecepatan sesuai dengan yang diatur pada aplikasi smarthome. Rata-rata tersebut menjadi pedoman kecepatan tampilan karakter yang diinputkan sesuai dengan tingkat kecepatan yang diatur.

4.1.4. Pengujian Koneksi Ulang

Pengujian ini dilakukan untuk mengetahui waktu yang diperlukan apabila pada saat tertentu koneksi terputus, maka pada pengujian ini penulis mengamati waktu yang diperlukan sistem untuk mencoba koneksi ulang ke *mikrokontroler*. Berikut tabel hasil pengujian :

Tabel 10. Pengujian Koneksi Ulang Setelah Terputus

Koneksi Terputus-putus		
Pengujian	Waktu	Status
1	10,11	Mencoba Ulang
2	17,92	Mencoba Ulang
3	18,11	Mencoba Ulang
4	17,98	Mencoba Ulang
5	17,92	Mencoba Ulang

Sumber : (Penulis)

Berdasarkan hasil pengujian yang dilakukan kita mendapatkan waktu rata – rata yang digunakan oleh sistem untuk menyambungkan kembali ke *mikrokontroler* setelah koneksi terputus. Waktu rata – rata yang diperlukan sistem adalah ± 17 detik sampai sistem mencoba menyambungkan kembali ke *mikrokontoler*.

4.1.5. Pengujian karakter berupa simbol

Pengujian ini dilakukan karena sistem diatur melalui aplikasi yang ada di smartphone maka keyboard default yang digunakan adalah *Google Keyboard*. Untuk karakter biasa *running text* dapat menampilkannya dengan sempurna yaitu huruf dan angka, pada pengujian ini penulis menggunakan simbol – simbol tambahan yang ada di *Google Keyboard* untuk mengetahui kondisi simbol - simbol tersebut apabila digunakan sebagai karakter masukan pada aplikasi dan bagaimana tampilan simbol – simbol tersebut pada papan *running text*. Berikut tabel hasil pengujian simbol – simbol pada *Google Keyboard* :

Tabel 11. Pengujian Karakter Simbol - Simbol *Google Keyboard*

NO	Karakter Simbol Keyboard	Keterangan
1	#	Sesuai
2	@	Sesuai
3	\$	Sesuai
4	-	Sesuai
5	&	Sesuai
6	–	Sesuai
7	+	Sesuai
8	(Sesuai
9)	Sesuai
10	/	Sesuai
11	*	Sesuai
12	“	Sesuai
13	‘	Sesuai
14	:	Sesuai
15	;	Sesuai
16		Tidak Sesuai
17	√	Tidak Sesuai
18	?	Sesuai
19	●	Tidak Sesuai

20	¶	Tidak Sesuai
21	%	Sesuai
22	÷	Tidak Sesuai
23	[Sesuai
24]	Sesuai
25	{	Sesuai
26	}	Sesuai
27	>	Sesuai
28	<	Sesuai
29	=	Sesuai
30	~	Sesuai

Sumber : (Penulis)

Dari tabel 11 hasil pengujian diatas, penulis menemukan beberapa simbol yang sesuai dengan masukan dan ditampilkan serta ada juga yang tidak dapat ditampilkan sesuai dengan yang dimasukkan dan tampilannya pada papan *running text*. Hal ini disebabkan modul *MAX7219* tidak menyediakan karakter yang sesuai dengan beberapa simbol – simbol yang ada di *Google Keyboard*.

4.1.6. Pengujian *Bluetooth*

Pengujian ini dilakukan untuk mengetahui jarak yang dapat dijangkau oleh koneksi *bluetooth*. Pengujian ini dilakukan melalui koneksi *bluetooth*, pengujian didasarkan dengan pengukuran jarak antara *Android* dengan

sepeda motor tanpa penghalang garis lurus dengan papan *running text*.

Dari pengujian ini didapatkan hasil sebagai berikut :

Tabel 12. Pengujian Jarak Koneksi *Bluetooth*

No	Jarak	Hasil	Catatan
1	±5	Terhubung	Respon cepat
2	±10	Terhubung	Respon cepat
3	±15	Terhubung	Respon cepat
4	±20	Terhubung	Respon cepat
5	±25	Terhubung	Respon lambat
6	±30	Terhubung	Respon lambat
7	±35	Terputus	Tidak ada respon

Sumber : (Penulis)

Menurut data dari hasil pengujian yang di tampilkan pada tabel 12 dapat diketahui bahwa :

1. Pada jarak ±5 sampai dengan ±20 meter koneksi masih tersambung dengan baik dan dapat merespon dengan cepat ke papan *running text*. Koneksi pada jarak ini masih stabil.
2. Pada jarak ±25 sampai ±30 meter koneksi mulai terganggu dan tidak tersambung dengan baik, respon yang diberikan sebagai *feedback* dari perintah dari aplikasi *Android* juga lambat agar ditampilkan pada papan *running text*. Koneksi pada jarak sejauh ini sudah mulai terganggu dan tidak stabil.
3. Pada jarak ±35 meter koneksi terputus dan tidak tersambung ke *bluetooth*, dan tidak ada respon dari sistem. Koneksi pada jarak ini sudah terputus.

BAB V

PENUTUP

5.1. Kesimpulan

Setelah penulis mengadakan penelitian dan menganalisa data yang diperoleh baik data yang diperoleh dari literatur maupun data dari lapangan, dengan membahas skripsi yang berjudul “Sistem Kendali Papan Pengumuman Digital Menggunakan *Mikrokontroler Arduino dan Android*”, maka dapat ditarik suatu kesimpulan sebagai berikut :

- a. Dengan dibuatnya,”Sistem Kendali Papan Pengumuman Digital Menggunakan *Mikrokontroler Arduino dan Android*”, kita dapat menjalankan *running text* untuk papan pengumuman digital.
- b. Aplikasi yang dibuat menggunakan bahasa pemrograman *android*.
- c. Mikrokontroler yang digunakan adalah *arduino uno R3*.
- d. Media yang digunakan untuk mentransfer perintah dari aplikasi ke *prototype Running Text* adalah *Bloetooth*.
- e. Media Transfer data yang di buat menggunakan *Bluetooth HC 05*
- f. Media Tampilan berjalanya *running text* menggunakan modul *display dotmatrix LED 8x32*.
- g. *development tool* sederhana yang *powerfull* untuk membangun aplikasi *Android* menggunakan *Basic4Android*.

5.2. Saran

Dari kesimpulan diatas, penulis mencoba memberikan saran-saran yang diharapkan dapat berguna untuk perkembangan sistem selanjutnya, anantara lain sebagai berikut :

- a. Perancangan aplikasi”,Sistem Kendali Papan Pengumuman Digital Menggunakan *Mikrokontroler Arduino* dan *Android* pada smartphome android, untuk penelitian selanjutnya diharapkan dapat memenuhi kekurangan dalam sistem yang dibuat penulis yaitu”, Sistem Kendali Papan Pengumuman Digital Menggunakan *Mikrokontroler Arduino* dan *Android*”,.
- b. Untuk meningkatkan fasilitas-fasilitas yang ada pada aplikasi tersebut untuk menjadi lebih baik maka perlu adanya pengembangan media transfer data tidak hanya dengan media *bloetooth*.

DAFTAR PUSTAKA

- Agung, M Bangun, 2014. *Arduino For Beginners*. Surya University. Tangerang.
- Febriani, Wike. 2017. *Alat Pendeteksi Maling Pada Rumah Menggunakan Sensor Passive Infrared Berbasis Arduino*. Skripsi FMIPA Universitas Sumatera Utara : Medan.
- Lilik , dan Rita. 2017. *Rancang Bangun Papan Informasi Running Text berbasis Arduino Uno*. Jurnal TrendTech.
- Linarti. 2014. *Module Bluetooth HC-05*. Skripsi Fakultas Komputer Politeknik Negri Sriwijaya. Sumatera Selatan.
- Maimunah. 2017. *Trainer Periferal Antarmuka Berbasis Mikrokontroler Arduino Uno*. Jurnal Teknik Elektro dan Komputer Universitas Sam Ratulangi : Manado.
- Nugroho, Sigit. 2017. *Aplikasi Kran Otomatis Berbasis Arduino*. Proyek Akhir Sekolah Tinggi Manajemen Informatika Dan Komputer Akakom : Yogyakarta.
- Prima Lecture. 2016. *Modul Praktikum Elektronika Digital*. Laboratorium Politeknik Elektronika Negeri Surabaya.
- Rusli, Sabri. 2011. *Running Text Menggunakan Keyboard AT PS/2 Berbasis Mikrokontroler Atmega 8535*. Skripsi Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim : Riau.
- Septiana, Mochamad Irvan. 2013. *Teks Berjalan Dotmatrix Dengan Keyboard Berbasis Atmega32*. Yogyakarta

- Santoso, H. 2015. *Panduan Praktis Arduino Untuk Pemula*. Elang Sakti Tenggerek.
- Santoso, H. 2017. *Panduan Praktis Belajar Arduino Untuk Pemula Serial Mosnter Arduino*. Elang Sakti. Tenggerek.
- Badawi, A. (2018). Evaluasi Pengaruh Modifikasi Three Pass Protocol Terhadap Transmisi Kunci Enkripsi.
- Batubara, Supina. "Analisis perbandingan metode fuzzy mamdani dan fuzzy sugeno untuk penentuan kualitas cor beton instan." *IT Journal Research and Development* 2.1 (2017): 1-11.
- Bahri, S. (2018). *Metodologi Penelitian Bisnis Lengkap Dengan Teknik Pengolahan Data SPSS*. Penerbit Andi (Anggota Ikapi). Percetakan Andi Ofsset. Yogyakarta.
- Erika, Winda, Heni Rachmawati, and Ibnu Surya. "Enkripsi Teks Surat Elektronik (E-Mail) Berbasis Algoritma Rivest Shamir Adleman (RSA)." *Jurnal Aksara Komputer Terapan* 1.2 (2012).
- Fitriani, W., Rahim, R., Oktaviana, B., & Siahaan, A. P. U. (2017). Vernam Encrypted Text in End of File Hiding Steganography Technique. *Int. J. Recent Trends Eng. Res*, 3(7), 214-219.
- Hardinata, R. S. (2019). Audit Tata Kelola Teknologi Informasi menggunakan Cobit 5 (Studi Kasus: Universitas Pembangunan Panca Budi Medan). *Jurnal Teknik dan Informatika*, 6(1), 42-45.
- Hariyanto, E., Lubis, S. A., & Sitorus, Z. (2017). Perancangan prototipe helm pengukur kualitas udara. *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, 1(1).
- Hariyanto, E., & Rahim, R. (2016). Arnold's cat map algorithm in digital image encryption. *International Journal of Science and Research (IJSR)*, 5(10), 1363-1365.
- Harumy, T. H. F., & Sulistianingsih, I. (2016). Sistem penunjang keputusan penentuan jabatan manager menggunakan metode mfep pada cv. Sapo durin. In *Seminar Nasional Teknologi Informasi dan Multimedia* (pp. 6-7).
- Iqbal, M., Siahaan, A. P. U., Purba, N. E., & Purwanto, D. (2017). Prim's Algorithm for Optimizing Fiber Optic Trajectory Planning. *Int. J. Sci. Res. Sci. Technol*, 3(6), 504-509.
- Marlina, L., Muslim, M., Siahaan, A. U., & Utama, P. (2016). Data Mining Classification Comparison (Naïve Bayes and C4. 5 Algorithms). *Int. J. Eng. Trends Technol*, 38(7), 380-383.

- Muttaqin, Muhammad. "ANALISA PEMANFAATAN SISTEM INFORMASI E-OFFICE PADA UNIVERSITAS PEMBANGUNAN PANCA BUDI MEDAN DENGAN MENGGUNAKAN METODE UTAUT." *Jurnal Teknik dan Informatika* 5.1 (2018): 40-43.
- Ramadhan, Z., Zarlis, M., Efendi, S., & Siahaan, A. P. U. (2018). Perbandingan Algoritma Prim dengan Algoritma Floyd-Warshall dalam Menentukan Rute Terpendek (Shortest Path Problem). *JURIKOM (Jurnal Riset Komputer)*, 5(2), 135-139.
- Rahim, R., Aryza, S., Wibowo, P., Harahap, A. K. Z., Suleman, A. R., Sihombing, E. E., ... & Agustina, I. (2018). Prototype file transfer protocol application for LAN and Wi-Fi communication. *Int. J. Eng. Technol.*, 7(2.13), 345-347.
- Wahyuni, Sri. "Implementasi Rapidminer Dalam Menganalisa Data Mahasiswa Drop Out." *Jurnal Abdi Ilmu* 10.2 (2018): 1899-1902.

LAMPIRAN SCRIPT PEMROGRAMAN

```
#include <SoftwareSerial.h>

#include <MD_MAX72xx.h>

MD_MAX72XX mx = MD_MAX72XX(4, 5, 6, 4); //(DATA_PIN, CLK_PIN, CS_PIN,
MAX_DEVICES)

SoftwareSerial HC05(2, 3);

static boolean Tampil = true;

long Waktu = 0;

char *Pesan = "...";

String TXT = "";

int KRN = 1, KPN = 100;

void setup(){

  HC05.begin(9600);

  mx.begin(); resetMatrix();

}

void loop() {

  while(HC05.available() > 0){

    char Data = HC05.read();

    if(Data == '\n'){

      //Set Text

      if(TXT.substring(0,4) == "TXT="){

        TXT.remove(0,4); TXT += '\0';

        Pesan = malloc(TXT.length());

        TXT.toCharArray(Pesan, TXT.length()+1);
```



```

    resetMatrix();
}
//Set Kecerahan
if(TXT.substring(0,4) == "KRN="){
    TXT.remove(0,4);
    KRN = TXT.toInt();
    mx.control(MD_MAX72XX::INTENSITY, KRN);
}
//Set Kecepatan
if(TXT.substring(0,4) == "KPN="){
    TXT.remove(0,4);
    KPN = TXT.toInt();
}
if(TXT == "STAT"){
    String STAT = String(KRN) + "#" + String(KPN) + "#" + String(Pesan) + "#";
    HC05.println(STAT);
}
TXT = "";
}else{
    if(Data != '\r'){
        TXT += Data;
    }
}
}

Tampil = RunningText(Tampil, Pesan);
}

void resetMatrix(){

```

```
mx.control(MD_MAX72XX::INTENSITY, KRN);  
mx.control(MD_MAX72XX::UPDATE, MD_MAX72XX::ON);  
mx.clear(); Waktu = 0; Tampil = true;  
}
```

```
bool RunningText(bool bInit, char *pmsg){  
    static char curMessage[100], *p = curMessage;  
    static uint8_t state = 0, curLen, showLen, cBuf[8];  
    uint8_t colData;  
  
    if (bInit){  
        resetMatrix();  
        strcpy(curMessage, pmsg);  
        state = 0; p = curMessage;  
        bInit = false;  
    }  
    if (millis() - Waktu < KPN) return(bInit);  
  
    mx.transform(MD_MAX72XX::TSL);  
    Waktu = millis();  
  
    switch (state)  
    {  
        case 0:  
            showLen = mx.getChar(*p++, sizeof(cBuf)/sizeof(cBuf[0]), cBuf);  
            curLen = 0; state = 1;  
        case 1:  
            colData = cBuf[curLen++];  
            mx.setColumn(0, colData);  
    }
```

```
if (curLen == showLen){
    showLen = ((*p != '\0') ? 1 : mx.getColumnCount()-1);
    curLen = 0; state = 2;
}
break;
case 2:
    mx.setColumn(0, 0);
    if (++curLen == showLen) {
        state = 0; blnit = (*p == '\0');
    }
    break;
default:
    state = 0;
}
return(blNit);
}
```