

PERANCANGAN SOFTWARE UNTUK MEMPERPANJANG TULISAN YANG DAPAT DITAMPILKAN RUNNING TEKS DISPLAY LED DOT MATRIKS BERBASIS ARDUINO UNO

Bagus Furwanda¹⁾, Marvin F.S Hutabarat²⁾, Jeremia Siregar³⁾

Fakultas Teknologi Mineral, Institut Sains Teknologi T.D Pardede,
Jl.DR.TD.Pardede No. 8, Medan 2015, Sumatera Utara, Indonesia

bagusfurwanda00@gmail.com¹⁾, marvinhutabarat@istp.ac.id, jeremiasiregar@istp.ac.id

ABSTRAK

Media penyampaian informasi *running text* merupakan suatu inovasi yang menarik dalam perkembangan teknologi informasi saat ini. Informasi yang disampaikan pada umumnya diterima melalui berita dengan tulisan yang ditulis pada media cetak, internet dan lain- lain yang terkadang ada kejenuhan dalam membaca. *Running text* yang beredar saat ini dengan kontrol komputer kadang terkendala pada ketersediaan PC saat mengontrol dan peralatan sinkronisasinya yang harus ada untuk bisa mengatur *running text* sesuai *input* yang dimasukkan oleh user. Kecepatan transfer informasi sangat berpengaruh terhadap *output*, melalui teknologi *android* saat ini yang sudah canggih dimanfaatkan untuk kontrol *running text* yang *up to date*, relatif lebih aman dan ramah lingkungan pada pemakaian dengan control LED dot matriks sebagai penampil informasi tombol keypad terprogram berbasis mikrokontroler Arduino uno yang memiliki kemudahan dalam akses data pada *keypad*. *Keypad* dot matriks yang digunakan terdiri dari 16 saklar (4 baris dan 4 kolom) kecepatan dan karakteristik *text*. Jumlah LED yang digunakan sebanyak 7 baris x 6 kolom LED atau 112 buah LED dan IC 4094. Tujuh buah LED dan satu IC 4094 untuk menghasilkan 1 kolom dan 7 baris display. Rangkaian minimum sistem dibangun dari C=10uF/ 25V sebagai waktu reset secara otomatis, Kristal 12.000 MHz ssebagai osilator, dan mikrokontroler AT89S51 sebagai CPU.

Kata Kunci : *Running text* , LED dot matriks, keypad terprogram , Mikrokontroler AT89S51, Arduino uno.

ABSTRACT

Information delivery media running text is an interesting innovation in the development of information technology today. The information conveyed is generally received through news written in print media, the internet and others where sometimes there is boredom in reading. The currently circulating running text with computer control is sometimes constrained by the availability of a PC when controlling and the synchronization equipment that must be available to be able to set the running text according to the input entered by the user. The speed of information transfer greatly affects the output, through today's sophisticated Android technology used to control running text which is up to date, relatively safer and environmentally friendly in use with the dot matrix LED control as a display of programmable keypad button information based on the Arduino Uno microcontroller which have easy access to data on the

keypad. The dot matrix keypad used consists of 16 switches (4 rows and 4 columns) for speed and text characteristics. The number of LEDs used is 7 rows x 6 column LEDs or 112 LEDs and 4094 IC. Seven LEDs and one 4094 IC to produce 1 column and 7 display rows. The minimum system circuit is built from $C = 10\mu F / 25V$ as an automatic reset time, a 12,000 MHz Crystal as an oscillator, and an AT89S51 microcontroller as a CPU.

Keywords: Running text, LED dot matrix, programmable keypad, AT89S51 microcontroller, Arduino uno.

1. PENDAHULUAN

Mendasarkan permasalahan running teks yang ada saat ini yaitu mengalami permasalahan didalam pengaturannya, maka dari itu kali ini saya menggunakan keypad sebagai media untuk memilih informasi yang ingin di tampilkan pada display LED.

Keypad dot matriks akan memudahkan kita untuk menampilkan running teks pada display LED. Keypad dot matriks yang digunakan terdiri dari 16 saklar (4 baris dan 4 kolom), sambungan baris dan kolom keypad dot matriks ke minimum sistem mikrokontroler Arduino uno.

Keypad berisi tombol memungkinkan pengguna untuk memasukkan huruf, angka atau symbol pada kontroler Keypad tidak berisi ASCII tombol keyboard dan sering ditemukan dalam peralatan khusus, yang sering digunakan sebagai suatu input pada beberapa peralatan yang berbasis mikrokontroler.

Tulisan yang ditampilkan running teks berbasis arduino uno dapat dibuat lebih panjang dengan cara menambah modul display LED Dot Matriks yang dipasang. Modul display LED Dot Matriks ditambah untuk memperpanjang tulisan yang ditampilkan namun programnya belum dirubah variabelnya maka tidak bisa terjadi. Supaya bisa terjadi maka perlu programnya dirubah juga dengan cara merubah variabel jumlah displaynya.

Tujuan dan Manfaat

1. Display LED yang dirangkai dalam bentuk dot matriks.
2. Mikrokontroler arduino uno menerima data 8 bit dari tombol keypad dot matriks dan menampilkan informasi pada display LED sesuai dengan tombol yang dipilih.
3. Kata informasi yang ditampilkan display dapat diganti melalui program yang diupload ke memori mikrokontroler arduino uno.
4. Arduino memiliki kelebihan tersendiri dibanding board mikrokontroler yang lain selain bersifat *open source*, arduino juga mempunyai bahasa pemrogramannya sendiri yang berupa bahasa C (cobol). Selain itu dalam

board arduino sendiri sudah terdapat loader yang berupa USB sehingga memudahkan dalam memprogram mikrokontroler yang ada di arduino.

5. kurva karakteristik komponen, konfigurasi pin-pin mikrokontroler Arduino uno dan pemrogramannya, konfigurasi pin-pin IC 4094, dan LED.
6. Rancang bangun hardware dan software yang dapat memproses data bit karakter yang diupload dan menerima sinyal bit data dari keypad dan menampilkan informasi dalam bentuk kata display LED dot matriks sesuai dengan tombol yang dipilih.

2. TINJAUAN PUSTAKA

Arduino

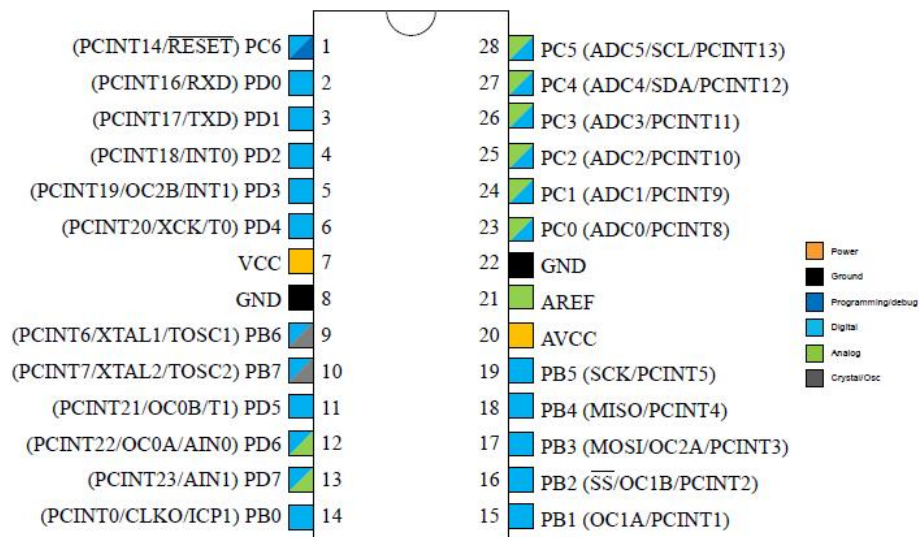
Arduino adalah sebuah board mikrokontroler yang berbasis ATmega328. Arduino memiliki 14 pin input/output yang mana 6 pin dapat digunakan sebagai *output* PWM, 6 analog input, *crystal* osilator 16 MHz, koneksi USB, *jack power*, kepala ICSP, dan tombol reset. Arduino mampu *men-support* mikrokontroler; dapat dikoneksikan dengan komputer menggunakan kabel USB.

Arduino menyediakan 20 pin I/O, yang terdiri dari 6 pin *input* analog dan 14 pin digital *input/output*. Untuk 6 pin analog sendiri bisa juga difungsikan sebagai *output* digital jika diperlukan *output* digital tambahan selain 14 pin yang sudah tersedia. Untuk mengubah pin analog menjadi digital cukup mengubah konfigurasi pin pada program. Dalam *board* dapat dilihat pin digital diberi keterangan 0-13, jadi untuk menggunakan pin analog menjadi digital, pin analog yang pada keterangan *board* 0-5 kita ubah menjadi pin 14-19. dengan kata lain pin analog 0-5 berfungsi juga sebagai pin *output* digital 14-16.

Sifat *open source* arduino juga banyak memberikan keuntungan tersendiri untuk kita dalam menggunakan *board* ini, karena dengan sifat *open source* komponen yang kita pakai tidak hanya tergantung pada satu merek, namun memungkinkan kita bisa memakai semua komponen yang ada dipasaran.

Tabel 1. Speksifikasi Arduino Uno ATmega328

Mikrokontroler	ATmega 328
Tegangan Pengoperasian	5 V
Tegangan Input yang disarankan	7 – 12 V
Batas Tegangan Input	6 – 20 V
Jumlah pin I/O digital	14 pin digital (6 diantaranya menyediakan keluaran PWM)
Jumlah pin input Analog	6 pin
Arus DC tiap pin I/O	40mA
Arus DC untuk pin 3,3 V	50mA
Memori Flash	32 KB (ATmega 328) sekitar 0,5 KB digunakan oleh bootloader
SRAM	2 KB (ATmega 328)
EPROM	1 KB (ATmega 328)
Clock Speed	16 MHz

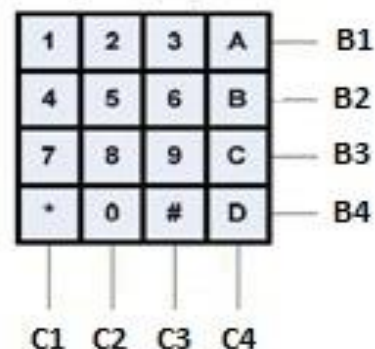


Gambar 2.4 Konfigurasi Pin-Pin ATmega328

Keypad Matriks 4x4

Keypad adalah "jenis perangkat" berisi tombol memungkinkan pengguna untuk memasukkan angka, huruf atau simbol pada kontroler. Keypad sesungguhnya terdiri dari sejumlah saklar, yang terhubung sebagai baris dan kolom dengan susunan seperti yang ditunjukkan pada gambar 4. Agar mikrokontroler dapat melakukan scan keypad, maka port mengeluarkan salah satu bit dari 4 bit yang terhubung pada kolom dengan logika low "0" dan selanjutnya membaca 4 bit pada baris untuk menguji jika ada tombol yang ditekan pada kolom tersebut. Sebagai konsekuensi, selama tidak ada tombol yang ditekan, maka mikrokontroler akan melihat sebagai

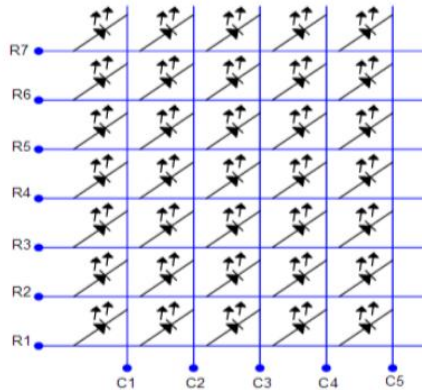
logika high "1" pada setiap pin yang terhubung ke baris.



Gambar 2. Rangkaian Ekuivalen Keypad 4x4

LED Dot Mtarix

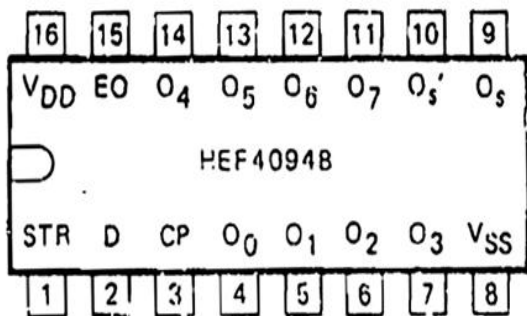
LED dot matrix adalah led yang disambung dan dirangkai membentuk kolom dan baris dengan jumlah tertentu sehingga jika lampu menyala akan membentuk sebuah karakter berupa huruf, angka, tanda baca, dan lain sebagainya. LED mempunyai dua kaki, yaitu anoda dan katoda. yang dimana untuk mengaktifkan LED tersebut kaki anoda kita beri tegangan VCC dan kaki katoda kita hubungkan ke ground.



Gambar 3. LED dot Matrix

IC 4094

IC (Integrated Circuit) adalah komponen elektronika aktif yang terdiri dari gabungan banyak komponen seperti resistor, kapasitor, transistor, dioda dan bahan ssemikonduktor lainnya yang diintegrasikan menjadi suatu rangkaian elektronika dalam sebuah kemasan kecil.



Gambar 4. Pin-pin IC 4094

Ic 4094 berfungsi untuk memindahkan data dari saluran serial ke saluran paralel dengan pergeseran bit Q0 sampai bit Q7 menuju output. Output paralel dapat dihubungkan langsung dengan LED.

Adaptor

Secara umum adaptor adalah sebuah rangkaian elektronika yang berguna untuk mengubah tegangan AC yang tinggi menjadi DC yang rendah. Adaptor merupakan sebuah alternatif pengganti dari tegangan DC (seperti: baterai, Aki) karena penggunaan tegangan AC lebih lama dan setiap orang dapat

menggunakannya asalkan ada aliran listrik di tempat tersebut.



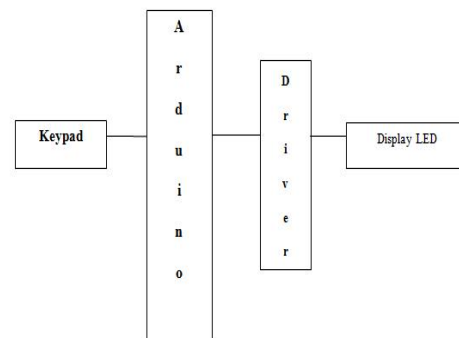
Gambar 5. Adaptor

Untuk keluaran adptor sendiri bermacam-macam ada yang 5V, 9V, 12V, dan sebagainya sesuai kebutuhan.

3. PERANCANGAN SISTEM

Diagram Blok Sistem

Blok diagram display LED dot matriks sebagai penampil informasi tombol keypad terprogram berbasis mikrokontroler arduino uno dirancang terlebih dulu secara blok diagram seperti ditunjukkan pada gambar 7.



Gambar 7. Diagram Blok Sistem penampil informasi terprogram

Keypad 4 x 4

Tombol pilih informasi yang akan ditampilkan di display LED dot matriks.

Mikro Arduino Uno

Pusat pengola bit data yang diterima dari keypad dan mengeluarkan bit data informasi ditampilkan di display.

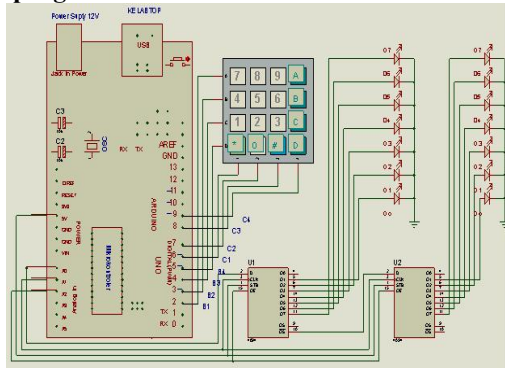
Driver 4094

Menerima data secara serial dari mikrokontroler dan mengeluarkannya secara paralel (SIPO).

Display LED dot matriks

Menkonversikan sinyal bit data menjadi titik cahaya membentuk karakter/ tulisan.

Rangkaian Display Dot Matriks Tombol Pilih Terprogram



Gambar 8. Rangkaian Display Dot Matriks Tombol Pilih Terprogram

Rangkaian *keypad dot* matriks berfungsi sebagai untuk memilih informasi yang ditampilkan pada display LED. Saklar pada *keypad dot* matriks dirancang untuk dapat menampilkan kata: BUKA, TAMU, MASUK, RAPAT, TUTUP, TENANG, KELUAR, SOLAT, KULIAH, MAKAN, TUNGGU, ISTIRAHAT. pada display. Pemakaian tombol pada keypad dot matriks dirancang sebagai berikut:

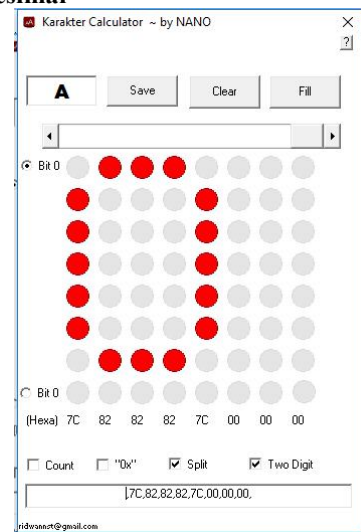
- Keypad 1 sebagai saklar pilih penampil kata BUKA
- Keypad 2 sebagai saklar pilih penampil kata TAMU
- Keypad 3 sebagai saklar pilih penampil kata MASUK
- Keypad 4 sebagai saklar pilih penampil kata RAPAT
- Keypad 5 sebagai saklar pilih penampil kata TUTUP
- Keypad 6 sebagai saklar pilih penampil kata TENANG
- Keypad 7 sebagai saklar pilih penampil kata KELUAR
- Keypad 8 sebagai saklar pilih penampil kata SOLAT
- Keypad 9 sebagai saklar pilih penampil kata KULIAH
- Keypad 0 sebagai saklar pilih penampil kata MAKAN
- Keypad A sebagai saklar pilih penampil kata TUNGGU
- Keypad B sebagai saklar pilih penampil kata ISTIRAHAT
- Keypad * sebagai saklar pilih mengosongkan/membersihkan layar display.

Keypad dot matriks yang digunakan terdiri dari 16 saklar (4 baris dan 4 kolom), sambungan baris dan kolom *keypad dot* matriks ke minimum sistem mikrokontroler arduino uno.

Jumlah LED yang digunakan sebanyak 7 baris x 16 kolom LED atau 112 buah LED dan 16 IC 4094.

Gambar 3.4 menunjukkan rangkaian dasar *display dot* matriks. Tujuh buah LED dan satu IC 4094 untuk menghasilkan 1 kolom dan 7 baris display. Informasi yang dapat ditampilkan oleh display dot matriks berbasis mikrokontroler arduino uno setelah diprogram sebelumnya adalah BUKA, TAMU, MASUK, RAPAT, TUTUP, TENANG, KELUAR, SOLAT, KULIAH, MAKAN, TUNGGU, ISTIRAHAT. Untuk memperagakan karakter dalam tampilan LED matrik yang digunakan tidak dinyalakan secara serentak, akan tetapi LED ini dinyalakan kolom demi kolom dengan cepat.

Konversi Pola Karakter Nol ke Heksadesimal



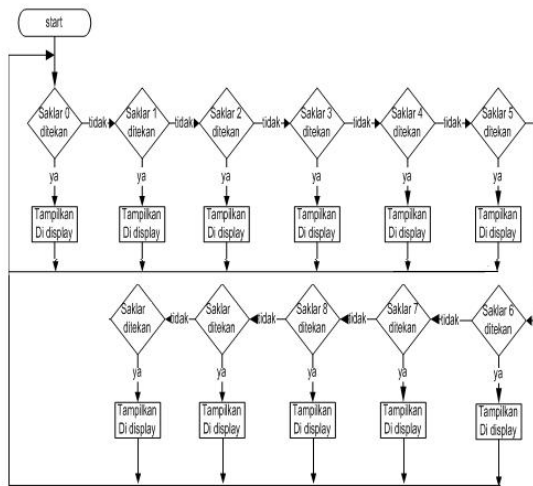
Gambar 10. Konversi Pola Karakter Nol ke Heksadesimal

Dari Gambar 3.7. dapat dijelaskan bahwa karakter nol dikonversikan ke bilangan heksadesimal 7C, 82, 82, 82, 7C, 00. Data heksadesimal inilah yang diketikkan pada program yang akan diupload ke mikrokontroler arduino uno berbantuan laptop dengan menggunakan aplikasi bernama Arduino IDE, pengetikannya seperti ini :

```
Disp.setByte(idx,0x7C); idx++;
Disp.setByte(idx,0x82); idx++;
Disp.setByte(idx,0x82); idx++;
Disp.setByte(idx,0x82); idx++;
Disp.setByte(idx,0x7C); idx++;
Break;
```

Diagram Alir Program

Setelah dilakukan perancangan alat maka langkah selanjutnya adalah membuat modul program untuk membangun sistem penampil informasi terprogram. Dalam proses perancangan program ini diawali dengan menentukan logika yang mendasari program tersebut, dimana pada perancangan alat ini yang digunakan adalah diagram alir dapat ditunjukkan pada Gambar 11.



Gambar 11. Diagram Alir Program

Program diawali dengan start yang artinya program dimulai. Kemudian program akan mengecek apakah ada tombol yang ditekan, jika ada, maka akan ditampilkan pada display.

4. PENGUJIAN DAN REALISASI

Pengujian Alat

Setelah perangkat keras dan perangkat lunak sudah selesai dirancang bangun, maka tahap selanjutnya ialah pengujian sistem. Pengujian dilakukan dengan dua cara:

1. saklar pada keypad dot matriks ditekan mulai dari angka 0 s/d 9; A, B, C, D, E, bintang (*), dan pagar (#). Setelah tampil informasi di display selanjutnya direset dengan cara tekan tombol pagar (#).

Pengujian dan analisis dilakukan untuk membuktikan apakah rangkaian bekerja sesuai dengan apa yang direncanakan.

Pengujian Tombol Pilih Terprogram

Pengujian display dilakukan untuk mengetahui baik tidaknya display dan apakah sudah bisa menampilkan informasi sesuai dengan tombol pilih yang ditekan. Pengujian dilakukan mulai dari tombol angka 0 s/d 9; A, B, C, D, tanda bintang (*) dan pagar (#).

Dan untuk programnya adalah sebagai berikut :

```

#include <Keypad.h>
#include <NanoTechDisp4094.h>

const byte ROWS = 4;
const byte COLS = 4;
char hexaKeys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {2, 3, 4, 5};

```

```
byte colPins[COLS] = {6, 7, 8, 9};
```

```

Keypad customKeypad =
Keypad( makeKeymap(hexaKeys), rowPins, colPins,
ROWS, COLS);
NanoTech Disp(1,A0,A2,A1);// jlh
board,data,clock,strobe
int maxNum;

```

```

void setup() {
  Serial.begin(9600);
  Disp.init();
}

```

```

void loop() {
  char customKey = customKeypad.getKey();

```

```

if (customKey) {
  if (customKey == '*') {
    Disp.buffClear();
    Disp.show();
    maxNum = 0;
  } else {
    Serial.println(customKey);
    maxNum = initChar(maxNum,customKey);
    Disp.show();
  }
}
}

```

```

int initChar(int idx,char chr) {
  switch(chr) {
    case ' ':
      Disp.setByte(idx,0x0); idx++;
      Disp.setByte(idx,0x0); idx++;
      Disp.setByte(idx,0x0); idx++;
      break;

    case '1':
      Disp.setByte(idx,0xFE); idx++;
      Disp.setByte(idx,0x92); idx++;
      Disp.setByte(idx,0x6C); idx++;
      Disp.setByte(idx,0x0); idx++;
      Disp.setByte(idx,0xFC); idx++;
      Disp.setByte(idx,0x02); idx++;
      Disp.setByte(idx,0xFC); idx++;
      Disp.setByte(idx,0x0); idx++;
      Disp.setByte(idx,0xFE); idx++;
      Disp.setByte(idx,0x28); idx++;
      Disp.setByte(idx,0x44); idx++;
      Disp.setByte(idx,0x82); idx++;
      Disp.setByte(idx,0x0); idx++;
      Disp.setByte(idx,0x7E); idx++;
      Disp.setByte(idx,0x90); idx++;
      Disp.setByte(idx,0x7E); idx++;
      Disp.setByte(idx,0x0); idx++;

```

```

break;

case '2':
    Disp.setByte(idx,0x80); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x80); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0x7E); idx++;
    Disp.setByte(idx,0x88); idx++;
    Disp.setByte(idx,0x7E); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x20); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0xFC); idx++;
    Disp.setByte(idx,0x02); idx++;
    Disp.setByte(idx,0xFC); idx++;
    Disp.setByte(idx,0x0); idx++;
    break;

case '3':
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x20); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0x7E); idx++;
    Disp.setByte(idx,0x88); idx++;
    Disp.setByte(idx,0x7E); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0x62); idx++;
    Disp.setByte(idx,0x92); idx++;
    Disp.setByte(idx,0x8C); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0xFC); idx++;
    Disp.setByte(idx,0x02); idx++;
    Disp.setByte(idx,0xFC); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x10); idx++;
    Disp.setByte(idx,0x28); idx++;
    Disp.setByte(idx,0x44); idx++;

    break;

case '4':
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x90); idx++;
    Disp.setByte(idx,0xEE); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0x7E); idx++;
    Disp.setByte(idx,0x88); idx++;
    Disp.setByte(idx,0x7E); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x90); idx++;
    Disp.setByte(idx,0xF0); idx++;

    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0x7E); idx++;
    Disp.setByte(idx,0x88); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x80); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0x92); idx++;

    case '5':
        Disp.setByte(idx,0x80); idx++;
        Disp.setByte(idx,0xFE); idx++;
        Disp.setByte(idx,0x80); idx++;
        Disp.setByte(idx,0x0); idx++;
        Disp.setByte(idx,0xFC); idx++;
        Disp.setByte(idx,0x02); idx++;
        Disp.setByte(idx,0xFC); idx++;
        Disp.setByte(idx,0x0); idx++;
        Disp.setByte(idx,0x80); idx++;
        Disp.setByte(idx,0xFE); idx++;
        Disp.setByte(idx,0x80); idx++;
        Disp.setByte(idx,0x0); idx++;
        Disp.setByte(idx,0xFC); idx++;
        Disp.setByte(idx,0x02); idx++;
        Disp.setByte(idx,0xFC); idx++;
        Disp.setByte(idx,0x0); idx++;
        Disp.setByte(idx,0xFE); idx++;
        Disp.setByte(idx,0x80); idx++;
        Disp.setByte(idx,0x0); idx++;
        Disp.setByte(idx,0x90); idx++;
        Disp.setByte(idx,0xF0); idx++;
        Disp.setByte(idx,0x0); idx++;
        break;

    case '6':
        Disp.setByte(idx,0x80); idx++;
        Disp.setByte(idx,0xFE); idx++;
        Disp.setByte(idx,0x80); idx++;
        Disp.setByte(idx,0x0); idx++;
        Disp.setByte(idx,0xFE); idx++;
        Disp.setByte(idx,0x92); idx++;
        Disp.setByte(idx,0x92); idx++;
        Disp.setByte(idx,0x0); idx++;
        Disp.setByte(idx,0xFE); idx++;
        Disp.setByte(idx,0x80); idx++;
        Disp.setByte(idx,0x0); idx++;
        Disp.setByte(idx,0xFE); idx++;
        Disp.setByte(idx,0x0); idx++;
        Disp.setByte(idx,0xFE); idx++;
        Disp.setByte(idx,0x92); idx++;

```

```

Disp.setByte(idx,0x9E); idx++;

break;

case '7':
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x10); idx++;
    Disp.setByte(idx,0x28); idx++;
    Disp.setByte(idx,0x44); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x92); idx++;
    Disp.setByte(idx,0x92); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x02); idx++;
    Disp.setByte(idx,0x02); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0xFC); idx++;
    Disp.setByte(idx,0x02); idx++;
    Disp.setByte(idx,0xFC); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0x7E); idx++;
    Disp.setByte(idx,0x88); idx++;
    Disp.setByte(idx,0x7E); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x90); idx++;
    Disp.setByte(idx,0xEE); idx++;
    Disp.setByte(idx,0x0); idx++;
    break;

case '8':
    Disp.setByte(idx,0x62); idx++;
    Disp.setByte(idx,0x92); idx++;
    Disp.setByte(idx,0x8C); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x82); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x02); idx++;
    Disp.setByte(idx,0x02); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0x7E); idx++;
    Disp.setByte(idx,0x88); idx++;
    Disp.setByte(idx,0x7E); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0x80); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x80); idx++;
    Disp.setByte(idx,0x0); idx++;

case '9':
    Disp.setByte(idx,0xFE); idx++;

```

```

Disp.setByte(idx,0x10); idx++;
Disp.setByte(idx,0x28); idx++;
Disp.setByte(idx,0x44); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFC); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0xFC); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x82); idx++;
Disp.setByte(idx,0xFC); idx++;
Disp.setByte(idx,0x82); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x7E); idx++;
Disp.setByte(idx,0x88); idx++;
Disp.setByte(idx,0x7E); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x10); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x0); idx++;

break;

case '0':
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x20); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0x7E); idx++;
    Disp.setByte(idx,0x88); idx++;
    Disp.setByte(idx,0x7E); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x10); idx++;
    Disp.setByte(idx,0x28); idx++;
    Disp.setByte(idx,0x44); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0x7E); idx++;
    Disp.setByte(idx,0x88); idx++;
    Disp.setByte(idx,0x7E); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x80); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x0); idx++;

break;

case 'A':
    Disp.setByte(idx,0x80); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x80); idx++;
    Disp.setByte(idx,0x0); idx++;

```



```

Disp.setByte(idx,0xFC); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0xFC); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x92); idx++;
Disp.setByte(idx,0x9E); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x92); idx++;
Disp.setByte(idx,0x9E); idx++;
Disp.setByte(idx,0xFC); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0xFC); idx++;

```

```
break;
```

```
case 'B':
```

```

Disp.setByte(idx,0x82); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x82); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x62); idx++;
Disp.setByte(idx,0x92); idx++;
Disp.setByte(idx,0x8C); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x82); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x82); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x90); idx++;
Disp.setByte(idx,0xEE); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x10); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x7E); idx++;
Disp.setByte(idx,0x88); idx++;
Disp.setByte(idx,0x7E); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0x0); idx++;
break;

```

```
}
```

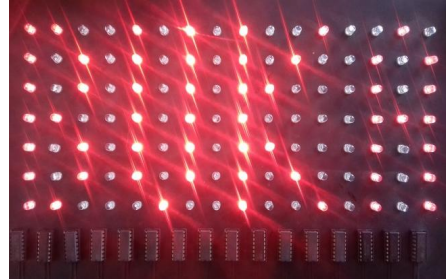
```

return idx;
}

```

Pengujian keypad tombol satu (BUKA)

Pengujian ini dilakukan untuk mengetahui hubungan tombol keypad angka 1 dalam menampilkan kata BUKA pada display, hasil pengujiannya seperti diperlihatkan pada Gambar 12



Gambar 12. Display Tampilkan Tombol Buka

Adapun program untuk dapat menampilkan kata BUKA adalah sebagai berikut:

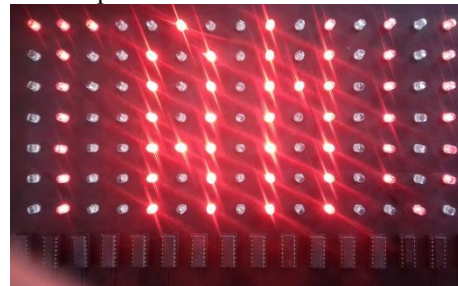
```

case '1':
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x92); idx++;
Disp.setByte(idx,0x6C); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFC); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0xFC); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x28); idx++;
Disp.setByte(idx,0x44); idx++;
Disp.setByte(idx,0x82); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x7E); idx++;
Disp.setByte(idx,0x90); idx++;
Disp.setByte(idx,0x7E); idx++;
Disp.setByte(idx,0x0); idx++;
break;

```

Pengujian keypad tombol dua (TAMU)

Pengujian ini dilakukan untuk mengetahui hubungan tombol keypad 2 dalam menampilkan kata TAMU pada display, hasil pengujiannya seperti diperlihatkan pada Gambar 13.



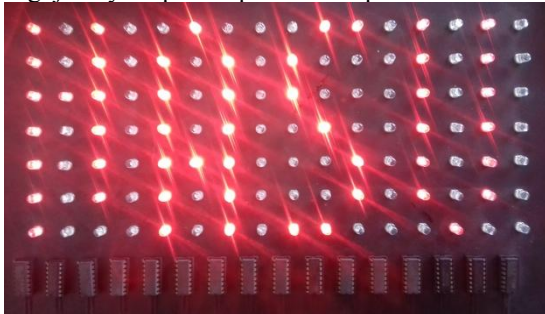
Gambar 13. Display Tampilkan Kata TAMU

Adapun program untuk dapat menampilkan kata TAMU adalah sebagai berikut:

```
case '2':
  Disp.setByte(idx,0x80); idx++;
  Disp.setByte(idx,0xFE); idx++;
  Disp.setByte(idx,0x80); idx++;
  Disp.setByte(idx,0x0); idx++;
  Disp.setByte(idx,0x7E); idx++;
  Disp.setByte(idx,0x88); idx++;
  Disp.setByte(idx,0x7E); idx++;
  Disp.setByte(idx,0x0); idx++;
  Disp.setByte(idx,0xFE); idx++;
  Disp.setByte(idx,0x20); idx++;
  Disp.setByte(idx,0xFE); idx++;
  Disp.setByte(idx,0x0); idx++;
  Disp.setByte(idx,0xFC); idx++;
  Disp.setByte(idx,0x02); idx++;
  Disp.setByte(idx,0xFC); idx++;
  Disp.setByte(idx,0x0); idx++;
  break;
```

Pengujian Keypad Tombol 3 (MASUK)

Pengujian ini dilakukan untuk mengetahui hubungan tombol keypad angka 3 dalam menampilkan kata MASUK pada display, hasil pengujiannya seperti diperlihatkan pada Gambar 14.



Gambar 14. Display Tampilkan Kata MASUK

Adapun program untuk dapat menampilkan kata MASUK adalah sebagai berikut:

```
case '3':
  Disp.setByte(idx,0xFE); idx++;
  Disp.setByte(idx,0x20); idx++;
  Disp.setByte(idx,0xFE); idx++;
  Disp.setByte(idx,0x0); idx++;
  Disp.setByte(idx,0x7E); idx++;
  Disp.setByte(idx,0x88); idx++;
  Disp.setByte(idx,0x7E); idx++;
  Disp.setByte(idx,0x0); idx++;
  Disp.setByte(idx,0x62); idx++;
  Disp.setByte(idx,0x92); idx++;
  Disp.setByte(idx,0x8C); idx++;
  Disp.setByte(idx,0x0); idx++;
  Disp.setByte(idx,0xFC); idx++;
  Disp.setByte(idx,0x02); idx++;
  Disp.setByte(idx,0xFC); idx++;
  Disp.setByte(idx,0x0); idx++;
```

```
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x10); idx++;
Disp.setByte(idx,0x28); idx++;
Disp.setByte(idx,0x44); idx++;
break;
```

a. Pengujian keypad tombol empat (RAPAT).

Pengujian ini dilakukan untuk mengetahui hubungan tombol keypad angka 4 dalam menampilkan kata RAPAT pada display, hasil pengujiannya seperti diperlihatkan pada gambar 4.4.



Gambar 4.4 Display Tampilkan Kata RAPAT

Adapun program untuk dapat menampilkan kata RAPAT adalah sebagai berikut:



case '4':

```
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x90); idx++;
Disp.setByte(idx,0x90); idx++;
Disp.setByte(idx,0x6E); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x7E); idx++;
Disp.setByte(idx,0x90); idx++;
Disp.setByte(idx,0x90); idx++;
Disp.setByte(idx,0x90); idx++;
Disp.setByte(idx,0x7E); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x90); idx++;
Disp.setByte(idx,0x90); idx++;
Disp.setByte(idx,0x60); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x7E); idx++;
Disp.setByte(idx,0x90); idx++;
Disp.setByte(idx,0x90); idx++;
Disp.setByte(idx,0x7E); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x80); idx++;
```

```
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0x0); idx++;
```

```
break;
```

b. Pengujian keypad tombol lima (TUTUP).

Pengujian ini dilakukan untuk mengetahui hubungan tombol keypad angka 5 dalam menampilkan kata TUTUP pada display, hasil pengujiannya seperti diperlihatkan pada gambar 4.5.

Gambar 4.5 Display Tampilkan Kata TUTUP.

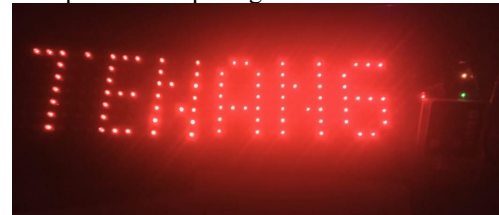
Adapun program untuk dapat menampilkan kata TUTUP adalah sebagai berikut:

case '5':

```
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFC); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0xFC); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFC); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0xFC); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x90); idx++;
Disp.setByte(idx,0x90); idx++;
Disp.setByte(idx,0x60); idx++;
Disp.setByte(idx,0x0); idx++;
break;
```

c. Pengujian keypad tombol enam (TENANG).

Pengujian ini dilakukan untuk mengetahui hubungan tombol keypad angka 6 dalam menampilkan kata TENANG pada display, hasil pengujiannya seperti diperlihatkan pada gambar 4.6.



Gambar 4.6 Display Tampilkan Kata TENANG.

Adapun program untuk dapat menampilkan kata TENANG adalah sebagai berikut:

case '6':

```
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x92); idx++;
Disp.setByte(idx,0x92); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x20); idx++;
Disp.setByte(idx,0x08); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x7E); idx++;
Disp.setByte(idx,0x88); idx++;
Disp.setByte(idx,0x88); idx++;
Disp.setByte(idx,0x7E); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x20); idx++;
Disp.setByte(idx,0x08); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x7C); idx++;
Disp.setByte(idx,0x92); idx++;
Disp.setByte(idx,0x92); idx++;
Disp.setByte(idx,0x9C); idx++;
break;
```

d. Pengujian keypad tombol tujuh (KELUAR).

Pengujian ini dilakukan untuk mengetahui hubungan tombol keypad angka 7 dalam menampilkan kata KELUAR pada display, hasil

pengujiannya seperti diperlihatkan pada gambar 4.7.



Gambar 4.7 Display Tampilkan Kata KELUAR.

Adapun program untuk dapat menampilkan kata KELUAR adalah sebagai berikut:

case '7':

```
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x10); idx++;
Disp.setByte(idx,0x28); idx++;
Disp.setByte(idx,0x44); idx++;
Disp.setByte(idx,0x82); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x92); idx++;
Disp.setByte(idx,0x92); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFC); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0xFC); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x7E); idx++;
Disp.setByte(idx,0x88); idx++;
Disp.setByte(idx,0x88); idx++;
Disp.setByte(idx,0x7E); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x90); idx++;
Disp.setByte(idx,0x90); idx++;
Disp.setByte(idx,0x6E); idx++;
Disp.setByte(idx,0x0); idx++;
break;
```

e. Pengujian keypad tombol delapan (SOLAT).

Pengujian ini dilakukan untuk mengetahui hubungan tombol keypad angka 8 dalam menampilkan kata SOLAT pada display, hasil pengujiannya seperti diperlihatkan pada gambar 4.8.



Gambar 4.8 Display Tampilkan Kata SOLAT.

Adapun program untuk dapat menampilkan kata SOLAT adalah sebagai berikut:

case '8':

```
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x62); idx++;
Disp.setByte(idx,0x92); idx++;
Disp.setByte(idx,0x8C); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x7C); idx++;
Disp.setByte(idx,0x82); idx++;
Disp.setByte(idx,0x82); idx++;
Disp.setByte(idx,0x7C); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x7E); idx++;
Disp.setByte(idx,0x88); idx++;
Disp.setByte(idx,0x88); idx++;
Disp.setByte(idx,0x7E); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0x0); idx++;
break;
```

f. Pengujian keypad tombol Sembilan (KULIAH).

Pengujian ini dilakukan untuk mengetahui hubungan tombol keypad angka 8 dalam menampilkan kata KULIAH pada display, hasil pengujiannya seperti diperlihatkan pada gambar 4.9.



Gambar 4.9 Display Tampilkan Kata KULIAH.

Adapun program untuk dapat menampilkan kata KULIAH sebagai berikut:

```
case '9':
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x10); idx++;
    Disp.setByte(idx,0x28); idx++;
    Disp.setByte(idx,0x44); idx++;
    Disp.setByte(idx,0x82); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0xFC); idx++;
    Disp.setByte(idx,0x02); idx++;
    Disp.setByte(idx,0x02); idx++;
    Disp.setByte(idx,0xFC); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x02); idx++;
    Disp.setByte(idx,0x02); idx++;
    Disp.setByte(idx,0x02); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0x82); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x82); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0x7E); idx++;
    Disp.setByte(idx,0x88); idx++;
    Disp.setByte(idx,0x88); idx++;
    Disp.setByte(idx,0x7E); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x10); idx++;
    Disp.setByte(idx,0x10); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x0); idx++;

break;
```

g. Pengujian keypad tombol A (TUNGGU).

Pengujian ini dilakukan untuk mengetahui hubungan tombol keypad angka 8 dalam menampilkan kata TUNGGU pada display, hasil pengujiannya seperti diperlihatkan pada gambar 4.10.



Gambar 4.10 Display Tampilkan Kata TUNGGU.

Adapun program untuk dapat menampilkan kata TUNGGU adalah sebagai berikut: case 'A':

```
case 'A':
    Disp.setByte(idx,0x80); idx++;
    Disp.setByte(idx,0x80); idx++;
    Disp.setByte(idx,0xFE); idx++;
```

```
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFC); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0xFC); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x20); idx++;
Disp.setByte(idx,0x08); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x7C); idx++;
Disp.setByte(idx,0x92); idx++;
Disp.setByte(idx,0x92); idx++;
Disp.setByte(idx,0x9C); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x7C); idx++;
Disp.setByte(idx,0x92); idx++;
Disp.setByte(idx,0x92); idx++;
Disp.setByte(idx,0x9C); idx++;
Disp.setByte(idx,0x00); idx++;
Disp.setByte(idx,0xFC); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0xFC); idx++;
```

break;

h. Pengujian keypad tombol B (ISTIRAHAT).

Pengujian ini dilakukan untuk mengetahui hubungan tombol keypad angka 8 dalam menampilkan kata ISTIRAHAT pada display, hasil pengujiannya seperti diperlihatkan pada gambar 4.11.



Gambar 4.11. Display Tampilkan Kata ISTIRAHAT.

Adapun program untuk dapat menampilkan kata ISTIRAHAT adalah sebagai berikut:

```
case 'B':
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x0); idx++;
    Disp.setByte(idx,0x62); idx++;
    Disp.setByte(idx,0x92); idx++;
    Disp.setByte(idx,0x8C); idx++;
    Disp.setByte(idx,0x0); idx++;
```

```

Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0x00); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x00); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x90); idx++;
Disp.setByte(idx,0x6E); idx++;
Disp.setByte(idx,0x00); idx++;
Disp.setByte(idx,0x7E); idx++;
Disp.setByte(idx,0x88); idx++;
Disp.setByte(idx,0x7E); idx++;
Disp.setByte(idx,0x00); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x10); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x00); idx++;
Disp.setByte(idx,0x7E); idx++;
Disp.setByte(idx,0x88); idx++;
Disp.setByte(idx,0x7E); idx++;
Disp.setByte(idx,0x00); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x80); idx++;
Disp.setByte(idx,0x00); idx++;

break;
}
return idx;
}

```

5. KESIMPULAN

Setelah penulis mengadakan pengujian dan analisis hasil dari perancangan display LED penampil informasi, maka dapat ditarik kesimpulan:

1. Display LED yang dirangkai dalam bentuk dot matriks dapat menampilkan kata BUKA, TAMU, MASUK, RAPAT, TUTUP, TENANG, KELUAR, SOLAT, KULIAH, MAKAN, TUNGGU, ISTIRAHAT.
2. Mikrokontroler arduino uno menerima data 8 bit dari tombol keypad dot matriks dan menampilkan informasi pada display LED sesuai dengan tombol yang dipilih.
3. Kata informasi yang ditampilkan display dapat diganti melalui program yang diupload ke memori mikrokontroler arduino uno.
4. Arduino memiliki kelebihan tersendiri dibanding *board* mikrokontroler yang lain selain bersifat *open source*, arduino juga mempunyai bahasa pemrogramannya sendiri yang berupa bahasa C (cobol). Selain itu dalam *board* arduino sendiri sudah terdapat *loader* yang berupa USB sehingga

memudahkan dalam memprogram mikrokontroler yang ada di arduino

5. Adaptor adalah sebuah rangkaian yang berguna untuk mengubah tegangan AC yang tinggi menjadi DC yang rendah. Adaptor merupakan sebuah alternatif pengganti dari tegangan DC (seperti ;baterai,Aki)

Saran

1. Alat yang saya rancang bangun, bahwa data bit dari keypad dot matriks 4 x 4 dikirim ke mikrokontroler melalui kabel maka diharapkan dipenelitian berikutnya bisa dikirim tanpa kabel (nir kabel).
2. Perancangan selanjutnya diharapkan memakai modul LED matriks sehingga tampilan informasi yang dimunculkan lebih bagus/ cantik.
3. Tombol pilih informasi yang ditampilkan pada display LED dot matriks masih menggunakan keypad dot matriks 4 x 4, dipeneliti berikutnya diharapkan dapat menggunakan tuts keypad *hand phone(remot)* dan *keyboard* komputer.

DAFTAR PUSTAKA

Aldyrazor, <https://www.aldyrazor.com/2020/04/gambar-arduino-uno.html>

Arduino-keypad4x4,
<https://www.teachmemicro.com/arduino-keypad-interfacing-4x4-matrix/>

Labelektronika, http://www.labelektronika.com/2017/02/arduino-uno-mikrokontroler-atmega-328.html#google_vignette

Melani Satyoadi, Ir. 2003, “*Elektronika Digital*”, Penerbit Andi, Yogyakarta.

Pulus Andi Nalwan, 2002, “*Panduan Praktis Teknik Antarmuka Dan Pemrograman Mikrokontroler AT89S51*”, PT Alex Komputindo, Gramedia Jakarta

Rizal Riskiawan, 1997, “*Tutorial Perancangan Hardware II*”, PT Elex Media Komputindo, Jakarta

Sulhan Setiawan, 2006, “*Mudah Dan Menyenangkan Belajar Mikrokontroler*”, Penerbit Andi, Yogyakarta