

PERANCANGAN DISPLAY LED DOT MATRIKS BERBASIS ARDUINO UNO YANG DAPAT DIFUNGSIKAN SEBAGAI PENAMPIL TULISAN BERJALAN DAN PENAMPIL INFORMASI TOMBOL PILIH TERPROGRAM

Kolombus Siringo ringo, Jeremia Siregar dan Alviando

Fakultas Teknologi Industri
Institut Sains Teknologi T.D Pardede, Jl.DR.TD.Pardede No. 8, Medan 20153

Email : kolombussiringo@yahoo.com
jeremiasiregar@istp.ac.id
alviando2804@gmail.com

ABSTRAK

Sekarang ini penyampaian informasi dilakukan dengan menggunakan spanduk, baliho, hanya bisa digunakan sekali pakai sedangkan dengan menggunakan media elektronik lainnya seperti radio dan televisi penyampaian informasi akan sangat mahal. Pada tugas akhir ini penulis mencoba membuat suatu alat yaitu running text, running text adalah suatu media untuk menyampaikan informasi lebih menarik dan dapat diubah sesuai dengan keinginan pemakai.

Proses dilakukan oleh mikro kontroler ATmega328 yang merupakan fasilitas pengaksesan data melalui koneksi keypad 4x4 digunakan untuk pengiriman data. Kemudian program diterima oleh mikrokontroler Arduino Uno lalu dikirim ke ic 4094 ke led dot matrix.

Kata Kunci : Arduino uno, Mikrokontroler, Ic 4094, Keypad, LED.

ABSTRACT

Currently the delivery of information is done using banners, billboards, can only be used once, while using other electronic media such as radio and television the delivery of information will be very expensive. In this final project the author tries to make a tool that is running text, running text is a medium to convey information more interesting and can be changed according to the wishes of the user.

The process is carried out by the ATmega328 micro controller which is a data access facility via a 4x4 keypad connection used for data transmission. Then the program is received by the Arduino Uno microcontroller and then sent to the ic 4094 to the led dot matrix.

Keywords: Arduino uno, Microcontroller, Ic 4094, Keypad, LED.

1. PENDAHULUAN

Perkembangan teknologi elektronika khususnya bidang teknologi mikrokontroler arduino uno telah membuat

peningkatan kuantitas maupun kualitas peralatan listrik/ elektronika. Penerapan teknologi mikrokontroler arduino uno pada rangkaian elektronika membuat rangkaian menjadi lebih sederhana dan memudahkan dalam perakitannya. Mikrokontroler arduino uno sudah banyak diterapkan pada berbagai rangkaian elektronika peralatan listrik,

seperti misalnya pada display running teks dan lain-lain. Display LED dot matriks yang sudah berbasis mikrokontroler dapat difungsikan sebagai running teks, penampil informasi terprogram, lampu hias variatif, dan lain-lain. Dikatakan display LED dot matriks sebagai penampil informasi terprogram, bahwa informasi yang ditampilkan dapat diganti dengan cara menekan tombol pilih.

Display sebagai penampil informasi pada umumnya ditempatkan dipintu-pintu kantor yang digunakan untuk menyampaikan informasi misalnya

kata TUTUP, BUKA, MASUK, KELUAR, RAPAT, TENANG, TUNGGU, TAMU, KULIAH, dan lain sebagainya masih dilakukan secara manual. Dikatakan dilakukan secara manual karena harus diganti dengan cara dihapus dan ditulis ulang atau dibalik. Memperhatikan hal tersebut maka penulis merencanakan merancang bangun display LED dot matriks yang dapat menampilkan berbagai informasi dalam bentuk kata dengan cara cukup tekan tombol pilih. Berdasarkan penjelasan ini maka penulis mengangkat judul tugas akhir **“Perancangan Display Dot Matriks sebagai Penampil Informasi Terprogram Berbasis Mikrokontroler Arduino Uno”**. Mikrokontroler Arduino Uno berfungsi untuk menerima dan memproses sinyal bit data yang diterima dari tombol pilih dan selanjutnya dikirimkan sinyal bit ke LED dot matriks untuk ditampilkan tulisan.

Tujuan dan Manfaat

Tujuan dan manfaat dari penulisan ini adalah:

1. Merancang bangun display LED dot matriks berbasis mikrokontroler arduino uno untuk dapat menampilkan kata MASUK, KELUAR, SHOLAT, RAPAT, TUTUP, BUKA, TENANG, dan TAMU.
2. Merancang pemakaian keypad 4 x 4 sebagai tombol pilih terprogram untuk dapat menampilkan informasi yang akan ditampilkan.
3. Merancang bangun pemakaian modul mikrokontroler arduino uno untuk dapat menerima sinyal bit dari tombol pilih dan menampilkan informasi pada display dot matriks.
4. Merancang bangun program cobol yang digunakan menjadi *softdriver* display LED dot matriks berbasis mikrokontroler arduino uno.
5. Dan manfaat hasil perancangan ini dapat digunakan sebagai salah satu media penyampaian informasi terprogram.

2. TINJAUAN PUSTAKA

Arduino

Arduino adalah sebuah board mikrokontroller yang berbasis ATmega328. Arduino memiliki 14 pin input/output yang mana 6 pin dapat digunakan

Keypad Matriks 4x4

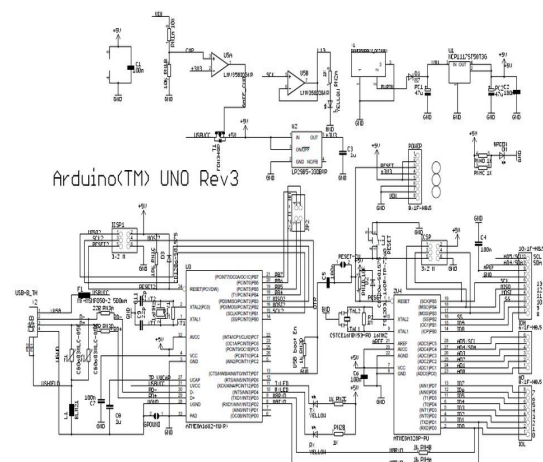
Keypad adalah "jenis perangkat" berisi tombol memungkinkan pengguna untuk memasukkan angka, huruf atau simbol pada kontroler. Keypad sesungguhnya terdiri dari sejumlah saklar, yang terhubung sebagai baris dan kolom dengan susunan seperti yang ditunjukkan pada gambar 4. Agar mikrokontroler dapat melakukan scan keypad, maka

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 bersifat open source, 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. Bahasa pemrograman arduino merupakan bahasa C yang sudah disederhanakan syntax bahasa pemrogramannya sehingga mempermudah kita dalam mempelajari dan mendalami mikrokontroler.

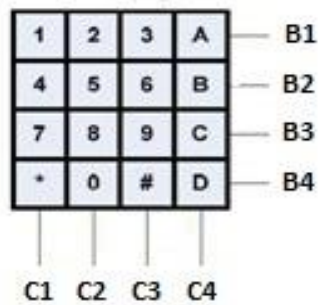
Tabel 1. Speksifikasi Arduino Uno ATmega328

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 1. Rangkaian Arduino Uno

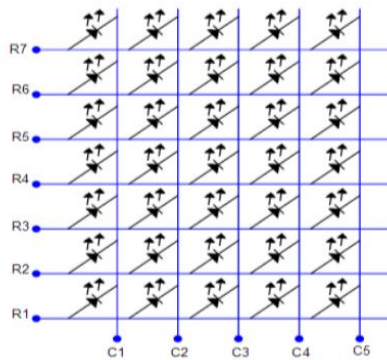
port mengeluarkan salah satu bit dari 4 bit yang terhubung pada kolom dengan logika low “0” dan selanjutnya membaca 4 bit pada baris untuk menguji jika ada tombol yang ditekan pada kolom tersebut. Sebagai konsekuensi, selama tidak ada tombol yang ditekan, maka mikrokontroler akan melihat sebagai logika high “1” pada setiap pin yang terhubung ke baris.



Gambar 2. Rangkaian Ekuivalen Keypad 4x4

LED Dot Matrix

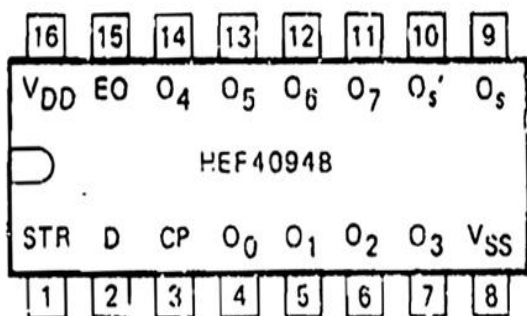
LED dot matrix adalah led yang disambung dan dirangkai membentuk kolom dan baris dengan jumlah tertentu sehingga jika lampu menyala akan membentuk sebuah karakter berupa huruf, angka, tanda baca, dan lain sebagainya. LED mempunyai dua kaki, yaitu anoda dan katoda. yang dimana untuk mengaktifkan LED tersebut 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 adaptor sendiri bermacam-macam ada yang 5V, 9V, 12V, dan sebagainya sesuai kebutuhan.

Buck Converter

Konverter jenis *buck* merupakan konverter penurun tegangan yang mengkonversikan tegangan masukan DC menjadi tegangan DC lainnya yang lebih rendah. Seperti terlihat pada gambar 2.14, rangkaian ini terdiri atas satu saklar aktif (MOSFET), satu saklar pasif (dioda), kapasitor dan induktor sebagai tapis keluarannya.



Gambar 6. Buck Converter LM2596

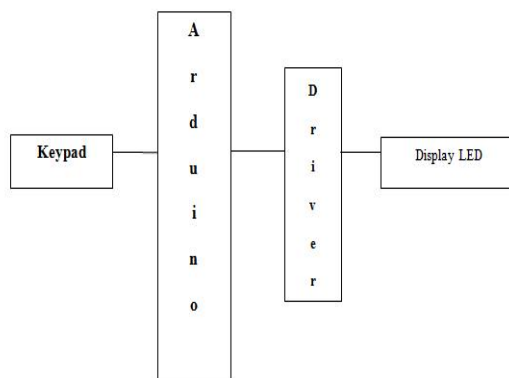
Untuk tegangan kerja yang rendah, saklar pasif (dioda) sering diganti dengan saklar aktif (MOSFET)

sehingga susut daya pada saklar bisa dikurangi. Apabila menggunakan 2 saklar aktif, kedua saklar ini akan bekerja secara bergantian, dan hanya ada satu saklar yang menutup setiap saat. Nilai rata-rata tegangan keluaran konverter sebanding dengan rasio antara waktu penutupan saklar (saklar konduksi/ON) terhadap periode penyaklarannya. Biasanya nilai faktor daya ini tidak lebih kecil dari 0.2, karena jika dioperasikan pada rasio tegangan yang lebih tinggi, saklar akan bekerja dibawah keandalannya dan menyebabkan efisiensi konverter turun. Untuk rasio (V_d/E_d) yang sangat tinggi, biasanya digunakan konverter DC-DC yang terisolasi atau topologi yang dilengkapi dengan trafo.

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.

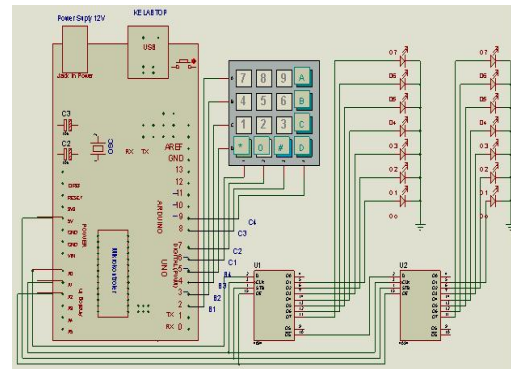
Driver 4094

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.

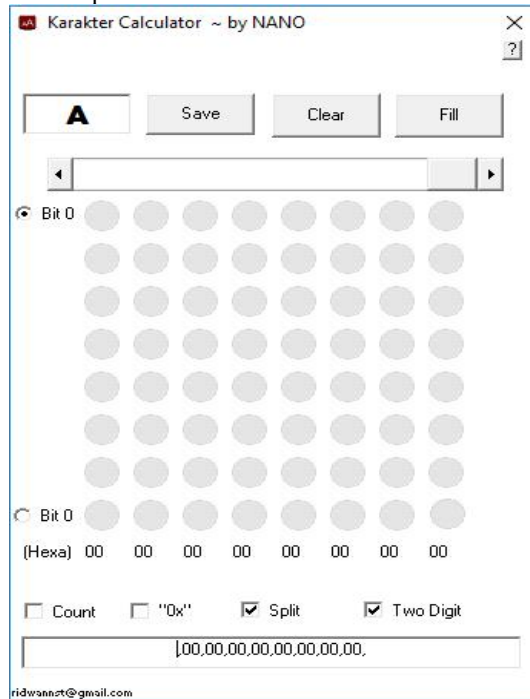
Keypad dot matriks yang digunakan terdiri dari 16 saklar (4 baris dan 4 kolom), sambungan baris dan kolom *keypad dot* matriks ke minimum sistem mikrokontroler arduino uno.

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.

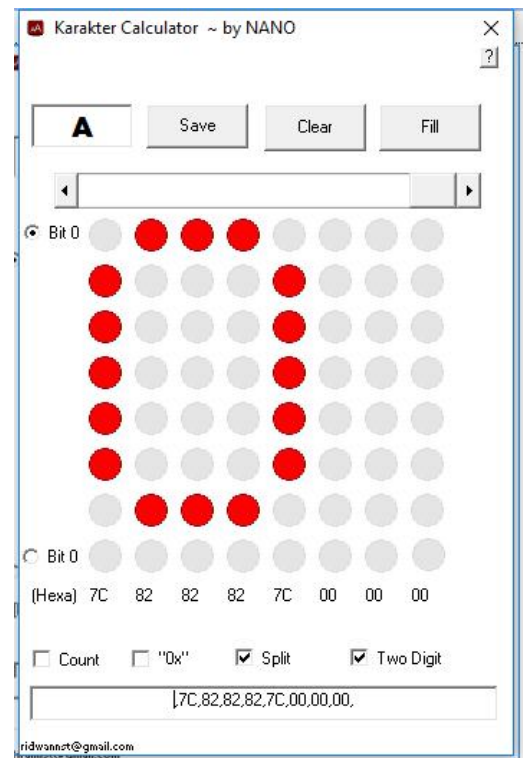
Konversi Bilangan Biner ke Hexadesimal

Setiap karakter yang akan ditampilkan pada display adalah kata yang bergerak dari kanan ke kiri. Untuk menampilkan sebuah karakter pada display, maka kolom-kolom tempat tampilnya karakter tersebut harus dihidupkan secara bergantian dalam waktu yang cepat, sehingga tampak bahwa sepertinya kolom-kolom tersebut hidup secara bersamaan. Gambar 9 menunjukkan karakter maker dalam membuat pola karakter.



Gambar 9. Aplikasi Karakter Maker

Pola karakter nol (0) yang dibentuk dengan menggunakan karakter maker langsung dikonversikan ke bilangan heksadesimal seperti ditunjukkan pada Gambar 9.



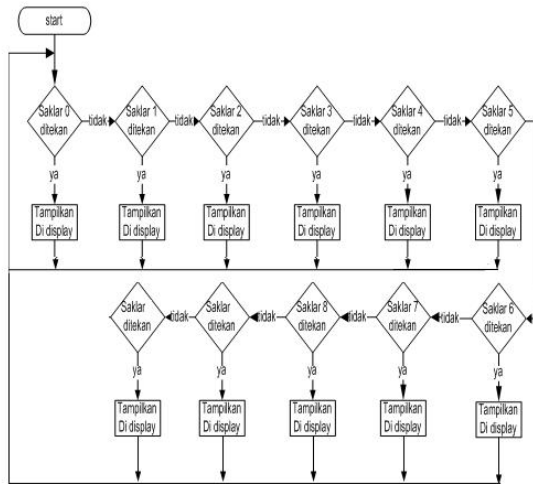
Gambar 10. Konversi Pola Karakter Nol ke Heksadesimal

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 ditunjukan 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.

Program Running Text Display LED Dot Matrix

Pada rangkaian *running text display LED dot matrix* sama seperti rangkaian tombol pilih terprogram, perbedaannya adalah pada rangkaian *running text dot matrix* tidak memerlukan keypad untuk mengganti kata pada *display LED dot matrix*. Kita hanya perlu mengetik pada program arduinonya yang bertuliskan “`initTeks("isi teks");`”.

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 (#).
2. Kata ALVIANDO ditampilkan pada display. Setelah kata ALVIANDO tampil pada display lalu diganti dengan kata ISTP pada display melalui program yang bertuliskan “`initTeks("ALVIANDO");`”.

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 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,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;
```

```
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,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,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,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++;

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':
```

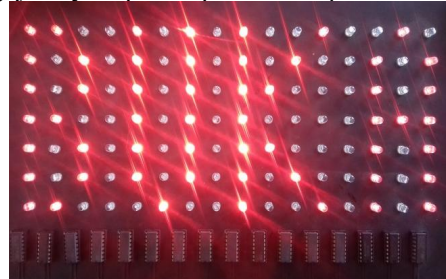
```
Disp.setByte(idx,0x82); idx++;
Disp.setByte(idx,0xFE); idx++;
Disp.setByte(idx,0x82); idx++;
Disp.setByte(idx,0x0); idx++;
Disp.setByte(idx,0x62); idx++;
Disp.setByte(idx,0x92); idx++;
Disp.setByte(idx,0x8C); idx++;
Disp.setByte(idx,0x0); idx++;
```

```
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

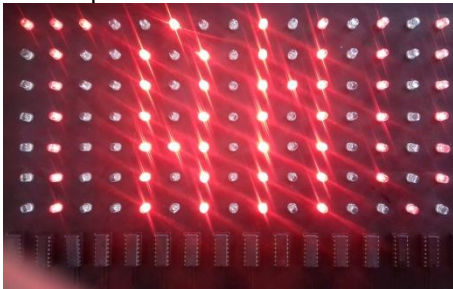
Adapun program untuk dapat menampilkan kata BUKA adalah sebagai berikut:

```
case 'I':
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.



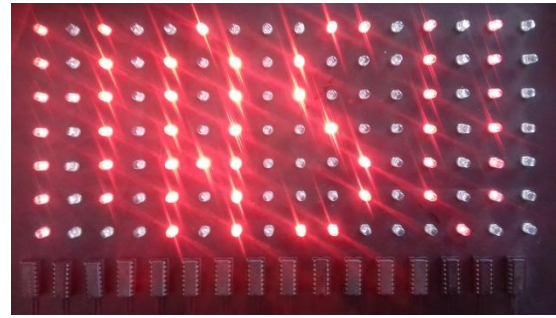
Gambar 13. Display Tampilkan Kata TAMU

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.



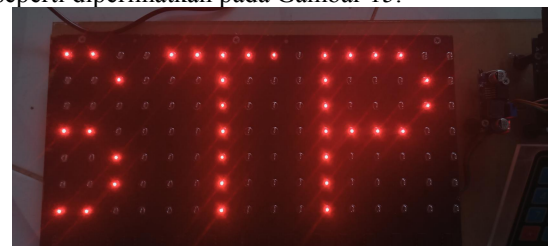
Gambar 14. Display Tampilkan Kata MASUK

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;
```

Pengujian Tulisan Berjalan Kata ISTP

Pengujian display dilakukan untuk mengetahui baik tidaknya display dan apakah sudah bisa menampilkan informasi sesuai dengan kata yang ditulis pada program arduino uno. hasil pengujiannya seperti diperlihatkan pada Gambar 15.



Gambar 15. display teks berjalan kata ISTP

Adapun program untuk dapat menampilkan tulisan berjalan ISTP adalah sebagai berikut:
#include <NanoTechDisp4094.h>

```
NanoTech Disp(2,A0,A2,A1);//jlh
board,data,clock,output enable
int maxNum;
int posX;
int speedTeks = 250;//mengatur kecepatan
pergerakan tulisan
void setup() {
  Disp.init();
  posX = Disp.getMaxBaris();
}
void loop() {
  if (maxNum > 0){
    maxNum--;
    Disp.moveLeft();
    Disp.show();
  } else {
    initTeks("ISTP");
    Disp.show();
  }
  delay(speedTeks);
}
void initTeks(String Teks){ // init teks
  maxNum = posX;
  int ln = Teks.length();
  for(int i=0;i<ln;i++){
    maxNum = initChar(maxNum,Teks[i]);
  }
}
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 'I':
      Disp.setByte(idx,0x42); idx++;
      Disp.setByte(idx,0xFE); idx++;
      Disp.setByte(idx,0x02); idx++;
      Disp.setByte(idx,0x0); idx++;
      break;
    case '2':
      Disp.setByte(idx,0x42); idx++;
      Disp.setByte(idx,0x86); idx++;
      Disp.setByte(idx,0x8A); idx++;
      Disp.setByte(idx,0x92); idx++;
      Disp.setByte(idx,0x62); idx++;
      Disp.setByte(idx,0x0); idx++;
      break;
    case '3':
      Disp.setByte(idx,0x44); idx++;
      Disp.setByte(idx,0x82); idx++;
      Disp.setByte(idx,0x92); idx++;
      Disp.setByte(idx,0x92); idx++;
      Disp.setByte(idx,0x6C); idx++;
```

```
Disp.setByte(idx,0x0); idx++;
      break;
    case '4':
      Disp.setByte(idx,0x18); idx++;
      Disp.setByte(idx,0x28); idx++;
      Disp.setByte(idx,0x48); idx++;
      Disp.setByte(idx,0x88); idx++;
      Disp.setByte(idx,0xFE); idx++;
      Disp.setByte(idx,0x0); idx++;
      break;
    case 'A':
      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++;
      break;
    case 'B':
      Disp.setByte(idx,0xFE); idx++;
      Disp.setByte(idx,0x92); idx++;
      Disp.setByte(idx,0x92); idx++;
      Disp.setByte(idx,0x92); idx++;
      Disp.setByte(idx,0x6C); idx++;
      Disp.setByte(idx,0x0); idx++;
      break;
    case 'C':
      Disp.setByte(idx,0x7C); idx++;
      Disp.setByte(idx,0x82); idx++;
      Disp.setByte(idx,0x82); idx++;
      Disp.setByte(idx,0x82); idx++;
      Disp.setByte(idx,0x44); idx++;
      Disp.setByte(idx,0x0); idx++;
      break;
    case 'D':
      Disp.setByte(idx,0xFE); idx++;
      Disp.setByte(idx,0x82); idx++;
      Disp.setByte(idx,0x82); idx++;
      Disp.setByte(idx,0x82); idx++;
      Disp.setByte(idx,0x7C); idx++;
      Disp.setByte(idx,0x0); idx++;
      break;
    case 'E':
      Disp.setByte(idx,0xFE); idx++;
      Disp.setByte(idx,0x92); idx++;
      Disp.setByte(idx,0x92); idx++;
      Disp.setByte(idx,0x92); idx++;
      Disp.setByte(idx,0x82); idx++;
      Disp.setByte(idx,0x0); idx++;
      break;
    case 'F':
      Disp.setByte(idx,0x7E); idx++;
      Disp.setByte(idx,0x90); idx++;
      Disp.setByte(idx,0x90); idx++;
      Disp.setByte(idx,0x80); idx++;
```

```
Disp.setByte(idx,0x40); idx++;
Disp.setByte(idx,0x0); idx++;
break;
case 'G':
    Disp.setByte(idx,0x7C); idx++;
    Disp.setByte(idx,0x82); idx++;
    Disp.setByte(idx,0x92); idx++;
    Disp.setByte(idx,0x92); idx++;
    Disp.setByte(idx,0x4C); idx++;
    Disp.setByte(idx,0x0); idx++;
    break;
case 'H':
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x10); idx++;
    Disp.setByte(idx,0x10); idx++;
    Disp.setByte(idx,0x10); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x0); idx++;
    break;
case 'I':
    Disp.setByte(idx,0x82); idx++;
    Disp.setByte(idx,0x82); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x82); idx++;
    Disp.setByte(idx,0x82); idx++;
    Disp.setByte(idx,0x0); idx++;
    break;
case 'J':
    Disp.setByte(idx,0x84); idx++;
    Disp.setByte(idx,0x86); idx++;
    Disp.setByte(idx,0x82); idx++;
    Disp.setByte(idx,0x82); idx++;
    Disp.setByte(idx,0xFC); idx++;
    Disp.setByte(idx,0x0); idx++;
    break;
case 'K':
    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++;
    break;
case 'L':
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x02); idx++;
    Disp.setByte(idx,0x02); idx++;
    Disp.setByte(idx,0x02); idx++;
    Disp.setByte(idx,0x02); idx++;
    Disp.setByte(idx,0x0); idx++;
    break;
case 'M':
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x20); idx++;
    Disp.setByte(idx,0x08); idx++;
    Disp.setByte(idx,0x20); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x0); idx++;
    break;
case 'N':
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x20); idx++;
    Disp.setByte(idx,0x10); idx++;
    Disp.setByte(idx,0x08); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x0); idx++;
    break;
case 'O':
    Disp.setByte(idx,0x7C); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x82); idx++;
    Disp.setByte(idx,0x82); idx++;
    Disp.setByte(idx,0x82); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x7C); idx++;
    Disp.setByte(idx,0x0); idx++;
    break;
case 'P':
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x90); idx++;
    Disp.setByte(idx,0x90); idx++;
    Disp.setByte(idx,0x90); idx++;
    Disp.setByte(idx,0x60); idx++;
    Disp.setByte(idx,0x0); idx++;
    break;
case 'Q':
    Disp.setByte(idx,0x7C); idx++;
    Disp.setByte(idx,0x82); idx++;
    Disp.setByte(idx,0x82); idx++;
    Disp.setByte(idx,0x86); idx++;
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x0); idx++;
    break;
case 'R':
    Disp.setByte(idx,0xFE); idx++;
    Disp.setByte(idx,0x90); idx++;
    Disp.setByte(idx,0x98); idx++;
    Disp.setByte(idx,0x94); idx++;
    Disp.setByte(idx,0x62); idx++;
    Disp.setByte(idx,0x0); idx++;
    break;
case 'S':
    Disp.setByte(idx,0x64); idx++;
    Disp.setByte(idx,0x92); idx++;
    Disp.setByte(idx,0x92); idx++;
    Disp.setByte(idx,0x92); idx++;
    Disp.setByte(idx,0x4C); idx++;
    Disp.setByte(idx,0x0); idx++;
    break;
case 'T':
```

```
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;
case 'U':
Disp.setByte(idx,0xFC); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0xFC); idx++;
Disp.setByte(idx,0x0); idx++;
break;
case 'V':
Disp.setByte(idx,0xF8); idx++;
Disp.setByte(idx,0x04); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0x04); idx++;
Disp.setByte(idx,0xF8); idx++;
Disp.setByte(idx,0x0); idx++;
break;
case 'W':
Disp.setByte(idx,0xFC); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0x04); idx++;
Disp.setByte(idx,0x18); idx++;
Disp.setByte(idx,0x04); idx++;
Disp.setByte(idx,0x02); idx++;
Disp.setByte(idx,0xFC); idx++;
Disp.setByte(idx,0x0); idx++;
break;
case 'X':
Disp.setByte(idx,0xC6); idx++;
Disp.setByte(idx,0x28); idx++;
Disp.setByte(idx,0x10); idx++;
Disp.setByte(idx,0x28); idx++;
Disp.setByte(idx,0xC6); idx++;
Disp.setByte(idx,0x0); idx++;
break;
case 'Z':
Disp.setByte(idx,0x86); idx++;
Disp.setByte(idx,0x8A); idx++;
Disp.setByte(idx,0x92); idx++;
Disp.setByte(idx,0xA2); idx++;
Disp.setByte(idx,0xC2); idx++;
Disp.setByte(idx,0x0); 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. Display LED yang dirangkai dalam bentuk dot matriks dapat menampilkan tulisan berjalan kata ALVIANDO dan ISTP.
3. Mikrokontroler arduino uno menerima data 8 bit dari tombol keypad dot matriks dan menampilkan informasi pada display LED sesuai dengan tombol yang dipilih yaitu tombol 1 untuk kata BUKA, tombol 2 untuk kata TAMU, tombol 3 untuk kata MASUK, tombol 4 untuk kata RAPAT, tombol 5 untuk kata TUTUP, tombol 6 untuk kata TENANG, tombol 7 untuk kata KELUAR, tombol 8 untuk kata SOLAT, tombol 9 untuk kata KULIAH, tombol 0 untuk kata MAKAN, tombol A untuk kata TUNGGU, tombol B untuk kata ISTIRAHAT.
4. Kata informasi yang ditampilkan display dapat diganti melalui program yang diupload ke memori mikrokontroler arduino uno.
5. Program cobol yang digunakan pada display LED dot matriks berbasis mikrokontroler arduino uno dirancang melalui laptop/komputer dengan menggunakan aplikasi Arduino IDE.
6. Display LED dot matriks berbasis mikrokontroler arduino uno menggunakan bahan keypad 4x4, arduino uno, ic 4094, adaptor, dan converter DC to DC.

Saran

1. Alat yang dirancang bangun oleh penulis, 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* dan *keyboard* komputer

DAFTAR PUSTAKA

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

<https://indone5ia.wordpress.com/2011/09/02/sekilas-mengenai-konverter-dc-dc/>

<https://wikielektronika.com/adaptor-adalah/>

<https://egsean.com/cara-membuat-sistem-minimum-mikrokontroler-atmega-328p/>

<https://elektrologi.iptek.web.id/rangkaian-programmer-pada-arduino-uno-dan-nano/>

<http://kursuselektronikaku.blogspot.com/2017/11/cara-mengakses-ic-4094-arduino.html>